Group Selection by Nodes in Wireless Sensor Networks Using Coalitional Game Theory

Fatemeh Kazemeyni, Einar Broch Johnsen, Olaf Owe
University of Oslo, Norway
Email: {fatemehk,einarj,olaf}@ifi.uio.no

Ilangko Balasingham
The Interventional Center,
Oslo University Hospital, Norway
Email: ilangkob@medisin.uio.no

Abstract

Wireless sensor networks consist of resource-constrained nodes; especially with respect to power resources. In many cases, the replacement of a dead node is difficult and costly, e.g. an implanted node in the human body. Our main goal in this paper is reducing the total power consumption of the network. For this purpose, we consider the cooperation of nodes in data transmission in terms of a group, *since the major consumer of power is the data transmission process. A mobile node may move to a new location, in which it is desirable for the node to join a group. In this paper, we propose an algorithm for nodes to choose the best group in their signal range, using coalitional game theory to determine what is beneficial in terms of power consumption. The protocol is formalized in rewriting logic, implemented in the Maude tool, and validated by means of Maude's model exploration facilities. Simulation-based tools are in general not able to prove the protocol. However, by using Maude, we prove the correctness of our proposed protocol, by searching for failures of the protocol, through all possible behaviors of sensors. These searches prove that grouping nodes is done correctly in all reachable states from a set of initial states of the model. In addition, we simulate our model in order to quantitatively analyze the efficiency of the proposed protocol. The results show significant improvements in power efficiency.*

1. Introduction

A wireless sensor network (WSN) typically consists of sensor nodes with sensing, computing, and communication devices. The main goal of the WSN is to gather data from the environment and transmit it to a sink node. WSNs are usually self-configured ad-hoc networks with mobile nodes.

The physical size of sensor nodes is very small, which introduces challenges for the design and management of WSNs. Especially, restrictions in power resources need to be considered in order to improve the longevity of the nodes. Data transmission is expensive, therefore, the management of communication between nodes is an important factor in power efficiency of the network. *Cooperation* between sensor nodes can potentially reduce the total power consumed for data transmission in the whole network by replacing multi-hop with traditional single-hop communication.

Grouping is a method to organize node cooperation in a WSN. A group has a selected node called the *leader* which is responsible for receiving data from the group members and communication with the outside of the group. Inside a group, nodes help each other to transmit data to the leader using multi-hop instead of single-hop communication thereby expecting to reduce the consumed power.

Nodes which are close to each other, may in principle communicate using less power. By cooperating inside a group, the group's members can decrease their transmission power to minimum and still reach the leader. However, if nodes do not have fixed locations, the network topology can change. Nodes should compute the most efficient way to communicate in the network. Consequently, the group structure of the network may need to evolve. In a self-organizing network with a dynamic topology, a node which moves may want to join a group to have a cheaper communication and the group needs to decide whether to accept the node.

This paper proposes a protocol to decide which group could be the best for the node to join. The node chooses a group such that joining it is beneficial for the node, for the group and also whole the network. In order to decentralize the grouping process, in this protocol the nodes choose the best group to join with respect to the total energy of the network. Our protocol uses coalitional game theory to decide on the best group membership. This work extends previous work [17], also using coalitional game theory for deciding group membership, but where the leader is responsible for deciding about the node's membership in the group, solely based on the local benefit of the group.

We use formal techniques instead of network simulation, in order to analyze our protocol in a more abstract model so that it is possible to prove its correctness in addition to simulate its behavior. Simulation-based tools can provide useful statistical results from protocol's behavior, but since it is practically impossible to test exhaustively all the behaviors of networks, these tools can not prove the correctness of a protocol. By using formal techniques, we can inspect all reachable states of a system and prove its correctness. We develop a formal, executable model of the proposed protocol in rewriting logic [20]. The resulting model is analyzed using Maude [4]. We show correctness of the proposed protocol.

1.1. Related Work

The energy conservation approaches that have been proposed in the literature could be categorized in three: duty cycling, data-driven, and mobility approaches [2]. The *duty cycling* approach is concerned with networking subsystems and sleep/wake-up scheduling algorithms. These methods, such as in [11], [32], try to find efficient subsets in the network and schedule the activity of network nodes. The purpose of the *data driven* approach is to reduce the data that is transmitted between sensors or the sink node, and considers data compression methods [30]. The *mobility-based* method such as presented in [5] and [28], can be categorized as mobile-sink and mobile-relay methods, depending on the type of the mobile entity. Mobile entities can gather the data from the nodes by using short range communication, which is an efficient way of communication with respect to energy. The grouping technique proposed in this paper is related to the duty cycling approach, because the group members and the group leader arrange their duties in order to cooperate with each other, and thereby conserve the total energy of the group and the network.

