Ad Hoc Networks 10 (2012) 789–802

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Dynamic cooperator selection in cognitive radio networks

Nemanja Vučević [a,*,1], Ian F. Akyildiz [b], Jordi Pérez-Romero [a]

[a] Signal Theory and Communications Dept., Universitat Politècnica de Catalunya, Barcelona 08034, Spain
[b] Broadband Wireless Networking Lab., School of Electrical and Computer Engineering Georgia Institute of Technology, Atlanta, GA 30332, USA

ARTICLE INFO

Article history:
Available online 24 February 2011

Keywords:
Cognitive radio networks
Cooperation
Reinforcement learning
Reliability
Security

ABSTRACT

The primary objective of cooperation in cognitive radio (CR) networks is to increase the spectrum access efficiency and improve the network performance. However, Byzantine adversaries or unintentional erroneous conduct in cooperation can lead to destructive behavior of CR users that can decrease their own and others' performances. This work presents a dynamic solution for cooperation reliability in conditions with constraints typical for a CR network. Specifically, in CR networks, the information on the success of cooperation can be limited only to cases with interference; when malicious, cooperators can be completely non-correlated and can alter behavior; and the set of available cooperators can dynamically change in time. In order to face these challenges, each CR user autonomously decides with whom to cooperate by learning cooperators behavior with a reinforcement learning (RL) algorithm. This RL algorithm determines the suitability of the available cooperators, and selects the most appropriate ones to cooperate with the objective to increase the efficiency of spectrum access in CR networks. The simulation results demonstrate the learning capabilities of the proposed solution and especially its reliable behavior under highly unreliable conditions.

1. Introduction

Cognitive radio (CR) technology, as the core of CR networks, is a promising solution to deal with the problem of spectrum scarcity and low spectrum use associated to classical fixed spectrum assignment schemes [1]. For a proper operation, CR networks need to perceive the behavior of primary users (PUs) in their assigned frequency bands and perform opportunistic spectrum access without or with minimal interference to them. To this end, CR networks need to be capable to learn from the environment and to dynamically adapt to environment conditions in accordance with the heterogeneity and randomness in PUs behavior. Besides, multiple CR networks that can be contending on the same spectrum resources introduce additional complexity in the process.

Opportunistic spectrum access procedures such as spectrum sensing, spectrum decision, spectrum sharing and spectrum mobility require that CR users continuously gather and process information. Cooperation is introduced in CR networks [2] in order to increase the efficiency of the network (e.g. time and energy consumption for obtaining and processing information). Despite the fact that the purpose of cooperation is to improve the overall network performance, it may also introduce malicious behavior and unreliability in CR networks.

CR networks need not only to dynamically adapt to the changes in the radio environment, but also they need to be resistant to various types of threats and system errors. Jammers, data falsification, or denial of service, are some of the threats in CR networks [3–6]. Different defenses can be developed to prevent these attacks. A first line of security is aimed to give protection against the outside threats (i.e., intrusions by unauthorized users). A second

[*] Corresponding author. Tel.: +34 934010738; fax: +34 934017195.
E-mail addresses: vucevic@tsc.upc.edu (N. Vučević), ian@ece.gatech.edu (I.F. Akyildiz), jorperez@tsc.upc.edu (J. Pérez-Romero).

[1] This work was conducted during his stay at the BWN Lab, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, in 2009. Short version of this paper has been presented in [18].

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.01.020

line of defense is against the so-called Byzantine threats (i.e., traitors among cooperators). Cooperative CR networks are especially vulnerable to Byzantine attacks, as the CR users rely on information obtained from others. The cooperation reliability in CR networks may also be jeopardized by unintentional malicious effects (i.e., from the lack of reliability of devices or systems). The cooperation in CR networks needs to be performed with unknown or known cooperators that may change their behavior, in conditions where the knowledge about the success of the cooperation is limited. Autonomous solutions which are capable to learn and react dynamically according to the degree of reliability should be developed to deal with these issues in real time.

This paper addresses the reliability problem in cooperative CR networks [18]. It presents a solution that determines the reliability of the cooperators and selects which are appropriate for cooperation. The cooperator evaluation and selection in the proposed solution is based on reinforcement learning (RL) [7] which is a branch of machine learning envisaged as a good candidate for dynamic control and adaptation in CR networks [8]. Specifically, CR networks are likely to rely on learning, which makes the learning from false beliefs one of the threats to a CR network (learning in erroneous or intruded conditions) [3]. Therefore, the learned information should not be permanent, i.e., the learned beliefs should expire in due time in order to disable long-term malicious effects [5]. The proposed solution uses exploratory properties of reinforcement learning to provide the ability to relearn the changes and react through appropriate reconfigurations.

The rest of this paper is organized as follows. In the next section, the detailed problem formulation typical for CR networks is given. In Section 3, the framework for the proposed solution is described, followed by the algorithm description in Section 4. Section 5 contains simulation results. Finally, the main conclusions are summarized in Section 6.

2. Problem statement

CR users are opportunistically utilizing the network resources, and their incorrect actions may result in serious network capacity degradation. Cooperation is usually needed in these cases to ease the resources utilization. In such a case, a CR user collects information (the advices) from cooperators, and makes a decision about its next actuation regarding opportunistic spectrum access. CR user and its cooperators may be a CR base station or a CR terminal operating in either centralized or ad hoc CR networks. The constraints explained in this section are characteristic for the cooperative spectrum sensing in CR networks, where the cooperators provide information about spectrum availability and the CR user decides whether to access a given spectrum or not.

2.1. Cooperation model

In the problem considered here a CR user needs to decide on the following unknown hypothesis in order to perform spectrum access: H_0 (spectrum access not possible), H_1 (spectrum access possible). In a cooperative scenario, CR user receives information messages from the cooperators in order to make a spectrum access decision. The communicated message can contain measured or estimated values in case of "soft decision", or it can contain a local advice (local decision) that each cooperator makes (usually a binary decision 0 or 1) in case of "hard decision". Hard decisions reduce the communication overhead, but are more challenging for final decision making since they provide only minimum information.

This paper considers the case with hard decision. This means the cooperator i makes a local decision and forwards it as an advice $x(i)$ to the CR user: advice $x(i) = 1$ means that local decision is H_1 and $x(i) = 0$ means local decision is H_0. For example, if the measured signal power from the PU is lower than a threshold, a cooperator would send $x(i) = 1$ meaning that the PU is not there so spectrum access is possible (H_1). Based on combining the received information from the cooperators, the CR user takes a global decision X (i.e. $X = 0$ for H_0 and $X = 1$ for H_1).

The usual data fusion rule for hard decisions in the literature is "K out of M" rule [9]. This means that at least K out of M received advices have to be H_1 so that the global decision X is H_1 (i.e., $X = 1$). Otherwise global decision is H_0 (i.e. $X = 0$):

$$X = \begin{cases} 1, & \sum_{i=1}^{M} x(i) \geqslant K \\ 0, & \sum_{i=1}^{M} x(i) < K \end{cases}. \tag{1}$$

Two special cases of "K out of M" rule are the extreme options: AND rule when $K = M$, and OR rule when $K = 1$.

It is also assumed in the paper that N cooperators are available. However, not all of them are actually necessary for cooperation. Then, the CR user can select only M ($M \leqslant N$) cooperators based on their suitability to achieve the desired goal (see Fig. 1). The suitability of cooperators should include both aspects of reliability and accuracy. In this paper, dynamic tracking of this suitability is carried out by means of reinforcement learning.

Based on the above discussion, the proposed method needs to solve the following points:

- Decide how many cooperators to use (i.e. decide the value of $M \leqslant N$).
- Learn how to select reliable cooperators (i.e. decide which are the most convenient M cooperators out of the N available ones).
- Make a decision from the M collected advices (i.e. apply "K out of M" rule).

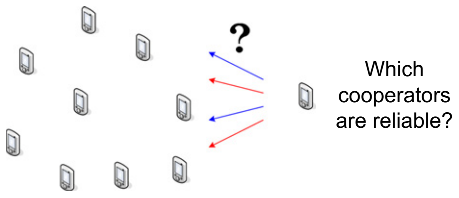

Fig. 1. The dilemma: Which cooperators are reliable?

2.2. Malicious behavior in cooperation

The malicious conduct in cooperation may be intentional and/or unintentional:

- The intentional malicious behavior is due to the Byzantine adversary that pretends to be a friend and uses its privilege to achieve its own desired goal.
- The unintentional incorrect behavior may be from:
  - Cooperators that are manipulated by other opponent systems and are unaware of their malicious effects (e.g., sensors in range of jammers giving impression of spectrum occupancy).
  - Technological limitations (i.e. hardware and software errors, failures and limitations of the sensing devices).
  - Environment conditions (e.g. sensors in high shadowing zone).

Additional complexity in the detection of the malicious cooperators comes from the fact that cooperators may change their behavior in time. A reliable cooperator during a certain period may turn into a malicious cooperator in the future, and vice versa. Thus, a system needs to be capable of capturing such dynamic changes. This variable behavior may especially be present in cases when cooperators are occasionally manipulated by another enemy system, or when the attacker is trying to gain the trust of the victim.

