Implementation of Dynamic Spectrum Allocation for Cognitive Radio Networks based on Iterative Water Filling in OMNeT++/MiXiM

Ir. D'HONDT Sébastien
Royal Military Academy
Brussels, Belgium
sebastien.dhondt@mil.be

Dr. ir. SCHEERS Bart
Royal Military Academy
Brussels, Belgium
bart.scheers@rma.ac.be

Dr. ir. LE NIR Vincent *
Royal Military Academy
Brussels, Belgium
vincent.lenir@rma.ac.be

Abstract

The growth in wireless technology and the increasing demand for wireless multimedia services creates a lack of spectrum. A potential solution to this issue is to allocate the spectrum dynamically by means of cognitive radio. Iterative Water Filling (IWF) can offer a practical solution to this dynamic spectrum allocation. Straightforward implementations of IWF in C/C++ or Matlab already exist. To our knowledge, this algorithm has not been studied yet in an event-driven simulator such as OMNeT++/MiXiM. For the implementation of the IWF, it is required to have multiple sub-channels with an adjustable power for each of them. In this work, several existing MiXiM modules are extended in order to permit the use of multiple sub-channels for the communication between nodes. Moreover, new classes, maps and events are created for the implementation of the IWF algorithm. The implementation is validated through simulations of a scenario where two tactical radio networks coexist in the same area.

Categories and Subject Descriptors

C.2.1 [**Computer-Communications Networks**]: Network Architecture and Design—*Wireless communications*; I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Discrete event*

1 Introduction

Nowadays, there are a lot of wireless applications sharing the same medium. This overload leads to a lack of spectrum in given frequency bands. At this time, the allocation of the spectrum is mostly static and based on licensing, as in the case of the FM-band. The purpose of this work is to present another approach of sharing the same medium by means of a dynamic allocation of the spectrum instead of the static allocation. The cognitive radio (CR) provides a solution to this dynamic spectrum access (DSA). This concept is an extension of the software defined radio (SDR).

Three approaches of DSA can be distinguished: the dynamic exclusive use, the open sharing and the hierarchical access model [3]. Those models vary according to the existence of priority to access the medium of the different nodes. In this work, all the nodes are considered equal in a open sharing model (based on unlicensed bands). This model can be applied in the case of a military international deployment with little preparation time in which dynamic spectrum allocation permits the coexistence of tactical radio networks in the same area. In such a situation, it is not convenient to consider a static allocation of frequencies between the participating states.

A practical implementation of the spectrum management can be achieved thanks to game theory [1]. Game theory is a mathematical modeling of strategic situations (= game), in which an individual's choice depends on the choices of the others. The choice is evaluated by an objective function. The result of this function can be optimal if the players cooperate with each other. Because of the absence of cooperation between networks of cognitive radios (= players), it is not a cooperative game anymore, but a competitive one. In this case a Nash equilibrium (NE) can be reached. In a NE, no player can increase the outcome of its objective function by unilaterally changing its strategy.

In this work, the iterative water filling (IWF) algorithm is used for solving the non-cooperative game for DSA [11]. This competitive algorithm gives a sub-optimal allocation of the power by distributing the total power available at the transmit nodes on different sub-channels. The total power used by the transmitters is also reduced if this power is higher than necessary for reaching the target data rate. This is really beneficial for the economy of power, and for reducing the interference with other networks.

The behavior of IWF can be simulated using sequential programming languages such as C/C++ and MatLab. These implementations give a global idea of the convergence of the algorithm, but do not give insights about the behavior in a more realistic environment. There is nowadays no implementation of the algorithm in a real event-driven simulator. The purpose of this work is to concretize this implementation in OMNeT++/MiXiM [10, 5]. A first approach would be to use the signal class of MiXiM for defining multiple sub-channels [4]. In this work, we propose a second approach in which we actually extend existing modules in the connection manager to control multiple sub-channels independently.

The IWF is presented in details in section 2. The algorithm is first described for a single network. Then, the iterative aspect is presented for networks coexisting in the same area. Finally, the IWF algorithm with the power control aspect is described. In section 3, the IWF is implemented in OMNeT++/MiXiM. Therefore, several existing MiXiM modules are extended in order to permit the

*S. D'hondt, B. Scheers and V. Le Nir are with the Royal Military Academy, Dept. Communication, Information Systems & Sensors (CISS), 30, Avenue de la Renaissance B-1000 Brussels BELGIUM. This research work was carried out in the frame of the Belgian Defense Scientific Research & Technology Study C4/19 funded by the Ministry of Defense (MoD). The scientific responsibility is assumed by its authors.

use of multiple sub-channels for the communication between nodes. Moreover, new classes, maps and events are created for the implementation of the IWF algorithm. The implementation is validated in section 4 through simulations of a scenario where two tactical radio networks coexist in the same area.

