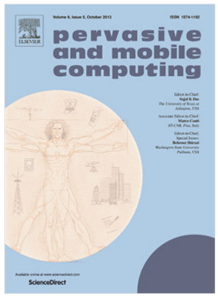
Accepted Manuscript



An evolutionary cluster-game approach for Wireless Sensor Networks in non-collaborative settings

Ketema Adere Gemeda, Gabriele Gianini, Mulugeta Libsie

PII: S1574-1192(17)30030-5
DOI: https://doi.org/10.1016/j.pmcj.2017.10.008
Reference: PMCJ 902

To appear in: *Pervasive and Mobile Computing*

Received date : 5 May 2017
Revised date : 30 September 2017
Accepted date : 13 October 2017

Please cite this article as: K.A. Gemeda, G. Gianini, M. Libsie, An evolutionary cluster-game approach for Wireless Sensor Networks in non-collaborative settings, *Pervasive and Mobile Computing* (2017), https://doi.org/10.1016/j.pmcj.2017.10.008

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

An Evolutionary Cluster-Game Approach for Wireless Sensor Networks in Non-collaborative Settings

Ketema Adere Gemeda[a], Gabriele Gianini[b,*], Mulugeta Libsie[a]

[a]*Computer Science Department, Addis Ababa University, Ethiopia*
[b]*Computer Science Department, Universita degli Studi di Milano,Italy*

Abstract

Wireless Sensor Networks typically consist of a large number of sensor nodes with constrained resources. Cluster-based routing algorithms for WSNs try to preserve battery power by grouping nodes into multiple clusters: a single node in each cluster, the Cluster Head (CH), communicates with a Base Station on behalf of the others. In an ideal collaborative setting, sensor nodes should alternate in the role of CH. However, the cooperation of nodes is not granted in WSNs with more than one governing authority, where sensor nodes can behave selfishly, in order to save their own resources. In this paper, we propose a novel evolutionary cluster-head determination algorithm called GREET, based on an Evolutionary Game Theory (EGT) approach. In the proposed algorithm, individual nodes adapt their strategies on the basis of the outcomes of the interactions with other nodes and converge to an Evolutionary Stable Strategy (ESS) equilibrium. We show that this ESS corresponds to one of the desired behavioral outcomes. This outcome is obtained without the support of external cooperation enforcement mechanisms. In the study, we use an analytic model of the population evolution, based on the so-called *replicator dynamics*, as a guide in the choice of the mechanisms, then we adapt the approach to realistic more scenarios. We show, by means of a systematic simulation study, that the algorithm extends the network lifetime and provides a better packet throughput, w.r.t other standard WSN algorithms, such as LEACH and CROSS.

Keywords: Wireless Sensor Networks, Cluster-based Routing, Evolutionary Game Theory, Snow Drift Game

1. Introduction

Wireless Sensor Networks (WSNs) can consist of hundreds of sensor nodes distributed over a geographical area. Sensor nodes are capable of observing physical phenomena, of processing data, and sometimes of taking appropriate actions [1, 2, 3]. WSNs are typically used for tracking and monitoring. Monitoring applications include environmental monitoring, health monitoring, inventory location monitoring and structural monitoring. These applications are often made possible by the fact that a WSN has a short system setup time and sensor nodes can be deployed with acceptable cost. The versatility of WSNs and their broad range of applications are increasingly attracting the interest by the industry and by the research community.

Sensor nodes are low-power devices equipped with one or more units devoted to sensing and processing, a memory unit, a power supply unit and a communication unit. They can be either stationary or mobile [4]. The hardware of a sensor node may also have additional application-dependent components such as a location finding system, a power generator or a mobilizer.

Sensor nodes collect and route information about the observed physical phenomena – possibly through multiple hops – to a central node, called *sink* or *base station* (BS), for further processing and decision-making. The BS has a dedicated power supply and a higher processing capability and can be connected to other networks, like the Internet. The WSN deployment can be either structured or unstructured [2]. In an *unstructured* WSN, nodes are deployed in an ad-hoc manner (for instance dropped from a plane or randomly placed in a field) and then left unattended to perform its monitoring and reporting functions. In a *structured* WSN, all or some of the nodes are deployed in a pre-planned manner, which results in lower network management and maintenance cost. The latter modality has a higher initial cost and it is not always feasible.

In general, WSNs are characterized by the following features [2]: (1) sensor nodes are highly constrained in power, computation, and storage capabilities; (2) sensor nodes have a modest and sometimes non-renewable battery power; (3) in most sensor network applications, the sensed data flow from multiple source sensor nodes to a particular sink (a many-to-one traffic pattern) or to few sinks; (4) sensor nodes are densely deployed in a region of interest and collaborate to accomplish a common sensing task; (5) due to the large number of sensor nodes deployed, it is usually not possible to build a global addressing scheme for a sensor network. Due to those peculiarities, designing resource efficient routing algorithms is challenging.

An effective approach to routing – able to preserve

*Corresponding author. Now at EBTIC (Emirates ICT Innovation Centre), Khalifa University, P.O. Box 127788, Abu Dhabi, United Arab Emirates.

communication resources – is the so called *cluster-based* routing, which is based on sensor nodes grouping [5, 6, 7, 8, 9, 10]. A recent survey can be found in [11]. In such routing algorithms, nodes are arranged into clusters. In each cluster, the task of communicating to the base station is delegated to a single node, called Cluster Head (CH). The CH role is taken in turn by different nodes. The CH aggregates the information collected from the nodes in its cluster and forwards it to the base station. CHs can also form multi-hop data transmissions, thus, after the head of each cluster has been determined, the routing problem of the WSN is reduced to a routing problem among CHs.

If the role of CH is fairly distributed among nodes, this technique brings many advantages: (1) it reduces the total transmission power; (2) it can reduce redundancy by aggregating data; (3) it balances power exhaustion among nodes; (4) it makes bandwidth utilization more efficient; (5) it increases the manageability and scalability of the network and (6) it reduces routing and topology maintenance overhead.

Cluster-based routing algorithms consist of a cluster formation phase, followed by the steady-state operation, subdivided into data aggregation and data transmission phases. In the present work, we focus on the cluster formation phase.

Most of the previous research work on cluster-based routing algorithms of WSNs (it is the case of LEACH [5] and HEED [6]) assumes full collaboration among sensor nodes. However, this assumption does not hold for WSNs with more than one governing authority, where sensor nodes may be owned and supported by different stakeholders or agencies. In that case, sensor nodes can be viewed as selfish players: their primary goal is to maximize the benefits to their agency and save their own resources, such as power, even at the expenses of the other nodes. In such conditions, guaranteeing fair spatial dispersion, rotating CHs, and having balanced cluster formation becomes a challenge [12]. In such settings, a different method for determining CHs is required.

A framework for modeling such interdependent and possibly interactive decision landscape is offered by Game Theory [13]. Game theory (GT) is used to model systems that consist of selfish *agents*, or *players*, with non-aligned interests and preferences, where each player, not only is determined to choose the optimal move or strategy to attain its goals, but (if endowed with the capability of strategic reasoning) may also be aware that the others will do so and take decisions consequently. The area of GT which models fully rational players adopting strategic reasoning is commonly called Classical GT as opposed to other areas where the player is endowed with lesser forms of rationality.

A special area of GT, assuming limited rationality is Evolutionary Game Theory (EGT) [14, 15, 16, 17, 18]. There, each player is assumed not to use strategic reasoning, but just to learn from recent experience and adapt its strategy consequently, pursuing a short-term selfish goal. As a result, sub-populations of interacting players evolve their behaviors: the percentage of players adopting any specific strategy evolves with time. In some cases, the percentages of the different strategies in a population can become stable. This situation is referred to as an *evolutionary equilibrium* or Evolutionary Stable Strategy (ESS) profile. In designing a protocol, a designer normally aims at setting up rules, such that the ESS outcome is a desirable state of the system. The advantages of EGT models is that they are more realistic than those postulating full rationality: they count upon the use of little computation by the player and assume decisions are made based only on local information. In a WSN setting, EGT requires each sensor node to simply observe its neighbor nodes' behavior and to change its own strategies based on those observations. The outcomes of the collective behaviors do not rely on any cooperation enforcement mechanism, but rather on the benefits obtained by individual sensor nodes.

Classical GT models [19] and EGT models [7, 20, 21] are becoming a reference tool for modeling of WSN protocols (for a survey see [22]). Based on an EGT model, in this work, we introduce a novel evolutionary Cluster-Head determination algorithm for sensor networks, called GREET (evolutionary Game theoRy based Energy EfficienT cluster-head determination algorithm). The algorithm takes into account both the positive incentives to the node – resulting from successfully spending its own resources for communicating its own information – and the negative incentives – consisting in spending power for routing other players' messages.

