# Use of Tree Traversal Algorithms for Chain Formation in the PEGASIS Data Gathering Protocol for Wireless Sensor Networks

**Natarajan Meghanathan**
Department of Computer Science, Jackson State University
Jackson, MS 39217 - USA
[e-mail: natarajan.meghanathan@jsums..edu]

*Received November 2, 2009; revised November 30, 2009; accepted December 14, 2009; published December 31, 2009*

## Abstract

The high-level contribution of this paper is to illustrate the effectiveness of using graph theory tree traversal algorithms (pre-order, in-order and post-order traversals) to generate the chain of sensor nodes in the classical Power Efficient-Gathering in Sensor Information Systems (PEGASIS) data aggregation protocol for wireless sensor networks. We first construct an undirected minimum-weight spanning tree (*ud-MST*) on a complete sensor network graph, wherein the weight of each edge is the Euclidean distance between the constituent nodes of the edge. A Breadth-First-Search of the *ud-MST*, starting with the node located closest to the center of the network, is now conducted to iteratively construct a rooted directed minimum-weight spanning tree (*rd-MST*). The three tree traversal algorithms are then executed on the *rd-MST* and the node sequence resulting from each of the traversals is used as the chain of nodes for the PEGASIS protocol. Simulation studies on PEGASIS conducted for both TDMA and CDMA systems illustrate that using the chain of nodes generated from the tree traversal algorithms, the node lifetime can improve as large as by 19%-30% and at the same time, the energy loss per node can be 19%-35% lower than that obtained with the currently used distance-based greedy heuristic.

***Keywords:*** Minimum-weight spanning tree, tree traversal, PEGASIS, data gathering, wireless sensor networks, simulation

**DOI: 10.3837/tiis.2009.06.003**

## 1. Introduction

A wireless sensor network is a network of smart sensors that collect data about the ambient environment and propagate the collected data to one or more control centers called sinks, through which the end user accesses the data. The network has limited bandwidth and sensor nodes operate with limited computing and memory capacity. Also, sensor nodes operate with limited transmission range to conserve battery charge and reduce interference. The sink is normally located far away from the network field and because of the above constraints, direct communication from each of the sensor nodes to the sink cannot be a viable solution from both energy as well as bandwidth points of view. This motivates the need for data gathering protocols that can be effectively and efficiently run at the sensor nodes to combine the data and send only the aggregated data (that is a representative of the data collected from all the sensor nodes) to the sink. Data gathering algorithms typically run in several rounds, wherein during each round, data from each of the sensor nodes are collected and aggregated, and then forwarded to the sink. Though several data gathering protocols have been proposed in the literature, the two commonly used and referred protocols are the LEACH (Low-Energy Adaptive Clustering Hierarchy) [1] and the PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [2][3] protocols.

In LEACH, a certain percentage of the sensor nodes are elected as cluster heads for each round of communication. Each cluster head forms a cluster around itself and a sensor node chooses to join the cluster whose cluster head is closest to it. If $P$ is the percentage of nodes that can be cluster heads, LEACH ensures that a sensor node is elected as cluster head exactly once within every $1/P$ rounds of data communication. PEGASIS forms a single chain of sensor nodes and the same chain is used for all the rounds of data communication. The chain of sensor nodes is formed using a greedy-heuristic based on the distance between the sensor nodes, starting from the node farthest to the sink. The nearest node to this node is added as the next node in the chain. This procedure is continued until all the nodes are included in the chain. For every round of data communication, a sensor node is uniform-randomly elected as the leader of the PEGASIS chain and data from either end of the chain gets forwarded towards the leader node. PEGASIS incurs a huge delay, especially for Time Division Multiple Access (TDMA) systems [4], as data moves across the complete chain of sensor nodes, one node at a time, before getting transmitted to the sink. For CDMA (Code Division Multiple Access) systems [5], PEGASIS has been later improved using a chain-based binary scheme to reduce the delay per round from $N$ time slots to $log\ N$ time slots, where $N$ is the number of nodes in the network.

Through several research articles [2][3][6][7], it has been shown that, for both TDMA and CDMA systems, PEGASIS yields a larger node lifetime and a lower energy consumption per round compared to LEACH. The lifetime of the nodes achieved with PEGASIS is 1.5 – 2 times more than that incurred with LEACH, whereas the energy consumed per round for LEACH is 2 – 3 times more than that incurred for PEGASIS. The performance of PEGASIS (as vindicated by our simulation results presented in the later sections of the paper) very much depends on the chain of sensor nodes along which data aggregation takes place. As of now, PEGASIS uses the chain of sensor nodes formed using the distance-based greedy heuristic for both TDMA and CDMA systems. Since greedy strategies rarely yield optimal solution, we hypothesize that the node lifetime and energy loss per node encountered with PEGASIS can be further optimized if we could use a better algorithm to construct the node chain.

The high-level technical contribution of this paper is to illustrate the effectiveness of using the traditional graph theory tree traversal algorithms (such as the pre-order, in-order and post-order traversals) to generate the chain of sensor nodes for PEGASIS data aggregation. In this pursuit, we first propose an algorithm to generate a rooted directed minimum-weight spanning tree of the entire sensor network graph. The algorithm takes as input a complete weighted network graph of the sensor nodes where the weight of an edge is the physical Euclidean distance between the constituent nodes of the edge. We first run the Kruskal's minimum-weight spanning tree algorithm [8] on the complete network graph to obtain an undirected unrooted minimum-weight spanning tree (*ud-MST*). A Breadth First Search [8] on the *ud-MST* is now conducted to obtain the rooted directed minimum-weight spanning tree (*rd-MST*). The root node of the *rd-MST* is the sensor node that is physically closest to the center of the network field. The three tree traversal algorithms (pre-order, in-order and post-order traversal) [8] are then executed on the *rd-MST* and the node sequence resulting from each of the traversals is used as the chain of nodes for the PEGASIS protocol. Simulation results illustrate that the node lifetime can be increased as large as by 30% and the energy lost per node can be relatively lower as much as by 35%, when using the tree traversal algorithms compared to the distance-based greedy heuristic. Throughout this paper, we use the terms 'data gathering' and 'data aggregation', 'node' and 'vertex', 'edge' and 'link' interchangeably. They mean the same.

