This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

Wakeup Scheduling in Roadside Directional Sensor Networks

Jian Tang, Binhai Zhu, Li Zhang and Roberto Hincapie

***Abstract*—In this paper, we focus on a roadside directional sensor network where sensors with directional Field Of Views (FOVs) are placed along a roadway. We study the problem of wakeup scheduling, with the objective of maximizing network lifetime under the constraint that full coverage and network connectivity are maintained at all times. First, we present centralized polynomial time algorithms to optimally solve the problems of scheduling sensor nodes with fixed sensing orientations. Moreover, an effective heuristic algorithm is proposed to solve the problem of scheduling sensor nodes with variable sensing orientations. In addition, we also present distributed algorithms for both fixed and variable cases. Simulation results based on a roadway in the Yellowstone National Park have been presented to show the performance of the proposed algorithms.**

***Index Terms*— Wireless sensor network, directional sensor, roadside, scheduling, coverage.**

I. Introduction

The lifetime of a Wireless Sensor Network (WSN) can be significantly extended by carefully scheduling sleep and wakeup for sensor nodes [15]. Traditional research has focused on WSNs with omni-directional (isotropic) sensors which have disk-like sensing coverage regions. However, some widely deployed sensors, such as video sensors, image sensors, ultrasonic sensors and infrared sensors, only have directional Field Of Views (FOVs), i.e., it can only sense events that happen in certain directions [4]. In this paper, we focus on such directional sensors. In addition, we consider a particular application scenario in which a set of directional sensor nodes are deployed on the two sides of a chain to monitor the related activities. For example, a set of directional sensors can be placed along a roadway to monitor car accidents and rock slides, or detect crossing of moving objects. As far as we know, the Western Transportation Institute at Montana State University has deployed sensors along the roadways near the Yellowstone National Park to detect the wildlife crossing for the road ecology research [13]. The applications belonging to such a scenario are not restricted to the roadside and transportation. A directional sensor network can also be placed along the borderline of a country or a territory to monitor and detect illegal intrusions.

In this paper, we study the scheduling problems for chain coverage in directional sensor networks. Based on a sleep-wakeup schedule, a subset of sensor nodes will work and the others will sleep during a certain period of time. However, the working and sleeping sensor node sets change over time. The ultimate goal is to maximize network lifetime while maintaining the full chain coverage and network connectivity at any time. We consider two kinds of directional sensors: fixed and variable. A *fixed directional sensor* has a fixed sensing orientation, however, the orientation of a *variable directional sensor* can be adjusted. We present centralized polynomial time algorithms to optimally solve the problems of scheduling fixed directional sensors. Moreover, an effective heuristic algorithm is proposed for the variable case. In addition, we also present distributed algorithms for both cases.

Extensive research has been conducted to study the connectivity and coverage issues, and the related scheduling problems in WSNs with omni-directional sensors [11], [12], [15]. Coverage issues have also been studied in directional sensor networks [1], [2], [4], [6], [14] and even in 3D directional sensor networks [8]. However, we address the sleep-wakeup scheduling problems in the context of *directional sensor networks and chain coverage* in this work, which have not been well studied before. We propose both theoretically well-founded and practically useful algorithms to solve them.

II. Problem Formulation

In this paper, we consider a WSN consisting of static directional sensor nodes, each of which is aware of its location. Without abuse of notation, v denotes both sensor node v and its location. Our sensing model is similar to that in [2], [4]. Each node v has a directional sensor whose sensing region is a sector of the disk centered at v with a sensing range of r and a Field Of View (FOV) of θ. The *orientation* of a sensor node is defined as the direction of the central axis of its sensing sector. We will consider sensors with the fixed orientations as well as sensors with variable orientations, which will be simply called *fixed directional sensors* and *variable directional sensors* respectively in the following. For the fixed case, the orientation of a sensor is given beforehand and is not allowed to be changed during its lifetime. For the variable case, the orientation of a sensor can be tuned to any direction in a given set Ω. Similar as in [2], [4], we assume that all the possible sensing sectors of a variable sensor v are mutually disjoint and they form a disk centered at v with a radius of r. The target area of the sensor network is a narrow chain which is composed of a set of landmarks points and straight line segments in between. Note that any road or borderline can be precisely approximated by such a chain as long as there are enough landmark points. We assume that the width of the chain is negligible because the sensing range r and its total length are much larger, and we are only interested in knowing whether some moving objectives cross the chain in many applications (e.g., animals cross the road or people pass the borderline). Sensor node placement is out of scope of this paper and will be addressed by our future research. We

Jian Tang is with Department of Electrical Engineering and Computer Science at Syracuse University, Syracuse, NY 13244. Binhai Zhu is with Department of Computer Science at Montana State University, Bozeman, MT 59717. Li Zhang is with Department of Computer Science at UC-Davis, Davis CA 95616. Roberto Hincapie is with Department of Telecommunications Engineering, Universidad Pontificia Bolivariana, Medellin, Colombia.