2.3. Erroneous advices and decisions

Due to the malicious behavior, two types of erroneous advices are possible in such a system: erroneous positive advice when the spectrum access is not possible ($x(i) = 1|H_0$) and erroneous negative advice when the spectrum access is possible ($x(i) = 0|H_1$). For a cooperator i, the probabilities of erroneous advices are:

$$p_{err}(i) = \Pr\{x(i) = 0|H_1\}, \tag{2}$$

$$q_{err}(i) = \Pr\{x(i) = 1|H_0\}. \tag{3}$$

The probabilities of erroneous advice (2) and (3) influence the final error probabilities of decision, i.e. the probabilities of erroneous non-actuation (P_{ERR}) and erroneous spectrum access (Q_{ERR}), i.e. interference. In particular, for “K out of M” rule, these are:

$$P_{ERR} = \Pr\{X = 0|H_1\} = 1 - \sum_{j=K}^{M} \Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = j|H_1\right\}, \tag{4}$$

$$Q_{ERR} = \Pr\{X = 1|H_0\} = \sum_{j=K}^{M} \Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = j|H_0\right\}. \tag{5}$$

An optimal system will tend to minimize both $P_{ERR} \to 0$ and $Q_{ERR} \to 0$. However, note that an erroneous spectrum access can lead to interference with primaries, which is more destructive than an erroneous non-actuation. Therefore, it is much more important to keep Q_{ERR} within very low limits than P_{ERR} to assure minimum interference.

An additional constraint in CR networks is that the evaluation of the actuation is only possible once the spectrum access has actually happened (e.g., a collision occurs after making an access when the PU is active). However, an erroneous non-actuation (i.e. a missed opportunity of spectrum access) is usually not possible to detect, as the CR user does not have a success feedback when idle. In other words, there is no feedback when $X = 0$ so that it is not possible to evaluate P_{ERR}. Thus, the way to minimize P_{ERR} is to maximize the spectrum access, i.e. to maximize $\Pr\{X = 1\}$.

2.4. Summary of considered challenges

Based on the above considerations, the goal of the proposed framework to deal with the problem of malicious behavior in cooperation is to maximize $\Pr\{X = 1\}$, but at the same time preserve Q_{ERR} within very low limits. The following additional challenges are addressed in the problem considered in this paper:

1. The local decision of the cooperators is unknown and independent in each cooperator (i.e. cooperators are heterogeneous).
2. This paper considers a worst case scenario where probabilities $q_{err}(i)$ and $p_{err}(i)$ are unknown, uncorrelated, and may vary dynamically and independently of each other.
3. Probabilities $\Pr\{H_1\}$ and $\Pr\{H_0\}$ are unknown and may change with time.
4. There is no knowledge on the success of decisions not to actuate, i.e. when $X = 0$.
5. The number of potential cooperators N can vary with time.

2.5. Related work

To the best of authors’ knowledge, the problem with all the aforementioned assumptions and conditions has not been addressed in the literature up to date. However, as the proposed solution combines suitability evaluation, data fusion and decision making in unreliable conditions, some partially related studies are briefly discussed in the following.

The classification of how good a cooperator is often results in trust evaluation, and is commonly compared with human behavior characteristics [10]. The application of trustworthiness for reliable path selection in ad hoc and wireless sensor networks is present in the literature, e.g. [11,12]. Existing studies distinguish between direct and indirect trusts, and address the influence of reputation, respect and rumors in trust construction. Similarly, the CR user in this paper also evaluates and assigns a grade of suitability to cooperators, reflecting their reliability.

Studies on how to use expert advices address the problem of how to use several experts to guess the unknown hypothesis as close as possible to the best expert’s guesses [13,14]. However, assumptions taken in the previous studies do not consider all the constraints and limitations explained in the previous section (challenges 1–5) at the same time. Moreover, the behavior of the best cooperator does not guarantee that $\Pr\{X = 1\}$ is maximized and does not preserve Q_{ERR} within very low limits with higher priority.

The problem of maximizing $\Pr\{X = 1\}$ while Q_{ERR} is preserved within low limits appears as a Neyman–Pearson detection problem in the literature [9], where the data fusion from sensors should optimize the decision following the "K out of N" rule. Due to their high complexity, the methods used for solving this problem have only limited applications in cases with small number of highly similar sensors, e.g. [9,15]. These solutions are also not applicable for the entire set of challenges 1–5 mentioned before.

3. Proposed framework

In the proposed framework, N potential cooperators offer advices to a CR user and it needs to determine which of the potential cooperators to use in order to maximize its goals. The final decision of CR used is computed as a combination of decisions of several available cooperators. Therefore, how much convenient for cooperation is a given cooperator, depends on its grade of reliability, on the reliability of other cooperators, and on the number of needed cooperators at a certain moment. Thus, even when one cooperator does not change its behavior, the variation of general conditions can make it more or less appropriate in different time instants.

The evaluation of cooperation suitability depends on the final outcomes, Q_{ERR} and P_{ERR}, when a cooperator is used. The proper learning mechanism should assure that the malicious cooperators are not used. A proper cooperator selection should also reduce the signaling load of cooperation process, as the unnecessary cooperators can be excluded from cooperation, while the goals are achieved with high accuracy.

3.1. Suitability list and cooperator selection

Each CR user performs its own evaluation and cooperator selection based on its own experience. In order to perform spectrum access, a CR user makes a suitability list for the N cooperators available to him at that moment. The appearance and disappearance of cooperators (e.g., due to mobility) does not affect the algorithm, as they can easily be added or removed from the list. Each of these cooperators is characterized with a learning parameter p_i, which is used to define their suitability π_i.

In Fig. 2, the general framework is presented where CR user selects M cooperators from the suitability list and starts an application period, which is a period in which the cooperation is carried out with one group of M cooperators. During this period T_A decisions are made about whether to access the spectrum or not. For each decision the resource access depends on the current advices of the M cooperators in use applying the "K out of M" rule. During one application period the group of M cooperators does not change.

The reward is computed at the end of each application period. This reward reflects the success of the decisions that a CR user has made. The reward will have two components r and ρ as will be explained in Section 4. Then, the RL algorithm is executed to update the controlled parameters (learned parameters) p_i for each cooperator i used in that period. The learned values p_i are the key parameters in defining the suitability of each cooperator. Once the p_i values are updated, the suitability of each cooperator π_i in the system is computed. Based on this, a new set of cooperators is selected for the cooperation in the following application period.

An additional degree of flexibility of the algorithm can be achieved by adapting M to CR user's needs through time. The change of M can be performed based on the tracked values of erroneous access (interference) during previous application periods. This process is independent from the RL algorithm that maintains the suitability list.

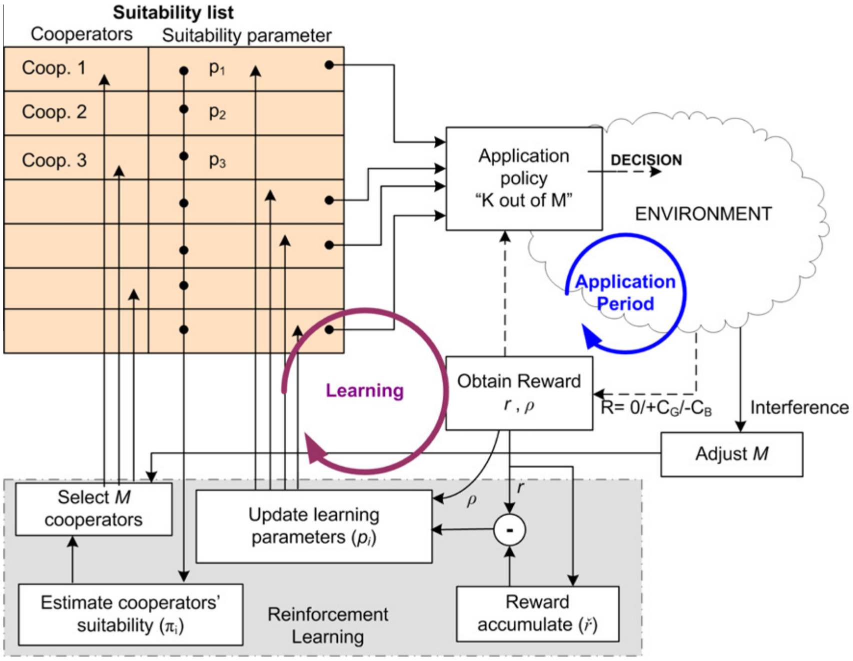

Fig. 2. The cooperator selection framework based on reinforcement learning and cooperation suitability list.

3.2. Cooperation communication

This work assumes that a small bandwidth control channel is available to exchange the cooperation messages. A CR user either invites cooperators for cooperation or listens to all cooperators and just ignores the ones denoted as unreliable. In the first case, the invitation to cooperate is performed only once at the beginning of each application period, and only if there has been a change in the set of used cooperators.

In case cooperators are invited for cooperation, the CR user also needs to perform occasional checkup of possibly new cooperators. This can be done only when the current performance is not satisfactory for a long period, or when a lot of cooperators have disappeared. Note that the CR user can optionally set a maximum limit for the number of cooperators in a suitability list. With this option it can also occasionally remove highly unreliable cooperators and replace them with the new ones.

