

# A New Density Control Algorithm for WSNs

Yi Shang, Jianli Shen, and Hongchi Shi  
Department of Computer Science  
University of Missouri-Columbia  
Columbia, MO 65211  
{ShangY,ShiH}@missouri.edu

**Abstract**— In dense sensor networks, density control is an important technique for prolonging network’s lifetime while providing sufficient sensing coverage. In this paper, we present a new density control algorithm called NODC (Non-Overlapping Density Control). The goal of NODC is maximizing coverage while avoiding the overlap of sensing areas of active sensors. We derive the optimality conditions and develop a distributed algorithm. Experimental results show that NODC can use up to 20% fewer active nodes than OGDC with less than 5% reduction in coverage.

## I. INTRODUCTION

Many sensor network applications require sensors to monitor and collect data in a region of interests. Due to the limited energy resource in each sensor node, sensors are usually deployed in high density and take turns to work in order to prolong the network lifetime [1]. Density control techniques have been developed to determine when and what sensors should be powered up and what should be put into energy saving mode.

A general approach is to divide the operating time into many rounds and set different nodes active in different rounds depending on their energy levels and other attributes.

In AF [3], the region of interest is divided into virtual grid, and one node in each grid is selected to be “on” in each round. In Ascent [4], each node self-determines its participation based on its contribution to network connectivity by checking its local node density and measuring message loss rate. In PEAS (Probing Environment and Adaptive Sleeping) [5], sleeping nodes wake up once a while to probe their neighborhood and replace any failed working node. It does not guarantee the sensing area of a sleeping node is completely covered by other nodes. In OGDC [2], an optimal geographical density control algorithm is proposed based on the optimality conditions under which a subset of working sensor nodes can be chosen to achieve full coverage.

## II. NON-OVERLAPPING DENSITY CONTROL (NODC)

In this paper, we extend OGDC and propose a new density control algorithm called NODC (Non-Overlapping Density Control) by considering the trade-offs between sensing area overlap and sensing gaps.

In the ideal case, the optimal conditions for NODC is driven based on the following assumptions: (i) the sensing area of each node is represented as a disk of the same size; (ii) sensor node density is sufficiently high that a sensor can be found at any desirable point; and (iii) the interested sensing space is much

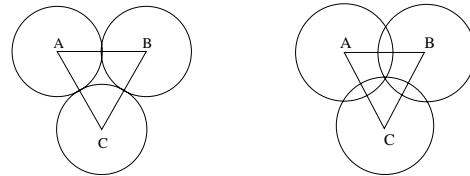


Fig. 1. Comparison of the basic idea of NODC (left) and OGDC (right).

larger than the sensing range of each sensor node, so the boundary effects can be ignored; (iv) the radio range is at least twice of the sensing range, and (v) each node knows its position.

The primary goal of NODC is to minimize the uncovered area (sensing gaps of the working nodes), while avoiding overlaps of the sensing areas of working nodes. A representative case of NODC is shown in Fig. 1. In contrast, the goal of OGDC is to minimize the overlaps of the sensing areas of working nodes (cannot make it zero), while avoiding sensing gaps.

The overall process of NODC is similar to OGDC. Time is divided into rounds. At the beginning of each round, all nodes wake up, set their states to “UNDECIDED,” and carry out the process of selecting the working nodes. The process ends when all the nodes change their states to either “ON” or “OFF.” All nodes remain in the state until the beginning of the next round.

The main difference of NODC from OGDC is in how to set backoff timer of each node when it receives a power-on message.

*Selection of starting node.* At the beginning of each round, every node is in the “UNDECIDED” state. A node volunteers to be a starting node with a certain probability. The volunteer changes its state to “ON”, and broadcasts a power-on message that contains (i) the position of the sender and (ii) the direction  $\alpha$ . This direction is randomly generated from a uniform distribution in  $[0, 2\pi]$ .

*Actions taken when a node receives a power-on message.* When a sensor node receives a power-on message, it checks whether or not the ratio of its sensing area covered by its “ON” neighbors to its overall sensing area is over a predetermined threshold. If so, it sets its state to “OFF”. Otherwise, it takes actions according to the following rules R1 to R4.

*Rule R1:* The message is the first power-on message received and is from a starting node. If the  $\alpha$  in the message is greater than or equals to 0, then the message was originated from a starting node. In this case, the node sets a timer of  $T_{c1}$  seconds.

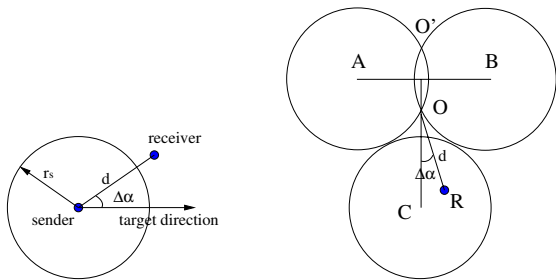


Fig. 2. Description of formulas  $T_{c1}$  (left) and  $T_{c2}$  (right).

When the timer expires, the node sets its state to “ON” and broadcasts a power-on message with  $\alpha$  set to a negative value. If the node hears any other power-on message before its timer expires, it carries out the operations specified in Rule R3.

