A Low-Latency and Energy-Efficient Algorithm for Convergecast in Wireless Sensor Networks

S. Upadhyayula, V. Annamalai, S. K. S. Gupta
Dept. of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287
(sandeep.gupta, sarma, valliappan)@asu.edu

***Abstract*— In Wireless Sensor Networks (WSN) the process of dissemination of data among various sensors (*broadcast*) and collection of data from all sensors (*convergecast* or *data aggregation*) are common communication operations. With increasing demands on efficient use of battery power, many efficient broadcast tree construction and channel allocation algorithms have been proposed. Generally convergecast is preceded by broadcast. Hence the tree used for broadcast is also used for convergecast. Our research shows that this approach is inefficient in terms of latency and energy consumption.**

In this paper we propose a heuristic solution for the problem of *minimum energy convergecast* which also works toward *minimizing data latency*. This algorithm constructs a tree using a greedy approach where new nodes are added to the tree such that weight on the branch to which it is added is less. The algorithm then allocates Direct Sequence Spread Spectrum or Frequency Hopping Spread Spectrum codes. Simulation results show that energy consumed and communication latency of our approach is lower than some of the existing approaches for convergecast. We have then used our algorithm to perform broadcast. Surprisingly our results show that this algorithm's performance for broadcasting is better compared to other broadcast techniques.

I. INTRODUCTION

Miniaturization of wireless sensor devices makes them power anemic. So, examination of inefficient energy expenditure indicates that sensor communication is a critical feature. The principal role of wireless sensors is collecting data from its environment. A survey of literature indicates the following research work in this area [1][3][4]. In [1] the algorithm proposed for tree construction adopts a greedy approach and is energy efficient. Simulation results show that latency of this algorithm is very high. Since, data aggregation or convergecast usually follows broadcast, the tree constructed for broadcast is used for convergecast. Our analysis shows that this approach is inefficient in terms of energy consumption and latency of communication. In this paper we address the problem of energy-efficient convergecast communication by proposing a CDMA/TDMA based algorithm that constructs a tree and schedules its nodes for collision-free transmission. In this work, latency, energy and reliability are used as metrics for measuring efficiency of the proposed method. Simualation results show that energy consumed and latency of our approach is less compared to protocol proposed in [1]. Our approach also performs better than reusing a tree constructed by broadcast approach of [2].

II. PROBLEM DESCRIPTION

A. Problem Definition

The problem of *convergecast* can be formally defined as follows: In an environment consisting of wireless sensor nodes, each node i has a unit of data which has to be transmitted to the base station. Each node has a transmission range that can be adjusted upto a maximum value of r. All nodes within the range are considered to be its neighbors. A sensor node, if within the range, can transmit its data to the base station directly (single hop) or through another node (multihop). Energy expended by a node i is a function of the amount of data units d_i to be transmitted and distance of transmission r_i: $E_i = f(d_i, r_i)$. In a multihop transmission each node in the path fuses its data along with the data it has received and transmits it to its parent node in the tree.

B. Goals of Communication

Total energy expended by the network is defined as the sum of energies used by each node ($E_{total} = \sum_{i \in N} E_i$). Data latency (or latency) is the total time taken for one round of data transmission from each node in the network to the base station. Wireless sensor devices are scarce on battery power. In wireless networks energy consumed for communication is very high. Therefore, one of the design goals is to minimize energy consumption during coomunication. Faster data transfer ensures low-latency. Reliability during transmission is also desirable. Otherwise collision may occur that requires retransmission and, hence, more demand on battery power. In this paper we try to address all the three issues namely energy-efficiency, latency and reliability.

C. Energy-Latency Trade-off

In a network consisting of n nodes, the average number of data units (D_{avg}) transmitted and the average distance of transmission (R_{avg}) is given by

$$D_{avg} = \frac{\sum_{i \in V} d_i}{n} \text{ and } R_{avg} = \frac{\sum_{i \in V} \ r_{i,parent(i)^2}}{n} \quad (1)$$

where d_i is data unit transmitted by node i, $r_{i,parent(i)}$ is the distance between transmitting node i and and its parent node $parent(i)$. V is the set of nodes in the network whose size is n.