This paper assumes direct data gathering from the cooperators. In general, data gathering can also be indirect (i.e. through an intermediate cooperator). However, with indirect data gathering, the proposed solution remains unchanged. In that case, two options would be possible:

- The intermediate cooperator makes a unique local decision from gathered data and forwards it to a CR user. This makes only the intermediate cooperator visible in suitability list.
- The intermediate cooperator forwards all the gathered data. This opens the possibility to data falsification [6], however, it would only change the original advice error probabilities of the indirect cooperators.

4. Learning and decision processes

In this section learning mechanism and the application policy are first described for a fixed number of used cooperators *M*. Afterwards, the framework is extended for a variable value of *M*.

4.1. Cooperator selection

The learning capabilities of the proposed solution rely on RL [7] algorithms to maintain the suitability parameter of the prospective cooperators. RL is a branch of machine learning where an agent through interaction with environment learns and decides on actions in order to maximize some long term reward. The reward is an input that the agent receives from the environment, which represents the quality of the actions taken by the agent, and that should reflect the goals and needs of the system.

This work starts from the actor-critic learning and the REINFORCE algorithm [16]. Actor critic methods require minimal computation in order to select actions, and use separate data structures for control policy (the "actor") and the value function (the "critic"). The task of the "critic" is to "criticizes" the actor's behavior, i.e. to determine after each action whether the results have gone better or worse than expected [7]. This is carried out based on the interaction with the environment through a reward function. The critic in this paper is performing the global reward accumulate, and the comparison of the current reward value to this accumulate. Based on the "critic", the "actor" makes the update of the learning parameters p_i that will determine the suitability, and selects the *M* cooperators for the next application period. Fig. 3 presents the learning process of the algorithm. Details of the learning algorithm from the figure are given in continuation.

The global reward function *r* is accumulated throughout the application period. For one application period, the reward *r* is

$$r = \frac{1}{T_A} \cdot \sum_{t=1}^{T_A} [X_t \cdot (\alpha_t \cdot C_G + (1 - \alpha_t) \cdot C_B)]. \tag{6}$$

This reward presents the sum of the rewards and penalties accumulated during one application period. Index *t* corresponds to each of the T_A decisions made by the CR user. Only when the decisions are to actuate (i.e. $X_t = 1$) the reward contribution is considered for that decision. Constants C_G and C_B define the ratio between reward increase and penalization for correct and wrong spectrum access. Parameter α_t equals:

$$\alpha_t = \begin{cases} 1, & X_t = 1 | H_1 \\ 0, & X_t = 1 | H_0 \end{cases}. \tag{7}$$

The reward obtained in Eq. (6) represents the overall actuation performance during one application period. Additionally, the reward correction ρ_i is also computed in every application period for each cooperator *i* among the *M* active cooperators:

$$\rho_i = \frac{1}{T_A} \sum_{t=1}^{T_A} [(1 - x_t(i)) \cdot X_t]. \tag{8}$$

This reward represents the number of times the cooperator *i* gave non-actuation advice ($x(i) = 0$) when decision was to access the resource ($X = 1$), divided with the number of decisions in one application period. This value helps the learning process by penalizing the cooperators that often advise not to access the resource when the rest $M - 1$ cooperators advise to do so. This is similar to the supervisor that contributes to faster learning [17].

Each application period is followed by the learning process. The update of the learning parameter for each of the active cooperators is:

$$p_i \leftarrow p_i + \beta \cdot (r - \breve{r} - \xi \cdot \rho_i) \cdot (1 - \pi_i). \tag{9}$$

Here, β ($0 < \beta \leqslant 1$) and ξ are positive constant parameters and π_i ($0 \leqslant \pi_i \leqslant 1$) is the suitability of the cooperator *i* to be selected for the cooperation. The values of β determine the influence of the recent RL decisions to future RL decisions, i.e. higher values β of give more influence to the more recent decisions. Parameter ξ determines the strength of the bias of the individual penalization. The value of this parameter should be a small number ($\xi \ll 1$) not to have big influence over the main reward *r*. The parameter $\breve{r}$ is the global reward accumulate, which is used as a reinforcement baseline or reward reference in the process of the behavior evaluation. After all the parameters p_i of the active cooperators have been updated, $\breve{r}$ is also updated as:

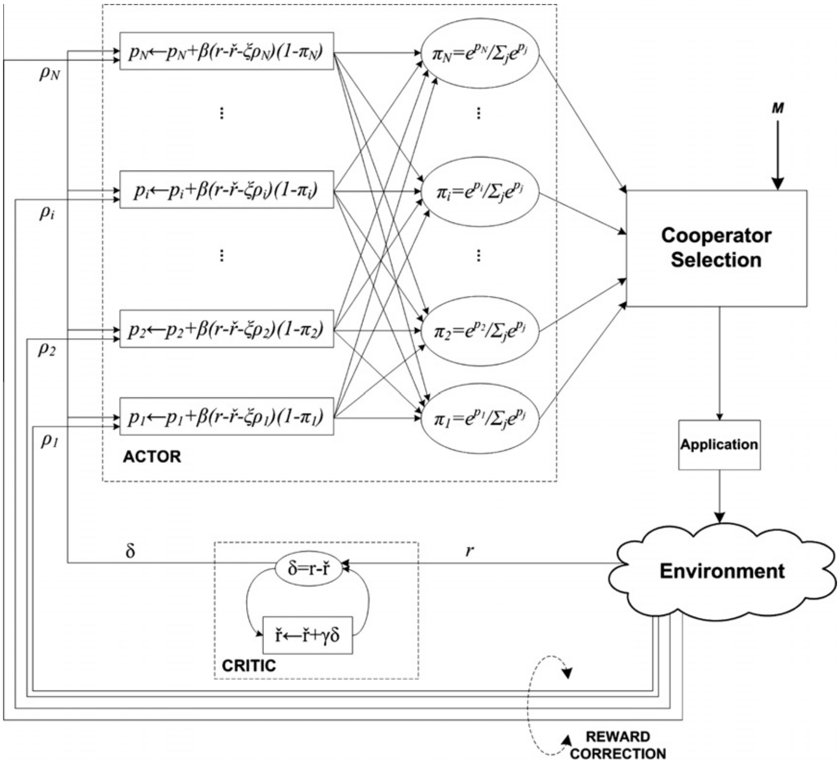
$p_N \leftarrow p_N + \beta(r - \check{r} - \xi\rho_N)(1-\pi_N)$
$p_i \leftarrow p_i + \beta(r - \check{r} - \xi\rho_i)(1-\pi_i)$
$p_2 \leftarrow p_2 + \beta(r - \check{r} - \xi\rho_2)(1-\pi_2)$
$p_1 \leftarrow p_1 + \beta(r - \check{r} - \xi\rho_1)(1-\pi_1)$
$\pi_N = e^{p_N}/\Sigma_j e^{p_j}$
$\pi_i = e^{p_i}/\Sigma_j e^{p_j}$
$\pi_2 = e^{p_2}/\Sigma_j e^{p_j}$
$\pi_1 = e^{p_1}/\Sigma_j e^{p_j}$
ρ_N
ρ_i
ρ_2
ρ_1
ACTOR
M
Cooperator Selection
Application
Environment
δ
r
δ=r-ř
ř←ř+γδ
CRITIC
REWARD CORRECTION

Fig. 3. The learning process.

$$\breve{r} \leftarrow \gamma \cdot r + (1-\gamma) \cdot \breve{r}, \tag{10}$$

where the parameter γ is constant, $0 < \gamma \leqslant 1$. The learning values p_i and the global reward accumulate $\breve{r}$ are the only values that need to be stored for the learning mechanism. Note that for $\gamma = 1$ the algorithm only uses the current reward to evaluate cooperator set. For values of γ close to 1 influence of latest rewards in the exponential averaging of $\breve{r}$ is more dominant, whereas for lower values of γ their influence is less dominant.

Finally, once the updates of parameters p_i are done, the suitability values π_i are computed for all cooperators:

$$\pi_i = \frac{e^{p_i}}{\sum_{j=1}^{N} e^{p_j}}, \tag{11}$$

M cooperators are randomly selected for cooperation for the following application period. Suitability value π_i presents the probability with which cooperator i will be selected among these M cooperators.

4.2. Application policy

The application policy defines how the advices of the M cooperators are interpreted by the CR user in order to make the decision X. As hard decisions are used, the policy applies the ‘‘K out of M’’ rule.

Error probabilities P_{ERR} and Q_{ERR} are defined in Eqs. (4) and (5), where it can be seen that for a given M, a value of K closer to M decreases Q_{ERR}, but increases P_{ERR} (Theorem 1 in the Appendix). As the primary objective of our solution is to maintain Q_{ERR} very low, $K = M$ can be the safest choice (i.e., all the cooperators have to decide $x(i) = 1$ in order to have $X = 1$). However, this paper starts from the value $K = M - 1$ as it allows one cooperator to advise $x(i) = 0$ and still to decide $X = 1$. This permits occasional exploration and faster learning of the RL algorithm on the cooperators that may be malicious and are among the selected M cooperators. In any case, other values of K could also be used, as it will be studied in Section 5.