Noncooperational game theory has been used to reduce the power consumption of sensor nodes, applying a utility function to find the Nash equilibrium [14], [21], [29]. Coalitional game theory is applied to reduce the power consumption in WSNs by [27], who propose a merge and split approach for coalition formation. They calculate the value of the utility function for every possible permutation of nodes and find groups with the best utility value. This is as far as we know the only previous work that uses coalitional game theory for grouping the sensor networks. In contrast, we develop and formalize a protocol which considers nodes which may need to join a new group without reorganizing the entire WSN.

WSNs present interesting challenges for formal methods, due to their resource restrictions and radio communication. This has led to research on how to develop modeling languages or extensions which faithfully capture typical features of sensors; e.g., mobility, location, radio communication, message collisions. In addition, WSNs need communication protocols which take resource usage into account. There is a very active field of research on protocol design for WSNs. However, protocol validation is mostly done with simulation-based tools, using NS-2, OMNeT+, and extensions such as Castalia [26] and SensorSim [24].

Formal techniques are much less explored in the development and analysis of WSNs, but start to appear. Among automata-based techniques, the TinyOS operating system has been modeled as a hybrid automaton [7] and UPPAAL has been applied to the LMAC protocol [9] and to the temporal configuration parameters of radio communication [31]. The CaVi tool combines simulation in Castalia with probabilistic model checking [8].

A recent process algebra for active sensor processes includes primitives for, e.g., sensing [6]. A CREOL extension for heterogeneous environments includes radio communication [15]. The Temporal Logic of Actions has been used for routing tree diffusion protocols [22].

Ölveczky and Thorvaldsen show how a rich specification language like Maude is well-suited to model WSNs, using Real-Time Maude to analyze the performance of the OGCD protocol [23]. Their approach has been combined with probabilistic model-checking to analyze the LMST protocol [16].

We follow this line of research and use Maude as a tool to develop a grouping protocol [18] for WSNs, applying coalitional game theory to estimate power consumption.

1.2. Paper Overview

Section 2 introduces WSNs and grouping, and Section 3 coalitional game theory. Section 4 proposes a group membership protocol based on coalitional game theory. Section 5 summarizes rewriting logic and Maude, used to develop a formal model of the protocol in Section 6 and for analysis of the model in Section 7. Section 8 concludes the paper.

2. Grouping the Sensor Nodes

A sensor network is typically a wireless ad-hoc network, in which the sensor nodes support a multi-hop routing algorithm. In these networks, communication between nodes is generally performed by direct connection (single-hop) or through multiple hop relays (multi-hop).

When a large number of sensor nodes are placed in the environment, neighbor nodes may be very close. In this case, the transmission power level for communication with a neighbor can be kept low. Since nodes can cooperate to transmit data, multi-hop communication in sensor networks is expected to consume less power than the traditional single-hop communication [1]. Furthermore, multi-hop communication can effectively overcome some signal propagation effects experienced in long-distance wireless communication.

Nodes broadcast their data to all nodes within the range of their data transmission, which is determined by the power used for transmission. Usually in protocols such as the standard AODV protocol [25], sensor nodes use their maximum data transmission power to cover a larger area and reach more nodes, both for data transmission and for routing.

Grouping is a method for cooperation between nodes, i.e., to transfer data, in which nodes belong to distinct groups [18]. Each group has a group *leader*; a node which is responsible for receiving data from the group members in order to route it to the sink node, also communicating with other group leaders. Inside the group, it is not always necessary for a node to use its maximum transmission power. Instead, by cooperation between the group members, nodes can use their minimum transmission power to reach the group leader, and consequently decrease the power consumed for communication in the group. The group formation techniques are different. The grouping can be done based on special characteristics or distance. Regarding to special characteristics, a special correlation among the sensors could be found by using vector quantization [12]. For example, all the sensor nodes that have similar sensed data could be placed in one group. The sensor nodes could also be placed in groups based on the distance. The location of the nodes can be determined using different methods, such as GPS. For a better grouping, other factors such as signal interference could also be considered for group formation.

3. Coalitional Game Theory

Game theory [10] can be used to analyze behavior in decentralized and self-organizing networks. Game theory typically models the nodes as players and choice of strategies of self-interested players, in order to capture the interaction of players in an environment such as a communication network. A game consists of

- a set of *players* $\mathbf{N} = \{1, 2, ..., n\}$;
- an indexed set of *possible actions* $A = A_1 \times A_2 \times ... \times A_n$, where A_i is the set of actions of player i (for $0 < i \leq n$);
- a set of *utility functions*, one for each player. The utility function u assigns a numerical value to the elements of the action set A; for actions $x, y \in A$ if $u(x) \geq u(y)$ then x must be at least as preferred as y.

Game theory can be categorized into noncooperative [3] and cooperative game theory [10]. Noncooperative game theory studies the interaction between competing players, where each player chooses its strategy independently and the goal of each player is to improve its utility or reduce its cost [27].

In cooperative games, groups of players are formed, called coalitions. players trying to find a coalition to strengthen their position in the game and make an agreement to act as a simple entity. Coalitional games have proved useful to design fair, robust, and efficient cooperation strategies in communication networks. In a coalitional game (N, v) with N players, the coalition value or utility of a coalition is determined by a *characteristic function* $v : 2^N \to \mathbb{R}$ which applies to coalitions of players.

