

# A Benchmarking Algorithm to Determine the Sequence of Stable Data Gathering Trees for Wireless Mobile Sensor Networks

Natarajan Meghanathan

Associate Professor, Department of Computer Science, Jackson State University

Jackson, MS, USA

E-mail: natarajan.meghanathan@jsums.edu

Philip Mumford

Senior Electronics Engineer, Sensors Directorate, Air Force Research Lab/RYWC

Wright-Patterson Air Force Base, OH, USA

**Keywords:** stability, data gathering tree, mobile sensor networks, tree lifetime, network lifetime, coverage loss time, simulations

**Received:** September 15, 2012

*We propose a benchmarking algorithm to determine the sequence of longest-living stable data gathering trees for wireless mobile sensor networks whose topology changes dynamically with time due to the random movement of the sensor nodes. Referred to as the Max.Stability-DG algorithm, the algorithm assumes the availability of complete knowledge of future topology changes and is based on the following greedy principle coupled with the idea of graph intersections: Whenever a new data gathering tree is required at time instant  $t$  corresponding to a round of data aggregation, choose the longest-living data gathering tree from time  $t$ . The above strategy is repeated for subsequent rounds over the duration of the lifetime of the sensor network to obtain the sequence of longest-living stable data gathering trees spanning all the live sensor nodes in the network such that the number of tree discoveries is the global minimum. Thus, the number of tree discoveries incurred with the Max.Stability-DG algorithm will serve as the lower bound for the number of discoveries for any network-wide communication topology (like spanning trees and connected dominating sets) determined through any other algorithm for data gathering in mobile sensor networks under identical operating conditions. In addition to theoretically proving the correctness of the Max.Stability-DG algorithm, we also conduct exhaustive simulations to evaluate the performance of the Max.Stability-DG trees and compare to that of the minimum-distance spanning tree based data gathering trees with respect to metrics such as tree lifetime, delay per round, node lifetime, network lifetime and coverage loss time, under both sufficient-energy and energy-constrained scenarios.*

*Povzetek: Predstavljen je izvorni referenčni algoritem za iskanje najbolj obstojnih dreves v brezžičnih senzorskih omrežjih.*

## 1 Introduction

A wireless sensor network is a network of several smart sensor nodes that can gather data about the ambient environment as well as intelligently process them before propagating to a control center called the sink, which is typically located far away from the field being monitored and used to remotely administer the sensor network. Even though widely used for data gathering in several real-time applications, wireless sensor networks are mostly deployed for static environments, wherein the mobility of the sensor nodes, the users and the monitored phenomenon are all totally ignored. A wireless mobile sensor network (WMSN) is the next logical evolutionary step for sensor networks in which mobility needs to be handled in all its forms. With the widespread growth of embedded systems and ubiquitous computing technologies, a mobile sensor network could be envisioned as a homogeneous or heterogeneous network of sensor-equipped computers, mobile phones and

vehicles, generally referred to as nodes (having one or more sensors like a camera sensor, microphone, GPS sensor, etc) [10]. The nodes of a WMSN often move in an arbitrary fashion, independent of each other. Some of the applications [9] of WMSNs could be traffic monitoring, route planning, civil infrastructure monitoring (say, attaching vibration sensors to cars and monitoring the conditions of roads/pot holes), geo-imaging, mobile target tracking [33] and etc. WMSNs can be used to monitor and collect data over a much larger geographical area with less number of sensor nodes compared to static sensor networks. With mobility, the entire area could be covered with fewer sensors/nodes over a period of time

Like their static counterparts, the mobile sensor nodes are likely to be constrained with limited battery charge, memory and processing capability as well as operate under a limited transmission range. Two sensor

nodes that are outside the transmission range of each other cannot communicate directly. The bandwidth of a WMSN is also expected to be as constrained as that of a static sensor network. Due to all of the above resource and operating constraints, it will not be a viable solution to require every sensor node to directly transmit their data to the sink over a longer distance. Also, if several signals are transmitted at the same time over a longer distance, it could lead to lot of interference and collisions. Thus, there is a need for employing energy-efficient data gathering algorithms that can effectively combine the data collected at these sensor nodes and send only the aggregated data (that is a representative of the entire network) to the sink.

Over the past few years, the sensor network research community has proposed a number of data gathering algorithms to effectively combine the data collected at these sensor nodes through a properly constructed communication topology and send only the aggregated data (that is a representative of the entire network) to the sink. However, a majority of these data gathering algorithms are meant for static sensor networks (i.e., static sensor nodes) with either a static (e.g., [8][12]) or mobile (e.g., [26][29]) sink. Tree-based data gathering is considered to be the most energy-efficient [16] in terms of the number of link transmissions; however, almost all of the tree-based data gathering algorithms have been proposed for static sensor networks without taking the mobility of the sensor nodes into consideration. In the presence of node mobility, the network topology changes dynamically with time – leading to frequent tree reconfigurations. Thus, mobility brings in an extra dimension of constraint to a WMSN and we need algorithms that can determine stable long-living data gathering trees that do not require frequent reconfigurations. To the best of our knowledge, we have not come across any work on stable data gathering trees for mobile sensor networks.

In this research, we address the issue of finding a sequence of longest-living stable data gathering trees for mobile sensor networks such that the number of tree discoveries is the global minimum. We present a simple but powerful polynomial-time greedy algorithm, referred to as the Max.Stability-DG algorithm, to determine the sequence of longest-living stable data gathering trees. Given the complete knowledge of the future topology changes, the Max.Stability-DG algorithm operates based on the following greedy principle: Whenever a data gathering tree is required at time instant  $t$ , choose the longest-living data gathering tree from  $t$ . The above strategy is repeated over the duration of the data gathering session. The sequence of such longest-living data gathering trees is called the *Stable-Mobile-DG-Tree*. The worst-case run-time complexity of the Max.Stability-DG tree algorithm is  $O(n^2T \log n)$  and  $O(n^3T \log n)$  when operated under sufficient-energy and energy-constrained scenarios respectively, where  $n$  is the number of nodes in the network and  $T$  is the total number of rounds of data gathering;  $O(n^2 \log n)$  is the worst-case run-time complexity of the minimum-weight spanning tree algorithm (we use Prim's algorithm [5]) used to

determine the underlying spanning trees from which the data gathering trees are derived.

The rest of the paper is organized as follows: Section 2 presents the system model and the terminology used in this research as well as a high-level overview of the working of the proposed Max.Stability-DG algorithm and highlights its key contributions. Section 3 presents related work on data gathering in mobile sensor networks. Section 4 describes in detail the working of the Max.Stability-DG algorithm, analyzes its run-time complexity for both sufficient-energy and energy-constrained scenarios, and provides a formal proof of correctness of the algorithm. We also present an algorithm to determine a minimum-distance spanning tree based data gathering (MST-DG) tree that has been observed (in previous research) to be the most energy-efficient approach [16] for data gathering in static sensor networks. We also present an example to illustrate the working of the Max.Stability-DG and MST-DG algorithms. Section 5 presents an exhaustive simulation study evaluating the performance of the Max.Stability-DG trees under diverse conditions of network dynamicity (node mobility and number of static nodes), network density (transmission range) and energy level at the nodes (sufficient-energy and energy-constrained scenarios). The performance metrics evaluated are the tree lifetime, delay per round, energy lost per node, energy lost per round, fairness of node usage, node lifetime, network lifetime, coverage loss time and fraction of coverage loss. We compare the performance of the Max.Stability-DG trees with that of the MST-DG trees. Section 6 presents the conclusions along with a summary of the simulation results. Section 7 discusses future work. For the rest of the paper, the terms 'node' and 'vertex', 'edge' and 'link', 'data aggregation' and 'data gathering' will be used interchangeably. They mean the same.

## 2 System model, terminology and overview

### 2.1 System model

The system model adopted in this research is as follows: (i) Each sensor node is assumed to operate with an identical and fixed *transmission range*. (ii) For the purpose of calculating the coverage loss, we also use the *sensing range* of a sensor node, considered in this research, as half the transmission range of the node. Basically, a sensor node can monitor and collect data at locations within the radius of its sensing range and transmit them to nodes within the radius of its transmission range. It has been proven in the literature [32] that the transmission range per node has to be at least twice the sensing range of the nodes to ensure that coverage implies connectivity. (iii) A data gathering tree is obtained by conducting a Breadth First Search (BFS) [5] of the spanning tree, starting from a root node that serves as the leader node for the tree. (iv) The leader node of a data gathering tree remains the same until the

tree exists and is randomly chosen each time a new tree needs to be determined. (v) Data gathering proceeds in rounds. During a round of data gathering, data gets aggregated starting from the leaf nodes of the tree and propagates all the way to the leader node. An intermediate node in the tree collects the aggregated data from its immediate child nodes and further aggregates with its own data before forwarding to its immediate parent node in the tree.

## 2.2 Terminology

We use the notions of static graphs and mobile graphs (adapted from [7]) to capture the sequence of topological changes in the network and determine a stable data gathering tree that spans over several time instants. A **static graph** is a snapshot of the network at any particular time instant and is modeled as a unit disk graph [11] wherein there exists a link between any two nodes if and only if the physical distance between the two end nodes of the link is less than or equal to the transmission range. The weight of an edge on a static graph is the Euclidean distance between the two end nodes of the edge. The Euclidean distance for a link  $i - j$  between two nodes  $i$  and  $j$ , currently at  $(X_i, Y_i)$  and  $(X_j, Y_j)$  is given by:  $\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$ .

A **mobile graph**  $G(i, j)$ , where  $1 \leq i \leq j \leq T$ , where  $T$  is the total number of rounds of the data gathering session corresponding to the network lifetime, is defined as  $G_i \cap G_{i+1} \cap \dots \cap G_j$ . Thus, a mobile graph is a logical graph that captures the presence or absence of edges in the individual static graphs. In this research work, we sample the network topology periodically for every round of data gathering to obtain the sequence of static graphs. The weight of an edge in the mobile graph  $G(i, j)$  is the geometric mean of the weights of the edge in the individual static graphs spanning  $G_i, \dots, G_j$ . Since there exist an edge in a mobile graph if and only if the edge exists in the corresponding individual static graphs, the geometric mean of these Euclidean distances would also be within the transmission range of the two end nodes for the entire duration spanned by the mobile graph. Note that at any time, a mobile graph includes only **live sensor nodes**, nodes that have positive available energy.

A **static spanning tree** is a minimum-weight spanning tree determined on a static graph. Since we use the Euclidean distance between the constituent nodes of an edge as the link weight, the minimum-weight spanning tree determined on a static graph will be a minimum-distance spanning tree for which the sum of the edge weights will be the minimum. A **static data gathering tree** is a rooted form of its corresponding static spanning tree with the root node being the leader node chosen for the round corresponding to the time instant represented by the static spanning tree. A **mobile spanning tree** is a minimum-weight spanning tree determined on a mobile graph whose edge weights are the geometric mean of the corresponding edge weights in the constituent static graphs. A **mobile data gathering tree** is a rooted mobile spanning tree whose root node is

the leader node chosen at the beginning time instant of the corresponding mobile graph. The leader node of a mobile data gathering tree remains the same until the mobile graph gets disconnected due to node mobility or a node failure occurs, whichever happens first.

## 2.3 Overview of the maximum stability based data gathering algorithm and key contributions

A high-level overview of the working of the Max.Stability-DG algorithm is as follows: To determine a stable data gathering at time instant  $t_i$  ( $1 \leq i \leq T$ , the total number of rounds of the data gathering session), we determine the mobile graph  $G(i, j)$ , where  $i \leq j$  such that there exists a spanning tree of the sensor nodes in  $G(i, j)$  and not in  $G(i, j+1)$ . We transform such a longest-living spanning tree existing in each of the static graphs of the mobile graph  $G(i, j)$  to a data gathering tree by simply running a breadth-first search (BFS) algorithm [5] starting from an arbitrarily chosen root node (also called the leader node). The data gathering tree rooted at the leader node is used for all the rounds from time instants  $t_i$  to  $t_j$ , which is considered as the lifetime of the spanning tree. The above procedure is repeated until the end of the data gathering session or the network lifetime, as appropriate. Any spanning tree algorithm can be used to determine the spanning tree on the mobile graph. In this research, we use the Prim's  $O(n^2 \log n)$  algorithm to determine a minimum-weight spanning tree on the mobile graph (of  $n$  nodes) whose edge weights are modeled as the geometric mean of the edges in the constituent static graphs.

A **key assumption** in this research is that the entire sequence of network topology changes is known beforehand at the time of running the Max.Stability-DG algorithm. This is required to generate the mobile graph spanning several static graphs, each representing snapshots of the network topology at time instants corresponding to successive rounds of data gathering, on which a stable long-living data gathering tree will be determined. The above assumption may not be practical for distributed systems of sensor networks. However, note that our goal in this research is not to develop a distributed algorithm for data gathering; but, to develop a **benchmarking algorithm** that can give us the sequence of long-living data gathering trees (over the duration of the data gathering session) whose **lifetime will be the upper bound for the data gathering trees** obtained using any other algorithm developed for this problem in the area of mobile sensor networks. The sequence of such stable longest-living data gathering trees determined using the Max.Stability-DG algorithm will involve the minimum number of discoveries. Thus, the number of data gathering tree discoveries incurred with the Max.Stability-DG algorithm **will form the lower bound for the number of data gathering tree discoveries** incurred with any other algorithm for mobile sensor networks.

**The proposed algorithm is very generic in nature** and it can be used to determine a sequence of stable

communication topologies of any type (for example, a connected dominating set [17], a chain [12], a cluster [8] etc) as long as there is an underlying algorithm to determine that communication topology on a given graph. In this research, we focus only on spanning tree as the communication topology for data gathering. Moreover, since the Max.Stability-DG trees are spanning-tree based and a spanning tree exists in a network if and only if the underlying network is connected, the stability of network-wide communication topologies (like a connected dominating set [17] that spans all the nodes) determined by any algorithm can be evaluated by comparing their lifetime with that obtained for the Max.Stability-DG trees under identical operating conditions. Henceforth, the relative stability of data gathering trees or any network-wide communication topology for mobile sensor networks, determined from any existing or newly proposed algorithm (very few of which is currently available in the literature, as reviewed in Section 3), either centralized or distributed, can be evaluated in comparison with the mobile data gathering trees obtained by running the Max.Stability-DG algorithm, developed in this research, under the same conditions in which the existing or prospective data gathering algorithm is run.

### 3 Related work on data gathering in wireless mobile sensor networks

The research on mobile sensor networks started with the deployment of mobile sink nodes on a network of static sensor nodes. A common approach of data gathering in such environments is to employ a mobile data collecting agent (e.g., [30][31][34]) that goes around the network in the shortest possible path towards the location from which the desired data is perceived to originate. In [35], the authors propose a distributed algorithm to optimize both coverage control and mobile collection using a Bayesian occupancy grid mapping technique that recursively updates the locations of potential data sources. In [18], the authors propose a 2-layer architecture comprising of mobile sinks and static sensor nodes for large scale wireless sensor networks. The top layer is a mobile ad hoc network of resource-rich sink nodes while the bottom-layer is a network of static resource-constrained sensor nodes. Each sink node is assigned a particular region to monitor and collect data. A sink node moves to the vicinity of the sensor nodes (within a few hops) to collect data. The collected data is exchanged with peer mobile sinks. A prototype implementation of the same is available in [19].

