

# Performance analysis and formal verification of cognitive wireless networks

G.L. Dei Rossi, L. Gallina, and S. Rossi

Università Ca' Foscari, Venezia (Italy)  
{deirossi,lgallina,srossi}@dais.unive.it

**Abstract.** Cognitive Networks are a class of communication networks, in which nodes can learn how to adjust their behaviour according to the present and past network conditions. In this paper we introduce a formal probabilistic model for the analysis of wireless networks in which nodes are seen as processes capable of adapting their course of action to the environmental conditions. In particular, we model a network made of mobile nodes using the gossip protocol, and we study how the energy performance of the network varies, according to the topology changes and the transmission power. The stochastic process underlying the model is a discrete time Markov chain. We use the PRISM model checker to obtain, through Monte-Carlo simulation, numerical results for our analysis, which show how the learning-driven dynamic adjustment of transmission power can improve the energy performance while preserving connectivity.

## 1 Introduction

Cognitive networks [4] are communication networks in which nodes can alter their behaviour according to changes of the environmental conditions. What differentiate this approach from the one of *cognitive radio* [8] networks are that, while in the latter the choices that nodes can take are restricted to radio channel selection, in the former nodes can take complex decisions, taking into account the global goals of the network. Cognitive processes are particularly useful when we have to deal with ad hoc networks, where the absence of a fixed infrastructure and the dynamic nature of the network topology, as well as the limited power capacities of nodes, make the network prone to problems such as link breakages, energy waste and interferences.

Topology Control is a technique aimed at guaranteeing network connectivity, while optimising network performance with respect to several metrics, depending on the specific objective of each single network.

Although several formal models for the analysis of wireless ad hoc and sensor networks and for cognitive radio networks were proposed in the literature (see, e.g., [13, 5]), to the best of our knowledge formal models for the analysis of cognitive networks are rare. In [9] the authors discuss the issues concerning the definition of a PEPA model for cognitive networks, although they do not propose any actual model, and thus they do not perform any quantitative or qualitative analysis.

PRISM [10] is a tool for modelling and analysing systems that exhibit a probabilistic behaviour. It supports, among others, the modelling of Markov Decision Processes (MDPs), where nondeterministic and probabilistic aspects coexist. In addition to the traditional model checking, PRISM provides *statistical model checking*, allowing one to compute probabilities of properties' satisfaction. In particular, PRISM also offers a discrete-event simulator, allowing one to generate approximate results for the verification of properties. This approach is particularly useful for very large models, when other approaches to model checking are not feasible, due to the well known problem of state space explosion.

This paper presents a probabilistic model for the analysis of networks exhibiting cognitive behaviours. The model is written in the PRISM language, and supports broadcast communications, node mobility, and the ability of nodes to dynamically adjust the transmission power during their operations.

*Paper structure.* The paper is organised as follows. In Section 2 we give an introduction to the use of cognitive networks for topology control, Section 3 reviews the basic features of PRISM that we use in the rest of the paper. In Section 4 we introduce a novel model for cognitive networks, and in Section 5 we use the PRISM tool to analyse its behaviour, giving numerical examples. Finally, in Section 6 we give some final remarks, concluding the paper.

## 2 Topology Control with Cognitive Networks

Topology Control [14] is a technique aimed at guaranteeing the connectivity of a communication network, while limiting other cost factors, such as the level of interference and the energy consumption, thus extending the network lifetime. In the presence of mobility this problem is not trivial, since the network topology continuously changes, causing frequent link breakages and variations in the interference levels. In wireless networks, this can be considered as the problem of finding a trade-off between power saving and network connectivity through the choice of the appropriate transmission power for each node. It is evident that if each node transmits at a low power, then its connectivity level, and potentially the one of the whole network, will be reduced, while if we assign high transmission power to the nodes, we generally enhance the connectivity of the network, but we consume far more energy. This relation is, indeed, not a trivial one, since increasing transmission power, and thus the coverage area of a radio station, can increase the chances of collisions and interferences, decreasing the whole network connectivity. For omni-directional antennas we can reasonably model the coverage radius as a function of the power used by the transmitter, and vice-versa. The function can be arbitrary, but usually the coverage radius is proportional to the square root of the transmission power [12]. Of course connectivity is also influenced by factors independent from the transmission power, such as routing and link-level protocols. However in this paper we focus on energy consumption, leaving all the other factors unchanged. In particular, we assume that the network uses the well-known *gossip* protocol to propagate messages.

