Transitive approach for topology control in Wireless Sensor Networks

Karim Bessaoud  
University of Versailles SQY  
45 avenue des Etats-Unis  
Versailles, France  
karim.bessaoud@prism.uvsq.fr

Alain Bui  
University of Versailles SQY  
45 avenue des Etats-Unis  
Versailles, France  
alain.bui@prism.uvsq.fr

Toshimitsu Masuzawa  
Osaka University  
1-3 MACHIKANEYAMA  
TOYONAKA, 5608531, JAPAN  
masuzawa@ist.osaka-u.ac.jp

Laurence Pilard  
University of Versailles SQY  
45 avenue des Etats-Unis  
Versailles, France  
laurence.pilard@prism.uvsq.fr

Abstract—The energy is a critical resource in Wireless Sensor Networks that impacts on networks lifetime. In this paper, we propose a distributed self-stabilizing algorithm of topology control to preserve energy in case of communications by broadcast. The topology control is achieved by the reduction of the transmission power of the nodes in the network. The self-stabilizing property is a very desirable property in Wireless Sensor Networks that guarantees to reach a correct behavior in a finite number of steps, regardless of its initial state. Our solution is validated by extensive simulations. The obtained results show the efficiency of our solution in case of communication by broadcast.

Keywords—wireless sensor networks; self-stabilization; topology control; simulation; distributed algorithm

I. INTRODUCTION

Wireless Sensor Network (WSN) consists of a large number of sensors deployed over an area that must be covered. These sensors collect data, share it with other sensors or send it to a sink for a specific treatment. Indeed, the sensors are able to monitor physical or environmental conditions, such as temperature or movement, that can be used for different tasks depending on the application. WSN is an ad hoc network where sensors communicate using multi-hop wireless communication. Sensors can be randomly deployed since geographical and technical conditions do not always allow customized placement. This constraint requires self-* (self-organization, self-stabilization, self-healing, ...) properties of the system.

Sensors are composed by four main components: a sensing component, a component for processing and storage, a radio and a battery. Some components can be added depending on the applications and the protocols. For example, GPS can be added for applications that require the knowledge of the exact position of the sensors. Sensors have a limited energy due to the use of a battery, the network lifetime depends on this energy. So, unlike traditional networks where everything is done to improve the performance (delay, throughput ...), in WSN the effort is mainly focused on the conservation of the energy resource.

Energy is used in WSN for sensing, processing and for the communication, i.e. listening to the media, sending and receiving messages. The communication consumes the most energy and the transmission range has a great influence on the energy of transmission. Indeed, the energy consumed for a message transmission increases quadratically with the transmission range [1]. In addition, the higher the transmission range is, the denser the network, and so the higher the risk of collision is due to concurrent transmission. Note that collisions cause multiple sending and receiving and thus is energy consuming. The idle listening is specific to wireless networks, this is the state where a sensor waits for a message. Since a node does not know when a message will arrive, it should keep its radio in receive mode whenever it is not transmitting. This mechanism can be very harmful in terms of energy due to the overhearing. The overhearing happens when a sensor receives messages that does not concern it and so will consume an unnecessarily reception energy. Indeed, when a sensor sends a message, all sensors that are in its transmission range receive this message, even if they are not intended. Thus, the higher the transmission range, the higher the number of overhearing.

As we just saw, the transmission range has a significant impact on the energy consumption. In this paper, we present an algorithm that reduces independently the transmission range of each node in a WSN in order to reduce the energy consumption and then to increase the network lifetime.

The reduction of the transmission range has to consider the communication category. The diversity of applications in WSN implies that no generic category of communication is adapted to every type of applications. In the following, we briefly present the main communication category in WSN.