Very few topology-based data gathering algorithms have been proposed for mobile sensor networks where the sensor nodes actually move. Among these, most of the work is focused around the use of clusters wherein researchers have tried to extend the classical LEACH (Low Energy Adaptive Clustering Hierarchy) [8] algorithm for dynamically changing network topologies. Variants of LEACH for WMSNs that have been proposed in the literature include those that take into consideration the available energy level [2] and the

mobility-level [23] of the nodes to decide on the choice of cluster heads; stability of the links between a regular node and its cluster head [6]; as well as set up a panel of cluster heads to facilitate cluster reconfiguration in the presence of node mobility [24]. In [13], the authors propose a distributed cluster-head based algorithm in which cluster-heads are elected based on node IDs (0 to  $C-1$ ,  $C$  to  $2C-1$  ..., to operate with  $C$  clusters at a time) or node locations (nodes that are closest to certain landmarks within a WMSN serve as the cluster-heads). In [15], the authors investigate the use of a directed acyclic graph as the underlying communication topology of a sensor network field, modeled according to the theory of thermal fields, to form propagation paths such that the temperature of the nodes on the path increases as data progresses towards the sink, which is considered to be the warmest.

The only tree-based data gathering algorithm we have come across for WMSNs is a shortest path-based spanning tree algorithm [25] wherein each sensor node is constrained to have at most a certain number of child nodes. Based on the results from the literature of mobile ad hoc networks (e.g., [20][21]), minimum hop shortest paths and trees in mobile network topologies are quite unstable and need to be frequently reconfigured. We could not find any other related work on tree-based data gathering for wireless mobile sensor networks.

## 4 Data gathering algorithms based on maximum stability and minimum-distance spanning trees

### 4.1 Maximum stability spanning tree-based data gathering (max.stability-DG) algorithm

The Max.Stability-DG algorithm is based on a greedy look-ahead principle and the intersection strategy of static graphs. When a mobile data gathering tree is required at a sampling time instant  $t_i$ , the strategy is to find a mobile graph  $G(i, j) = G_i \cap G_{i+1} \cap \dots \cap G_j$  such that there exists a spanning tree in  $G(i, j)$  and no spanning tree exists in  $G(i, j+1) = G_i \cap G_{i+1} \cap \dots \cap G_{j+1}$ . We find such an epoch  $t_i, \dots, t_j$  as follows: Once a mobile graph  $G(i, j)$  is constructed with the edges assigned the weights corresponding to the geometric mean of the weights in the constituent static graphs  $G_i, G_{i+1}, \dots, G_j$ , we run the Prim's minimum-weight spanning tree algorithm on the mobile graph  $G(i, j)$ . If  $G(i, j)$  is connected, we will be able to find a spanning tree in it. We repeat the above procedure until we reach a mobile graph  $G(i, j+1)$  in which no spanning tree exists and there existed a spanning tree in  $G(i, j)$ . It implies that a spanning tree basically existed in each of the static graphs  $G_i, G_{i+1}, \dots, G_j$  and we refer to it as the mobile spanning tree for the time instants  $t_i, \dots, t_j$ . To obtain the corresponding mobile data gathering tree, we choose an arbitrary root node for this mobile spanning tree and run the Breadth First Search (BFS) algorithm on it starting

from the root node. The direction of the edges in the spanning tree and the parent-child relationships are set as we traverse its vertices using BFS. The resulting mobile data gathering tree with the chosen root node (as the leader node) is used for every round of data gathering spanning time instants  $t_i, \dots, t_j$ . We then set  $i = j+1$  and repeat the above procedure to find a mobile spanning tree and its corresponding mobile data gathering tree that exists for the maximum amount of time since  $t_{j+1}$ . A sequence of such maximum lifetime (i.e., longest-living) mobile data gathering trees over the timescale  $T$  corresponding to the number of rounds of a data gathering session is referred to as the **Stable Mobile Data Gathering Tree**. Figure 1 presents the pseudo code of the Max.Stability-DG algorithm that takes as input the sequence of static graphs spanning the entire duration of the data gathering session.

**Input:** Sequence of static graphs  $G_1, G_2, \dots, G_T$ ; Total number of rounds of the data gathering session –  $T$

**Output:** *Stable-Mobile-DG-Tree*

**Auxiliary Variables:**  $i, j$

**Initialization:**  $i = 1; j = 1; \text{Stable-Mobile-DG-Tree} = \Phi$

**Begin** *Max.Stability-DG Algorithm*

```

1  while ( $i \leq T$ ) do
2    Find a mobile graph  $G(i, j) = G_i \cap G_{i+1} \cap \dots \cap G_j$  such that there exists at least one spanning tree in  $G(i, j)$  and {no spanning tree exists in  $G(i, j+1)$  or  $j = T$ }
3    Mobile-Spanning-Tree( $i, j$ ) = Prim's Alg ( $G(i, j)$ )
4    Root( $i, j$ ) = Choose a node randomly in  $G(i, j)$ 
5    Mobile-DG-Tree( $i, j$ ) = Breadth First Search (Mobile-Spanning-Tree( $i, j$ ), Root( $i, j$ ))
6    Stable-Mobile-DG-Tree = Stable-Mobile-DG-Tree  $\cup$  { Mobile-DG-Tree( $i, j$ ) }
7    for each time instant  $t_k \in \{t_i, t_{i+1}, \dots, t_j\}$  do
        Use the Mobile-DG-Tree( $i, j$ ) in  $t_k$ 
8      if node failure occurs at  $t_k$  then
           $j = k - 1$ 
          break
      end if
    end for
9     $i = j + 1$ 
10 end while
11 return Stable-Mobile-DG-Tree

```

**End** *Max.Stability-DG Algorithm*

Figure 1: Pseudo Code for the Maximum Stability-based Data Gathering Tree Algorithm

While operating the algorithm under energy-constrained scenarios, one or more sensor nodes may die due to exhaustion of battery charge even though the underlying spanning tree may topologically exist. For

example, if we have determined a data gathering tree spanning across time instants  $t_i$  to  $t_j$  using the above approach, and we come across a time instant  $t_k$  ( $i \leq k \leq j$ ) at which a node in the tree fails, we simply restart the Max.Stability-DG algorithm starting from time instant  $t_k$  considering only the live sensor nodes (i.e., the sensor nodes that have positive available energy) and determine the longest-living data gathering tree that spans all the live sensor nodes since  $t_k$ . The pseudo code of the Max.Stability-DG algorithm in Figure 1 handles node failures, when run under energy-constrained scenarios, through the **if** block segment in statement 8. If all nodes have sufficient-energy and there are no node failures, the algorithm does not execute statement 8.

## 4.2 Run-time complexity Analysis of the max.stability-DG algorithm

To expand a mobile graph  $G(i, j) = G_i \cap G_{i+1} \cap \dots \cap G_j$  to  $G(i, j+1)$ , all we had to do is to check whether each of the edges in the mobile graph  $G(i, j)$  existed at time instant  $t_{j+1}$ . This can be done in  $O(n^2)$  time on a mobile graph of  $n$  nodes, as there can be at most  $O(n^2)$  edges on a graph of  $n$  vertices. The overall complexity of the Max.Stability-DG algorithm is the sum of the time complexity to construct the mobile graphs, the time complexity to run the spanning tree algorithm on these mobile graphs and the time complexity to transform these spanning trees to data gathering trees using BFS.

**Sufficient-energy Scenarios:** When the network operates under sufficient-energy scenarios (i.e., no node failures), for a data gathering session comprising of  $T$  rounds, we will have to construct  $T$  mobile graphs, resulting in a time complexity of  $O(n^2 T)$  to construct the mobile graphs. On each of these  $T$  mobile graphs, we will have to run a spanning tree algorithm. If we use the  $O(n^2 \log n)$  Prim's algorithm to construct a spanning tree, the complexity to run the spanning tree algorithm on the  $T$  mobile graphs becomes  $O(n^2 \log n * T)$ . A spanning tree on  $n$  vertices has  $n-1$  edges. The time-complexity of running BFS on a spanning tree of  $n$  vertices with  $n-1$  edges is  $O(n)$  [5]. To run BFS on the  $O(T)$  spanning trees, we incur  $O(nT)$  time. Thus, the overall complexity of the Max.Stability-DG algorithm under sufficient-energy scenarios is  $O(n^2 T) + O(n^2 T \log n) + O(nT) = O(n^2 T \log n)$ .

**Energy-Constrained Scenarios:** There can be at most  $n-1$  node failures (on an  $n$  node network) that trigger the execution of statement 8 in the pseudo code of Figure 1 for the Max.Stability-DG algorithm. A node failure occurring at time instant  $t_k$  ( $i \leq k \leq j$ ), while using a mobile data gathering tree that has been determined on a mobile graph for time instants  $t_i, \dots, t_j$ , would require us to construct a mobile graph starting from  $t_k$  and the number of mobile graphs that we have to construct and run the spanning tree algorithm increases by  $j-k+1$ . At the worst case, if there are  $n-1$  node failures, the number of mobile graphs that we have to construct and run the spanning tree algorithm increases by  $(T-1) + (T-2) +$

$(T - (n-1)) = (n-1)T - [1 + 2 + \dots + (n-1)] = O(nT) + O(n^2)$ . Under the sufficient-energy scenarios, we had to construct  $T$  mobile graphs and run the spanning tree algorithm on each of them. In the energy-constrained scenarios, we will have to construct at most  $T + O(nT) + O(n^2)$  mobile graphs and run the spanning tree algorithm on each of them. The number of rounds of data gathering is generally far greater than the number of nodes in the network. For example, in our simulations, we use a value of  $T = 4000$  rounds (4 rounds per second, for 1000 seconds) and  $n = 100$  nodes. Thus, since  $n \ll T$ , we can say that  $n^2 \ll nT$ . Therefore, a total of  $T + O(nT) + O(n^2) = T + O(nT) = O(nT)$  mobile graphs are constructed. The time complexity to construct these mobile graphs is  $O(n^2 * nT) = O(n^3T)$ . We run the  $O(n^2 \log n)$  Prim's spanning tree algorithm on the  $O(nT)$  mobile graphs, resulting in a time-complexity of  $O(n^3 T \log n)$  to determine the spanning trees. The time-complexity of running the  $O(n)$ -BFS algorithm on the  $O(nT)$  spanning trees is  $O(n^2T)$ . Thus, the overall time-complexity of the Max.Stability-DG algorithm under the energy-constrained scenarios is  $O(n^3T) + O(n^3 T \log n) + O(n^2T) = O(n^3 T \log n)$ .

### 4.3 Minimum-distance spanning tree based data gathering algorithm

In Section 5 (Simulations), we compare the performance of the Max.Stability-DG trees with that of the minimum-distance spanning tree based data gathering (MST-DG) trees. The sequence of MST-DG trees for the duration of the data gathering session is generated as follows: If a MST-DG tree is not known for a particular round, we run the Prim's minimum-weight spanning tree algorithm on the static graph representing the snapshot of the network topology generated at the time instant corresponding to the round. Since the weights of the edges in a static graph represent the physical Euclidean distance between the constituent end nodes of the edges, the Prim's algorithm will return the minimum-distance spanning tree on the static graph. We then choose an arbitrary root node and run the Breadth First Search (BFS) algorithm starting from this node. The MST-DG tree is the rooted form of the minimum-distance spanning tree with the chosen root node as the leader node. We continue to use the MST-DG tree as long as it exists. The leader node of the MST-DG tree remains the same until the tree breaks due to node mobility or node failures. When the MST-DG tree ceases to exist for a round, we repeat the above procedure. This way, we generate a sequence of MST-DG trees, referred to as the **MST Mobile Data Gathering Tree**. The MST-DG algorithm emulates the general strategy (referred to as Least Overhead Routing Approach, LORA [1]) of routing protocols and data gathering algorithms for ad hoc networks and sensor networks. That is, the algorithm chooses a data gathering tree that appears to be the best at the current time instant and continues to use it as long as it exists. In a recent work [16], the authors have observed the minimum-distance spanning tree-based data gathering trees to be the most energy-efficient communication topology for data gathering in static sensor networks.

To be fair to the Max.Stability-DG algorithm that is proposed and evaluated in this research, the MST-DG algorithm is also run in a centralized fashion with the assumption that the entire static graph information is available at the beginning of each round. The time-complexity of generating the sequence of MST-DG trees on a network of  $n$  nodes for a total of  $T$  rounds for the data gathering session is  $O(n^2 T \log n)$  for both the sufficient-energy and energy-constrained scenarios. The time-complexity of the MST-DG algorithm remains the same for both the sufficient-energy and energy-constrained scenarios; because, we do not need to backtrack on the sequence of static graphs upon node failure and repeat the algorithm more than once on a static graph.

### 4.4 Example

We run both the Max.Stability-DG and MST-DG algorithms on the same sequence of static graphs  $G_1 G_2 G_3 G_4 G_5$  (shown in the first part of Figures 2 and 3), generated by sampling the network topology for every second. For simplicity in the representation, we do not use weights for the edges. In both Figures 2 and 3, one could assume that the spanning trees determined on the static graphs and mobile graphs at different instances of execution of the algorithms are the minimum-weight spanning trees on the corresponding graphs.

In Figure 2, we could find a connected mobile graph spanning  $G_1$ ,  $G_2$  and  $G_3$ ; and could not find a connected mobile graph from  $G_1$  through  $G_4$ . A spanning tree exists for a graph if and only if the graph is connected. We determine a spanning tree on  $G_1 \cap G_2 \cap G_3$  and derive a data gathering tree rooted at an arbitrarily selected node (node 3). This stable data gathering tree is to be used for the rounds corresponding to time instants of the static graphs  $G_1$ ,  $G_2$  and  $G_3$ . Similarly, we continue with the subsequent two static graphs and find a data gathering tree (with an arbitrary root node – node 6) that exists in both  $G_4$  and  $G_5$ . Thus, we require two tree discoveries for the sequence of static graphs  $G_1 G_2 G_3 G_4 G_5$ .

We apply the MST-DG algorithm on the same sequence of static graphs  $G_1 G_2 G_3 G_4 G_5$  (Figure 3). Note that the MST-DG algorithm chooses a spanning tree that appears to be the best at the time of needing a new data gathering tree. With this locally optimal strategy, we observe that we need to use a total of four data gathering trees (one for  $G_1$  and  $G_2$ ; and one each for  $G_3$ ,  $G_4$  and  $G_5$ ); thus, requiring four tree discoveries for the sequence of static graphs  $G_1 G_2 G_3 G_4 G_5$ . We observe similar trends on the lifetime of the MST-DG trees in the simulations. Such a behaviour is not just a characteristic of MST-DG trees. We conjecture that any non-stability based data gathering algorithm that does not take into consideration the mobility of the nodes and chooses a data gathering tree that appears to be locally optimal with respect to any other metric (like energy consumption, delay etc.) will end up determining data gathering trees that require to be frequently reconfigured in the presence of a dynamically changing topology, characteristic of mobile sensor networks.