A common tool adopted in EGT for analytically modeling the population evolution is the so-called *replicator dynamics* [17]. This model of the dynamics posits that – from a round of interactions to the next – the proportion of sensor nodes using a certain strategy increases proportionally to the relative advantage provided by the strategy w.r.t. the average payoff of the population. Replicator dynamics is one of the stylized models that use the so-called population approximation. This approximation assumes that the ensemble of individuals participating to the process is so well-mixed, that any individual has the same probability of meeting any other individual. The population approximation does not take into account the fact that individuals are bound by physical constraints to specific regions of space and interact only with neighbors; however, it can provide a "first-order" analytic model of the phenomena, especially good if the individuals are characterized by a high mobility. The addition of physical space to the picture is most of the time achieved by giving up the analytic tractability and passing to simulative models.

In this work, by modeling a system of sensor nodes, first through replicator dynamics, then by a more realistic WSN simulation, we show that GREET guides the overall state of the system towards one of the desired behaviors, which can be set by the network designer. Simulation results show that GREET extends network lifetime and yields better packet throughput when compared to other cluster-based algorithms such as LEACH [5] and CROSS [7].

In order to challenge the capabilities of the algorithm, we consider – both in the analytic model and in the simulation – the worst case scenario, where each node belongs to a distinct agency. As a complement, in the Appendix, we model analytically the case where more sensors can belong the same agency. In that case, the selfish player is the agency: nodes from the same agency behave collaboratively, forming a coalition whenever they happen to be in the same cluster.

The remainder of this paper is organized as follows. Section 2 reviews the related works both for collaborative and non-collaborative sensor nodes. The description of the network model and the evolutionary cluster game are presented in Section 3. In Section 4 we analyze the replicator dynamics model for the developed evolutionary cluster game and find its evolutionary stable strategies. A description of GREET is presented in Section 5. The simulation setup and the results are presented in Section 6. Conclusions are drawn in Section 7. The Appendix describes the case where the number of agencies is lower than the number of sensors.

2. Related Work

Clustering in sensor networks is an active research area [11]. While the ultimate objective behind all the algorithms is to extend the network lifetime and enhance network performance, each algorithm focuses on improving the clustering attributes in a specific phase. Here, focusing on the cluster formation phase, we review representative cluster routing algorithms for WSNs, some not using GT, and some based on GT.

LEACH (Low Energy Adaptive Clustering Hierarchy) [5] is one of the first hierarchical routing protocols used for WSNs. It performs self-organizing and re-clustering functions for every round. A CH is randomly self-elected and rotated in a probabilistic way. It incorporates data fusion into the routing protocol in order to avoid sending redundant information to the base station. However, in LEACH only a single hop cluster is formed that might lead to a large number of clusters and it is assumed that all CHs should directly transmit data to the sink. The protocol does not address the problem of the optimization in the CH selection. In order to overcome this issue, the LEACH-C (LEACH-Centralized) algorithm [23] allows the base station to determine the CH and to notify the information of head node selection to every node in a network.

The PEGASIS (Power-Efficient Gathering in Sensor Information System) [24] protocol was developed to provide improvements over LEACH. PEGASIS builds chains of nodes instead of clusters to address the overhead caused by the cluster formation in LEACH. A greedy algorithm is used to perform the construction of the chain: nodes select their closest neighbors as next hops node. Each node keeps track of its previous and next neighbors in the chain. It uses the assumption that nodes have a partial global knowledge of the network: construction starts from the nodes that are farthest from the sink. During the communication along the chain, each node aggregates data from its neighbors so that eventually all the data are aggregated at one of the sensor nodes, called chain leader. In this protocol the role of chain leader is taken by a single node, hence bottlenecks may be an issue.

In general, the works over the protocols LEACH [5], HEED [6], LEACH-C [23] , PEGASIS [24], BCDCP [25], WCA [26] and BCSP [27] assumed that sensor nodes are fully collaborating (i.e., sensor nodes are acting honestly toward cluster routing algorithm). This is not always the case in practical settings. Selfishness is one of the key problems that confront developers of cooperative distributed systems: selfish behaviors are performed by participants that benefit from the system without contributing their fair share to it It has the potential to severely degrade system performance in several scenarios [28]. The problem of building selfish-resilient systems is often approached using Game Theory [29, 30, 31].

Also in WSNs the performance of cluster routing algorithms can be highly affected by selfish nodes, as we demonstrated in a previous study [12]: the network lifetime and packet throughput decrease, the Quality of Service (QoS) deteriorates, and so on. Also in the design of WSN cluster routing protocols Game Theory has been extensively used [7, 19, 32, 33, 34].

In [33] the authors proposed ACHGT (Adaptive Clustering Hierarchy based on Game-theoretic Techniques), a Game Theory based algorithm where individual sensor nodes are modeled as players, but the CH selection and the number of clusters are mediated by the base station. The selection is based on the information of location and residual power of every node. The authors show that this approach is more efficient than a random one followed by LEACH. However, no theoretical analysis is provided and the ultimate decisions are centralized: this requires excessive overhead and energy costs.

Another game theoretic approach for CH selection is proposed in [34] based on distance and power as parameters. A *fuzzy* clustering approach is applied to find initial clusters along with the member nodes. The Euclidean distance from the base station to each static node is calculated followed by the estimate of power consumption for sending and receiving messages. For every reference time interval, the total cost to benefit ration of different nodes in each cluster is recomputed to grant that the system is game-theoretically stable with those elected CHs or to trigger a new CH selection phase. In terms of network lifetime and optimal CHs selection, this approach is better than the one of the LEACH and HEEDs algorithms. In this approach, GT is not used in the clustering phase.

Few works used Game Theory to model the clustering phase. Among them are those that proposed the protocols CROSS [7], CORE [9], LGCA [10] and the works [32, 35, 36]. Before describing shortly the most relevant ones, we recall that models of competition and collaboration by selfish agents endowed with non-aligned preferences are called Social Dilemmas. There are three qualitative classes of social dilemmas involving symmetric players. Those are players who have the availability of the same set of strategies – conventionally termed Collaborate (C) and Defect (D) – and similar costs and benefits, assigned to their joint strategy choices [13]. We identify the strategy Collaborate with the action of volunteering/bidding as CH for a period, and the strategy Defect as not bidding.

The three classes are conventionally called Prisoner Dilemma, Stag Hunt and Snow Drift. They can be characterized based on two criteria that, for the sake of simplicity, we express for the case of two-player games, where there can be only 4 possible outcomes: CC,DC,DC,DD. The first criterion is whether the outcome CD (the first player volunteers as CH, the second does not) is more beneficial to the first player than mutual defection DD: if this is the case, the class is Snow Drift; if not, it is one of the other two games. The second criterion is whether the outcome DC (the first player does not volunteer, the second does) is more beneficial to the first player than mutual collaboration CC: if this is the case the game is a Prisoner Dilemma, otherwise, it is a Stag Hunt.

In the model proposed in this paper, we considered that, if everyone else played Defect, is much better for a node to have chosen Collaborate and be the CH for that period, rather than to have chosen Defect. In the former case, its messages reach the BS, in the latter the node saves power, but it does not accomplish its mission. This feature situates our game model in the Snow Drift class.

In Classical Game Theory the equilibrium state (a joint behavior determined by the spontaneous individual choice of the fully rational players) is achieved in the Snow Drift game by a suitable randomization of the strategies.

This is the model adopted in CROSS (Clustered Routing for Selfish Sensors) [7], where a clustering mechanism, is proposed based on a Snow Drift game. CROSS achieves also a relatively uniform power consumption distribution by reselecting CHs in every round, thus it extends the network life-time w.r.t. LEACH. Nonetheless, the mechanisms of CROSS are based on some ideal assumptions, among them, that all nodes are simultaneously playing the game. This makes the game too large and inefficient.

In [10] a Localized Game-theoretical Clustering Algorithm (LGCA) for WSNs is proposed, so as to overcome the shortcomings of CROSS. In LGCA each node selfishly plays a localized clustering game only with its neighbors within a communication radius R_c to select the potential CHs. During this time it is likely that more than two nodes in close proximity happen to be selected as potential CHs. Then the contention procedure based on CSMA/CA mechanism is carried out to announce only one real CH in one region. The network lifetime under LGCA is better when compared with CROSS and LEACH. However, both under CROSS and LGCA, since the CH selection is random and does not take into account residual power, one might select as CH a node having low residual power and eventually discontinue network operation.

A repeated game theoretical approach with limited punishment mechanism for clustering in mobile ad hoc networks is proposed in [32], in order to prevent a node from playing non-honest strategies by reporting deceitful self-description values. By doing so, all nodes will act honestly and further guarantee correctness and fairness of CH selection. In this model, each node is assigned a randomization probability taking into account residual power, average moving speed and connectivity with its neighbors.