2 Presentation of the algorithm

A cognitive radio is equipped with a cognitive manager responsible for the control of the total power and the power allocated to each individual sub-channel. In this work, IWF is used for the practical realization of the cognitive manager. The IWF approach is purely competitive and leads to the determination of a sub-optimum. To better understand the implementation of IWF, we will introduce in this section the IWF in three steps. The water filling algorithm is first introduced for a single network and then, it is generalized for several networks. Finally the power control is introduced.

2.1 Water Filling

The water filling (WF) algorithm is considered for one single couple of users composed of a transmitter and a receiver. The two nodes are in an environment with noise and they dispose of multiple sub-channels which they can use in parallel. The problem can be considered as an optimization problem. The optimization has to be performed on the transmitted power. For the considered network, a given data rate has to be achieved with the lowest possible power. In order to solve this, the Lagrange multipliers are used (for a theoretical overview, see [2]). The problem is described by the following problem

$$\begin{array}{l}\text{Minimize} \sum_{i=1}^{N_c} P_i \\ \text{subject to } \; T \leq R\end{array} \tag{1}$$

with $R = \Delta f \sum_{i=1}^{N_c} \log_2\left(1 + \frac{P_i.|H_i|^2}{\Gamma\sigma_i^2}\right)$. In this equation, the different variables are given in Table 1.

N_c	amount of sub-channels
P_i	power delivered to channel ”i” by the transmitter
T	target data rate for the network
R	Effective data rate for the network
H_i	channel property for channel ”i”
σ_i	Standard deviation of the noise for channel ”i”
Γ	The signal to noise ratio (SNR) gap
Δf	The sub-channel bandwidth

Table 1. Variables used in the optimization problem

The problem can also be expressed by its dual form

$$L(\lambda, T) = P_i - \lambda\left[\Delta f \sum_{i=1}^{N_c} \log_2\left(1 + \frac{P_i.|H_i|^2}{\Gamma\sigma_i^2}\right) - T\right] \tag{2}$$

with λ the Lagrange multiplier for the target rate constraint. The derivative is given by

$$\frac{\partial L(\lambda, T)}{\partial P_i} = 1 - \lambda\left[\Delta f \frac{|H_i|^2}{\Gamma\sigma_i^2.(1 + \frac{P_i.|H_i|^2}{\Gamma\sigma_i^2})}\right] \tag{3}$$

The solution of this problem is determined by the following equation

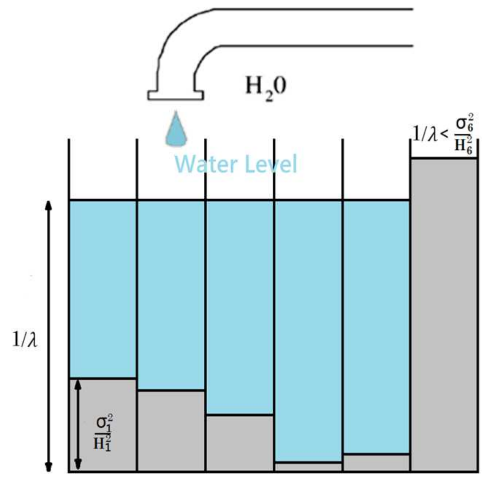

Figure 1. Water Filling - Analogy with water

$$P_i = \left[\frac{1}{\lambda\Delta f} - \frac{\Gamma\sigma_i^2}{|H_i|^2}\right]^+ \tag{4}$$

The solution gives the optimum for the dual form. By consequence, this solution is also an optimum for the initial function [2]. At this step, it is possible to define what is called the water level which is equal to $\frac{1}{\lambda\Delta f}$. The water level is compared to the value of $\frac{\Gamma\sigma_i^2}{|H_i|^2}$ called the normalized noise.

