Energy-Efficient Composite Event Detection in Wireless Sensor Networks

Mirela Marta, Yinying Yang, and Mihaela Cardei*

Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431, U.S.A.
`{mmarta,yyang4}@fau.edu, mihaela@cse.fau.edu`

Abstract. Wireless sensor networks are deployed to monitor and control the physical environment. Sensors can be equipped with one or more sensing components, such as temperature, light, humidity, etc. An atomic event can be detected using one sensing component. A composite event is the combination of several atomic events. We consider a wireless sensor network densely deployed in a monitored area for reliable detection of a predefined composite event. In this paper, we study the energy-efficient k-watching composite event detection problem, concerned with designing a sensor scheduling mechanism that increases network lifetime, when the set of active sensor nodes are connected and collectively k-watch the composite event at all times. We propose a localized connected dominating set based approach and analyze its performance by simulations.

Keywords: Wireless sensor networks, composite event detection, energy efficiency, sensor scheduling, reliability.

1 Introduction

Sensors are used to monitor and control the physical environment. A Wireless Sensor Network (WSN) is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it [1], [3]. Sensor nodes measure various parameters of the environment and transmit data collected to one or more sinks. Once a sink receives sensed data, it processes and forwards it to the users.

A WSN can detect single (or *atomic*) events or *composite* events [9]. Considering the sensors manufactured by Crossbow Technology, Inc. [10] as an example, a sensor equipped with MTS400 multi sensor board can sense temperature, humidity, barometric pressure, and ambient light. Thus it can detect multiple atomic events.

Let us consider a single sensing component, for example the temperature. If the sensed temperature value rises above a predefined threshold, then we say that an *atomic* event has occurred. A *composite* event is the combination of several *atomic* events. For example, the composite event fire may be defined as the combination of events temperature and light. The *composite* event fire occurs only when both the temperature and the light rise above some predefined thresholds.

---

[*] This work was supported in part by NSF grant CCF 0545488.

B. Liu et al. (Eds.): WASA 2009, LNCS 5682, pp. 94–103, 2009.

In this paper, we consider a heterogeneous WSN deployed to reliably detect a predefined composite event. For a reliable detection, each atomic event part of the composite event has to be k-watched, that means k sensor nodes equipped with the corresponding sensing component have to be active.

A large number of sensors can be distributed in mass by scattering them from airplanes, rockets, or missiles [1]. Generally, more sensors are deployed than the minimum required to perform the proposed task. This compensates for the lack of exact positioning and improves fault tolerance. An important issue in WSNs is power scarcity, driven in part by battery size and weight limitations. One power saving technique that highly improves network lifetime is to schedule sensor nodes to alternate between active and sleep mode. The active nodes are then in charge with sensing and data gathering tasks.

We propose the energy-efficient k-watching composite event detection (k-ECED) problem. Given the initial deployment of a power-constrained heterogeneous WSNs containing sensors with different sensing capabilities, our objective is to design a sensor scheduling mechanism that maximizes network lifetime. The active sensor set must k-watch the composite event and must be connected. Providing a connected topology is an important property needed for event and data reporting. We propose a localized connected dominating set based approach for designing the sensor scheduling mechanism and we analyze its performance through simulations.

2 Related Work

Sensor nodes have limited power supply and therefore they can work a limited amount of time before they deplete their energy resources and become unfunctional. Many routing algorithms introduced in literature try to minimize sensor energy consumption using different approaches. One such method is sensor scheduling using Connected Dominated Set (CDS). Here, the nodes are either in the dominated set or at most one hop away from the CDS. In this way, all the nodes can communicate with the nodes in the CDS and can transmit their readings to the sink. The problem of routing in ad hoc networks using the CDS approach is described in [11]. The authors propose a three step algorithm for data delivery starting at the source gateways, forwarding the messages through the induced graph, and ending with the destination gateways.

Adjusting the transmission range represents another approach for saving energy in WSNs. Messages sent to a shorter distance will consume less energy than messages transmitted over a larger distance, but they might require multiple retransmissions.

