

Table des matières

Table des matières	5
Liste des figures.....	9
Liste des tableaux.....	10
Liste des abréviations.....	11
Introduction générale	12
Chapitre I : Radio Cognitive	14
I. Introduction	15
II. Définition.....	15
III. Historique.....	16
IV. Architecture de la RC.....	17
V. Principe.....	19
VI. Fonctions	20
VI.1 Détection du spectre (Spectrum sensing).....	20
VI.2 Gestion du spectre (Spectrum management).....	20
VI.2.1 Analyse du spectre	20
VI.2.2 Décision sur le spectre	20
VI.3 Mobilité du spectre (Spectrum mobility).....	21
VII. Cycle cognitif	21
VII.1 Phase d'observation (détecter et percevoir)	22
VII.2 Phase d'orientation.....	22
VII.3 Phase de planification	22
VII.4 Phase de décision.....	23
VII.5 Phase d'action	23
VII.6 Phase d'apprentissage	23
VIII. Relation entre la RC et la radio logicielle restreinte(SDR).....	23
IX. Composantes de la RC.....	24
IX.1 Emetteur / Récepteur	25

IX.2	Analyseur de spectre (Spectrum analyser)	25
IX.3	Extraction de connaissances et apprentissage (Knowledge extraction/learning) ..	25
IX.4	Prise de décision (Decision making)	25
X.	Langage pour la radio cognitive	25
XI.	Domaines d'application.....	26
XI.1	Les réseaux sans fil de prochaine génération	26
XI.2	Coexistence de différentes technologies sans fil.....	26
XI.3	Services de cyber santé (eHealth services)	26
XI.3.1	Réseaux d'urgence	27
XI.3.2	Réseaux militaires	27
XII.	Conclusion	27
Chapitre II : Exclusion Mutuelle Distribuée		28
I.	Introduction	29
II.	Différence entre un système distribué et un système centralisé	29
III.	Définition de l'exclusion mutuelle	30
III.1	États des processus	30
III.2	Propriétés.....	31
III.3	Ordonnancement.....	32
III.4	Extensions de l'exclusion mutuelle	32
III.4.1	Exclusion mutuelle à priorité	32
III.4.2	Exclusion mutuelle à contraintes de temps	33
III.4.3	Exclusion mutuelle généralisée.....	33
III.4.4	Exclusion mutuelle de groupe.....	33
III.5	Métriques d'évaluation de performances	33
III.6	Mécanismes d'exclusion mutuelle	34
IV.	Taxonomie des algorithmes distribués	35
IV.1	Algorithmes à permissions.....	35
IV.1.1	Permissions individuelles	35

IV.1.2	Permissions d'arbitre	35
IV.1.3	Permissions généralisées	36
IV.2	Algorithmes à jeton	36
IV.2.1	Algorithmes à jeton structurés.....	36
IV.2.1.1	Topologie statique	37
IV.2.1.2	Topologie dynamique	37
IV.2.2	Algorithmes à jeton non-structurés.....	37
IV.2.2.1	Algorithmes statiques.....	37
IV.2.2.2	Algorithmes dynamiques	38
V.	Etat de l'art.....	38
VI.	Conclusion	40
Chapitre III : Contribution et résultats		41
I.	Introduction	42
II.	Suppositions	43
III.	Système Multi-Agents (SMA)	44
IV.	Outils de développement.....	44
IV.1	Plateforme JADE.....	44
IV.2	JFreeChart	45
V.	Topologie du réseau utilisé	45
VI.	Scénario.....	46
VII.	Organigramme	47
VIII.	Algorithme proposé	49
VIII.1	Pour la gestion de spectre.....	49
VIII.2	Pour la tolérance aux pannes.....	50
IX.	Présentation du fonctionnement.....	51
X.	Présentation de l'application	52
XI.	Simulation	54
XII.	Résultats obtenus.....	56

XIII. Évaluation des performances.....	61
XIV. Conclusion.....	62
Conclusion générale.....	63
BIBLIOGRAPHIE.....	64
Bibliographie.....	65
Annexe A.....	68