In this article we also assume that every node in the network is somewhat smart, and capable of applying some strategies to decide its transmission power, based on the conditions in which it operates. In particular, we assume that, observing the past behaviour of the network, or using some link-level techniques usually employed for interference, collision and congestion detection [17], each node is able to guess how many other stations are present in a given radius. Given that information, the node can perform a very simple decision, i.e.,

- If there is a radius  $r < r_{\max}$  for which there are at least  $n$  other nodes, use the minimum transmission power capable of transmitting with radius  $r$ .
- Use the maximum allowed power, corresponding to radius  $r_{\max}$ , otherwise.

It is clear that, due to mobility and interferences, the guess of the aforementioned node can be wrong, however this mistake will have an effect on the next retransmissions of the node itself. In this way, we have just defined a *cognitive network* in which nodes are able to learn, from the observed environment, an appropriate behaviour for the net itself.

### 3 The PRISM Model Checker

PRISM [10] is a probabilistic model checker which supports several types of models, such as *discrete-time Markov chains* (DTMCs), *continuous-time Markov chains* (CTMCs), *Markov decision processes* (MDPs), and Probabilistic Timed Automata (PTA). Models are expressed using PRISM’s own language.

This paper deals with models that can be represented by Discrete Time Markov Chains [15], and studies their qualitative and quantitative properties using model checking techniques. In the following we briefly introduce the main aspects of the PRISM language.

#### 3.1 Modules.

PRISM models consist of modules, expressed through a simple state-based language. A *module* is specified as:

```
module name
...
endmodule
```

and it is composed of variables and commands. *Variables* are names associated to values. The syntax for variables is:

```
name : [ range ] init initial_value;
```

*Commands* describe all the possible behaviours of the modules, i.e. all the possible transitions from one state to another. They include *guards*, which indicate the states where the transitions can occur, and the *updates*, which modify the variables in order to reach the arrival states. The syntax for a command is:

```
[action] guards -> p1:update1 + p2:update2 .. pn:updaten;
```

where  $p_1, \dots, p_n$  express the probability of each possible update ( $\sum_{i=1}^n p_i = 1$ ), `guards` is the list of conditions associated to that transitions, and `action` is the label of the transitions, which is used to synchronise different modules, since two modules can synchronise if they can execute an action with the same label.

### 3.2 The Property Specification Language

PRISM provides a specification language to express rewards and quantitative properties and it supports the automated analysis of these properties with respect to the probabilistic models. It supports several temporal logics, such as PCTL (Probabilistic Computation Tree Logic) and LTL (Linear Temporal Logic) [7]. In particular, when dealing with DTMCs, the PRISM property specification language enables us to study many important properties, such as the probability to reach a particular state under some conditions.

The **P** operator is used to reason about the probability of the occurrence of an event. Formally, we write:

```
P bound [ pathprop ]
```

which is **true** if the probability that the path property `pathprop` is satisfied by the paths reachable from the initial states respects the bound `bound`.

We can also adopt a quantitative approach, by computing the actual probability that a path property is satisfied. An example is:

```
P =? [pathprop]
```

which computes the probability of satisfying `pathprop`.

The PRISM property specification language introduces a set of *temporal operators* in order to express the PCTL path formulas or the LTL formulas which can be verified for a single path of a model. Among these operators, the most used are **F**, which expresses the property that the condition will be eventually satisfied by the path, and **G** which expresses the property that the condition is always true (i.e., it expresses the *invariancy* property).

### 3.3 Costs and Rewards.

Reward properties are based on the possibility of defining rewards associated with a given PRISM model. Rewards can assign values, or costs, either to states or transitions. We are interested in transitions rewards, whose syntax is:

```
rewards ‘‘name’’  
[action_1] constraint_1 : cost_1  
[action_2] constraint_2 : cost_2  
...  
[action_n]constraint_n : cost_n  
endrewards
```

where, for each  $i \in [1 - n]$  assigns the cost `cost_i` to the transitions labelled with `[action_i]` satisfying the constraint `constraint_i`.

