Modeling DoS Attacks in WSNs with Quantitative Games

Quentin Monnet
Lab. LACL,
Université Paris-Est, France
quentin.monnet@lacl.fr

Mathieu Sassolas
Lab. LACL,
Université Paris-Est, France
mathieu.sassolas@lacl.fr

Lynda Mokdad
Lab. LACL,
Université Paris-Est, France
lynda.mokdad@u-pec.fr

***Abstract*—In this article, we propose to use game theory to model our WSN network. In this setting, the goal of the compromised node is to keep disrupting the network while remaining alive. The game studied is a two-player quantitative infinite game on a finite graph, where each transition can change some energy levels and some reward. The goal of the compromised node is hence to maximize its reward while maintaining a positive energy level. On the theoretical side, we show that solving these games is not algorithmically possible if the objective is too complex. We can however provide solutions in some restricted cases. The ultimate purpose is to demonstrate that, with the presented detection solution, a compromised node cannot "win the game", and hence either gets detected, dies, or behaves as an normal (sane) node would.**

Keywords: WSN, Security, Game theory

I. Introduction

Projects such as the *Internet of things* or *smart cities* are most probably going to interconnect a multitude of devices, and to bring many functionalities to the end user through an extensive use of connected sensors. Ambient light, temperature, air pollution degree measurement, or traffic monitoring are mere examples of civil applications involving those sensors. There are also military uses: sensors may be deployed as networks to detect the presence of biological, chemical or nuclear agent, or to monitor infantry units.

Such networks are called *wireless sensor networks* (WSNs). The sensors (or *nodes*) are small devices able to gather data on their physical environment. They communicate with one another through radio transmission, but they have low resources at their disposal: limited computing power, limited memory, as well as a limited battery. They are often dropped into hostile areas (by helicopter for instance), or may generally be difficult to access, so the batteries must be considered as single-use. The sensors have to self-organize themselves and to deploy low-consuming routing algorithms so as to create a functional network. All relevant data is forwarded to an entity called *base station* (BS), which does not have the same limitations as the sensors, and acts as an interface between the WSN and the user (or the external world).

An important issue with WSNs is the security of the network. It may be necessary to ensure that data flowing in the network cannot be overheard (confidentiality), or that each participant in a data exchange is actually the one it pretends to be (authentication). This is especially important in the case of military uses, as collected data must not be accessible to the enemy. Availability is also an important property: once the network is deployed, it should be accessible for its intended use. Sometimes an opponent disagrees with that, and tries to launch a denial of service (DoS) attack to disrupt the network. In this article we try to prevent some of these attacks. More precisely, we consider a compromised node trying to send more messages than other legitimate sensors and dropping the packets it should forward. Then we try to create a situation in which the rogue node either fails to reach is goal, dies (*i.e.* runs out of power), or gets detected.

To this aim, we model the running network as a game. Two quantitative aspects are taken into account: the energy levels of the sensors and the number of messages that they successfully send to the base station. While legitimate sensors try to collaborate so as to provide an efficient service, the compromised node may drop received packets and tries to send as many of its own messages as possible. The number of successfully sent messages may be seen as a "reward", a payoff; we mostly consider the mean-payoff value, that is to say the payoff of a player divided by the number of its actions. The energy, on the other hand, leads to a strong constraint: a node whose energy level drops to zero dies and immediately looses the game.

The contributions of this paper are twofold: we provide a game model to deal with wireless sensor networks which is well-suited to deal with the constraints of such networks (energy consumption and availability as a goal). Then we study the theoretical properties of these games. We show that if a broad range of objectives on the energy and payoff values are allowed, deciding which player wins the game becomes undecidable. However, when limiting the scope of allowed objectives, we obtain algorithms to decide whether a compromised node may harm the system.

The rest of the paper is organized as follows: Section II briefly introduces some related work achieved over the same research area. Section III presents the problems and model we use for our quantitative games. We introduce the theoretical results in Section IV. At last, the conclusion in Section V permits us to sum up our contribution and to consider future work leads.

II. Related work

A. WSNs and DoS attacks

Security in wireless sensor networks has been deeply investigated over the last years. While generally facing the same security issues as generic wireless network, WSNs cannot resort to heavy intrusion detection systems (IDSs) or to highly consuming cryptographic protocols. Thus a variety of mechanisms addressed to WSNs have been proposed. Many of them provide solutions to ensure data privacy [7], or authentication [11] in the network.