Liste des figures

Figure I. 1 : Architecture de la radio cognitive	17
Figure I. 2 : Protocoles utilisés par la radio cognitive	19
Figure I. 3 : Principe de gestion du spectre	19
Figure I. 4 : Fonctions de gestion du spectre	21
Figure I. 5 : Cycle de cognition de Mitola	22
Figure I. 6 : Relation entre la RC et la radio logicielle restreinte(SDR).....	24
Figure I. 7 : Composantes de la radio cognitive	24
Figure II. 1 : Automate de l'exclusion mutuelle	31
Figure II. 2 : Taxonomie des algorithmes d'exclusion mutuelle distribuée	38
Figure III. 1 : Topologie proposée.....	46
Figure III. 2 : Scénario proposé.....	47
Figure III. 3 : Organigramme proposé	48
Figure III. 4 : Une partie d'algorithme proposé pour la gestion de spectre coté SU.....	49
Figure III. 5 : Une partie d'algorithme proposé pour la gestion de spectre coté PU	50
Figure III. 6 : Une partie d'algorithme proposé pour la tolérance aux fautes coté SU	51
Figure III. 7 : Une partie d'algorithme proposé pour la tolérance aux fautes coté PU	51
Figure III. 8 : Interface d'accueil	53
Figure III. 9 : Interface principal	54
Figure III. 10 : Un exemple de tableau jeton	54
Figure III. 11 : Résultat obtenu des différentes interactions entre les agents dans l'exemple en utilisant l'agent sniffer	56
Figure III. 12 : Impact d'augmentation du nombre des agents sur le nombre de messages échangés	57
Figure III. 13 : Impact d'augmentation du nombre de PU sur le nombre de messages échangés avant d'entrer en SC.....	58
Figure III. 14 : Impact d'augmentation du nombre de SU sur le temps de réponse.....	59

Figure III. 15 : Impact du nombre de PU sur le nombre de messages échangés en utilisant la méthode FIFO et la fidélité	60
Figure III. 16 : Impact du nombre de SU sur le nombre de messages échangés en utilisant la méthode FIFO et la fidélité	60
Figure III. 17 : Comparaison entre le nombre de SC exécuté en se basant sur la fidélité et sur la méthode FIFO.....	61

Liste des tableaux

Tableau II. 1 : Comparaison entre le système distribué et le système centralisé.....	30
Tableau II. 2 : Comparaison entre l'approche distribuée et l'approche centralisée	35

Liste des abréviations

Acronymes	Signification
ACC	Agent Communication Channel
ACL	Agent Communication Language
AMS	Agent Management System
CORBA	Common Object Request Broker Architecture
DF	Director Facilitator
FCC	Federal Communications Commission
FIFO	First In First Out
FIPA	Foundation for Intelligent Physical Agents
JADE	Java Agent Development
IDL	Interface Definition Language
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
LGPL	Lesser General Public Licence
MAC	Medium Access Control
PU	Primary User
QoI	Quality of Information
RC	Radio Cognitive
RF	Radio Frequency
RL	Reverse Link
SC	Section Critique
SDL	Service Description Language
SDR	Software Defined Radio (radio logicielle restreinte)
SMA	Système Multi-Agents
SU	Secondary User
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
UML	Unified Modeling Language

Introduction générale

Nous vivons une époque où l'évolution de la technologie suit un rythme effréné, les systèmes de communication sans fil font la transition de la téléphonie sans fil aux données internet interactives et aux applications multimédia, pour une transmission de débit plus élevée souhaitée. Au fur et à mesure que de plus en plus d'appareils fonctionnent sans fil, il n'est pas difficile d'imaginer que les technologies futures seront confrontées à un encombrement spectral, et la coexistence d'appareils sans fil sera un problème majeur. Compte tenu de la disponibilité limitée de la bande passante, répondre à la demande de capacités et de débits de données plus élevés est une tâche difficile, nécessitant des technologies innovantes qui peuvent offrir de nouvelles façons d'exploiter le spectre radioélectrique disponible. La radio cognitive (RC) est la technologie passionnante qui offre de nouvelles approches à l'utilisation du spectre. Cette dernière est un nouveau concept pour les communications sans fil futures, et elle a suscité un intérêt considérable. La RC offre une solution tentante au problème de l'encombrement spectral en introduisant un accès dynamique et opportuniste aux ressources partagées qui ne sont pas fortement occupées par leurs utilisateurs autorisés.