Now, for the case $K = M - 1$ and $M \geqslant 2$, P_{ERR} and Q_{ERR} from Eqs. (4) and (5) become:

$$P_{ERR} = 1 - \left(\prod_{i=1}^{M}(1 - p_{err}(i))\right) \cdot \left(1 + \sum_{i=1}^{M} \frac{p_{err}(i)}{1 - p_{err}(i)}\right), \tag{12}$$

$$Q_{ERR} = \left(\prod_{i=1}^{M} q_{err}(i)\right)\left(1 + \sum_{i=1}^{M} \frac{1 - q_{err}(i)}{q_{err}(i)}\right), \tag{13}$$

where the advice error probabilities $p_{err}(i)$, $q_{err}(i)$ are defined in Eqs. (2) and (3).

4.3. Variation of M

A more adaptive system can be achieved if the number of used cooperators M can be variable. In particular, when Q_{ERR} is high, M should be increased. In turn, when Q_{ERR} is low, M may be decreased so that the P_{ERR} may be decreased more easily. Theorem 2 of the Appendix proves that adding one cooperator (with whichever value $q_{err} \neq 1$) can reduce Q_{ERR} when necessary. In that case P_{ERR} increases for sure (unless $p_{err} = 0$ of the new cooperator), so the algorithm relies on a proper learning mechanism to encounter the best M cooperators to maintain P_{ERR} value as low as possible. Similarly, in the cases when Q_{ERR} is sufficiently low it is possible to improve the value of P_{ERR} by reducing the number of cooperators M without increasing Q_{ERR} too much.

To carry out the adaptation of M, CR user tracks the amount of interferences it made over the last T application periods as follows:

$$S(T) = \frac{1}{T}\frac{1}{T_A}\sum_{s=-T}^{0}\sum_{t=1}^{T_A}[X_{s,t}\cdot(1-\alpha_{s,t})]. \tag{14}$$

Here, s stands for the index of the application period, whereas t is the index of the decisions in one application period. $X_{s,t}$ is the decision t made in the application period s, $\alpha_{s,t}$ is the parameter defined in Eq. (7) for the decision t in application period s, i.e. $1 - \alpha_{s,t} = 1$ when the spectrum is not free for CR user in application period s.The average $S(T)$ is independently tracked for the previous T_X and T_Y application periods. Two threshold values are defined to control M: D_H^U and $D_H^L (D_H^L < D_H^U)$. M is changed as:

$$S(T_X) > D_H^U \rightarrow M = M + 1, \tag{15}$$

$$S(T_Y) < D_H^L \rightarrow M = M - 1. \tag{16}$$

Note that longer periods T_X and T_Y will make more accurate averaging, but lower values of T_X and T_Y will allow a faster reaction to change M. The proposed mechanism will use smaller value T_X than T_Y ($T_X < T_Y$) so the change of M due to the interference (increment of M) can be performed after the average interference is measured over less application periods. On the other side, adjustment of P_{ERR} is of lower priority, so the decrement of M can be performed based on the average over more values. Threshold parameters D_H^U and D_H^L determine the reference values for the change of M based on interference.

Let us assume now that the number of the application periods from the last increment of M is T_X^*. Then, as long as $T_X^* < T_X$, instead of using condition from Eq. (15), the modified condition to increment M is used:

$$T_X^* < T_X \rightarrow S(T_X^*) > D_H^U \cdot (T_X + T_X^*)/2T_X^* \rightarrow M = M + 1. \tag{17}$$

Similarly, if M has been decreased before T_Y^* application periods, and as long as $T_Y^* < T_Y$, instead of (16), the modified condition for the next decrement of M is:

$$T_Y^* < T_Y \rightarrow S(T_Y^*) < D_H^L \cdot (2T_Y^* - T_Y)/T_Y^* \rightarrow M = M - 1. \tag{18}$$

The previously explained use of Conditions (17) and (18) is done in order to disable consecutive changes of M due to the values that already have contributed to a change of M. Thus, the learning algorithm has time to readapt itself to the new number of cooperators in use. Note that the mechanism uses unitary change for M values in order to perform a more gradual and stable adaptation, as the learning algorithm can adapt to each newly added or removed cooperator.

In order to avoid having very low values of M, the minimum number for M is set to be 3 whenever there are more than 3 cooperators available ($N \geqslant 3$). When this is not the case ($N < 3$), one or two cooperators may also be used. Note that for $M = 1$, it needs to be $K = 1$.

5. Simulation results

This section presents some simulation results to evaluate the performance of the proposed approach. First the operation of the learning process is presented in a particular example, followed by the simulation results in more complex scenarios.

5.1. Learning and reconfiguration

The purpose of this part is to demonstrate the operation of learning process. It is assumed that there are 10 cooperators ($C1$–$C10$) available and that the error probabilities $p_{err}(i)$ and $q_{err}(i)$ change as given in Table 1. After every 2000 application periods some of the probabilities change. Notation "–" indicates that the cooperator is unavailable in that period. The values for $p_{err}(i)$ and $q_{err}(i)$ are chosen to represent different illustrative behaviors of the cooperators. For this set of results, the spectrum availability is randomly generated with $\Pr\{H_1\} = \Pr\{H_0\} = 0.5$. Statistics are averaged over the last 200 application periods.

Table 2 summarizes the list of parameters used in simulations. Initial values for the learning mechanism are $M = 5$ (when M is variable) and $p_i = 0$. In case, new cooperators appear while the algorithm is already running, p_i takes a random value from the range ($\min\{p_j\}$, $\max\{p_j\}$) from the cooperators j in the suitability list.

In Fig. 4, an example of the adaptation process is shown for the case in which M is constant, $M = 5$. The probabilities to select the cooperators (π_i) change through time in accordance with the variation of the error probabilities $p_{err}(i)$ and $q_{err}(i)$. At the beginning more than five cooperators are performing well enough to be included in cooperation (e.g. cooperators $C3$–$C7$ which have low $p_{err}(i)$ or $q_{err}(i)$ and $C8$ that has low $p_{err}(i)$ and sufficiently low $q_{err}(i)$ to maintain low Q_{ERR} when combined with any four cooperators from $C3$ to $C7$. Afterwards, the learning mechanism is trying to correct possible degradation of performances that can happen each time one of the used cooperators increases the error probability (either $p_{err}(i)$ or $q_{err}(i)$). This can be observed, for instance, when $C7$ increases p_{err} from 0.01 to 0.99 after 2000th application period and, correspondingly, the algorithm decreases the probability to cooperate with it. Appearance and disappearance of cooperators is also followed by similar expected reaction. As it has been mentioned before, at the appearance of cooperators, the learning parameter $p(i)$ is randomly initialized. So when $C1$ and $C4$ appear at the same time at the 18,000th application period, the probability to use $C4$ is high, whereas the probability to use $C1$ is low. However, the quality in the behavior of these two cooperators is perceived and these values are corrected fast so that the probability to use $C4$, having actually a large value of p_{err}, is progressively decreased, while the probability to use $C1$, exhibiting good values of both p_{err} and q_{err}, is progressively increased.

In order to illustrate the need for an adaptive value of M, note that during the application periods from 6000th to 8000th there are only three cooperators with $p_{err}(i) = 0.01$

Table 1
Behavior of cooperators through simulation.

Application period		Cooperators									
		[C1	C2	C3	C4	C5	C6	C7	C8	C9	C10]
0	p_{err} =	[0.01	0.50	0.01	0.01	0.01	0.01	0.01	0.01	0.99	–]
	q_{err} =	[0.50	0.50	0.01	0.01	0.01	0.01	0.01	0.50	0.01	–]
2000	p_{err} =	[0.01	0.50	0.01	0.01	0.01	0.01	0.99	0.01	0.99	–]
	q_{err} =	[0.50	0.50	0.01	0.01	0.01	0.01	0.01	0.50	0.01	–]
4000	p_{err} =	[–	0.50	0.01	0.01	0.01	0.99	0.99	0.01	0.99	–]
	q_{err}=	[–	0.50	0.01	0.01	0.01	0.01	0.01	0.50	0.01	–]
6000	p_{err} =	[–	0.50	0.01	0.01	0.01	0.99	0.99	0.99	0.99	–]
	q_{err}=	[–	0.50	0.01	0.01	0.01	0.01	0.01	0.50	0.01	–]
8000	p_{err} =	[–	0.50	0.01	0.01	0.01	0.99	0.99	0.99	0.99	0.01]
	q_{err} =	[–	0.50	0.01	0.01	0.01	0.01	0.01	0.50	0.01	0.01]
10,000	p_{err} =	[–	0.50	0.01	0.01	0.01	0.99	0.99	0.99	0.01	0.01]
	q_{err} =	[–	0.50	0.01	0.01	0.01	0.01	0.01	0.50	0.01	0.01]
12,000	p_{err} =	[–	0.50	0.01	–	0.01	0.99	0.99	0.99	0.01	0.01]
	q_{err} =	[–	0.50	0.01	–	0.01	0.01	0.01	0.50	0.01	0.01]
14,000	p_{err} =	[–	0.50	0.01	–	0.01	0.01	0.99	0.01	0.01	0.01]
	q_{err} =	[–	0.50	0.01	–	0.01	0.50	0.50	0.50	0.50	0.01]
16,000	p_{err} =	[–	0.50	0.01	–	0.01	0.01	0.99	0.01	0.01	0.01]
	q_{err} =	[–	0.50	0.01	–	0.50	0.50	0.50	0.50	0.50	0.01]
18,000	p_{err} =	[0.01	0.50	0.01	0.99	0.01	0.01	0.99	0.01	0.01	0.01]
	q_{err} =	[0.01	0.50	0.01	0.50	0.01	0.50	0.50	0.50	0.50	0.01]

Table 2
List of parameters.