The core of the coalitional game (N, v) guarantees that no player has an incentive to leave N to form another coalition [27].

4. A Protocol for Deciding Group Membership

Consider the grouping problem for wireless sensor networks as a coalitional game. The sensor nodes are the players and the game is concerned with whether a node should join a group or not. The goal is to reduce the total power consumption in the network, so we need a utility function which reflects the power consumed for data transmission and signal interference. The utility function proposed by Goodman *et al.* [13] appears to be a suitable choice when power consumption is an important factor of the model [19]:

$$w(P_j, \delta_j) = (\frac{R}{P_j})(1 - e^{-0.5\delta_j})^L. \qquad (1)$$

When applying w to a node j, P_j is the power used for message transfer by j and δ_j is the signal to interference and noise ratio (SINR) for j. In addition, R is the rate of information transmission in L bit packets in the WSN.

Nodes can transfer data with different amounts of power. Let P^{max} denote the maximum transmission power and P_j^{min} the minimum power for each node j, such that $0 \leq P_j^{min} \leq P^{max}$. When a node j cooperates in a group, it uses P_j^{min} for message transmission, and otherwise P^{max}. Consider a network of nodes $N = \{1, \ldots, n\}$. If all the nodes in N cooperate, we have:

$$\sum_{j=1}^{n} w(P_j, \delta) = \sum_{j=1}^{n} w(P_j^{min}, \delta)$$

In the case without cooperation, P_{max} is assigned to P_j. Observe that if this utility function were applied naively, it would always be beneficial for nodes to form a coalition, as the result of decision making is the same for every topology of the network and every group.

However, in reality all the cooperating nodes use power in order to transmit data to the group leader, so it is not sufficient to only consider the power consumption of the original sender of data in the utility function. Although each node uses its minimum power to transmit data, the node's total power usage depends on the number of messages it needs to transmit. Each node on the route between the source node and the leader, needs to send its own data as well as the data that it has received from the previous node. In general, the power consumption for the intermediate nodes will increase.

We modify the utility function (Formula 1) to capture the overall power usage needed to transmit the data from the node to the leader following a given path:

$$u(P_j, \delta_j) = \left(\frac{R}{\sum_{n \in RP_{j,Leader}} P_n^{min}}\right)(1 - e^{-0.5\delta_j})^L, \tag{2}$$

where the set $RP_{j,Leader}$ contains all nodes in the routing path between node j and the leader. This utility function is similar to Formula 1 except that the power that is applied is the sum of the power consumed by all the nodes in the routing path through which data is transmitted from the sender to the leader.

The power consumed for routing data from a non-member to the leader follows Goodman *et al.* (Formula 1) and is based on maximum power single hop:

$$w(j, \delta_j) = \left(\frac{R}{P_j^{max}}\right)(1 - e^{-0.5\delta_j})^L$$

Using the utility function u, the leader can decide about the membership of a new node with more realistic estimations. The result depends on the specific topology, so coalition is not always beneficial. Consequently, it is more beneficial for the node to follow a path through the group than to act individually when $w(j, \delta_j) < u(j, \delta_j)$ holds. To calculate the power that is used in the cooperation, we have proposed a *power-sensitive AODV* routing protocol [17], modifying AODV to find the cheapest path between the node and the leader in terms of power. The leader may then decide to add the node to its group and sends an `Invite` message to the node. A node may receive several `Invite` messages from different leaders due to the intersection of groups. In this paper, we consider how the node may select the best group to join, using game theory. It chooses to join the group which is the most beneficial for the overall network.

Consider a group i with leader *Leader*. Let M be the set of nodes which can reach *Leader* with P^{max} and N the set of group members. Let the accumulated *group utility value* g_i be determined by the sum of the utility values for communication with *Leader*:

$$g_i(M, N) = \sum_{j \in M} w(j, \delta_j) + \sum_{j \in N} u(j, \delta_j)$$

The group membership protocol extends the power-sensitive AODV protocol as follows:

1) Node j sends a `Hello` message with maximum power to all group leaders within range;
2) Each group leader runs the power-sensitive AODV protocol to find the cheapest path for j as a potential group member and evaluates the benefit of group membership for j: $g_i(M \cup \{j\}, N) < g_i(M, N \cup \{j\})$;
3) If membership is beneficial, group i's leader sends an *Invite* message to j, including the utility values $v^i_{\text{old}} = g_i(M \cup \{j\}, N)$ and $v^i_{\text{new}} = g_i(M, N \cup \{j\})$;
4) Node j may receive many `Invite` messages, which are processed sequentially. By assumption, j is currently in group a and knows v^a_{old} and v^a_{new}; For each invitation, j computes $v^i_{\text{old}} - v^i_{\text{new}} > v^a_{\text{old}} - v^a_{\text{new}}$. If this is the case, j accepts the invitation from i and sends a *Leave* message to a with the value $v^a_{\text{new}} - v^a_{\text{old}}$;
5) The *Leader* receives an `Accept` message from j and updates its utility approximation;
6) The *Leader* receives a `Leave` message with value v_j, and updates its utility approximation v to $v - v_j$.

