Scheduling on Sensor Hybrid Network

Hongsik Choi, Ju Wang and Esther A. Hughes
School of Engineering
Virginia Commonwealth University
Richmond, VA 23284
Email:{hongsik,jwang3,eahughes}@vcu.edu.

Abstract—We investigate a unique scheduling problem in wireless sensor networks where all nodes in a cluster send exactly one packet to a designated sink node with goal of minimized transmission time. The difficulty lies in the fact that node transmissions must be sufficiently isolated either in time or in space to avoid collisions. The problem is formulated and solved via graph representation. We prove that with specific network topologies (either line or tree), an optimal transmission schedule can be obtained efficiently through a pipeline-like schedule. The minimum time required for a line (or tree) topology with n nodes is $3(n-2)$. We further prove that our scheduling problem is NP-hard for general graphs. We propose a heuristic algorithm for general graphs. Our heuristic tries to schedule as many *independent segments* as possible to increase the degree of parallel transmission. This algorithm is compared to an RTS/CTS based distributed algorithm. Preliminary simulated results indicate that our heuristic algorithm outperforms the RTS/CTS based distributed algorithm (up to 30%) and exhibits stable scheduling behavior.

Index Terms—Sensor network, hybrid network, scheduling, all-to-one information gathering

I. INTRODUCTION

Advances in wireless networking and the desire for real time data collection has resulted in tremendous growth in research activities in the area of wireless sensor networks. An important technology trend is the development of large-scale heterogeneous network architectures integrated with various communication technologies for better performance. In such network architectures, low cost battery powered sensor nodes are typically equipped with (several) data sensing components, limited data processing components, and short-range communication components. The baseline data collection functionality is provided by packet relaying among sensor nodes, where the data is eventually received in a central location. However, this method is not scalable and cannot handle aggregated traffic when hundreds or thousands of sensor nodes are used.

To support large-scale sensor networks and the fast collection of time critical data, the network is often organized into clusters [2] where information is first gathered at the cluster head, then further collected by mobile base stations such as unmanned vehicles for example. An important link to such a network paradigm is efficient packet routing inside of the clusters.

Much research has been performed on routing in sensor networks based on the ad hoc nature of sensor networks [4], [5], [1], [3] resulting in distributed routing algorithms. These distributed algorithms are inherently competition-based, such as in [5] and [7]. Thus, their performance is heavily dependent on the density of the sensor nodes and on traffic patterns.

In this paper, we will confine our discussion to the information gathering algorithm based on the deployment of mobile base stations. The scenario we consider consists of a group of sensor nodes, one of which is designated as the local cluster head (or sink). We assume that our traffic is specifically generated for information gathering by the sink node and each node has exactly one packet of information to be sent to the sink. The goal is to determine the minimum schedule time for all-to-one information gathering in which the sink node receives messages from each node in the network.

Unlike the traditional distributed solutions, we are more interested in a centralized scheduling algorithm. The motivation for considering a centralized algorithm is as follows: (1) it allows rigorous analysis techniques for the upper bound and optimality of schedule solutions, (2) the analysis for centralized scheduling leads to a more comprehensive understanding of the nature of wireless transmissions, (3) a centralized scheduler allows sensor nodes to transmit in a synchronized manner, which considerably saves the signaling (time and energy) overhead in a distributed access protocol, and (4) centralized scheduling is practically feasible in the cluster by broadcasting from the sink node. In a wireless sensor network with a relative static topology, a schedule can be reused repeatedly for the most part.

Because the network is wirelessly sharing a single channel, there is a possible conflict among transmissions. When a node u sends a message to a node v , several constraints are imposed on the network.

First, no nodes within the transmission range of node u (except v) can receive a message from u . Second, v cannot send a message while receiving from u , neither can it receive a message from nodes other than u . Third, none of v 's neighbors can send any messages. As argued by Gupta et al. in [5], these constraints limit the actual capacity of wireless networks. The media access problem in wireless sensor networks is also addressed in [6].