With the PRISM property specification language we can use the **R** operator to compute the expected value of the rewards associated with the model. As for the *reachability properties* (the **P** operator), we can verify if the cost of reaching the states satisfying some particular property respects a certain bound:

`R bound [ rewardprop ]`

We can also compute the expected cost of reaching states satisfying a given property:

`R = ? [F rewardprop]`

### 3.4 Statistical Model Checking

Due to the well-known problem of state space explosion, in addition to the standard model checking techniques, which need to build the entire model for the verification of properties, PRISM also provides a *discrete-event simulator*, which can be used to perform approximate (or statistical) model checking. Approximate results can be obtained by generating a large number of paths through the model, without building the entire state space, evaluating the properties on each run, and using the information to generate approximate results. This technique can be used to analyse both reachability and reward properties, and it is particularly useful to study models with a large number of modules and interactions (see [11]).

## 4 The Model

We consider a wireless network with both static and mobile devices, where communications are carried on using a basic gossip protocol. Nodes can, through radio-frequency channels, broadcast messages, which are receivable by all the nodes which are inside the sender node’s transmission area and are listening to the same channel. We analyse the energy costs of a multi-hop communication between two random network nodes, and we study how the ability of learning and reasoning in the processes behaviour can improve the performance of the network.

In particular, we model 15 mobile nodes, and 10 static nodes, evenly distributed in a network area of  $50 \times 100$  square meters, as depicted in Figure 1. The static nodes are located at positions  $\{7, 9, 17, 19, 27, 29, 37, 39, 47, 49\}$ , while the movements for all the other nodes are described by the bidirectional arrows in Figure 1. We model the network area as a grid of  $5 \times 10$  cells. The distances between cells are determined by considering the centre of each cell and calculating the euclidean distance between each pair of centres (each cell is  $10 \times 10$  square metres). Moreover, we consider each node as a cognitive process, that can dynamically change the transmission power for its communications, depending on the position of its active neighbours, with the global aim of an efficient topology control. Usually, modern technologies allow the devices to choose among a

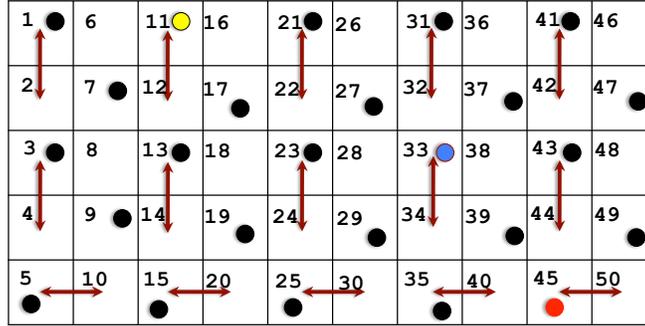


Fig. 1: Topology of the Network

discrete set of possible power levels. In what follows we will use the transmission radius to represent the transmission power, since those quantities are strictly related. As we mentioned in Section 2, usually the power spent for a transmission is proportional to the squared radius. The processes that model nodes listen to the channel and, when they receive a message, they forward it, according to the gossip strategy, i.e., they will forward the message with a certain probability  $\mathbf{psend}$ , and discard it with probability  $1 - \mathbf{psend}$ . We will study the performance of the network, for different gossip strategies, i.e., with the value of the forwarding probability ranging in the set  $\{0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0\}$ .

Several papers, such as [6, 1–3], already present analysis of gossip-based protocols, comparing modifications which are particularly appropriate for ad hoc and wireless networks. In this paper we analyse how the presence of cognitive processes in the network can strongly improve the performance of these kinds of communication protocols. In our model, each node can choose its transmission radius in the set  $\{10\text{m}, 15\text{m}, 20\text{m}\}$ . Specifically, it will choose the minimum radius which ensure the possibility to receive the message for at least two receivers or, if there are not enough available neighbours in the transmission area, it will transmit with its maximum power (radius = 20m).

As introduced in Section 3, the PRISM model checker supports different model types. Here we model the network as a DTMC, where probabilities are used to model both the possible topology changes, and the behaviour of the processes. In what follows we will give the essential elements of the mapping of the aforementioned model in PRISM’s own language. Table 1 shows the representation of a single network node.

*Variables.* The most important variables of our model mapping are the following:

- $\mathbf{steps}_i$  controls the sequentiality of the process executed by the sensor node  $i$ . In particular,  $\mathbf{steps}_i = 2$  means that the node is ready to receive,  $\mathbf{steps}_i = 1$  means that the node is ready to transmit, and  $\mathbf{steps}_i = 0$  means that the node has completed a transmission.
- $\mathbf{l}_i$ : is the variable containing the actual location of the sensor node  $i$ .