The core of this game is not empty and nodes can form groups. Assuming two disjoint groups i and a with $v(a \cup i) = 0$, the core is empty [27]. In our game, $i \cap a \neq \emptyset$, $g_i(M, N) = v(M, N)$ and $g_i(M \cup \{j\}, N) < g_i(M, N \cup \{j\})$, which proves the core is not empty and no node or group of nodes has incentive to leave the coalition.

5. Rewriting Logic and Maude

The formal model of the protocol is defined in rewriting logic (RL) [20] and is executable on the RL system Maude [4]. A rewrite theory is a 4-tuple (Σ, E, L, R) where the signature Σ defines the function symbols, E defines equations between terms, L is a set of labels, and R is a set of labeled rewrite rules. Rewrite rules apply to terms of given sorts. Sorts are specified in (membership) equational logic (Σ, E). When modeling computational systems, different system components are typically modeled by terms of suitable sorts defined in the equational logic. The global state configuration is defined as a multiset of these terms. RL extends algebraic specification techniques with transition rules: The dynamic behavior of a system is captured by rewrite rules supplementing the equations which define the term language. From a computational viewpoint, a rewrite rule $t \longrightarrow t'$ may be interpreted as a *local transition rule* allowing an instance of the pattern t to evolve into the corresponding instance of the pattern t'. When auxiliary functions are needed in the semantics, these are defined in equational logic, and are evaluated in between the state transitions [20]. If rewrite rules apply to non-overlapping sub-configurations, the transitions may be performed in parallel. Consequently, concurrency is implicit in RL. Conditional rewrite rules $t \longrightarrow t'$ **if** cond are allowed, where the condition cond is a conjunction of rewrites and equations that must hold for the main rule to apply. Maude provides model checking tools to check desired properties of a model and a search tool that searches through all reachable states while checking given properties.

6. A Formal Model of Regrouping

In this section, we define a formal model of the group membership protocol in rewriting logic. In the model, we assume that there is no message loss in the protocol, that messages do not expire, and that the topology of the network consists of a fixed number of nodes, but nodes can move. A *system configuration* is a multiset of objects and messages inside curly brackets. Following RL conventions, whitespace denotes the associative and commutative constructor for configurations. The term ⟨O : Node | Attributes⟩ denotes a Node object, where O is the object identifier, and Attributes a set of attributes of the form Attr : X where Attr is the attribute name and X the associated value.

In the sequel, we explain the rules and equations modeling wireless message passing, node movement, and the routing protocol, as well as the evaluation of the utility function.

6.1. Unicast and Broadcast

Unicast messages have the form

```
(M from O X Y P to O′)
```

where M is the messages body (possibly with parameters), O the source with current location (X,Y), O′ the destination, and P the sending power used. A message will not reach its destination unless it is within the range. This is modeled by the conditional equation (ceq)

```
ceq (M from O X Y P to O´)
⟨O´:Node|xLoc:X´,yLoc:Y´,A⟩ =
⟨O´:Node|xLoc:X´,yLoc:Y′,A⟩
if not inrange(X,Y,X´,Y´,P ).
```

where inrange is a Boolean function checking that the two locations (X,Y) and (X′,Y′) are in range of each other with power P (using the calculated distance and the network parameters including the interference level). Note that this equation removes a message which cannot reach its destination, depending on the location values at sending time. *Multicasting* is modeled by allowing a set of destinations and equations which expand the destination set:

```
eq (M from O X Y P to noneOids) = none .
eq (M from O X Y P to O′; Os) =
 (M from O X Y P to O′) (M from O X Y P to Os).
```

Here, Os denotes a set of object identities (with “;” as multiset constructor). *Wireless broadcasting* uses messages

```
(M from O X Y P to all)
```

where all is a constructor indicating that the message is sent to *all nodes within range*.

6.2. Node Movements

In most WSNs, nodes can move and change their location. Therefore, a WSN model should provide suitable rules for changing the position of nodes. We have modeled three different methods for node movement.

In the first method, the node can move freely everywhere, captured by rules that non-deterministically change the location of the node. In the second one, a node can move directly to a desired location. A rule will change the location of the node in one step. In the last method, a node can move to a desired location through a non-deterministic path, there are rules that determine the final destination and do non-deterministic but finite steps toward it.