The rest of the paper is organized as follows: Section 2 briefly reviews PEGASIS for both TDMA and CDMA systems and also discusses other related studies on PEGASIS. Section 3 introduces our graph theory algorithm to generate a rooted directed minimum-weight spanning tree of the sensor network and describes the execution of the pre-order, in-order and post-order traversal algorithms. Section 4 describes our simulation environment and illustrates the simulation results comparing the performance of PEGASIS based on the chain generated from the three tree traversal algorithms and the original PEGASIS protocol that is based on the distance-based greedy heuristic for chain construction. Section 5 concludes the paper.

## 2. Description of the PEGASIS Protocol and Related Studies

The PEGASIS (Power-Efficient Gathering in Sensor Information Systems) protocol [2] forms a chain of the sensor nodes and uses this chain as the basis for data aggregation. The chain is formed using a greedy approach, starting from the node farthest to the sink. The nearest node to this node is added as the next node in the chain. This procedure is continued until all the nodes are included in the chain. A node can be in the chain at only one position.

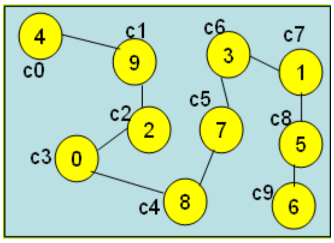

**Fig. 1**. Example for PEGASIS Chain

During each round, a leader node is randomly selected. The leader node is responsible for forwarding the aggregated data to the sink. Once the leader node is selected and notified by the sink node, each node in both sides of the chain (with respect to the leader node), receives and transmits the aggregated data to the next node in the chain, until the data reaches the leader node. For example, consider the chain formed in **Fig. 1** for a 10-node network. The index of the nodes in the chain is different from the identification numbers for the nodes (i.e., the node ID). If node 3 at chain index 6 is selected as the leader node, the flow of data would be in the following order: c0 → c1 → c2 → c3 → c4 → c5 → **c6** ← c7 ← c8 ← c9. PEGASIS can lead to significant delays in data aggregation because of the waiting time at the leader node to receive data from both sides of the chain.

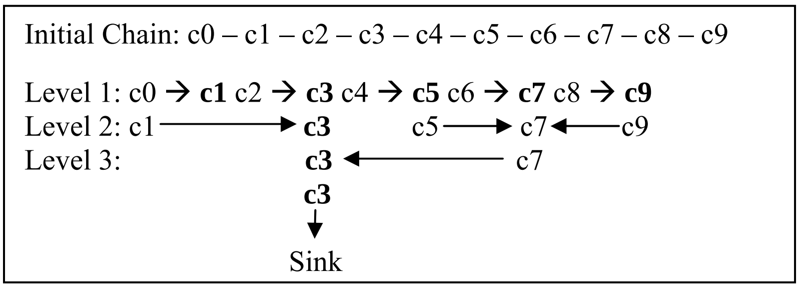
Initial Chain: c0 – c1 – c2 – c3 – c4 – c5 – c6 – c7 – c8 – c9
Level 1: c0 → c1 c2 → c3 c4 → c5 c6 → c7 c8 → c9
Level 2: c1 → c3 c5 → c7 ← c9
Level 3: c3 ← c7
c3
Sink

**Fig. 2**. Example for Chain-based Binary Scheme of PEGASIS

## 2.1 PEGASIS for CDMA Systems

The improved chain-based binary scheme of PEGASIS [3] works primarily for CDMA systems [5] where there can be simultaneous communication between any pair of nodes if each node is assigned unique CDMA code and each node knows the CDMA code for communication with every other node. The chain formed using the greedy distance-based heuristic is still used as the basis for data aggregation. Each round of data aggregation and transmission is comprised of log $N$ levels, where $N$ is the number of nodes in the network. Each node transmits the data to a close neighbor in a given level of the hierarchy. Nodes that receive data at a given level are the only nodes that rise to the next level. In order to minimize the delay, data is aggregated simultaneously using as many pairs as possible at each level. **Fig. 2** shows an example of data aggregation at different levels for a 10-node chain. Here, node at chain index 3 is selected as the leader of the round and all data gets aggregated towards this node, which is responsible for transmission to the sink.

## 2.2 Concentric Clustering based PEGASIS Protocol

An enhanced version of the PEGASIS protocol based on the idea of a concentric clustering scheme has been proposed in [9]. According to this scheme, the sensor network is logically divided into several concentric circles or levels, with the sink node being the center of each circle. The radius of each concentric circle and the number of such levels is dependent on the network density and the location of the sink. The level that is closest to the sink is said to be the lowest level and the level that is farthest from the sink is said to be the highest level. A chain is constructed among the sensor nodes in each level. The same greedy-distance based heuristic used in the original PEGASIS protocol is used for the chain construction. A head node is randomly selected for each level and a head node informs its location information to peer head nodes above and below its level in the network. At each level, data gets aggregated towards the head node at that level. The aggregated data at each level is further aggregated along a chain of the head nodes, starting from the highest level and ending at the lowest level. The head node at the lowest level forwards the aggregated data to the sink node.

## 2.3 Diamond-shaped Routing Method for PEGASIS

The diamond-shaped routing method for the PEGASIS protocol proposed in [10] is a slightly modified version of the concentric clustering based protocol. Here, the number of head nodes at the different concentric circular levels is altered between one and two to provide more reliability for data transmission. Data aggregation is simultaneously conducted at each level and the aggregated data flows from the highest level to the lowest level along a sequence of diamond-shaped communication structures formed by the head nodes. The head node at the lowest level is still the node responsible for forwarding the data to sink. In both the concentric clustering based PEGASIS and the diamond-shaped PEGASIS protocols, if the sink is located reasonably far away from the network, there can be premature node failures at the lowest level.