A GT based clustering algorithm called COst and REward based clustering (CORE) is proposed in [9]: it integrates a virtual reward to the CROSS clustering game to improve its performance. To support energy preservation as well as to extend network lifetime, the cost is coupled with the residual power, so that sensor nodes with high residual power experience a low cost for being CH and sensor nodes with low residual power experience a high cost for being CH. The mechanism of CORE is only applicable under the ideal assumptions that all deployed sensor nodes have large enough communication radius to play in the same game simultaneously. Such assumption is particularly inefficient when nodes are scattered in a large area.

All the works [7, 9, 10, 32, 34, 35, 36] adopted assumptions from Classical GT: complete knowledge about the game, complete information of opponents' type, and full rationality of players. However, such assumptions are unrealistic in a WSN setting [20]. In the present work, we overcome this issue by using EGT [20, 21, 32] as a sounder model for ensembles of selfish sensor nodes.

We assume each node updates its own strategies based on the benefit obtained in the latest rounds of interactions with peers. In many multi-agent systems of practical relevance, this is expected to bring the population of sensor nodes towards an Evolutionary Stable Strategy profile, or ESS. If all members of the population adopt the ESS, then the statistical distribution of the strategies does not change. In other words, no mutant strategy could invade the population. The approach taken by the present work towards nodes' selfishness consists of designing an evolutionary mechanism that brings them to collaborate, so as to improve the global network performance.

As mentioned, the game discussed in the present paper is an instance of the Evolutionary Snow Drift game [17]. More specifically, we develop along the lines of [37, 38] a generalization of the standard version of the game. In all the standard evolutionary games one assumes that the interactions take place between *pairs* of players (randomly drawn from a very large and well-mixed population, so that the probability of meeting a player of a given kind matches the proportion of that kind in the population). In the present work case, we consider interactions of sets of M players at time (each player being randomly and independently drawn from the population): the M players represent, in a stylized way, the members of a cluster.

This model is an instance of an Evolutionary Snowdrift game with M-person interactions [37]. To the best of our knowledge, no other work on the application of this game

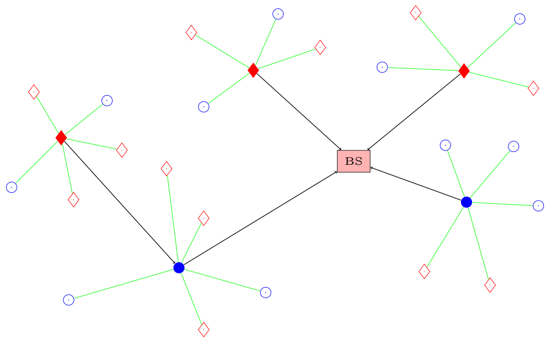


Figure 1: *Common Base Station scenario for sensor nodes belonging to two authorities. Circles: first authority. Diamonds: second authority. Solid circles and diamonds: Cluster Heads. Solid lines: intra-authority communication. Dashed lines: inter-authority communication*

to Cluster Head determination in WSN is available. We study the game first in terms of analytic models depicting a large and well mixed population (a mean-field model) using the two-player interaction, then the many-player interaction. Those scenarios are not realistic and would be appropriate only in the limit of an ideal very-high mobility scenario. Therefore, we pass subsequently to structured population assumptions by studying the evolution of the WSN after deployment within a physical space. In a structured population setting it is not granted that all the desirable properties of the well-mixed population model are preserved [39, 40]. We develop a detailed algorithm for CH determination and study its effectiveness by simulation in a structured population setting: we check that the system converges to equilibrium. Finally, we quantify the overall efficiency of the algorithm by using standard networking performance metrics.

3. Network Model and Evolutionary Cluster Game

3.1. Network Model

We consider a model of a static WSN consisting in N sensor nodes that are randomly distributed over a region and adopt the following simplifying assumptions: all nodes have the same initial battery power; the BS is not power constrained; sensor nodes are self-configured in that, they dynamically adjust the radio power according to the communication distance to the destination; the communication channel is an ideal channel without packet loss (the latter occurs only due to misbehavior of sensor nodes); two sensor nodes are able to communicate with one another if they reside within one's another transmission range; interoperability is ensured by the device manufacturers.

An schematic illustration of this setting, with two authorities, is shown in Figure 1. Hereafter, we consider the worst-case setting in which each node represents a different authority.

3.2. Evolutionary Cluster Game Definition

We model the system using Evolutionary Game Theory (EGT) [17]. EGT has a dynamic nature and can model the adaptation of players, that change their strategies by reacting to simple observation.

We define an Evolutionary Cluster Head game as a triplet $\mathcal{G} \equiv \{\mathcal{N}, \mathcal{S}, \mathcal{U}\}$. The players of the game are selfish sensor nodes, denoted by $\mathcal{N} = \{1, 2, 3, \ldots, N\}$. Let us assume the actions of the nodes are organized in *rounds*. Each sensor node player, at each round, can adopt a strategy from the strategy space $\mathcal{S} = \{s_C, s_D\} \equiv \{C, D\}$, where C corresponds to bidding, i.e. announcing oneself as a candidate CH, and D corresponds to not bidding. At a given time a node adopts either strategy C or strategy D. For the sake of brevity, hereafter, those adopting strategy s_C are called *C-nodes*, those adopting strategy s_D are called *D-nodes*.

$\mathcal{U}$ represents the *utilities or payoffs* to the sensor nodes. The payoff to a node depends on its strategy and on the strategies of the other nodes it interacts with. For instance, if the data sensed by a node and then sent out are successfully relayed by the other nodes to the BS, the source node experiences a positive payoff. The utilities $\mathcal{U}$ relevant to our model are specified in Section 4.

3.3. Evolution Dynamics

The nodes try different strategies in each round and learn from the interactions with other nodes. During this process, the percentage of sensor nodes using a certain strategy may change. The objective, in studying the game, is to predict the dynamics of the system in terms of relative proportions of strategies [17]. We wish to set up the system so that the evolution, in terms of strategy proportions, tends asymptotically towards a desirable behavior.

To obtain a preliminary model the evolution of the system and use it as a hint of the much complex behavior of the real-world systems, it is customary to use *replicator dynamics*, a simplified representation of the evolution mechanisms, where interactions are assumed to take place between any pairs of players, with no regard to actual locations nor to the restrictions they would imply [17, 38].

Given an evolutionary game, players use pure strategies taken from a set $\mathcal{S} = \{s_1, \ldots, s_i, \ldots, s_{|\mathcal{S}|}\}$. Let us denote by n_i the number of sensor nodes that, during a given round, use strategy s_i then $x_i = n_i/N$ is the portion of population playing strategy s_i. The overall state of the population is described by the vector $x = (x_1, \ldots, x_i, \ldots, x_{|\mathcal{S}|})$.

Let us indicate by $U_{s_i}(x)$ the average payoff to a player resulting from the adoption of a specific strategy s_i, when the overall population state is x (the average is taken over all the players adopting strategy s_i), and let $U_{avg}(x)$ the average payoff to a player computed over the whole population when in state x.

The core of replicator dynamics is the following assumption. If adopting a strategy represents an advantage, in terms of payoff $U_{s_i}(x)$, with respect to the average payoff $U_{avg}(x)$ of the whole population, then the players will tend to adopt that strategy: as a consequence its share will increase in the next round (the converse also holds). Going to the continuous time limit, we obtain the following evolution equation, where x'_i represents the time derivative of the proportion of the population adopting strategy s_i

$$x'_i = x_i \Big(U_{s_i}(x) - U_{avg}(x) \Big) \tag{1}$$

The above equation is called *Replicator Dynamics Equation* [17]. It expresses the fact that at time t the proportion of sensor nodes using strategy s_i increases (decreases) when their payoff is larger (smaller) than the average payoff in the sensor nodes population.

An Evolutionary Stable Strategy (ESS) is a fixed point of the evolution (i.e. at ESS $x'_i = 0$, for all i) at which also some stability conditions (expressed in terms of second derivatives, specified below) are fulfilled.

4. Analytic models of the dynamics

In this Section, we address first the problem of finding the Steady State of the Replicator Dynamics equation for our two-strategy game. This corresponds to studying the problem within a *population dynamics* approximation (where any player is assumed to interact with the same probability with any one other player in the large population). Then, we refine the analysis using a slightly more accurate (though still approximate) model, which takes into account that, in practice, players interact with a small set of neighbors. To this purpose, we use a (finite) many-player game. The simulative study in Section (6) provides the final validation of the approach.

4.1. Replicator Dynamics for the two-strategy game

Since the strategy space of our Evolutionary Cluster Game is $\mathcal{S} = \{C, D\}$, the proportions in the population are indicated by x_C and x_D, with the normalization condition $x_C + x_D = 1$. The overall state of the population is described by the vector $x = (x_C, x_D)$.