Table 1: The PRISM module for a node

---

```

module P8
steps8 : [0..2] init 2;
18 : [15..20] init 15;
[move] (18 = 15) → 0.8 : (18' = 20) + 0.8 : (18' = 15);
[movee] (18 = 20) → 0.8 : (18' = 15) + 0.8 : (18' = 20);

//beginning of a new round
[round] no_one_sending → (steps8' = 2);

//transmission
//[c8] (steps8 = 1) → (steps8' = 0);

//reception
[c3] (steps8 = 2) & s1p3 & s1p38 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c3] (steps8 = 2) & s2p3 & s2p38 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c3] (steps8 = 2) & s3p3 & s3p38 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c3] (steps8! = 2) |!((s1p3 & s1p38) | (s2p3 & s2p38) | (s3p3 & s3p38)) → (steps8' = steps8)

[c5] (steps8 = 2) & s2p5 & s2p58 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c5] (steps8 = 2) & s3p5 & s3p58 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c5] (steps8! = 2) |!((s2p5 & s2p58) | (s3p5 & s3p58)) → (steps8' = steps8)

[c7] (steps8 = 2) & s1p7 & s1p78 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c7] (steps8 = 2) & s2p7 & s2p78 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c7] (steps8 = 2) & s3p7 & s3p78 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c7] (steps8! = 2) |!((s1p7 & s1p78) | (s2p7 & s2p78) | (s3p7 & s3p78)) → (steps8' = steps8)

[c10] (steps8 = 2) & s1p10 & s1p108 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c10] (steps8 = 2) & s2p10 & s2p108 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c10] (steps8 = 2) & s3p10 & s3p108 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c10] (steps8! = 2) |!((s1p10 & s1p108) | (s2p10 & s2p108) | (s3p10 & s3p108)) → (steps8' = steps8)

[c12] (steps8 = 2) & s2p12 & s2p128 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c12] (steps8 = 2) & s3p12 & s3p128 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c12] (steps8! = 2) |!((s2p12 & s2p128) | (s3p12 & s3p128)) → (steps8' = steps8)

[c13] (steps8 = 2) & s1p13 & s1p138 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c13] (steps8 = 2) & s2p13 & s2p138 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c13] (steps8 = 2) & s3p13 & s3p138 → psend : (steps8' = 1) + (1 - psend) : (steps8' = 0);
[c13] (steps8! = 2) |!((s1p13 & s1p138) | (s2p13 & s2p138) | (s3p13 & s3p138)) → (steps8' = steps8)

endmodule

```

---

Table 2: Connectivity formulas

---

```

//P1 strategies
formula s1p12 = ((steps2 = 2) & (l2 - l1 = 1));
formula s1p14 = ((steps4 = 2) & (l4 = l1 - 5));
formula s2p12 = s1p12;
formula s2p14 = (steps4 = 2);
formula s2p15 = ((steps5 = 2) & (l5 - l1 = 6));
formula s3p12 = ((steps2 = 2) & (l2 - l1 < 3));
formula s3p14 = (steps4 = 2);
formula s3p15 = s2p15;
formula s3p16 = ((steps6 = 2) & (l6 - l1 = 10));

formula s1p1 = (s1p12 & s1p14);
formula s2p1 = !s1p1 & ((s2p12 & s2p14) | (s2p12 & s2p15) | (s2p14 & s2p15));
formula s3p1 = !s1p1 & !s2p1;

```

---

*Modelling the Network Topology.* In order to model the level of connectivity of the network, which dynamically changes depending on the positions of the nodes inside the network area, and before defining the modules for the network nodes, we introduce a list of formulas, which allow us to verify the distance between each pair of possible neighbours. In particular, for each pair  $i, j \in \{1, \dots, 25\}$  and for each  $h \in \{2, 3, 4\}$ , if the formula  $\text{shp}ij$  is true, it means that the node  $P_j$  is actually able to listen to a  $P_i$ 's transmission with radius  $5 \times h$ . Moreover, for each  $i \in \{1, \dots, 25\}$  and for  $h \in \{2, 3, 4\}$ , if the formula  $\text{shpi}$  is true, then there exists at least two possible receiver nodes inside the transmission area of the sender, when transmitting with radius  $5 \times h$ . Table 2 shows the set of formulas modelling the connectivity of  $P_1$ . As an example,

$$\text{formula s1p12} = ((\text{steps2} = 2) \& (l2 - l1 = 1));$$

is true when node  $P_2$  is ready to receive ( $\text{steps2} = 1$ ), and the distance between  $P_1$  and  $P_2$  is 1, i.e., looking at Figure 1, is true only when  $l1 = 2$  and  $l2 = 3$ , which, since we consider the nodes lying in the centre of each cell, means that radius 10m guarantees their connection.