RL parameters						Parameters for adaptive M			
β	ξ	γ	C_G	C_B	T_A	D_H^U	D_H^L	T_Y	T_X
0.4	0.05	0.4	1	−10	20	0.001	10^{-5}	100	10

and one with $p_{err}(i) = 0.5$, whereas the rest of them have $p_{err}(i) = 0.99$. Then, there is no combination of $M = 5$ cooperators that can give lower P_{ERR} than 0.5 (see Eq. (12)). Similarly, for the application periods between 16,000th and 18,000th, the combination of available $q_{err}(i)$ values does not permit Q_{ERR} values as low as D_H^U for $M = 5$ (see Eq. (13)). So, in this period average Q_{ERR} is $E(Q_{ERR}) > 2D_H^U$ (Fig. 4c). Consequently, in these cases it is appropriate to modify the value of M.

In Fig. 5a, the performances (P_{ERR} and Q_{ERR}) are also presented for the case $M = 4$. Now, P_{ERR} is significantly lower between the 6000–8000th application periods. However, as expected, having one cooperator less can be a limitation for Q_{ERR}. So, between the 16,000th and 18,000th application periods, the average value of Q_{ERR} is even higher, $E(Q_{ERR}) \approx 6D_H^U$ (as indicated in Fig. 5a).

Finally, in Fig. 5b and c the performance results are given for variable M. Now, the variation of M allows lowering of P_{ERR} in periods in which Q_{ERR} is not close to the threshold (between 6000th and 8000th application periods). In the periods from 16,000th to 18,000th, the values of P_{ERR} are higher than in both simulations with fixed M. Nevertheless,

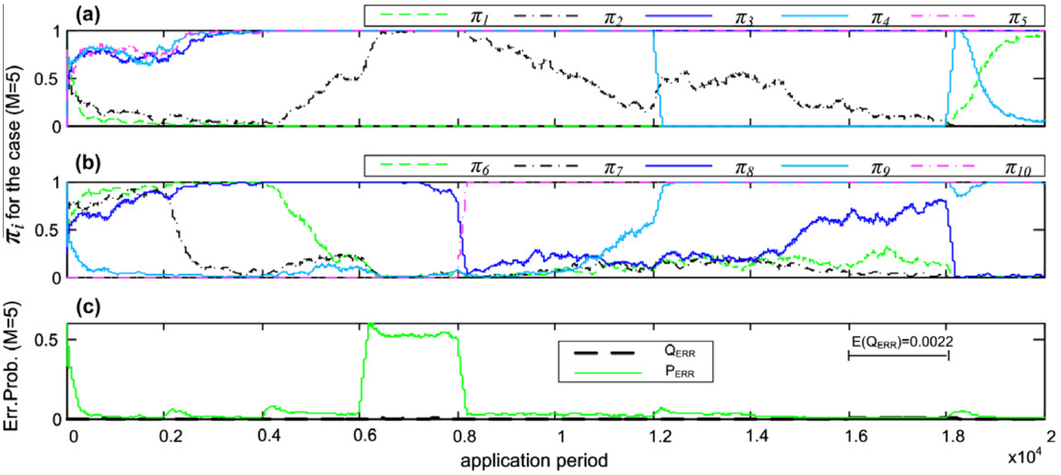

Fig. 4. (a and b) Suitability of different cooperators (probability to cooperate) and (c) performance results for $M = 5$.

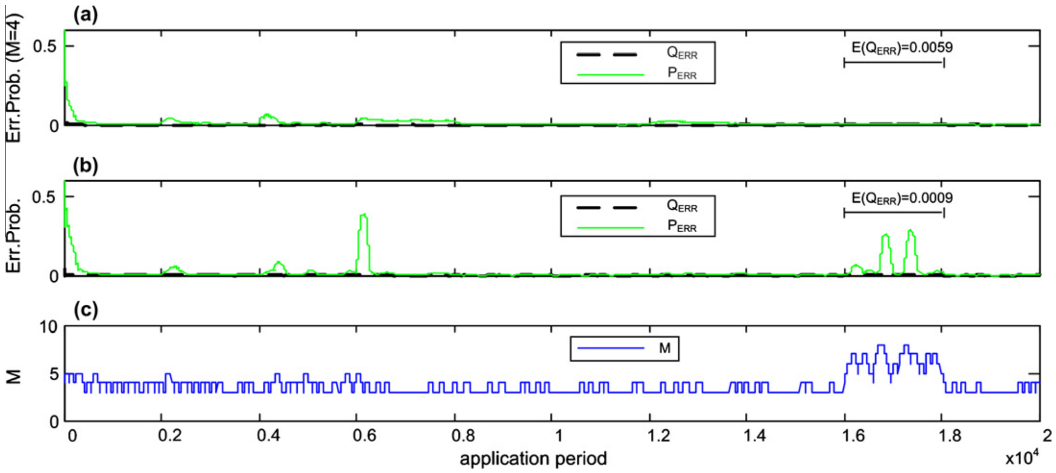

Fig. 5. (a) Performance results for $M = 4$; (b) performance results when M is variable and (c) corresponding M value.

this tradeoff is done so the average value of Q_{ERR} is now below the D_H^U, i.e. $E(Q_{ERR}) \approx 0.9D_H^U$.

As it is presented in Section 2, the primary objective was to maintain Q_{ERR} very low and then to minimize P_{ERR} as much as possible. The presented results show how the mechanism maintains Q_{ERR} low even in transitional periods (i.e., simulation initialization and preference changes when conditions worsen). At the same time, fast convergence of the RL algorithm lowers P_{ERR} relatively fast (e.g., after only ~300 application periods, even for $M = 5$, average P_{ERR} is below 0.05 without any previous knowledge - at session initialization).

5.2. RL convergence

In order to analyze convergence, simulations in specific and controlled scenarios have been performed. In particular, the test has been done with $N = 6$ cooperators and two different scenarios:

Scenario 1: There are N_1 cooperators that perform spectrum access blocking, i.e. $p_{err} = 1$, $q_{err} = 0$. The rest of the cooperators are ideal.

Scenario 2*:* There are N_2 cooperators that are stuck to state H_0 (spectrum free) and lead to interference, i.e. $p_{err} = 0$, $q_{err} = 1$. The rest of the cooperators are ideal.

In both scenarios the algorithm is considered to converge when the average values $P_{ERR} \to 0$ and $Q_{ERR} \to 0$. Fig. 6a presents the evolution of P_{ERR} in scenario 1 (note that in this scenario $Q_{ERR} = 0$) for different values N_1, whereas Fig. 6b presents the evolution of Q_{ERR} in scenario 2 (note that in this scenario $P_{ERR} = 0$) for different values N_2. The figure shows that the convergence of P_{ERR} towards very small values (below 0.01) for scenario 1 can last between 300 and 400 application periods depending on the value N_1. However, the convergence of the Q_{ERR} which is defined as more critical is significantly faster in the mechanism. It requires less than 100–200 application periods for Q_{ERR} to reach values below 0.01 and some 200–350 application periods to reach values below 0.001 depending on N_2 (except for the case $N_2 = 4$ which is somewhat slower).

The algorithm uses $K = M - 1$, which in scenario 1 makes the system completely immune to the case with one bad cooperator (i.e. $P_{ERR} = 0$ for $N_1 = 1$). On the other side, in scenario 2, the system is completely immune to having one bad cooperator as well (i.e. $Q_{ERR} = 0$ for $N_2 = 1$) due to having $K = M - 1$ and the fact that the minimum value used for M is 3.

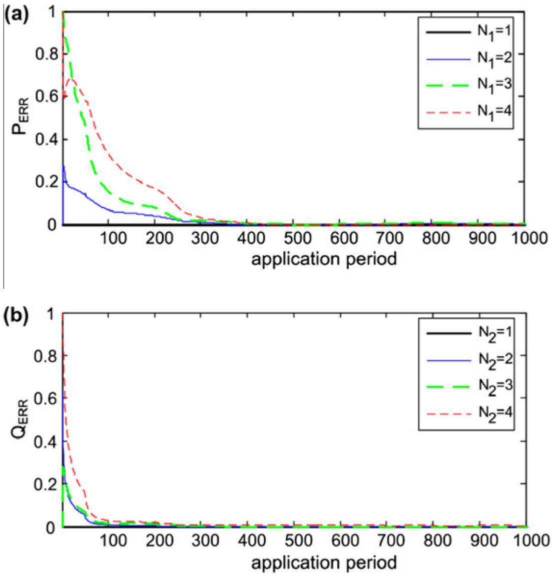

Fig. 6. Evolution of: (a) P_{ERR} in scenario 1 and (b) Q_{ERR} in scenario 2.