$$\begin{array}{llll} \text{if} & \frac{\Gamma\sigma_i^2}{|H_i|^2} \geq \frac{1}{\lambda\Delta f} & \text{then} & P_i = 0 \\ \text{if} & \frac{\Gamma\sigma_i^2}{|H_i|^2} < \frac{1}{\lambda\Delta f} & \text{then} & P_i = \frac{1}{\lambda\Delta f} - \frac{\Gamma\sigma_i^2}{|H_i|^2} \end{array} \tag{5}$$

From equation (5), an allocation of power is determined. This allocation provides the optimal power configuration for optimizing the data rate between a couple of transmitter/receiver, regarding the noise in the surrounding. An illustration of this algorithm is presented in Figure 1. The water (= the power) has to be distributed in the reservoir (= the normalized noise in the sub-channels). Therefore, the water is poured into the reservoir, and the liquid takes the form of the recipient. If the depth available at a given place is consequent, the amount of water at this place is important. It is the same for the power: a limited depth means a lot of noise being present in this band, and the power allocated to this band will be reduced. Finally, the sum of the allocated power in a sub-channel and the ambient noise is fixed to a constant value, the water level. This water level is the same for each sub-channel.

2.2 Iterative Water Filling

The Water Filling principle has been introduced in the previous section and an illustration has been given for a single network. For the implementation of this algorithm in case of multiple networks, it is necessary to introduce the iterative aspect [11, 8, 6, 7]. The principle is that each network in turn will execute the IWF (see previous section) considering the interference of all other networks as noise.

Figure 2 shows two networks interfering with each other. Every network is composed of one receiver and one transmitter. They communicate using the same sub-channels, which means that inter-

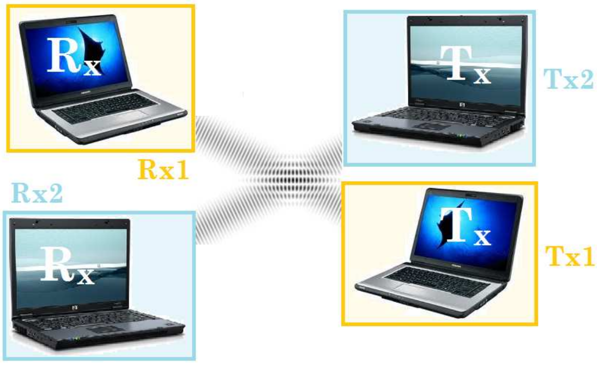

Figure 2. Two networks close to each other

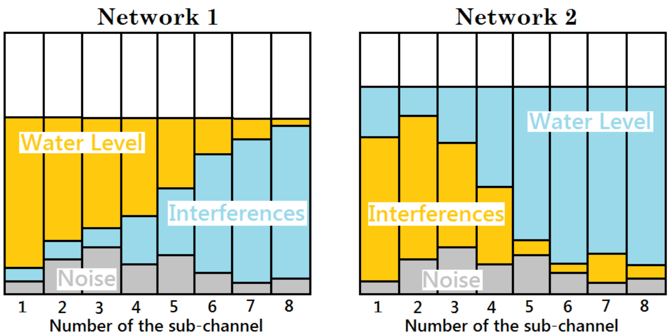

Figure 3. Distribution of the power on the sub-channels of the two networks

ference exists. In the case of this example, eight sub-channels are considered (numbered from 1 to 8).

The four nodes are in a noisy environment, but this noise is not necessary the same on all the sub-channels. The noise is presented in Figure 3. Both couples of transmitter/receiver want to ensure an optimal communication. The first step consists in sensing the medium and in detecting the noise present in each sub-channel. In practice, the node is not able to make a distinction between noise inherent to the surrounding and the interference due to other transmitters. The interference is also represented in Figure 3. At this time, the power allocation of the first network (left) takes place mainly on the four lowest sub-channels. The second network (right) uses mainly the four highest sub-channels for its communication. This illustration shows intuitively the existence of an equilibrium between those two networks. Because the first network sends a lot of power at the lowest frequencies, it causes a lot of interference for the lowest sub-channels of the second network and vice versa. If the networks allocate the power iteratively, it will lead to a NE.

Note that in the two figures, there is a difference in the water level of the two networks. This difference exists because the total power delivered by the two transmitters is not necessarily the same. At this time, the water level of the second network is higher, it means that the transmitter of the second network emits with more power.