*Transitions.*

- `[move]`: is the transition modelling the periodic topology changes. Node mobility is expressed in terms of the transition matrix of a discrete time markov chain: each entry of the matrix denotes the probability that a sensor node moves from a location to another. In particular, static nodes are associated with the identity matrix. When the transition `move` is performed, a node will change location with probability  $\varepsilon$ , and will remain in the same location with probability  $1 - \varepsilon$ . Here we choose 0.8 as the value for  $\varepsilon$ .

- [round]: is the transition occurring when no more transmissions are possible. At the end all the nodes will be in the reception state ( $\mathbf{steps}_i = 2$ ), except for the sender node, whose *steps* variable will be set to 1.
- [ci]: is the transition modelling a broadcast transmission. In particular, if a node is in the state *ready to transmit* ( $\mathbf{steps}_i = 1$ ), it will execute the following transition:

$$[ci] (\mathbf{steps}_i = 1) \rightarrow (\mathbf{steps}_i' = 0);$$

meaning that the node  $i$  transmits the message and then transits in a sleeping phase. If another node  $P_j$  is in the state *ready to transmit* ( $\mathbf{steps}_j = 2$ ), and it is inside the transmission area of the sender node ( $\mathbf{s1p}_{ij}$ ,  $\mathbf{s2p}_{ij}$ ,  $\mathbf{s3p}_{ij}$ ), it will synchronize with the sender node and receive the message. Transition [ci] ( $\mathbf{steps}_j = 2$ )& $\mathbf{s1p}_{ij}$ & $\mathbf{s1p}_{ij} \rightarrow \mathbf{psend} : (\mathbf{steps}_j' = 1) + (1 - \mathbf{psend}) : (\mathbf{steps}_j' = 0)$ ; models the basic gossip strategy: the node receiving the message will forward it with probability  $\mathbf{psend}$ , and discard it with probability  $1 - \mathbf{psend}$ .

*Rewards.* As introduced in Section 3, PRISM allows us to specify rewards (or costs), associated to both states and transitions. In order to study the energy performance of the networks, we associate a cost to each transition. In particular, for each transition [ci] (meaning that  $P_i$  is sending a message) we verify which transmission power has been used for the transmission ( $\mathbf{s1p}_1$ ,  $\mathbf{s2p}_1$  or  $\mathbf{s3p}_1$ ), and we use the values 1 for radius 10 m, 1.5 for radius 15m and 2 for radius 20m.

We are interested also in studying how many retransmissions the sender must perform before the communication is successfully completed. In order to do so, we introduce another reward, simply assigning 1 to each transition tagged with [round].

Formally, rewards are written as follows:

```
rewards "rounds"
[round] true : 1;
endrewards
```

```
rewards "costs"
[c1] s1p1 : 1;
[c1] s2p1 : 1.5;
[c1] s3p1 : 2;
[c2] s1p2 : 1;
[c2] s2p2 : 1.5;
[c2] s3p2 : 2;}
[c3] s1p3 : 1;
[c3] s2p3 : 1.5;
[c3] s3p3 : 2;}
. . .
endrewards
```

## 5 Simulations and Results

In this section we show some numerical results obtained using our model for the analysis of connectivity and performance properties of wireless networks. As usual for large models, we use statistical model checking, using the discrete-event simulator of PRISM.

We show how, using a cognitive process, which is able to dynamically adjust the transmission power of a node depending on the relative positions of the surrounding ones, it is possible to improve the performance of the network, guaranteeing a high level of connectivity, while limiting the energy consumption.

In the following examples, we use the same network that we have seen in Section 4, and we set the node  $P_{23}$ , i.e., the red node in Figure 1, as the final destination for the communications, while we change the sender node, in order to study how the performance of the network depend on the relative distance between sender and receiver. In particular we will show numerical results using as sender either the node  $P_{17}$  or the node  $P_6$ , i.e., the blue and yellow nodes in Figure 1, respectively.

We compare the connectivity and the power consumption of the cognitive network with other networks having exactly the same topology and using the same gossip strategy, but with a fixed transmission power.