The replicator dynamics equations are

$$\begin{aligned} x'_C &= x_C \Big(U_C(x) - U_{avg}(x) \Big) \\ x'_D &= x_D \Big(U_D(x) - U_{avg}(x) \Big) \end{aligned}$$

However – thanks to the normalization condition – the whole dynamics can be expressed in terms of a single degree of freedom $x_C = (1 - x_D)$ as

$$x'_C = x_C \Big(U_C(x_C) - U_{avg}(x_C) \Big) \tag{2}$$

The normalization condition grants that other equation will be given by $x'_D = -x'_C$.

4.2. Two-player interaction

Now we formalize our assumptions about the utilities U_C and U_D. We assume that they are determined by the two parameters, b and c: b is the benefit experienced by a node when its data reach the BS, and c is the costs in which a node incurs when sending data to the BS. That cost is always lower than the benefit coming from a successful message delivery: $0 < c < b < 1$. The payoff matrix for 2-player interactions, from the point of view of a player, is

$$A = \begin{bmatrix} A_{CC} & A_{CD} \\ A_{DC} & A_{DD} \end{bmatrix} = \begin{bmatrix} (b - \frac{c}{2}) & (b - c) \\ b & 0 \end{bmatrix} \tag{3}$$

The value of the matrix element A_{DD} expresses the fact that, if both nodes decide to play D, then no data is relayed and each player gets a payoff 0. The value of A_{CC} tells that if both nodes play C (and then choose at random the CH) each of them obtains an expected payoff $(b - c/2)$. The off-diagonal elements tell that if a node decides to play C when the other node chooses to play D, then the latter succeeds in sending its data to the BS at no cost and gets a payoff b, while the former gets a payoff $(b - c)$.

Thus, when a C-node meets another C-node, which happens with probability x_C, it obtains a payoff $a_{CC} = (b - c/2)$, whereas if it meets a D-node, which happens with probability x_D, it obtains $a_{CD} = (b - c)$. The expected value is the following utility

$$U_C(x_C) = (1 - x_C)(b - c) + x_C \left(b - \frac{c}{2} \right)$$

Similarly, the expected payoff for a D-node is

$$U_D(x_C) = x_C b$$

Using the average payoff for the whole the population

$$x_C\, U(s_C, x) + x_D\, U(s_D, x)$$

the replicator dynamics equations can be developed as

$$\begin{aligned} x'_C &= x_C \Big(U_C(x_C) - U_{avg}(x_C) \Big) \\ &= x_C \Big(U_C(x) - x_C U_C(x_C) - x_D U_D(x_C) \Big) \\ &= x_C \Big((1 - x_C) U_C(x_C) - x_D U_D(x_C) \Big) \\ &= x_C (1 - x_C) \Big(U_C(x_C) - U_D(x_C) \Big) \end{aligned} \tag{4}$$

This first derivative vanishes at three points, that we denote by $x_C^{(0)}$, $x_C^{(1)}$ and $x_C^{(*)}$. The first two are trivially $x_C^{(0)} = 0$ (no one plays C) and $x_C^{(1)} = 1$ (everyone plays C); the reminder corresponds to

$$(U_C(x_C) - U_D(x_C)) = 0 \tag{5}$$

or, explicitly,

$$(1 - x_C)(b - c) - \frac{c}{2} x_C = 0 \tag{6}$$

which yields

$$x_C^{(*)} = \frac{2(b - c)}{2b - c} \tag{7}$$

The stability analysis can be performed by differentiating (4) to get the second derivative x''_C: a positive value of the second derivative corresponds to stable points. Substituting each of the three points one can check that $x_C^{(0)}$ and $x_C^{(1)}$ are *unstable* equilibrium points. The point $x_C^{(*)}$, on the contrary, is stable. Thus, in the population approximation, $x_C^{(*)}$ is an *ESS* point.

4.3. Many-player interaction

In practice, the nodes of a sensor network do not interact pairwise, but rather group-wise, with a restricted number of neighbors. Thus, a more accurate game model is be defined by the collection of a number of independent sensor node games, each involving only a finite number M of nodes. In an M-node game, a sensor node competes against the other $M-1$ sensor nodes (the complement set), and its payoff depends on the number of C-nodes in the complement set. The replication dynamics of such a game has been studied in [41], building on the works in [38, 37]. Here, along the lines of those works, we redevelop those results that apply to our case.

If at least a sensor node, plays the C strategy, then all the nodes in that sensing region will get a benefit b; within them, the C-nodes will incur in a cost. Since only one of the volunteering nodes is chosen at random as CH, that cost in average will be shared equally among the C-nodes: if there is only a C-node, its payoff is $(b-c)$; if there are two C-nodes in the same sensing region, then the payoff is $(b-\frac{c}{2})$, and so on. Let the number of C-nodes, in a cluster of M nodes, be indicated by h, with $h \in \{0, 1, \ldots, M\}$ and let us denote by $A(s_i, h)$ the payoff to a node playing strategy s_i in a cluster containing exactly h C-nodes. Then

$$A(C, k) = b - \frac{c}{h}$$

As to $A(D, h)$, if $h = 0$ the payoff to the node playing D is zero, but as soon as $h > 0$ one has $A(D, h) = b$. The overall payoff matrix $A^{(M)}$ (where the apex indicates the fact that the player set contains M players) is

$$A^{(M)} = \begin{bmatrix} A(C,M) & \cdots & A(C,k) & \cdots & A(C,1) & A(C,0) \\ A(D,M) & \cdots & A(D,k) & \cdots & A(D,1) & A(D,0) \end{bmatrix}$$

$$= \begin{bmatrix} b-\frac{c}{M} & \cdots & b-\frac{c}{h} & \cdots & b-c & \\ b & \cdots & b & \cdots & b & 0 \end{bmatrix} \tag{8}$$

$$\tag{9}$$

The element $A(C, 0)$ does not apply: if the node plays C there is at least a C-node in the cluster. Notice, in passing, that the payoff matrix (3) is a special case of (8): it corresponds to $A^{(M=2)}$.

Now we can compute the average payoffs $U_C^M(x_C)$ and $U_D^M(x_C)$ of each strategy (with the usual meaning of the apex M). Consider the former. If the strategy of a node is fixed to C, then the number k of other C-nodes in the subset of M nodes chosen at random from a population where the fraction of C-nodes is x_C, is a Binomial variable: such variable has probability parameter $p = x_C$ and multiplicity parameter $n = (M-1)$. The expected utility $U_C^M(x_C)$ of the C-node is obtained averaging over k

$$\begin{aligned} U_C^M(x_C) &= \sum_{k=0}^{n} \binom{n}{k} x_C^k (1-x_C)^{n-k} A(C, k+1) \\ &= b - c \sum_{k=0}^{n} \binom{n}{k} x_C^k (1-x_C)^{n-k} \frac{1}{k+1} \end{aligned}$$

As to $U_D^M(x_C)$ one has

$$\begin{aligned} U_D^M(x_C) &= \sum_{k=0}^{n} \binom{n}{k} x_C^k (1-x_C)^{n-k} A(D, k) \\ &= \sum_{k=1}^{n} \binom{n}{k} x_C^k (1-x_C)^{n-k}\, b \\ &= \sum_{k=0}^{n} \binom{n}{k} x_C^k (1-x_C)^{n-k}\, b - \left[\ b(1-x_C)^n\ \right] \\ &= b - \left[\ b(1-x_C)^n\ \right] = b\left[\ 1-(1-x_C)^n\ \right] \end{aligned}$$

The second last simplification comes from the normalization condition of the Binomial. The average utility is $U_{avg}^M(x) = x_C U_C^M(x) + (1-x_C) U_D^M(x)$. Notice that for $M = 2$ the above utilities equal the utilities of the game in the previous subsection.

We have all the elements to write the evolution equation. We have two competing strategies in the population, as in the game of the previous section, therefore, starting from the standard replicator dynamics equation (1), we can obtain an equation of the form

$$x_C' = x_C(1-x_C)(U_C^M(x_C) - U_D^M(x_C)) \tag{10}$$

Also this equation, as equation (4), has three critical points: two, with $x_C^{(0)} = 0$ and $x_C^{(1)} = 1$, both associated to unstable equilibria, and one non-trivial, in $x_C^{(*)}$ such that

$$\left(\ U_C^M(x_C) - U_D^M(x_C)\ \right) = 0 \tag{11}$$

Also in this case, by appropriate stability analysis [38] one can check that this is a stable equilibrium. This equation can be expanded as

$$b x_C - b + c \sum_{k=0}^{n} \binom{n}{k} x_C^k (1-x_C)^{n-k} \frac{1}{k+1} = 0 \tag{12}$$

One can check that for $M = 2$ (i.e. $n = 1$) one recovers the same solution of the previous section (equation (7)).