From [6], energy loss in a free space for RF communication has a quadratic dependence on the transmission distance and a linear dependence on the number of data bits transmitted. In single hop communication energy spent by a node i is given as $f(r_{i,BS}, d_i)$ and in multi hop it is given as $f(r_{i,parent(i)}, \sum_{k \in children(i)} d_k \ + \ d_i)$. In multi hop on an average $r_{i,parent(i)}$ is less compared to $r_{i,BS}$. But the amount of data it transmits is high. In multi hop the energy gain due to reduction in R_{avg} is negated by the energy loss due to increase in average data bits transmitted D_{avg}. Therefore, depending on distance transmitted and size of data either of communication can be usefull.

III. System Model

The problem of convergecast communication is addressed in this paper using the following system model. The system consists of n wireless sensors each equipped with a transceiver that can transmit data up to a maximum range of r. All the sensors that are in communicating range of each other are called neighbors.

We construct a graph $G(V, E)$ from the network where nodes represent sensors, arcs between two nodes indicates that they are neighbors of each other and the weights on the arc represents the communication cost. We assume that the network is synchronized and the nodes are static. All nodes in the graph have a unit of data to be transmitted to base station. As mentioned in Section II, data can be transmitted in a single hop or in multiple hops.

A. Rationale For Tree Construction

Generally, direct transmission of sensed data to base station will drain the energy of the node quickly [4]. Since in free space, energy dependence on transmission range is quadratic compared to number of data bits transfered (which is linear) [6], we choose multihop transmission as our model of communication. In this model intermediate nodes concatenates the data packets it receives from other nodes. If a node is in the path of multiple routes, it will wait until it has received data from all the nodes. We achieve multihop transmission by constructing a tree. Then each node in the tree is allocated a code,timeslot pair called a **channel**. Channel assigned to a node indicates the time slot in which it will transmit data and the code it will use while transmitting data. The tree construction algorithm follows an iterative greedy approach determining parent-child relationship among nodes.

B. Assumptions

Each node will have two codes for communication. One for transmission and another for reception. For any node in the network, its reception code is same to the transmission code of its children. Each node has only one transceiver. So, in a given time slot a parent cannot receive messages from more than one child. A node can either transmit or receive messages but cannot do both at the same time. A node has two states: sleep and active state. Prior to the scheduled time slot for communication, a node switches from the sleep state (energy conserving state) to the active state, transmits or receives data and then goes back to the sleep state.

C. Energy Model

We use the model proposed in [3] to describe energy expended in the whole network

$$\begin{aligned} E_{total} &= \sum_{i \in V} E_i \quad \text{where,} \\ E_i &= E_{Tx} + E_{Rx}, \\ E_{Tx} &= E_{elec} \times k + \epsilon_{amp} \times k \times r^2 \quad \text{and} \\ E_{Rx} &= E_{elec} \times k \end{aligned} \quad (2)$$

where k is number of bits received by node i, r is the transmission distance, E_{Tx} and E_{Rx} are the energy consumed for transmission and reception respectively. Electrical energy required to run the circuitry of transmitter and receiver are same(E_{elec}) and ϵ_{amp} is the amplification energy for transmission.

D. Latency Model

A node, during its time slot, transmits its data using the code it was alloted. If d_{c_i} represents data transmitted by child c_i then the length of time slot can be defined using

$$D = max\{d_{c_1}, d_{c_2}, \ldots, d_{c_n}\} \quad (3)$$

where $c_1, c_2, \ldots, c_n$ are children of root node. Let T_D be time required to transmit D units of data. T_D determines the length of one time slot. Therefore, for a node with k children, at least kT_D time is required for the node to receive data from all its children. For convergecast, latency of communication is defined as time taken from start of transmission from leaf nodes until all the data is received by base station. For broadcast, latency is time taken for the message transmitted by the root node to reach all the nodes in the network. Latency in broadcast is determined by latency of critical path - path along which time to deliver packets is the longest. In convergecast, latency depends on the number of parallel transmissions. Higher the number, lower is the latency.

We improve latency for convergecast by constructing a balanced tree. Motivation for balanced tree construction is that it enhances the likelihood of multiple simultaneous transmissions in a given timeslot. Multiple simultaneous transmissions might

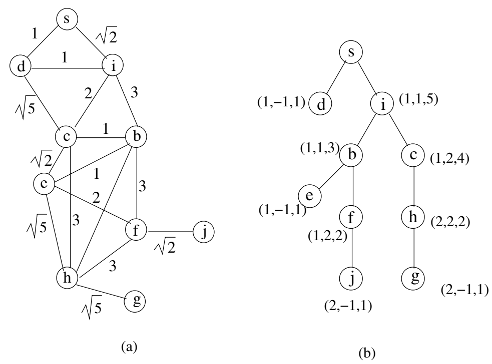


Fig. 1. (a) An Example Graph (b) Tree with nodes scheduled. The 3-tuple indicates (Transmission Code, Reception Code, Time Slot)

lead to collisions. These collisions can be avoided by allocating channels. To ensure that the tree is balanced, we introduce β-**rule** which states that a node cannot have more than β children. However, if an intended child node is likely to be left out of the tree due to β-rule, then the rule is overlooked.

