Distributed Saturation Degree Based TDMA Scheduling Algorithm for Target Tracking

Fan Zhang*, Peng Cheng *, Jiming Chen*†, Youxian Sun* and Xuemin (Sherman) Shen†
*State Key Lab. of Industrial Control Technology, Zhejiang University, Hangzhou, 310027 China
Email: {zhangfan, pcheng, yxsun}@iipc.zju.edu.cn
† Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
Email: {jmchen, xshen}@bbcr.uwaterloo.ca

Abstract—**In target tracking applications, active ultrasonic sensors can provide satisfactory distance estimations, but also suffer from inter-sensor-interference when they are not well scheduled. In this paper, we propose a distributed saturation degree based algorithm (DSDA), which assigns the TDMA slot in wireless sensor networks distributively in order to avoid the interference. By adopting a graph coloring technique, *Saturation Degree Heuristic*, this new algorithm can provide near-optimal slot number in a totally distributed way. Simulation results demonstrate the efficiency of DSDA in terms of slot number, system scalability, tracking accuracy and energy consumption.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have been considered as a promising technique for area surveillance applications [1] [2], and target localization/tracking is essential for these applications. Many approaches have been proposed for target tracking within WSNs [2] [3] [4]. According to target behavior, most of the previous works can be categorized into two classes, cooperative [2] [3] and non-cooperative [4] [5]. A cooperative target is part of the network and emits certain forms of physical signals that reveal its presence or report its own identification. In the non-cooperative scenario, however, there exists no information exchange between the target and the network infrastructure. Therefore, sensor nodes need to detect and identify the target *actively* by emitting energy and measuring the feedback. Our previous work [5], a tracking system aiming at non-cooperative targets, utilized passive infrared sensors for target detection and the active ultrasonic sensors for ranging.

In non-cooperative tracking systems using echo-based sensors like ultrasonic sensors, there is severe Inter-Sensor Interference (ISI) when nearby active sensors work simultaneously in the same frequency band. Such interference results in erroneous sensor readings and leads to unacceptable tracking results. Therefore, sensor scheduling is necessary to ensure that in an ISI region the number of sensors which are detecting should never be more than one at any specific time. In fact, by regarding the ranging operation of a sensor node as the occupation of a shared channel, the ISI problem among active ultrasonic sensors in WSNs can be converted to the problem of multiple access in a shared channel. Hence the schemes for media access control (MAC) can be used to solve the ISI problem.

A common MAC paradigm is CSMA (Carrier Sense Multiple Access), which is considered as a simple, flexible and robust contention based access protocol. However, it still suffers from the collisions which can cause serious energy waste, high overhead and throughput degradation [6]. The DSS algorithm proposed in [7] is kind of this fashion and it requires nodes negotiate with each other frequently to decide the tasking node which results in high energy consumption.

Another classic access scheme is TDMA (Time Division Multiple Access) which schedules transmission slots of neighboring nodes to occur at different times. It often uses topology information as a basis for access scheduling and needs clock synchronization among neighbors, thus lacks efficiency and scalability when the networks subject to frequent topology changes. However, due to its collision-free and energy efficient properties, TDMA still has a unique place in the access design space for WSNs [8], particularly where most of nodes can be assumed to be stationary as in our tracking scenario.

In this paper, in order to tackle the ISI problem for active sensors in WSNs, we model it as a TDMA slot assignment problem. By utilizing the graph coloring theory, we design a Distributed Saturation Degree based Algorithm (DSDA). This new algorithm assigns the time slots for each sensor in a distributed and parallel way which has a low complexity. Furthermore, it also benefits better tracking performance due to the increased sampling rate in terms of reduced slot number.

The remainder of the paper is organized as follows: Section II provides a discussion about the relationship between the TDMA slot assignment and the graph coloring problem. Section III presents the details of DSDA, and Section IV evaluate the algorithm with extensive simulations. Finally, conclusions and future work are given in Section V.

II. GRAPH COLORING AND TDMA SLOT ASSIGNMENT

Graph coloring problem has been regarded as a convenient equivalent for the channel assignment problem in time, frequency and code domains [9]. In this section, we first provide some background of graph theory and then show how it can be mapped to TDMA slot assignment problem for active sensors to avoid the ISI. Finally, we define some metrics which will be used to evaluate the proposed algorithm later.