Since the behavior depends not on the absolute values of the costs and benefits, but on their relative value, we can re-write this equation introducing, the parameter $w \equiv c/b$, representing the cost-to-benefit ratio.

$$x_C - 1 + w \sum_{k=0}^{n} \binom{n}{k} x_C^k (1-x_C)^{n-k} \frac{1}{k+1} = 0 \tag{13}$$

The solution $x_C^{(*,M)}$ to this equation (hereafter, for brevity $x_C^{(*)}$) can be worked out analytically for lower degrees n and numerically for higher degrees: Figures 2 and 3 show the dependence of the solution $x_C^{(*)}$ from w and M. The curves in Figure 2 show $x_C^{(*)}$ vs. w for different values of M, while the curves in Figure 3 show $x_C^{(*)}$ vs. N for different values of w. The equilibrium probability $x_C^{(*)}$ decreases with the growth of w and increases with M.

Notice, for later use, that those equilibrium probabilities can be pre-computed as a function of the relevant pa-

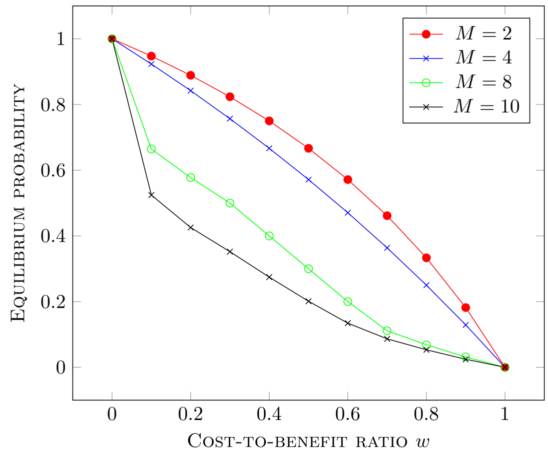


Figure 2: *Equilibrium probability x_C^* (probability of volunteering as candidate Cluster Head) as a function of the cost to benefit ratio w.*

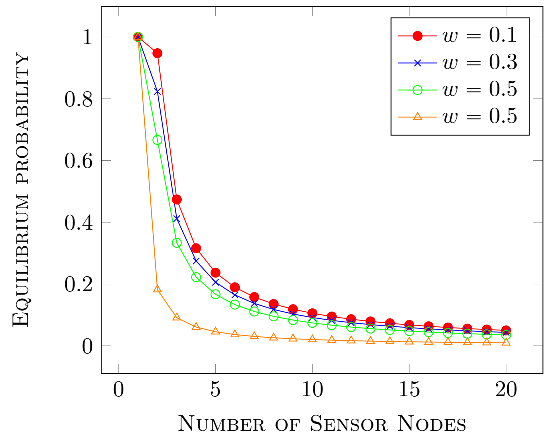


Figure 3: *Equilibrium probability x_C^* (probability of volunteering as candidate Cluster Head) as a function of the number of nodes N.*

rameters (number of neighbors, cost-to-benefit ratio) and efficiently stored in a node memory. Every node will know, given the parameters, what probabilities to adopt keep the overall system on a statistically stable state.

5. The Algorithm

In this Section, we present the proposed Cluster Head determination algorithm, the GREET algorithm, inspired to the evolutionary cluster game studied above. In the next Section, we demonstrate, by simulation, that this algorithm takes the system to an equilibrium, in analogy to what happens with the replicator dynamics.

5.1. Algorithm scope

The algorithm is part of a specific operational phase of the sensor nodes life cycle. Such a life cycle – starting after node placement over the field – is typically structured into a self-organized *initiation* phase followed by a number of cyclic operational phases. The latter phases, in cluster-based algorithms, are the cluster set-up phase and data communication phase, as shown in Fig. 4. The set-up phase is subdivided into candidate CHs bidding, CH determination, and cluster formation. The data-transmission phase consists of data aggregation and transmission.

GREET operates within the CH bidding sub-phase and the CH determination sub-phase. We focus on those two sub-phases. For the remaining sections of the life-cycle, we adopt standard assumptions, which are recalled hereafter. In particular, we do not focus on the details of the initialization phase and give for granted some conditions such as the synchronization (see Section 5.5).

5.2. Initiation Phase

After the deployment of the sensor nodes, which we assume uniform random, the set of sensors has to undergo a self-organization phase [42]. We assume each node can broadcast its data in low-power consumption short-range mode or in high-power consumption long-range mode. The latter is used when the node is CH and sends data to the BS, the former in the other cases. For the sake of simplicity, we assume all the nodes have the same power value for the short-range mode and the same high power value for the long-range mode. In ideal conditions this translates into all the nodes having the same short-range distance R_c and the same long-range distance L_c.

At the end of the self-organizing phase, each node is aware of the neighbor nodes within short-range distance. We denote by $Nh(n_i)$ the number of neighbor nodes of n_i, that the node can sense during a round.

5.3. Set-up Phase: the GREET algorithm

A node can be in one of the following states, w.r.t. the CH determination goal: *CH-enabled*, *CH-disallowed*, *candidate CH*, *non-candidate-CH*, *appointed-CH*, *normal* state and *loner* state. A the beginning of every Set-up phase, all the nodes are in *CH-enabled* state, except for the ones that served as CH in the previous rounds, that are *CH-disallowed*.

5.3.1. Bidding for the CH role

As the set-up phase starts, each *CH-enabled* node can explicitly announce the C strategy (and become *candidate CH*) or tacitly adopt the D strategy (and become *non-candidate CH*).

Those nodes are in a $M = Nh+1$ node game, as the one described in the previous section, which admits an ESS. In the well-mixed population hypothesis, we have seen that whatever the x_C adopted at the start would lead to the stable ESS equilibrium. Here we posit that, since the beginning, the sensor nodes adopt the collaborative strategy with probability $x_C^{(*,M)}$. However, even if this is not the case, the mechanism of learning from interaction with the neighbors drives system towards equilibrium. This finding is confirmed by simulation.

5.3.2. Determining the actual CH

If only one sensor node plays C, that node becomes CH (and passes to the state of *appointed CH*). If two or more sensor nodes play C the node to be designed CH is the one

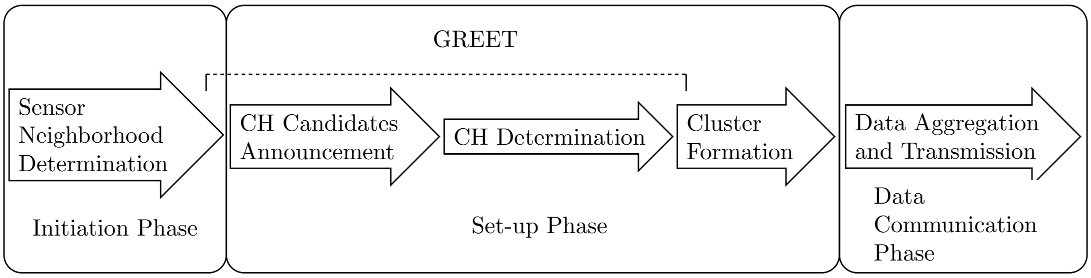


Figure 4: *Phases of the GREET algorithm: we focus on CH Candidates Announcement and CH Determination.*

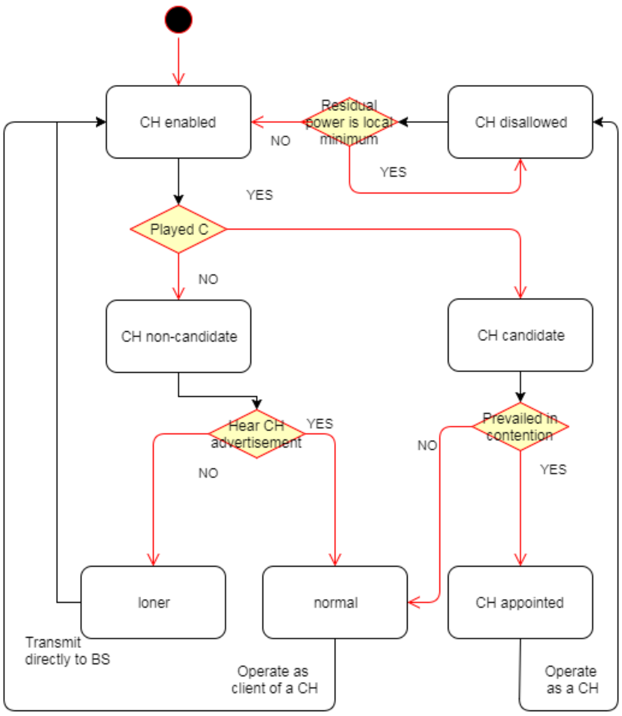


Figure 5: *State diagram for the sensor node.*

with the highest residual power. In hypothetical case of a tie, the contention can be resolved by randomization.