- sensors-to-sink: Commonly, sensor networks applications use a sink that handles the collected informations. In this case, sensors must root the messages to the sink. In WSN, sensors use multi-hop transmission to communicate with the sink. They must create paths to transmit messages to the sink. In that case, the routing is quite simple, since each sensor only needs to know its successor on the path to the sink.
- sensor-to-sensor: When two sensors need to communicate, they are either neighbors and can communicate by a single transmission, or distant, and the messages has to be routed. Here, the routing is more complicated than in the previous category, since sensors must handle
routing tables to forward the messages.

- broadcast: Is another way to route messages when a sensor needs to transmit a message to the whole network. To perform a broadcast, a source sensor sends a message to its neighbors, then when a sensor receives the message for the first time, it forwards it to its neighbors.

In this paper, we propose an algorithm of topology control broadcast communication that reduces the transmission range of each sensor. Our algorithm is self-stabilizing that allow our system to return to a regular activity after any transient fault (variable corruption, node failure, etc.).

The remainder of this paper is structured as follows. Section II presents related work on algorithms for energy saving in WSN. Our system model is presented in section III. Section IV gives a detailed description of our algorithm. A performance evaluation is analyzed in section V followed by a conclusion and some perspectives in section VI.

II. RELATED WORK

Energy is a critical resource for WSN since the network’s lifetime depends on the energy. The issue of reducing the energy consumption in WSN has been extensively discussed in the literature and several solutions have been proposed.

In [2], [3], [4], [5] solutions, the main objective is to minimize the energy of end-to-end communications using energy-efficient routing protocols. The energy-efficient routing has to reduce the number of messages transmission and reception. However, an energy-efficient routing protocol may consume itself a lot of energy. Hence a good compromise should be found. When the application allows it, a low energy solution for reducing messages transmission is the data aggregation. Indeed, this solution enables to limit the number of transmitted messages for a node at the price of loosing some information.

The solutions proposed in [6], [7], [8] are based on node’s activity scheduling by alternating node’s states between sleep and active state. Node in the sleep state cannot send or receive messages, reducing then the node’s energy consumption. The difficulty of these solutions is to minimize energy consumption while ensuring network’s connectivity and application functionalities. At each time, sensors in the active state should cover the whole network, and form a connected network.

In [9], [10], [11], [12], [13], [14], sensors reduce their transmission range while maintaining network connectivity. The solution allows to reduce energy consumption due to transmission power, overhearing and collision by decreasing the nodes transmission range. However, if transmission ranges are reduced too much, the network can be partitioned. In the following, we describe in more details two solutions close to ours and that we use to compare with our solution in the evaluation section.

In [9], an adjustment of the power transmission of nodes in an ad hoc network is performed to reduce the energy consumption of the nodes while ensuring the connectivity or the bi-connectivity of the network. The authors proposed two centralized algorithms CONNECT and BICONN-AUGMENT conceived for static networks. For mobile ad hoc networks, the authors designed two distributed heuristics LINT and LILT.

In this paper, we focus in CONNECT that minimizes the maximal power used by the nodes of the network while maintaining the connectivity of the network. In the initial state, all the nodes form a connected component. Then, the algorithm merges the closest connected components in Euclidean distance to form a single connected component.

In [12], an energy-efficient self-stabilizing topology control protocol for WSN is proposed. This algorithm is based on a low cost backbone construction that helps to maintain network connectivity while saving maximum energy. The transmission power of each node is independently reduced while preserving the backbone. The objective of [12] is to reduce the energy consumed in the network to route end-to-end messages for sensor-to-sensor communications.

III. SYSTEM MODEL

The network is formed by a set of $n$ wireless nodes. Each node has a unique identifier and uses an omni-directional antenna with a transmission range $R_u$ fixed by its transmission power $p_u$. Each node $u$ has a maximum transmission range $MaxR_u$ (thus $R_u \leq MaxR_u$). When a node $u$ sends a message, all nodes that are in the disk centered on $u$ with range $R_u$ receive the message. The network is represented by a disk graph with unidirectional link (DGU) $G = (V, E)$, where $V$ represents the set of nodes and $E$ the set of wireless links. $E$ is the set of directed edges such that $(u, v) \in E$ iff $u \in V, v \in V$ and the Euclidean distance between $u$ and $v$ is lower than $MaxR_u$ (so the node $v$ can receive messages from $u$).