Cependant, l'accès dynamique au spectre dans un réseau à radio cognitive soulève plusieurs défis notamment le partage équitable des ressources. C'est pourquoi l'intérêt de trouver des techniques pour résoudre ce problème. L'exclusion mutuelle est l'une des techniques utilisées, elle concerne la façon de contrôler les nœuds pour qu'ils entrent dans la section critique afin d'accéder à une ressource partagée de manière mutuellement exclusive.

Les algorithmes distribués d'exclusion mutuelle sont l'une des briques de base pour de nombreuses applications réparties. Elles les utilisent pour protéger la partie du code (section critique) qui accède à des ressources partagées. Ainsi, on retrouve ce mécanisme dans tous les niveaux des applications réparties. On distingue généralement deux classes d'algorithmes d'exclusion mutuelle distribués. Dans les approches à base de permissions, un nœud peut entrer en section critique uniquement s'il a obtenu la permission de tous les nœuds ou d'une majorité d'entre. Dans les autres approches, un jeton unique donne le droit d'accès à la ressource critique (Sopena, 2008).

Ce projet de fin d'étude prend pour cadre l'utilisation des algorithmes d'exclusion mutuelle distribuée pour le partage des ressources spectrales dans un réseau radio cognitive. Nous avons choisi de nous intéresser plus particulièrement aux algorithmes basés sur les jetons ainsi que la gestion des défaillances.

D'une façon général, notre solution consiste à mettre en place un algorithme de gestion du spectre basé sur la négociation « plusieurs à plusieurs » et bien adapté pour les réseaux mobiles ad hoc. Cet algorithme permet de gérer les éventuels problèmes de la concurrence d'accès aux ressources partagées en se basant sur la notion de fidélité c'est à dire le nombre de fois qu'un processus a déjà accédé à la SC. L'algorithme permet de minimiser un certain nombre de messages échangés dès le départ de simulation. Les résultats obtenus montrent que notre proposition assure de très bonnes performances et qu'elles sont prometteuses en termes de gestion de spectre dans un réseau à radio cognitive.

Notre mémoire est structuré comme suit :

Dans le premier chapitre, nous définirons les principaux concepts liés à l'étude de la RC (sa définition, son fonctionnement, son architecture ainsi que ses différents domaines d'applications).

Dans le second chapitre, nous présenterons d'une façon générale le concept d'exclusion mutuelle distribuée, en énonçant les différentes classes d'algorithmes qui la garantit. Nous présenterons ensuite un état de l'art sur l'utilisation de l'exclusion mutuelle dans la RC qui nous sert de support pour positionner nos travaux de recherche.

Dans le troisième chapitre, nous montrons notre algorithme d'exclusion mutuelle. Ce dernier est réalisé à l'aide du simulateur JADE. Une discussion par rapport aux résultats obtenus est aussi présentée dans ce chapitre.

Chapitre I
Radio Cognitive

I. Introduction

L'évolution des Systèmes sans fil a déclenché une forte demande en terme de ressources spectrales, ces dernières sont devenues de plus en plus précieuses avec la prolifération rapide de standards et de services de radiocommunication. Le problème réside dans le fait que les systèmes sans fil de communication numérique n'exploitent pas l'intégralité de la bande de fréquence disponible. Pour pallier à ce problème, beaucoup de travaux de recherche ont été réalisés, et l'une des initiatives est l'idée de la radio cognitive (RC).

La radio cognitive est une technique prometteuse pour l'utilisation efficace du spectre. Elle doit surveiller l'activité dynamique dans le spectre de l'utilisateur primaire et adapter la transmission des utilisateurs secondaires pour une meilleure allocation des ressources spectrales (Zayen, 2010).

Dans ce chapitre, nous commençons par une définition du concept radio cognitive ainsi qu'un bref historique. Ensuite, nous nous intéressons successivement aux différents aspects de la RC : principe de fonctionnement, fonctions, architecture, etc.