978-1-4244-9268-8/11/$26.00 ©2011 IEEE

assume that all the sensor nodes have been placed on the two sides of the chain in advance and they can cover it at least once. No matter which kind of directional sensor is used, a sensing sector of a sensor node can always cover a continuous chain *segment* with a fixed length. In addition, each node is also equipped with an omni-directional antenna and a radio with a transmission range of R.

Given a set of fixed directional sensor nodes V and a chain L, a *Fixed Directional Sensor Cover (FDSC)* is a minimum subset of sensor nodes of V which can cover L and form a connected network. For the variable case, a tuple (v, i) is used to denote the ith *usable* sector of node v. Note that only those sectors which overlaps with L are usable. The set of all usable sectors is denoted as $\Pi = \{(v, i)|v \in V, 1 \leq i \leq q_v\}$, where q_v is the total number of usable sectors of node v. A *Variable Directional Sensor Cover (VDSC)* is a minimum subset of sectors of Π such that they can fully cover L, the subset includes no more than one sector from each node and the corresponding nodes form a connected network. The lifetime of a FDSC S is $T(S) = \min\{\frac{E_v}{P}|v \in S\}$, where E_v is the initial energy of node v and P is the energy consumption rate of a working sensor node. P is assumed to be a constant since a sensor node usually stays idle in most of its working time in our application scenarios. Accordingly, the lifetime of a VDSC S_C is $T(S_C) = \min\{\frac{E_v}{P}|(v, i) \in S_C\}$. A fixed (variable) directional sensor schedule is a collection of FDSCs (VDSCs) whose lifetime is the summation of their lifetime values. Note that two FDSCs (VDSCs) in a schedule may not be disjoint.

We will consider the following two optimization problems.

Definition 1: (Fixed Directional Sensor Scheduling Problem (FDSSP)): Given a set of fixed directional sensor nodes V and a chain L, the **Fixed Directional Sensor Scheduling Problem (FDSSP)** seeks a fixed directional sensor schedule Γ such that Γ has the maximum lifetime among all fixed directional sensor schedules corresponding to V and L.

Definition 2: (Variable Directional Sensor Scheduling Problem (VDSSP)): Given a set of variable directional sensor nodes V and a chain L, the **Variable Directional Sensor Scheduling Problem (VDSSP)** seeks a variable directional sensor schedule Γ such that Γ has the maximum lifetime among all variable directional sensor schedules corresponding to V and L.

III. Scheduling Algorithms

Similar as in [15], it is easy to show that if $R \geq 2r$, then a minimum subset of sensor nodes of V covering L is also connected, i.e., the complete coverage of a chain implies connectivity among the set of working sensor nodes. Due to space limitation, details are omitted here. In practise, this condition can usually be satisfied. For example, $R = 300$m for some sensor nodes from Crossbow such as Mica2 and MPR410CB [9], [15] and $r \leq 100$m for most sensors [4], [15]. Therefore, we will consider the case where $R \geq 2r$ and focus only on the coverage in the following.

A. Fixed Directional Sensor Scheduling

Given a set of fixed directional sensors which have already been placed, we show that the FDSSP is polynomially solvable. We consider two versions of the FDSSP: the uniform initial energy version (FDSSP-UE) in which all the sensor nodes are assumed to have the same initial energy as well as the non-uniform initial energy version (FDSSP-NUE) in which different nodes may have different initial energy. Obviously, the algorithm for the FDSSP-NUE can be used to solve the corresponding FDSSP-UE problem. However, we propose an algorithm particularly for the FDSSP-UE which is much faster than our algorithm for the FDSSP-NUE.

We have the following theorem and the corresponding algorithm is given after its proof.

Theorem 1: The FDSSP-UE problem can be solved by Algorithm 1 within $O(kn^2)$ time, where k is the minimum number of sensor nodes covering any point on L and n is the number of sensor nodes.

Proof: Let $L = (b_0, c_0)$ and let $s_i = (b_i, c_i)$, which is a segment of L covered by sensor node v_i for $i = 1, 2, ..., n$. *From now on, when the context is clear, we will mix the use of a sensor node v_i and the segment $s_i = (b_i, c_i)$ covered by v_i.* If we cut all such segments with vertical lines through all endpoints within L, we can obtain at most $(2n - 1)$ open *base intervals*. Note that each base interval is free of any endpoint of some s_i as shown by Fig. 1. For each base interval I, let the number of segments covering I (or, the number segments *stabbed* by I) be the *stabbing number* of I, denoted as $SN(I)$. Let SN^* be the minimum stabbing number over all base intervals. All base intervals with stabbing number SN^* are called *critical* base intervals. In Fig. 1, there are 11 base intervals, 6 of which are critical base intervals whose stabbing number is 2, all at odd positions ordered from left to right. Let the lifetime of each sensor node be $t = \frac{E}{P}$, where E is the initial energy of every sensor node. We claim the following: the minimum stabbing number $SN^* = k$ if and only if the maximum network lifetime is kt.