### 5.1 Reachability Property

We first study the reachability properties of the system, i.e., the probability to reach a successful state of the model, which corresponds to the correct reception of a message by the final destination of the network.

In our PRISM representation of the model, since `steps23 = 1` means that the node  $P_{23}$  has correctly received the message, the formula which represents the success of the communication is

$$\text{formula goal} = (\text{steps23} = 1);$$

and the property that we are interested in verifying is

$$P=?[F \text{ goal}]$$

which gives us the probability that the sender and the receiver nodes will eventually complete their communication successfully.

As stated before, in order to perform statistical model checking, i.e., to get approximate results for the verification of properties, we use the PRISM simulator, that relies on Monte Carlo simulations. As we expected, since we assume that the sender node may retransmit a possibly infinite number of times, the probability to reach the goal state was correctly computed as 1.0 for all the network configurations, where the confidence interval was  $+/- 0$ , based on a confidence level 95%. This result ensures us that, in our setting, using a fixed transmission power or dynamically changing the transmission power, depending on the surrounding environment, does not affect the network connectivity. Moreover, this result ensures that a message will always reach the destination in a finite number of steps.

Table 3: Results for Energy Costs, Distance = 28,3 m

VariableRadius		FixedRadius = 15	
<b>psend</b>	<b>cost</b>	<b>psend</b>	<b>cost</b>
0.65	26.15733	0.65	25.5838
0.7	23.741	0.7	24.2405
0.75	22.5360	0.75	22.5333
0.8	20.7675	0.8	20.2982
0.85	18.2167	0.85	17.8995
0.9	15.7207	0.9	15.8523
0.95	13.3402	0.95	13.3570
1.0	11.21633	1.0	10.8015

## 5.2 Energy Cost Properties

As stated before, it is possible to analyse the performance of the network, in terms of energy consumption. As already introduced in Section 4, the transmission radius of a node in a wireless network is usually strictly related to its transmission power. In the literature we can find several formulas to estimate both reception and transmission energy costs (see, e.g., [12], [16]).

Here we abstract from those possible formulas, and we simply assign to each transmission the correspondent transmission radius as a reward. Notice that this is a choice that doesn't affect the complexity of the model or of its analysis. Moreover, we do not consider the energy spent for receiving data or to move, since the former is usually a fixed quantity, which does not depend on the actual activity of the node, and the latter usually come from a different power source, e.g., the legs of the mobile device user.

Again, we analyse the costs using statistical model checking. The reward property that has been studied is:

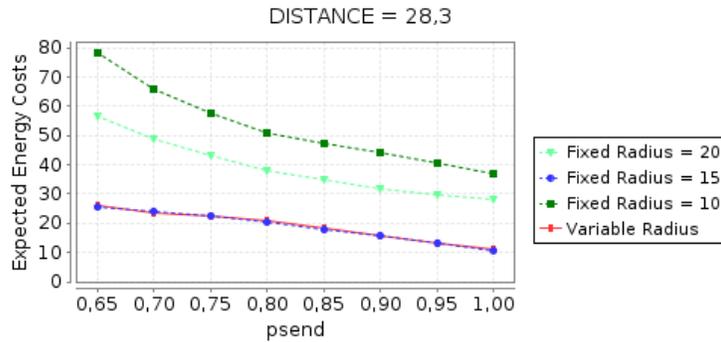
$R\{\text{'costs'}\}=?[F \text{ goal}]$

As in the previous case, we used a Monte Carlo simulation, and we obtained a maximum confidence interval of 2 – 3% with respect to the averages, based on a confidence level of 95%.

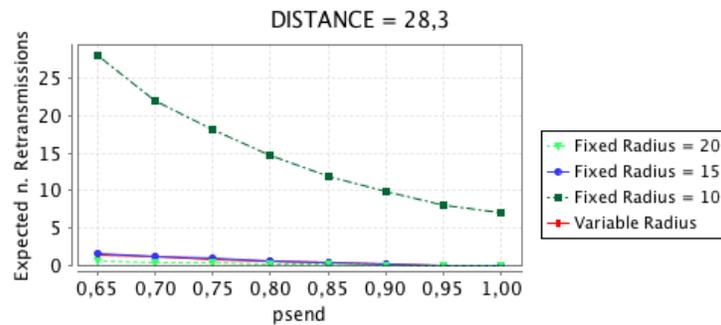
The results for a distance of 28.3m are shown in Figure 2.(a).