IV. Algorithms for Tree Construction and Channel Allocation

A. Tree Construction Algorithm

Given a topology graph $G(V, E)$, our algorithm begins with the root node and starts iteratively establishing parent-child relationship with other nodes. It makes use of the notations shown in Table I.

Algorithm 1 Tree Construction(G,s)

while $\mathcal{P} \neq \emptyset$
  for all $c \in \mathcal{C}_{poss}$
    $d = \infty$
    for all $p \in \mathcal{P}$
      if $\delta(p, c) < d \wedge (|\mathcal{C}_p| < \beta \vee |Adj(c)| = 1)$
        $d = \delta(p, c)$
        $\mathcal{C}_{parent(c)} = \mathcal{C}_{parent(c)} - \{c\}$
        $parent(c) = p$
        $\mathcal{C}_p = \mathcal{C}_p \cup \{c\}$
        $update(Adj(c)),\ update(Adj(parent(c)))$
    $\mathcal{C} = \mathcal{C} \cup \{c\}$
  $\mathcal{P} = \mathcal{C},\ \mathcal{C} = \emptyset,\ update(\mathcal{C}_{poss})$

Working of Algorithm 1 is explained using Figure 1. C_{poss} consists of all the neighbors of nodes in $\mathcal{P}$. For the graph in Figure 1, $\mathcal{P} = \{a\}$ and $\mathcal{C}_{poss} = \{d, i\}$ and β = 2 (user input). Elements of $\mathcal{C}_{poss}$ are intended children of elements of $\mathcal{P}$. Each element $c \in \mathcal{C}_{poss}$ is compared with each element $p \in \mathcal{P}$. Since a, the source node, is the only node in set $\mathcal{P}$, d becomes child of a. Similarly for node i. For next iteration, $\mathcal{P} = \{d, i\}$ and $\mathcal{C}_{poss}$ = $\{b, c\}$. Node c will become child of node i since it is closer to i than d.

TABLE I
NOTATIONS FOR TREE CONSTRUCTION ALGORITHM

Notation	Definition	Initial Value
$\mathcal{P}$	$\{ p \mid p \in V\} \wedge \mathcal{P} \subset V$	$\{s\}$, root node
$\mathcal{C}$	$\{ c \mid c \in V$ $\wedge parent(c) \in \mathcal{P}\}$	$\emptyset$
$Adj(p)$	Adjacent nodes of p that are NOT part of the tree	-
$\mathcal{C}_{poss}$	$\{ c \mid c \in Adj(p) \forall p \in \mathcal{P}\}$	$Adj(s)$
$\mathcal{C}_i$	$\{ c \mid parent(c) = i$ $\forall\, c, i \in V\}$	$\emptyset$
β	A positive integer such that $\lvert\mathcal{C}_i\rvert = \beta$	Input
$\delta(u, v)$	Distance between u and v	-
$update(X)$	Update the set X reflecting their definitions	
$Rx(p)$	Reception code of p	-
$Tx(p)$	Transmission code of p	-
$\Gamma(p)$	Time slot for transmission	-
Π_i	$\{ p \mid Rx(p) = i\}$	$\emptyset$
π_i	$\{ p \mid Tx(p) = i\}$	$\emptyset$
$\mathcal{L}$	$\{ i \mid \mathcal{C}_i = \emptyset$ or all children of i are assigned a channel$\}$	Input
Θ	Total number of codes	Input

β-rule implementation can be seen in next iteration. When $\{b, c\}$ becomes parent set then C_{poss} = $\{e, f, h\}$. Node e and f will be children of b. But when h is examined it is closer to b. However, since b does not satisfy β-rule, h will become child of c. This algorithm continues until all the nodes in the graph joins the tree. In this approach, during each iteration, no node in C_{poss} goes without being assigned a parent.

B. Channel Allocation Algorithm

Communication among parent-child pairs should be performed with minimum interference among different transmissions. To achieve this we use CDMA technology. We propose an algorithm that takes a tree and number of codes as input and allocates channel. This algorithm makes use of the notations shown in Table I.