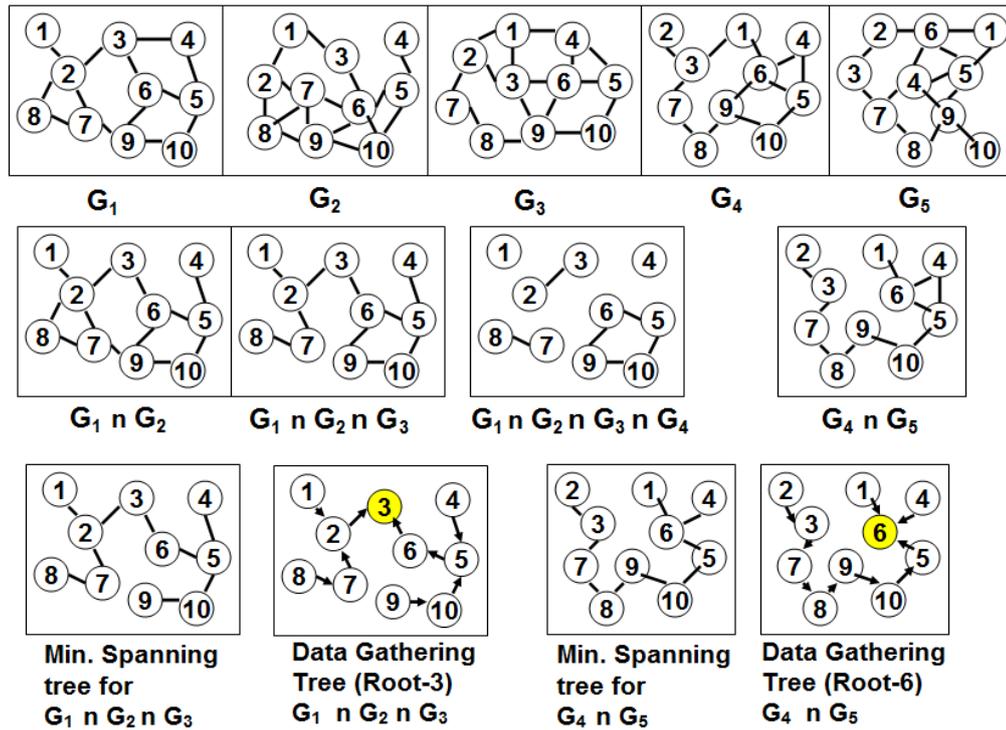


Figure 2: Example to Illustrate the Execution of the Maximum Stability Spanning Tree-based Data Gathering Tree Algorithm that uses the Globally Optimal Approach

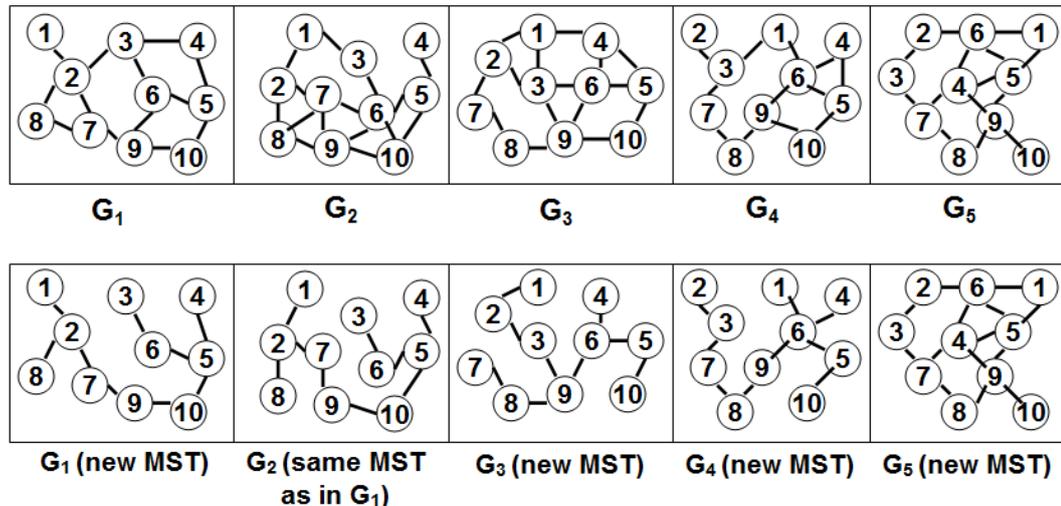


Figure 3: Example to Illustrate the Execution of the Minimum-distance Spanning Tree based Data Gathering Algorithm that uses the Locally Optimal Approach

#### 4.5 Proof of correctness of the maximum stability-based data gathering algorithm

In this section, we prove that the Max.Stability-DG algorithm does determine the sequence of long-living stable mobile data gathering trees such that the number of tree discoveries is the global minimum (i.e. optimum). We use the approach of Proof by Contradiction. Let  $m$  be the number of data gathering tree discoveries incurred using the Max.Stability-DG algorithm on a sequence of static graphs  $G_1 G_2 \dots G_T$ . Let there be another algorithm

(a hypothetical algorithm) that returns a sequence of mobile data gathering trees for the same sequence of static graphs such that the number of tree discoveries is  $n < m$ . If such an algorithm exists, then without loss of generality, there has to be one mobile data gathering tree, determined using this hypothetical algorithm, existing for the entire duration of a mobile graph  $G(p, s)$ ; whereas, the Max.Stability-DG algorithm had to have at least one data gathering tree transition in  $G(p, s)$ . However, there cannot be such a data gathering tree that spanned through the entire mobile graph  $G(p, s)$  and was not discovered by the Max.Stability-DG algorithm. Because, the Max.Stability-DG algorithm takes intersection of the

static graphs  $G_p \cap G_{p+1} \cap \dots \cap G_s$  and runs a spanning tree algorithm on the intersection graph  $G(p, s)$  – if at all a spanning tree existed in  $G(p, s)$ , then  $G(p, s)$  would have to be connected. If the Max.Stability-DG algorithm could not determine a spanning tree/data gathering tree for the mobile graph  $G(p, s)$ , it implies the mobile graph  $G(p, s)$  is not connected. It is not possible for any algorithm, including our hypothetical algorithm, to find a spanning tree/data gathering tree that covers all the vertices of a disconnected graph. Thus, the hypothetical algorithm would also have to have at least one tree transition in  $G(p, s)$ . The above proof holds good for any value of static graph indices  $p$  and  $s$ , where  $1 \leq p \leq s \leq T$ , and  $T$  is the total number of rounds corresponding to the duration of the data gathering session. Thus, the number of data gathering tree discoveries incurred with using the Max.Stability-DG algorithm is the global minimum.

Note that in the above proof, we have implicitly assumed that all the sensor nodes are alive for the entire duration of the data gathering session. In other words, we have proved that when operated under sufficient-energy scenarios, the Max.Stability-DG algorithm returns the stable sequence of data gathering trees such that the number of tree discoveries is the global minimum. It is not possible to theoretically prove the optimality of the Max.Stability-DG algorithm under energy-constrained scenarios. One can only validate the optimality of the lifetime of the Max.Stability-DG trees under energy-constrained scenarios through simulations, as we do in Section 5, wherein we observe the Max.Stability-DG trees to yield a significantly longer lifetime compared to the MST-DG trees under energy-constrained scenarios.

## 5 Simulations

In this section, we present an exhaustive simulation study on the performance of the Max.Stability-DG trees and compare them with that of the MST-DG trees under diverse conditions of network density and mobility. The simulations are conducted in a discrete-event simulator developed (in Java) by us exclusively for data gathering in mobile sensor networks. The MAC (medium access control) layer is assumed to be collision-free and considered an ideal channel without any interference. We opine that the use of any MAC-scheme proposed for energy-efficient and low-latency tree-based data gathering in sensor networks [14] can only be complementary to the performance of our benchmarking algorithm when implemented in a distributed context. Sensor nodes are assumed to be both TDMA (Time Division Multiple Access) and CDMA (Code Division Multiple Access)-enabled [28]. Every upstream node broadcasts a time schedule (for data gathering) to its immediate downstream nodes; a downstream node transmits its data to the upstream node according to this schedule. Such a TDMA-based communication between every upstream node and its immediate downstream child nodes can occur in parallel, with each upstream node using a unique CDMA code.

The network dimension is 100m x 100m. The number of nodes in the network is 100 and initially, the

nodes are uniform-randomly distributed throughout the network. The sink is located at (50, 300), outside the network field. For a given simulation run, the transmission range per sensor node is fixed and is the same across all nodes. The network density is varied by varying the transmission range per sensor node from 20m to 50m, in increments of 5m. For brevity, we only present results obtained for transmission ranges per node of 25m and 30m (representative of moderate density, with connectivity of 97% and above), and for 40m (representative of high density, with 100% connectivity).

Simulations are conducted for two kinds of energy scenarios: One scenario wherein each node is provided with abundant supply of energy (50 J per node) and there are no node failures due to exhaustion of battery charge; the simulations in these *sufficient-energy scenarios* are conducted for 1000 seconds. The second scenario is an *energy-constrained scenario* in which each node is supplied with limited initial energy (2 J per node) and the simulations are conducted until the network of live sensor nodes gets disconnected due to the failures of one or more nodes.

The energy consumption model used is a first order radio model [22] that has been also used in several of the well-known previous work (e.g., [8][12]) in the literature. According to this model, the energy expended by a radio to run the transmitter or receiver circuitry is  $E_{elec} = 50$  nJ/bit and  $\epsilon_{amp} = 100$  pJ/bit/m<sup>2</sup> for the transmitter amplifier. The radios are turned off when a node wants to avoid receiving unintended transmissions. The energy lost in transmitting a  $k$ -bit message over a distance  $d$  is given by:  $E_{TX}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2$ . The energy

lost to receive a  $k$ -bit message is:  $E_{RX}(k) = E_{elec} * k$ .

We conduct constant-bit rate data gathering at the rate of 4 rounds per second (one round for every 0.25 seconds). The size of the data packet is 2000 bits; the size of the control messages used for tree discoveries is assumed to be 400 bits. We assume that a tree discovery requires network-wide flooding of the 400-bit control messages such that each sensor node will broadcast the message exactly once in its neighborhood. As a result, each sensor node will lose energy to transmit the 400-bit message over its entire transmission range and receive the message from each of its neighbor nodes. In high density networks, the energy lost due to receipt of the redundant copies of the tree discovery control messages dominates the energy lost at a node for tree discovery. All of these mimic the energy loss observed for flooding-based tree discovery in ad hoc and sensor networks.

The node mobility model used is the well-known Random Waypoint mobility model [3] with the maximum node velocity being 3 m/s, 10 m/s and 20 m/s representing scenarios of low, moderate and high mobility respectively. According to this model, each node chooses a random target location to move with a velocity uniform-randomly chosen from  $[0, \dots, v_{max}]$ , and after moving to the chosen destination location, the node continues to move by randomly choosing another new location and a new velocity. Each node continues to

move like this, independent of the other nodes and also independent of its mobility history, until the end of the simulation. For a given  $v_{max}$  value, we also vary the dynamicity of the network by conducting the simulations with a variable number of static nodes (out of the 100 nodes) in the network. The values for the number of static nodes used are: 0 (all nodes are mobile), 20, 50 and 80.

## 5.1 Performance metrics

We generated 200 mobility profiles of the network for a total duration of 6000 seconds, for every combination of the maximum node velocity and the number of static nodes. Every data point in the results presented in Figures 5 through 25 is averaged over these 200 mobility profiles. The tree lifetime and delay per round are measured for both the sufficient-energy and energy-constrained (appropriately prefixed as ‘EC’ next to the names of the data gathering trees) scenarios. The trio of the energy consumption metrics – energy lost per round, energy lost per node and fairness of node usage are all measured only for the sufficient-energy scenarios in order to accurately capture the impact of the topological structure, network dynamicity and the stability of the data gathering trees on energy consumption. The node and network lifetimes as well as the fraction of coverage loss and coverage loss time are measured only for the energy-constrained scenarios.

The performance metrics measured in the simulations are:

- (i) **Tree Lifetime** – the duration for which a data gathering tree existed, averaged over the entire simulation time period.
- (ii) **Delay per Round** – measured in terms of the number of time slots needed per round of data aggregation at the intermediate nodes, all the way to the leader node of the data gathering tree, averaged across all the rounds of the simulation. A brief description of the algorithm used to compute the delay per round is given in Section 5.2 along with an illustration in Figure 4.
- (iii) **Energy Lost per Round** – measured as the (sum of the energy lost due to the transmission and reception of data across all the links of a data gathering tree used for each round, the energy lost in transmitting the aggregated data from the leader node to the sink for each round plus the energy lost due to the network-wide flooding-based discovery of all the data gathering trees) divided by (the number of rounds of data gathering conducted on the network).
- (iv) **Energy Lost per Node** – measured as the (sum of the energy lost at each node in the network due to transmission and reception of the data packets depending on their position in the data gathering trees used for the different rounds plus the energy lost due to broadcast transmission and reception of control messages in the

neighborhood) divided by (the number of nodes in the network).

- (v) **Fairness of Node Usage** – measured as the standard deviation (SD) of energy lost per node. The SD of energy lost per node should be ideally zero for maximum fairness of node usage. However, due to the stochastic nature of the network, nodes are not equally used. The lower the value for the SD of energy lost per node, the larger the fairness of node usage, and vice-versa.
- (vi) **Node Lifetime** – measured as the time of first node failure due to exhaustion of battery charge.
- (vii) **Network Lifetime** – measured as the time of disconnection of the network of live sensor nodes (i.e., the sensor nodes that have positive available battery charge), while the network would have stayed connected if all the nodes were alive at that time instant. So, before confirming whether an observed time instant is the network lifetime (at which the network of live sensor nodes is noticed to be disconnected), we test for connectivity of the underlying network if all the sensor nodes were alive.

We obtain the distribution of node failures as follows: The probability for ‘x’ number of node failures (x from ranging from 1 to 100 as we have a total of 100 nodes in our network for all the simulations) for a given combination of the operating conditions (transmission range per node, maximum node velocity and number of static nodes) is measured as the number of mobility profile files that reported x number of node failures divided by 200, which is the total number of mobility profiles used for every combination of maximum node velocity and number of static nodes. Similarly, we keep track of the time at which ‘x’ (x ranging from 1 to 100) number of node failures occurred in each of the 200 mobility profiles for a given combination of operating conditions and the values for the time of node failures reported in Figures 17, 18 and 19 are an average of these data collected over all the mobility profile files. We discuss the results for the distribution of the time and probability of node failures along with the discussion on node lifetime and network lifetime in Section 5.7.

(viii) **Fraction of Coverage Loss and Coverage Loss Time**: If  $f$  is denoted as ‘Fraction of Coverage Loss’ (ranging from 0.01 to 1.0, measured in increments of 0.01), the coverage loss time is the time at which any  $f$  randomly chosen locations (X, Y co-ordinates) among 100 locations in the network is not within the sensing range of any node (explained in more detail below). Since the number of node failures increases monotonically with time and network coverage depends on the number of live nodes, our assumption in the calculations for network

coverage loss is that the fraction of coverage loss increases monotonically with time. We keep track of the largest fraction of coverage loss the network has incurred so far, and at the beginning of each round we check whether the network has incurred the next largest fraction of coverage loss, referred to as the *target fraction of coverage loss*. The first time instant during which we observe the network to have incurred the target coverage loss is recorded as the coverage loss time for the particular fraction of coverage loss, and from then on, we increment the target coverage loss by 0.01 and keep testing for the first occurrence of the new target fraction of coverage loss in the subsequent rounds. We repeat the above procedure until the network lifetime is encountered for the simulation with the individual data gathering algorithm.

At the beginning of each round, we check for network coverage as follows: We choose 100 random locations in the network and find out whether each of these locations is within the sensing range of at least one sensor node. We count the number of locations that are not within the sensing range of any node. If the fraction of the number of locations (actual number of locations that are not covered / total number of locations considered, which is 100) not within the sensing range of any node equals the target fraction of coverage loss, we record the time instant for that particular round of data gathering as the coverage loss time corresponding to the target fraction of coverage loss. We then increment the target fraction of coverage loss by 0.01 and repeat the above procedure to determine the coverage loss time corresponding to the new incremented value of the target fraction of coverage loss.

Each coverage loss time data point reported for particular fractions of coverage loss in Figures 23, 24 and 25 are the average values of the coverage loss times observed when the individual data gathering tree algorithms are run with the mobility profile files corresponding to a particular condition of network dynamicity (max. node velocity and number of static nodes) and transmission range per node. The probability for a particular fraction of coverage loss is computed as the ratio of the number of mobility profile files in which the corresponding fraction of coverage loss was observed divided by the total number of mobility profile files (200 mobility profile files for each operating condition).