The problem is formulated as the **Minimum Information Gathering Time Problem (MIGTP)** based on a graph representation of the network topology. The problem is first attacked for a linear graph topology. We discover that in this case, the optimum scheduling is obtained by allowing parallel transmission in an interleaved manner. We develop the **LPIPE** algorithms to explicitly obtain the transmission schedule of each node while satisfying all constraints. Furthermore, it is proved that **LPIPE** algorithm minimizes the total transmission time for the entire network. It turns out that a linear graph will require

a minimum of $3(n - 2)$ time slots, where n is the number of nodes in the network. A similar result has been discussed in [5].

The **LPIPE** is then extended to **TPIPE** for tree topologies. Interestingly enough, all claims for the line topology are proved to be true for the tree topology. With these claims for the line topology, we can prove the optimality of **TPIPE**. Further, the optimum schedule for a given tree requires the same number of time slots as does the line topology with the same size.

However, the optimal schedule for general graph topology is difficult to obtain. This is due to the followings: (1) the number of possible routing trees grows exponentially with the size of the graph, and (2) the transmission schedule is subject to more constraints for the general graph than in the case of the line or tree topology. In fact, we prove that finding the optimum schedule for a general graph is NP-hard. The proof is constructed by reducing from a *partition problem* (a known NP hard problem) to a simplified graph scheduling problem.

We thus propose a heuristic algorithm for sub-optimal solutions. The heuristic is based on the observations (from line and tree cases) that a good schedule must rely on a good routing tree and allow as much parallel transmission as possible. We construct a minimum spanning tree using Dijkstra's algorithm such that the hop-distance between any sensor node and the sink is minimal. Our heuristic algorithm then iteratively identifies all independent line segments in the routing tree that can be scheduled in parallel. In the selection process, the longest segment is selected. This will divide the graph into a set of independent line segments with some remaining nodes which can not be scheduled in parallel with the independent segments. The final scheduling with these segments is greedy based: the schedule for the independent segments is determined first using **LPIPE** algorithm, this will generate an initial schedule for nodes in the independent segments. Then the remaining nodes will be scheduled according to their hop-distance from the sink node. To summarize, the proposed algorithm is greedy in the sense that it schedules the longer path in the routing tree first. This is because the longest tree path can be treated as the critical path. The performance of the heuristic algorithm is evaluated via simulation where we consider a cluster of sensor nodes in a 1000*1000 area. The number of nodes in the simulation ranges from 2 nodes to 40 nodes. For a fair comparison, a RTS/CTS based distributed transmission scheme is also simulated, based on the same routing tree obtained by our heuristic algorithm. The results indicate that the proposed heuristic algorithm outperforms the RTS/CTS method in all cases. When the number of nodes increases, the average number of slots of our heuristic algorithm is significantly smaller than the RTS/CTS base methods. Furthermore, we find that the RTS/CTS based method becomes increasingly unstable for large cluster sizes: the average number of slots can fluctuate by more than 40% if the graph size is large. In contrast, the performance of our method is stable: it shows a steady (but smooth) increasing scheduling time as the number of the nodes in graph grows. These preliminary simulation results indicate that the proposed heuristic algorithm can deliver a good schedule and is scalable.

II. NETWORK MODEL AND PROBLEM DEFINITION

We represent our sensor hybrid network by a graph, G . Each node in the graph corresponds to a node in the network. Edges represent two nodes within communication range which implies they can communicate directly.

Our problem can be defined as follows: Given a network with a sink node and all to one traffic, find a proper minimum time schedule. A schedule finds a path from one node to the sink node for each node in the network (except the sink node) and assigns a time slot to each link in the path. To be a proper schedule, no two links can occupy the same time slot if they are within a conflict range.