Graph coloring is a classic problem in graph theory. Given a simple graph $G = (V, E)$, where V is the set of vertices, and

978-1-61284-233-2/11/$26.00 ©2011 IEEE

E is the set of edges. For each vertex v define its neighborhood $N(v) = \{u : \{u, v\} \in E\}$ and vertex degree $d(v) = |N(v)|$. Then the vertex coloring problem is to find a color assignment for the vertices of G: $V(G) \rightarrow F$, where F is a set of colors, usually represented by some small subset of positive integers, so that no two adjacent vertices are given the same color.

In the TDMA slot assignment, the real time is divided into non-overlapping periodic cycles called *time frames* which are also divided into non-overlapping equal time periods called *time slots*. From the view of graph theory, we consider the nodes as vertices and the interference relations as edges. There is an edge between nodes u and v if and only if the Euclidean distance between them does not exceed $2r_d$, where r_d is the detection range of the ultrasonic sensor [4]. After each node gets its color which represents a TDMA time slot, it uses that slot at each time frame for interference-free sensing.

Therefore, in order to improve the tracking performance, it is ideal to minimize the number of used colors so that a higher sampling rate can be achieved. However, the optimal solution is NP-hard [10]. Considering time efficiency and energy cost, we will use three metrics to evaluate the heuristic algorithm for the TDMA slot assignment problem:

- **Number of Used Color**: the number of colors required to properly color the entire graph by the given algorithm.
- **Running Time**: the amount of time taken by the given algorithm to color all the nodes in the graph.
- **Message Complexity**: the number of messages transmitted by all the nodes during the color assignment phase with the given algorithm.

III. DISTRIBUTED SATURATION DEGREE BASED ALGORITHM

In this section, we present the proposed DSDA, which is inspired by a centralized heuristic coloring algorithm DSATUR [11]. DSATUR addressed *Saturation Degree* to dynamically arrange the order of vertices to be colored, and used less extra colors than the other heuristic solutions. Therefore, to meet the main design goal, we use the saturation degree heuristic as the basic of our algorithm.

A. Why Saturation Degree?

Saturation Degree of vertex v, denoted by $sd(v)$, is defined as the number of different colors assigned to the neighbors of v. Intuitively, a vertex with higher saturation degree should be more careful when selecting a color for itself. The DSATUR algorithm always colors the vertex with the largest saturation degree in the entire graph, therefore it arrives at a very efficient order for coloring the graph.

The distributed algorithms are preferred in most WSN applications as it is usually difficult for each node to access global information of the whole network. Here, in order to utilize *Saturation Degree Heuristic* in a distributed way, we introduce a quantity named *SDI* (Saturation Degree Indicator) for each node to calculate the priority of coloring for itself and its uncolored neighbors distributively:

$$SDI_v(u) = (\Delta(v) + 1) \cdot sd(u) + d(u) + rand(u)$$

in which

$$u \in S(v) = UN(v) \bigcup \{v\}$$
$$\Delta(v) = \max_{u \in S(v)} d(u)$$

where $UN(v)$ denotes the uncolored neighborhood of v, and $rand(u) \in (0, 1)$ which is used to break the ties is a random value generate by node u. Note that $\Delta(v)$ is a constant value for node v if the graph structure does not change.

We call node v an *Extremum Node* if condition $SDI_v(v) = \max_{u \in S(v)} SDI_v(u)$ is satisfied, which means v has higher priority of coloring than all its uncolored neighbors.

B. Algorithm Specification

1) Assumptions: Before we present the details of DSDA, several assumptions are made below.

- Each node has a unique node ID, and all nodes in the network are synchronized.
- At the initial state, each node has information about IDs, degrees and the random values of its neighbors, thus $\Delta(v)$ is known for each node v.
- The range of the radio signal is set to be equal to the ISI range, i.e., $2r_d$ by modifying the transmission power.
- The transmission is guarantied by the low level protocol so that each one-hop message can be successfully delivered.

2) Algorithm Details: The idea behind DSDA is to let each node decide its own slot according to the information collected from its neighbor nodes. There are two data types used to store the neighborhood information for each node. Let $LC(v)$ be a data type for a list of colored neighbors of v and their colors, $LUC(v)$ be a data type to record the information of uncolored neighbors of v, including their saturation degrees, degrees and random values. Thus, we can get $sd(v)$ from $LC(v)$, and $SDI_v(u)$ from $LUC(v)$. In addition, there is a positive real value named age attached to each item in LUC, as explained below.