### 5.2 Algorithm to compute the delay per round of data gathering

The delay incurred at a node is measured in terms of the number of time slots it takes to gather data from all of its

immediate child nodes. The delay for the data gathering tree is one plus the delay incurred at the leader node (root node). We assume that it takes one time slot per child node to transfer data to its immediate predecessor node in the tree. However, a node cannot transfer the aggregated data to its parent node until it receives the data from its own child nodes. The delay calculations start from the bottom of the data gathering tree. The delay incurred at a leaf node is 0. To calculate the delay incurred at an intermediate node  $u$ ,  $Delay(u)$ , located at a particular level in the data gathering tree, we maintain a sorted list,  $Child-Nodes(u)$ , of the delay associated with each of its immediate child nodes and use a temporary running variable  $Temp-Delay(u)$ , initialized to zero, to explore the sorted list of the delays at the child nodes. For every child node  $v \in Child-Nodes(u)$ ,  $Temp-Delay(u) = \text{Maximum} [Temp-Delay(u) + 1, Delay(v) + 1]$ , as we assume it takes one time slot for a child node to transfer its aggregated data to its immediate predecessor node in the tree. The delay associated with an intermediate node  $u$ ,  $Delay(u)$ , is the final value of the  $Temp-Delay(u)$  variable, after we iterate through the sorted list of the delays associated with the list  $Child-Nodes(u)$ . The above procedure is repeated at all the intermediate nodes, from levels one less than the *Height* of the tree all the way to zero (i.e., the root node). We illustrate the working of the above explained procedure for delay computation on a data gathering tree through an example presented in Figure 4. The integer inside a circle indicates the node ID and the integer outside a circle indicates the delay for data aggregation at the node.

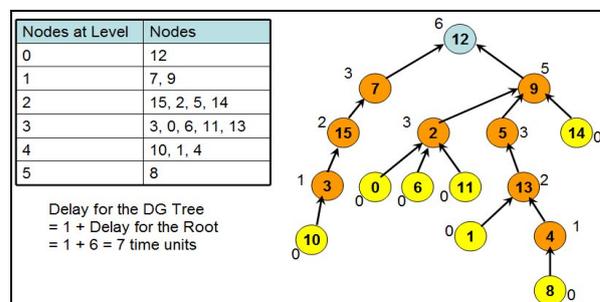


Figure 4: Example to Illustrate the Calculation of Delay per Round of Data Gathering.

### 5.3 Tree lifetime

Among the three key operating parameters (maximum node velocity, number of static nodes and transmission range per node) of the simulations, we observe the stability of the data gathering trees to be highly influenced by the maximum node velocity ( $v_{max}$ ) of the nodes. When operated under sufficient-energy scenarios, for a fixed number of static nodes and transmission range per node, we observe the lifetime incurred for both the Max.Stability-DG trees and MST-DG trees to proportionally decrease with a corresponding increase in the  $v_{max}$  values from 3 m/s to 10 m/s and further to 20 m/s. In the energy-constrained scenarios, even though a data gathering tree may topologically exist, the tree would require reconfiguration if one / more

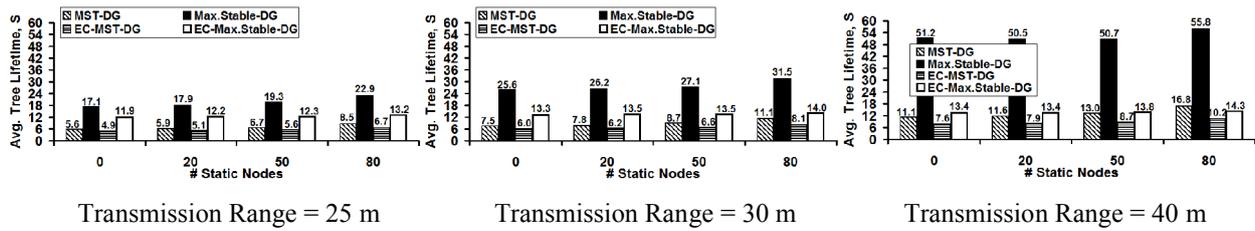


Figure 5: Average Tree Lifetime (Low Node Mobility:  $v_{max} = 3$  m/s).

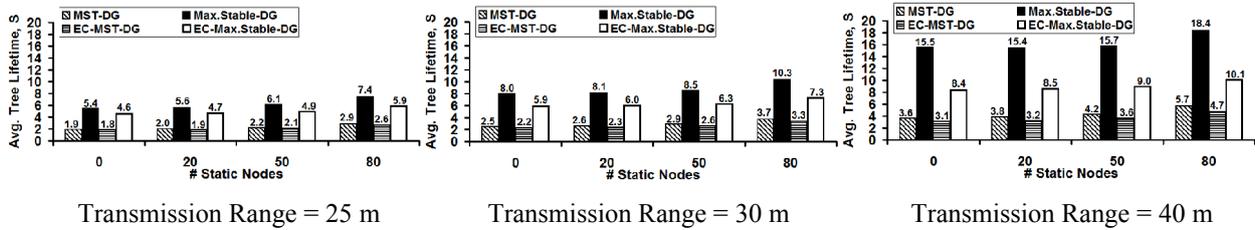


Figure 6: Average Tree Lifetime (Moderate Node Mobility:  $v_{max} = 10$  m/s).

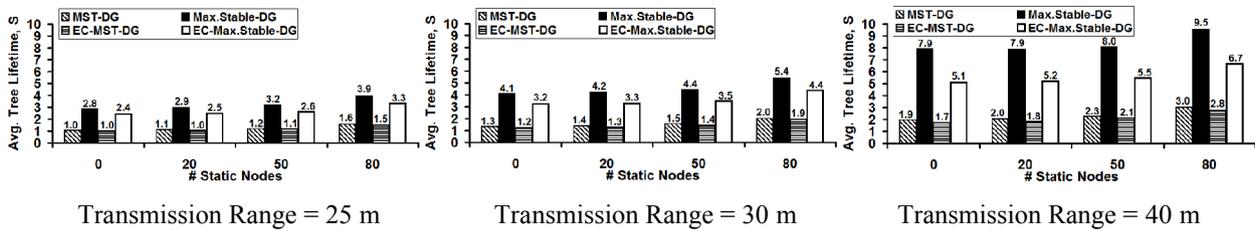


Figure 7: Average Tree Lifetime (High Node Mobility:  $v_{max} = 20$  m/s).

nodes in the tree fail due to exhaustion of battery charge. Since a tree also needs to be reconfigured due to node mobility, the lifetime of the data gathering trees observed for energy-constrained scenarios is always less than or equal to that observed for sufficient-energy scenarios. In the case of both the Max.Stability-DG and MST-DG trees, for a fixed transmission range and # static nodes, we observe the largest difference between the tree lifetimes for the sufficient-energy scenarios vis-à-vis the energy-constrained scenarios to occur when the network is operated under low node mobility conditions ( $v_{max} = 3$  m/s). This could be attributed to the significantly longer lifetime observed for the data gathering trees at low node mobility conditions when operated with sufficient-energy for the nodes.

In low mobility scenarios (refer Figure 5), we also observe the difference in the tree lifetimes under sufficient-energy vs. energy-constrained scenarios to increase with increase in the transmission range per node. At higher transmission ranges, the links are more stable as nodes of a link have relatively higher freedom to move around (compared to operating at low and moderate transmission ranges) and still remain as neighbors. Hence, the data gathering trees are bound to be the most stable at low node mobility and larger transmission ranges per node. At these conditions – under sufficient-energy scenarios, we observe the Max.Stability-DG trees to sustain a lifetime that is larger than that of the MST-DG trees by a factor of about 3 to 4.5. However, under energy-constrained scenarios, when operated at low node velocity and larger transmission range per node, the

Max.Stability-DG trees are only 50-75% more stable than that of the MST-DG Trees, even though the absolute magnitude of the tree lifetime incurred with both the trees is the maximum compared to the other operating conditions. On the contrary, larger differences between the lifetimes of the Max.Stability-DG trees and MST-DG trees under energy-constrained scenarios are observed when operated at moderate transmission ranges per node and the difference increases with increase in node mobility. This could be attributed to the significant energy savings sustained by the Max.Stability-DG algorithm with respect to tree discoveries under moderate and high node mobility levels (refer Figures 6 and 7). On the other hand, the MST-DG trees are quite unstable with increase in node mobility, resulting in frequent flooding-based tree discoveries that consume significant node energy. As a result, nodes on a Max.Stability-DG tree exist for a relatively much longer time compared to that of the MST-DG trees, contributing to the increasing difference in the lifetime of the two data gathering trees in energy-constrained scenarios when operated under moderate and high levels of node mobility.

With regards to the impact of the transmission range per node, the difference in the lifetime of the Max.Stability-DG trees and the MST-DG trees increases with increase in the transmission range per node, for a given level of node mobility. For a fixed  $v_{max}$  value, the lifetime of the Max.Stability-DG trees increases by a factor of 3 and above as we increase the transmission range from 25m to 40m; whereas the lifetime of the MST-DG trees increases only at most by a factor of 2.

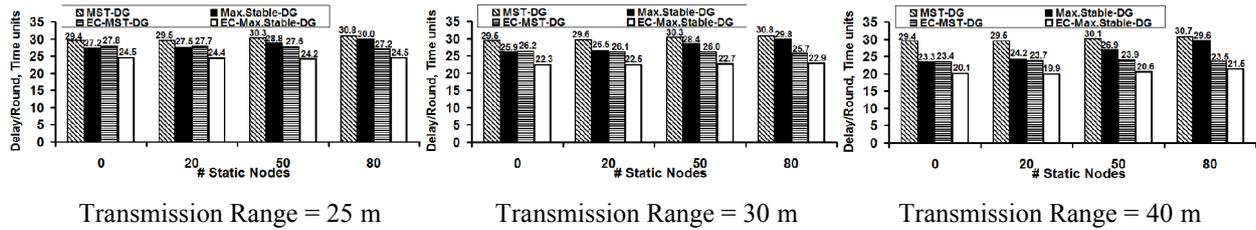


Figure 8: Average Delay per Round (Low Node Mobility:  $v_{max} = 3$  m/s).

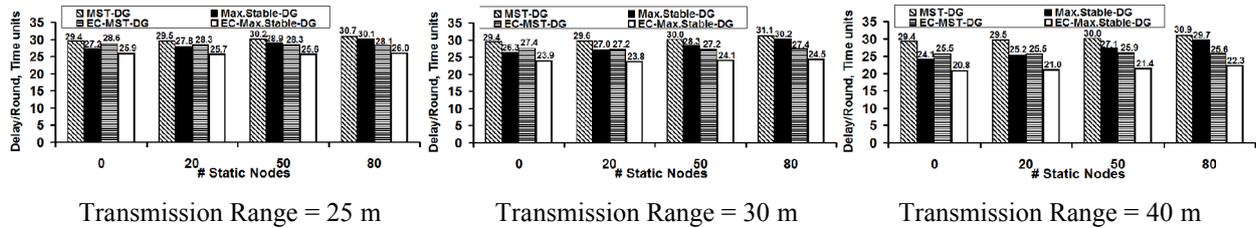


Figure 9: Average Delay per Round (Moderate Node Mobility:  $v_{max} = 10$  m/s).

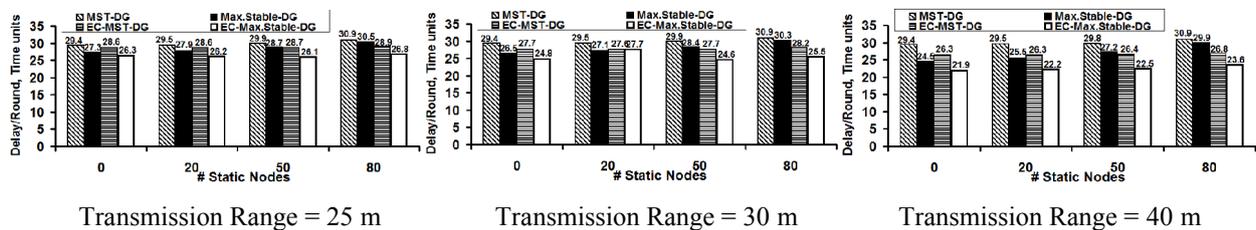


Figure 10: Average Delay per Round (High Node Mobility:  $v_{max} = 20$  m/s).

This could be again attributed to the optimal usage of the availability of stable links (facilitated by the larger transmission ranges per node) by the Max.Stability-DG algorithm through its look-ahead and graph intersection approach. However, as is the bane of the algorithms based on the local optimum approach, the MST-DG trees are formed with relatively less stable links even when operated with higher transmission ranges per node.

With regards to the impact of the number of static nodes, we observe that for both the sufficient-energy and energy-constrained scenarios, the lifetime of both the Max.Stability-DG trees and the MST-DG trees increases by about 50% when the number of static nodes is increased from 0 to 80 nodes. There is not much of a significant increase (only at most about 10-15% increase) in the lifetime of both the data gathering trees when we run the network with 20 and 50 static nodes instead of 0 nodes. This vindicates the impact of node mobility on the stability of the data gathering trees. Even if half of the nodes in the network are operated static, we observe the data gathering trees to have about the same vulnerability for a link failure vis-à-vis operating the network with all mobile nodes.

### 5.4 Delay per round

A minimum-distance based spanning tree tends to have relatively fewer leaf nodes, and as a result more nodes are likely to end up as intermediate nodes – leading to a much larger depth for the MST-DG trees. The MST-DG tree is also observed to be more unbalanced with respect to the distribution of the number of children per

intermediate node as well as the distribution of the leaf nodes at different levels. Not all leaf nodes are located at the bottommost level of the tree. Due to all these structural complexities, the MST-DG trees have been observed to incur a much larger delay per round of data gathering. On the other hand, the Max.Stability-DG trees have been observed to be more shallow (i.e., lower depth) with more leaf nodes and the distribution of the number of child nodes per intermediate node is relatively more balanced. All of these factors contribute to a much lower delay per round of data gathering.

We observe the Max.Stability-DG trees to incur a lower delay per round of data gathering compared to that of the MST-DG trees under all operating conditions (the difference is as large as 25%). The delay per round is not much affected by the dynamicity of the network and is more impacted by the topological structure of the two spanning trees. We observe a relatively lower delay per round, especially for the MST-DG trees, at energy-constrained scenarios vis-à-vis sufficient-energy scenarios due to the decrease in the number of nodes and slightly better distribution of nodes when fewer in number. The delay per round for the Max.Stability-DG trees is influenced more by the transmission range per node in energy-constrained scenarios and by the number of static nodes in sufficient-energy scenarios. For a given transmission range per node and # static nodes, variations in the value of the maximum node velocity have the least impact on the delay per round for both the data gathering trees.

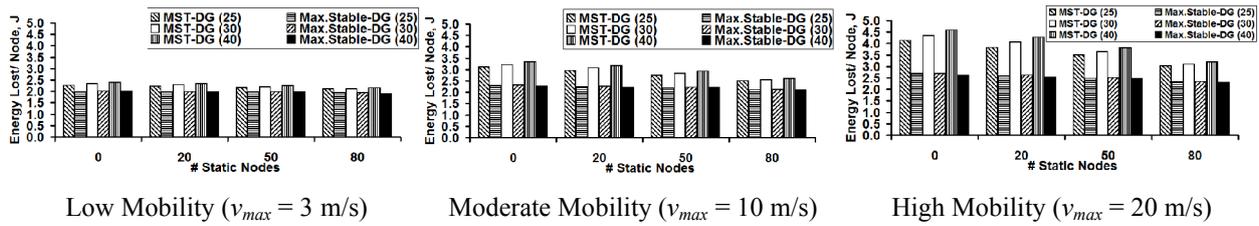


Figure 11: Average Energy Lost per Node.