Our algorithm computes for each node the reduced transmission range $r_u$, that has to be used for all communication but our algorithm communications. We then obtain a sub-graph of $G, G' = (V', E')$ where $V' = V$ and $E' \subseteq E$ such as $(u, v) \in E'$ iff $(u, v) \in E$ and the Euclidean distance between $u$ and $v$ is lower than $r_u$ fixed by our algorithm.

We assume each node $u$ can compute its distance from each of its neighbors $v$ by using the signal power of the messages sent by $v$ and the transmission power of $v$. The function $dist(u, v)$ returns the distance between $u$ and $v$.

We use a message passing communication model. Each node $u$ waits for some HELLO messages from its neighbors in order to compute its neighbors list. Thus $u$ has to distinguish between a dead node $v$ that will never send a message again, and an alive node $v$ that sent a message to $u$ that is very long to be delivered. According to [17], this distinction is impossible in a purely asynchronous message.
passing system where there is no bound on the message delivery time. We assume then a pseudo-synchronous message passing system, i.e., there is an unknown bound on the message delivery time. Thus, our algorithm relies on timers.

A node maintains a set of variables that makes up the local state of the node. Each node executes a set of guarded rules. Each rule has the form \(<Guard> \rightarrow <Action>\), where \(<Guard>\) is a boolean condition over the node’s local state or a message reception, and \(<Action>\) is a set of statements assigning new values to the variables of the node. The set of the local states of all nodes in the network union the set of all in transit messages is called a configuration.

A node is enabled if one of its guards is true. Nevertheless, the execution of this guarded rule depends on a daemon that is a predicate over the execution. Our algorithm works under the fair asynchronous daemon.

Definition 1 (fair asynchronous daemon): Is defined as follow: (i) to move from one configuration to the next one, the daemon selects a non-empty set of enabled nodes to execute one guarded rule, and (ii) any node that is continuously enabled will eventually be selected.

Our solution is self-stabilizing [15, that is a very valuable property in WSN since it ensures the achievement of a correct behavior in a finite number of steps, regardless to the initial state. This allows our solution to eventually recover to a correct behavior after any transient fault, without any external intervention.

Definition 2 (Self-stabilization): Let \( \Gamma \) be a set of all configurations. A system \( S \) is self-stabilizing with respect to \( \Lambda \), with \( \Lambda \subseteq \Gamma \) if and only if it satisfies the following two conditions: (convergence) every execution of \( S \) contains at least a configuration in \( \Lambda \), and (closure) for any configuration \( \lambda \in \Lambda \), any configuration \( \gamma \) that follows \( \lambda \) is also in \( \Lambda \).

IV. ALGORITHM DESCRIPTION

The algorithm is composed by two phases. The first phase for the neighborhood discovery and the second phase for the computation of the relevant transmission range. Our algorithm uses only HELLO messages sent by each node to give one-hop neighborhood information. This message is sent to all neighbors and is never forwarded.

During the first phase, each node computes its potential neighbors (PN variable). A potential neighbor for a node \( u \) is a node \( v \) such that there is a bidirectional link between \( u \) and \( v \) when they both use their maximum transmission range. We recall that in this paper, we assume the wireless network is modeled using a DGU graph and so, the links are not necessarily bidirectional. So all nodes will first detect all the directional links (RF variable for Received From), from the received messages and then compute the PN. During this first phase, each node also computes the potential neighbors of all its own potential neighbors (TPN variable for Transitive Potential Neighbors). Each time a node observes some changes in RF, PN or TPN, it launches the second phase.