The issue at hand in this paper is the availability of the network. We want to ensure that the network remains up and running, no matter what an attacker attempts. There are many existing denial of service attacks. Some may consist in basically jamming the channel used for communication with radio noise. Other attacks can target higher level such as the MAC protocols or the routing protocols [12]. For instance we previously introduced a mechanism to elect monitoring nodes in clustered networks so as to watch and detect potential rogue nodes [7].

B. WSNs and game theory

The authors of [5] categorize game-theoretic approaches in WSNs into three main categories: energy efficiency, security, and pursuit-evasion. Pursuit-evasion games consist in a set of mobile players trying to "capture" another set, to optimize tracking, while the opponents aim at avoiding detection. The other two categories are rather explicit: when looking for energy efficiency, games are used to save as much energy as possible, by optimizing either the network topology or their own behavior [3]. Security games, of course, oppose normal sensors to attackers from inside or outside the network. In a more recent survey [10] the pursuit-evasions games are just part of a broader "application" category (along with data collection for instance), whereas energy efficiency has been split into network management (resources, power) and communication (QoS, topology, routing design).

In [6] the authors model the interactions between the nodes and an IDS as a Bayesian game (*i.e.* with partial information: the IDS does not know *a priori* whether a given node is compromised). They analyze the Nash equilibrium of this game to design a secure routing protocol.

Repeated games are models involving sequences of history-dependent game strategies: the players perform a sequence of actions, and their strategies is influenced by what the other players have done in the past. Those games are used in [1] to set up an IDS, or in a less generic solution in [9], which relies on the acknowledgments upon transmissions to detect a malicious node located in the forward data path.

Our approach differs as the potential death (by exhaustion) of nodes is included in the game constraints, alongside payoff values. To the best of our knowledge, works related to this setting consider all dimensions either as energy or as payoff, and deal only with conjunction [14]. In this case, the algorithms follow the same structure: the game objective is decomposed according to each dimension and finite-memory winning strategies for each dimension are retrieved. Then these strategies are combined, possibly yielding an infinite-memory strategy.

Some other works [13] deal with more involved objectives based on combination of mean-payoff objectives using sum, $\max$, and $\min$ operators.

A little bit further from our approach, some authors consider games with both a payoff requirement and a parity objective [2]. In that case the parity objective ensures that the system behaves correctly, while the payoff represents a quantitative goal. In this case, winning strategies may require infinite memory, although an approximation can be obtained with finite memory. This kind of game is however ill suited to the modeling of wireless sensor network, since here the energy is an important factor to the life of the system.

III. Infinite quantitative games on finite graph

We consider a network of sensors in the wild; one of them is corrupted and may try to send lots of messages, forgetting its *relay* role towards the others. The "normal" sensors try to collaborate, hence they can be seen as the same *coalition* whose goal is for the whole network to function properly. On the other hand, the corrupted sensor tries to transmit as many messages as possible. They are all facing one limitation: energy. Sending messages requires more energy than waiting, hence it may lead to a timely death of a sensor.

A. The arena

The *arena* of the game is a graph $\mathcal{G} = (V, E)$ where $V = V_c \uplus V_g$ is the set of states, partitioned between the states V_g of the *good* sensors and the states V_c of the *corrupted* sensors. The edges E is a subset of $V \times V$ such that for every $q \in V, \exists q' \in V, (q, q') \in E$, *i.e.* there are no end states. The graph is *weighted* over k dimensions, meaning there is a function $w : E \to \mathbb{Z}^k$ that assigns weights for each dimension to every edge.

B. The semantics

A *configuration* of the game is a tuple $(q, p) \in V \times \mathbb{Z}^k$: q is the current state, p is the accumulated payoff. From a configuration (q, p), if an edge $e = (q, q')$ is taken, then subsequent configuration is $(q', p + w(e))$.

A *run* is an infinite sequence of such configurations. A finite prefix of a run is called a *history*.

The partition of V decides which player chooses the next edge. A strategy for a player is a function that gives this chosen state (hence edge): $\sigma_c : V^*V_c \to V$, $\sigma_g : V^*V_g \to V$.

Given a pair of such strategies and an initial configuration, the game yields a single run called *outcome* of the strategies: $outcome(\sigma_c, \sigma_g)$.

Remark 1: All the above definitions extend naturally to the case of more than two players.