### 2.4 PEGASIS with Multiple Smaller Chains

In [11], the authors propose to reduce the latency associated with the original TDMA-based PEGASIS protocol by forming multiple smaller chains instead of a single longer chain. Simulation results showed optimal results if the number of nodes in each chain is 15- 20% of the number of the nodes in the network. The multiple chains are formed sequentially. Each chain is constructed starting from the sensor node that is farthest from the sink and has not been included in any other chain. The chain is constructed using the same distance-based greedy heuristic of the original PEGASIS protocol until a pre-set number of nodes (a tunable parameter of the protocol) are added to the chain. The above procedure is repeated until every node is part of one of the multiple chains. A leader node is selected for each of the smaller chains. The leader node selection is based on several parameters such as the distance to the sink, remaining energy-level and the number of times the node has been earlier selected as the leader. Leader nodes are selected for certain number of rounds (another tunable parameter of the protocol). The leader node having the highest remaining energy is selected as the global leader. Data aggregated at each leader node aggregates towards this global leader node, which forwards it further to the sink. The multiple chains used in this protocol are nothing but a sequence of broken segments of the single longer chain (of the original PEGASIS protocol). Hence, even though, this protocol can lower the latency for data aggregation, the energy efficiency is the same as that of the original PEGASIS protocol.

### 2.5 Impact of Leader Node Selection Strategies on PEGASIS

In addition to the commonly strategy of randomly selecting the leader node for every round of data aggregation, four other strategies for leader node selection were explored in [12]. The four strategies are referred to as: Shuffle (a node is selected as leader only once in $N$ rounds in a network of $N$ nodes), High-energy (node with the highest remaining energy is selected as the leader), 2-block and 4-block (the network is divided into 2 or 4 blocks and the leader node is the highest energy node in the randomly chosen block of a round). The impact of these 5 leader node selection strategies on PEGASIS was studied for both TDMA and CDMA systems. Simulation studies show that to maximize the node lifetime, the energy-aware selection strategies such as High-energy, 2-block and 4-block should be considered when the sink is located outside the network field for any network topology; on the other hand, the Random and Shuffle strategies should be preferred when the sink is located at the center of the network field. However, the leader node selection strategy did not appear to have any significant impact on the energy consumed per round and latency per round of data aggregation.

## 3. Graph Theory Algorithms

All of the methods and studies described in Section 2 still use the distance-based greedy heuristic of the original PEGASIS protocol for chain formation. The proposed approach in this paper is the first initiative to construct the PEGASIS chain based on a traversal of the minimum spanning tree formed using the sensor nodes. Minimum spanning tree-based data forwarding [13][14] has been observed to be the most suitable energy-efficient strategy for environments where all the nodes have data to send and the data can be aggregated at the forwarding nodes. PEGASIS is a classical data gathering protocol used for such environments.

In this section, we describe the graph theory algorithms used for constructing a rooted directed minimum-weight spanning tree (*rd-MST*) based on a complete network graph of wireless sensor nodes and for traversing the *rd-MST* in pre-order, in-order and post-order fashion to generate the chain of nodes for the PEGASIS data gathering protocol. An example comprehensively illustrating the generation of the *rd-MST* and its traversal to list the nodes in pre-order, in-order and post-order fashion is shown in **Fig. 7**.

### 3.1 Algorithm to Generate a Rooted, Directed Minimum-weight Spanning Tree (*rd-MST*)

The *rd-MST* algorithm (pseudo code in **Fig. 3**) takes as input a complete graph $G$ ($V$, $E$) of the wireless sensor nodes in a network of dimensions $X_{MAX}$ and $Y_{MAX}$. The set of vertices ($V$) of this graph is the sensor nodes and the set of edges ($E$) comprises of an edge for every pair of nodes in the network. The weight of an edge in $E$ is the physical Euclidean distance between the constituent nodes of the edge in the network field. The algorithm works as follows: We first generate an undirected minimum-weight spanning tree of $G$ ($V$, $E$), represented as *ud-MST* ($V^{ud}$, $E^{ud}$), using Kruskal's algorithm. Note that, $V^{ud} = V$ and $E^{ud} \subseteq E$. We then generate the rooted directed minimum-weight spanning tee, represented as *rd-MST* ($V^{rd}$, $E^{rd}$), by conducting a breadth-first search on the *ud-MST*, starting from the node (root node, *r-node*) that lies physically closest to the center of the network field. The breadth-first search algorithm visits each vertex exactly once and the sets $V^{rd}$ and $E^{rd}$ are updated as each vertex is visited. Also, as each vertex is newly visited, it is added to the set *Queue-to-Visit*, whose constituent elements (i.e., the vertices) are extracted on a First-In First-Out (FIFO) basis and the edges incident on these extracted vertices are explored to visit new vertices. For example, if a vertex $v$ is newly visited from an already visited vertex $u$ (i.e., $v \in V^{ud}$ and $u \in$ *Queue-to-Visit*, $V^{ud}$, $V^{rd}$), then the vertex $v$ is added to the sets $V^{rd}$ and *Queue-to-Visit* and the edge $u - v$ is added as a directed edge $u \rightarrow v$ to the set $E^{rd}$. The *rd-MST* is basically constructed in an iterative fashion according to the above procedure.

```
Input: Complete Graph $G$ ($V$, $E$)
          Network Dimensions: $X_{MAX}$ and $Y_{MAX}$
Output: A rooted directed minimum-weight spanning tree, rd-MST ($V^{rd}$, $E^{rd}$)
Auxiliary Variables: Queue-to-Visit, tempQueue
                     Coordinates of network center ($x_{center}$, $y_{center}$)
                     An undirected minimum-weight spanning tree, ud-MST ($V^{ud}$, $E^{ud}$)
Initialization:
$x_{center} = X_{MAX}/2$ and $y_{center} = Y_{MAX}/2$
Queue-to-Visit = $\Phi$
ud-MST ($V^{ud} = \Phi$, $E^{ud} = \Phi$)
```

```
rd-MST ($V^{rd} = \Phi$, $E^{rd} = \Phi$)

Begin
    Run Kruskal's Algorithm on G to obtain an undirected minimum-weight spanning tree ud-MST ($V^{ud}$, $E^{ud}$), where $V^{ud} = V$ and $E^{ud} \subseteq E$.
    Root r-node = $\left\{ v \mid \underset{v \in V}{Min}\left[\text{distance}\left((x_v, y_v), (x_{center}, y_{center})\right)\right] \right\}$
    Queue-to-Visit = {r-node}

    While ( $|E^{rd}| < |E^{ud}|$ ) do
        tempQueue = $\Phi$
        for every vertex $u \in$ Queue-to-Visit do
            $V^{rd} = V^{rd}$ U {u}
            for every edge $(u, v) \in E^{ud}$ do
                tempQueue = tempQueue U {v}
                $E^{rd} = E^{rd}$ U {u→v}
            end for
        end for
        Queue-to-Visit = tempQueue
    end While

    return rd-MST ($V^{rd}$, $E^{rd}$)

End
```