2.3 IWF with distributed power control

In this last step, we will describe the distributed power control mechanism of the algorithm (see Figure 4). For this, it is necessary to make a distinction between the inner and the outer loop.

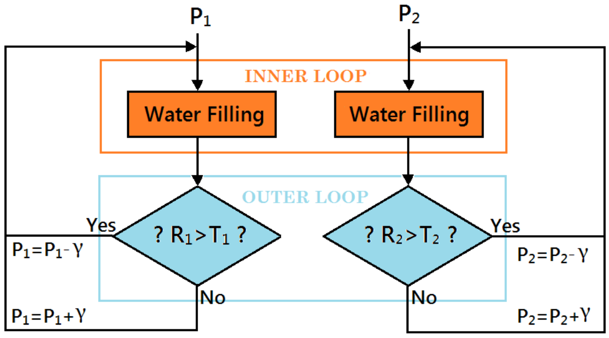

Figure 4. Schematic representation of the Iterative Water Filling Algorithm

The first executed loop is the inner loop. During its execution, each node is looking for the optimal allocation of power between the sub-channels to maximize its data rate subject to a total power constraint. The determination of the best allocation is calculated thanks to the formula (5). This is done iteratively by all the networks until convergence is reached (mostly after five or six iterations).

When convergence is achieved, the outer loop is started. At this level, the effective data rate (R_j) is compared to the target data rate (T_j). The difference between those two values determines the decision to take. If the target rate is smaller than the effective rate, it means that the transmitter delivers too much power and less power would be enough for ensuring the desired data rate. Therefore, the maximum delivered power of the considered transmitter will be decreased by a given quantity γ (dB).
If the target rate is not reached, it means that the power delivered by the transmitter is not enough to ensure the desired data rate. In this case, the maximum delivered power of the considered transmitter will be increased. The outer loop is continuously executed, until every transmitter has reduced its power to the minimum.

3 Multichannel model in MiXiM

MiXiM is an package that adds the mobility dimension to static networks. In OMNeT++, nodes are composed of different layers. Messages can be exchanged between the nodes by using the input and output points (cGate). A message is characterized by two main parameters: its transmitter and its receiver. Using the MiXiM framework, messages can be transmitted by means of a wireless channel and will only be processed by the destination receiver if a connection exists between the two parties. The existence of this connection is determined by the connection manager.

In the original simulation, the connection manager decides if a connection is possible between two nodes based on the signal to noise ratio (SNR) at the receiver side. This leads to the calculation of a maximum interference distance, which is compared to the real distance between both nodes (see Figure 5).

The purpose of this work is to concretize the implementation of the IWF algorithm in OMNeT++/MiXiM [10, 5]. A first approach would be to use the signal class of MiXiM for defining multiple sub-channels [4]. In this work, we propose a second approach in which we actually extend existing modules in the connection manager to control multiple sub-channels independently. To do so, it is necessary to add the sub-channel dimension to several functions and to specify the sub-channel through which a message has to be sent from the transmitter to the receiver. The extension of cGate

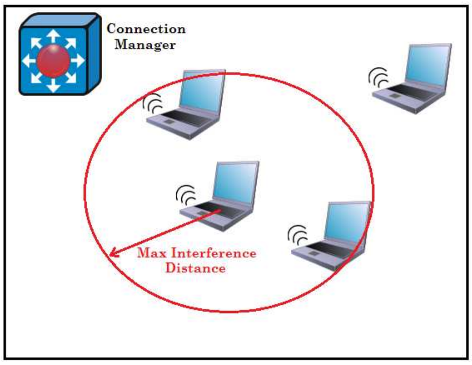

Figure 5. Overview of the initial simulation

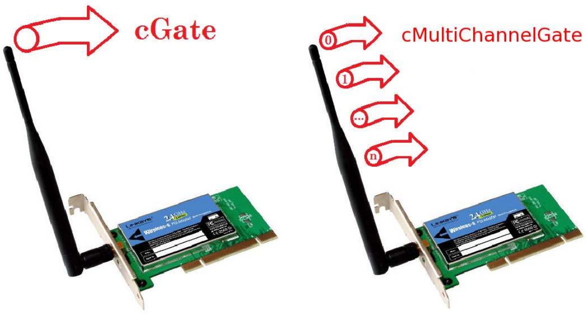

Figure 6. Extension of a cGate to a cMultiChannelGate

class is achieved by the creation of cMultiChannelGate (see Figure 6) which allows to specify the number of the sub-channel used for the transmission. Other functions need also to be customized as *"SendToChannel()"*, *"ConnectTo()"* and *"DisconnectFrom()"* to take into account the sub-channels created by the cMultiChannelGate extension. The consequence of this customization is that the connection manager doesn't only manage the connection between nodes, but also the sub-channels that can be used for that purpose. The class AirFrame has also been extended to the wAirframe by adding the sub-channel dimension. The connection manager is now able to determine if two nodes are connected through given sub-channels. This is achieved thanks to the *"UpdateNicConnections()"*.