Note that scenarios 1 and 2 are the worst case scenarios for P_{ERR} and Q_{ERR}, respectively, in the sense that in each application period whenever the wrong set of cooperators is used the probabilities P_{ERR} and Q_{ERR} for that application period will be 1. Additionally, the CR user has no knowledge of the suitability of any of the N cooperators at the beginning.

5.3. Comparison with “K out of N”

In the following the analysis of the performances is done of the proposed algorithm under a completely random and unknown behavior of the different cooperators. In the second set of results, number of cooperators is $N = 15$, and the aim is to compare the performance of the proposed approach that adaptively varies M contributors with the “K out of N” data fusion dynamic highly hostile conditions.

Spectrum availability is simulated as two states of exponentially distributed length with mean of $1/\lambda = 500$ application periods. The spectrum access is randomly available with probabilities $\Pr\{H_1\} = 0.2$ and $\Pr\{H_1\} = 0.8$ in the two states. The simulation length is 2×10^5 application periods.

All cooperators change their behavior independently for $p_{err}(i)$ and $q_{err}(i)$ after separately generated number of application periods with mean $1/\lambda_p$ for p_{err} and $1/\lambda_q$ for q_{err}, $1/\lambda_p = 1/\lambda_q = 2000$. The new value p_{err} is uniformly selected within a specific interval I_P and value p_{err} is uniformly selected within a specific interval I_Q. In order to simulate different types of cooperators three behavior types are considered, defined by the following intervals: $I_{1P} = I_{1Q} = [0–0.05]$, $I_{2P} = I_{2Q} = [0.05–0.75]$, $I_{3P} = I_{3Q} = [0.75–1.00]$. I_{1P} and I_{1Q} present relatively good behavior, whereas other behavior types are for less reliable or malicious cooperators. Then, when cooperator i wants to change $p_{err}(i)$ or $q_{err}(i)$, it first randomly selects the range interval: I_{1P} (or I_{1Q} for q_{err}) with probability P_1 (or Q_1 for q_{err}), I_{2P} (I_{2Q}) with probability P_2 (Q_2) and I_{3P} (I_{3Q}) with probability P_3 (Q_3). Afterwards, the error rate $p_{err}(i)$ or $q_{err}(i)$ is uniformly selected from the chosen interval.

Probability values for eight case studies (a–h) considered in this section are given in Table 3. The values in the presented case studies are chosen arbitrarily to present the performance of the proposed solution with different grade of malicious behavior. This is because the behavior of the cooperators is assumed to be completely random and unknown in worst case scenarios with malicious users.

The proposed algorithm is compared with fixed “K out of N” solution for every K, $K = 1, \ldots, N$, denoted by $Q_{ERR}(K/N)$ and $P_{ERR}(K/N)$. Results for eight case studies (a–h) from Table 3 are presented in Fig. 7. In each graph, horizontal dashed lines represent the values P_{ERR} obtained with the proposed solution based on RL, denoted by $P_{ERR}(RL)$. In turn, the value $Q_{ERR}(RL)$ is not plotted since it is in all the cases below 0.001.

The case studies (a–h) grade from less hostile to more hostile conditions. In case study (a) there are more values K for which both P_{ERR} and Q_{ERR} can be preserved low with “K put of N”. From cases (b)–(d) the zone with both low P_{ERR} and Q_{ERR} is becoming tighter. Finally, for case studies after (e) the tradeoff between setting low P_{ERR} or Q_{ERR} by choosing K becomes obvious. When smaller P_{ERR} is desired, lower K is preferable; however, when smaller Q_{ERR} is desired, higher K is preferable. It can also be observed how P_{ERR} and Q_{ERR} can take different values depending on the case condition, e.g. Q_{ERR} is low with much smaller K in case (g) than (f), whereas P_{ERR} is low with much higher K in case (f) than (g).

Table 3
Case studies.

Case	Q_1	Q_2	Q_3	P_1	P_2	P_3
(a)	0.9	0.1	0	0.9	0.1	0
(b)	0.6	0.2	0.2	0.8	0.1	0.1
(c)	0.8	0.1	0.1	0.7	0	0.3
(d)	0.6	0.2	0.2	0.6	0.2	0.2
(e)	0.6	0	0.4	0.6	0	0.4
(f)	0.5	0	0.5	0.7	0	0.3
(g)	0.8	0	0.2	0.3	0	0.7
(h)	0.4	0.3	0.3	0.4	0.3	0.3

Fig. 8 compares values P_{ERR} obtained with the proposed solution, for eight case studies, with “K out of N” for the K that has the lowest P_{ERR} while assuring that $Q_{ERR} < D_H^U = 0.001$. This is denoted as Best “K out of N” solution in the figure. Depending on the case conditions best K can be different K, e.g. $K = 8$ for cases (c) and (g), and $K = 13$ for case (h).

The proposed solution expresses higher robustness to malicious behavior and also maintains very low error probability values for P_{ERR} and Q_{ERR} in less hostile conditions. In the case (a), $P_{ERR} = 0.004$ for the proposed algorithm due to the exploration needed for reinforcement learning. For the cases (b) and (c) there still exists K in “K out of N” with slightly lower P_{ERR}. As the conditions become more hostile, the proposed solution outperforms all “K out of N” solutions (d–h).

5.4. Performance in the presence of perfect cooperators

In this study, only two behavior types are considered, defined with following error probabilities $I_{1P} = I_{1Q} = [0–0.05]$ and $I_{2P} = I_{2Q} = [0.95–1.00]$ for $p_{err}(i)$ and $q_{err}(i)$. These behavior types are selected by the users with probabilities $P_1 = Q_1$ and $P_2 = Q_2$ ($P_2 = 1 - P_1 = Q_2 = 1 - Q_1$). As in the previous scenario, from Section 5.3, behavior intervals are changed independently for $p_{err}(i)$ and $q_{err}(i)$. Once a behavior type is assigned, values for $p_{err}(i)$ and $q_{err}(i)$ are uniformly chosen from the corresponding interval. Two cases are compared now:

- *Case* 1*:* All 15 cooperators assign random values to $p_{err}(i)$ and $q_{err}(i)$.
- *Case* 2*:* There are two perfect cooperators, that have constantly $p_{err}(i) = q_{err}(i) = 0$, whereas other 13 cooperators behave as in case 1.

The results for P_{ERR} when $P_2 = Q_2$ is incrementing are presented in Fig. 9. For illustrative purposes best “K out of N” solution, as defined in previous study (lowest P_{ERR} for $Q_{ERR} < 0.001$), is compared with the results obtained with proposed algorithm. The result obtained with the best fix K for the “K out of N” is taken independently for each simulated point. The results demonstrate that best “K out of N” obtains lower P_{ERR} in the presence of two perfect cooperators. However, it does not perform cooperator selection, so the final decision still needs to be made through combining with other 13 imperfect cooperators.

Results show that in the case with two perfect cooperators the proposed solution reduces P_{ERR} to almost 0. As expected from Section 4, the learning mechanism manages to identify the good cooperators and achieves very good performances with very low number of cooperators. Therefore, a reduction of M through cooperation leads to a significant improvement of performances.

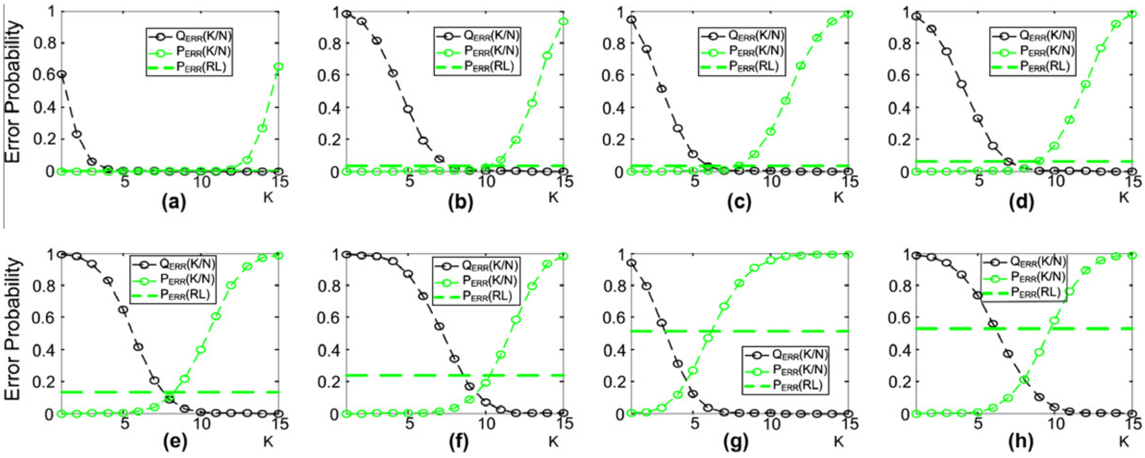

Fig. 7. Error probabilities $P_{ERR}(K/N)$ and $Q_{ERR}(K/N)$ for "K out of N" fusion ($K = 1, \ldots, N$) and $P_{ERR}(RL)$ for proposed algorithm in eight case studies: (a–h).