**Fig. 3**. Algorithm to Generate a Rooted directed Minimum Spanning Tree of the Sensor Network Graph

The run-time complexity of the *rd-MST* algorithm is O(|*E*|*log |*E*|) + O(|*V*| + |*E*|) = O(|*V*| + |*E*|*log |*E*|), where O(|*E*|*log |*E*|) is the run-time complexity of the Kruskal's minimum-weight spanning tree algorithm [8] and O(|*V*| + |*E*|) is the run-time complexity of Breadth-First Search [8], both on a graph of |*V*| vertices and |*E*| edges. The complexity of Kruskal's algorithm can be lowered from O(|*E*|*log |*E*|) to O(|E|*α(|V|)) time, where α is the extremely slowly-increasing inverse of the signle-valued Ackermann function [8]. The run-time complexity of the distance-based greedy chain formation heuristic used in the original PEGASIS protocol on a network of |*V*| nodes would be O(|*V*|$^2$). This is because we add nodes to the chain, one at a time, and it takes O(|*V* |) time to find the node that is closest to the last added node to the chain, during the recent iteration. Such iterations are repeated |*V*| times.

### 3.2 Algorithms to Traverse the Rooted Directed Minimum-weight Spanning Tree (*rd-MST*)

The vertices of the *rd-MST* can be listed, exactly once, in a sequence by traversing the tree in a systematic order. Depending on the order in which the vertices are visited, starting with the root vertex, there exist three different types of tree traversal algorithms: pre-order, in-order and post-order traversals. The three traversal algorithms can be easily described through recursion (as illustrated in the pseudo codes of **Fig. 4**, **5** and **6**). Each of the three traversal algorithms takes the *rd-MST* and the root node (*r-node*) as input and generates the sequence of vertices depending on the order in which the vertices are visited, starting with the *r-node*. In pre-order traversal, a node is visited first before visiting the nodes in any of its subtrees. In post-order traversal, we visit the nodes in each subtree of a node before visiting the node itself. In in-order traversal, if a node has *m* subtrees, we visit the nodes in the first *m*/2 subtrees, visit the node itself, and then visit the nodes in the remaining *m*/2 subtrees. The run-time complexity of each of the tree-traversal algorithms is O(|*V*|) on a tree with |*V*| vertices.

```
Input: Rooted Directed Tree rd-MST ($V^{rd}$, $E^{rd}$), Root node r-node
Output: PreOrder-Sequence
Auxiliary Variables: Child-Nodes-List, Sensor node u
Initialization: PreOrder-Sequence = $\Phi$
                $u \leftarrow$ r-node
Begin PreOrder-Traversal (u)
    PreOrder-Sequence = PreOrder-Sequence U {u}
    Child-Nodes-List = $\Phi$
    for every edge $u \rightarrow v \in E^{rd}$ do
        Child-Nodes-List = Child-Nodes-List U {v}
    end for
    for every $v \in$ Child-Nodes-List do
        PreOrder-Traversal (v)
    end for
    return PreOrder-Sequence
End PreOrder-Traversal (u)
```

**Fig. 4**. Algorithm for Pre-Order Traversal of the Rooted Directed Minimum-weight Spanning Tree

```
Input: Rooted Directed Tree rd-MST ($V^{rd}$, $E^{rd}$), Root node r-node
Output: PostOrder-Sequence
Auxiliary Variables: Child-Nodes-List, Sensor node u
Initialization: PostOrder-Sequence = $\Phi$
                $u \leftarrow$ r-node
Begin PostOrder-Traversal (u)
    Child-Nodes-List = $\Phi$
    for every edge $u \rightarrow v \in E^{rd}$ do
        Child-Nodes-List = Child-Nodes-List U {v}
    end for
    for every $v \in$ Child-Nodes-List do
        PostOrder-Traversal (v)
    end for
    PostOrder-Sequence = PostOrder-Sequence U {u}
    return PostOrder-Sequence
End PostOrder-Traversal (u)
```

**Fig. 5**. Algorithm for Post-Order Traversal of the Rooted Directed Minimum-weight Spanning Tree

```
Input: Rooted Directed Tree rd-MST ($V^{rd}$, $E^{rd}$), Root node r-node
Output: InOrder-Sequence
Auxiliary Variables: Child-Nodes-List, Sensor node u, Half-List-Size
```

```
Initialization: InOrder-Sequence = $\Phi$
                u ← r-node

Begin InOrder-Traversal (u)
    Child-Nodes-List = $\Phi$
    for every edge u→ v $\in E^{rd}$ do
        Child-Nodes-List = Child-Nodes-List U {v}
    end for
    Half-List-Size = | Child-Nodes-List | / 2
    for every 1 ≤ index ≤ Half-List-Size do
        v ← Child-Nodes-List[index]
        InOrder-Traversal (v)
    end for
    InOrder-Sequence = InOrder-Sequence U {u}
    for every Half-List-Size < index ≤ | Child-Nodes-List | do
        v ← Child-Nodes-List[index]
        InOrder-Traversal (v)
    end for
    return InOrder-Sequence
End InOrder-Traversal (u)
```