C. Winning conditions

Here, we consider zero-sum games: the goal of the good sensors is to have the corrupted sensor lose. In the general sense, a winning condition is a subset of the runs. For practical reasons, winning conditions studied in the literature are ones that can be expressed finitely: ω-regular conditions [4], and ones based on the values of counters.

In our case, the dimensions of the weight can express either an energy level or a payoff. We hence assume that dimensions are separated into k_e energy dimensions and k_v payoff ones, and a configuration is now $(q, \mathit{val}, \mathit{energ})$. Energy levels are meant to remain above zero, regardless of their value. Payoffs can be negative, although the goal of the player receiving the payoff is to maximize it. Since we consider infinite runs, it is sensible to average the payoff with respect to the length of the run, hence considering *mean-payoff* as an objective.

The set of winning runs is then given by the set of runs that satisfy a given *payoff formula* in the following grammar:

$$\varphi ::= \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg at \qquad at ::= p_e \bowtie c \mid p_v \bowtie c$$

where $c \in \mathbb{N}$, $\bowtie \in \{\geq, >\}$ is a comparison operator p_e is an energy component and p_v a payoff component. Note that constants c could also be in $\mathbb{Q}$ and scaled back to $\mathbb{N}$ along the weights of transitions.

The semantics is as follows. A run ρ satisfies an atom $p_e \bowtie c$ if for every $i \in \mathbb{N}$, $\mathit{energ}(\rho_i)_e \bowtie c$. A run ρ satisfies an atom $p_v \geq c$ if $\limsup_{n\to\infty} \frac{\mathit{val}(\rho_{\leq n})_v}{n} \geq c$. That is, for a given component, we consider the sum of all the weights divided by the number of steps. Since runs are infinite, we consider the limit of this average for prefixes of increasing length. Since the limit may not always exist, we chose superior limit. This is consistent with the fact that the network considers the worst case scenario. Although inferior limit can be used from a modeling point of view, the solving of the games is more involved.

The satisfaction of boolean combination of atoms is defined in the classical way. Note that the negation can only happen to atoms. Following classical vocabulary, an atom or its negation is called a *literal*. In solving those games, we consider the *positive fragment* of this logic, *i.e.* conjunctions of literals.

A run is hence winning if the formula is satisfied, written $\rho \vDash \varphi$.

Remark 2: Without loss of generality, payoffs start with value 0, while energy levels start with a non-zero positive value, called the *initial credit*.

D. Decision problems

The decision problems that arise on this setting are the following:

- **Winning problem:** Given an initial configuration and a payoff formula φ, is there a strategy σ_c for the corrupted sensor such that for every strategy σ_g of the good sensors, *outcome*$(\sigma_c, \sigma_g) \vDash \varphi$?
- **Initial credit problem:** Given an initial state q and a payoff formula , is there a value $\chi \in \mathbb{N}$ such that the winning problem from configuration $(q, 0^{k_p}, \chi^{k_e})$ answers positively?

IV. Solving games with energy and payoff

We address in this section the problem of solving these games. We first show that it is undecidable to solve games where the winning condition is an arbitrary payoff formula. Then we focus on the positive fragment of such formulas, and give sufficient conditions for the game to be solvable.

In order to give a general presentation, the two players in the games will be called player 0 and player 1. Player 0 is the one that has to fulfill the objective given by the payoff formula, hence he represents the compromised node. On graphs provided as figures, all its states are depicted as squares. On the other hand, player 1 represents the other nodes, and its states are depicted as circles.

A. Undecidability of the general case

In this section we show that there is no algorithmic solution to solve the winning problem on these games if the winning condition can be specified by any payoff formula. Namely, we show the following:

Theorem 1: The winning problem and the initial credit problem are undecidable for objective defined by a given payoff formula with four energy components and one payoff component.

The proof consists in encoding the halting problem on two-counter machines – which is known to be undecidable with such a game. Each counter is represented by two energy components (one for each player) carrying a copy of the value of said counter. In the case of the 0-test, player 0 can claim that the counter was or was not null and the second one can check the validity of the claim. If the machine reaches its halting state, a reward component is incremented, and all other components are reset to 0. Hence if the machine halts, player 0 has a winning strategy (encoding faithfully the machine's behavior) to get a strictly positive reward. Otherwise the reward remains null.