Notice that, since appointed CHs are at least at a distance R_c meters away from each other, the GREET algorithm can reasonably uniformly distribute CHs over the sensor field (provided that, as we assume, the sensors are scattered approximately uniformly over the field).

5.3.3. Cluster Formation

At this point, the appointed CH advertises itself as such. The nodes receiving the advertisement go back to *normal state* and prepare to subscribe to that CH. In case of reception of more than one CH advertisement a node will prepare to subscribe to the one with the higher residual power. Now, normal state nodes and CH disallowed nodes send a message to their CH to subscribe as members. The appointed CH creates a time slot schedule and broadcasts it to all of its candidates, that – from this point on – become its cluster members.

There can be also nodes that did not bid for CH and that did not receive any CH advertisement: those nodes will not be able, during the current round, to route their messages through a CH: we call them *loner* nodes. They will remember this info and use it to play C in the following round. As to the current information to be sent by the loner nodes, any implementation of the algorithm should specify whether they should wait until the next round for trying to route the information or they should send the information to the BS directly. In the simulation below, we adopted the latter option.

At the beginning of the next round, all the nodes in *normal state* will become *CH-enabled*, while those that served as CH will become *CH-disallowed*. A node that served as CH in a round, will return *CH-enabled* only when its residual power is not the minimum of its neighborhood; in the meanwhile it will stay in the *CH-disallowed* state. The corresponding state diagram is shown in Figure 5.

5.4. Data Transmission Phase

After cluster formation, the data transmission phase starts. The non-CH nodes start to send data to their CH according to the assigned time slot in the TDMA schedule.

5.5. Synchronization and scheduling

GREET is an algorithm that can work under different settings and can be supported in different ways by infrastructural services: it is organized in phases and communication rounds, thus it assumes that a synchronization mechanism and a scheduling schema are provided. The choice of those depends on how constraining are the requirements of the application setting. A synthesis of the tradeoffs among the main synchronization schemas is provided in [43]. For the sake of simplicity, in the simulation, we assumed that synchronization is provided by broadcast signals from the BS. The results of the power consumption comparisons in the next section are expected to hold even under different synchronization schemas, provided that the the involved cluster based algorithms use the same schema.

As to scheduling, it depends on the varying degrees of structural organization available in the network during the different sub-phases. In the simulation, we assumed that, after the cluster has been formed, the scheduling can be coordinated by the CH, within a TDMA schema (as in LEACH): this avoids collisions and saves power. In the other less structured sub-phases, nodes should resort to contention-based schemas. As to the CH-to-BS communication, we assumed that scheduling is responsibility of the BS.

Table 1: Simulation Parameters

Size of the region	$L \times L = 100 * 100\ m^2$
Nodes' placement	uniform random
Number of Nodes Deployed	100
Av.distance node from BS	$d_{toBS} = \frac{0.765 \times x_m}{2}$
Av.distance node close CH	$d_{toCH} = \frac{L}{\sqrt{2\,\pi\,a}}$
Number of iterations	2500
Packet Size	$k = 4000\ bits$
Energy to receive a bit	$E_{elec} = 50\ nJ/bit$
Initial Battery Power	$E_0 = 0.5\ J$
Data Fusion Energy cost	$E_{DA} = 5\ nJ/bit/signal$
Free space power loss	$E_{fs} = 10\ pJ/bit/m^2$
Multi path fading power loss	$E_{mp} = 0.0013\ pJ/bit/m^4$
Parameter w	$0 \leq w \leq 1$

6. Simulative Model

A simulation using MatLab has been carried on to compare study the performance of GREET and to compare it to the cluster-based algorithms LEACH, CROSS. The GREET simulation results are in good qualitative agreement with the replicator dynamics results.

6.1. Simulation Setup Parameters

We randomly deployed 100 sensor nodes over a simulation field of $100 \times 100\ m^2$. The BS, is placed at the center of the field, and accessible by all sensor nodes in their high-power long-range transmission mode. Table I shows the parameters of the simulation: in order to ease the comparison, we used, wherever possible, the same values as the LEACH study [23].

6.2. Power Dissipation

The power dissipated for data transmission and reception is modeled using the first order radio model described in [5, 23]. Denoting the distance by d, we assume d^2 free-space power loss when propagation is in the line of sight, and d^4 power loss for long-distance communication, due to multi-path fading. Each sensor node consumes an amount of energy E_T to transmit a k-bits packet over a distance d as shown in equation (14). Let us define the critical distance $d_0 = \sqrt{E_{fs}/E_{amp}}$, where E_{fs} and E_{mp} represent the transmitter energy of free space and multi-path fading, respectively. Let E_{elec} be the power dissipated per bit by the transmitter or receiver circuit. The transmission energy is

$$E_T(k,d) = \begin{cases} k\,(E_{elec} + E_{fs}\,d^2) & d \quad <= \quad d_0 \\ k\,(E_{elec} + E_{mp}\,d^4) & d \quad > \quad d_0 \end{cases} \tag{14}$$

Where k is the number of bits in the packet. To receive a k-bit packet, a sensor consumes an amount of energy E_R

$$E_R = kE_{elec} \tag{15}$$

The total energy, E_{round}, consumed in the network during a round is expressed as follows

$$\begin{aligned} E_{round} &= k\,[2 \times N\,E_{elec} + N\,E_{DA} + \\ &+ \quad E_{mp}\,a\,d_{toBS}^4 + E_{fs}\,N\,d_{toCH}^2] \end{aligned}$$

where a is the number of CHs, N the total number of sensor nodes, E_{DA} is the energy for data aggregation spent by the CHs, d_{toCH} is the average distance between CH and its member nodes and d_{toBS} the average distance between CH and BS. The average energy E_{avg} of the network at the r^{th} round is computed as

$$E_{avg} = \frac{1}{N} E_{total} \left(1 - \frac{r}{r_{max}}\right) \tag{16}$$

where r_{max} denote the number of iterations, and E_{total} is the total energy of the network at initial deployment $E_{total} = \sum_i E_i$, where E_i the initial energy of node i.

6.3. Simulation Results and Discussion

The simulation results confirm the qualitative findings of the analytic model. In the latter, the main parameter was the cost-to-benefit ratio w. In the simulation we selected the representative values $w = \{0.1, 0.2, \ldots, 0.9\}$. According to the results obtained as w tends to one, the equilibrium probability of the candidate CHs decreases: selfish sensor nodes tend to wait for some other nodes to perform the task of sending data on behalf of them. The lower the number of nodes available as a candidate CH, the fewer CHs are present. Thus, each sensor consumes much communication power by directly communicating with the BS. As a result, the performance of GREET deteriorates as w increases. This is also the case for the CROSS algorithm. To evaluate GREET, we used the metrics measuring network lifetime, network residual power, and number of data packets successfully sent to the BS.

We took note of the network lifetime of the node that first runs out of power (First Node Dead time, or FND) and the lifetime of the last node whose power is depleted (Last Node Dead time, or LND). The FND of LEACH, CROSS, and GREET for different values of w is shown in Fig. 6 (left). Notice that in LEACH, by construction the FND is not a function of the parameter w; on the contrary, it depends on w in CROSS and GREET. With respect to this metrics, it is apparent that GREET outperforms both LEACH and CROSS. This happens because, in GREET, the sensor node that served as CH during the previous round is exempted from the current CH determination process until its residual power is not similar to that of its member nodes. The fact that, broadly speaking, all the sensor nodes in an area have to serve as CH before being considered for a further turn, makes GREET better than the other algorithms in terms of FND.

The LND for LEACH, CROSS, and GREET is shown in Fig. 6 (right), for different values of w. Again GREET outperforms both LEACH and CROSS. An interesting observation is the following: GREET achieves a maximum

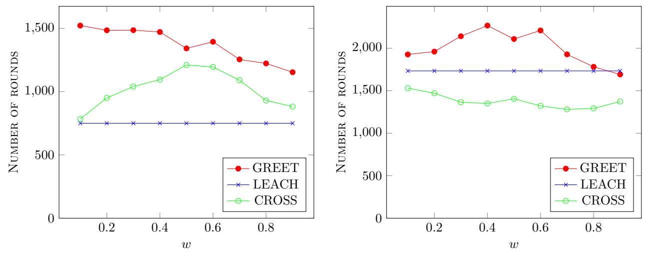


Figure 6: *First Node Dead (left) and Last Node Dead (right) time for different values of the parameter w*

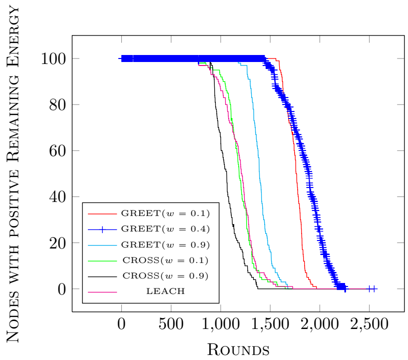


Figure 7: *Number of sensor nodes with positive residual power*

LND for a value around $w = 0.4$, i.e. approximately where the simplified population game of the previous section was reaching an equilibrium state.