**Fig. 6**. Algorithm for In-Order Traversal of the Rooted Directed Minimum-weight Spanning Tree

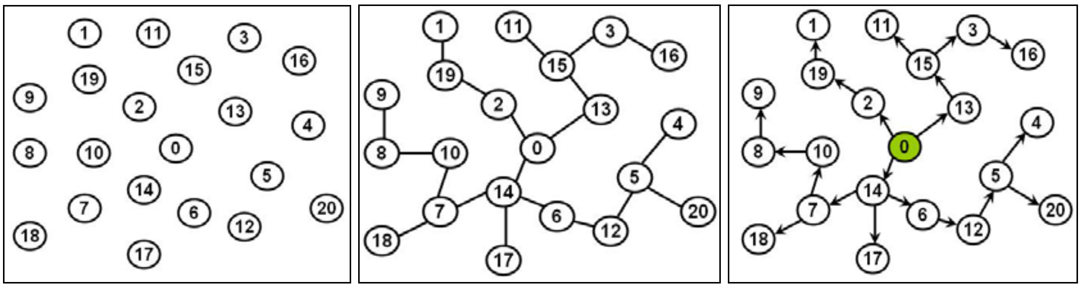

**Fig. 7.1**. Node Distribution

**Fig. 7.2**. Undirected MST

**Fig. 7.3**. Rooted Directed MST (root node: node 0)

**Pre-order chain: [0, 2, 19, 1, 13, 15, 3, 16, 11, 14, 6, 12, 5, 4, 20, 7, 10, 8, 9, 18, 17]**

**In-order chain: [2, 19, 1, 0, 13, 3, 16, 15, 11, 6, 12, 4, 5, 20, 14, 10, 8, 9, 7, 18, 17]**

**Post-order chain: [1, 19, 2, 16, 3, 11, 15, 13, 4, 20, 5, 12, 6, 9, 8, 10, 18, 7, 17, 14, 0]**

**Fig. 7**. Example to Illustrate the Generation of Node Chain using the Tree Traversal Algorithms

## 4. Simulations

We conducted all of our simulations in a discrete-event simulator developed in Java. This simulator has also been recently used [6][7][15] by us to successfully report simulation results for data aggregation in sensor networks. We conducted the simulations with square network fields of dimensions 100m x 100m and 50m x 50m. The number of nodes used in both these network fields is 100. The sink is located outside the network field, at co-ordinates (50, 300) for the 100m x 100m network field and at co-ordinates (25, 150) for the 50m x 50m network field. Simulations have been conducted for both TDMA and CDMA systems. Each sensor node is assumed to be capable of conducting transmission power control: i.e., the sensor node will be able to adjust its transmission range depending on the distance to the receiver node. For CDMA systems, each sensor node has a unique CDMA code and it is assumed to be known to all the other sensor nodes. The initial energy supplied to each node in all of our simulations is 1J. The size of the data packet is 2000 bits. We assume that an aggregating node fuses its own data with the data collected from its peer node in the chain and sends a data packet of the same size to the next node in the chain. In other words, the size of the data packets does not increase with data aggregation.

We use the following first order radio model [16] that has been also previously used (e.g., [2][3][6][7][15]) to model energy consumption. According to this model, the energy expended by a radio to run the transmitter or receiver circuitry is $E_{elec}$ = 50 nJ/bit and $\in_{amp}$ = 100 pJ/bit/m$^2$ for the transmitter amplifier. The radios are turned off when a node wants to avoid receiving unintended transmissions. An $r^2$ energy loss model is used to compute the transmission costs. The energy lost in transmitting a $k$-bit message over a distance $d$ is given by: $E_{TX}(k, d) = E_{elec} * k + \in_{amp} * k * d^2$. The energy lost in receiving a $k$-bit message is $E_{RX}(k) = \mathrm{E}_{\mathrm{elec}} * k$. The cost of fusion is 5 nJ/bit/message.

The performance metrics measured are: (i) Node Lifetime, (ii) Energy Lost per Node and (iii) Energy Lost per Round. The simulation results presented in **Fig. 8.1 - 8.4**, **9.1 - 9.4** and **10.1 - 10.4** are average values obtained for 1000 trials under each of the different combinations of simulation conditions presented above. The Node Lifetime, sometimes considered in the literature as a measure of also the network lifetime [1][2], is the number of successful rounds of data gathering that have been completed at the time of the first node failure in the network due to the exhaustion of battery charge. The Energy Lost per Node is the energy lost for transmission, reception and fusion of data, averaged over all the nodes for each trial until the round of first node failure. The Energy Lost per Round is the sum of the energy lost at all the nodes per round for transmission, reception and fusion of data, averaged over all the rounds of a trial until the round of first node failure. For sake of clarity, we refer to the version of PEGASIS that uses the node chain obtained from the distance-based greedy heuristic as PEGASIS-original and the versions of PEGASIS that use the node chain obtained from the tree traversal algorithms as PEGASIS-preOrder, PEGASIS-inOrder and PEGASIS-postOrder.

## 4.1 Node Lifetime

For all the scenarios studied in this paper, the values for the node lifetime obtained for PEGASIS based on any of the three tree traversal algorithms are greater than those obtained for PEGASIS-original. The maximum relative improvement in the round of first node failure can be as large as 30% (obtained for post-order traversal on 100m x 100m network, TDMA system). The distance-based greedy heuristic has been observed to yield network chain, wherein the physical distance between the neighboring nodes in the chain gets significantly larger as we progress along the length of the chain. Especially, the physical distance between the neighboring nodes in the second half of the chain is significantly larger than the physical distance between the neighboring nodes in the first half of the chain. As a result, nodes in the second half of the chain have to transmit data over a longer distance compared to the nodes in the first half of the chain. This triggers the premature node failures of the nodes in the second half of the chain compared to the nodes in the first half of the chain. With the tree traversal algorithms, especially the post-order and in-order traversal algorithms, the nodes are relatively more equally spaced in the resulting chain and thus the energy lost per node does not vary significantly along the chain.

The choice of root node for the *rd-MST* could also be attributed to the improvement in node lifetime. In PEGASIS-original, there is no concept of using a rooted tree and the starting node of the chain is the node that is farthest away from the sink location. Our approach of choosing the root node to be the node that is physically closest to the center of the network field yields a tree with greater depth (average depth of a 100-node *rd-MST* observed in our simulations is 34.0). Consequently, the physical distance between a majority of node pairs sharing an edge in the *rd-MST* and the node chain is not significantly larger.