At this step, the model is composed of multiple sub-channels instead of a single channel. It is still necessary to add some elements in order to stock data related to the execution of the algorithm. For the implementation of IWF, three new classes have been created. The variables and the classes created for the implementation in OMNeT++/MiXiM are described in Table 2. Once those classes have been defined, it is possible to refer to them in a map. A map is a kind of table with data, but characterized by two parameters. The first parameter is called the "key" and is different for every element of the map. The second parameter is the "value". A key is linked to a value, but a value can be composed of a collection of data. The approach chosen in this work is to put a pointer to the MultiChannelNicEntry as key value. This key is related to a value, which is a list of object belonging to one of the classes presented above. As example, the map related to the receiver class is presented in Figure 7. On this figure, it is clear that a key value is linked on a collection of data for each node (=key). In other words, the map has the role of a local list of data available for each node.

Variable	Description	Class
node	refers to the memory address of a given node of the network	node
indexchannel	describes the sub-channel on which the transmission is established	node
pathloss	characterizes the losses in the sub-channel connected to this node	node
power	gives the power delivered by this node on this sub-channel	node
network	gives the id of the network to which the node belongs	node
status	describes the status of the node: "0"=Rx and "1"=Tx	node
staticrandom	gives a random character to the sub-channel	node
indexchannel	describes the sub-channel on which the transmission is established	receiver
noiseandinterference	includes noise and interference due to transmitter of other networks	receiver
powerowntx	gives the power delivered by the own transmitter	receiver
pathlossowntx	gives the pathloss caused during the transmission from the own transmitter	receiver
maxpower	limits the absolute total power delivered by the transmitter	waterfilling
constraintpower	sums all the powers delivered by the transmitter on all the different sub-channels	waterfilling
iteration	gives the number of inner loop executed in this outer loop	waterfilling
targetrate	fixes the target rate for this network	waterfilling

Table 2. Description of the variables created in OMNeT++/MiXiM

Key	Value			
MultiChannelNicEntry*	list<receiver>			
	int	**double**	**double**	**double**
	indexchannel	noiseandinterference	powerowntx	pathlossowntx
Node 1	0	$3,46 \times 10^{-11}$	0,0218	$2,23 \times 10^{-11}$
	...	...	...	...
	4	$3,22 \times 10^{-11}$	0,0195	$2,12 \times 10^{-11}$
Node 2	0	$8,72 \times 10^{-10}$	0	$4,28 \times 10^{-11}$
	...	...	...	...
	4	$1,18 \times 10^{-11}$	0,121	$4,43 \times 10^{-11}$
...	...	...	...	...

Figure 7. Illustration of the map related to the receiver class

After the creation of classes and maps related to each of them, it is now possible to start the implementation of IWF with distributed power control. Therefore, three functions independent from each other are created.

The first one is the function *"calcpower()"*. Its aim is to fill in the map linked to the the node class. This function is called together with the *"UpdateNicConnections()"*. Another important role of this function is to keep updating the pathloss value. Because the considered nodes are mobile, the pathloss varies continuously.
The second function is a function that executes the inner loop. The goal of this *"IterationInnerLoop()"* function is to determine which allocation of power of the respective transmitter matches with the current configuration of the network. In other words, this function distributes the available power on the different sub-channels. For that purpose, it is necessary to calculate the water level corresponding to a given transmitter.
The third function implemented for the IWF is related to the execution of the outer loop. The *"IterationOuterLoop()"* is based on a rudimentary approach. The decision of increasing or decreasing the total power of a transmitter is related to the difference between