DSDA runs in rounds and each round is divided into two subrounds: $subrnd1$ and $subrnd2$. The algorithm, running in parallel on each node and during each round the detailed procedure is summarized below.

During $subrnd1$, by using a constant $maxAge$ which is set as an upper bound for the item age in LUC, each uncolored node A firstly updates the age of each item in $LUC(A)$ and deletes the expired items which is caused by node death. After that, A calculates $SDI_A(B)$ for itself and its uncolored neighbors to find out whether it is an *Extremum Node*.

If A is an *Extremum Node* which implies that it can be colored in this round, it picks its color to be the minimum of the colors that have not been taken by its neighbors before this round and this information is known from $LC(A)$. Then A enters the *colored* state and broadcasts a $release$ message containing its selected color.

During $subrnd2$, on receiving the $release$ message, each A's uncolored neighbor B will update its neighborhood information and broadcast an $update$ message containing the updated $sd(B)$, $d(B)$, and $rand(B)$ to its neighbors, so that they can update the item of B in LUC and refreshes the age of this item to zero. On the other hand, if an uncolored node C has not receive any $release$ message in recent $maxAge$ rounds, it will also broadcast an $update$ message with the unchanged $sd(C)$, $d(C)$, but a new $rand(C)$ to prevent being deleted by its neighbors.

The pseudo code of proposed algorithm executed on each node is described in Algorithm 1.

Algorithm 1 Distributed Saturation Degree Based Algorithm

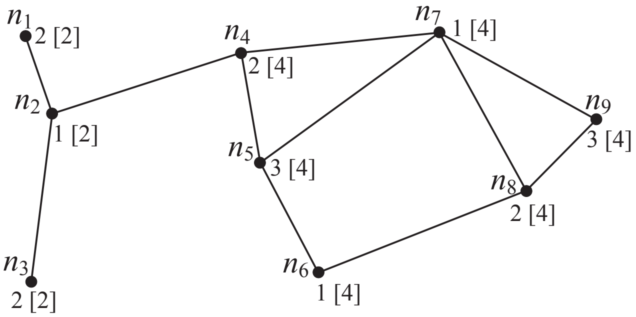
```
let $A$ be an arbitrary node in the network

During subrnd1
if $A.color = 0$ (uncolored) then
    for each item $B$ in $LUC(A)$ do
        $age_B = age_B + 1$;
        if $age_B > maxAge$ then
            delete item $B$ from $LUC(A)$;
        else
            calculate $SDI_A(B)$;
        end if
    end for
    if $A$ is Extremum Node then
        $A.color$ = minimum unused color within its neighbors;
        broadcast release;
    end if
end if

During subrnd2
if $A.color = 0$ (uncolored) then
    if receive release from $B$ in subrnd1 then
        transfer the item of $B$ from $LUC(A)$ to $LC(A)$;
        calculate the updated $sd(A)$;
        generate a new $rand(A)$;
        broadcast update;
    else
        if not receive any release message in recent $maxAge$ rounds then
            generate a new $rand(A)$;
            broadcast update;
        end if
    end if
    if receive update from $B$ then
        update the item of $B$ in $LUC(A)$;
        $age_B = 0$;
    end if
end if
```

3) Local Time Framing: After picking a time slot, each node needs to decide the time frame size during which it can use the slot for sensing periodically. Once a node finds that all its neighbors and itself have picked a time slot, it changes into *framing* state to decide its local time frame. Consider a node v, and let C_v denote the maximum color number of any node in $N(v)\bigcup\{v\}$. Then, the time frame for v is set to be 2^a where a is the positive number satisfying $2^{a-1} < C_v \leq 2^a$.

With this scheme, each node can determine its own time frame size only on the basis of local neighborhood information and thus makes DSDA able to handle local topology changes gracefully without incurring any global changes. Fig. 1 shows an example of a TDMA schedule derived from DSDA.



The number beside a node indicates the color assigned by DSDA and the number in the bracket is its frame size after local time framing. Then the slot schedule of all nodes is shown as follows.

1	2	3	4	1	···
n_2	n_1	n_5	n_1	n_2	···
n_6	n_3	n_9	n_3	n_6	···
n_7	n_4	n_2		n_7	···
	n_8				···

Fig. 1. Schedule Derived Using DSDA

C. Analysis and Discussion

In this section we analyze the correctness and performance of DSDA. We first show that the execution of DSDA results in a valid conflict-free TDMA time slot schedule, and then analyze the time complexity and message complexity of DSDA, respectively.