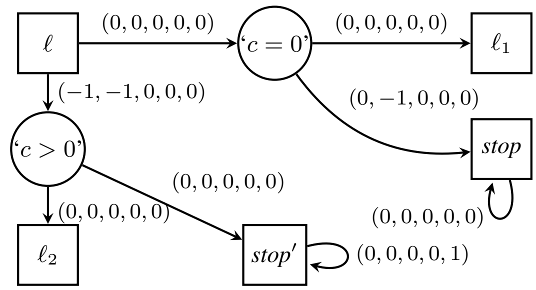

Figure 1. Module for encoding a 0-test instruction into a game: "if $c = 0$ then goto ℓ_1 else decrement c and goto ℓ_2".

More precisely, a two-counter machine is a list of labeled instructions which can be:

- **increment:** increment counter c (resp. d) and goto label ℓ
- **0-test:** if $c = 0$ (resp. $d = 0$) then goto ℓ_1 else decrement c (resp. d) and goto ℓ_2
- **halting:** *halt*

The first label is the initial one. Initially, both counters are set to 0.

Now given a machine $\mathcal{M}$, we build an arena $\mathcal{G}_{\mathcal{M}}$ based on small modules for each kind of instructions (for counter c, it is symmetrical for d). The arena has 5 components c, c', d, d', p, where only p is a payoff. Components c and c' (resp. d and d') are supposed to always retain the same value, except in the 0-test. Initially, all components have value 0.

The goal of player 0 is to obtain a strictly positive payoff while faithfully simulating the machine. The faithful simulation depends on the component's values: c and d must remain positive. Components c' and d' ensure that player 1 cannot wrongfully claim that player 0 cheated. If so, their value will be strictly negative. As a result, the winning condition is the satisfaction of the following payoff formula:

$$(c \geq 0 \land d \geq 0 \land p > 0) \lor (c' < 0 \land c \geq 0) \lor (d' < 0 \land d \geq 0)$$

Each labeled instruction has a state in the arena belonging to player 0, with the same label (some other states will be added). The case of incrementation is trivial if instruction ℓ is "increment c then goto ℓ'", then there is an edge from ℓ to ℓ' with weights $(1, 1, 0, 0, 0)$. There is no other edge from ℓ, so the player has no choice.

The 0-test is trickier, since player 0 could "cheat" by choosing the wrong branch of the conditional. Formally, assume ℓ is "if $c = 0$ then goto ℓ_1 else decrement c and goto ℓ_2". Then there are two edges stemming from state ℓ: one claiming that $c = 0$, the other claiming that $c > 0$, as depicted in Figure 1. The edge claiming that $c = 0$ goes to a state owned by player 1, which can either accept the claim (hence going into ℓ_1) or reject it. Rejecting the claim means decrementing c' (but not c) and going to a *stop* state: this state has one edge to itself labeled $(0, 0, 0, 0, 0)$, meaning the simulation has stopped. Note that wrongfully rejecting the claim means that $c' < 0$ while $c = 0$, since c and c' were equal. Rightfully rejecting it means that the payoff is null.

Similarly, the claim that $c > 0$ is made by an edge decrementing both c and c' (hence $(-1, -1, 0, 0, 0)$), and going to state of player 1. He can either follow with the simulation or go to a *stop'* state. This state has a single edge to itself labeled $(0, 0, 0, 0, 1)$, meaning the

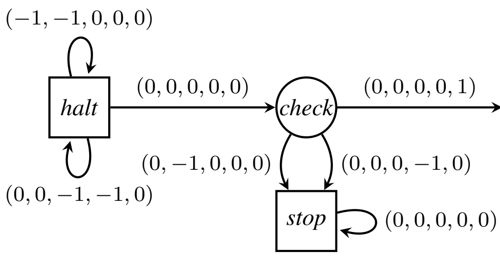

Figure 2. Module for encoding the halting instruction into a game.

simulation has stopped but the payoff is 1. If player 0 cheated, then $c < 0$ and the simulation need not go further. If he didn't, then both $c > 0$ and $c' > 0$, and $p > 1$.

Finally, upon reaching the *halt* state, player 0 decreases both c and c' at will, then go to a *check* state. State *check* belongs to player 1, and behaves as in the case of 0-test where player 0 claimed that $c = 0$. The same is also done for d, and finally an edge goes back from *check* to the initial state giving a reward to the p component: $(0, 0, 0, 0, 1)$. This module is depicted in Figure 2.