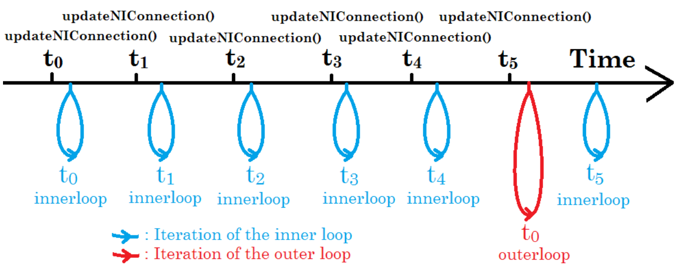

Figure 8. Planning of the events related to IWF

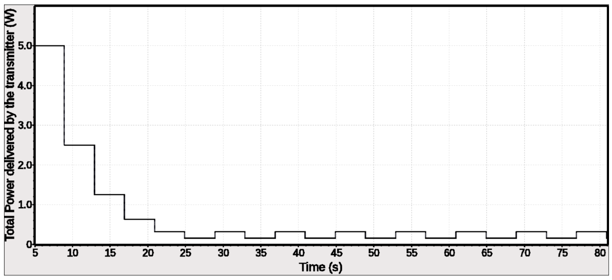

Figure 9. Convergence of the outer loop - overview of the total power

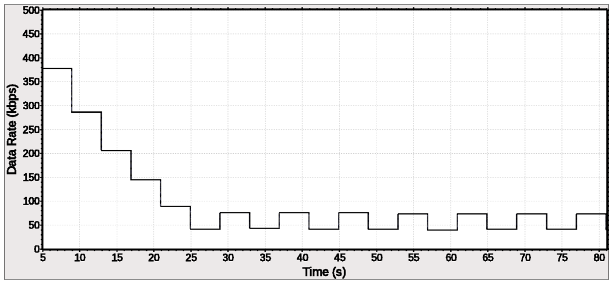

Figure 10. Convergence of the outer loop - overview of the data rate

the target and the real rate. In this work, it is assumed that a given data rate has to be achieved in each network. At the beginning of the simulation, this value is initialized at the maximum power that each transmitter is able to deliver in order to initiate a handshake between a transmitter and its receiver.. If the data rate in a network is higher than the target rate, it means that the transmitter of this network delivers too much power. The constraintpower of this network is set to (0.5·constraintpower). On the other side, if the real data rate is smaller than the target one, the transmitter has to increase its delivered power for reaching the desired target rate. Therefore, its new constraintpower is equal to (2·constraintpower). A limitation to this increase of power exists. The algorithm checks if this constraintpower is smaller than the maxpower of this transmitter.

The scheduling of those functions is also an important issue in the implementation of the IWF. For the periodic scheduling of the inner and outer loop, some functions of the connection manager have been customized. Two events have been added to the existing ones: EXECUTE-IL and EXECUTE-OL. Those events are declared in the *"initialize()"* function [9], and defined in the *"handleMessage()"*. Their execution is planned thanks to the *"scheduleAt()"* function. An overview of the execution of those events is presented in Figure 8.

For the execution of the simulation, a period of 0.5 second is chosen for the inner loop, and a period of 4 seconds for the outer loop. The convergence of the inner loop is mostly reached after five iterations.

4 Simulation results

The algorithm has been fully implemented and this section is dedicated to the results of the simulation. For that purpose, two networks of two nodes are considered. One of those networks has a static behavior, and the other is composed of mobile nodes. Every network is composed of a receiver and a transmitter and this configuration is kept during the entire simulation. The communication in each network is realized by means of four sub-channels. The attenuation between a transmitter and its receiver is driven by the path loss and independent complex Gaussian variables for the sub-channels. The simulation set-up is quite simple and reflects a realistic scenario. There are two vehicles at the North-East side of the considered area. The two trucks follow a pre-defined trajectory with a constant velocity (about 90 km/h). The itinerary followed by the vehicles is described in the "trajectories.txt" file in the simulator and the mobility model is the "BonnMotionMobility" (see MIXIM [5]). The vehicles drive through the city and go to the west. The most critical configuration is obviously reached when the two vehicles are in the city, between the two buildings. In this situation, the two networks are close to each other, and the interference is maximal. In order to ensure a communication in the two networks, a target data rate of 64 kbps is set.

A first step in the evaluation of the IWF implementation consists in the verification of the convergence of the outer loop. Therefore, one considers a single network composed of a couple transmitter/receiver in a noisy environment without other networks in its surrounding. The initial power delivered by the transmitter is equal to the maximum power it can deliver (in this case: 10W). The target data rate between the two nodes is 64 kbps. This power is obviously too large for delivering this data rate.