1) Correctness: Given the above scheduling algorithm, the correctness of DSDA is guaranteed by the following theorem:

***Theorem** 1:* Each two adjacent nodes are assigned different colors.

Proof: This can be easily proved by the following three arguments: (1) a node does not select a color unless and until it becomes *Extremum Node*. (2) At any round, since all nodes have different random values, the *Extremum Nodes* who have locally highest priority can never be adjacent. (3) *Extremum Node* always selects the minimum color that is not taken by its neighbors. ■

***Theorem** 2:* After local time framing, each two adjacent nodes never use the same time slot.

Proof: We prove the theorem by contradiction. Consider two adjacent nodes i and j with the assigned color s_i and s_j respectively, $s_i \neq s_j$. After local time framing, i's time frame is set to be 2^a and j's is 2^b, i.e., i uses only slots $k \cdot 2^a + s_i$ and j uses $l \cdot 2^b + s_j$, for all $k, l = 0, 1, 2, \cdots$. If it happens that j uses one of the slots that is used by i, then for some k, l, we have $k \cdot 2^a + s_i = l \cdot 2^b + s_j$. Without loss of generality, we assume that $a \leq b$. Therefore we can get $s_i \equiv s_j (mod 2^a)$. Note that $1 \leq s_i, s_j \leq C_i \leq 2^a$, so s_i and s_j must be the same, which is a contradiction. ■

With the above two theorems, we can conclude that the TDMA time slot schedule created by DSDA is conflict-free.

***Theorem** 3:* DSDA never uses more than $\delta + 1$ colors.

It easily follows from the executions of the algorithm.

2) Complexity Analysis: For the time complexity, if no node is dead during the execution of color assign, then in each round, at least one node picks its color, i.e., the upper bound for the number of rounds that a node takes to acquire a slot is $|V|$. While, if node's death occurs during the color assignment, after up to $maxAge$ rounds, the information about death nodes will be deleted, thus, in every $maxAge$ round, at least one node is deleted or picks its color, then the upper bound on the number of rounds is $maxAge \cdot |V|$.

For the message complexity, in one round, a node can send $O(1)$ messages. Since there are no more than $O(|V|)$ rounds, each node can send $O(|V|)$ messages at most.

Although the analytic results shows that the time complexity and message complexity of DSDA are bounded by $O(|V|)$, while in practice, the number of rounds and the average number of message transmissions per node can be far less than that, and we conjecture that DSDA has an expected time and message complexity of $O(\delta)$, as we shall see from the simulation results, where δ is the maximum neighborhood size in the network, which is usually much smaller than $|V|$.

IV. SIMULATION RESULTS

The proposed algorithm is evaluated in the following two aspects: (1) the performance analysis in the previous section is verified; and (2) the effectiveness of the TDMA schedule generated by DSDA in target tracking scenario is evaluated by comparing with the existing DSS algorithm [7].

A. Validation of Analysis

Nodes are placed randomly on a $15m \times 15m$ square with a communication range of $4m$. The neighborhood size of the network δ is changed between 3 and 50 by varying the number of placed nodes from 16 to 150 and the presented results are based on 20 independent simulation runs for high confidence.

Time complexity and Message Complexity: Fig. 2 shows the number of rounds that all nodes have taken to decide their slots, and Fig. 3 shows the average number of message transmissions per node during the executions of the algorithm. In each figure, the data points represent the numerical results of DSDA for different networks and the solid line indicates the average value of these data points. We can see that both the the running time and the exchanged messages of DSDA follow the analytic results as they grow linearly with the size of the neighborhood.

Number of Used Color: According to Theorem 3, the number of used color is bounded by $\delta + 1$ in DSDA. While in practice, it can be much less than that, as shown in Fig. 4 and less used color always results in more efficiency TDMA time slot schedule.

Scalability: We examine the scalability of DSDA by ranging the number of nodes from 100 to 1000 in step of 100 and correspondingly increase the network size with the same node density. Numerical results given in Table I indicate that DSDA is scalable since, while the size of the network is increased, the running time (i.e., number of rounds) and the number of message transmissions per node remain relatively constant. DSDA performance does not directly affected by the network size, but only depend on the neighborhood size. The reason is that DSDA is a distributed scheduling algorithm that runs in parallel over different neighborhood areas within the network. In addition, the number of used color generated by DSDA is quite close to the result of DSATUR, the centralized near-optimal coloring algorithm, which reveals the efficiency of DSDA.