Our problem is formulated as the Minimum Information Gathering Time Problem (MIGTP) as follow: **MIGTP**(Minimum Information Gathering Time Problem)

Instance: A directed graph $G = (V, E)$, a sink $s \in V$, a message set $M = \{m_i, \text{ for all } v_i \in V/s\}$

Objective: Find a minimum schedule time K and a proper schedule for all to one gathering, i.e., find a set of paths, $P = \{p(v_i, s) | \forall v_i \in V\}$, and a function $f : E(P) \rightarrow \{1, 2, \dots, K\}$ such that $f(e_1) \neq f(e_2)$ whenever $\{e_1, e_2\} \in E(P)$

To describe the problem more precisely, let us define some terminology. Let $*$ indicate all possibilities, for example, v_* means any $v_i \in V$. Let $N(v_i), (n(v_i))$ be the neighboring node set (and neighbor node) of vertex v_i , respectively. The problem is defined to minimize K and find function $f : E \times M \rightarrow \{0, 1, 2, \dots, K\}$ such that

- (a) $f((v_i, *), m_p) \neq f((v_i, *), m_q)$ iff $p \neq q$ for all $v_i \in V$,
- (b) $f((u, v_i), m_*) \neq f((u, v), m_p)$ for all $v_i \in N(u)/v$
- (c) $f((v_i, u), m_*) \neq f((u, v), m_p)$ for all $v_i \in N(u)$
- (d) $f((v, *), m_*) \neq f(u, v), m_p)$
- (e) $f((v_i, *), m_*) \neq f((u, v), m_p)$ for all $v_i \in N(v)/\{N(u), u\}$
- (f) $f((v_i, v_j), m_p) < f((v_j, *), m_p)$ for all $0 < i, j, p \leq n - 1$
- (g) For all $m_p, 0 \leq p \leq n - 1$, there exist a path $P_p = ((v_p, v_i) = e_1, e_2, \dots, e_j = (v_j, s))$ such that $f(e_i, m_p) \neq \text{null}$ for all $e_i \in P_p$,
- (h) $P_q \subseteq P_p$ iff $v_q \in P_p$.

Constraint (a) ensures that no node tries to send more than two messages at the same time. Constraint (b) through (e) ensure that no nodes can transmit a message that might interfere with the transmission from node u to v . As in Figure 1, when node u transmits a message to node v , transmission through the edges labeled b, c, d , or e is prohibited since they would collide with the message sent from u to v . Edges labeled $b, (c, d, e)$ are matched with constraint (b), ((c), (d), (e)) of the problem, respectively. Dotted lines exist if two nodes are within transmission range. Constraint (f) ensures that no node can transfer a message before it actually receives that message. Constraint (g) ensures that $n - 1$ unique messages from each $n - 1$ node can be delivered to the sink node. Intuitively, we think of the problem as two subproblems: a routing problem and a time assignment problem. Constraint (h) makes message forwarding stateless so that each sensor node does not need to maintain costly state information. Otherwise, the intermediate sensor nodes have to forward data depending on the data's source and destination information.

Fig. 1. Links conflicting with (u, v) in G

III. MAIN RESULTS

Throughout this section, we speak of the network and topology interchangeably. Proofs for all lemmas and theorems discussed are referred in [9].

A. Line Topology

Fig. 2. Links conflicting with $(i, i-1)$ in line

In this section, we consider a line network with the nodes numbered from 0 to $n-1$ from left to right. The line network makes the problem rather simple because the topology itself fixes the routing. The problem reduces to a time slot assignment problem.

For simplicity, we assume the sink node is always labeled as node 0. This assumption can be relaxed later without any further difficulty.

TAPL(Time Slot Assignment Problem for Line)

Given a line graph $L = (V, E)$ and $M = \{m_v, \forall v \in V\}$ where $(v_i, v_{i+1}) \in E$ and $(v_i, v_j) \notin E$ iff $|i-j| \neq 1$, find a function $f : V \times M \rightarrow \{0, 1, \dots, K-1\}$ such that

- (1) $f(v_i, m_j) \neq f(v_i, m_k)$ iff $j \neq k$ for all $v_i \in V$,

- (2) $f(v_i, m_*) \neq f(v_j, m_*)$ if $|i-j| \leq 2$ for $1 \leq i, j \leq n-1$,
- (3) $f(v_i, m_j) < f(v_{i-1}, m_j)$ for all $1 < i \leq j \leq n-1$,
- (4) $f(v_i, m_j) \neq \text{null}$, for all $1 \leq i \leq j \leq n-1$,
- (5) satisfying (1) - (4), K is minimized.