II. Définition

La RC est une technologie émergente en matière d'accès sans fil, elle sert à améliorer l'utilisation du spectre radio en permettant l'accès aux opportunités spectrales disponibles de manière dynamique. Une radio cognitive était définie par Mitola dans son travail séminal comme « un système radio ayant la capacité de détecter, de prendre conscience de son environnement opérationnel et peut ajuster de façon dynamique et autonome ses paramètres de fonctionnement radio en conséquence ». Cette définition a été généralisée par la FCC d'être « un système radio ayant la capacité de détecter son environnement électromagnétique et d'ajuster de façon dynamique et autonome ses paramètres de fonctionnement radio afin de modifier le fonctionnement du système» (Khattab, 2012).

La RC est une forme de communication sans fil dans laquelle un émetteur/récepteur détecte l'existence des canaux de communication inoccupés, et peut se déplacer dans ces canaux. Ceci permet d'optimiser l'utilisation des fréquences radio disponibles du spectre tout en minimisant les interférences avec d'autres utilisateurs (Benmammar, 2014).

Cette capacité permet d'adapter chaque appareil aux conditions spectrales du moment et offre donc un accès plus souple aux utilisateurs, efficace et complet à cette ressource. Cette approche peut améliorer considérablement le débit des données et la portée des liaisons sans augmenter la bande passante ni la puissance de transmissions. La radio cognitive offre également une solution équilibrée au problème de l'encombrement du spectre.

A partir de toutes ces définitions, une radio cognitive a deux caractéristiques principales qui la distinguent d'une radio traditionnelle: la cognition et la reconfigurabilité (Khattab, 2012).

La cognition regroupe les divers processus mentaux allant de l'analyse perceptive de l'environnement à la commande motrice (en passant par la mémorisation, le raisonnement, les émotions, le langage...) (Benmamar, 2014).

La deuxième caractéristique clé est la reconfigurabilité. Elle représente la capacité à s'adapter à divers paramètres radio de communication à savoir les bandes de fréquences, les largeurs de bandes des canaux radio, les techniques de modulation et de codage, etc. et cela sans avoir à changer la partie matérielle. Ceci afin de pouvoir utiliser la nouvelle bande (Nguyen, 2012).

III. Historique

Au cours des 15 dernières années, les notions sur les radios évoluent loin des radios matérielles à base de radios qui impliquent une combinaison de matériel et de logiciels. Au début des années 1990, Joseph Mitola a introduit l'idée des radios à logiciel (SDR). Ces radios sont typiquement des systèmes complexes et reconfigurables permettant de mélanger l'adaptation matérielle et logicielle avec une grande flexibilité. Elles tendent à y rajouter de l'intelligence artificielle et une conscience de l'environnement (Nguyen, 2012).

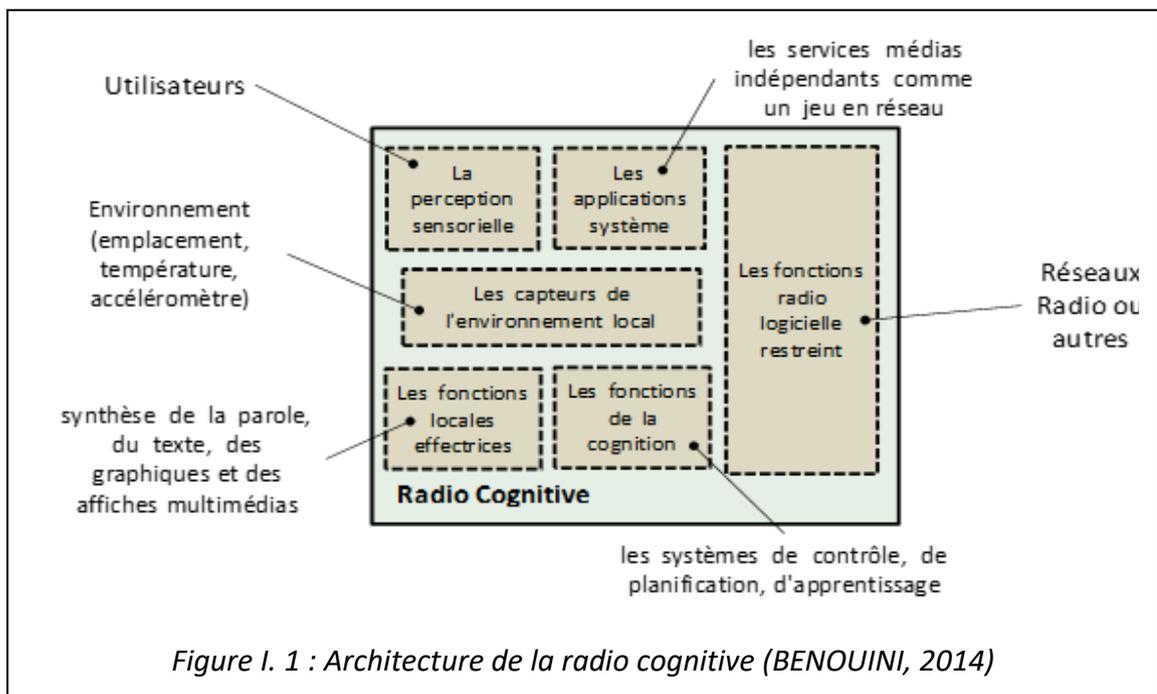
En 1998, Mitola a pris le concept SDR un pas plus loin, en inventant le terme de la radio cognitive. Les RC sont essentiellement des SDR dotés d'une intelligence artificielle capables de détecter et de réagir à leur environnement.