6.3. The Regrouping

Each node should inform neighboring leader nodes about its movements. This is done by broadcasting a hello message with maximum power when the node has changed position. The following rule represents the hello broadcasting:

```
rl[moving-done]:
(movemsg Xn Yn from O X Y P to O)
⟨O: Node | xLoc:Xn,yLoc:Yn,power:P,energy:E,A⟩
⟶
⟨O: Node | xLoc:Xn,yLoc:Yn,power:P,energy:E−Pmax,A⟩
(hello from O X Y Pmax to all ) .
```

Here, the new location of the node is defined by (Xn,Yn). When a neighboring group leader receives this hello message, a new node has entered the group's signal range. Each message transmission reduces the node's total energy E with respect to the amount of energy that is consumed for sending the message. The leader starts the process to decide whether it is beneficial to accept the new node as a group member based on the power usage in the result path. The leader first runs the power-sensitive AODV protocol (presented below in Section 6.4) with minimum power to find the cheapest path to the new node. If a path is found, the modified AODV protocol ends by letting the leader send a message membershipMsg to itself. This message starts the decision making process about the node's membership, which is captured by the following rules:

```
crl[Membership-decision]:
(membershipMsg Oc from O)
⟨O:Node | leaderNode:true,members:OS1,id:Id,xLoc:X,
yLoc:Y,utility:U,routingTable:RT,energy:E,A' ⟩
⟨Oc: Node | id:Id1,Ac⟩
⟶
⟨O:Node | leaderNode:true,members:OS1,id:Id,xLoc:X,
yLoc:Y,utility:U,routingTable:RT,energy:E−Pmin,A' ⟩
⟨Oc:Node | id:Id1,Ac⟩
InvitationMsg Id X Y U NU from O to Oc
if NU:=newUtility(U,findPower(RT,Id1))
∧ joinGroup(U,findPower(RT,Id1)).

rl [Joining]:
(join from O X Y P to Oc)
⟨Oc: Node | leader:[O' X' Y'],A⟩
⟶
⟨Oc: Node | leader:[ O X Y ],A⟩ .
```

Here, O is the leader, Oc is the new node, U is the old utility value and NU is the new utility value of the group. The function findpower extracts the value of required power for data transmission from the routing table, the function newUtility calculates the new value of the utility function after joining the node, and the function joinGroup represents the computation of the utility function (Formula 2), formalized as follows:

```
op joinGroup : Nat → Bool .
eq joinGroup (P)  =
(RATE quo P * ((1 - 2.71^(0.5 * I))^PACK) >
(RATE quo Pmax) * ((1 - 2.71^(0.5 * I))^PACK).
```

Here, P is the total power consumed in the routing path, and Pmax, I, RATE, and PACK, are constants reflecting the maximum sending power, the transmission rate, and the packet size, respectively. These constants can be seen as network parameters, and suitable values given as parameters to the initial configuration. The output of joinGroup is a Boolean value. The leader uses this function to decide if a new node could be added as a member. If the leader decides to add the node, it sends an invitation message to the node. The node may receive several invitation messages in case of multiple groups, therefore it should choose one of membership offers that is best for it and also the network. The following rule represents the node's behavior after receiving the invitation:

```
crl[Grouping]:
(InvitationMsg Id X Y U NU from O' to Oc)
⟨O': Node | leaderNode:true,id:Id,xLoc:X,yLoc:Y,
utility:U,routingTable:RT,A' ⟩
⟨Oc: Node | leader:LEADER,id:Id1,utility:Uc,
oldUtility:OU,power:P,energy:E,Ac⟩
⟶
⟨O': Node | leaderNode:true,id:Id,xLoc:X,yLoc:Y,
utility:U,routingTable:RT,A' ⟩
⟨Oc: Node | leader:[Id X Y],id:Id1,utility:NU,
oldUtility:U,power:P,energy:E-(Pmin+Pmax),Ac⟩
AcceptMsg NU OU from Oc to O'
LeaveMsg NU OU from Oc to LEADER
if bestGroup(LEADER,OU,Uc,X,Y,P,U,NU).
```

Here, NU and OU are the new and old utility values of the group and bestGroup is a function which compares the different membership offers. In this model, like the real environment, messages are queued and received one by one. So, each time we just need to compare two offers. The bestGroup function works as follows:

```
op bestGroup: ListListNat Nat
 Nat Nat Nat Nat Nat Nat → Bool

ceq bestGroup(LEADER,OU,Uc,U,NU,X,Y,P) =
abs(Uc−OU) < abs(NU−U)
if inrange(X(LEADER),Y(LEADER),X,Y,P) .
```

Inputs of this function are the leader LEADER, old utility value of the current group OU, current utility value of the current group Uc, the utility value of the the new group without the node U, the utility value of the new group with the node NU and also location (X,Y) and power P of the node. It compares the difference between the utility that is gain by joining to the new and previous group and accept the new offer if it increase the utility of the new group more than the previous group. If the node decides to change to the new group, it will inform the new and previous groups' leaders by sending AcceptMsg and LeaveMsg messages.

6.4. The Routing Protocol

The routing protocol of Section 4 is now formalized. The main difference between our protocol and AODV is that we find the *cheapest path* instead of the shortest one. In the model, each node has its own routing table that stores the path to each destination. For each destination, the routing table stores the following information: the next node on the path to the destination and the required power to send data to the destination. When the node finds a cheaper path to a destination (a path which requires less power), it updates its routing table and replaces the old path with the cheaper one. The neighbors of a node are stored in a list neighbors.