Constraint (1) ensures that no node tries to send more than two messages at the same time. Constraint (2) ensures that no nodes can transmit a message that possibly interferes with the transmission from node v_i to node v_{i-1} . As in Figure 2, when node i transmits a message to node $i-1$, none of the nodes $i+1, i+2, i-1$, or $i-2$ can send a message because they may conflict with node i 's transmission or conflict with the transmission from node i to node $i-1$. The possible conflicting transmission is represented as a dotted line. Constraint (3) ensures that no node can transfer a message before it actually receives that message. Constraint (4) ensures that $n-1$ unique messages from each of the $n-1$ nodes can be delivered to the sink node. Constraint (5) ensures the minimum schedule among those schedules satisfying constraints (1) through (4).

We present the following algorithm, LPIPE.

Algorithm 1: LPIPE

Input: A line graph $G(N, E)$, $|N| = n$
 $(v_i, v_{i+1}) \in E$ for $0 \leq i < n-1$
 $M = \{m_i, 1 \leq i \leq n-1\}$

Output: Schedule S , i.e., $f : E \times M \rightarrow t$,

begin

1. **for** $k = 0$ to $n-4$
2. **for** $i = 1$ to $n-1-k$
3. $S(e(v_i, v_{i-1}), m_{i+k}) = t_{(i-1)\%3+3+k}$
- endfor**
4. $S(e(v_1, v_0), m_{n-2}) = t_{(n-3)*3}$
5. $S(e(v_2, v_1), m_{n-1}) = t_{(n-3)*3+1}$
6. $S(e(v_1, v_0), m_{n-1}) = t_{(n-3)*3+2}$

end.

Figure 3 shows a sample schedule for a 10 node line network.

edge	(1,0)	(2,1)	(3,2)	(4,3)	(5,4)	(6,5)	(7,6)	(8,7)	(9,8)
time									
0	m(1)			m(4)			m(7)		
1		m(2)			m(5)			m(8)	
2			m(3)			m(6)			m(9)
3	m(2)			m(5)			m(8)		
4		m(3)			m(6)			m(9)	
5			m(4)			m(7)			
6	m(3)			m(6)			m(9)		
7		m(4)			m(7)				
8			m(5)			m(8)			
9	m(4)			m(7)					
10		m(5)			m(8)				
11			m(6)			m(9)			
12	m(5)			m(8)					
13		m(6)			m(9)				
14			m(7)						
15	m(6)			m(9)					
16		m(7)							
17			m(8)						
18	m(7)								
19		m(8)							
20			m(9)						
21	m(8)								
22		m(9)							
23	m(9)								

Fig. 3. An example schedule for 10 node line network

Lemma 1: The algorithm LPIPE makes a feasible solution for TAPL, i.e., produce a schedule that meets constraints (1)

through (4) of TAPL.

Lemma 2: The schedule from the algorithm LPIPE uses $3(n - 2)$ time slots, i.e., $K = 3(n - 2)$.

Next we introduce the Dual Graph Coloring Problem (**DGCP**) to discuss the optimality of the LPIPE algorithm.

Given $G_m = (V_m, E_m)$, find the minimum number of colors for proper coloring of the graph, i.e., the color assigned to u and v is different whenever $\{u, v\} \in E_m$, where $V_m = \{v_{i,j}, 0 < j \leq m, j \leq i \leq m\}$ and E_m can be defined as a union of sets, i.e.,
 $\{(v_{i,k}, v_{j,k}), i \neq j, 1 \leq k \leq m - 1\} \cup$
 $\{(v_{i,k}, v_{j,k+1}), 0 < k < m, k < i \leq m, k + 1 \leq j \leq m\} \cup$
 $\{(v_{i,k}, v_{j,k+2}), 1 \leq k \leq m - 2, k < i \leq m, k + 2 \leq j \leq m\}.$

Figure 4 shows a sample graph instance of G_m .

Fig. 4. An example instance of G_4

Lemma 3: The optimal solution for DGCP is $3(m - 1)$, i.e., the chromatic number $\chi(G_m) = 3(m - 1)$.