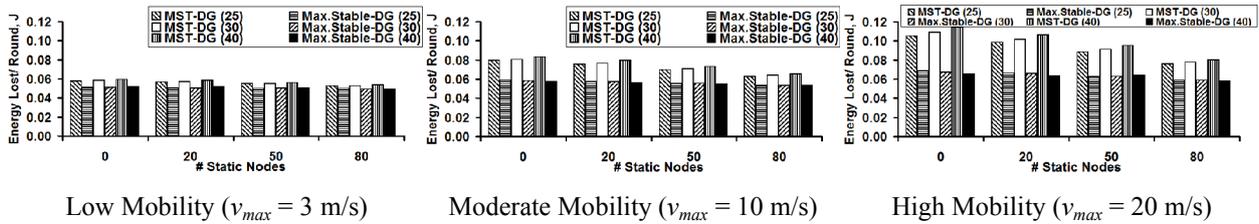


Figure 12: Average Energy Lost per Round.

With node failures, the number of nodes in the data gathering trees decreases and this has a positive impact on the delay per round of data gathering. The decrease in the delay per round under energy-constrained scenarios ranges from 10-40%, with the larger percentage decrease in delay observed when the data gathering algorithms are operated with transmission range per node of 40m (observed for all node mobility levels) under energy-constrained scenarios compared to sufficient-energy scenarios. As a result, for a given energy scenario, the difference in the delay per round of the Max.Stability-DG trees and the MST-DG trees decreases with increase in the transmission range per node for a given node mobility.

For a given level of node mobility and transmission range per node, the delay per round for a data gathering tree, under both the sufficient-energy and energy-constrained scenarios, increases with increase in the number of static nodes. The increase is more predominantly observed for the Max.Stability-DG trees when operated at 80 static nodes under sufficient-energy scenarios. The Max.Stability-DG trees incurred about 10-25% lower delay than that of the MST-DG trees when all nodes are mobile. As the number of static nodes increases, the delay per round incurred with the Max.Stability-DG trees converges to that of the MST-DG trees, especially when operated under sufficient-energy scenarios. The relatively lower delay per round for the Max.Stability-DG trees under energy-constrained scenarios can be attributed to the decrease in the number of nodes coupled with the shallow topological structure of the data gathering tree.

### 5.5 Energy consumption

The energy lost per node and energy lost per round for the Max.Stability-DG trees are lower than that of the MST-DG trees for all the operating conditions. In this section, we combine the discussion for energy lost per node and energy lost per round as similar trends are observed for the two performance metrics (see Figures 11 and 12) with respect to the two data gathering trees

under the different operating conditions. The maximum node velocity and the number of static nodes have been observed to have a significant impact on the energy lost per node and per round for the two data gathering trees. The lifetime of the MST-DG trees decreases significantly with increase in the maximum node velocity, requiring frequent network-wide flooding that consumes the energy level at the nodes. On the other hand, the Max.Stability-DG algorithm incurs significantly fewer tree discoveries and hence sustains a lower overall energy loss. The energy lost per node (and energy lost per round) for the MST-DG trees are about 10-20%, 15-45% and 30-75% more than that incurred for the Max.Stability-DG trees at conditions of low, moderate and high node mobility respectively. For a fixed  $v_{max}$  and transmission range per node, the difference in the energy lost per node (and energy lost per round) for the two data gathering trees decreases with increase in the number of static nodes. This is attributed to the relatively lower number of tree discoveries needed in the presence of static nodes and overall, the nodes incur a lower energy loss for the duration of the data gathering session. The decrease in the energy lost per node (and energy lost per round) with increase in the number of static nodes is significantly observed as we increase the maximum node velocity (about 5-10% decrease at  $v_{max} = 3$  m/s and about 25%-35% decrease at  $v_{max} = 20$  m/s).

For a given level of node mobility and number of static nodes, we observe the Max.Stability-DG trees to sustain almost similar values for the energy lost per node (and energy lost per round) as we increase the transmission range per node from 25m to 40m. Even though one can imagine that the nodes will lose more energy when operated under higher transmission range, the potential energy savings (obtained due to the reduced number of flooding-based tree discoveries attributed at larger transmission ranges per node) equally compensates for the increase in the transmission energy loss. Even the MST-DG trees have been observed to incur only about 5-10% increase in the energy lost per node (and energy lost per round) when operated at range

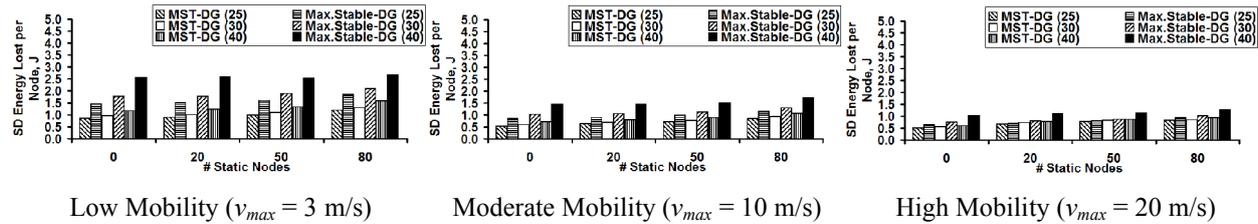


Figure 13: Fairness of Node Usage: Standard Deviation of Energy Lost per Node.

per node of 40m compared to 25m per node. Thus, among the three operating parameters – the transmission energy per node has the least impact on the energy lost per node and energy lost per round for the two data gathering trees.

## 5.6 Fairness of node usage

Ideally, all nodes are to be equally used. However, due to node mobility, node distribution and the varied roles for the nodes (leader node, intermediate node and leaf node) that change at different instants of the data gathering session depending on the topological stability of the data gathering tree used, the energy consumption across nodes is not uniform. As a result, certain nodes fail prematurely ahead of others.

We observe the Max.Stability-DG trees to be unfair with respect to node usage. This could be attributed to the repeated use of certain nodes (the intermediate nodes and the leader node) of the stable data gathering tree for a longer time. This has a profound effect on the node lifetime (the time of first node failure) as observed in Section 5.7. However, the energy savings brought by reduced number of tree discoveries significantly compensates for the unfairness in node usage, leading to a larger network lifetime (the time the network of live nodes gets disconnected), as also observed in Section 5.7. Since the MST-DG trees are relatively more frequently reconfigured, we observe these data gathering trees to incur a lower standard deviation of energy lost per node under all simulation conditions. This plays a significant role in the MST-DG trees sustaining a relatively longer node lifetime (the time of first node failure). However, the relatively more equal, but higher energy lost per node for the MST-DG trees expedites node failures beyond the first node failure, eventually leading to a lower network lifetime than those incurred with the Max.Stability-DG trees.

For a given transmission range per node and number of static nodes, we observe the standard deviation of energy usage at the nodes to decrease with increase in maximum node velocity. Node mobility triggers more frequent tree reconfigurations and as a result, the chances of the role of the energy-consuming intermediate nodes and leader node gets rotated gets high, leading to increased fairness in node usage. On the other hand, the standard deviation of energy lost per node increases with increase in the transmission range per node (attributed to the increased stability at larger transmission ranges) as well as with increase in the number of static nodes (again attributed to the stability of the data gathering trees).

## 5.7 Node lifetime and network lifetime

We observe a tradeoff between node lifetime and network lifetime for maximum stability vs. minimum-distance spanning tree based data gathering in mobile sensor networks. The MST-DG trees incur larger node lifetimes (the time of first node failure) for all the 48 operating combinations of maximum node velocity, number of static nodes and transmission range per node. The Max.Stability-DG trees incur larger network lifetime for most of the operating conditions. The lower node lifetime incurred with the Max.Stability-DG trees is attributed to the continued use of stable data gathering trees for a longer time and that too without changing the leader node. It would involve too much of message complexity and energy consumption to have the sensor nodes coordinate among themselves to choose a leader node for every round. Hence, we choose the leader node for a data gathering tree at the time of discovering it and let the leader node remain the same for the duration of the tree (i.e., until the tree fails). The same argument applies for the continued use of the intermediate nodes that receive aggregate data from one or more child nodes and transmit them to an upstream node in the tree. Due to the unfairness in node usage resulting from the overuse of certain nodes as intermediate nodes and leader node, the Max.Stability-DG trees have been observed to yield a lower node lifetime, especially under operating conditions (like low and moderate node mobility with moderate and larger transmission range per node) that facilitate greater stability. The node lifetime incurred with the Max.Stability-DG trees increases significantly with increase in the maximum node velocity, especially when operated in moderate transmission ranges per node.

The node lifetime incurred for the MST-DG trees can be larger than that of the Max.Stability-DG trees by as large as 400% at low and moderate levels of node mobility and by as large as 135% at higher levels of node mobility. For a given level of node mobility, the difference in the node lifetimes incurred for the MST-DG trees and Max.Stability-DG trees increases with increase in the transmission range per node (for a fixed number of static nodes) and either remain the same or slightly increase with increase in the number of static nodes (for a fixed transmission range per node). The MST-DG trees too suffer a decrease in node lifetime with increase in transmission range per node; but, at a lower scale – due to the relative instability of the trees. At larger transmission ranges per node, the data gathering trees are bound to be more stable, and the negative impact of this

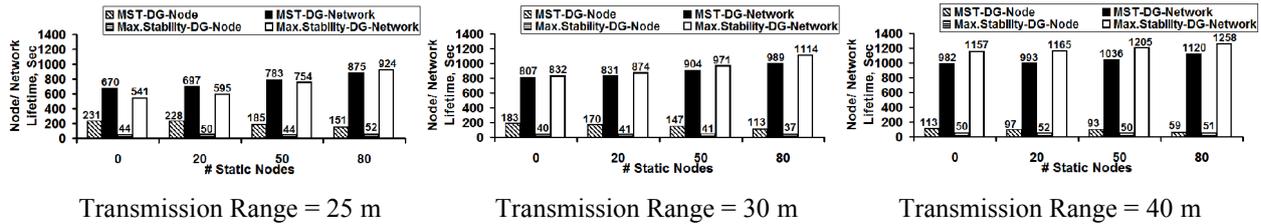


Figure 14: Average Node and Network Lifetime (Low Node Mobility:  $v_{max} = 3$  m/s).

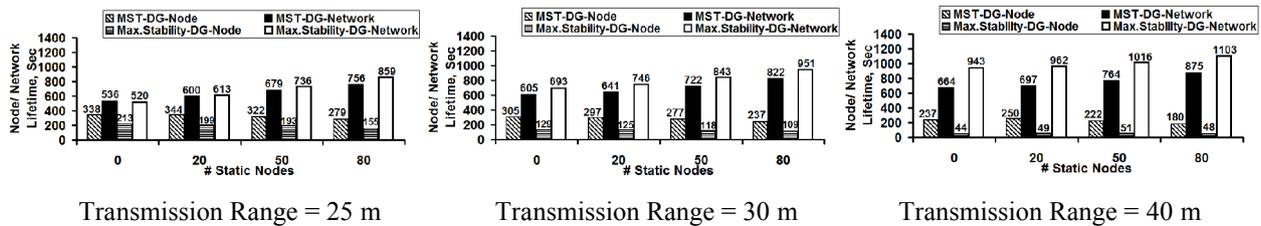


Figure 15: Average Node and Network Lifetime (Moderate Node Mobility:  $v_{max} = 10$  m/s).

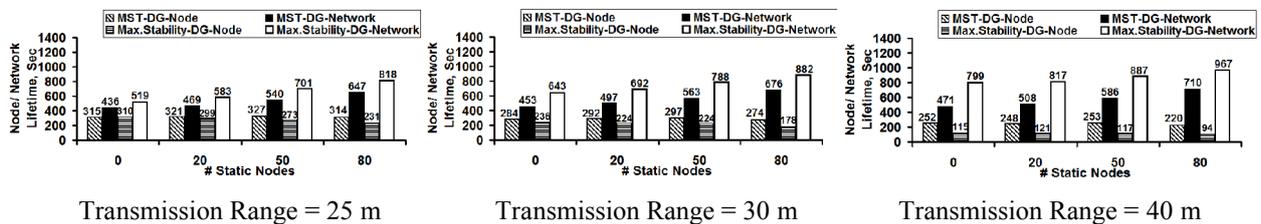


Figure 16: Average Node and Network Lifetime (High Node Mobility:  $v_{max} = 20$  m/s).

on node lifetime is significantly felt in the case of the Max.Stability-DG trees. For a given transmission range per node, the negative impact associated with the use of static nodes on node lifetime is increasingly observed at  $v_{max}$  values of 3 m/s and 10 m/s. At  $v_{max} = 20$  m/s, since the network topology changes dynamically, even the use of 80 static nodes is not likely to overuse certain nodes and result in their premature failures. The node lifetime incurred with MST-DG trees is more impacted with the use of static nodes at low node mobility scenarios (Figure 14) and the node lifetime incurred with the Max.Stability-DG trees is more impacted with the use of static nodes at moderate and higher node mobility scenarios (Figures 15 and 16).

The Max.Stability-DG trees compensate for the premature failures of certain nodes by incurring a lower energy loss per round and energy loss per node due to lower tree discoveries and shorter tree height with more even distribution of the number of child nodes per intermediate node. As the dynamicity of the network increases, the data gathering trees become less stable, and this helps to rotate the roles of the intermediate nodes and leader node among the nodes to increase the fairness of node usage. All of these save significantly more energy at the remaining nodes that withstand the initial set of failures. As a result, we observe the Max.Stability-DG trees to observe a significantly longer network lifetime compared to that of the MST-DG trees. The difference in the network lifetime incurred for the Max.Stability-DG trees and that of the MST-DG trees increases with increase in the maximum node velocity and transmission range per node. For a given  $v_{max}$  and

transmission range per node, the number of static nodes does not make a significant impact on the difference in the network lifetime incurred with the two data gathering trees, especially at moderate transmission ranges per node of 25 and 30m.

With respect the impact of the operating parameters on the absolute magnitude of the network lifetime, we observe the network lifetime incurred with the two data gathering trees increases with increase in the number of static nodes for a given value of  $v_{max}$  and transmission range per node. For a given level of node mobility, the network lifetime increases with increase in transmission range per node; however, for the MST-DG trees, the rate of increase decreases with increase in the maximum node velocity. This could be attributed to the relative instability of the MST-DG trees at high node mobility levels, requiring frequent tree reconfigurations. During a network-wide flooding, all nodes in the network tend to lose energy, almost equally. The Max.Stability-DG trees maintain a steady increase in the network lifetime with increase in transmission range per node for all levels of node mobility. For a given transmission range per node and number of static nodes, the network lifetime incurred for the two data gathering trees decreases with increase in the maximum node velocity, especially for the MST-DG trees due to their instability. This could be attributed to the energy loss incurred due to frequent tree discoveries.