For smaller values of w, LND decreases, since the average number of candidate CHs per round is increasing. In other words, the smaller the cost-to-benefit ratio of being a CH, the higher the number of sensor nodes bidding for the CH role, the higher the expected number of appointed CHs per round. Conversely, for larger values of w, LND degrades: fewer sensor nodes are willing to be a candidate CH. In such cases sensor nodes have to communicate directly with the BS, thus reducing their lifetime. All sensor nodes being CHs and no nodes being CHs is the same as direct communication with the BS.

Fig. 7 shows the algorithms' performance in terms of the number of sensor nodes with some residual power. This metrics measures the total number of sensor nodes still alive, which is evaluated at each transmission round of the algorithm. As for the equilibrium probability in the population game, GREET has the highest power saving around $w = 0.4$. Obviously, the power expenditure rate of GREET is more uniform than the one of LEACH and of CROSS, since we have incorporated residual power as a parameter to determine the actual CH. Moreover, the actual CHs are approximately uniformly scattered over the whole area, because every CH is at least R_c meters away from any other. As a consequence, the power of all nodes is used more fairly.

Lastly, as shown on Fig. 8, the average number of data packets sent to the BS has been used for evaluation. Both for the values of $w = 0.4$ and $w = 0.1$, GREET has a higher throughput than LEACH and CROSS. This happens because around the former value, GREET has achieved better network life time, while around the latter

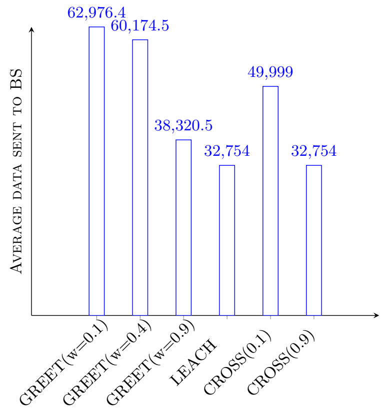


Figure 8: *Average data sent to the BS (number of packets).*

value, it has the highest number of candidate CHs. However, around the value $w = 0.9$, GREET features a lower throughput due to the highest cost of collaboration: selfish nodes do not forward the other nodes' data packets.

7. Discussion and Conclusions

We have introduced GREET, an energy-efficient Cluster Head determination algorithm for selfish nodes in a WSN. The algorithm is inspired to an Evolutionary Game Theory (EGT) approach with M-player interactions.

We point out that the structure of this game is rather general and applies to many contexts in the real world, where the costs of an achievement are shared by the contributors only, while the benefits of the achievement are shared among contributors and non-contributors alike. However, to the best of our knowledge, this is the first work in which the EGT form of the game is applied to Cluster Head determination in WSN and studied by simulation in a realistic context.

As to the EGT model, we adopted the worst case scenario, in which each node belongs to a different (selfish) agency. The EGT model with a finite number of selfish agencies each owning more than one sensor node is discussed in the Appendix. Its simulative study will be the object of a future work.

Furthermore, in the simulation here, we assumed that nodes were endowed with homogeneous characteristics (they would differ only in position and residual power); we plan to extend the study to heterogeneous networks.

Appendix: Players grouping in agencies

Form the point of view of the number of controlling authorities or *agencies*, so far we have assumed the worst case scenario, where each sensor node represents an agency by itself. We can generalize the model to the case where an agency can own more than one sensor node. Here we outline such a generalization in terms of the analytic model. This setting is not covered by the simulation Section 6. The change w.r.t. the previous setting is that if two or more sensor nodes belonging to the same agency meet in a region, they must collaborate with one another, i.e. form a *coalition*, adopting a coordinated strategy behavior. We denote coordinated coalition strategies with a hat sign. A coalition can play $\widehat{C}$ or $\widehat{D}$. Playing $\widehat{D}$ for the coalition means that all the members play D; playing $\widehat{C}$ at coalition level means that only one chosen member plays C and the remainder play D.

Let us indicate by G the number of agencies – each agency being indexed by g – and let N_g the number of sensor nodes of agency g; let the total number of sensor nodes be $N = \sum_g^G N_g$. So far, we had assumed $N_g = 1$, i.e. $G = N$. Now we allow $1 \leq N_g < N$.

Hereafter we compute the payoffs in those scenarios, furthermore, we discuss the impact of the payoff on the replicator equation and on the equilibria. Notice that in the payoff computation we take the point of view of the agency (as, in fact, we already did in Sections 3 and 4, where $N = G$).

An important point is that the dynamics of the system can depend on whether we are assuming that the players are aware of each other's affiliation before announcing their choice (informed choice) or not (uninformed choice). The former case corresponds to the setting where the information about the owning agency is broadcasted with the *Hello* message, which precedes the announcement; the latter case corresponds to a setting where that information is broadcasted at the same time of the announcement.

Hereafter we assume that the nodes broadcast their agency affiliation with the *Hello* message, i.e. we are in the *informed choice* setting. It follows that nodes can adopt some form of inter-agency collaboration before making their choice. A simple collaboration model consists in assuming that sensor nodes have some way of agreeing on a common strategy with the members of their agency (either by predetermined rules or with a round of further information exchange or negotiation). In this case, each coalition can behave as a single ("extended") player.

What counts, thus, is how many extended players, i.e. distinct coalitions meet in a round: we fall back to the case of many-player interactions, discussed in Section 4, except for the fact that the number of players is equal to the number of non-empty coalitions involved in the interaction and that this number fluctuates from one round to the next. A formal statement of the solution strategies follows.

Assume that, at each round, clusters are formed by drawing M players at random from the population. The fact that they are owned by distinct agencies determines a partition of the M-player set: each partition block is a coalition with k_g elements belonging to the same agency g. Let $K = card(\{ g \in \{1, 2, \dots, G\} \mid k_g > 0 \})$ the number of the agencies actually represented in the M interacting player set of a round.

Let us consider the point of view of a coalition g', represented by at least a player. The nontrivial equilibrium strategy for a player belonging to agency g' is the following conditional strategy. At the beginning of the round, find out $k_{g'}$ and K; if $k_{g'} = 1$, i.e. there are no agency-fellows, play C with probability $x_C^{(*,K)}$, solution of equation (13); if $k_{g'} > 1$, i.e. there are agency fellows, agree with them a common strategy, so that the coalition plays $\widehat{C}$ with probability $x_C^{(*,K)}$ (equation (13)).

Acknowledgements

The authors would like to acknowledge the support by the COOPERA program (Region Rhône-Alpes, France) and by the IRIXYS, International Research and Innovation Center in Intelligent Digital Systems.

References

[1] I. F. Akyildiz, I. H. Kasimoglu, Wireless sensor and actor networks: research challenges, Ad hoc networks 2 (4) (2004) 351–36. doi:10.1016/j.adhoc.2004.04.003.

[2] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, Ad hoc networks 3 (3) (2005) 325–349. doi:10.1016/j.adhoc.2003.09.010.

[3] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Computer networks 52 (12) (2008) 2292–2330. doi:10.1016/j.comnet.2008.04.002.

[4] A. J. Swati, R. Priyanka, Wireless sensor network (wsn): Architectural design issues and challenges, International Journal on Computer Science and Engineering 02 (09) (2010) 3089–3094.

[5] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on, IEEE, 2000, pp. 10–pp.

[6] O. Younis, S. Fahmy, Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Transactions on mobile computing 3 (4) (2004) 366–379.

[7] G. Koltsidas, F.-N. Pavlidou, A game theoretical approach to clustering of ad-hoc and sensor networks, Telecommunication Systems 47 (1-2) (2011) 81–93. doi:10.1007/s11235-010-9303-5.

[8] S. Naeimi, H. Ghafghazi, C.-O. Chow, H. Ishii, A survey on the taxonomy of cluster-based routing protocols for homogeneous wireless sensor networks, Sensors 12 (6) (2012) 7350–7409. doi:10.3390/s120607350.

[9] S. Salim, X. Li, S. Moh, Cost-and reward-based clustering for wireless sensor networks: A performance tradeoff, in: 22nd Wireless and Optical Communication Conference (WOCC), 2013, IEEE, 2013, pp. 421–425.

[10] D. Xie, Q. Sun, Q. Zhou, Y. Qiu, X. Yuan, An efficient clustering protocol for wireless sensor networks based on localized game theoretical approach, International Journal of Distributed Sensor Networks 2013. doi:10.1155/2013/476313.