In Figure 9, the different iterations of the outer loop are presented as a function of time. It is clear that the power decreases in a first time until a real data rate of about 64 kbps is reached. At this moment, the total power has indeed converged. Its value doesn't vary significantly. The evolution of the data rate for the same simulation is showed in Figure 10. A second significant step is to confirm the convergence of the entire algorithm. The simulation can be divided in three key-moments. Those moments are linked to the position of

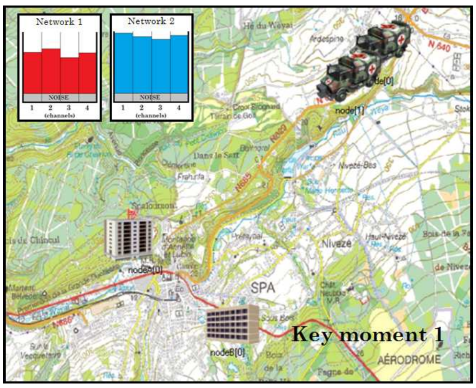

Figure 11. Networks far from each other

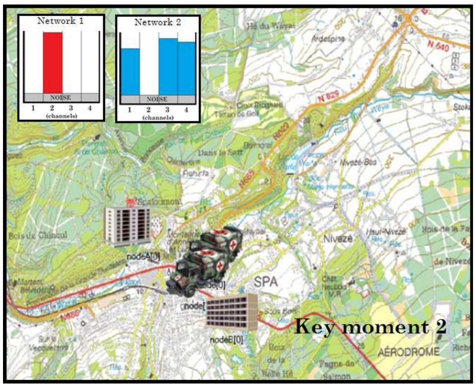

Figure 12. Networks in crossed configuration

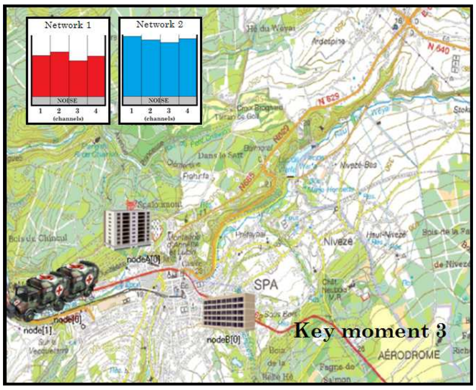

Figure 13. Networks in final configuration

one network to another.

At the beginning of the simulation, the two networks are far enough from each other so that they do not interfer. In this configuration, both transmitters can use the four sub-channels for the communication with their respective receiver. In Figure 11, the distribution of power is presented in the up-left corner. This allocation of power is not uniform, because the sub-channel gain is different for every sub-channel.

In the second key-moment, the networks are close to each other and the interference is maximal. The communication is maintained in this configuration thanks to the IWF (see Figure 12). The two networks use indeed different sub-channels, as it is shown in Figure 12. The first network uses sub-channel 2 and the second network uses sub-channels 1, 3 and 4.

In the last key-moment, the networks are far enough from each other, and the interference is therefore small. The communication is again possible on the four sub-channels (see Figure 13). The allocation of power in this configuration is similar to the initial allocation of power. Note that we have stressed this software by simulating more than two networks (up to ten networks) and it has always shown a very good convergence behavior.

An overview of the power allocation as a function of the time is presented in Figure 14 for the first network. This time evolution is also given for the second network in Figure 15. At a simulation time of about 180 seconds, the networks are close to each other, and the distribution of the sub-channels is maintained until the last