During the second phase, each node computes its reduced transmission range using compute-reduced-transmission-range() procedure. This procedure call two other procedures, the first one to compute the potential forwarders (PF variable) and the second one to compute the reduced neighbors (RN variable). A potential forwarder is a potential neighbor that can forward a message to another potential neighbor computed using TPN and PN variables. The idea is to find the transitive neighbors such as a node \( u \) that can communicate with \( v \) and \( w \) with \( u \) with \( w \) to try to delete this transitivity by deleting the link with the highest distance. Reduced neighbors is a subset of potential neighbors. The idea is to delete from the potential neighbor the most distant ones. Such a neighbor \( v \) can be deleted (i) if there exists another neighbor, closer, that can forward messages to \( v \) and (ii) if \( v \) is not himself the last forwarder for another node even further. The transmission range is then reduced to \( r_u \), the minimum range allowing the node to communicate with all its reduced neighbors.

A. Variables for node \( u \):

In the following, we present the variables of a node \( u \) used in our algorithm. All these variables represent sets.

- \( RF \) for Received From. RF contains a set of triplets \((id, dist, time)\) such that:
  - \( id \) is the identifier of a node \( v \) such that \( u \) can receive messages from \( v \);
  - \( dist \) is the distance from \( v \) to \( u \);
  - \( time \) since when \( u \) did not receive messages from \( v \).

- \( PN \) for Potential neighbors. PN contains a set of couples \((id, dist)\) such that:
  - \( id \) is the identifier of a node \( v \) bidirectionally linked with \( u \);
  - \( dist \) is the distance from \( u \) to \( v \);

- \( TPN \) for Transitive Potential Neighbors. TPN contains the set of the potential neighbors of \( u \)’s potential neighbors, it is a set of triplet \((id_1, id_2, dist)\) such that:
  - \( id_1 \) and \( id_2 \) are respectively the identifiers of 2 nodes \( v \) and \( w \) such that \( v \) is a potential neighbor of \( u \) and \( w \) is a potential neighbor of \( v \);
  - \( dist \) is the distance from \( v \) to \( w \).

- \( PF \) for Potential Forwarder. PF contains for each \( u \)’s potential neighbor \( v \), the set of nodes that can forward at low cost a message from \( u \) to \( v \). A node \( w \) in \( PF \) has to check that the distance between \( u \) and \( v \) is higher than both distances between \( u \) and \( w \) and between \( w \) and \( v \). Hence, PF contains a set of couples \((id_1, id_2)\) such that:
  - \( id_1 \) and \( id_2 \) are respectively the identifiers of 2 nodes \( v \) and \( w \),
- v and w are two potential neighbors of u;
- v and w are potential neighbors of each other;
- w can forward a message from u to v;
- dist(u, v) > dist(u, w) and dist(u, v) > dist(w, v).

- **RN** for Reduced Neighbors. RN contains the set of neighbors of u computed by the algorithm. The transmission range of u is computed according to the farthest node of u in RN.

### B. Example

![Diagram](image)

#### Variables values for node 0:
- \( RF_0 = \{1, 5, \_, (2, 12, \_), (3, 4, \_)\} \)
- \( PN_0 = \{1, 5, (2, 12), (3, 4)\} \)
- \( TPN_0 = \{1, 0, 5, (1, 2, 8), (1, 4, 5), (2, 0, 12), (2, 1, 8), (2, 3, 10), (3, 0, 4), (3, 2, 10)\} \)
- \( PF_0 = \{(2, 1), (2, 3)\} \)
- \( RN_0 = \{1, 3\} \)

#### C. Guarded rules for node u

1) **Phase 1**: computation of RF, PN and TPN. During this phase, u does not write neither read in PF and RN.