The value,  $T_{c1}$ , of the backoff timer is set as follows:

$$T_{c1} = t_0(c((2r_s - d)^2 + (d\Delta\alpha)^2) + u) \quad (1)$$

where  $t_0$  is the time it takes to send a power-on message,  $c$  is a constant that determines the backoff scale,  $r_s$  is the sensing range,  $d$  is the distance from the sender to the receiver,  $\Delta\alpha$  is the angle between  $\alpha$  and the direction from the sender to the receiver, as shown in Fig. 2, and  $u$  is a random number drawn from the uniform distribution  $[0, 1]$ .

*Rule R2:* The message is the first power-on message received but is from a non-starting node. The node sets a timer of  $T_e$  seconds, where the value of  $T_e$  is much larger than that of  $T_{c1}$ .

*Rule R3:* The message is the second power-on message received. If the coverage areas of the two senders do not intersect, then if the second sender is further away from the receiver than the first one, the receiver simply ignores the new message but retains the timer set for the first power-on message; otherwise, it cancels the existing  $T_e$  or  $T_{c1}$  timer and sets a timer of  $T_{c1}$  based on the second sender. If the two senders intersect, the receiver cancels the existing timer and sets a timer of  $T_{c2}$  seconds:

$$T_{c2} = t_0(c((\overline{OC} - d)^2 + (d\Delta\alpha)^2) + u) \quad (2)$$

where  $\overline{OC} = \sqrt{(2r_s)^2 - (\frac{1}{2}\overline{AB})^2} - \frac{1}{2}\overline{OO'}$ , and  $t_0$ ,  $c$ ,  $r_s$ , and  $u$  are the same as in Eq. (1). Figure 2 illustrates the basic idea, where  $A$  and  $B$  are two senders and  $R$  is the receiver.

*Rule R4:* More than two power-on messages have been received. This case can be further divided into two subcases.

a) None of the coverage areas of the senders overlaps with each other. The receiver sets the timer to the  $T_{c1}$  value of the closest sender.

b) The coverage areas of the senders overlap. Find the two closest senders that overlap and set the timer to their  $T_{c2}$  value.

In R1 to R4, the timer of the receiver can be set based on up to two “ON” neighbors. For comparison purpose, we also propose a simpler algorithm that only set the timer based on up to one “ON” neighbor. To distinguish these two algorithms, we call the one using Rules R1-R4 *NODC-2*, and the simpler one using the following two Rules, R1’ and R2’, *NODC-1*.

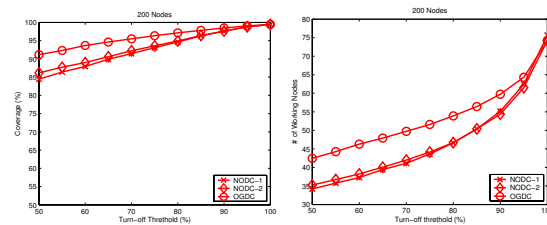


Fig. 3. Performance comparison of NODC-2, NODC-1, and OGDC.

*Rule R1’:* The message is the first power-on message received. The receiver sets its timer to the value of  $T_{c1}$ .

*Rule R2’:* More than one power-on messages have been received. If the new sender is the closest to the receiver among the previous senders, then the receiver sets its timer to the value of  $T_{c1}$  of the new sender. Otherwise, the receiver keeps its timer unchanged.

### III. EXPERIMENTAL RESULTS

The parameters of the test problems are as follows:

- The region of interest is a 50m by 50m square.
- The sensing range of each node  $r_s$  is 5m.
- The time it takes to send a power-on message  $t_0$  is 7ms.
- The constant  $c$  used in Eqs. (1) and (2) is  $10/r_s^2$ .

The other parameters for OGDC are set to the same values as in [2].

In the simulation, we compare the results of the three algorithms, NODC-2, NODC-1, and OGDC, for different power-off threshold of the nodes. Each data point is the averages of 100 random runs. Fig. 3 shows the result of coverage percentage versus the turn-off threshold percentage for sensor networks with 200 sensor nodes. OGDC has the best coverage, but uses the most working nodes. NODC-2 and NODC-1 use much fewer working nodes than OGDC, and the difference increases as the turn-off threshold decreases. The saving of working nodes by NODC-2 and NODC-1 against OGDC goes up to 25%. At the same time, the reduction of the overall coverage of NODC-2 and NODC-1 against OGDC is less than 5%. The performance difference between NODC-2 and NODC-1 is small.

### REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Czirci, “Wireless sensor networks: A survey,” in *Computer Networks*, March 2002, pp. 38(4):393–422.
- [2] Honghai Zhang and Jennifer Hou, “Maintaining coverage and connectivity in large sensor networks,” in *International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless and Peer-to-Peer Networks*, Feb 2004.
- [3] Ya Xu, Solomon Bien, Yutaka Mori, John Heidemann, and Deborah Estrin, “Topology control protocols to conserve energy in wireless ad hoc networks,” Tech. Rep. 6, University of California, Los Angeles, Center for Embedded Networked Computing, January 2003, submitted for publication.
- [4] Alberto Cerpa and Deborah Estrin, “Ascent: Adaptive self-configuring sensor networks topologies,” in *Proc. IEEE INFOCOM 2002*, New York, NY, USA, June 2002.
- [5] Fan Ye, Gary Zhong, Songwu Lu, and Lixia Zhang, “Peas: A robust energy conserving protocol for long-lived sensor networks,” in *Proceedings of the 10th IEEE International Conference on Network Protocols*, 2002, pp. 200–201, IEEE Computer Society.