All messages in the routing protocol are modeled as Maude messages and behave as explained in Section 6.1. There are several rules to control the message propagation in the model by receiving a route request or a route reply message and sending a new message which is either a reply or a request.

7. Analysis of the Case Study

Maude provides different tools for testing and validating the model. It can run the model through one path of the state space like a simulator. As a case study, we consider a topology with six nodes. In this topology, the nodes b and f are leaders of different groups. We simulate our model with different initial states. Initial state consists of a topology of the network and a set of nodes' movements. In our analysis the topology was the same for all the initial states, but movements of the nodes were different. The set of the movements includes the following cases:

- Movement of one node in each run that is repeated for different nodes regardless of being a normal node or leader, in the area of the same group or to the range of another group and also out of the range of any group.
- Simultaneous movement of two nodes in each run. Several permutation of nodes' movements were considered in this part of the test, including moving nodes to the same group or different groups.

We simulated the model and analyze the final states to find out if the model behaves correctly. The correctness is formalized as follows:

$$\Box\neg(\mathit{Member}(O,L) \Leftrightarrow \mathit{UtilityEnhancement}(L)) \tag{3}$$

Where $\mathit{Member}(O,L)$ is true if node O is a member of the group of leader L. Likewise, $\mathit{UtilityEnhancement}(L)$ is true when the new utility of the group of L is more than its previous utility, considering all nodes. Formula 3 means that in all the states of the system, the membership of the node in the group is accepted by the leader only when this membership is beneficial for the group and enhance the utility. In other words, there is no state that the utility value decreases but the node's membership is accepted.

To exemplify, consider the scenario in which nodes c and d change their location such that node c stays and d comes within the range of the leader f. We first use Maude to check this property by simulating the model. The result of the simulation is

```
⟨"a":Node|leaderNode:false,id:1,leader:[2 1 1],
members:noneOids,xLoc:2,yLoc:2,power:1,
utility:4,routingTable:[0 0 0],reqid:0,
oldUtility:0,energy:870⟩

⟨"b":Node|leaderNode:true,id:2,leader:[2 1 1],
members:("a";"e"),xLoc:1,yLoc:1,power: 1,
utility:0,routingTable:[0 0 0],reqid:0,
oldUtility:0,energy:805⟩

⟨"c":Node|leaderNode:false,id:3,leader:[6 10 3],
members:noneOids,xLoc:8,yLoc:4,power:1,
utility:5,routingTable:[0 0 0],reqid:1,
oldUtility:0,energy:835⟩

⟨"d":Node|leaderNode:false,id:4,leader:[6 10 3],
members:noneOids,xLoc:9,yLoc:3,power:1,
utility:7,routingTable:[0 0 0],reqid:1,
oldUtility:5,energy:820⟩

⟨"e":Node|leaderNode:false,id:5,leader:[2 1 1],
members:noneOids,xLoc:8,yLoc:4,power:1,
utility:0,routingTable:[0 0 0],reqid:1,
oldUtility:0,energy:800⟩

⟨"f":Node|leaderNode:true,id:6,leader:[6 10 3],
members:("c";"d"),xLoc:10,yLoc:3,power:1,
utility:7, routingTable:([4 4 1][0 0 0]),
reqid:1,oldUtility:0,energy:765⟩
```

By inspecting the members attributes of leader f, we see that node c and d now are members of f's group.

The simulation results showed that the model works as expected in all the cases.

Although we did several simulations to improve the trustworthiness of the results, simulation can not prove the correctness of the model because it just checks one path in the system's state space, whereas to prove the validity of the model all the possible paths of the state space should be checked for failure. To achieve this goal, we search all possible states of our model using Maude's search command for a number of given initial states. The search result proves that for all possible traces the model works correctly. For example, when node d (id=4) moves to the point(10,3) which is closer to leader f, the following search command gives all possible final states:

```
search initState( [ 4 10 3 ] )
⟶ ! C:Configuration
```

All the solutions show that at the final state, node d is a member of leader f and there is no case of failure. In addition, we have analyzed the effects of the grouping protocol on the energy consumption of the WSNs. For this purpose, the Maude's simulation tool is used repeatedly. The final result is the average of the results of all the simulations.

In the beginning of the model execution, the nodes start sending data messages. During the execution, they can move and join a new group. We ran simulations for two distinct scenarios, namely, when the WSN uses the grouping protocol vs. when it does not. Our purpose is to compare the power consumption of the nodes and the leaders, in each separate scenario. The network's architecture could be designed to provide low cost communications between leaders and sink nodes, such as MULE-based architecture for WSNs [28]. Therefore, We can assume that the leaders use the minimum transmission power to send messages to sink nodes.