Periodic data reporting and event detection are two main applications of WSNs. Periodic data reporting involves sensor nodes sending their sensed data to a sink periodically. The latter approach implies sending a message each time an event occurs. In [5], authors introduce a framework that deals with event detection in a distributed way by using nodes collaboration. Their goal is to have a distributed system that is resilient to nodes failures and low energy of nodes. The two protocols introduced here are simple event detection and composite event detection. Both build a tree using a communication model based on the Publish-Subscriber paradigm. They work in two phases: initialization phase and collection phase. In the first phase, the application advertises events of interest in the network, together with the region in the network (in terms of start and

Table 1. Notations

M	The number of atomic events which form the composite event
N	The number of sensor nodes
R_c	Sensor communication range
A	Deployment area
x_j	Sensing component which detects atomic event j
k	Fault tolerance level
E	Initial energy of each sensor
h	The number of hops in the local topology

end coordinates) where the event/events are desired to be detected. Based on this, an *Event-Based-Tree* (EBT) is constructed and is used to propagate the information from the source to the destination. In the collection phase, results are collected after an event happened and relied to the sink. In order to save energy, the results are collected using aggregation.

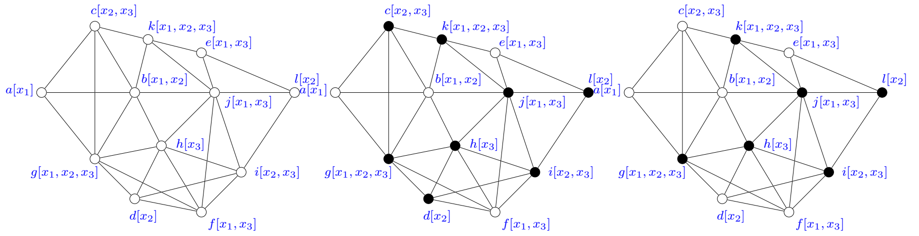
In [9], the authors introduce the *k-watching Event Detection Problem*. The goal is to have an area being k-watched, which means that there must be at least k sensor nodes in the respective area that have together at least k sensing units for each atomic event of interest. A topology and routing supported algorithm is introduced to compute the maximum number of detection sets such that each detection set ensures the k-watching property. The detection sets (or data collection trees) are constructed using the Breadth First Search algorithm starting from a gateway, which can be any sensor node with richer energy resources. One drawback of the proposed approach is the global knowledge required by the algorithm constructing the detection sets. The sets are computed by the gateway node and the decision if an event, simple or composite, occurs is also made by the gateway node. In this paper we proposed a localized approach for constructing detection sets, which is more scalable and thus more appropriate for large scale WSNs.

3 Problem Definition and Network Model

3.1 Problem Definition

Sensors can have single or multiple sensing components, such as the temperature, humidity, barometric pressure, and ambient light. In [9], authors introduce the definitions of an *atomic event* and a *composite event*. When we consider a single sensing component, for example, the temperature, if the temperature rises above some predefined threshold, an *atomic event* is detected. A *composite event* is the combination of several *atomic events*.

For example, consider a fire-detection application using sensors to measure various parameters of the environment. A *composite event* fire might be defined as the combination of the *atomic events* temperature $> th_1$, light $> th_2$, and smoke $> th_3$, where "th" denotes a threshold for the corresponding attribute. That is $fire = (temperature > th_1) \wedge (light > th_2) \wedge (smoke > th_3)$. It is more accurate to report the fire when all these atomic events occur, instead of the case when only one attribute is above the threshold.



(a) (b) (c)

Fig. 1. CDS-LocalRule example (a) WSN topology, (b) CDS-LocalRule for $h = 1$, and (c) CDS-LocalRule for $h = 2$

The main notations used in this section are introduced in Table 1. We denote by M the total number of sensing components, $x_1, x_2, ..., x_M$. Sensors are equipped with single or multiple sensing components. The number and types of sensing components may be different among the sensors in the network. For each sensing component type, a sensor node can be equipped with at most one such sensing component. All of a sensor's sensing components turn on or off simultaneously.

There are several reasons why sensor nodes could have different sets of sensing components [5]:

- The deployed sensors might be manufactured with different sensing capabilities.
- Some sensing capabilities might fail over time.
- Some sensor nodes might have purposefully stopped some sensing components due to energy constraints.
- A sensor node might be unable to use some of its sensor data due to the lack of memory for storing data.