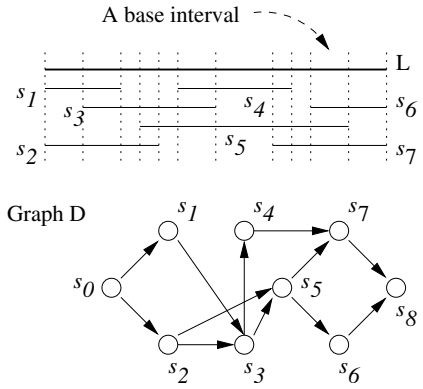

Fig. 1. Base intervals and the corresponding graph D

We define a directed acyclic graph $D(V_D, E_D)$ where each vertex in V_D corresponds to a sensor node and there is an edge from sensor node v_i to sensor node v_j iff $b_i < b_j \leq c_i < c_j$ (i.e., their coverage regions overlap). In addition, we create a virtual source node v_0 and a virtual sink node v_{n+1}. There is a directed edge from v_0 to each sensor node covering the first base interval and a directed edge from each sensor node covering the last base interval to v_{n+1} in D. s_0 and s_8 are

virtual nodes in Fig. 1. We can see a stronger version of the above claim: $SN^* = k$ if and only if there are k vertex-disjoint paths connecting v_0 and v_{n+1}. Then the maximum network lifetime is kt because the sensors on each path form a FDSC. We will prove this claim next.

If $SN^* = k$ then by induction on k, we can show that between v_0 and v_{n+1} there are k vertex-disjoint paths. This is due to the fact that once we identify in sequence all the critical base intervals (i.e., those covered by exactly k sensors), $I_1, I_2, ..., I_d$, we can peel off, from left to right, a set of intersecting segments (sensors) such that they cover all the base intervals and form a path from v_0 to v_{n+1}; moreover, the sensor nodes on this path cover each critical base interval exactly once. In Fig. 1, we have $k = 2, d = 6$. Then, after deleting these sensor nodes, the minimum stabbing number for the remaining segments is $(k-1)$. This can be done formally as follows. For a sensor node $v_{i,1}$ covering I_i, identify $v_{i+1,1}$ covering I_{i+1} such that $v_{i,1}$ *and* $v_{i+1,1}$ *do not cover the same critical base interval in* $\{I_1, I_2, ..., I_d\}$ *unless* $v_{i,1} = v_{i+1,1}$; moreover, the path between $v_{i,1}$ and $v_{i+1,1}$ is a shortest path in graph D, for $i = 1, 2, ..., d-1$. Such a path must exist as otherwise $SN^* < k$. Let this shortest path be $\langle v_{i,1} \rightsquigarrow v_{i+1,1} \rangle$. Now we can concatenate these paths to have a path between v_0 and v_{n+1}, $\langle v_0 \rightsquigarrow v_{1,1} \rangle \circ \langle v_{1,1} \rightsquigarrow v_{2,1} \rangle \circ \cdots \circ \langle v_{i,1} \rightsquigarrow v_{i+1,1} \rangle \circ \cdots \circ \langle v_{d-1,1} \rightsquigarrow v_{d,1} \rangle \circ \langle v_{d,1} \rightsquigarrow v_{n+1} \rangle$, for $i = 1, 2, ..., d-1$. After deleting such a path between v_0 and v_{n+1} from D, whose sensor nodes form a FDSC, apparently, $SN^* = k - 1$ for the remaining segments (sensors). Here is an example: In Fig. 1, initially, $SN^* = 2$. After peeling off a path $\langle s_0, s_2, s_5, s_6, s_8 \rangle$ (s_0 and s_8 obviously won't be removed as they are virtual nodes), we have $SN^* = 1$. Note that as s_5 and s_7 are stabbed by the same critical base interval, the algorithm will not peel off a path like $\langle s_0, s_2, s_5, s_7, s_8 \rangle$.

If there are k vertex-disjoint paths between v_0 and v_{n+1}, then we can prove that $SN^* = k$ by contradiction. Assume that some critical base interval I only stabs $k' < k$ sensors (segments). As the paths between v_0 and v_{n+1} are vertex disjoint, some paths must use some sensors stabbed by I more than once. This is a contradiction.