Figure 1 and Figure 2 represent the saved energy of a sensor node and of a leader, with (red) and without (blue) using the grouping protocol. To generate each of the graphs in the figures we mentioned, we ran 5 simulations, with each simulation lasting for 1000 time units that is captured by rewrite steps, i.e. one time unit corresponds to one rewrite step. In the initial configuration, an initial value is assigned to the total energy of each node. After each message transmission, including data messages that are sending regularly by nodes and the messages that are related to the grouping protocol, the value of total energy is modified and captured to be shown in the graph. The results show that for a normal sensor node, it is always beneficial to join a group. Even for a leader, when it uses its minimum transmission power for all of its communications, it is more beneficial to be in a group than to be a separate node which should send only its own messages but by using the maximum transmission power.

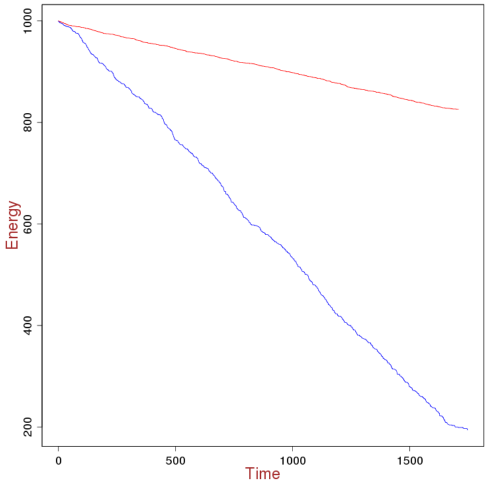

Figure 1. The remaining energy of a node.

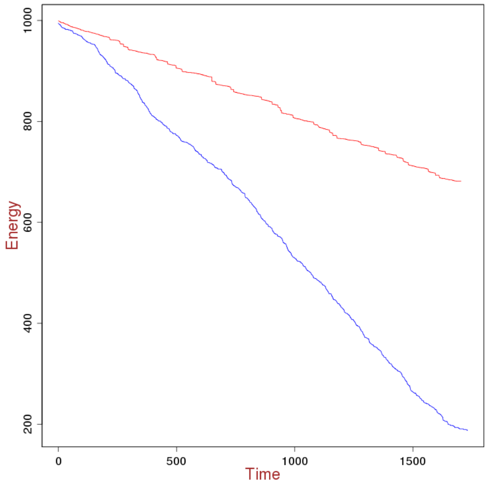

Figure 2. The remaining energy of a leader.

8. Conclusion

In this paper, we propose an improved group membership protocol for WSNs to choose the best available group by the node. In this protocol, members cooperate with each other to transmit data, in order to decrease the total power consumption of the group and also the network. A node may move towards the range of several groups. It should choose the best group to join, applying coalitional game theory with respect to the total power consumption. The analysis of the protocol was done by formalizing the protocol in rewriting logic and Maude is used to analyze its behavior for several scenarios.

In future work, we intend to build on our current Maude model as well as extending the model to capture real-time aspects of WSNs. Furthermore, we plan to refine the utility function used in this paper, i.e. to capture the interference of the transmission signals of the nodes. We are also going to study changing the nodes role in a group and selecting a new leader. In addition, we want to capture the correlation of transmitted data to send them more efficiently by considering this correlation in the cooperation of the nodes. In the other hand, we plan to do probabilistic model checking in order to statistically prove the correctness of the model.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Netw.*, 7:537–568, May 2009.

[3] T. Başar and G. J. Olsder. *Dynamic non-cooperative game theory*. SIAM, 1999.

[4] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. F. Quesada. Maude: Specification and programming in rewriting logic. *Theoretical Computer Science*, 285:187–243, Aug. 2002.

[5] M. Conti, A. Passarella, and L. Pelusi. Mobile-relay forwarding in opportunistic networks. In *Chapter in Adaptive Techniques in Wireless Networks (M. Ibnkahla, Editor*. CRC Press, 2009.

[6] J. S. Dong, J. Sun, J. S. 0001, K. Taguchi, and X. Zhang. Specifying and verifying sensor networks: An experiment of formal methods. In S. Liu, T. S. E. Maibaum, and K. Araki, editors, *10th International Conference on Formal Engineering Methods (ICFEM'08)*, volume 5256 of *Lecture Notes in Computer Science*, pages 318–337. Springer, 2008.

[7] S. C. Ergen, M. Ergen, and T. J. Koo. Lifetime analysis of a sensor network with hybrid automata modelling. In C. S. Raghavendra and K. M. Sivalingam, editors, *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pages 98–104. ACM, 2002.

[8] A. Fehnker, M. Fruth, and A. McIver. Graphical modelling for simulation and formal analysis of wireless network protocols. In M. Butler, C. B. Jones, A. Romanovsky, and E. Troubitsyna, editors, *Methods, Models and Tools for Fault Tolerance*, volume 5454 of *Lecture Notes in Computer Science*, pages 1–24. Springer, 2009.