Algorithm 2 NodeInfo($u, i, \mathcal{L}_{new}, time$)

$Tx(u) = i$
$\Gamma(u) = time$
for all $v \in Adj(u) \wedge \delta(u, v) \leq \delta(u, parrent(u))$
  $\Pi_i = \Pi_i \cup \{v\}$
for all $v \in Adj(parent(u)) \wedge \delta(v, parent(u)) \leq \delta(v, parent(v))$
  $\pi_i = \pi_i \cup \{v\}$
if all children of u are scheduled
  $\mathcal{L}_{new} = \mathcal{L}_{new} \cup \{u\}$

Algorithm 3 Channel Allocation($\mathcal{L}$)

var: $\mathcal{L}_{new} = \emptyset$
int: $code$,i, $time = 1$
while $\mathcal{L} \neq \emptyset$
  for all $u \in \mathcal{L}$
    if $Rx(parent(u)) > 0$
      $i = Rx(parent(u))$
      if $u \notin \pi_i \wedge parent(u) \notin \Pi_i$
        NodeInfo($u, i, \overline{\mathcal{L}}, time$)
      else
        $\mathcal{L}_{new} = \mathcal{L}_{new} \cup \{u\}$
    else
      for all $0 \leq i \leq \Theta$
        if $u \notin \pi_i \wedge parent(u) \notin \Pi_i$
          $Rx(parent(u) = i$
          NodeInfo($u, i, \mathcal{L}_{new}, time$)
      if $i > \Theta$
        $\mathcal{L}_{new} = \mathcal{L}_{new} \cup \{u\}$
  $\mathcal{L} = \mathcal{L}_{new}$, $\mathcal{L}_{new} = \emptyset$, $time = time + 1$
  $\pi_i = \Pi_i = \emptyset \, \forall \, 0 \leq i \leq \Theta$

Tree constructed by earlier algorithm is shown in Figure 1(b). This tree is taken as input by the algorithm. The set of leaf nodes $\{d, e, j, g\}$ are examined for channel allocation. These nodes can be assigned in any order. Since we have used linked lists to store information of leaf nodes, we schedule them from left to right in the tree. Node d will be assigned first code that is available (code 1). Node a, parent of d, and all other neighbors of d within the hearing range of $\delta(a, d)$ will be added to the set π_1. All the neighbors of a will be added to the set Π_1. Significance of set Π_i is that nodes $p \in \Pi_i$ have one of their neighbor transmitting data using code i. If p listens for data from child node then it will receive a garbled message due to interference from other transmissions. Similarly, $c \in \pi_i$ have one of their neighbors receiving data with i^{th} code. If c starts transmitting its data with code i then it will cause interference to other receivers. When a parent-child pair is examined for channel, the child node should not be present in π_i and parent node should not be present in Π_i. If both these conditions are met then that pair can communicate using code i unless parent node has already decided upon a channel from its earlier child node. Complete schedule of the tree generated by our algorithm is shown in Figure 1(b)

V. Simulation Results

In the following figures, energy consumptions of the three algorithms ([1][2] and CCA) are plotted. [2] constructs broadcast tree by choosing child nodes in a collision free manner. [1]'s tree construction follows a greedy approach in child node selection. CCA's tree construction algorithm is an improvement on [1] which is described in section IV.

In our experiments, among the three algorithms, CCA consumes least energy and [2] consumes most. For a fixed network size, all the algorithms cosume less energy with the decrease in node density. The reason is: for high density graphs, average link weight decreases in the constructed tree. Therefore, for same number of nodes, denser graphs consume less energy. However, the relationship among the three algorithms still remain the same with CCA consuming least energy and [2] consuming maximum energy.

In terms of latency, performance of [1] is poorest followed by [2]. CCA performs convergecast in least time units. With the change in node density and number of nodes, latencies due to CCA and [1] remains mostly unchanged. For a fixed number of nodes, latency due to [2] increases with node density. This behavior can be explained as follows: with increase in node density, [2]'s tree construction algorithm finds it difficult to schedule nodes because of high interference. [1] performance remains unaffected because of the *cluttering effect* of the tree construction algorithm. If a node is in vicinity of many nodes, then that node is likely to become parent of all the nodes. This will result in one node having many children, thereby *cluttering* the communication space. Due to this behavior latency of [1] will remain unchanged. CCA, due to its balanced tree construction property, is able to accomodate variations in density and size of network without effecting latency.