Lemma 4: The solution for DGCP with G_{n-1} can be constructed from the solution of TALP with n nodes.

Lemma 5: The optimal solution for DGCP with G_{n-1} is a lower bound of TALP with n nodes, i.e., the chromatic number $\chi(G_{n-1}) = 3(n - 2) \leq K$.

Theorem 1: LPIPE gives an optimal schedule for the TALP

Lemma 6: When the i^{th} node is a sink node, the lower bound for TALP with n nodes is

$$\max\{3(i - 2), 3(n - i - 1)\} \quad \text{if } i \neq n - i + 1 \quad (1)$$

$$3(i - 2) + 1 \quad \text{if } i = n - i - 1 \quad (2)$$

Theorem 2: One can get an optimal solution for TALP with n nodes using LPIPE when the i^{th} node is a sink node.

B. Tree Topology

In this section, we consider the tree network. Without loss of generality, we can assume that the sink node is always the root node. Since the route from any node to the root node is fixed in a tree topology, the problem is also reduced to a time slot assignment problem like line topology problem. We will refer this problem as a Time Slot Assignment Problem for a Tree (TAPT) and is defined as follows:

TAPT(Time Slot Assignment Problem for Tree)

Given a tree graph $T = (V, E)$ and $M = \{m_v, \forall v \in V\}$, find

a function $f : V \times M \rightarrow \{0, 1, \dots, K - 1\}$ such that

- (1) $f((v_i, *), m_p) \neq f((v_i, *), m_q)$ iff $p \neq q$ for all $v_i \in V$,
- (2) $f((u, *), m_p) \neq f((v, *), m_*)$ for all $v \in N(u)$,
- (3) $f((u, *), m_p) \neq f((*, v), m_*)$ for all $v \in N(u)$,
- (4) $f((*, u), m_p) < f((u, *), m_p)$ for all $m_p \in M$ and $u \in V$,
- (5) There exists a path $P_p = ((v_p, v_i) = e_1, e_2, \dots, e_k = (v_k, s))$ such that $f(e_i, m_p) \neq \text{null}$ for all $e_i \in P_p$ and $m_p \in M$,
- (6) satisfying (1) through (5), K is minimized.

Constraint (1) ensures that no node attempts to send more than two messages at the same time. Constraint (2) and (3) ensure that no node can transmit a message that possibly interferes with a transmission from node u to its parent node. Constraint (4) ensures that no node can transfer a message before it actually receives that message. Constraint (5) ensures that a message from each node will be delivered to the sink node. Constraint (6) ensures the minimum schedule among those schedule satisfying constraints (1) through (4).

We present the following algorithm, TPIPE. Without loss of generality, we assume that $v_0, v_1, v_2, \dots, v_{n-1}$ are visited in sequence using a depth-first search. Let $d(v_i)$ be the degree of the node v_i , and $v_{i,1}, v_{i,2}, \dots, v_{i,d(v_i)-1}$ be the children of node v_i . Assume that DFS visits node $v_{i,j}$ before visiting node $v_{i,j+1}$.

For simplicity, we assume the degree of both the root node and the direct child of the root node is one. This assumption can be relaxed without any further difficulty.

Lemma 7: The TPIPE algorithm produces a schedule with $3(n - 2)$ time slots, i.e., $K \leq 3(n - 2)$.

Lemma 8: The lower bound of TAPT is $3(n - 2)$, i.e., $3(n - 2) \leq K$.

Theorem 3: TPIPE gives an optimal schedule for TAPT.

proof: By lemma 7 and 8, $3(n - 2) \leq K \leq 3(n - 2)$. Therefore, $K = 3(n - 2)$