TABLE I
SCALABILITY OF DSDA

Node Number	Color Number (DSDA)	Color Number (DSATUR)	Round Number	Msg Number	δ
100	14.03	13.85	41.29	29.78	28.37
200	15.34	14.98	51.45	31.98	31.2
300	16.25	15.81	57.48	33.13	32.96
400	16.46	16.17	58.11	33.7	33.56
500	16.99	16.72	60.32	34.27	34.84
600	17.01	16.64	61.97	34.53	34.78
700	17.47	17.22	63	34.63	35.52
800	17.48	17.21	63.29	34.84	36.04
900	17.77	17.36	65.88	35.01	36.34
1000	17.73	17.46	65.22	35.06	36.4

B. Comparison of Tacking Performance

We compare the tracking performance of DSDA with DSS. Unless specified otherwise, we use the default settings of DSS and the same simulation setups as described in [7].

Firstly, we compare the two algorithms under a different setting of sensor quantity, ranging from 3 to 24. Then, we fix the node density, but increase the side length of the square monitored area from 3 meters to 24 meters in steps of 3 meters. As depicted in Fig. 5(a) and (b), and it can be seen that: (i) by increasing the number of deployed sensor nodes, the tracking error for both algorithms can be reduced; (ii) both DSDA and DSS are insensitive to the network scale which proves the scalability of our new algorithm; and (iii) the DSDA outperforms DSS when the network is not too dense which is understandable as the DSDA selects the tasking node in a deterministic way so that it can achieve a higher sampling rate. On the other hand, since DSDA can not completely eliminate the empty detection, it loses its ascendancy along with the increase of node density or network scale. Despite of this, the gap between the two algorithms is quite small.

Finally, we examine the energy cost for these two algorithms to get an effective schedule. By recording the total number of radio operations for each node, we can calculate the total energy by using the data reported by [12]. Fig. 5(c) shows that the energy cost for DSDA is constant because it only needs nodes to negotiate with each other at the setup phase. While for DSS, since it works like CSMA and requires nodes to