La RC a fait l'objet d'un intérêt particulier comme étant une technologie qui peut améliorer les performances des réseaux sans fil et l'efficacité de l'usage des bandes spectrales (BEN DHAOU, 2011).

IV. Architecture de la RC

L'architecture de la RC est un ensemble complet et cohérent des règles de conception grâce auxquelles un ensemble de composants spécifiés permet d'obtenir un ensemble de fonctions spécifiées dans des produits et des services qui évoluent à travers de multiples points de conception au fil du temps. Cette dernière, dans laquelle des SDR, des capteurs... seront intégrés pour créer une meilleure qualité d'information (QoI) et développer des capacités à observer, orienter, planifier, décider, agir et apprendre l'environnement de l'utilisateur et les ressources fréquentielles (Mitola, 2006).

L'architecture de la RC est composée d'une série de 6 composantes fonctionnelles tels que présentés par la figure I.1 ci-dessous:



Les six composantes fonctionnelles de l'architecture sont:

1. L'interface de perception sensorielle de l'utilisateur comprend les fonctions de capture (sensing) et de perception, visuelles et acoustiques (Bedrane, 2014).
2. Les capteurs de l'environnement local (position, température, accélération, etc.).

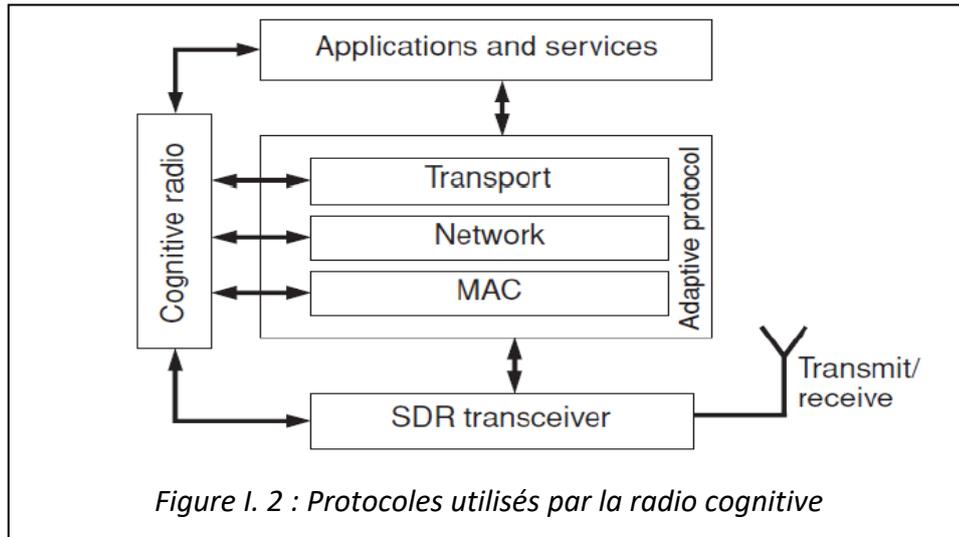
3. Les applications système (les services médias indépendants comme un jeu en réseau).
4. Les fonctions de la radio logicielle restreinte (SDR) (qui incluent la détection Radio Frequency (RF) et les applications radio de la SDR).
5. Les fonctions de la cognition (contrôle, planification, apprentissage).
6. Les fonctions locales effectrices (synthèse de parole, texte, graphiques, affichages multimédias).