[11] S. P. Singh, S. Sharma, A survey on cluster based routing protocols in wireless sensor networks, Procedia Computer Science 45 (2015) 687 – 695, international Conference on Advanced Computing Technologies and Applications (ICACTA). doi:http://dx.doi.org/10.1016/j.procs.2015.03.133.

[12] K. A. Gemeda, G. Gianini, M. Libsie, The effect of node selfishness on the performance of wsn cluster-based routing algorithms, in: AFRICON, 2015, IEEE, 2015, pp. 1–5.

[13] G. Owen, Game theory. 1995.

[14] J. M. Smith, Evolution and the theory of games, in: Did Darwin Get It Right?, Springer, 1988, pp. 202–215.

[15] R. M. Axelrod, The evolution of cooperation: revised edition, Basic books, 2006.

[16] J. W. Weibull, Evolutionary game theory, MIT press, 1997.

[17] J. Hofbauer, K. Sigmund, Evolutionary games and population dynamics, Cambridge university press, 1998.

[18] M. A. Nowak, Evolutionary dynamics: Exploring the equations of life. 2006, Massachusetts: Belknap Press Google Scholar.
[19] H.-Y. Shi, W.-L. Wang, N.-M. Kwok, S.-Y. Chen, Game theory for wireless sensor networks: a survey, Sensors 12 (7) (2012) 9055–9097. doi:10.3390/s120709055.
[20] G. V. Crosby, N. Pissinou, Evolution of cooperation in multi-class wireless sensor networks, in: Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on, IEEE, 2007, pp. 489–495. doi:10.1109/LCN.2007.149.
[21] A. Rogers, E. David, N. R. Jennings, Selfish sensors in wireless micro-sensor networks, in: Conf. on Autonomous Agents and Multiagent Systems-Volume 3, IEEE Computer Society, 2004, pp. 1322–1323.
[22] H. Bhattacharya, S. Chattopadhyay, M. Chattopadhyay, Mathematical analysis of wsns using game theory: A survey, Soft Computing Applications in Sensor Networks (2016) 123.
[23] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, Wireless Communications, IEEE Transactions on 1 (4) (2002) 660–670. doi:10.1109/TWC.2002.804190.
[24] S. Lindsey, C. S. Raghavendra, Pegasis: Power-efficient gathering in sensor information systems, in: Aerospace conference proceedings, 2002, Vol. 3, IEEE, 2002, pp. 3–1125.
[25] S. D. Muruganathan, D. C. Ma, R. I. Bhasin, A. O. Fapojuwo, A centralized energy-efficient routing protocol for wireless sensor networks, IEEE Communications Mag. 43 (3) (2005) S8–13.
[26] M. Chatterjee, S. K. Das, D. Turgut, Wca: A weighted clustering algorithm for mobile ad hoc networks, Cluster computing 5 (2) (2002) 193–204.
[27] G. Lee, M. Lee, W. Seok, J. Kong, O. Byeon, A base station centralized simple clustering protocol for sensor networks, in: Embedded and Ubiquitous Computing, Springer, 2006, pp. 682–691.
[28] G. L. Cota, S. B. Mokhtar, G. Gianini, E. Damiani, J. Lawall, G. Muller, L. Brunie, Analysing selfishness flooding with seine, in: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2017, pp. 603–614. doi:10.1109/DSN.2017.51.
[29] G. Gianini, E. Damiani, T. R. Mayer, D. Coquil, H. Kosch, L. Brunie, Many-player inspection games in networked environments, in: Digital Ecosystems and Technologies (DEST), 2013 7th IEEE International Conference on, IEEE, 2013, pp. 1–6.
[30] G. L. Cota, S. B. Mokhtar, J. Lawall, G. Muller, G. Gianini, E. Damiani, L. Brunie, A framework for the design configuration of accountable selfish-resilient peer-to-peer systems, in: Reliable Distributed Systems (SRDS), 2015 IEEE 34th Symposium on, IEEE, 2015, pp. 276–285.
[31] G. L. Cota, S. B. Mokhtar, G. Gianini, E. Damiani, J. Lawall, G. Muller, L. Brunie, Racoon++: A semi-automatic framework for the selfishness-aware design of cooperative systems, IEEE Transactions on Dependable and Secure Computing PP (99) (2017) 1–1. doi:10.1109/TDSC.2017.2706286.
[32] L.-h. Sun, H. Sun, B.-q. Yang, G.-j. Xu, A repeated game theoretical approach for clustering in mobile ad hoc networks, in: IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC),2011, IEEE, 2011, pp. 1–6.
[33] Z. Zeng-wei, W. Zhao-hui, L. Huai-zhong, Clustering routing algorithm using game-theoretic techniques for wsns, in: Proceedings of the International Symposium on Circuits and Systems (ISCAS), 2004, Vol. 4, IEEE, 2004, pp. IV–904.
[34] S. Dasgupta, P. Dutta, A novel game theoretic approach for cluster head selection in wsn, International journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN 2 (3) (2013) 2278–3075.
[35] W.-D. Wang, Q.-X. Zhu, Hierarchical clustering algorithm and cooperation analysis for wireless sensor networks, Ruan Jian Xue Bao(Journal of Software) 17 (5) (2006) 1157–1167. doi:10.1360/jos171157.
[36] Z. Xu, Y. Yin, J. Wang, An density-based energy-efficient routing algorithm in wireless sensor networks using game theory, International Journal of Future Generation Communication and Networking 5 (4) (2012) 99–112.
[37] D.-F. Zheng, H. Yin, C.-H. Chan, P. Hui, Cooperative behavior in a model of evolutionary snowdrift games with n-person interactions, Europhysics Letters(EPL) 80 (1) (2007) 18002.
[38] M. O. Souza, J. M. Pacheco, F. C. Santos, Evolution of cooperation under n-person snowdrift games, Journal of Theoretical Biology 260 (4) (2009) 581–588.
[39] C. Hauert, M. Doebeli, Spatial structure often inhibits the evolution of cooperation in the snowdrift game, Nature 428 (6983) (2004) 643.
[40] G. Szabó, G. Fath, Evolutionary games on graphs, Physics reports 446 (4) (2007) 97–216. doi:10.1016/j.physrep.2007.04.004.
[41] B. Wang, K. R. Liu, T. C. Clancy, Evolutionary cooperative spectrum sensing game: how to collaborate?, IEEE Transactions on Communications 58 (3) (2010) 890–900.
[42] L. Clare, G. Pottie, J. Agre, Self-organizing distributed sensor networks, in: Proc. SPIE Conf. Unattended Ground Sensor Technologies and Applications, Vol. 3713, 1999, pp. 229–237.
[43] F. Sivrikaya, B. Yener, Time synchronization in sensor networks: a survey, IEEE Network 18 (4) (2004) 45–50. doi:10.1109/MNET.2004.1316761.

ACCEPTED MANUSCRIPT

{Ketema Adere Gemeda} received his BSc. degree in Computer Science and IT from the ArbaMinch University, Ethiopia, and M.Tech. Degree in Computer Science and Engineering from the IIIT-Hyderabad, India in 2007 and 2010 respectively. He has been working as a Graduate Assistant from 2007-2008 Computer Science Department of Adama Science and Technology University; he has also been Lecturer from 2010-2012 in the same university. He is currently a PhD candidate at Addis Ababa University in collaboration of INSA Lyon, France. His research interests include efficient routing algorithm design for ad-hoc wireless and sensor networks.

{Gabriele Gianini} (MSc Physics 1992, Ph.D Physics 1996), since 2005 has been is Assistant Professor at the Department of Computer Science of the Universita degli Studi di Milano (Italy).

He has held visiting positions at a number of international institutions,

including INSA de Lyon (France), University of Passau (Germany), CERN (Geneva, Switzerland), Fermilab (Chicago, USA), CBPF (Brasil). Since 2016 he holds a Senior Researcher Fellowship at the EBTIC Laboratory/ Khalifa University of Abu Dhabi (UAE). From 2005 to 2012 he has been Adjoint professor at the Free University of Bolzano (Italy). Among his research interests: statistical and soft computing techniques for Big Data, quantitative modeling of processes, game theoretic applications to networking and to security. He has been conducting research activities within several EU FP7 and H2020 projects.

From 1992 to present has co-authored over 150 papers published in internationally refereed journals and conferences.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Mulugeta Libsie has received a PhD in Computer Science from the University of Klagenfurt, Austria. Since 2004, he is Assistant Professor at the Department of Computer Science of Addis Ababa University (Ethiopia), where he lectures courses in Operating Systems, Computer Networks, Computer Security, and Distributed Systems. Among his research interests are networking, distributed information systems, and wireless networks. He is the author and co-author of several journal papers, conference proceedings and textbooks.

ACCEPTED MANUSCRIPT