We notice that, while with a fixed radius of 10 or 20m, the energy costs of the communications critically increase, especially for small value of the gossip probability **psend**, using cognitive processes, or a fixed radius of 15m, the performance is consistently improved. Since the curves for the variable radius and the fixed radius 15m almost overlap, Table 3 reports the results in detail.

We analyse the average number of retransmissions, after the first one, that the sender node must perform to complete the communication with the receiver node, since it is useful to better understand the results of the previous reward



(a) Energy Costs

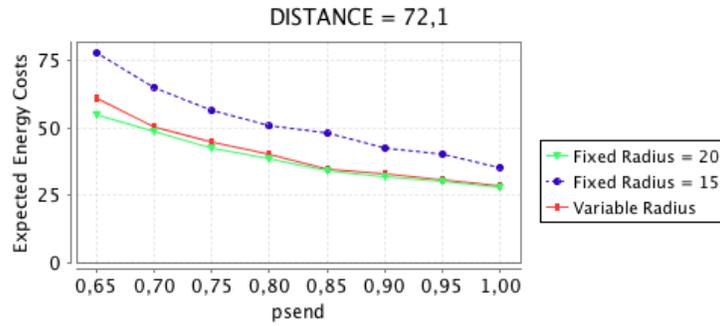


(b) Expected Number of Sender's Retransmissions

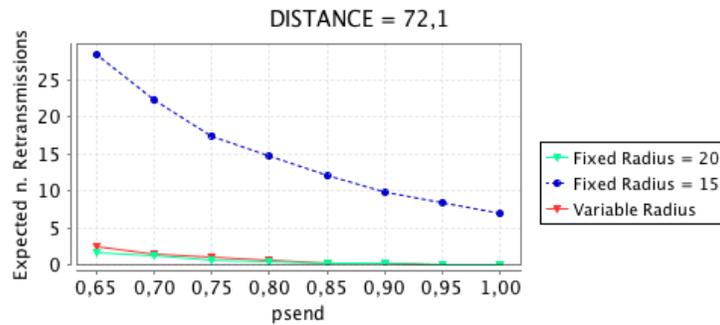
Fig. 2: Distance between sender and receiver: 28,3 m

property verification. Figure 2.(b) shows some results for this kind of analysis. Notice that, by fixing the radius to the maximum value, on average the communication reaches the successful state after less than 1 retransmissions. As an instance, the result for  $psend = 0.65$  is 0.67567. Here the energy waste is given by the high power employed for each forwarding, rather than by the number of transmissions to reach the success. Again the curves for the Variable Radius, and the Fixed Radius 15m are almost overlapping. This result lead us to the conclusion that, with this particular network configuration, if the processes can dynamically choose their transmission radius, depending on the neighbours' positions, the average radius will be 15m.

We now perform the same kind of analysis changing the sender. In this case the distance between sender and receiver is 72,1m. Figure 3 shows the results for energy consumption: in this case, with a fixed radius of 15m ,the energy performance of the network critically deteriorates. However, the results for a fixed radius 20m and a variable radius are similar. Table 4 gives the precise values for each  $psend$ . Notice that results for a fixed Radius of 10m are not



(a) Energy Costs



(b) Expected Number of Sender's Retransmissions

Fig. 3: Distance between sender and receiver: 72,1 m

reported: this is due to the fact that the power needed for small values of  $psend$  is very high and this would have led to an unreadable graph.

Again the analysis of the number of retransmissions by the sender nodes is helpful to understand the behaviour of the network: the curves for a fixed radius and a variable radius are similar. For  $psend = 0.65$  we have, on average, 1.6834 retransmissions for the fixed radius network, and 2.519 for the cognitive networks, while for  $psend = 1.0$  we have 0 on average for both the network configurations), meaning that, for a larger distance a fixed radius 20m is close to the ideal value of the transmission radius to guarantee the energy performance optimisation.

The results prove that, using a fixed radius, the performance of the network strictly depends on the relative positions of the sender and the receiver, while using a variable radius, we always get a power consumption that is closed to the minimum (that is closed to the fixed radius 15m in the first case, and to the fixed radius 20m in the second case).