We consider a WSN deployed to detect a predetermined *composite event*, which is a combination of M atomic events $x_1 > th_1, x_2 > th_2, ..., x_M > th_M$. Sensor nodes are equipped with one or more sensing components and they cooperate to detect the composite event. Each sensor node is equipped with a subset of the sensing components set $\{x_1, x_2, ..., x_M\}$. Only when information about the occurrence of all the M atomic events is collected, the occurrence of the composite event is concluded.

Fig. 1a shows an example of a network with $N = 12$ sensor nodes equipped with one or more sensing components. There are three sensing components ($M = 3$) x_1, x_2, x_3 whose measurements have to be used to detect the composite event.

To achieve a reliable surveillance, an event must be observed by more than one sensor. We adopt the k-watched atomic/composite event definitions from [9]:

Definition 1. (k-watched atomic event) *An atomic event is k-watched by a set of sensors if at any time this event occurs at any point within the interested area, at least k sensors in the network can detect this occurrence.*

Definition 2. (k-watched composite event) *A composite event is k-watched by a set of sensors if every atomic event part of the composite event is k-watched by the set of sensors.*

In this paper, the objective is that the composite event is k-watched by the sensors in the area. A WSN is densely deployed, so more sensors are deployed than the number required to provide k-watching of a composite event. In order to prolong network lifetime, one solution is to schedule sensor activity to alternate between sleep and active mode, with the requirement that the active sensors ensure k-watching of the composite event.

Another important step is collecting of sensed data from the active nodes, to determine if the composite event has occurred. For this, the set of active nodes must provide a connected topology. The problem definition is presented next.

Definition 3. (Energy-efficient k-watching Composite Event Detection) *k-ECED Given a set of N sensors with different sensing capabilities, a monitored area A, a composite event which is a combination of M atomic events involving sensing components x_1, x_2, ..., x_M, and the energy constraint of each sensor E_{init}, design a sensor scheduling mechanism such that:*

1. *the composite event is k-watched in the area A*
2. *the set of active sensor nodes is connected*
3. *network lifetime is maximized.*

3.2 Network Model

In this paper we consider a heterogeneous WSN consisting of sensor nodes with different sensing capabilities and a sink, deployed to reliably detect when a predefined composite event takes place. The main objective of the *k-ECED* problem is to prolong network lifetime while ensuring that the composite event under consideration is k-watched and that the set of active nodes are connected.

In order to prolong network lifetime, we propose to use a sensor scheduling mechanism. Network activity is organized in rounds. Each round consists of two phases: *initialization phase* and *event detection phase*.

In the initialization phase, sensor nodes decide if they will be active or if they go to sleep during the next round. The active sensor set must provide a connected topology and must ensure k-watching property across the deployment area.

In this paper, we are not concerned with a specific location for the sink, but we assume that the sink is located within communication range of at least one active node. The sink could be static or it can move in the deployment area. If the sink is mobile, then flooding the alerts (when a composite event is detected) in the network will reach the sink. If the sink is static or moving at a slow speed, the sink could form a data collection tree rooted at the sink, as follows. At the beginning of each round, after the sensor initialization phase, the sink is flooding a query message in the network. Sensors will forward the first copy of the message received and set-up reverse links. When a composite event is detected, the alert is sent to the sink along the reverse links.

The decision of whether to use data collection trees depends on the application. If the composite event detection is expected to be very rare (e.g. forest fire detection), then sending an alert using flooding could be more efficient. If more event detections and reporting are expected in each round, then using a data collection tree could be more energy efficient.

We consider that all rounds take the same time. Network lifetime is measured as the number of rounds until no more detection sets can be formed due to the energy constraints. We consider a similar energy model as that presented in LEACH [4]. The energy used to transmit a l-bit message over a distance d is: $E_{Tx}(l,d) = E_{elec} * l + \varepsilon_{amp} * l * d^2$, and the energy consumed to receive a l-bit message is: $E_{Rx}(l) = E_{elec} * l$, where E_{elec} = 50 nJ/bit and ε_{amp} = 100 pJ/bit/m^2. We assume that the packet size of the $Hello$ messages is much smaller than the size of data messages.

Network topology is modeled as an undirected graph $G = (V, E)$, where the set of vertices V has one vertex for each sensor node. An edge is added between two vertices u and v if the Euclidean distance between nodes is less than or equal to the communication range R_c. We define the h-hop neighborhood of a node u as $N_h(u) = \{v | distance(u, v) \leq h\ hops\}$.