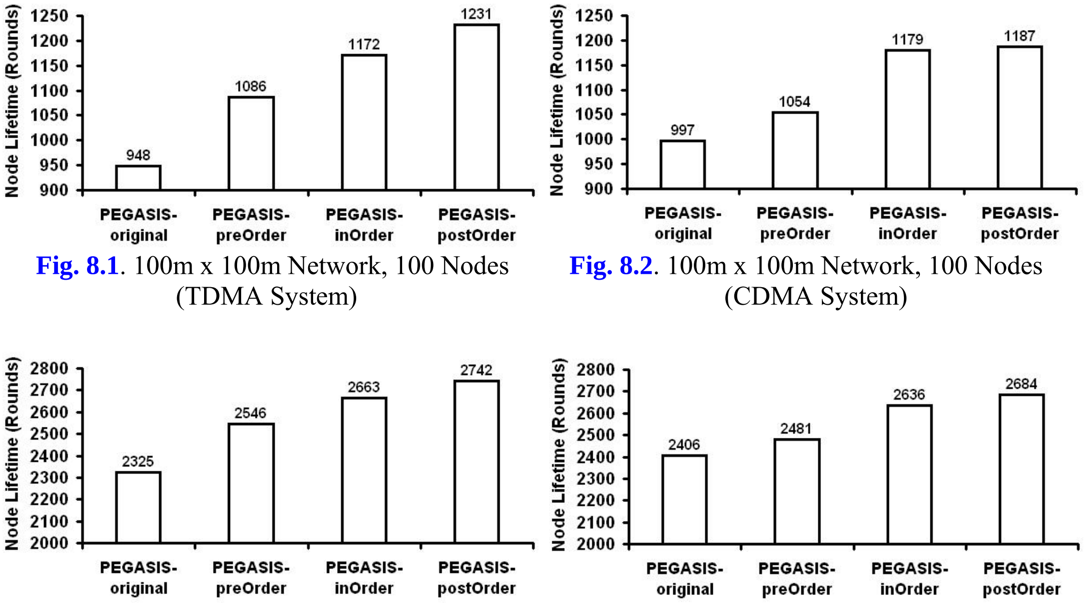

**Fig. 8.1**. 100m x 100m Network, 100 Nodes (TDMA System)

**Fig. 8.2**. 100m x 100m Network, 100 Nodes (CDMA System)

**Fig. 8.3**. 50m x 50m Network, 100 Nodes (TDMA System)

**Fig. 8.4**. 50m x 50m Network, 100 Nodes (CDMA System)

**Fig. 8**. Node Lifetime (Round of First Node Failure) for TDMA and CDMA Systems

In terms of the absolute values, the chain obtained through a post-order traversal of the *rd-MST* yields the largest value for node lifetime in all the four scenarios shown in **Fig. 8**. The node chain obtained through a pre-order traversal of the *rd-MST* yields the lowest value (among the three traversal algorithms) for the round of first node failure in all the four scenarios. Nevertheless, for a given simulation scenario, the absolute values for node lifetime obtained for PEGASIS based on any of the three traversal algorithms is larger than that obtained with PEGASIS-original. As we quadruple the area of the network from 2,500 m$^2$ (50m x 50m) to 10,000 m$^2$ (100m x 100m), the node lifetime decreases by a factor of 2.2 to 2.5, for both TDMA and CDMA systems. In a 100m x 100m network, the 100 nodes are more sparsely distributed compared to the case of a 50m x 50m network. Hence, the nodes have to communicate over a relatively longer distance in a 100m x 100m network compared to that in a 50m x 50m network.

With respect to the impact of TDMA vs. CDMA, we observe that for a given simulation scenario, PEGASIS-original yields a slightly larger value (about 5%) for node lifetime in CDMA systems compared to TDMA systems. On the other hand, PEGASIS based on the node chain generated from the tree traversal algorithms is more likely to yield a larger value for the round of first node failure in TDMA systems compared to CDMA systems. For a given network field, the relative improvement in node lifetime brought about by the three tree traversal algorithms vis-à-vis the distance-based greedy heuristic is greater for TDMA systems (30% for 100m x 100m network and 18% for 50m x 50m network) compared to CDMA systems (19% for 100m x 100m network and 12% for 50m x 50m network).

## 4.2 Energy Lost per Node

For all the simulation scenarios studied in this paper, PEGASIS-original incurs the largest energy loss per node compared to PEGASIS based on all the three tree traversal algorithms. Among the tree traversal algorithms, PEGASIS chain constructed based on pre-order traversal resulted in the largest energy loss per node and the PEGASIS chain constructed based on post-order traversal yielded the lowest energy loss per node. All of these observations vindicate the improvement in node lifetime as a result of using the tree traversal algorithms for chain formation in PEGASIS vis-à-vis the distance-based greedy heuristic.