Now assume that machine $\mathcal{M}$ halts. Consider the strategy of player 0 that plays faithfully *i.e.* claims exactly what the counter value tells (and in the end decrements components so that they exactly reach 0). Then if player 1 never claims wrongdoing, counters c and d remain positive as in $\mathcal{M}$. In addition, assume $\mathcal{M}$ finished in k steps and let c_k and d_k be the values of c and d, respectively, upon reaching the halting instruction. Then the run of the machine is then at most $2k$ steps in the game (because of the claims that must be accepted), then from the *halt* state the game needs $c_k + d_k + 2$ steps to reach the initial state again: it is the number of steps required in order to decrement both counters and traverse the module of Figure 2. Hence the mean payoff is at least $\frac{1}{2k+c_k+d_k+2} > 0$.

If player 1 rejected a claim:

- if $c = c' = 0$ but player 1 decremented c' to contest the claim, then the game "stops" but condition $c' < 0 \wedge c \geq 0$ is satisfied;
- if $c = c' \geq 0$ after decrementation but the game was "stopped", payoff is 1 while both c and d are above 0, hence condition $c \geq 0 \wedge d \geq 0 \wedge p > 0$ is satisfied.

As a result, this faithful simulation strategy is winning.

On the other hand if $\mathcal{M}$ does not halt, faithful simulation yields a payoff of 0 provided player 1 never wrongfully attacks a claim (the halt state is never reached, and if player 1 does not claim cheating, stop$'$ is not reached either). Note that in case of unfaithful simulation from player 0, player 1 only needs to detect this fault: either $c < 0$ or the payoff is irremediably null.

This construction can be adapted to the initial credit problem: one needs to add a module before the initial state that resembles the *halt* module, in the sense that it ensures bringing all energy levels to 0. Hence any initial credit is irrelevant and player 0 wins if and only if $\mathcal{M}$ halts.

B. The positive fragment

We now consider the initial credit problem. It is "easier" for the compromised node to win for this game rather than for the corresponding winning game. Additionally, it is common when combining strategies that an initial credit required to win is increased; in other words, strategies for the winning problem are usually not robust enough. From the model point of view, solving this problem yields more information: if a strategy and a credit are found, they give an actual value to which the initial energy level must be set.

We show in this section that winning strategies for objectives defined by literals may be combined into a winning strategy for the whole objective.

First, the case of a literal is the well-studied case of either energy games or mean-payoff games. These games have simple solutions, in the sense that (1) these games are determined, *i.e.* one of the player has a winning strategy; (2) if a winning strategy exists, there exists one with finite memory; (3) which player wins the game can be decided in $\mathsf{NP} \cap \mathsf{coNP}$ [15]. Moreover, it was proved in [14] that finite memory suffice to win a conjunction of energy requirements. The case of conjunction of payoff requirement has also been studied in [14], where the memoryless strategies for each payoff condition is combined into infinite memory strategies for the whole objective. For example, when mean-payoff is defined with the superior limit: each strategy is played in increasingly longer phases until reaching the desired value.

As a result we here focus on mixing energy and payoff objectives. Remark that since the objective of the compromised node is a conjunction of literals, it is clear that if the objective specified by one of the literals cannot be achieved, then the conjunction cannot, hence the game cannot be won.

If there are strategies for each of the objectives, it may be possible to combine finite-memory winning strategies for each of the literals into a single winning strategy, possibly needing infinite memory. We provide sufficient conditions to do so; finding an algorithm for this problem is still open.

1) Attractors: In two-player games, it is common to use the notion of *attractor* of a set Q to denote states where a player can force the play to reach Q.

Definition 1: The 1-step *attractors* for players 0 and 1 of a set Q of states are defined as follows:

$$\begin{aligned} \mathit{1Attr}_0(Q) &= \left\{q \in V_0 \,\middle|\, \exists (q, q') \in E \text{ s.t. } q' \in Q\right\} \\ &\quad \cup \left\{q \in V_1 \,\middle|\, \forall (q, q') \in E, q' \in Q\right\} \end{aligned}$$

$$\begin{aligned} \mathit{1Attr}_1(Q) &= \left\{q \in V_0 \,\middle|\, \forall (q, q') \in E, q' \in Q\right\} \\ &\quad \cup \left\{q \in V_1 \,\middle|\, \exists (q, q') \in E \text{ s.t. } q' \in Q\right\} \end{aligned}$$