A subset of vertices C form a Connected Dominating Set (CDS) if the subgraph induced by C is connected and for any node $v \in V$, either $v \in C$ or v has a neighbor in C.

4 CDS-Approach to the k-ECED Problem

Our solution addresses the *initialization phase*, the decision mechanism used by a node with sufficient energy resources to decide whether it will be an active or sleeping during the next round.

The decision mechanism that we propose is a local algorithm, where each node relies only on local information from its h-hop neighborhood, where h is a given parameter. A local solution is more scalable, but may obtain sub-optimal solutions compared to a global solution.

The CDS-based local algorithm decides the sensor scheduling at the beginning of each round. Let us assume that each node has the h-hop neighborhood information which can be obtained by having nodes exchange Hello messages. For constructing h-hop neighborhood, each node s broadcasts h Hello messages. First Hello message contains only the node s's ID. Then the ith Hello message ($i = 2, ..., h$) contains s's $(i-1)$-hop neighborhood. We also consider that each node u has a priority $p(u)$ which is totally ordered within the network. The priority is defined as a 2-tuple $p(u) = (E(u), ID(u))$, where $E(u)$ is the node u's residual energy and ID(u) is the node u's ID. A node with higher residual energy has a higher priority. If two nodes have the same residual energy, then the node with higher ID has higher priority.

At the beginning of the initialization phase, all the nodes are active. For a node u, we define $N'_h(u)$ to be the set of nodes within the h-hop neighborhood which have priority higher than u's priority: $N'_h(u) = \{v \in N_h(u) | p(v) > p(u)\}$. Each node decides whether it will go to sleep during the current round based on the following rule:

CDS-LocalRule. *The default status of a sensor node is active. A sensor u is in sleep mode if the following two conditions hold:*

1. *(k-watching property) the set of nodes $N'_h(u)$ provides the k-watching of each of u's sensing components.*
2. *(Connectivity property) any two of u's neighbors in $N_1(u)$, w and v, are connected by a path with all intermediate nodes in $N'_h(u)$.*

The intuition behind this rule is that a sensor u can go to sleep if the nodes with higher priority in its h-hop neighborhood can provide the k-watching property and the connectivity property on behalf of u. Note that a node with higher priority can also go to sleep if the rule holds true in its h-hop neighborhood. To avoid inconsistencies, a mechanism that assigns global priorities to the nodes is used. If a node u has only one neighbor, then the connectivity requirement is fulfilled and u goes to sleep if the k-watching property holds.

h represents the number of hops in the local topology and is a given parameter which usually takes small values, e. g. $h \leq 3$. There is a trade-off in choosing h. When h is larger, more Hello messages are exchanged, resulting in a higher possibility of collisions and a higher overhead in collecting local information. But more nodes are expected to go to sleep since the local neighborhood $N'_h(u)$ has a larger cardinality, see Fig. 3. Let S be the set of sleeping sensors, $V' = V - S$, and let G' be the subgraph induced by V'.

Theorem. *Assume that the WSN $G(V, E)$ provides the composite event k-watching property and forms a connected topology. Let S be the set of sleeping sensor nodes after applying the CDS-LocalRule. Then the following properties hold:*

- *Each sensor in S has a neighbor in V'*
- *Sensors in V' are connected*
- *Sensors in V' provide the k-watching property.*

Proof: The first two properties ensure that the set of active sensor nodes V' forms a CDS. Let us take any sensor node a in the network, $a \in V$. We will show that any other node in the network is connected to a through a path with nodes in V'.

Let X be the set of nodes which is not connected to a by a path of active nodes and let Y be the set of sleeping neighbors of the set X which are connected to a. The set $Y \neq 0$ since the nodes in X are connected to a in the original graph G. Let y be the node with the highest priority in Y. Then the node y has two neighbors, one neighbor $b \in X$ and one neighbor $c \in V - X - Y$. Since node y is sleeping, then according to the *CDS-LocalRule*, any two neighbors b and c must be connected through a path with intermediate nodes of higher priority. Such a path must contain at least one node in Y, let us denote it y'. It follows that $p(y') > p(y)$ which contradicts the assumption that y has the highest priority in Y.