The above proof implies a simple optimal algorithm for the FDSSP-UE whose pseudo code is given as Algorithm 1. In Algorithm 1, a shortest path (in terms of hops) between v_0 and v_{n+1} can be computed by the breadth-first search, which takes $O(|V_D| + |E_D|) = O(n^2)$ time in the worst case. So the overall running time of Algorithm 2 is $O(kn^2)$. ■

The algorithm for the FDSSP-NUE is presented as Algorithm 2. We summarize our result as follows.

Theorem 2: The FDSSP-NUE problem can be solved by Algorithm 2 in $O(n^4)$ time , where n is the number of sensors.

Proof: As in Theorem 1, we construct a directed *weighted* graph $W(V_W, E_W)$. Graph W is the same as graph D except that its edges are weighted. The weight w_e of an edge $e = \langle v_i, v_j \rangle$ is the minimum lifetime of v_i and v_j, i.e., $\min\{t_i, t_j\}$ where $t_i = \frac{E_{v_i}}{P}, t_j = \frac{E_{v_j}}{P}$. Let w^* be the minimum sum of lifetimes of all sensors covering any base interval. Let $I(w^*)$ be the base interval corresponding to w^*. We have the following claim: the maximum network lifetime $T = w^*$.

Algorithm 1 Algorithm for the FDSSP-UE

Step 1 Construct a directed acyclic graph $D(V_D, E_D)$ where $V_D = V \cup \{v_0, v_{n+1}\}$ and there is an edge from v_i to v_j iff $b_i < b_j \leq c_i < c_j$. v_0 and v_{n+1} are the two virtual (leftmost and rightmost) nodes.

Step 2 Compute all the base intervals of L and identify the critical base intervals. Let these critical base intervals be ordered as $I_1, I_2, ..., I_d$. For each $v_{i,l}$ covering I_i and $v_{i+1,l}$ covering I_{i+1} such that $v_{i,l}$ and $v_{i+1,l}$ do not cover the same base interval in $\{I_1, I_2, ..., I_d\}$ unless $v_{i,l} = v_{i+1,l}$, we compute the shortest path between $v_{i,l}$ and $v_{i+1,l}$ in graph D, for $i = 1, 2, ..., d-1$ and $l = 1, 2, ..., k$. Let this path be $\langle v_{i,l} \rightsquigarrow v_{i+1,l} \rangle$.

Step 3 Concatenate these shortest paths to obtain k vertex-disjoint paths between v_0 and v_{n+1}; i.e., $\langle v_0 \rightsquigarrow v_{1,l} \rangle \circ \langle v_{1,l} \rightsquigarrow v_{2,l} \rangle \circ \cdots \circ \langle v_{i,l} \rightsquigarrow v_{i+1,l} \rangle \circ \cdots \circ \langle v_{d-1,l} \rightsquigarrow v_{d,l} \rangle \circ \langle v_{d,l} \rightsquigarrow v_{n+1} \rangle$, for $i = 1, 2, ..., d-1$ and $l = 1, 2, ..., k$. The sensor nodes on each path between v_0 and v_{n+1} form a FDSC.

It is not difficult to prove the above claim. First we show that $T \leq w^*$. Think of network lifetime as a network flow, then the maximum flow corresponds to the lifetime of a FDSC. Obviously, w^* is such a maximum flow, and using up all these coverage lifetimes of $I(w^*)$ implies that $I(w^*)$ can not be covered anymore, i.e., the graph W is not connected anymore. We now show $T \geq w^*$ in a constructive way. The algorithm is iterative and similar to Algorithm 1. At step $i = 1$, we can identify w^* and compute a shortest path P_1 from v_0 to v_{n+1}. On this path, we identify the minimum edge weight w_1 and then subtract w_1 from all the edge/node weights for edges/nodes on P_1 (the edge/node with a weight of zero after the subtraction should be deleted from W). Then we update $w^* = w^* - w_1$ and repeat the procedure with $i = 2, 3, ...$ This process can be repeated at most $O(n^2)$ times in the worst case since at each iteration, we delete an edge from W and there are at most $O(n^2)$ edges in W. At each iteration, we need to compute the shortest path in the current graph W, which takes $O(n^2)$ time. So the overall time complexity of this algorithm is $O(n^4)$. ■

B. Variable Directional Sensor Scheduling

We suspect the VDSSP is NP-hard since it is similar to the well-known set cover problem. We present a simple and fast heuristic algorithm in the following to solve it, whose effectiveness will be justified by simulation results in Section V. The input includes the sensor node set V and the chain L to be covered. The output includes a variable directional sensor schedule Γ and its lifetime T.