For a given transmission range per node, with the absolute values of the node lifetime increasing with increase in the maximum node velocity and the network lifetime decreasing with increase in the maximum node

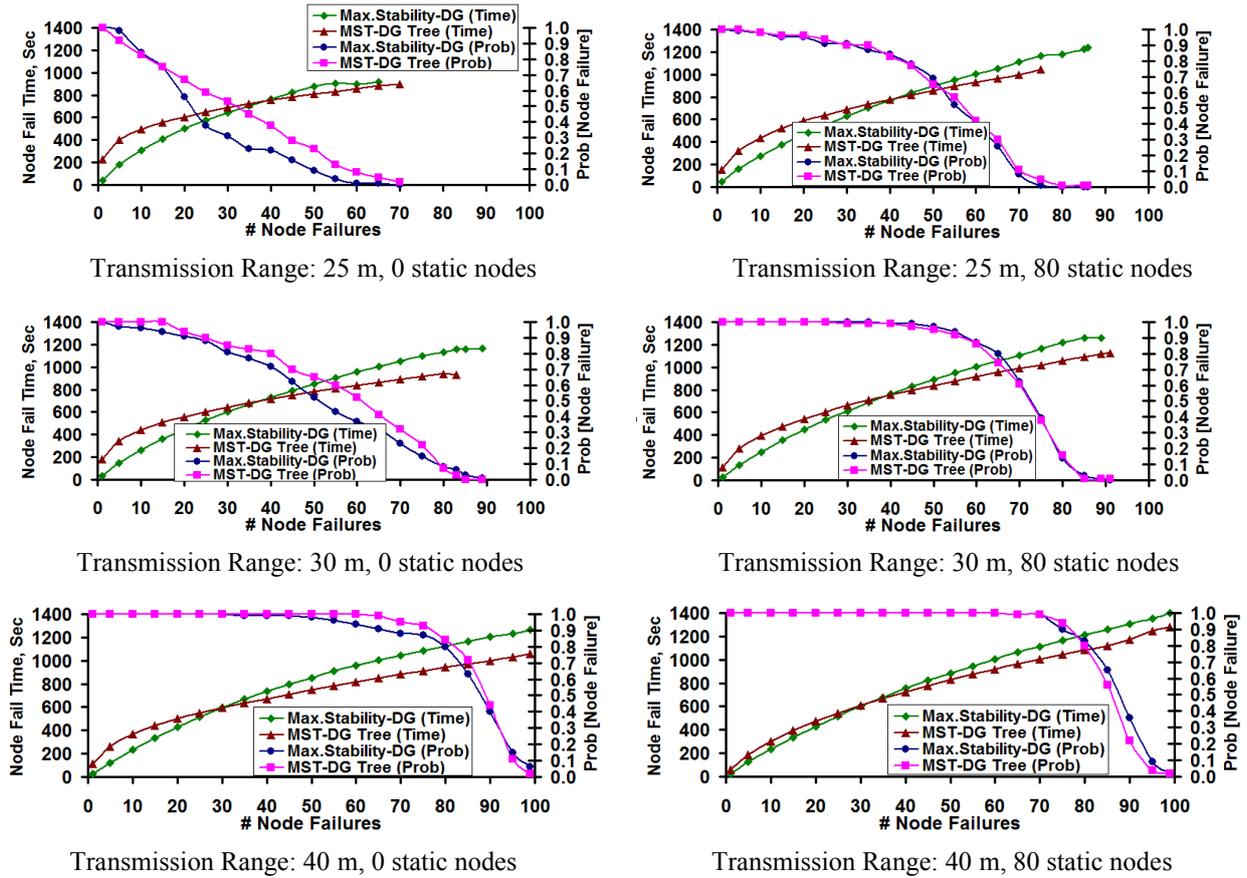


Figure 17: Distribution of Node Failure Times and Probability of Node Failures [ $v_{max} = 3$  m/s].

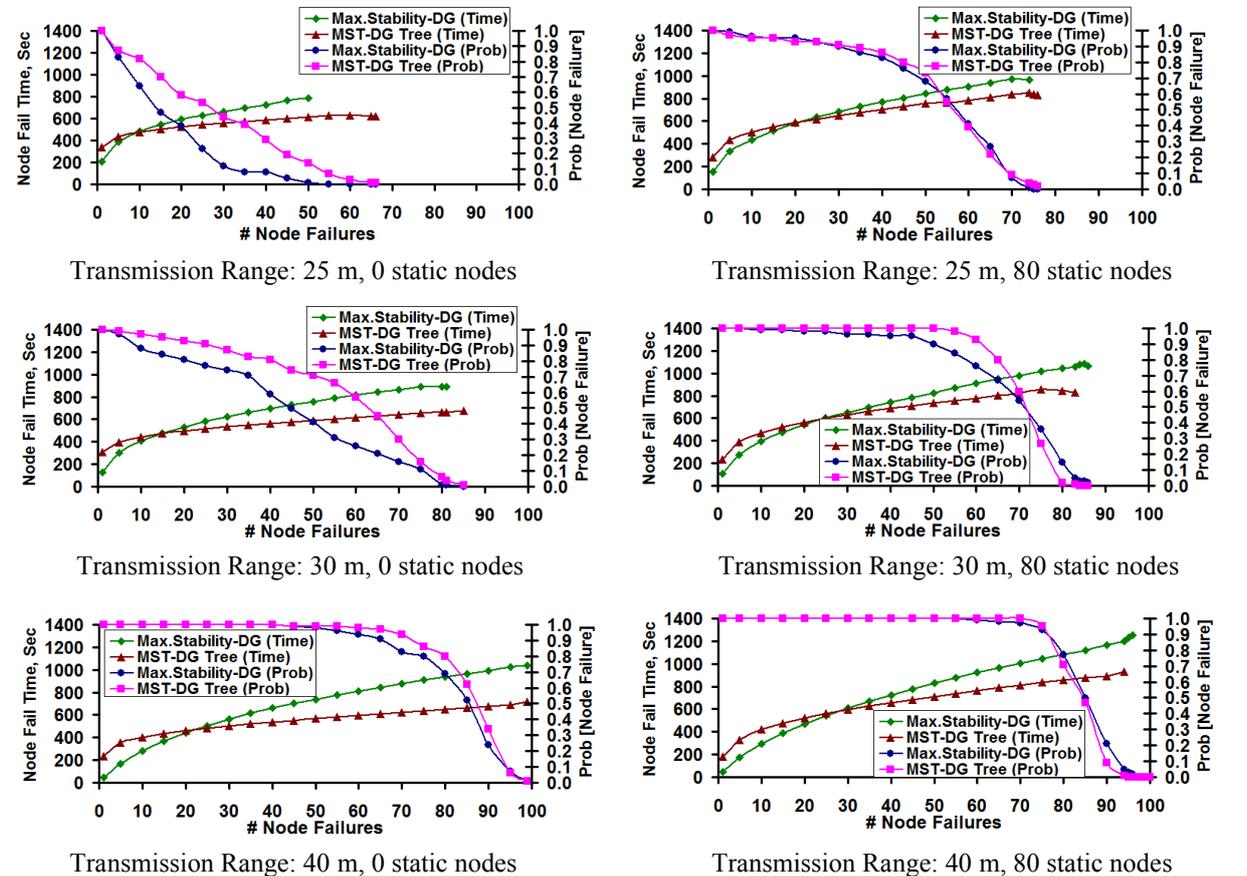


Figure 18: Distribution of Node Failure Times and Probability of Node Failures [ $v_{max} = 10$  m/s].

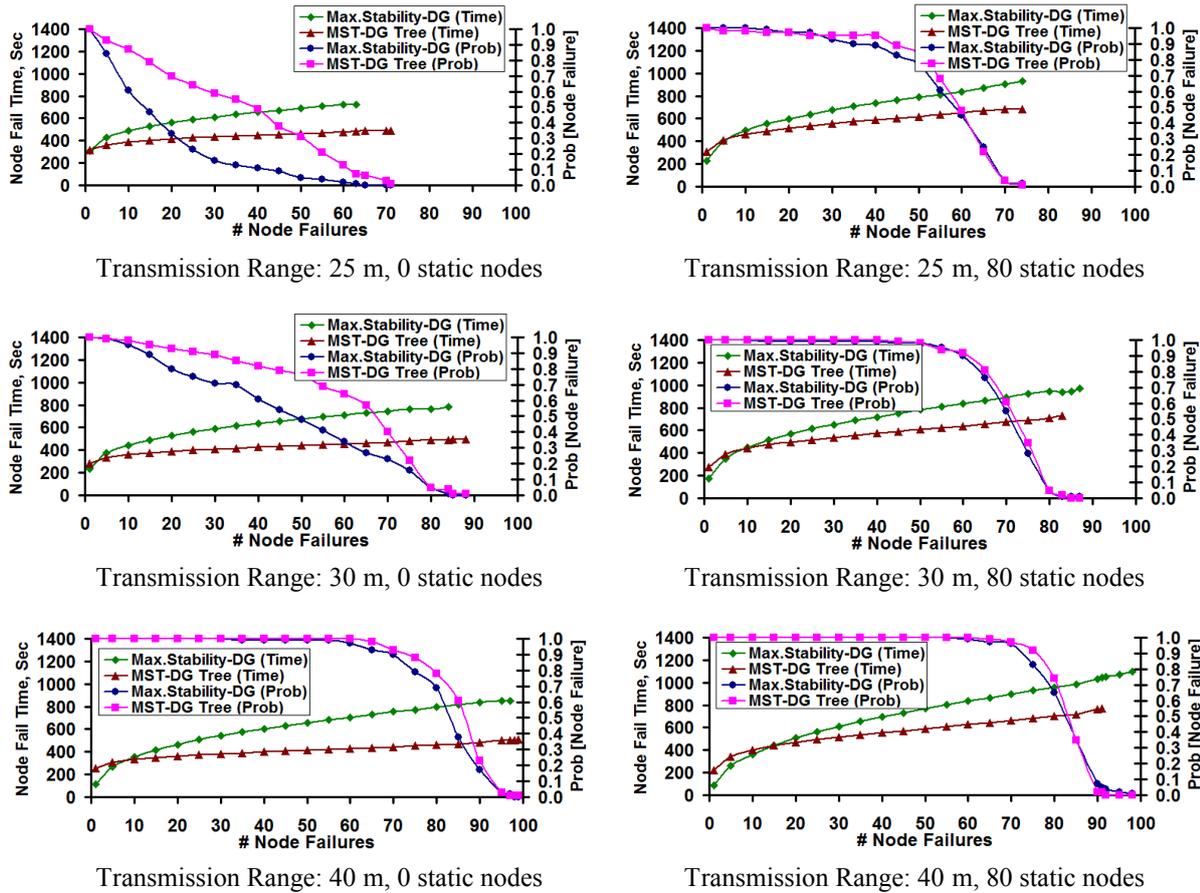


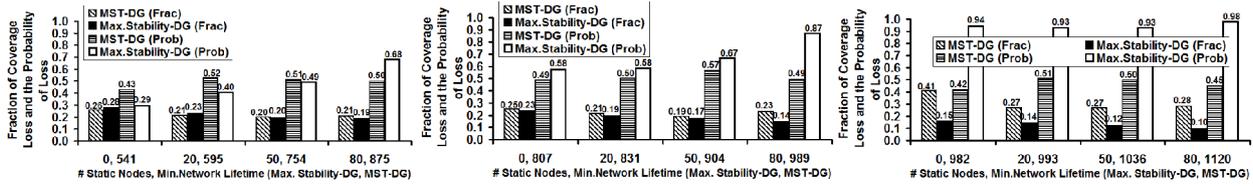
Figure 19: Distribution of Node Failure Times and Probability of Node Failures [ $v_{max} = 20$  m/s].

velocity, we observe the maximum increase in the absolute time of node failures to occur at low node mobility. This vindicates the impact of network-wide flooding based tree discoveries on energy consumption at the nodes. Since all nodes are likely to lose the same amount of energy with flooding, the more we conduct flooding, the larger is the network-wide energy consumption. As a result, node failures tend to occur more frequently when we conduct frequent flooding. Thus, even though operating the network at moderate and high levels of node mobility helps us to extend the time of first node failure, the subsequent node failures occur too soon after the first node failure. This could be justified with the observation of flat curves for the MST-DG trees with respect to the distribution of node failure times (in Figures 17, 18 and 19). The distribution of node failure times is relatively steeper for the Max.Stability-DG trees. The unfair usage of nodes in the initial stages does help the Max.Stability-DG trees to prolong the network lifetime. Aided with node mobility, it is possible for certain energy-rich nodes (that might have been leaf nodes in an earlier data gathering tree) to keep the network connected for a longer time by serving as intermediate nodes, and the energy-deficient nodes serve as leaf nodes during the later rounds of data gathering.

The impact of mobility in prolonging node failure lifetimes could also be explained by the lower probability of node failure observed for the Max.Stability-DG trees in comparison to the MST-DG trees when there are 0

static nodes (the plots to the left in Figures 17, 18 and 19). At 80 static nodes, the probability of node failures for the two data gathering trees is about the same and is higher than that observed when all nodes are mobile. This could be attributed to the repeated overuse of certain nodes as intermediate nodes and leader node on relatively more stable data gathering trees. Thus, with the use of static nodes, even though the absolute magnitude of the network lifetime can be marginally increased (by about 10-70%; the increase is larger at moderate transmission range per node and larger values of  $v_{max}$ ), the probability of node failures to occur also increases.

In terms of the percentage difference in the values for the network lifetime and node lifetime incurred with the two data gathering trees, we observe the Max.Stability-DG trees to incur a significantly prolonged network lifetime, beyond the time of first node failure. For a given transmission range per node and maximum node velocity, we observe the difference between the node lifetime and network lifetime for the Max.Stability-DG trees to increase significantly with increase in the number of static nodes. This could be attributed to the reduction in the number of flooding-based tree discoveries. For a given level of node mobility, we observe the difference in the node lifetime and network lifetime for the Max.Stability-DG trees to increase with increase in the transmission range per node. This could be again attributed to the decrease in the number of network-wide flooding based tree discoveries when used

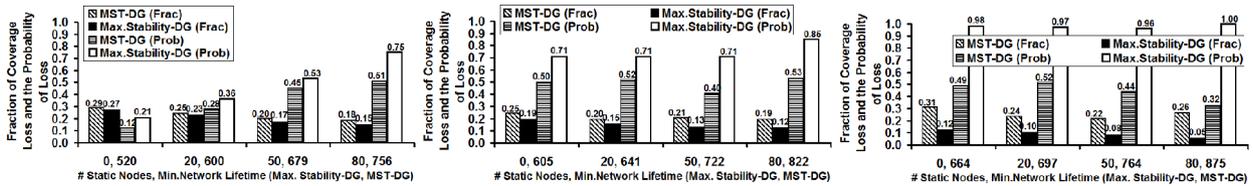


Transmission Range = 25 m

Transmission Range = 30 m

Transmission Range = 40 m

Figure 20: Fraction of Coverage Loss and Associated Probability (Low Node Mobility:  $v_{max} = 3$  m/s)

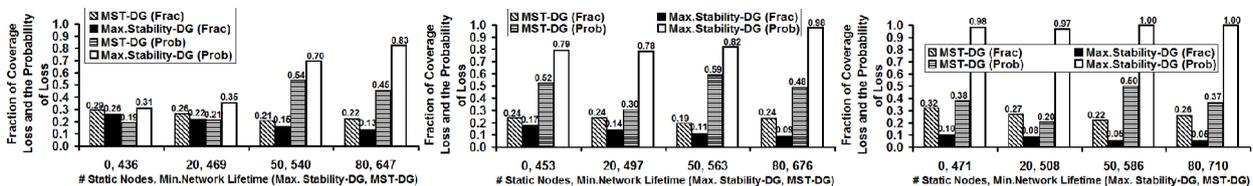


Transmission Range = 25 m

Transmission Range = 30 m

Transmission Range = 40 m

Figure 21: Fraction of Coverage Loss and Associated Probability (Moderate Node Mobility:  $v_{max} = 10$  m/s)

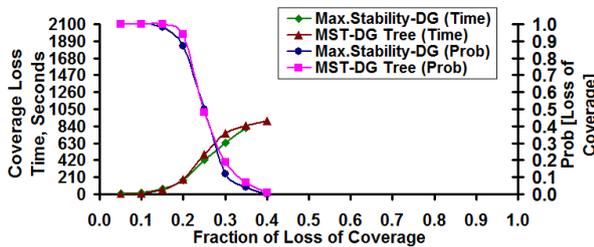


Transmission Range = 25 m

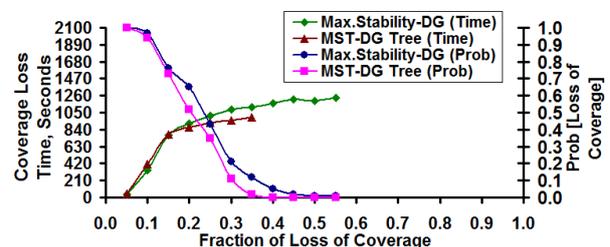
Transmission Range = 30 m

Transmission Range = 40 m

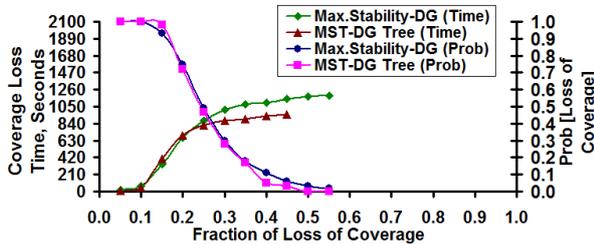
Figure 22: Fraction of Coverage Loss and Associated Probability (High Node Mobility:  $v_{max} = 20$  m/s).