The *attractors* for player 0 (resp. player 1) of a set Q of states are the fix-point of the 1-step attractor, starting from Q: for $i \in \{0, 1\}$,

$$\mathit{Attr}_i(Q) = \bigcup_{j \in \mathbb{N}} \left(\mathit{1Attr}_i\right)^j (Q)$$

The attractor of player i is therefore the set of states from which he can ensure that the play reaches Q. Note that from any state in $(\mathit{1Attr}_i)^j(Q)$, player i has a memoryless strategy to reach Q in at most j steps. Since the fix-point is reached in at most $|V|$ iterations,

Q can be reached in at most $|V|$ steps from any state of $\mathit{Attr}_i(Q)$. Note that this bound also shows that attractors can be computed in polynomial time.

Lemma 2: From any state of $\mathit{Attr}_i(Q)$, player i has a memoryless strategy that ensures reaching Q in at most $|V|$ steps.

A property of attractors in games is that they can be "safely" removed from a game while leaving the graph structure still a game (*i.e.* without end-states):

Lemma 3: Let $\mathcal{G} = \langle V_0, V_1, E\rangle$ be a game graph (*i.e.* such that every state of V has an outgoing edge in E). Let $j \in \{0, 1\}$ be a player. Let $Q \subseteq V$ and consider the graph $\mathcal{G}' = (V_0', V_1', E \cap (V' \times V'))$ with $V_i' = V_i \setminus \mathit{Attr}_j(Q)$ for $i \in \{0, 1\}$. Then every state of V' has an outgoing edge in $E \cap (V' \times V')$, *i.e.* $\mathcal{G}'$ is also a game graph.

Proof: Assume by contradiction that $q \in V'$ has no outgoing edge. Since q had an outgoing edge in E, it means that all successor states of q belong to $\mathit{Attr}_j(Q)$. Then by definition of an attractor, so does q, and $q \notin V'$. ■

2) Conjunction of an energy and payoff objective: In this first simpler case, we study objectives for player 0 of the form $p_e \geq c_e \wedge p_v \geq c_v$, with $c_e, c_v \in \mathbb{N}$. These objectives state that a certain reward must be achieved while maintaining the energy level above a given limit. A simple example of this kind of objective for a compromised node is the *greedy* objective: to remain alive ($p_e \geq 1$) while sending at least a message every 6 steps ($p_v \geq \frac{1}{6}$, as noted before, this can be transformed into an integral threshold by multiplying each weight of this component by 6).

First, it is clear that from states where one of the objective cannot be fulfilled, player 0 cannot win. In addition, if player 1 can force to reach such states, then the objective of player 0 cannot be fulfilled. Namely, we use the classical notion of *attractors* defined above. We write L_e the states where player 0 loses for $p_e \geq c_e$ (*i.e.* player 1 has a strategy to prevent that), and L_v the states where player 0 loses for $p_v \geq c_v$. It is clear that player 1 can prevent player 0 from winning for objective $p_e \geq c_e \wedge p_v \geq c_v$ from any state in $\mathit{Attr}_g(L_e \cup L_v)$. Hence any winning strategy for player 0 must remain in $\mathcal{G} \setminus \mathit{Attr}_1(L_e \cup L_v)$. Conversely, a winning strategy that remains in $\mathcal{G} \setminus \mathit{Attr}_1(L_e \cup L_v)$ is also a winning strategy in $\mathcal{G}$ since player 1 cannot force the play into $\mathit{Attr}_1(L_e \cup L_v)$, by the definition of attractors.

As a result, one can recursively remove states in $\mathcal{G}$ until player 0 wins for both objectives in every state. Note that if the game is empty at that point, then player 0 cannot win from any state.

Now assume player 0 has winning strategies λ_e, λ_v for objectives $p_e \geq 1$, $p_v \geq \frac{1}{15}$, respectively, that win from every state. Remark that these strategies can be assumed to be memoryless, hence functions from V_c to V.

Consider $\mathcal{G}_e$ the single player game obtained when player 0 plays λ_e: each transition entering a state $v \in V_c$ goes instead to $\lambda_e(v)$, and weights are added: $v_0 \xrightarrow{w_1} v \xrightarrow{w_2} \lambda_e(v)$ is replaced by $v_0 \xrightarrow{w_1+w_2} \lambda_e(v)$. Similarly, let $\mathcal{G}_v$ the single player game obtained when playing λ_v.