In Step 2 of this algorithm, a VDSC is constructed and stored in S_C. Note that for the VDSSP, we not only need to determine which sensor node should be selected but also which sector of that sensor node should be used. S is used to include the current set of sectors that can potentially be selected into S_C. In order to guarantee the full coverage,

Algorithm 2 Algorithm for the FDSSP-NUE

Step 1 Construct a weighted directed acyclic graph $W(V_W, E_W)$, where $V_W = V \cup \{v_0, v_{n+1}\}$ and there is an edge from v_i to v_j iff $b_i < b_j \leq c_i < c_j$. The weight of a node v_i is its lifetime t_i; moreover, the weight of edge $\langle v_i, v_j \rangle$ is $\min\{t_i, t_j\}$. v_0 and v_{n+1} are the two virtual nodes.

Step 2 Compute all the base intervals of L and identify the base interval B which is covered by some sensor nodes $B_V \subseteq V$ whose sum of lifetime w^* is the minimum.

Step 3 i:=1.

Step 4 Repeat Step $5-6$ until $w^* = 0$.

Step 5 Compute a shortest path P_i from v_0 to v_{n+1}. Identify the edge e_i on P_i with the minimum edge weight w_i. Output all sensor nodes on P_i as a FDSC with lifetime w_i.

Step 6 Update the edge weight for every edge on P_i by subtracting w_i from all the node/edge weights for all sensor nodes/edges on P_i. Delete e_i from W. Update $w^* := w^* - w_i$ and $i := i + 1$.

Algorithm 3 Algorithm for the VDSSP

Step 1 $\Gamma := \emptyset$; $T := 0$;

Step 2 $S_C := \emptyset$;
$S := \{(v,i) | (v,i)$ covers the left end of L$\}$;
while (Chain L is not fully covered)
forall $(v,i) \in S$
$$f(v,i) := \alpha \times \frac{E_v}{E_v^{\max}} + \beta \times \frac{l_{vi}^{'}}{l_v^{\max}} + \gamma \times \frac{\sum_{v \in N_{vi}^{'}} (\frac{E_v}{P})}{\sum_{v \in N_{vi}} (\frac{E_v}{P})};$$
endfor
$(v_{\max}, i_{\max}) = \text{argmax}\{f(v,i) | (v,i) \in S\}$;
$S_C := S_C + \{(v_{\max}, i_{\max})\}$;
$S := \{(v,i) | (v,i) \in \Pi \setminus S_C$, overlaps with $(v_{\max}, i_{\max})$, and $l_{vi}^{'} > 0\}$;
if $(S = \emptyset)$ $S_C := \emptyset$; **break**; **endif**
endwhile

Step 3 **if** $(S_C = \emptyset)$ **return** Γ, T; **endif**
$\Gamma := \Gamma + S_C$; $T := T + T(S_C)$;
forall $(v,i) \in S_C$
$E_v := E_v - T * P$;
endfor
goto Step 2;

the VDSC selection starts from the left end of the chain, i.e., only the set of sectors which can cover the left end are considered first and are added to S. Moreover, the selection procedure progresses sequentially along the chain, i.e., the set of sectors which cover the uncovered portion of the chain and overlap with the newly selected sector are considered. Every time, the sector with the maximum weight value (given by the weight function $f(\cdot)$) will be selected. The algorithm intends to select the node with relatively high residual energy and the sector covering relatively a large uncovered portion of the chain. In the weight function $f(\cdot)$, E_v is the residual energy of node v and $E_{\max} = \max\{E_v | (v,i) \in S\}$. $l_{vi}^{'}$ is the length of the uncovered portion of the chain which can be covered by sector (v,i). $l_v^{\max}$ is the maximum length of the uncovered portion of the chain which can be covered by a usable sector in node v. In addition, the sector creating a critical area which can not be covered by the other sectors that have not been selected for a relatively long time, is not preferred to be selected. This is achieved by the third item of the weight function $f(\cdot)$, where $N_{vi}^{'}$ is the set of sensor nodes which can cover the bottleneck base interval created by $S_C + \{(v,i)\}$ but has not yet be selected, and N_{vi} includes all sensor nodes in V which can cover such a bottleneck base interval. Here the bottleneck interval is the base interval with the minimum remaining coverage lifetime among all the based intervals created by $S_C + \{(v,i)\}$. In addition, after conducting extensive simulations, we found out the algorithm offers the best performance when we set $\alpha = \beta = 0.4$, $\gamma = 0.2$.

Step 2 can be done in $O(nq_{\max})$ time, where $n = |V|$ and $q_{\max}$ is the maximum number of usable sectors of a sensor node. There are at most $O(n)$ VDSCs. So the time complexity of Algorithm 3 is $O(n^2 q_{\max})$. In practice, usually there are only $O(1)$ VDSCs and $q_{\max}$ is also small. In this case, the time complexity of Algorithm 3 is only $O(n)$.