   **Regularly** →
   //— RF cleaning ————
   ∀x ∈ RF:
   - if x.time > “time bound” ∨ x.time < 0 then
     delete x from RF
   - else increase x.time by 1
   //— PN cleaning ————
   ∀x ∈ PN:
   - if ∃(x.id, y.id) ∈ RF then
     delete x from PN
   //— TPN cleaning ————
   ∀x ∈ TPN:
   - if ∃(x.id, y.id) ∈ PN then
     delete x from TPN

#### On the reception of HELLO((v, RF_v, PN_v) →

   //— RF update ————
   d := distance from v to u
   - if (v, y, _ ) ∉ RF then
     add (v, d, 0) to RF
   - else if x = (v, d, _ ) ∈ RF ∧ d’ ≠ d then
     delete x from RF
   - else x.time := 0
   //— PN update ————
   if Ǝx ∈ RF_v : x.id = u then
   - if (v, _ ) ∉ PN then
     add (v, x.dist) to PN
   - else if (v, d’) ∈ PN ∧ d’ ≠ x.dist then
     delete (v, d’) from PN
     add (v, x.dist) to PN
   - else delete all y in PN such that y.id = v

2) **Phase 2**: computation of RF, RN and the reduced transmission range r.

**Procedure compute-reduced-transmission-range()**

Sort PN according to the distance dist, from the highest distance to the smallest one

- Update-PF()
- Update-RN()

\( r := \max\{d : (v, d) \in PN \wedge v \in RN\} \)

**Procedure update-PF()**

Reset PF to empty set

∀x ∈ PN chosen from the highest distance (x.dist) to the smallest one do

   - if ∃x’ ∈ PN : x ≠ x’ ∧ x’.dist < x.dist do
     if (x’.id, x.id, d) ∈ TPN ∧ x’.dist > d then
       add (x.id, x.id, d) to PF
   continue := true

**Procedure update-RN()**

\( PF’ := PF \)

RN := all identifiers in PN

continue := true

∀x ∈ PN chosen from the highest distance (x.dist) to the smallest one and if (continue) do

   - continue := false
   - if (∀x’ ∈ PN : x ≠ x’ ∧ x’.dist > x.dist ∧
       (x’.id, x.id) ∈ PF’ ∧
       |{(x’.id, _ ) ∈ PF’}| > 1) then
     delete x from RN
     ∀x’ ∈ PN : x’ ≠ x ∧ x’.dist > x.dist do
     delete (x’.id, x.id) from PF’
   continue := true

**V. Performance evaluation**

To evaluate our distributed self-stabilizing algorithm of transitive topology control in WSN, we compare it with two other algorithms proposed in the literature CONNECT [9], and CDS_based [12]. We implement the algorithms and different observers to collect the results by using Peersim simulation platform [16]. In this section, we present a comparative of the transmission range and the energy used for broadcast between the implemented algorithms. To this purpose, we first describe the parameters setting and the energy consumption model used for our simulations.

**A. Simulation configuration**

For each simulation configuration, 100 simulations are performed and confidence intervals are computed at 95%.

Sensor nodes are randomly scattered in 200m × 200m square and each node has a maximum transmission range of 25m.
The network size varies between 500 and 5 000 nodes. The energy consumption model [1] is given by equations (1) and (2).

\[ E_t(k, d) = E_{elec}k + E_{amp}k d^\alpha \]
\[ E_r(k) = E_{elec}k \]

\( E_t \) and \( E_r \) are respectively the energy used for transmission and receiving \( k \) bits with Euclidean distance \( d \). In these formulas, the radio dissipates \( E_{elec} = 50nJ/\text{bit} \) to run the transmitter or receiver circuitry, and the radio dissipates \( E_{amp} = 100pJ/(\text{bit.m}^2) \) to run transmit amplifier. The model of energy consumed is used for a noise-free environment, so we take \( \alpha \) equal to 2. Nevertheless, in the presence of noise, \( \alpha \) is greater than 2, therefore the benefit in terms of energy from the reduction of transmission range induced by our algorithm is even better.

The simulation parameters are summarized in TABLE I.