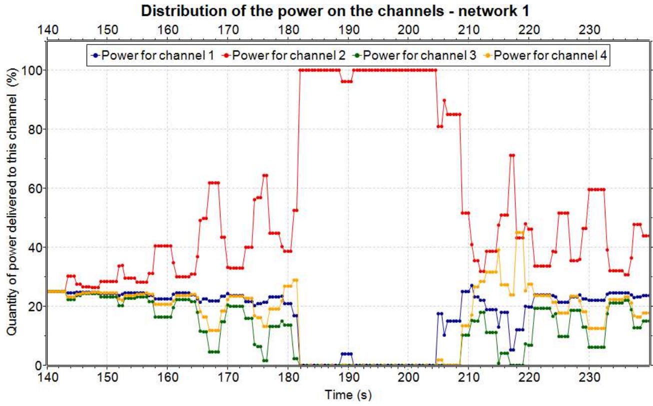

Figure 14. Distribution of power on the different sub-channels in network 1

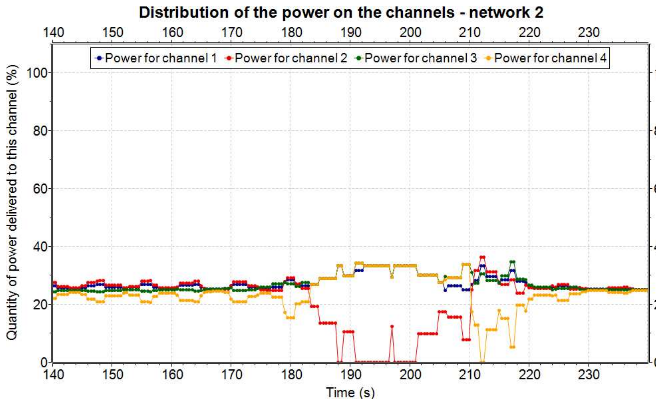

Figure 15. Distribution of power on the different sub-channels in network 2

key-moment (until about 210 seconds).

5 Conclusions

The purpose of this work was to simulate an implementation of the IWF algorithm with distributed power control for cognitive radio networks. This has been done for the first time in an event-driven simulator. OMNeT++/MiXiM has been used for that purpose. A good knowledge of game theory and of the Nash Equilibrium was obviously necessary for the development of the simulation. In this paper, we explained how existing MiXiM modules have been extended in order to permit the use of multiple sub-channels for the communication between nodes. Moreover, we detailed the new classes and maps that have been created for the implementation of IWF. We also detailed the creation of two distinct events: one for the execution of the inner loop and another for the execution of the outer loop. The algorithm has been tested in the case of a simulation with realistic parameters. For that purpose, a scenario has been presented with a convoy of two trucks. Those two vehicles belong to the first network which is mobile. When they come in the direct environment of a second network transmitting on the same sub-channels, they automatically tend to take different sub-channels. The convergence of the inner and outer loop has been confirmed. The IWF is a potential solution for the practical implementation of dynamic spectrum allocation in the case of cognitive radio networks.

6 References

[1] A. Anandkumar, S. Lambotharan, and J. Chambers. *A Game-theoretic approach to Transmitter Covariance Matrix Design for Broadband MIMO Gaussian Interference Channels*. Dept. of Electronic and Electrical Engineering, Loughborough, 2009.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University, Mar. 2004.

[3] S. Couturier and B. Scheers. The state of the art of dynamic spectrum access. In *Military Communication and Information Systems Conference (MCC'2009)*. Prague, Czech Republic, September 29-30 2009.

[4] A. Kopke, M. Swigulski, K. Wessel, D. Willkomm, P. Haneveld, T. Parker, O. Visser, H. Lichte, and S. Valentin. Simulating wireless and mobile networks in omnet++ the mixim vision. In *1st International Workshop on OMNet++*. Marseille, France, March 2008.

[5] MiXiM. http://sourceforge.net/apps/trac/mixim/wiki/Tutorials.

[6] V. Le Nir and B. Scheers. Autonomous dynamic spectrum management for coexistence of multiple cognitive tactical radio networks. In *5th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM'2010)*. Cannes, France, June 2010.

[7] V. Le Nir and B. Scheers. Coexistence of tactical cognitive radio networks. In *SDR Forum European Reconfigurable Radio Technologies Workshop (ERRT'2010)*. Mainz, Germany, June 2010.

[8] V. Le Nir and B. Scheers. Distributed power allocation for parallel broadcast channels with only common information in cognitive tactical radio networks. *EURASIP Journal on Wireless Communications and Networking Volume 2010, Article ID 172013, 10 pages doi:10.1155/2010/172013*, 2010.

[9] A. Varga. *OMNeT++ - Discrete Event Simulation System, Version 3.2*. User Manual, 29 Mar 2005.

[10] K. Wessel, M. Swigulski, A. Kopke, and D. Willkomm. Mixim - the physical layer an architecture overview. In *2nd International Workshop on OMNet++*. Rome, Italy, March 2009.

[11] W. Yu. *Competition and Cooperation in Multi-User Communication Environments*. Stanford University, Stanford, Jun 2002.