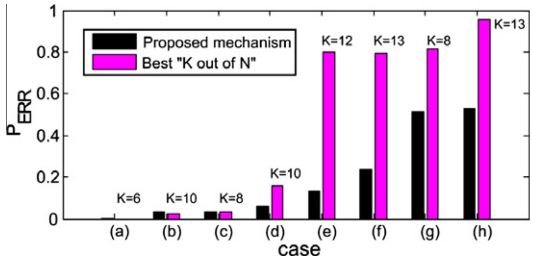

Fig. 8. P_{ERR} in different case studies, the best K values for "K out of N" solutions are indicated next to the corresponding P_{ERR} values.

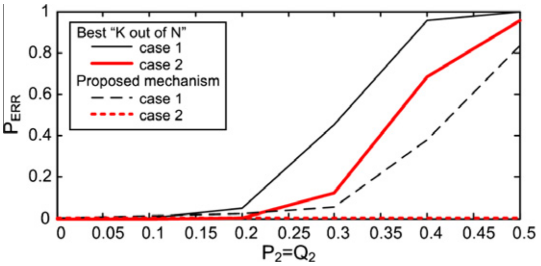

Fig. 9. P_{ERR} as a function of $P_2 = Q_2$.

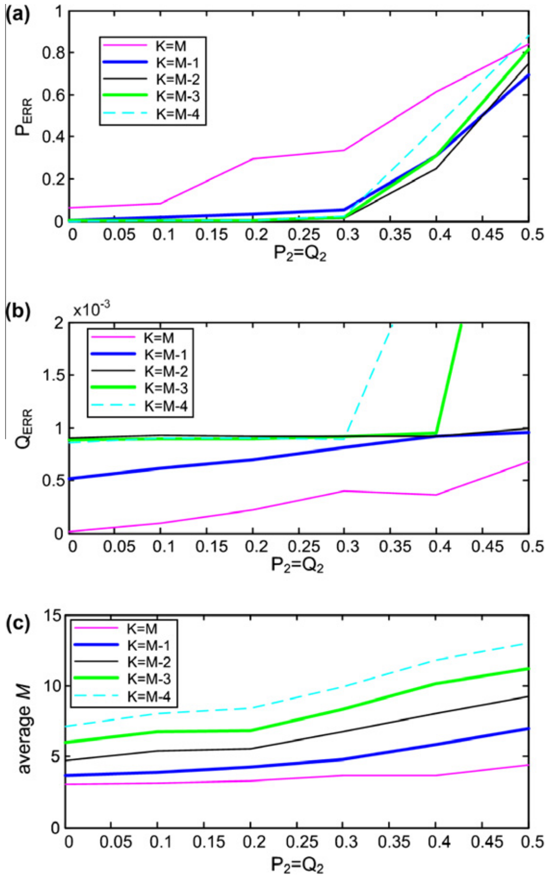

Fig. 10. Results for variable K: (a) P_{ERR}, (b) Q_{ERR} and (c) average M.

5.5. Variation of K

Different values K of the decision rule "K out of M" in the proposed solution are tested in this section. The scenario case 1 from Section 5.4 is considered. Fig. 10a and b compare P_{ERR} and Q_{ERR} for cases where $K = M$ to $K = M - 4$. As the results show, $K = M$ is the case that has the worst performances in terms of P_{ERR}, but, as expected, is the safest choice in terms of Q_{ERR}. For $K = M - 2$, the results are very similar to the case $K = M - 1$. Values K ($K < M - 2$) give similar results in terms of P_{ERR}, but increment Q_{ERR} in more hostile conditions, i.e. when $P_2 = Q_2$ increases.

Finally, Fig. 10c compares the average value M in the considered tests. As it has been seen, $K = M - 2$ achieved the same results as $K = M - 1$ in terms of P_{ERR} and Q_{ERR}. However, $K = M - 1$ achieved this with lower M, which enables reduction of communication load necessary for cooperation.

5.6. Variation of T_A

This section compares performances for the different number of decisions per application period T_A. Values

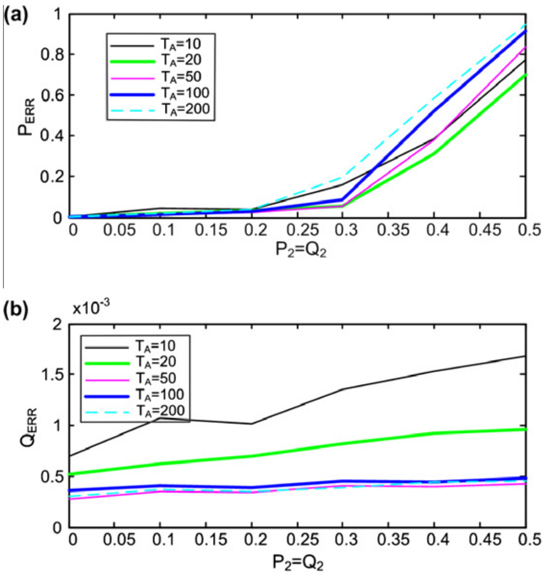

Fig. 11. Results for different number of decisions in one application period T_A: (a) P_{ERR} and (b) Q_{ERR}.

P_{ERR} and Q_{ERR} obtained for the case 1 from Section 5.4 are compared in Fig. 11. In terms of Q_{ERR} most difference is achieved when $T_A = 10$ and $T_A = 20$. This reveals that when interference probability is critical, higher number of decisions per application can be preferred. However, note that there is no need to take $T_A > 50$, as the number of decisions T_A higher than 50 is not decreasing Q_{ERR} (below approximately 0.0005), whereas P_{ERR} is increased.

6. Conclusions

This paper had addressed the reliability issue for cooperative cognitive radio networks, trying to maximize opportunistic radio access, with minimal interference to primary users, when cooperators advising on resource access can be unreliable or malicious. This paper has presented a solution that by means of reinforcement learning maintains a list of cooperators and their suitability, and selects the appropriate ones to in order to maximize correct resource access. The results demonstrate the capability of the proposed solution to successfully learn and act in dynamic hostile environments. The proposed solution offers robustness to highly erroneous cooperation conditions that could be due to either lowered reliability or possible Byzantine attacks. Additionally, the proposed solution adapts the number of used cooperators in accordance with their performances. When there are few highly accurate cooperators, the proposed solution successfully encounters them and bases the spectrum access decision only on their advices to achieve very high performances.

Acknowledgements

This work is supported by the US National Science Foundation under contract ECCS-0900930; by the Spanish Research Council and FEDER funds under ARCO grant (ref. TEC2010-15198); and MEC-FPU Scholarship (AP2005-3739). The authors would like to thank Josep Miquel Jornet for his valuable comments that improved this work.

Appendix A

In this section, some additional expressions that illustrate the dependence of P_{ERR} and Q_{ERR} on M and K are derived. For the general case P_{ERR} (Eq. (4)) and Q_{ERR} (Eq. (5)) can be presented as functions of M and K:

$$P_{ERR}(M,K) = 1 - \sum_{j=K}^{M} W(M,j), \tag{19}$$

$$Q_{ERR}(M,K) = \sum_{j=K}^{M} F(M,j), \tag{20}$$

where

$$W(M,j) = \Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = j|H_1\right\}, \tag{21}$$

$$F(M,j) = \Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = j|H_0\right\}. \tag{22}$$

As the values $W(M, j)$ and $F(M, j)$ are probabilities, it stands: $0 \leqslant W(M, j) \leqslant 1$ and $0 \leqslant F(M, j) \leqslant 1$.

Theorem 1. For a given number of cooperators M, for a hard decision with "K out of M" rule, case using $K'' > K$ will have higher or equal error probability P_{ERR} but will have lower or equal error probability Q_{ERR}, than the case using K.

Proof. From Eqs. (19) and (20) it can be observed that if instead of K, $K'' = K + 1$ cooperators are used in the "K out of M" rule, the following expressions can be obtained:

$$\begin{aligned} P_{ERR}(M,K+1) &= 1 - \sum_{j=K+1}^{M} W(M,j) \\ &= P_{ERR}(M,K) + W(M,K), \end{aligned} \tag{23}$$

$$Q_{ERR}(M,K+1) = \sum_{j=K+1}^{M} F(M,j) = Q_{ERR}(M,K) - F(M,j). \tag{24}$$

It is clear from the previous equations that:

$$P_{ERR}(M,K'') \geqslant P_{ERR}(M,K), \quad K'' > K, \tag{25}$$

$$Q_{ERR}(M,K'') \leqslant Q_{ERR}(M,K), \quad K'' > K. \quad \square \tag{26}$$

Theorem 2. For a number of cooperators M and a fixed difference $\Delta = M - K$ ($K = M - \Delta$) in the "K out of M" rule, the error probabilities are $P_{ERR}(M, K)$ and $Q_{ERR}(M, K)$. Then, if one additional cooperator is added, i.e. M is incremented ($M'' = M + 1$), then, the new error probabilities will be $P^*_{ERR} \geqslant P_{ERR}(M,K)$ and $Q^*_{ERR} \leqslant Q_{ERR}(M,K)$, independently of the erroneous probability advices of the new cooperator p^*_{err} and q^*_{err}.