Ces éléments fonctionnels sont implémentés sur une plate-forme radio cognitive; une infrastructure matériel-logiciel supporte les six fonctions. Tout d'abord, la traditionnelle interface utilisateur est divisée en un grand sous-système sensoriel et un sous-ensemble distinct d'effecteurs locaux. Ensuite, le sous-système audio n'est pas simplement un encodeur audio pour la transmission, mais aussi il analyse et interprète le son provenant des haut-parleurs pour une haute performance de l'interface du langage naturel parlé. De même, le sous-système texte analyse et interprète le langage pour suivre l'état de l'information de l'utilisateur, en détectant discrètement des plans et des besoins potentiels de communication et d'information, et de façon sécurisée pour protéger les informations personnelles de l'utilisateur qui effectue des activités normales. Les effecteurs synthétisent les discours avec du texte, des graphiques et des affiches multimédias (Mitola, 2006).

L'architecture du protocole de la radio cognitive est représentée dans la figure ci-dessous.

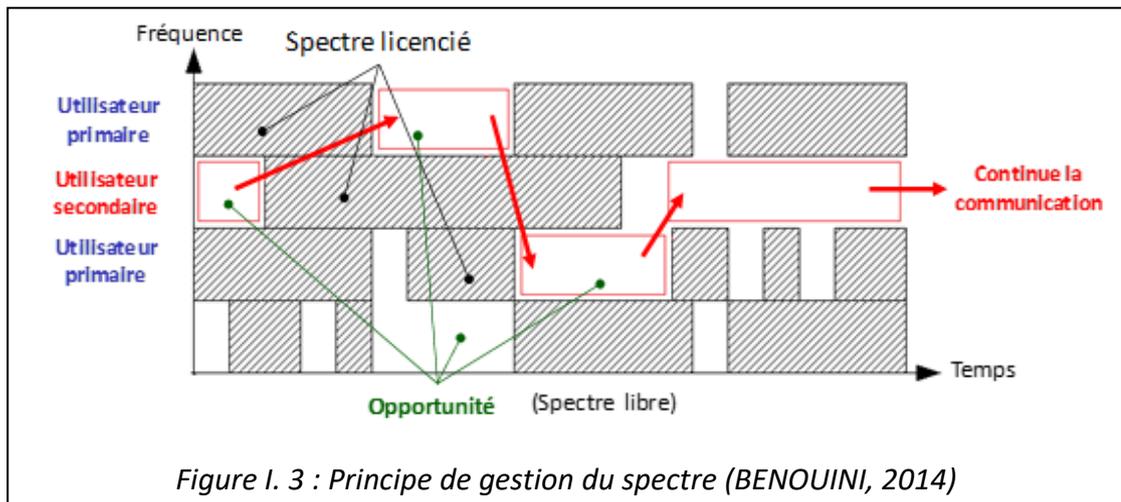
Dans la couche physique, le RF est mis en œuvre à base de radio définie par logiciel. Les protocoles d'adaptation de la couche MAC, réseau, transport, et applications doivent être conscients des variations de l'environnement radio cognitif (Benmammam, 2014).

Pour relier tous les modules, un contrôle radio cognitif est utilisé pour établir des interfaces entre l'émetteur/récepteur SDR et les applications et services sans fil.



V. Principe

Le principe de la RC nécessite une gestion alternative du spectre qui est la suivante: un utilisateur dit secondaire pourra à tout moment accéder à des bandes de fréquence qu'il trouve libres, c'est-à-dire, non occupées par l'utilisateur dit primaire possédant une licence sur cette bande. L'utilisateur secondaire (SU) devra les céder une fois le service terminé ou une fois qu'un utilisateur primaire (PU) aura montré des velléités de connexion (Amraoui, 2012). Ce principe est illustré par la figure I.3.