We then investigated performance of our algorithm when it is used for broadcast communication. Figure V(e-h) shows the ratio of the output of energy consumption and data latency of [2] and our algorithm. In notation $\frac{E_{b,b}}{E_{b,c}}$ used in Figure V(e,g) numerator $E_{b,b}$ indicate the energy expended when a broadcast is performed on a broadcast tree. $E_{b,c}$ indicate energy consumed when a broadcast is performed on a tree constructed for convergecast. Apparently the ratio is higher than 1 in all the plots. Thus even for broadcast communication our algorithm is energy and latency efficient.

VI. Conclusion

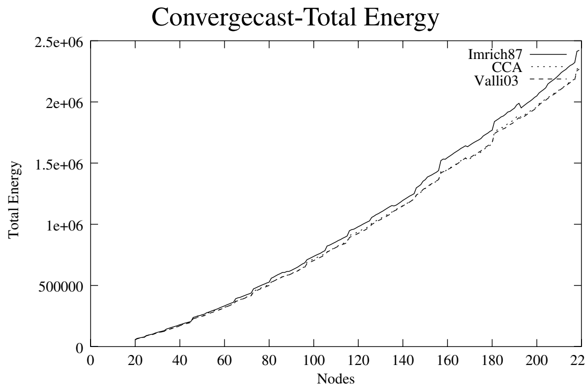
In this paper we have proposed an algorithm that constructs a tree and assigns channel for convergecasting. We have also shown the need for convergecasting using a tree and the disadvantage of using a broadcast tree. The CCA algorithm has good communication "quality" [5] because its demand on energy is low and data latency is also minimum. Also it is efficient in comparison to [1] and [2]. When CCA is used for broadcasting, we observed that it is very energy efficient and latency is low.

VII. Acknowledgements

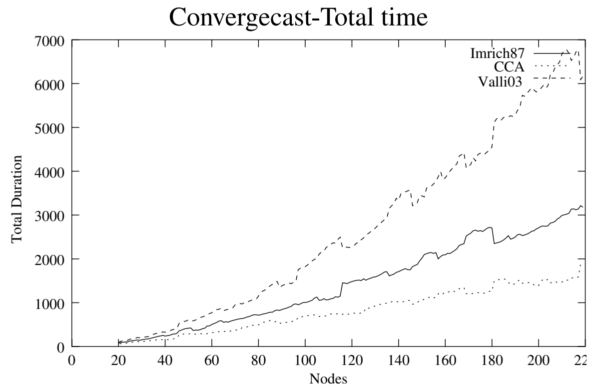
This work is supported in parts by NSF grants ANI-00196156, ANI-0086020, ANI-0123980, and DGE-9870720..

References

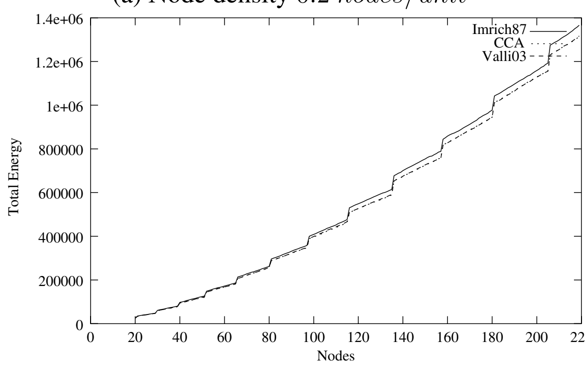
[1] V. Annamalai, S. K. Gupta and L. Schwiebert. On Tree-Based Convergecasting in Wireless Sensor Networks. *IEEE Wireless Communications and Networking Conference 2003*, New Orleans, 2003.