Table 4: Results for Energy Costs, Distance = 71,2 m

VariableRadius		FixedRadius = 20	
<b>psend</b>	<b>cost</b>	<b>psend</b>	<b>cost</b>
0.65	60.9090	0.65	54.7933
0.7	50.61383	0.7	48.95267
0.75	44.64067	0.75	42.7287
0.8	40.1177	0.8	38.3973
0.85	34.9950	0.85	34.2673
0.9	32.8725	0.9	32.1087
0.95	30.8292	0.95	30.2807
1.0	28.6423	1.0	28.1893

## 6 Conclusion

In this paper we have presented a probabilistic model for a class of cognitive networks in a wireless setting, in which nodes dynamically choose their transmission power, using data collected from the network itself. We have shown how this model can be encoded in the PRISM language, allowing for the analysis of its performances and for the verification of properties of its behaviour. Moreover, we have used that kind of analysis to compare the energy efficiency of those networks with others based on different strategies, namely ones in which a static transmission power is set. We have given some numerical results about this comparison, and we have concluded that cognitive-networks-based strategies could be effective in the analysed setting.

*Future works.* As a further enhancement of our model, we plan to consider more sophisticated routing protocols, and different decision strategies as well. On the other hand, further simplifications of the model could lead to a faster solution, even for models with a greater number of nodes. Moreover, the analysis of different kinds of rewards, such as latencies or throughputs, could, and should, be performed in order to better understand any possible advantage or drawback of a power allocation strategy in wireless settings.

## References

1. A.G. Dimakis, A.D. Sarwate, , and M.J. Wainwright. Geographic Gossip: Efficient Aggregation for Sensor Networks. In *Proc. of the 5th international conference on Information processing in sensor networks*, pages 69–76. ACM, 2006.
2. J. Scott Donald and Alec Yasinac. Dynamic probabilistic retransmission in ad hoc networks. In *Proc of the Int. Conference on Wireless Networks (ICWN04)*, pages 158–164. CSREA Press, 2004.
3. A. Fehnker and P. Gao. Formal Verification and Simulation for Performance Analysis for Probabilistic Broadcast Protocols. In *Ad-Hoc, Mobile, and Wireless Net-*

- works, volume 4104 of *Lecture Notes in Computer Science*, pages 128–141. Springer Berlin Heidelberg, 2006.
4. C. Fortuna and M. Mohorcic. Trends in the Development of Communication Networks: Cognitive Networks. *Computer Networks*, 53(9):1354 – 1376, 2009.
  5. E. Gelenbe and R. Lent. Power-aware ad hoc cognitive packet networks. *Ad Hoc Networks*, 2(3):205 – 216, 2004.
  6. Z.J. Haas, J.Y. Halpern, and L. Li. Gossip-based Ad Hoc Routing. *IEEE/ACM Trans. Netw.*, 14(3):479–491, 2006.
  7. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
  8. J. Mitola III. *Cognitive Radio - An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2000.
  9. L. Guo J. Wang and G. Zhao. Study on Formal Modeling and Analysis Method Oriented Cognitive Network. In *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on*, volume 2, pages 402–405, 2012.
  10. M. Z. Kwiatkowska, G. Norman, and D. Parker. Prism 4.0: Verification of probabilistic real-time systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *CAV*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011.
  11. G. Norman M. Kwiatkowska and D. Parker. Advances and Challenges of Probabilistic Model Checking. In *48th Annual Allerton Conference on Communication, Control, and Computing*, pages 1691–1698. IEEE, 2010.
  12. T. V. Madhav and N.V.S.N. Sarma. Maximizing Network Lifetime through Varying Transmission Radii with Energy Efficient Cluster Routing Algorithm in Wireless Sensor Networks. *International Journal of Information and Electronics Engineering*, 2(2):205–209, 2012.
  13. T. Mahmoodi. Energy-aware routing in the cognitive packet network. *Performance Evaluation*, 68(4):338 – 346, 2011.
  14. P. Santi. Topology Control in Wireless Ad Hoc and Sensor Networks. *ACM Computing Surveys (CSUR)*, 37(2):164–194, 2005.
  15. W. J. Stewart. *Probability, Markov Chains, Queues, and Simulation*. Princeton University Press, UK, 2009.
  16. O. Younis and S. Fahmy. HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *Mobile Computing, IEEE Transactions on*, 3(4):366–379, 2004.
  17. Hongqiang Zhai and Yuguang Fang. Physical carrier sensing and spatial reuse in multirate and multihop wireless ad hoc networks. In *Proc. of INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, pages 1–12, 2006.