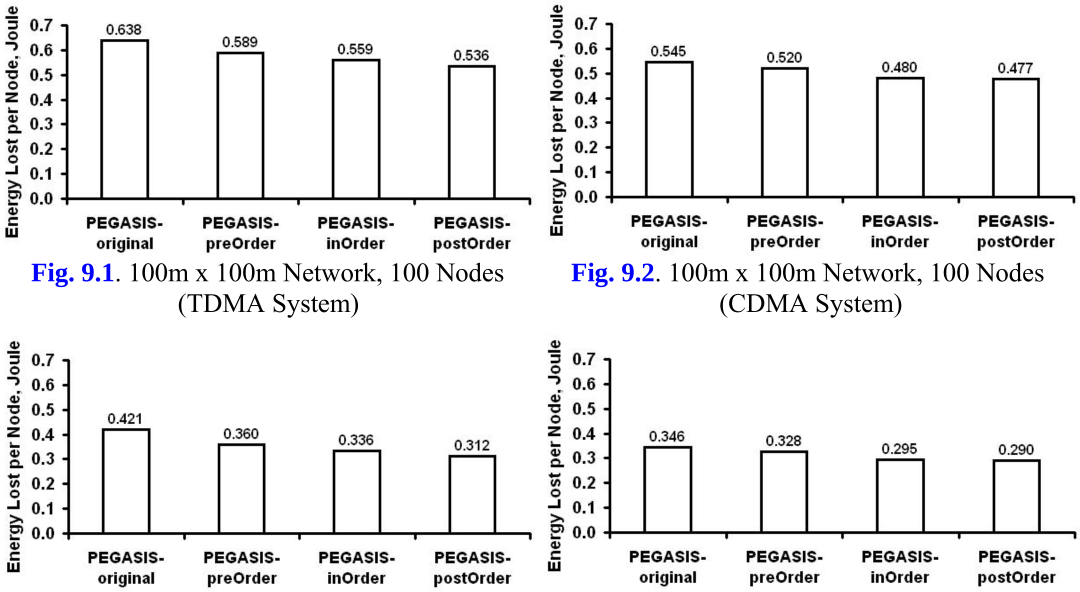

**Fig. 9.1**. 100m x 100m Network, 100 Nodes (TDMA System)

**Fig. 9.2**. 100m x 100m Network, 100 Nodes (CDMA System)

**Fig. 9.3**. 50m x 50m Network, 100 Nodes (TDMA System)

**Fig. 9.4**. 50m x 50m Network, 100 Nodes (CDMA System)

**Fig. 9**. Energy Lost per Node for TDMA and CDMA Systems

The significantly larger energy loss per node with the distance-based chain of PEGASIS-original can be attributed to the failure of the greedy approach in minimizing the node-node distance as we progress along the chain. Greedy approach for algorithm design very often yields sub-optimal solutions [8] and the distance-based greedy heuristic for constructing the node chain is another example to justify this fact (Dijsktra algorithm [8] for shortest path tree, Prim's and Kruskal algorithms [8] for minimum-weight spanning tree and the *OptTreeTrans* algorithm [17] for minimizing the number of multicast tree transitions in mobile ad hoc networks are some of the few examples of tree-related algorithms where the greedy strategy yields optimal solution).

For a 100m x 100m network field, the energy lost per node for PEGASIS-original has been observed to be 8%-19% (for TDMA systems) and 5%-14% (for CDMA systems) larger than that of the PEGASIS versions based on tree traversals. The difference increases as we increase the network density (i.e., as we reduce the size of the network field). For a 50m x 50m network field, the energy lost per node for PEGASIS-original has been observed to be 17%-35% (for TDMA systems) and 6%-19% (for CDMA systems) larger than that of the PEGASIS versions based on tree traversals.

For a given network field (i.e., for fixed network dimensions), for each of the four PEGASIS versions, the absolute value for the energy lost per node for TDMA systems is more than the energy lost per node for CDMA systems. The difference in the energy lost per node for TDMA systems and CDMA systems is the greatest for PEGASIS-original (as large as 21%) and lowest for PEGASIS-postOrder (as low as 7%). We also observe that for both TDMA and CDMA systems and for each of the four PEGASIS versions, as we reduce the size of the network field from 100m x 100m to 50m x 50m and place the sink appropriately closer to the network field, the energy lost per node is reduced by 33%-40%.

The relatively larger node lifetime and lower energy loss per node observed for the tree-traversal based PEGASIS versions in the simulations can be theoretically justified by the observation that a majority of the links (i.e., 66%-78% of the links in our simulations) in the node chain generated as a result of the tree traversals are also edges in the *ud-MST* based on which the *rd-MST* and the node chain are generated. As the *ud-MST* is the minimum-weight spanning tree that includes edges whose physical length is significantly smaller, the physical distance between two successive nodes in the chain generated by the tree traversals is very rarely a large value. Thus, the transmission energy loss per node for the data packets is relatively low compared to that incurred with the distance-based chain of the original PEGASIS.

## 4.3 Energy Lost per Round

For a given simulation scenario, the energy lost per round is almost the same for PEGASIS-original and the versions of PEGASIS based on the three traversal algorithms. The difference in magnitude is within 2%. With no appreciable difference in the energy consumed per round, the relatively low values for node failure in the case of PEGASIS-original indicate the unfairness of node usage associated with the distance-based chain compared to the versions of PEGASIS based on the tree traversal algorithms. Nodes in the later half of the distance-based chain lose more energy compared to nodes in the first half of the chain. On the other hand, in the case of the node chains formed as a result of tree traversals, the energy consumed per node during a round of data gathering is not significantly different among the nodes and is also lower compared to that obtained from the node chain formed using the distance-based greedy heuristic (refer Section 4.2).

For a given network field and for all of the PEGASIS versions studied, the energy consumed per round for CDMA systems is greater than that obtained for TDMA systems. This can be attributed to the fact that the nodes that communicate along the binary tree (formed from the node chain) may not be close enough to each other. The difference is more predominant in the 100m x 100m network as it has a relatively larger area and nodes are more sparsely distributed. The energy consumed per round for a CDMA system is 21% greater than that obtained for a TDMA system in a 100m x 100m network; whereas, the energy consumed per round for a CDMA system is only 8% greater than that obtained for a TDMA system in a 50m x 50m network. Similarly, for a given technology (TDMA or CDMA), the energy consumed per round is greater for the larger network. For a TDMA system, the energy consumed per round in a 100m x 100m network is 50% more than that obtained in a 50m x

50m network; whereas, for a CDMA system, the energy consumed per round in a 100m x 100m network is 66% more than that obtained in a 50m x 50m network.