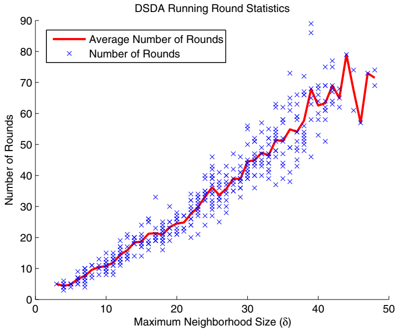


Fig. 2. DSDA Time Complexity

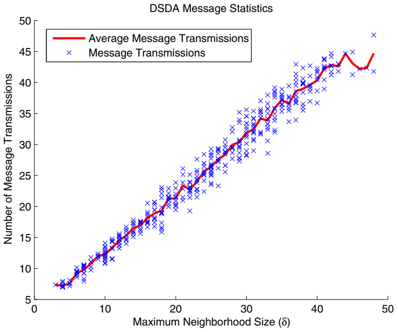


Fig. 3. DSDA Message Complexity

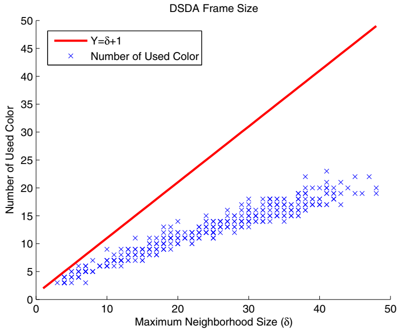


Fig. 4. Number of Used Color for DSDA

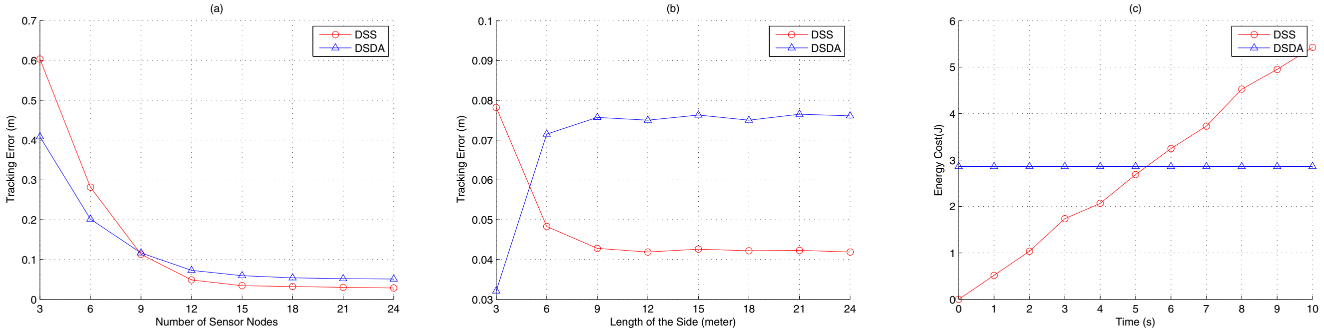


Fig. 5. Tracking Performance Comparison between DSS and DSDA

keep negotiating with others, the energy cost grows linearly with the simulation time.

V. CONCLUSION

In this paper, we have presented DSDA, a distributed, scalable and efficient TDMA scheduling algorithm, for active sensors in wireless sensor network to solve the ISI problem. By using the saturation degree heuristic, DSDA gives quite efficient slot assignments with near-optimal time frame size. Meanwhile, the running time and message complexity of DSDA is linear proportional to the size of the local neighborhood which ensures the adaptability and scalability of the algorithm. Simulation examples demonstrate that, compared with the DSS, our new algorithm obtains much better energy efficiency while guarantees similar tracking accuracy.

In the future, by considering the tradeoff between the tracking accuracy and the energy consumption, we will design a hybrid scheduling algorithm that combines the strengths of TDMA and CSMA. In addition, more efficient sensor scheduling algorithm could be designed with the estimation/prediction techniques such as EKF, unscented Kalman filter and particle filter. Tracking and adaptive sensor scheduling for multiple targets using multiple modalities are also our future work.

ACKNOWLEDGEMENT

This research is supported in part by Natural Science Foundation of China (NSFC) under grant no. 609741229, by Joint Funds of NSFC-Guangdong under grant no. U0735003, and by and under grant no. NSFZJ R1100324.

REFERENCES

[1] G. Werner-Allen, J. Johnson, M. Ruiz, J.Lees. Monitoring Volcanic Eruptions with a Wireless Sensor Network. In *Proceedings of EWSN*, pages 108-120, Istanbul, Turkey, Jan.-Feb. 2005.

[2] L. Klingbeil, T. Wark. A Wireless Sensor Network for Real-time Indoor Localisation and Motion Monitoring. In *Proceedings of IPSN*, pages 39-50, St. Louis, Missouri, USA, Apr. 2008.

[3] B. Kusy, A. Ledeczi, X. Koutsoukos. Tracking Mobile Nodes Using RF Doppler Shifts. In *Proceedings of ACM SenSys*, pages 29-42, Sydney, Australia, Nov. 2007.

[4] W. Xiao, J. Wu, L. Xie, L. Dong. Sensor Scheduling for Target Tracking in Networks of Active Sensors. *Acta Automatica Sinica*, vol. 32, no. 6, pages 922-928, Nov. 2006.

[5] H. Li, D. Miao, J. Chen, Y. Sun. X. Shen. Networked Ultrasonic Sensors for Target Tracking: An Experimental Study. In *Proceedings of IEEE GlobeCom*, Honolulu, Hawaii, USA, Nov.-Dec. 2009.

[6] K. Kredo. II, P. Mohapatra. Medium Access Control in Wireless Sensor Networks. *Computer Networks*, vol. 51, no. 4, pages 961-994, Mar. 2007.

[7] F. Zhang, J. Chen, H. Li, Y. Sun, X. Shen. Distributed Interfering Sensor Scheduling Scheme for Target Tracking. In *Proceedings of IEEE ChinaCom*, Aug. 2010.

[8] I. Rhee, A. Warrier, J. Min, L. Xu. DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks. In *Proceedings of ACM MobiHoc*, pages 190-201, Florence, Italy, May. 2006.

[9] S. Ramanathan. A Unified Framework and Algorithm for Channel Assignment in Wireless Networks. *Wireless Networks*, vol. 5, no. 2, pages 81-94, Mar. 1999.

[10] D. B. West. Introduction to Graph Theory. Senond Edition, Prentice Hall, 2001.

[11] D. Brélaz. New Methods to Color The Vertices of a Graph. *Communications of the ACM*, vol. 22, no. 4, pages 251-256, Apr. 1979.

[12] J. Polastre, J. Hill, D. Ciller. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of ACM SenSys*, pages 95-107, Baltimore, MD, USA, Nov. 2004.