Algorithm 2: TPIPE

Input: A tree graph $G(N, E)$, $|N| = n$,

$M = \{m_i, 1 \leq i \leq n - 1\}$

Output: Schedule S , i.e., $f : E \times M \rightarrow t$

begin

1. Make a line graph $G_L(N_L, E_L)$ such that
 $N_L = N, M_L = M,$
 $E_L = \{(v_i, v_{i+1}) | 0 \leq i \leq n - 2\}$
2. call LPIPE with G_L, M_L
3. $S = S_L$
4. **for** $i = 1$ to $n - 2$
5. **if** $d(v_i) > 2$
6. **for** $v_j = v_{i,2}$ to $v_{i,d(v_i)-1}$
7. **for** $k = j$ to $n - 1$
8. $S(e(v_j, v_i), m_k) = S(v_j, v_{j-1}, m_k)$
9. **if** $S(e^*, m_k) \leq 3(j - i - 1)$
10. $S(e^*, m_k) = \text{null}$
11. **endif**
12. **endfor**
13. **endif**
14. **endfor**
15. **endfor**

end.

C. Arbitrary Topology: Complexity Analysis

Following theorem shows that the decision version of MIGP is NP-complete in a weak sense. It implies that MIGP is NP-hard and is unlikely to have a polynomial solution unless $P = NP$.

Theorem 4: The decision version of the Minimum Information Gathering Problem (MIGP) is NP-complete in a weak sense.

D. Arbitrary Topology: Heuristic Algorithm

Since MIGTP is proven as NP-hard by theorem 4, we must rely on the heuristic solutions. Our heuristic algorithm breaks down the schedule into two subtasks: (1) finding the optimum routing tree, (2) finding the best transmission schedule for the given routing tree.

Our Heuristic algorithm starts with the minimum spanning tree T_s of the given general graph G using any MST algorithm such as Dijkstra's. Scheduling for T_s is not trivial: nodes on different branches of the tree might fall within each other's transmission range, thus making the scheduling among branches dependent on each other. Thus, tree T_s must be trimmed such that packet transmission on different branches will not interfere with each other and can be transmitted in parallel. The following algorithm describes the trimming process:

Algorithm 3: BUILD_BACKBONE

Input: A tree T_s

Output: A backbone forest T_b

begin

1. find the longest path P_0 in T_s
2. let $T_b = P_0$ and mark all nodes in P_0
3. on the k^{th} loop
4. find an unmarked longest path P_k in T_s
5. Let $A_k = \{v_k^i \in P_k | N(v_k^i) \cap T_b \neq \emptyset\}$
6. make $m(k)$ disjoint segments such that

$$P_k = \bigcup_{i=1}^{m(k)} P_k^i$$
7. add all segments P_k^i into T_b
8. mark P_k in T_s
9. go to 3, if it is not stable
where $m(k) = |A_k|$,
segments P_k^i are path start with v_k^i

end.

Algorithm 4: GREEDY_HEURISTICS

Input: A backbone forest T_b

Output: A schedule S

begin

1. **for all** $P_i \in T_b$
2. **for all** $P_i^j \in P_i$
3. calculate forbidden time slot $F(P_i^j)$
4. schedule on the path from s to $v_l(P_i^j)$
for messages originated from nodes in
 P_i^j using LPIPE w/o time slots in $F(P_i^j)$
6. **endfor**
7. **endfor**
where $v_l(P_i^j)$ be the farthest node from s
among nodes in the segment P_i^j and

end.

This will generate a backbone T_b consisting of edge disjoint segments. Furthermore, each segment in T_b can be scheduled independently with other segments as long as the schedule does not use forbidden time slots. Forbidden time slots for segment P_k^i are defined as slots used by nodes $v \in \{N(v_k^i) \text{ or } N(N(v_k^i))\}$ in the segments so far scheduled. To simplify the algorithm description, define the forbidden time slots $F(P_i^j)$ for segment P_i^j more formally as follows:

$$F(P_i^j) = \{s(v) | v \in (N(v_i^j) \vee N(N(v_i^j))) \wedge (P_k, 1 \leq k < i \vee P_i^k, 1 \leq k < j)\}$$

The schedule can now be described as Greedy_Heuristics.