<table>
<thead>
<tr>
<th>Simulation Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>200m * 200m</td>
</tr>
<tr>
<td>Network size</td>
<td>500 - 5 000</td>
</tr>
<tr>
<td>Maximum transmission Range</td>
<td>25m</td>
</tr>
<tr>
<td>( E_{amp} )</td>
<td>100pJ/(bit.m^2)</td>
</tr>
<tr>
<td>( E_{elec} )</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>2</td>
</tr>
</tbody>
</table>

**B. The transmission range**

Our algorithm reduces as much as possible the transmission range of the nodes in the network while maintaining the connectivity of the network. We evaluate the average transmission range in the network by comparing our algorithm with the centralized solution CONNECT and with the distributed solution that is proposed for sensor-to-sensor communication category CDS-based solution.

In this evaluation, we simulate the three algorithms for different network sizes varying between 500 and 5 000 sensors. The average transmission ranges are collected after stabilization of the algorithms.

![Figure 2. The average range transmission](image)

In figure 2 we can observe that the average transmission range of our solution is close to CONNECT, that is a centralized solution and is more than twice lower than CDS-based for a networks size of 2 000 nodes and more.

The performance of CONNECT can be explained by a global knowledge of the centralized algorithm that allows it to take better decisions. However, we can observe that by using local information, our algorithm gives an average transmission range very close to that of CONNECT that is a very interesting result.

Besides, our algorithm gives better results than CDS-based solution that is logical because of the objective of CDS-based that is dedicated to communication between sensors. Indeed, in sensor-to-sensor communication, we have to preserve the diameter of the network.

**C. The energy consumption for a broadcast**

Reducing energy consumption for the communication by broadcast is our main goal in this work. To evaluate our algorithm, we compute the average energy consumed in the network for a single broadcast. We consider that if one node initiates a broadcast, it sends a message to its neighbors. For all the nodes, if a node receives the message for the first time, it forward it. Given that the network is connected, all the nodes of the network receive the message.

In the following, we describe the computation of the energy consumed for the broadcast of one bit message. The final result is the average energy consumed for all the broadcast.

The energy consumption for the broadcast is based on energy consumption model presented in section V-A and data collected from the simulations.

The average energy consumed in the network for a broadcast (\( E_{broadcast} \)) is the sum of the energies consumed for the successive transmissions(\( E_{ST} \)) and the energies consumed for successive receptions(\( E_{SR} \)) as shown by equation (3).

Equation (4) and (5) represent respectively the calculation formulas for (\( E_{ST} \)) and (\( E_{SR} \)).

\[ E_{broadcast} = E_{ST} + E_{SR} \]  \hspace{1cm} (3)
\[ E_{ST} = n \times E_t \]  \hspace{1cm} (4)
\[ E_{SR} = n \times D \times E_r \]  \hspace{1cm} (5)

Where \( n \) is the network size and \( D \) is the average degree.

To compute this energy, we collect from the simulations \( n \), \( D \) and the transmission range squared average after stabilizing of the algorithms.

It is worth noticing that in equation (4), the Euclidean distance used in \( E_t \) is the transmission range squared average.

We compare our algorithm with CONNECT, CDS-based and with a network where the nodes use their maximum transmission range (Range Max).

We can observe, from figure 3, that our solution and CONNECT give a good result comparing to CDS-based and Max range. With our solution and CONNECT, the increase
Figure 3. The average energy consumed in the network for a broadcast of the energy is linear with the increase of the size, whereas for the other two is exponential increase.

The obtained results show that the behavior of the system with our algorithm is the desirable behavior. Our algorithm with local information gave results very close to CONNECT that is a centralized algorithm.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we presented an innovative solution for topology control in Wireless Sensors Networks. The diversity of applications in WSN is that there is no category of communication that can be adapted to any type of application. Our solution is proposed for energy saving in case of communication by broadcast.

The evaluation of our algorithm shows that in this context, our solution gives very promising results. Besides, the self-stabilizing property of our solution allows the system to recover after any transient fault such as variables corruption, failure, restart or addition of nodes in the network.

As future work, we plan to investigate the behavior of our algorithm in other contexts of communication such as sensor to sensor communication.

REFERENCES