IV. Distributed Scheduling Algorithms

In this section, we present two distributed algorithms for the FDSSP-NUE and the VDSSP respectively. In both of our algorithms, a sensor node is in one of the three states: DECIDE, WORK, and SLEEP during its lifetime. The time domain is divided into multiple rounds. The duration of a round depends on the residual energy of the working nodes in that round, which will be discussed shortly. At the beginning of each round, all sensor nodes turn on their radios, set their states to DECIDE and carry out the operations of selecting working nodes and deciding the duration of this round. At the end of such operations, each node either turns off its sensor and radio, and changes its state to SLEEP, or turns on its sensor and sets its state to WORK. In addition, all nodes remain their SLEEP/WORK states until the next round.

Similar to Algorithm 3, the basic idea of the distributed algorithms is to determine the set of working sensor nodes based on certain weight functions. We first describe our distributed algorithms for the FDSSP. At the beginning of each round, every node v tries to elect itself as a working node. It starts a backoff timer expiring after t_v time units which is given by Equation (1).

$$t_v = (1 - \alpha \times \frac{E_v}{E_v^{\max}} - \beta \times \frac{l_v^{'}}{l_v} - \gamma \times \frac{\sum_{u \in C_v^{'}} (\frac{E_u}{P})}{\sum_{u \in C_v} (\frac{E_u}{P})}) \times t_s \quad (1)$$

In this equation, E_v is the residual energy of node v. $E_v^{\max} = \max\{E_u | u \in N_v\}$, where N_v includes those nodes whose coverage regions overlap with v but whose states have not been decided. $l_v^{'}$ is the length of the portion of the *uncovered* chain which can be covered by sensor node v. Here by uncovered chain, we mean the portion of the chain that has not been covered by sensor nodes deciding to work. l_v is

the total length of the portion of chain which can be covered by v. Every base interval I within sensor node v's coverage range has a remaining coverage lifetime equal to $\sum_{u \in N_I} (\frac{E_u}{P})$, where N_I includes all nodes which can cover the base interval I but have not been selected as working nodes. Note that N_I does not include node v. Similar as above, the *bottleneck* base interval of node v is defined as a base interval within node v's coverage range with the minimum remaining coverage lifetime. C_v is the set of sensor nodes covering node v's bottleneck base interval. $C_v^{'}$ is a subset of C_v only including those nodes which are not v and have not been selected as working nodes yet. t_s is the maximum backoff duration, which was set to 10ms in our simulation. Note that all the above information can be easily obtained since $R \geq 2r$, i.e., all those nodes whose coverage regions overlap with node v's region are their direct neighbors. According to Equation (1), those nodes having relatively high residual energy and covering relatively large portion of the uncovered chain will have a relatively smaller backoff duration and will be more likely to become a working node. Moreover, if selecting v as a working node leads to a situation where there is a certain critical area on the chain which can not be covered by the sensor nodes that have not been selected as working nodes for a relatively long time, then it is not favorable to make v work in this round. This can be achieved by the third item of the equation.

When the timer fires, if node v does not receive any NODE-ON message from its neighbors, it turns on its sensor, changes its state to WORK and computes the round duration by $t_d = \frac{aE_v}{P}$, where P is the energy consumption rate and a is a tunable parameter that can be set to any real number in $[0, 1]$. We set it to 0.5 in our simulation. It then broadcasts a NODE-ON message including its location and the round duration t_d. Note that $\frac{E_v}{P}$ gives a prediction for the node's remaining lifetime, i.e., how much longer it can work. If we set the round duration to its remaining lifetime, this node may die at the end of this round. However, our algorithm may not yield the best solution for every round. So with the parameter a, the network can have chances to re-select the working sensor node sets, which hopefully will lead to longer network lifetime.

Whenever a node u receives a NODE-ON message from one of its neighbors before its own timer fires, it first checks whether the portion of the chain in its coverage range has been covered by the existing set of working nodes in the neighborhood. If so, it broadcasts a NODE-OFF message, turns off its sensor and radio, and sets its state to SLEEP. Otherwise, it updates all the information needed by Equation (1), and updates its timer to $(t_v - t_p)$, where t_p is the time elapsed since the beginning of the round, and t_v is the new timeout value computed by Equation (1) based on the updated information. In addition, it compares the round duration value included in the NODE-ON message t_d and its current duration t_d^u. If $t_d < t_d^u$, it will update $t_d^u := t_d$ and broadcast a DURATION message including t_d. All the nodes receiving such a message will compare their current duration value with t_d. If t_d is smaller, they will update their own t_d^u and further broadcast it to its neighbors. Eventually, all the working nodes are supposed to reach an agreement on the value of round duration. Note that a node deciding to sleep will turn off its radio at $(T_0 + t_s + t_g)$, where T_0 is the round starting time and t_g is a guard duration that can be set to a small value. This is because that even a sleeping node needs to know the round duration, i.e., when to wake up. Our working node selection procedure will roughly end at $(T_0 + t_s)$. In addition, the node will stop trying to be a working node if its remaining lifetime is less than t_s.