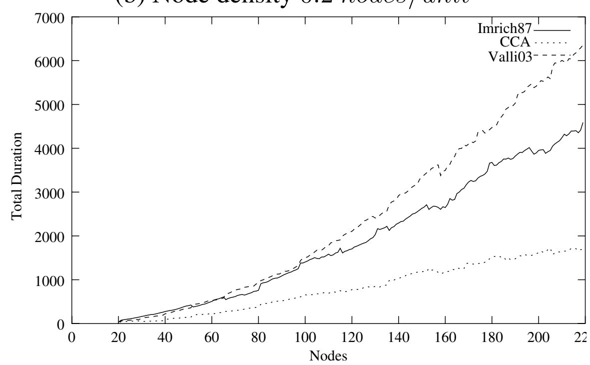
(a) Node density 0.2 $nodes/unit^2$



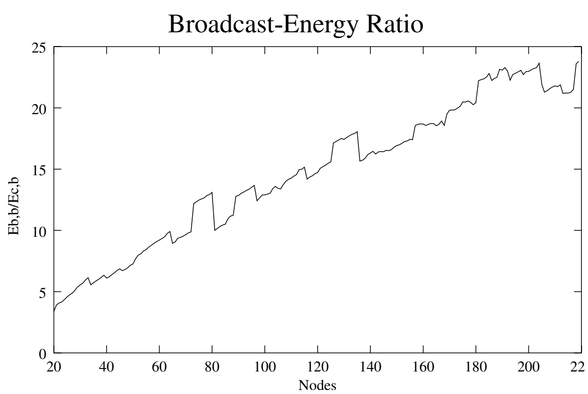
(b) Node density 0.2 $nodes/unit^2$



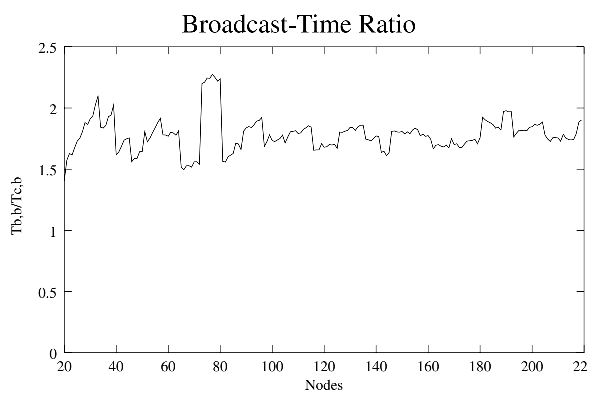
(c) Node density 0.8 $nodes/unit^2$



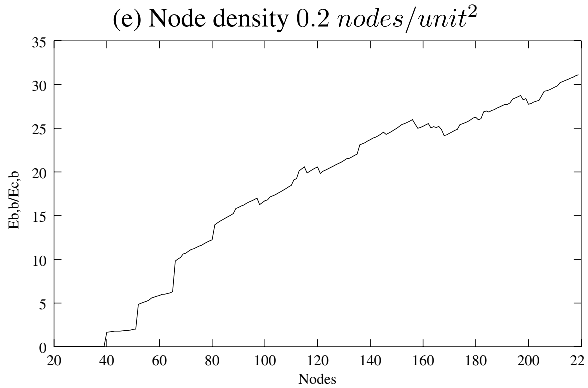
(d) Node density 0.8 $nodes/unit^2$



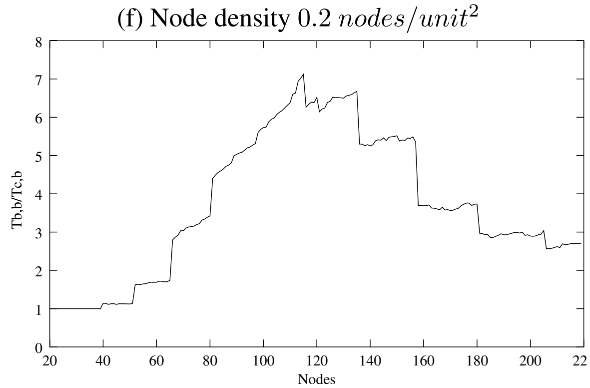
(e) Node density 0.2 $nodes/unit^2$



(f) Node density 0.2 $nodes/unit^2$



(g) Node density 0.8 $nodes/unit^2$



(h) Node density 0.8 $nodes/unit^2$

[2] I. Chlamtac and S. Kutten. Tree-based Broadcasting in Multihop Radio Networks. *IEEE Transactions on Computers* Vol. C-36, No. 10, Oct 1987.
[3] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Micro Sensor Networks. *Published in Proceedings of the Hawaii International Conference on System Science Jan 2000*.
[4] S. Lindsey, C. Raghavendra, K. M. Sivalingam. Data Gathering Algorithms in Sensor Networks Using Energy Metrics. *IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 9* Sept 2002.
[5] R. Min and A. Chandrakasan. Energy-Efficient Communication for Ad-Hoc Wireless Sensor Networks. *Proceedings of IEEE International Conference on Mobile Communications (MobiCom '02)*, 2002.
[6] T.S. Rappaport. *Wireless Communications*. Prentice-Hall, 1996.