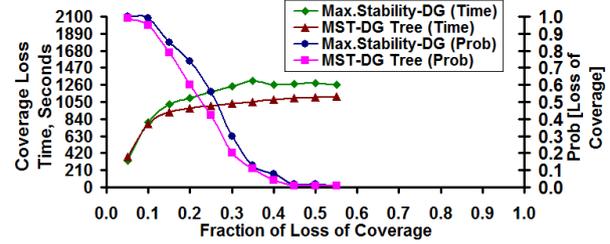
Transmission Range: 25 m, 0 static nodes



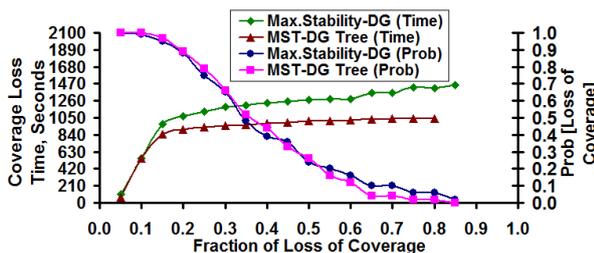
Transmission Range: 25 m, 80 static nodes



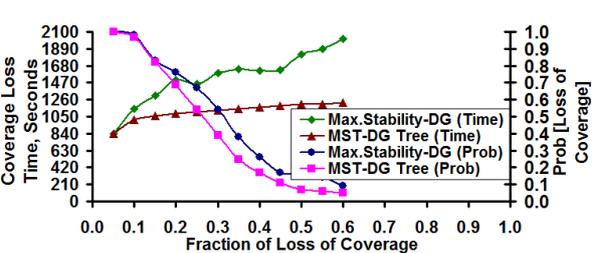
Transmission Range: 30 m, 0 static nodes



Transmission Range: 30 m, 80 static nodes



Transmission Range: 40 m, 0 static nodes



Transmission Range: 40 m, 80 static nodes

Figure 23: Coverage Loss Time and the Probability of Coverage Loss [Low Mobility:  $v_{max} = 3$  m/s].

at larger transmission ranges per node. Relatively, the MST-DG trees incur a very minimal increase in the network lifetime compared to the node lifetime, especially when operated at higher levels of node mobility.

One can also observe from Figures 17, 18 and 19 that the number of node failures that require for the node failure time incurred with the Max.Stability-DG trees to exceed that of the node failure time incurred with the MST-DG trees decreases with increase in maximum node mobility. This could be attributed to the premature very early node failure occurring for the Max.Stability-DG trees when operated under low node mobility scenarios, with the time of first node failure for the MST-DG tree being as large as 400% more than the time of first node failure for the Max.Stability-DG tree. On the other hand, at high levels of node mobility, the time of first node failure incurred with the MST-DG trees is at most 100% larger than that of the Max.Stability-DG trees. Hence, the node failure times incurred with the Max.Stability-DG trees could quickly exceed that of the MST-DG trees at higher levels of node mobility. At the same time, the probability for node failures to occur (that was relatively low at moderate transmission ranges per node, low and moderate levels of node mobility) with the Max.Stability-DG trees converges to that of the MST-DG trees when operated at higher levels of node mobility as well as with larger transmission ranges per node. For a given  $v_{max}$  value and transmission range per node, we also observe that the number of node failures required for the failure times incurred with the Max.Stability-DG trees to exceed that of the MST-DG trees increases with increase in the number of static nodes.

### 5.8 Coverage loss at a common timeline

In this section, we compare the loss of coverage incurred with both the Max.Stability-DG and MST-DG trees with respect to a *common timeline, chosen to be the minimum of the network lifetime obtained for the two data gathering trees* under every operating condition of transmission range per node, maximum node velocity and the number of static nodes. Given the nature of the results obtained for the network lifetime under different operating conditions, the minimum of the network lifetime for the two data gathering trees ended up mostly being the network lifetime observed for the MST-DG trees. For this value of network lifetime, we measured the fraction of coverage loss in the network incurred for each of the two data gathering trees, as well as measured the probability with which the corresponding fraction of coverage loss is observed.

Under the above measurement model, we observe the Max.Stability-DG trees incur lower values of the fractions of coverage loss at the minimum of the network lifetime incurred for the two data gathering trees for most of all the 48 combinations of the operating conditions of maximum node velocity, number of static nodes and transmission range per node (see Figures 20, 21 and 22). However, the fraction of coverage loss observed for the

Max.Stability-DG trees is bound to occur with a higher probability than that of the coverage loss to be incurred by using the MST-DG trees. The difference in the fraction of coverage loss incurred for the Max.Stability-DG trees vis-à-vis could be as large as 0.18-0.21, observed at transmission range per node of 40m and 80 static nodes, under all levels of node mobility. The only three combinations of operating conditions for which the Max.Stability-DG trees sustain a larger value for the fraction of coverage loss (that too, only by 0.02) are at a transmission range per node of 25m -  $v_{max} = 3$  m/s, 0 and 20 static nodes; and  $v_{max} = 10$  m/s, 0 static nodes.

In the case of the Max.Stability-DG trees, for a fixed  $v_{max}$  value, we observe the fraction of loss of coverage to decrease with increase in transmission range per node from 25m to 40m, of course with a higher probability. The significant decrease in the loss of coverage (as low as 0.05) at higher transmission range per node of 40m could also be attributed to the increase in the network lifetime, and also due to the reason that we measure the loss of coverage at a time value (corresponding to the network lifetime of the MST-DG trees), which is lower than the network lifetime of the Max.Stability-DG trees. For fixed  $v_{max}$  and transmission range, as we increase the number of static nodes, the fraction of coverage loss decreases significantly for Max.Stability-DG trees by about 0.05 to 0.1; whereas, the fraction of coverage loss for the MST-DG trees suffers a very minimal decrease or remains the same. For a fixed # static nodes and transmission range per node, node velocity has minimal impact on coverage loss for the MST-DG trees.

### 5.9 Distribution of coverage loss

In Figures 23, 24 and 25, we illustrate the distribution of the time (referred to as the coverage loss time) at which particular fractions of coverage loss occurs in the network when run with the Max.Stability-DG and MST-DG trees (until the network lifetime of the individual data gathering tree). The Max.Stability-DG trees incur larger values of coverage loss time for moderate and higher values of the fractions of coverage loss (generally above 0.15 or 0.2), under most of the combinations of the operating conditions of maximum node velocity, 0 and 80 static nodes and transmission range per node. For quantitative comparison purposes, we base our discussion in this section on the coverage loss time observed when the fraction of coverage loss is 0.3. For most of the combinations of operating conditions, we observe the coverage loss times incurred with the Max.Stability-DG and MST-DG trees to flatten out (i.e., not appreciably increase) starting from this fraction of coverage loss.

In terms of the percentage difference in the coverage loss time incurred at a fraction of coverage loss of 0.3, we observe the coverage loss time incurred with the Max.Stability-DG trees to be about 15-40%, 15-45% and 30-70% greater than the coverage loss time incurred with the MST-DG trees at low, moderate and high levels of node mobility respectively. For fixed transmission range

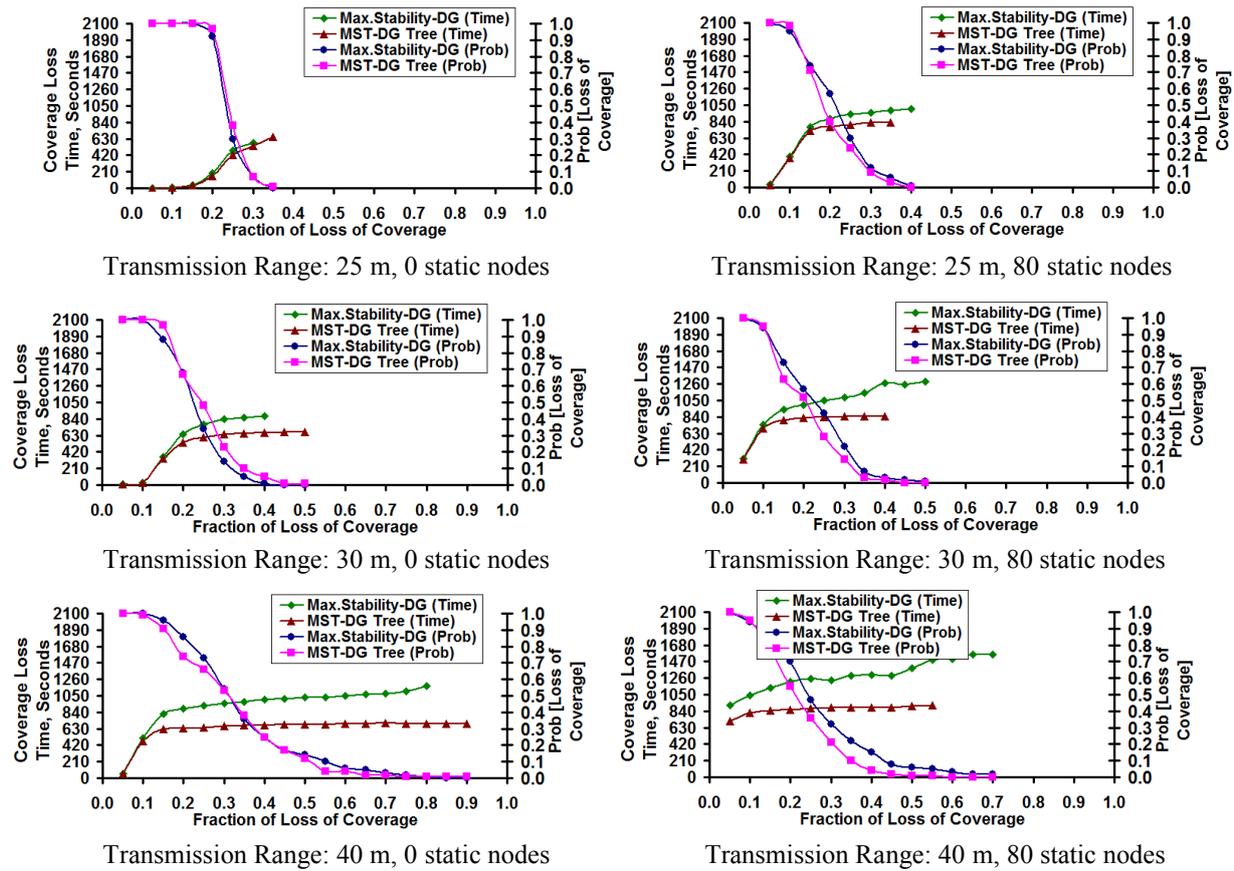


Figure 24: Coverage Loss Time and the Probability of Coverage Loss [Moderate Mobility:  $v_{max} = 10$  m/s].

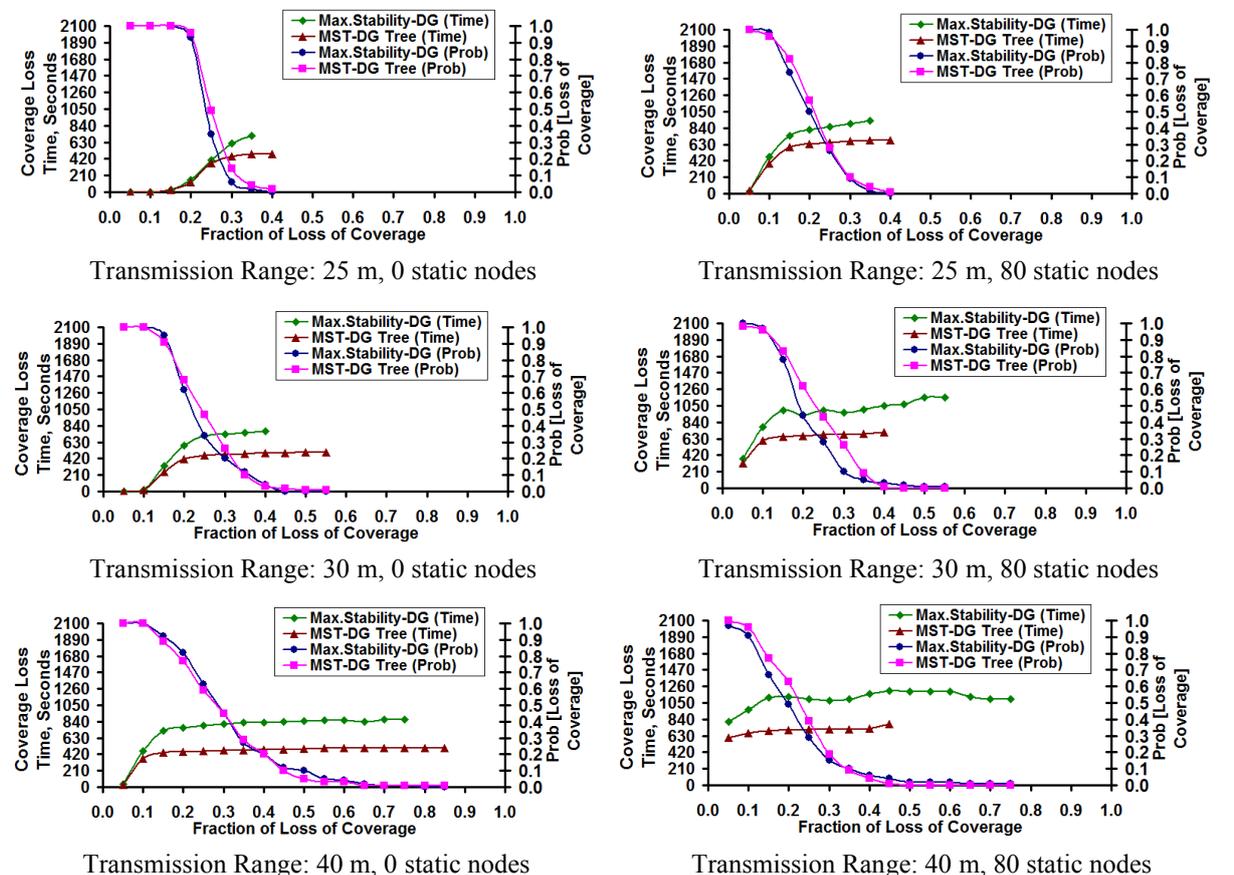


Figure 25: Coverage Loss Time and the Probability of Coverage Loss [High Mobility:  $v_{max} = 20$  m/s].

per node and number of static nodes, the absolute magnitude for the coverage loss time incurred for both the data gathering trees decreases with increase in the  $v_{max}$  value.