[9] A. Fehnker, L. van Hoesel, and A. Mader. Modelling and verification of the LMAC protocol for wireless sensor networks. In J. Davies and J. Gibbons, editors, *Proc. 6th Intl. Conf. on Integrated Formal Methods (IFM'07)*, volume 4591 of *Lecture Notes in Computer Science*, pages 253–272. Springer, 2007.

[10] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

[11] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin. Networking issues in wireless sensor networks. *J. Parallel Distrib. Comput.*, 64(7):799–814, 2004.

[12] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[13] D. Goodman and N. Mandayam. Power control for wireless data. *IEEE Personal Communications*, 7:48–54, 2000.

[14] H. Inaltekin and S. B. Wicker. The analysis of nash equilibria of the one-shot random-access game for wireless networks and the behavior of selfish nodes. *IEEE/ACM Trans. Netw.*, 16(5):1094–1107, 2008.

[15] E. B. Johnsen, O. Owe, J. Bjørk, and M. Kyas. An object-oriented component model for heterogeneous nets. In F. S. de Boer, M. M. Bonsangue, S. Graf, and W.-P. de Roever, editors, *Proc. 6th International Symposium on Formal Methods for Components and Objects (FMCO 2007)*, volume 5382 of *Lecture Notes in Computer Science*, pages 257–279. Springer, 2008.

[16] M. Katelman, J. Meseguer, and J. C. Hou. Redesign of the LMST wireless sensor protocol through formal modeling and statistical model checking. In G. Barthe and F. S. de Boer, editors, *Proc. 10th Intl. Conf. on Formal Methods for Open Object-Based Distributed Systems (FMOODS'08)*, volume 5051 of *Lecture Notes in Computer Science*, pages 150–169. Springer, 2008.

[17] F. Kazemeyni, E. B. Johnsen, O. Owe, and I. Balasingham. Grouping nodes in wsns using coalitional game theory. In J. Hatcliff and E. Zucca, editors, *Formal Techniques for Distributed Systems, Proc. FMOODS/-FORTE'09)*, volume 6117 of *Lecture Notes in Computer Science*, pages 95–109. Springer, 2010.

[18] J. Lloret, C. E. Palau, F. Boronat, and J. Tomás. Improving networks using group-based topologies. *Computer Communications*, 31(14):3438–3450, 2008.

[19] A. B. Mackenzie and S. B. Wicker. Game theory and the design of self-configuring, adaptive wireless networks. *IEEE Communications Magazine*, 39(11):126–131, Nov. 2001.

[20] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96:73–155, 1992.

[21] D. A. Miller, S. Tilak, and T. Fountain. "token" equilibria in sensor networks with multiple sponsors. In *Proceedings of the 1st International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, 2005.

[22] S. Nair and R. Cardell-Oliver. Formal specification and analysis of performance variation in sensor network diffusion protocols. In S. Balsamo, C.-F. Chiasserini, and L. Donatiello, editors, *Proceedings of the 7th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04)*, pages 170–173. ACM, 2004.

[23] P. C. Ölveczky and S. Thorvaldsen. Formal modeling, performance estimation, and model checking of wireless sensor network algorithms in Real-Time Maude. *Theoretical Computer Science*, 410(2-3):254–280, 2009.

[24] S. Park, A. Savvides, and M. B. Srivastava. SensorSim: a simulation framework for sensor networks. In A. Boukerche, M. Meo, and C. Tropper, editors, *Proc. 3rd Intl. Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2000)*, pages 104–111. ACM, 2000.

[25] C. E. Perkins and E. M. Belding-Royer. Ad-hoc on-demand distance vector routing. In *WMCSA*, pages 90–100. IEEE Computer Society, 1999.

[26] H. N. Pham, D. Pediaditakis, and A. Boulis. From simulation to real deployments in WSN and back. In *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'07)*, pages 1–6. IEEE, 2007.

[27] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Başar. Coalitional game theory for communication networks: A tutorial. *IEEE Signal Processing Magazine*, 26(5):77–97, Sept. 2009. Special Issue on Game Theory.

[28] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *IEEE SNPA Workshop*, pages 30–41, 2003.

[29] R. Strauss and A. Abedi. Game theoretic power allocation in sparsely distributed clusters of wireless sensors (gpas). In M. Guizani, P. Mueller, K.-P. Fähnrich, A. V. Vasilakos, Y. Zhang, and J. Zhang, editors, *IWCMC*, pages 1454–1458. ACM, 2009.

[30] C. Tang and C. S. Raghavendra. *Compression techniques for wireless sensor networks*, pages 207–231. Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[31] S. Tschirner, L. Xuedong, and W. Yi. Model-based validation of QoS properties of biomedical sensor networks. In L. de Alfaro and J. Palsberg, editors, *Proceedings of the 8th ACM & IEEE International conference on Embedded software (EMSOFT'08)*, pages 69–78. ACM, 2008.

[32] A. Warrier, S. Park, J. Min, and I. Rhee. How much energy saving does topology control offer for wireless sensor networks? - a practical study. *Comput. Commun.*, 30:2867–2879, October 2007.