Let α be the lowest value that can be obtained for component k_v in a simple cycle in $\mathcal{G}_e$, and dually β the lowest value that can be obtained for k_e in $\mathcal{G}_v$. In other words, α is the worst that can happen to reward when playing the strategy ensuring adequate level of energy, while β is the worst that can happen to energy level when

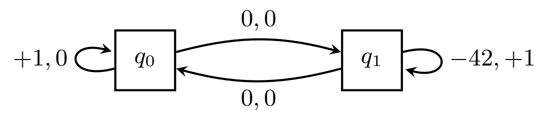

Figure 3. A simple game that requires infinite memory. The first component is an energy level while the second is a payoff.

ensuring adequate reward. Note that if $\alpha \geq c_v$, then λ_e is also a winning strategy for objective $p_v \geq c_v$, hence is a winning strategy for the whole objective. Similarly, if $\beta > 0$, then λ_v ensures objective $p_e \geq c_e$ provided the initial credit is adapted to the minimal sum of weights reached along a cycle:

$$\mathit{initial\_credit} = 1 + c + \min_{\substack{\rho \text{ prefix of } \mathcal{C} \\ \mathcal{C} \text{ cycle in } \mathcal{G}_v}} w_e(\rho)$$

If the above sufficient condition is not fulfilled, we do not so far have a solution for solving these games in the general setting. Indeed, strategies for each component or the other can be incompatible. For example, consider the (single-player) game of Figure 3. In this example, q_0 acts as a recharging state while q_1 is an active state, producing a useful effect rewarded by a payoff. One can see that recharging the battery is much slower than using it, since it takes 42 "energy units" per active step.

One can achieve a positive energy (first component) at all times by remaining in q_0 (or going there if starting from q_1). It is also possible to achieve a payoff (second component) of 1 by remaining in q_1 (or going there if starting from q_0, the transitive effect is negligible in the long run). However, one cannot achieve a payoff of 1 while maintaining the energy positive, since it takes 42 turns of "recharging" before being allowed to do something rewarding.

3) Using bounded memory: However, one can consider that nodes of a wireless sensor network are very limited in their resources, hence can only implement finite memory strategies. In this case, bounding *a priori* the amount of memory that can be used by player provides a solution for solving the initial credit problem for games with winning condition in the positive fragment.

A finite memory strategy is a strategy that can be implemented by a finite deterministic Mealy machine: given the current state of the machine and of the game, the machine produces an edge to be played and the next state of the machine. The size of the memory is the size of the Mealy machine.

For example, a finite memory strategy for the game of Figure 3 that loops 42 times in q_0 then goes to q_1, loops once there, and goes back to q_0 to start again can be represented by the machine of Figure 4. This machine has 46 states, since it needs to count how many times it has accumulated energy in q_0. Note that if starting in q_1, the machine first goes back to q_0 then applies the aforementioned strategy. Also, a strategy must be complete on its input, hence it must allow q_1 to occur in any memory state (in this case it goes back to the initial memory state and to q_0 in the game).

Remark 3: Note that the strategies with given memory may not be optimal. Consider for example the game of Figure 3 with objective being to maintain the energy level above 0 and ensure a mean-payoff greater than or equal to $\frac{1}{43}$.

An infinite memory strategy can win this game. It is defined by phases, as follows. At the k-th phase, loop $42k$ times in q_0, then go

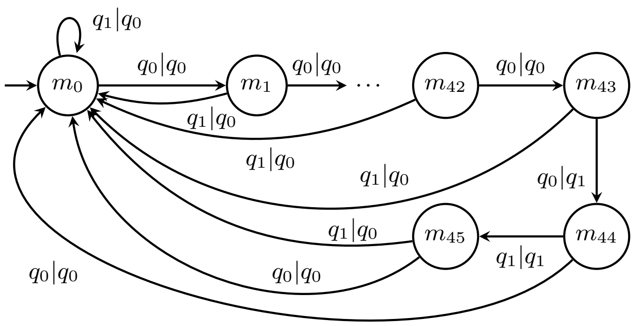

Figure 4. A Mealy machine representing a finite memory strategy (46 states). The input of edges is the current state of the game, the output is the chosen next state.

to q_1, loop k times, go back to q_0. This ensures that the transitions between q_0 and q_1 are negligible, hence the limit of the payoff is $\frac{1}{43}$, since 43 steps are needed to increment the payoff counter once. In addition, at each phase the energy goes back to its initial value (which can be 1), while only encountering positive values.