Our distributed algorithm for the VDSSP is the same as our FDSSP algorithm described above except for the backoff timer setup part. For the VDSSP, each node not only needs to decide whether it should go to work or sleep but also which sector to use if it decides to work. So each usable sector (v, i) of a node v will have a corresponding backoff duration which can be computed by Equation (2).

$$t_{vi} = (1 - \alpha \times \frac{E_v}{E_v^{\max}} - \beta \times \frac{l_{vi}^{'}}{l_v^{\max}} - \gamma \times \frac{\sum_{u \in C_{vi}^{'}} (\frac{E_u}{P})}{\sum_{u \in C_{vi}} (\frac{E_u}{P})}) \times t_s \quad (2)$$

In the equation, $l_{vi}^{'}$ is the length of the uncovered portion of the chain which can be covered by sector i of sensor node v. $l_v^{\max}$ is the maximum length of the uncovered portion of the chain which can be covered by a sector of sensor node v. If $l_{vi}^{'} = 0$, then this sector will not be considered since it can not make any contribution for the coverage. If a node v finds out this value is equal to 0 for every usable sector, then it will go to sleep. C_{vi} is the set of sensor nodes covering the bottleneck base interval of sector (v, i). $C_{vi}^{'}$ is a subset of C_{vi} only including those nodes which are not v and have not been selected as working nodes yet. We say that a node which has not been selected as a working node can cover a base interval if at least one of its sectors can cover it. Here, we only consider those base intervals created by sector i of node v and the sectors of the nodes which already decide to work. The definition for the bottleneck base interval is the same as before. Node v's backoff duration $t_v = \min\{t_{vi} | 1 \leq i \leq q_v\}$, where q_v is the number of usable sectors of node v. If node v happens to become a working node, it will use the sector that gives the minimum backoff duration. In addition, same as the fixed case, before the backoff timer fires, a node needs to update its timer whenever it receives a NODE-ON message.

V. Performance Evaluation

We evaluated the performance of the proposed algorithms via simulations. The distributed algorithms were evaluated using NS2 [10]. Simulations were conducted on a portion of the Grand Loop Road near Tower Roosevelt in the Yellowstone National Park with a length of 1000m. The roadway information was obtained from the Google Maps. In our simulations, the sensing range $r = 100$m [4] and the transmission range $R = 300$m [9]. For every fixed directional sensor, we assumed that it has a FOV of $\frac{\pi}{3}$ and an orientation of $\frac{\pi}{2}$ or $\frac{3\pi}{2}$ (depending on which side of the road it is placed on). We first randomly placed a certain number of sensor nodes on the two sides of the selected roadway. We then identified coverage gaps and covered them by placing more sensor nodes nearby. Similar as in [2], the energy consumption rate of a working sensor node was set to 0.02. The other algorithm

related parameters were set as follows: $\alpha = \beta = 0.4$, $\gamma = 0.2$, $t_s = 10$ms and $a = 0.5$. The network lifetime is used as the performance metric.

We set the number of sensor nodes (n), FOV (θ) and the initial energy (E_v) to different values in different scenarios. The results are presented in Fig. 2–4. Here, $E_v = [400, 600]$ means E_v was set to a random integer uniformly distributed in $[400, 600]$. In the figures, the label "Fixed" refers to the centralized optimal algorithms (Algorithm 1 or Algorithm 2) for the FDSSP. If all the sensor nodes have the same initial energy, Algorithm 1 was used. Otherwise, Algorithm 2 was used. "Fixed-D" refers to the distributed algorithm for the FDSSP. Similarly, "Variable" refers to Algorithm 3 and "Variable-D" refers to the distributed algorithm for the VDSSP.