Let us show now that nodes V' provide the k-watching property. Assume by contradiction that V' does not provide the k-watching property for a sensing component x_i. Let v be the sleeping node in $V - V'$ with the highest priority which is equipped with sensing component x_i. Then, according to the CDS-LocalRule k-watching property, v went to sleep if all its components are k-watched by nodes in $N'_h(u)$. It follows that the nodes in $N'_h(u)$ have k components x_i. Also these components didn't go to sleep since the nodes in $N'_h(u)$ have higher priority than u. This contradicts our assumption and as a result the nodes in V' provide the k-watching property. ■

Let us consider an example of a WSN with 12 sensor nodes, deployed as illustrated in Fig. 1a. The composite event consists of three sensing components $\{x_1, x_2, x_3\}$, and each sensor is equipped with one or more sensing components. The fault tolerance level is $k = 2$. We consider the nodes priority in alphabetical order, that is $p(a) < p(b) < ... < p(l)$. Fig. 1b shows the active nodes after applying *CDS-LocalRule* for $h = 1$. The blackened nodes are the active nodes.

Let us consider node b for example. $N_1'(b) = \{c, k, j, h, g\}$ and this set provides 2-watching for x_1, x_2 and x_3. Also, taken any two neighbors (e.g. a and h) they are connected through a path with all intermediate nodes in $N_1'(b)$. Thus, node b goes to sleep. For node c, $N_1'(c) = \{g, k\}$. The 2-watching property holds, but the connectivity property does not. The neighbors g and k are not connected through a path with intermediate nodes in $N_1'(c)$. Thus node c does not go to sleep.

Fig. 1c shows the active nodes after applying *CDS-LocalRule* for $h = 2$. As h increases, more sensor nodes are expected to go to sleep. Let us consider node c with $N_2'(c) = \{g, k, h, d, f, j, e\}$. In this case both the 2-watching property and the connectivity property hold, thus node c goes to sleep.

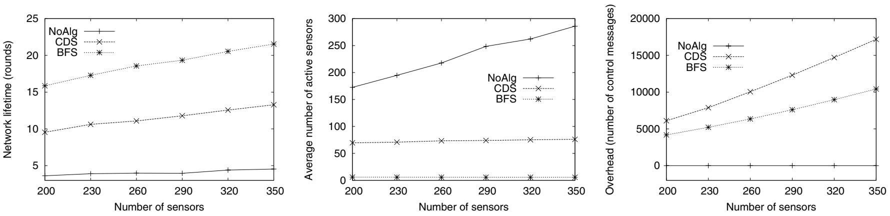
5 Simulation

5.1 Simulation Environment and Settings

Metrics in the simulations include the network lifetime, the average number of active sensors, and the overhead. In each round, we choose one set of active sensors and only active sensors report data to the sink. The network lifetime is computed as the number of rounds where both the k-watching and connectivity requirements are met. The average number of active sensors is computed as $\frac{numActiveSensor}{numRound}$, where $numActiveSensor$ is the total number of active sensors during network lifetime and $numRound$ is the network lifetime in terms of the number of rounds. The overhead is the number of Hello messages sent and received for choosing the set of active sensors.

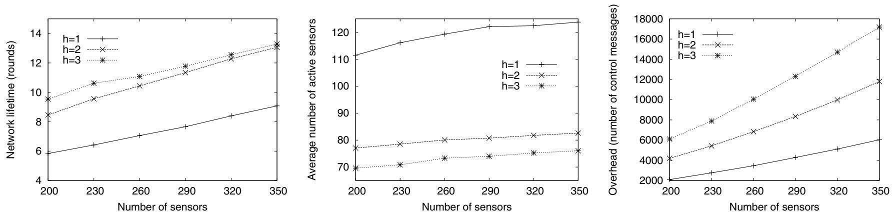
In the simulation, the monitored area is 100×100 units. The communication range of sensors is 15 units. We consider the composite event is a combination of three sensing components $M = 3$ and $k = 3$. The energy consumed for sending and receiving a data message is 60 units and 40 units respectively. The energy consumed for sending and receiving a Hello message is 3 and 2 units respectively. The sink is located at the center of the monitored area.