Proof. If number of cooperators M is incremented, then K is also incremented $M = M + 1 \rightarrow K = K + 1$. So the new error probabilities become:

$$P^*_{ERR} = P_{ERR}(M+1,K+1) = 1 - \sum_{j=K+1}^{M+1} \Pr\left\{\left(\sum_{i=1}^{M+1} x(i)\right) = j|H_1\right\}, \quad (27)$$

$$Q^*_{ERR} = Q_{ERR}(M+1,K+1) = \sum_{j=K+1}^{M+1} \Pr\left\{\left(\sum_{i=1}^{M+1} x(i)\right) = j|H_0\right\}. \quad (28)$$

If the cases when the new cooperator's advice is 1 (with probabilities $1 - p^*_{err}$ and q^*_{err}) and 0 (with probabilities p^*_{err} and $1 - q^*_{err}$) are separated, the previous equations become:

$$P^*_{ERR} = 1 - \left((1 - p^*_{err}) \sum_{j=K+1}^{M+1} \Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = (j-1)|H_1\right\} + p^*_{err} \sum_{j=K+1}^{M+1} \Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = j|H_1\right\}\right), \quad (29)$$

$$Q^*_{ERR} = \left(q^*_{err} \sum_{j=K+1}^{M+1} \Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = (j-1)|H_0\right\} + (1 - q^*_{err}) \sum_{j=K+1}^{M+1} \Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = j|H_0\right\}\right). \quad (30)$$

Taking into account:

$$\Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = M+1|H_1\right\} = 0, \quad i.e. W(M, M+1) = 0, \quad (31)$$

$$\Pr\left\{\left(\sum_{i=1}^{M} x(i)\right) = M+1|H_0\right\} = 0, \quad i.e. F(M, M+1) = 0. \quad (32)$$

Using Expressions (21) and (22), Expressions (29) and (30) become:

$$P^*_{ERR} = 1 - \left[(1 - p^*_{err}) \sum_{j=K}^{M} W(M,j) + p^*_{err} \sum_{j=K+1}^{M} W(M,j)\right], \quad (33)$$

$$Q^*_{ERR} = q^*_{err} \sum_{j=K}^{M} F(M,j) + (1 - q^*_{err}) \sum_{j=K+1}^{M} F(M,j). \quad (34)$$

Finally, from the last expressions and Eqs. (19) and (20) it can be obtained:

$$P^*_{ERR} = P_{ERR}(M,K) + p^*_{err} W(M,K), \quad (35)$$

$$Q^*_{ERR} = Q_{ERR}(M,K) - (1 - q^*_{err}) F(M,K). \quad (36)$$

It is clear from the last equations that:

$$P_{ERR}(M+1,K+1) \geqslant P_{ERR}(M,K), \quad \forall p^*_{err}, \quad (37)$$

$$Q_{ERR}(M+1,K+1) \leqslant Q_{ERR}(M,K), \quad \forall q^*_{err}. \quad \square \quad (38)$$

References

[1] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, S. Mohanty, Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey, Computer Networks 50 (13) (2006).

[2] S.M. Mishra, A. Sahai, R.W. Brodersen, Cooperative sensing among cognitive radios, in: IEEE International Conference on Communications (IEEE ICC), June 2006.

[3] Yuan Zhang, Gaochao Xu, Xiaozhong Geng, Security threats in cognitive radio networks, in: Int. Conference on High Performance Computing and Communications (HPCC), September 2008.

[4] J.L. Burbank, Security in cognitive radio networks: the required evolution in approaches to wireless network security, in: IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications (IEEE CrownCom), May 2008.

[5] T. Clancy, N. Goergen, Security in cognitive radio networks: threats and mitigation, in: IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications (IEEE CrownCom), May 2008.

[6] Chen Ruiliang, Park Jung-Min, Y.T. Hou, J.H. Reed, Toward secure distributed spectrum sensing in cognitive radio networks, IEEE Communications Magazine 46 (4) (2008).

[7] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, A Bradford Book, MIT Press, Cambridge, MA, 1998.

[8] R.W. Thomas, D.H. Friend, L.A. DaSilva, A.B. MacKenzie, Cognitive networks: adaptation and learning to achieve end-to-end performance objectives, IEEE Communication Magazine 44 (12) (2006).

[9] Q. Zhang, P.K. Varshney, R.D. Wesel, Optimal bi-level quantization of i.i.d. sensor observations for binary hypotesis testing, IEEE Transactions on Information Theory 48 (7) (2002).

[10] J.L. Burbank, W.T.M. Kasch, The application of human and social behavioral-inspired security models for self-aware collaborative cognitive radio networks, in: International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), November 2008.

[11] G. Han, D. Choi, W. Lim, A reliable and efficient approach of establishing trust for wireless sensor networks, in: Int. Conf. on Intelligent Computer Communication and Processing, September 2007.

[12] S. Buchegger, J.-Y. Le Boudec, Self-policing mobile ad hoc networks by reputation systems, IEEE Communications Magazine 43 (7) (2005).

[13] N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, M.K. Warmuth, How to use expert advice, Journal of the ACM 44 (3) (1997).

[14] D. Puci de Farias, N. Meggido, Combining expert advice in reactive environments, Journal of the ACM 53 (5) (2006).

[15] S. Mosleh, A.A. Tadaion, M. Derakhtian, Performance analysis of the Neyman–Pearson fusion center for spectrum sensing in cognitive radio networks, in: International Conference on Computer as a Tool (Eurocon), May 2009.

[16] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning 8 (1992).

[17] M.T. Rosenstein, A.G. Barto, in: J. Si, A. Barto, W. Powell, D. Wunsch (Eds.), Learning and Approximate Dynamic Programming: Scaling up to the Real World, John Wiley & Sons, Inc., New York, 2004.

[18] N. Vucevic, I.F. Akyildiz, J. Pérez-Romero, Cooperation reliability based on reinforcement learning for cognitive radio networks, in: IEEE Workshop on Networking Technologies for Software Defined Radio and White Space, IEEE SECON Workshops, June 2010.

Nemanja Vučević received the Telecommunications Engineering degree from the Faculty of Electrical Engineering, Belgrade University, Belgrade, Serbia, in 2003. He is currently working toward the Ph.D. degree with the Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, Barcelona, Spain. From July to December 2009, he was a Visiting Researcher with the Broadband Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta GA. His current research interests include radio resource and QoS management in multi-access wireless and all-IP networks, dynamic spectrum management, and cognitive radio.

Ian F. Akyildiz received the B.S., M.S., and Ph.D. degrees in Computer Engineering from the University of Erlangen-Nurnberg, Germany, in 1978, 1981 and 1984, respectively. Currently, he is the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, the Director of the Broadband Wireless Networking Laboratory and Chair of the Telecommunication Group at Georgia Tech.

He is an Honorary Professor with the School of Electrical Engineering at the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, since June 2008. He is also the Founder and Director of the recently created NaNoNetworking Center in Catalunya (N3Cat) at the UPC. Since March 2009, he is an Honorary Professor with the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria, South Africa. He is also a Visiting Professor at the King Saud University in Riyadh, Saudi Arabia, since January 2010.

He is the Editor-in-Chief of Computer Networks (Elsevier) Journal, and the founding Editor-in-Chief of the Ad Hoc Networks (Elsevier) Journal, the Physical Communication (Elsevier) Journal and the Nano Communication Networks (Elsevier) Journal. He serves on the advisory boards of several research centers, journals, conferences and publication companies. He is an IEEE Fellow (1996) and an ACM Fellow (1997). He has received numerous awards from IEEE and ACM.

He received the "Don Federico Santa Maria Medal" for his services to the Universidad of Federico Santa Maria, in 1986. From 1989 to 1998, he served as a National Lecturer for ACM and received the ACM Outstanding Distinguished Lecturer Award in 1994. He received the 1997 IEEE Leonard G. Abraham Prize Award (IEEE Communications Society) for his paper entitled "Multimedia Group Synchronization Protocols for Integrated Services Architectures" published in the IEEE Journal of Selected Areas in Communications (JSAC) in January 1996. He received the 2002 IEEE Harry M. Goode Memorial Award (IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking". He received the 2003 IEEE Best Tutorial Award (IEEE Communication Society) for his paper entitled "A Survey on Sensor Networks", published in IEEE Communications Magazine, in August 2002. He also received the 2003 ACM SigMobile Outstanding Contribution Award with the citation "for pioneering contributions in the area of mobility and resource management for wireless communication networks".

He received the 2004 Georgia Tech Faculty Research Author Award for his "outstanding record of publications of papers between 1999 and 2003". He also received the 2005 Distinguished Faculty Achievement Award from School of ECE, Georgia Tech. In 2009, he received the Georgia Tech Outstanding Doctoral Thesis Advisor Award for his 20+ years service and dedication to Georgia Tech and producing outstanding PhD students. He also received the 2009 ECE Distinguished Mentor Award from the School of Electrical and Computer Engineering Faculty Honors Committee, Georgia Tech.

His current research interests are in Nanonetworks, Cognitive Radio Networks and Wireless Sensor Networks.

Jordi Pérez-Romero (M'98) received the Telecommunications Engineering and Ph.D. degrees from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1997 and 2001, respectively. He is currently an Associate Professor at the Department of Signal Theory and Communications, UPC. He has been involved in different European Projects as well as in projects for private companies. He is the author or coauthor of various papers published in various international journals and conference proceedings. He is also a coauthor of one book on mobile communications. His research interests include mobile communication systems, especially packet radio techniques, radio resource and QoS management, heterogeneous wireless networks, and cognitive networks.