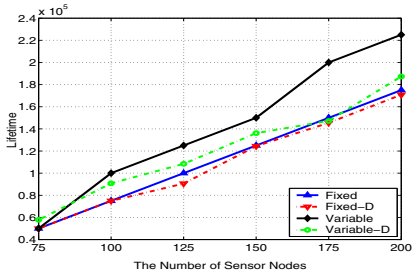

Fig. 2. Scenario 1: $E_v = 500$, $\theta = \frac{\pi}{3}$, $R = 300$m

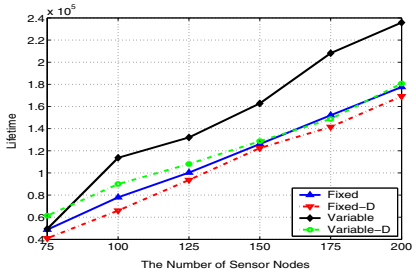

Fig. 3. Scenario 2: $E_v = [400, 600]$, $\theta = \frac{\pi}{3}$, $R = 300$m

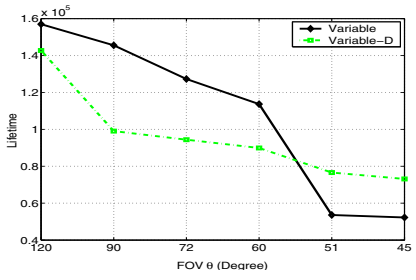

Fig. 4. Scenario 3: $E_v = [400, 600]$, $n = 100$, $R = 300$m

We make the following observations from the results:

1) The lifetime values given by the optimal centralized algorithms for the FDSSP are used as a baseline for comparisons since there is no previous solution available. From Figs. 2–3,we can see that the lifetime values given by the distributed FDSSP algorithm are always very close to the corresponding optimal values. On average, there is only a 4.6% difference. Our centralized VDSSP algorithm outperforms the centralized FDSSP algorithms by 19.5% on average. This is because the VDSSP algorithm can select the most proper sector to cover the uncovered chain at each step, which will hopefully lead to smaller working sensor sets therefore longer network lifetimes. As expected, the performance of the distributed VDSSP algorithm is not as good as that of the centralized VDSSP algorithm in most cases due to the lack of global information. However, it still outperforms the centralized FDSSP algorithms by 4.5% on average. In addition, the network lifetime always increases with the number of sensor nodes no matter which algorithm is employed.

2) Basically, the network lifetime decreases with the decrease of the FOV, which can be seen from Fig. 4. This is because a smaller FOV usually leads to larger working sensor sets therefore shorter network lifetimes.

VI. Conclusions

In this paper, we studied the scheduling problems for maximum lifetime chain coverage in directional sensor networks. We have formally formulated them as the FDSSP and the VDSSP. We presented optimal algorithms to solve two versions of the FDSSP in polynomial time. Moreover, we provided an effective heuristic algorithm to solve the VDSSP. We also presented distributed algorithms for both problems. The simulation results based on a roadway in the Yellowstone National Park have shown that our distributed algorithm for the FDSSP provides close-to-optimal performance. Moreover, our VDSSP algorithms (centralized and distributed) outperform the FDSSP algorithms.

References

[1] J. Adriaens, S. Megerian and M. Potkonjak, Optimal worst-case coverage of directional field-of-view sensor networks, *IEEE SECON'2006*, pp. 336–345.

[2] J. Ai and A. A. Abouzeid, Coverage by directional sensors, *IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2006.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: a Survey, *Computer Networks Journal*, Vol. 38, No. 4, 2002, pp. 393–422.

[4] Y. Cai, W. Lou, M. Li and X-Y Li, Target-oriented scheduling in directional sensor networks, *IEEE Infocom'2007*, pp. 1550–1558.

[5] M. Cardei, M. T. Thai, Y. Li and W. Wu, Energy-efficient target coverage in wireless sensor networks, *IEEE Infocom'2005*, pp. 1976–1984.

[6] G. Fusco and H. Gupta, Selection and orientation of directional sensors for coverage maximization, *IEEE SECON'2009*.

[7] C-F. Huang and Y-C. Tseng, The coverage problem in a wireless sensor network, *ACM Journal of Mobile Networks and Applications*, Vol. 10, No. 4, 2005, pp. 519–528.

[8] H. Ma, X. Zhang, A. Ming, A coverage-enhancing method for 3D directional sensor networks, *IEEE Infocom'2009*, pp. 2791–2795.

[9] Mica2 data sheet, *http://www.xbow.com*.

[10] The Network Simulator - NS-2, *http://www.isi.edu/nsnam/ns/*

[11] D. Tian and N. D. Georganas, A coverage-preserving node scheduling scheme for large wireless sensor networks, *ACM International Workshop on Wireless Sensor Networks and Applications*, 2002, pp. 32–41.

[12] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, *ACM SenSys'2003*, pp. 28–39.

[13] Road Ecology Program in Western Transportation Institute, *http://www.coe.montana.edu/wti/road_ecology/home.php*.

[14] H. Yang, D. Li and H. Chen, Coverage quality based target-oriented scheduling in directional sensor networks, *IEEE ICC'2010*.

[15] H. Zhang and J.C. Hou, Maintaining sensing coverage and connectivity in large sensor network, *Journal of Ad Hoc and Sensor Wireless Networks*, Vol. 1, No. 1–2, 2005, pp. 89–124.