For a given level of node mobility, the coverage loss time incurred with the Max.Stability-DG trees almost doubles, if not more, as we increase the transmission range per node from 25m to 40m and the number of static nodes from 0 to 80. This could be attributed to the significant energy savings obtained as a result of the need for very few network-wide flooding tree discoveries with the use of the Max.Stability-DG algorithm when operated at larger transmission ranges per node and/or more static nodes. We observe significant gains in the coverage loss time when the number of static nodes is also simultaneously increased with increase in the transmission range per node. In fact, at moderate and high levels of node mobility, the coverage loss time incurred when we run the network at transmission range per node of 25m and increase the number of static nodes from 0 to 80 is greater than or equal to the coverage loss time incurred when we run the network with 0 static nodes and increase the transmission range per node from 25m to 40m. In the case of MST-DG trees, the percentage increase in the coverage loss time with increase in the number of static nodes vis-à-vis increase in the transmission range per node is more obvious.

For both the data gathering trees, especially in the case of MST-DG trees, the potential energy savings obtained with respect to reduction in the number of network-wide flooding discoveries is much more when we operate at a moderate transmission range per node and increase the number of static nodes from 0 to 80 rather than operating at a larger transmission range per node with 0 static nodes. It is to be noted that larger the transmission range, the larger is the energy lost in transmission, and also larger is the energy lost due to receipt of the control messages from a larger number of neighbor nodes. For both the data gathering trees, we observe the increase in coverage loss time with the use of more static nodes vis-à-vis a larger transmission range per node to occur with a relatively lower probability of coverage loss.

## 6 Conclusions

The high-level contribution of this research is the design and development of a benchmarking algorithm (Max.Stability-DG algorithm) to obtain the upper bounds for the maximum lifetime that can be incurred with data gathering trees for mobile sensor networks. Given the entire sequence of topology changes over the duration of the data gathering session as input, the Max.Stability-DG algorithm returns the sequence of longest-living stable data gathering trees such that the number of tree discoveries is the global minimum. The run-time complexity of the algorithm has been observed to be  $O(n^2 T \log n)$  and  $O(n^3 T \log n)$  when operated under sufficient-energy and energy-constrained scenarios respectively, where  $n$  is the number of nodes in the network and  $T$  is the duration of the data gathering

session. Since the Max.Stability-DG trees are spanning tree-based and a spanning tree exists in a network if and only if the network is connected, the stability of a spanning tree or any network-wide communication topology (like a connected dominating set) discovered by an existing or prospective data gathering algorithm can be evaluated by comparing its lifetime with that obtained for the Max.Stability-DG trees. With a polynomial-time complexity and a much broader scope of application, as described above, the Max.Stability-DG algorithm has all the characteristics to become a global standard for evaluating the stability of communication topologies for data gathering in mobile sensor networks.

We have shown that under sufficient-energy scenarios, the number of spanning tree discoveries incurred with the Max.Stability-DG algorithm is the theoretical minimum for any network-wide communication topology used for data gathering. In addition to the theoretical evaluation and proof of correctness, we have also shown through extensive simulations that the Max.Stability-DG trees are significantly more stable than the MST-DG trees under both sufficient-energy and energy-constrained scenarios. We evaluate the performance of the data gathering trees obtained with the Max.Stability-DG algorithm under diverse conditions of network dynamicity (varied by changing the maximum node velocity and number of static nodes) and network density (varied by changing the transmission range per node). Due to its nature to use a long-living data gathering tree as long as it exists, we observe the Max.Stability-DG algorithm to incur a lower time for the first node failure. However, the tradeoff between stability and fairness of node usage ceases to exist beyond the first few node failures; the reduced number of network-wide flooding discoveries coupled with the shallow structure and even distribution of nodes across the intermediate nodes (which also contribute to a lower delay per round) contribute to a longer lifetime for the remaining nodes in the network and significantly prolong the network lifetime as well as the coverage loss time. On the contrary, the MST-DG trees that incur a larger time for the first node failure are observed to incur a significantly lower network lifetime and lower coverage loss time for a given fraction of loss of coverage (and correspondingly incur a larger fraction of coverage loss at any time), owing to frequent network-wide flooding-based tree discoveries that expedite the node failures after the first node failure. We did not come across such a comprehensive analysis for node failure times, network lifetime, coverage loss times and fraction of coverage loss in any prior work in the literature.

Table 1 summarizes the overall performance gains obtained with the Max.Stability-DG tree vis-à-vis the MST-DG trees under both the sufficient-energy and energy-constrained scenarios, as applicable. Table 2 ranks the three operating parameters in the decreasing order of influence on the performance of the two data gathering trees. As can be seen from Table 2, the nature of influence of the operating parameters on the performance of the two data gathering trees is more or less the same.

Performance Metric	Better Data Gathering Tree	Range of Performance Gain Compared to the other Data Gathering Tree	
		Sufficient-Energy Scenario	Energy-Constrained Scenario
Tree Lifetime	Max.Stability-DG tree	150% to 360% larger	40% to 200% larger
Delay per Round	Max.Stability-DG tree	4% to 25% lower	7% to 18% lower
Energy Lost per Node	Max.Stability-DG tree	7% to 45% lower	Not applicable
Energy Lost per Round	Max.Stability-DG tree	7% to 45% lower	Not applicable
Fairness of Node Usage	MST-DG tree	10% to 50% better	Not applicable
Node Lifetime	MST-DG tree	Not applicable	10% to 420% larger
Network Lifetime	Max.Stability-DG tree	Not applicable	5% to 60% larger
Coverage Loss Time	Max.Stability-DG tree	Not applicable	15% to 70% larger
Fraction of Coverage Loss	Max.Stability-DG tree	Not applicable	2% to 25% lower

Table 1: Overall Performance Gains for the Maximum Stability Spanning Tree based Data Gathering (Max.Stability-DG) Tree and Minimum-distance Spanning Tree based Data Gathering (MST-DG) Tree.

Performance Metric	Ranking of the Operating Parameters in the Order of Influence [1-Highest Influence]					
	Max. Stability-based Data Gathering			Min. distance-based Data Gathering		
	Node Velocity	Static Nodes	Transmission Range/ Node	Node Velocity	Static Nodes	Transmission Range/ Node
Tree Lifetime	1	3	2	1	3	2
Delay per Round	3	2	1	3	2	1
Energy Lost / Node	1	2	3	1	2	3
Energy Lost / Round	1	2	3	1	2	3
Fairness of Node Use	1	3	2	1	3	2
Node Lifetime	1	3	2	1	3	2
Network Lifetime	1	2	3	1	2	3
Coverage Loss Time	1	2	3	1	2	3
Frac. Coverage Loss	3	2	1	2	3	1

Table 2: Influence of the Operating Parameters on the Performance of the Data Gathering Trees.

## 7 Future work

As part of future work, we plan to do the following:

- (i) We will develop a distributed stability-based data gathering algorithm for mobile sensor networks based on the notion of the predicted link expiration time (LET) [27] – a link model successfully proposed for stable path routing in mobile ad hoc networks. From the lessons learnt in this research, we conjecture that a simple LET-based data gathering algorithm would discover stable trees at the cost of premature node failures (at least the first few node failures would occur much earlier, like we observed for the Max.Stability-DG trees). Hence, we plan to model the edge weight as a function of both the LET and the residual energy (available energy) of the end nodes of the link, so that we can balance the tradeoff between stability and unfairness in node usage. We can then compare the stability-node lifetime tradeoff of the LET-energy-DG trees with that of the Max.Stability-DG trees.

- (ii) As stable data gathering trees are likely to be used for a longer time, the trustworthiness of the data aggregated at the intermediate nodes needs to be validated and maintained through proper trust-evaluation schemes. We plan to develop and integrate a trust-evaluation model as part of stable data aggregation in mobile sensor networks.
- (iii) We plan to compare the stability of the Max.Stability-DG trees under several different node mobility models [4] vis-à-vis the Random waypoint model, the mobility model used in our simulations that has been widely used in the ad hoc network literature.

## Acknowledgment

This research was sponsored by the U. S. Air Force Office of Scientific Research (AFOSR) through the Summer Faculty Fellowship Program for the lead author (Natarajan Meghanathan) in June-July 2012. The research was conducted under the supervision of the co-author (Philip D. Mumford) at the U. S. Air Force Research Lab (AFRL), Wright-Patterson Air Force Base

(WPAFB) Dayton, OH. The public release approval number is 88ABW-2012-4935. The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

## References

- [1] M. Abolhasan, T. Wysocki and E. Dutkiewicz, "A Review of Routing Protocols for Mobile Ad hoc Networks," *Ad hoc Networks*, vol. 2, no. 1, pp. 1-22, January 2004.
- [2] T. Banerjee, B. Xie, J. H. Jun, and D. P. Agarwal, "LIMOC: Enhancing the Lifetime of a Sensor Network with Mobile Clusterheads," *Proceedings of the Vehicular Technology Conference Fall*, pp. 133-137, 2007.
- [3] C. Bettstetter, H. Hartenstein and X. Perez-Costa, "Stochastic Properties of the Random-Way Point Mobility Model," *Wireless Networks*, vol. 10, no. 5, pp. 555 – 567, September 2004.
- [4] T. Camp, J. Boleng and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communication and Mobile Computing*, vol. 2, no. 5, pp. 483-502, September 2002.
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms," 3<sup>rd</sup> Edition, MIT Press, July 2009.
- [6] S. Deng, J. Li and L. Shen, "Mobility-based Clustering Protocol for Wireless Sensor Networks with Mobile Nodes," *IET Wireless Sensor Systems*, vol. 1, no. 1, pp. 39-47, 2011.
- [7] A. Farago and V. R. Syrotiuk, "MERIT: A Scalable Approach for Protocol Assessment," *Mobile Networks and Applications*, Vol. 8, No. 5, pp. 567 – 577, October 2003.
- [8] W. Heinzelman, A. Chandrakasan and H. Balakarishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," *Proceedings of the Hawaiian International Conference on Systems Science*, January 2000.
- [9] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan and S. Madden, "CarTel: A Distributed Mobile Sensor Computing System," *Proceedings of the 4<sup>th</sup> International Conference on Embedded Networked Sensor Systems*, November 2006.
- [10] A. Kansal, M. Goraczko and F. Zhao, "Building a Sensor Network of Mobile Phones," *Proceedings of International Symposium on Information Processing in Sensor Networks*, pp. 547-548, April 2007.
- [11] F. Kuhn, T. Moscibroda and R. Wattenhofer, "Unit Disk Graph Approximation," *Proceedings of the Workshop on Foundations of Mobile Computing*, pp. 17-23, October 2004.
- [12] S. Lindsey, C. Raghavendra and K. M. Sivalingam, "Data Gathering Algorithms in Sensor Networks using Energy Metrics," *IEEE Transactions on Parallel and Distributed Systems*, 13(9):924-935, September 2002.
- [13] C-M. Liu, C-H. Lee and L-C. Wang, "Distributed Clustering Algorithms for Data Gathering in Wireless Mobile Sensor Networks," *Journal of Parallel and Distributed Computing*, vol. 67, no. 11, pp. 1187-1200, November 2007.
- [14] G. Lu, B. Krishnamachari and C. S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Tree-based Data Gathering in Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 7, no. 7, pp. 863-875, September 2007.
- [15] M. Macuha, M. Tariq and T. Sato, "Data Collection Method for Mobile Sensor Networks based on the Theory of Thermal Fields," *Sensors*, vol. 11, no. 7, pp. 7188-7203, July 2011.
- [16] N. Meghanathan, "A Comprehensive Review and Performance Analysis of Data Gathering Algorithms for Wireless Sensor Networks," *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, vol. 4, no. 2, pp. 1-29, April-June 2012.
- [17] N. Meghanathan, "A Data Gathering Algorithm based on Energy-aware Connected Dominating Sets to Minimize Energy Consumption and Maximize Node Lifetime in Wireless Sensor Networks," *International Journal of Interdisciplinary Telecommunications and Networking*, vol. 2, no. 3, pp. 1-17, July-September 2010.
- [18] N. Meghanathan and G. W. Skelton, "A Two Layer Architecture of Mobile Sinks and Static Sensors," *Proceedings of the 15th International Conference on Advanced Computing and Communication*, pp. 249-254, Guwahati, India, December 2007.
- [19] N. Meghanathan, S. Sharma and G. W. Skelton, "On Energy Efficient Dissemination in Wireless sensor Networks using Mobile Sinks," *Journal of Theoretical and Applied Information Technology*, vol. 19, no. 2, pp. 79-91, September 2010.
- [20] N. Meghanathan, "Performance Comparison of Minimum Hop and Minimum Edge Based Multicast Routing Under Different Mobility Models for Mobile Ad Hoc Networks," *International Journal of Wireless and Mobile Networks*, vol. 3, no. 3, pp. 1-14, June 2011.
- [21] N. Meghanathan and A. Farago, "On the Stability of Paths, Steiner Trees and Connected Dominating Sets in Mobile Ad Hoc Networks," *Ad Hoc Networks*, vol. 6, no. 5, pp. 744 - 769, July 2008.
- [22] T. S. Rappaport, "Wireless Communications: Principles and Practice," 2<sup>nd</sup> edition, Prentice Hall, January 2002.
- [23] G. Santhosh Kumar, M. V. Vinu Paul and K. Jacob Poulouse, "Mobility Metric based LEACH-Mobile Protocol," *Proceedings of the 16<sup>th</sup> International Conference on Advanced Computing and Communications*, pp. 248-253, December 2008.

- [24] H. K. D. Sarma, A. Kar and R. Mall, “Energy Efficient and Reliable Routing for Mobile Wireless Sensor Networks,” *Proceedings of the 6<sup>th</sup> IEEE International Conference on Distributed Computing in Sensor Systems Workshops*, June 2010.
- [25] M. Singh, M. Sethi, N. Lal and S. Poonia, “A Tree Based Routing Protocol for Mobile Sensor Networks (MSNs),” *International Journal on Computer Science and Engineering*, vol. 2, no. 1S, pp. 55-60, 2010.
- [26] A. Srinivasan and J. Wu, “TRACK: A Novel Connected Dominating Set based Sink Mobility Model for WSNs,” *Proceedings of the 17<sup>th</sup> International Conference on Computer Communications and Networks*, August 2008.
- [27] W. Su and M. Gerla, “IPv6 Flow Handoff in Ad hoc Wireless Networks using Mobility Prediction,” *Proceedings of the IEEE Global Telecommunications Conference*, pp. 271-275, December 1999.
- [28] A. J. Viterbi, “CDMA: Principles of Spread Spectrum Communication,” 1<sup>st</sup> edition, Prentice Hall, April 1995.
- [29] N. Vljajic and D. Stevanovic, “Sink Mobility in Wireless Sensor Networks: When Theory meets Reality,” *Proceedings of the IEEE Sarnoff Symposium*, March-April 2009.
- [30] W. Wu, H. Beng Lim and K-L. Tan, “Query-driven Data Collection and Data Forwarding in Intermittently Connected Mobile Sensor Networks,” *Proceedings of the 7<sup>th</sup> International Workshop on Data Management for Sensor Networks*, pp. 20-25, 2010.
- [31] G. Xing, T. Wang, W. Jia and M. Li, “Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station,” *Proceedings of the 9<sup>th</sup> ACM International Symposium on Mobile Ad hoc Networking and Computing*, pp. 231-240, 2008.
- [32] H. Zhang and J. C. Hou, “Maintaining Sensing Coverage and Connectivity in Large Sensor Networks,” *Wireless Ad hoc and Sensor Networks: An International Journal*, vol. 1, no. 1-2, pp. 89-123, January 2005.
- [33] [33] W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-based Collaboration for Target Tracking in Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1689-1701, September 2004.
- [34] M. Zhao and Y. Yang, “Bounded Relay Hop Mobile Data Gathering in Wireless Sensor Networks,” *Proceedings of the 6<sup>th</sup> IEEE International Conference on Mobile Ad hoc and Sensor Systems*, pp. 373-382, October 2009.
- [35] M. Zhong and C. G. Cassandras, “Distributed Coverage Control and Data Collection with Mobile Sensor Networks,” *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2445-2455, October 2011.