E. Arbitrary Topology: Simulation Results

The above heuristic scheduling algorithm is evaluated via simulation. Our simulator generates locations for mobile stations randomly in a 1000*1000 square meter rectangle area. The number of nodes in the simulation varies from 2 to 40. For a given number of nodes, we generate 50 different topologies. Instead of using a fixed transmission range for determining the neighbor set, each mobile base station can tune its transmission power such that the transmission range is sufficiently big to reach a node connected to the destination. Figure 5 (a) shows a random graph with 40 nodes. Notice that each node uses different transmission ranges.

Fig. 5. Random graphs

For an arbitrary generated random graph, the routing tree is obtained by the **Build_Backbone** algorithm such that each node knows exactly its next hop. The corresponding backbone tree to random graph (a) in figure 5 is (b).

To understand the performance of the proposed heuristic, we include the CSMA-CA based distributed access method in

our comparison. In practice, CSMA-CA is implemented via RTS/CTS signaling to achieve media access. It is easily seen that RTS/CTS is greedy in the sense that it allows the maximum degree of parallelism via active media sensing at all nodes. In this paper, we assume a flat 15% overhead to count for these consideration for the RTS/CTS method.

The simulation results are shown at Figure 6. Figure 6 (a) shows the average time slots for the proposed heuristic algorithm and the RTS/CTS based method. We observe that the average time slots increase almost linearly as the number of nodes increases. However, the curve corresponding to the randomly selected case also indicates strong dynamic as more nodes join to the area. For instance, the total number of slots for the random case at 37 nodes is 100; while for the random case at 40 nodes, the number of slots is about 30; which is only 30% of the former case.

Between our proposed heuristic and the RTS/CTS method, our heuristic algorithm results in smaller slot numbers in all cases, and shows considerable improvement in most cases. The improvement becomes more apparent when the size of the graph increases. Also, as the graph size increases, the RTS/CTS method become increasingly unstable. The average slot number at 37 nodes is 120, and this number dramatically drops to 87 at 38 nodes. Such instability in the large graph size makes the RTS/CTS method less scalable. The average slot number of the proposed heuristic shows a smooth increase. For a given network (graph) size, we also plot the standard deviation of the slot number among the 50 random cases. As the number of node increase, the standard deviation also increases. This is expected since the larger the graph size is, the more diverse the topology is, and thus a higher degree of fluctuation in terms of scheduling time.

IV. CONCLUSIONS

We have studied the problem of all-to-one information gathering in a wireless hybrid sensor network with n nodes, where one node's transmission range is far longer than that of others. The base station (node n_0) can send the transmission schedule to all $n - 1$ sensor nodes within the base station's transmission range. The significance of this model considered in this paper is i stressed with the sensor network in large scale deployment.

We first showed an optimal solution when the graph topology is a line or a tree. We then showed that the problem with a general topology is NP-hard, and then presented a heuristic algorithm along with simulation results.

REFERENCES

- [1] Curt Schurgers, M.B. Srivastava, "Energy Efficient Routing in Wireless Sensor Networks," *MILCOM'01*, Vienna, VA, pp. 357-361, October 28-31, 2001.
- [2] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," *Proceedings of IEEE INFOCOM*, March 2004.
- [3] G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Multiple-Access for Multiple Destinations in Ad-hoc Networks," *Proc. WiOpt 03*, Sophia-Antipolis, France, Mar. 2003.
- [4] A. Ephremides and T. Truong, "Scheduling Broadcasts in Multihop Radio Networks," *IEEE Trans. Commun.*, vol. 38, no. 4, Apr. 1990, pp. 456-460.
- [5] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Info. Theory*, vol. 46, no. 2, Mar. 2000, pp. 388-404.

Fig. 6. Scheduling results: (a) average slots, (b) std

- [6] Y.E. Sagduyu and A. Ephremides, "The problem of medium access control in wireless sensor networks," *IEEE Wireless Communications*, vol. 11, Issue 6, 2004 pp. 44-53.
- [7] Y.C. Tay, K. Jamieson and H. Balakrishnan, "Collision-minimizing CSMA and its applications to wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, 2004 pp.1048 - 1057.
- [8] M. A. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.
- [9] H. Choi, J. Wang, and E. A. Hughes, "Scheduling Issues in 3D Networks," *VCU-LNCR Technical Report*, No. 2005-4-1, 2005.