In each round during the network lifetime, active sensors form a data delivery tree initiated by the sink and each sensor generates 10 data messages. The data delivery tree is formed using controlled flooding. The sink broadcasts a message containing the number of hops, which is forwarded by active nodes which also keep a reference to the parent from which the message was received. We assume that each active sensor applies a data aggregation algorithm. The packet size that an active sensor reports is computed as $\lceil \alpha \cdot (msgReceived + 1) \rceil$, where α is the aggregation factor and we assume $\alpha = 0.5$, and $msgReceived$ is the total packet size it receives from its children in the data delivery tree. We conduct the simulations on a custom discrete event simulator, which generates a random initial sensor deployment. All the tests are repeated 50 times and the results are averaged.



(a) (b) (c)

Fig. 2. Comparison among 3 cases. (a) Network lifetime. (b) The average number of active sensors. (c) Overhead.



(a) (b) (c)

Fig. 3. Comparison among different h values for CDS algorithm. (a) Network lifetime. (b) The average number of active sensors. (c) Overhead.

5.2 Simulation Results

Fig. 2 compares three cases. $NoAlg$ is the case when no scheduling algorithm is applied and no sensor goes to sleep. CDS is our localized CDS-LocalRule approach. BFS is the centralized algorithm proposed in [9]. In Fig. 2b, BFS has the smallest average number of active sensors, while $NoAlg$ has the largest. Consequently, BFS has the longest network lifetime and $NoAlg$ has the shortest, see Fig. 2a. The overhead for choosing the set of active sensors are compared in Fig. 2c. Since BFS is centralized and all computations are done by the sink, each sensor first needs to report to the sink its location and other information and after the sink decides the set of active sensors, it sends back the scheduling plan to the sensors. In the CDS approach, sensors exchange information for h-hop neighborhood information. In $NoAlg$, no overhead is involved.

Fig. 3 compares the performance of the CDS algorithm for different values of h. When h is 3, the smallest number of sensors are chosen to be active and therefore, it has the longest network lifetime. When $h = 1$, it has the largest number of active sensors and the shortest network lifetime. That is because when checking the connectivity condition, when h is larger, the sensor has more neighborhood information and has more chances to find a path connecting pairs of two neighbors, thus the sensor has higher probability to go to sleep. As a trade-off, Fig. 3c shows that compared with the cases $h = 2$ and $h = 1$, the case $h = 3$ involves higher overhead to get more neighbor information.

6 Conclusions

In this paper, we focus on the energy efficient k-watching composite event detection problem. Given a WSN deployed for watching a composite event, our goal is to design a sensor scheduling mechanism such that the monitored area is k-watched, the connectivity condition is met and the lifetime is maximized. One localized CDS-based algorithm is proposed. Simulation results show that our method has low overhead and is effective in prolonging network lifetime.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38(4), 393–422 (2002)
2. Cheng, X., Thaeler, A., Xue, G., Chen, D.: TPS: A time-based positioning scheme for outdoor wireless sensor networks. In: IEEE INFOCOM (2004)
3. Culler, D., Estrin, D., Srivastava, M.: Overview of sensor networks. IEEE Computer 37(8), 41–49 (2004)
4. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Hawaii Intlernational Conference on System Sciences (HICSS) (2000)
5. Kumar, A.V.U.P., Reddy, A.M., Janakiram, V.D.: Distributed collaboration for event detection in wireless sensor networks. In: MPAC 2005 (2005)
6. Marta, M., Cardei, M.: Using sink mobility to increase wireless sensor networks lifetime. In: IEEE WoWMoM 2008 (2008)
7. Romer, K., Mattern, F.: Event based systems for detecting real-world states with sensor networks: a critical analysis. In: DEST Workshop on Signal Processing in Sensor Networks at ISSNIP, December 2004, pp. 389–395 (2004)
8. Stojmenovic, I., Lin, X.: Power-aware routing in ad hoc wireless networks. IEEE Transaction on Parallel and Distributed Systems (2001)
9. Vu, C.T., Beyah, R.A., Li, Y.: Composite event detection in wireless sensor networks. In: IEEE International Performance, Computing, and Communications Conference (2007)
10. `http://www.xbow.com/Home/HomePage.aspx`
11. Wu, J., Li, H.: An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. In: IEEE Transactions on Parallel and Distributed Systems, October 2004, pp. 908–920 (2004)
12. Xu, Y., Heidemann, J., Estring, D.: Geography-informed energy conservation for ad hoc routing. In: ACM/IEEE International Conference on Mobile Computing and Networking (July 2001)