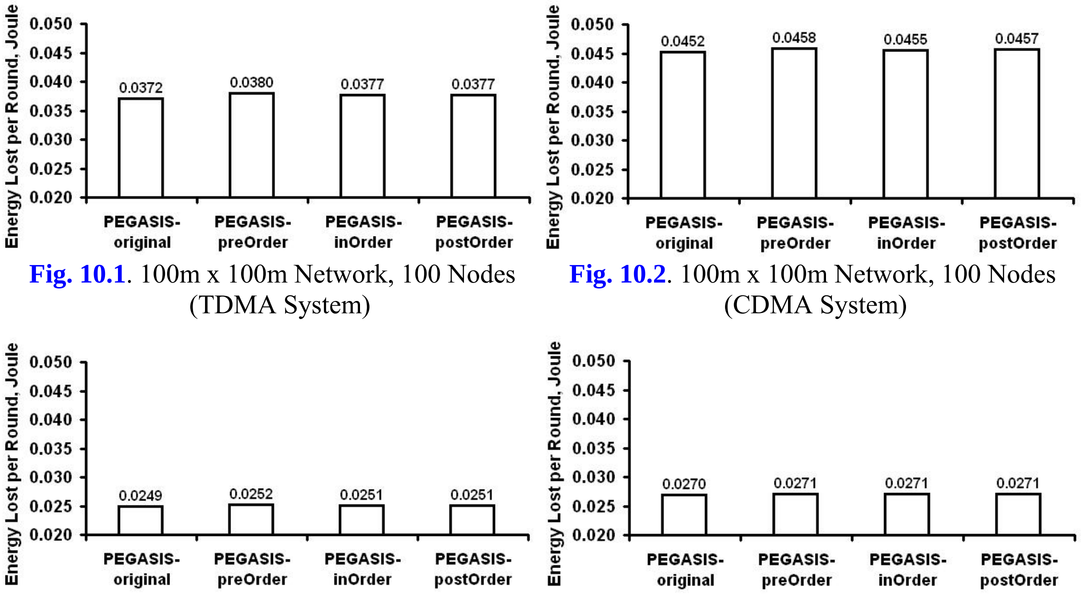

**Fig. 10.1**. 100m x 100m Network, 100 Nodes (TDMA System)

**Fig. 10.2**. 100m x 100m Network, 100 Nodes (CDMA System)

**Fig. 10.3**. 50m x 50m Network, 100 Nodes (TDMA System)

**Fig. 10.4**. 50m x 50m Network, 100 Nodes (CDMA System)

**Fig. 10**. Energy Lost per Round of Data Gathering for TDMA and CDMA Systems

## 5. Conclusions

The simulation results illustrate the effectiveness of employing the tree traversal algorithms to construct the chain of nodes for the PEGASIS data gathering protocol. The node lifetime can be improved as large as by 30% for TDMA systems and 19% for CDMA systems, using the tree traversal algorithms as the basis for constructing the PEGASIS node chain. At the same time, the energy lost per node obtained with the node chains based on tree traversals can be 35% and 19% lower than that obtained with the distance-based greedy heuristic for TDMA and CDMA systems respectively. Each of the three traversal algorithms (pre-order, in-order and post-order traversals) yield an improvement in node lifetime and a reduction in the energy lost per node compared to the distance-based greedy heuristic employed in the original version of the PEGASIS protocol. Among the three traversal algorithms, the post-order traversal approach yields the largest improvement in the node lifetime and largest reduction in the energy lost per node, while the pre-order traversal approach yields the least improvement in node lifetime and lowest reduction in the energy lost per node. The energy lost per round is almost the same for PEGASIS-original and the versions of PEGASIS based on the three traversal algorithms. So, overall, without any additional network-wide energy consumption, we can employ the tree traversal algorithms for forming the PEGASIS node chain and significantly enhance the node lifetime as well as reduce the energy lost per node.

## References

[1] W. Heinzelman, A. Chandrakasan and H. Balakarishnan, "Energy-Efficient Communication

Protocols for Wireless Microsensor Networks," *Proceedings of the Hawaaian International Conference on Systems Science*, Jan. 2000.

[2] S. Lindsey, C. Raghavendra and K. M. Sivalingam, "Data Gathering Algorithms in Sensor Networks using Energy Metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924-935, Sep. 2002.

[3] S. Lindsey, C. Raghavendra and K. M. Sivalingam, "Data Gathering in Sensor Networks using the Energy*Delay Metric," *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, pp. 2001-08, Apr. 2001.

[4] R. Osso, "Handbook of Emerging Communications Technologies: The Next Decade," 1st edition, CRC, ISBN: 0849395941, Aug. 1999.

[5] A. J. Viterbi, "CDMA: Principles of Spread Spectrum Communication," 1st edition, Prentice Hall, ISBN: 0201633744, Apr. 1995.

[6] N. Meghanathan, "An Algorithm to Determine Energy-aware Connected Dominating Set and Data Gathering Tree for Wireless Sensor Networks," *Proceedings of the 2009 International Conference on Wireless Networks*, pp. 608-614, Las Vegas, USA, July 13-16, 2009.

[7] N. Meghanathan, "Grid Block Energy based Data Gathering Algorithm for Lower Energy*Delay and Longer Lifetime in Wireless Sensor Networks," *Proceedings of the ISCA First International Conference on Sensor Networks and Applications*, San Francisco, USA, Nov. 2009.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms," 2nd Edition, MIT Press, 2001.

[9] S-M. Jung, Y-J. Han and T-M. Chung, "The Concentric Clustering Scheme for Efficient Energy Consumption in the PEGASIS," *Proceedings of the 9th International Conference on Advanced Communication Technology*, vol. 1, pp. 260-265, Gangwon-Do, South Korea, Feb. 2007.

[10] J-E. Lee and K. Kim, "Diamond-Shaped Routing Method for Reliable Data Transmission in Wireless Sensor Networks," *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 799 – 801, Sydney, Australia, Dec. 2008.

[11] N. Tabassum, Q. E. K. M. Mamun and Y. Urano, "COSEN: A Chain Oriented Sensor Network for Efficient Data Collection," *Proceedings of the 3rd International Conference on Information Technology: New Generations*, pp. 262-267, Las Vegas, USA, Apr. 2006.

[12] I. Shukla and N. Meghanathan, "Impact of Leader Selection Strategies on the PEGASIS Data Gathering Protocol for Wireless Sensor Networks," accepted for publication in *Ubiquitous Computing and Communication Journal*.

[13] H. O. Tan and I. Korpeoglu, "Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks," *SIGMOD Record*, vol. 32, no. 4, pp. 66-71, 2003.

[14] H. O. Tan, I. Korpeoglu and I. Stojmenovic, "A Distributed and Dynamic Data Gathering Protocol for Sensor Networks," *Proceedings of the 21st IEEE International Conference on Advanced Networking and Applications*, pp. 220-227, Niagara Falls, Canada, May 2007.

[15] L. King and N. Meghanathan, "A Weighted-Density Connected Dominating Set Data Gathering Algorithm for Wireless Sensor Networks," *Journal of Computer and Information Science*, vol. 2, no. 4, pp. 3-13, Nov. 2009.

[16] T. S. Rappaport, "Wireless Communications: Principles and Practice," 2nd edition, Prentice Hall, ISBN: 0130422323, Jan. 2002.

[17] 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.

**Natarajan Meghanathan** is an Assistant Professor in the Department of Computer Science at Jackson State University, Jackson, MS, USA. He received his MS and Ph. D in Computer Science from Auburn University, AL and The University of Texas at Dallas in August 2002 and May 2005 respectively. His main areas of research are wireless ad hoc networks and sensor networks. He has more than 50 peer-reviewed publications in leading international journals and conferences in these two areas together. He has also recently received grants as Principal Investigator from the Army Research Laboratory (ARL) and National Science Foundation (NSF). Besides ad hoc networks, Meghanathan is currently conducting active research in the areas of graph theory, distributed systems, network security and bioinformatics. He serves as the editor of a number of international journals and also in the technical program committees and organization committees of several leading international conferences in the area of networks.