On the other hand, if only k states of memory are allowed, the best payoff achievable is $\frac{k}{43k+2}$ (the corresponding strategy consists in repeating phase k). Hence not only the optimal payoff cannot be achieved, but allowing more memory allows to achieve better payoff.

Solving the game assuming player 0 has bounded memory k given as input consists in guessing this strategy as a machine, which is an exponential object if k is given in binary. Then the machine is synchronized with the game, yielding a single-player game, to be played by player 1. In this game, only memoryless strategies need to be considered.

Indeed, any infinite path not satisfying $p_e \geq c_e \wedge p_v \geq c_v$ either is such that p_e falls below c_e or the limit average of p_v is below c_v. In both cases this amounts to finding a lasso path in the graph, which can be guessed.

Regarding complexity, the procedure described above is in $\mathsf{NEXPSPACE}$ (equivalent to $\mathsf{EXPSPACE}$ [8, Chap. 20]), although the bound is not tight.

V. Conclusion

In this paper, we presented a model of games well suited to study the behavior of wireless sensor networks, combining energy and payoff constraints. We considered the theoretical properties of such games, first showing that a broad range of objectives yield undecidability, then showing sufficient conditions where the game can be decided. Future work include bridging this theoretical gap by finding an exact algorithmic solution for the solving of such games in order to detect greedy attacks; such attacks are not evident in wireless sensor networks as the latter are built as distributed systems.

References

[1] A. Agah and S. K. Das. Preventing DoS attacks in wireless sensor networks. A repeated game theory approach. *International Journal of Network Security*, 5(2):145–153, Sept. 2007.

[2] K. Chatterjee, T. Henzinger, and M. Jurdziński. Mean-payoff parity games. In *Proceedings of the 20th International Symposium on Logic in Computer Science (LICS'05)*, June 2005.

[3] P. Closas, A. Pagès-Zamora, and J. A. Fernández-Rubio. A game theoretical algorithm for joint power and topology control in distributed wsn. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, pages 462–468, Athens, Greece, Apr. 2009.

[4] E. Grädel, W. Thomas, and Th. Wilke, editors. *Automata, Logics, and Infinite Games*. Lecture Notes in Computer Science. Springer, Oct. 2002.

[5] R. Machado and S. Tekinay. A survey of game-theoric approaches in wireless sensor networks. *Computer Networks*, 52(16):3047–3061, Nov. 2008.

[6] M. Mohi, A. Movaghar, and P. M. Zadeh. A bayesian game approach for preventing DoS attacks in wireless sensor networks. In *Proceedings of the 2009 International Conference on Communications and Mobile Computing (CMC'09)*, Kunming, Yunnan, China, Feb. 2009.

[7] Q. Monnet, L. Mokdad, and J. Ben-Othman. Energy-balancing method to detect denial of service attacks in wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications*, Sydney, NSW, Australia, 2014.

[8] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[9] Y. B. Reddy. a game theory approach to detect malicious nodes in wireless sensor networks. In *Proceedings of the third international conference on sensor technologies and applications*, pages 462–468, Athens, Greece, June 2009.

[10] H.-Y. Shi, W.-L. Wang, N.-M. Kwok, and S.-Y. Chen. Game theory for wireless sensor networks: a survey. *Sensors*, 12(7):9055–9097, July 2012.

[11] M. A. Simplicio, Jr, B. T. de Oliveira, P. S. L. M. Barreto, C. B. Margi, T. C. M. B. Carvalho, and M. Naslund. Comparison of authenticated-encryption schemes in wireless sensor networks. In *Proceedings of the 36th Annual IEEE Conference on Local Computer Networks*, pages 454–461, Bonn, Germany, Oct. 2011.

[12] S. K. Singh, M. P. Singh, and D. K. Singh. A survey on network security and attack defense mechanism for wireless sensor networks. *International Journal of Computer Trends and Technology*, May 2011.

[13] Y. Velner. The complexity of mean-payoff automaton expression. In A. Czumaj, K. Mehlhorn, A. M. Pitts, and R. Wattenhofer, editors, *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming (ICALP'12, Part II)*, volume 7392 of *Lecture Notes in Computer Science*, pages 390–402. Springer, July 2012.

[14] Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. Rabinovich, and J.-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *CoRR*, 2012.

[15] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1&2):343–359, 1996.