VI. Fonctions

L'objectif principal de la radio cognitive consiste à gérer le spectre de manière opportuniste de façon à optimiser l'exploitation des ressources radio disponibles (TRIGUI, 2013).

Par conséquent, la RC doit réaliser plusieurs fonctions regroupées dans trois étapes pour assurer son bon fonctionnement à savoir:

VI.1 Détection du spectre (Spectrum sensing)

Cette fonctionnalité permet de détecter ainsi les parties libres du spectre et les partager sans interférence avec d'autres utilisateurs. Ceci permet à un SU de consulter le spectre en vertu de la contrainte d'interférence (Benmammar, 2014).

VI.2 Gestion du spectre (Spectrum management)

Les RC devraient capter les meilleures fréquences disponibles pour répondre aux besoins de communication des utilisateurs, donc les fonctions de gestion du spectre sont nécessaires pour les radios cognitives. Ces fonctions de gestion peuvent être classées comme suit:

VI.2.1 Analyse du spectre

Les résultats obtenus de la détection du spectre sont analysés pour estimer la qualité du spectre. Une des questions ici est de savoir comment mesurer la qualité du spectre qui peut être accédée par un utilisateur secondaire.

VI.2.2 Décision sur le spectre

Le système à radio cognitive doit choisir le canal le plus approprié au besoin de l'utilisateur parmi la liste des bandes de spectres libres détectées. Cette décision doit tenir compte de plusieurs critères afin de choisir le canal adéquat (TRIGUI, 2013).

L'existence de plusieurs utilisateurs, ces utilisateurs peuvent être coopératifs ou non coopératifs dans l'accès au spectre, dans la même bande de spectre nécessite des mécanismes et des protocoles d'allocation pour éviter les collisions et les conflits. Ces

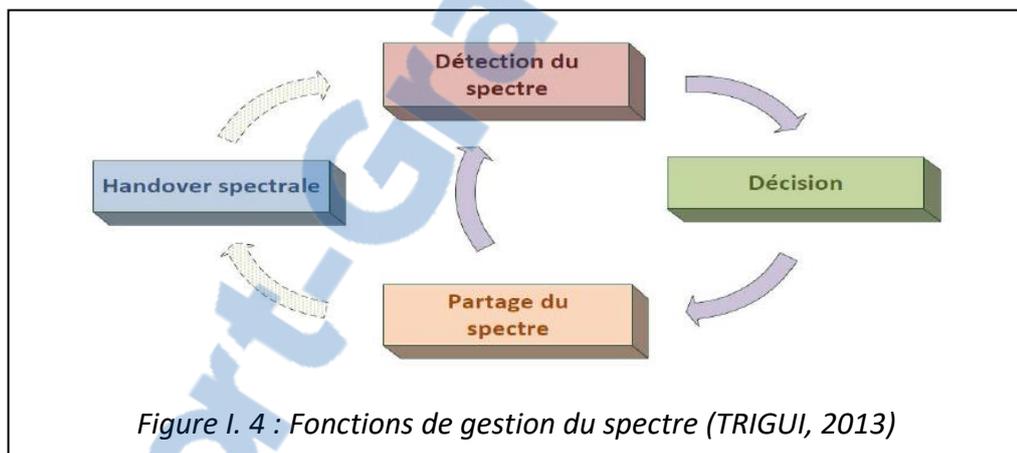
mécanismes permettent de gérer le partage de spectre avec les autres utilisateurs et éviter les interférences.

Dans un environnement non-coopératif, chaque utilisateur a son propre objectif, tandis que dans un environnement coopératif, tous les utilisateurs peuvent collaborer pour atteindre un seul objectif (Benmammar, 2014).

VI.3 Mobilité du spectre (Spectrum mobility)

C'est le processus qui permet à l'utilisateur de la radio cognitive de changer sa fréquence de fonctionnement vers une nouvelle fréquence, meilleure que celle utilisée.

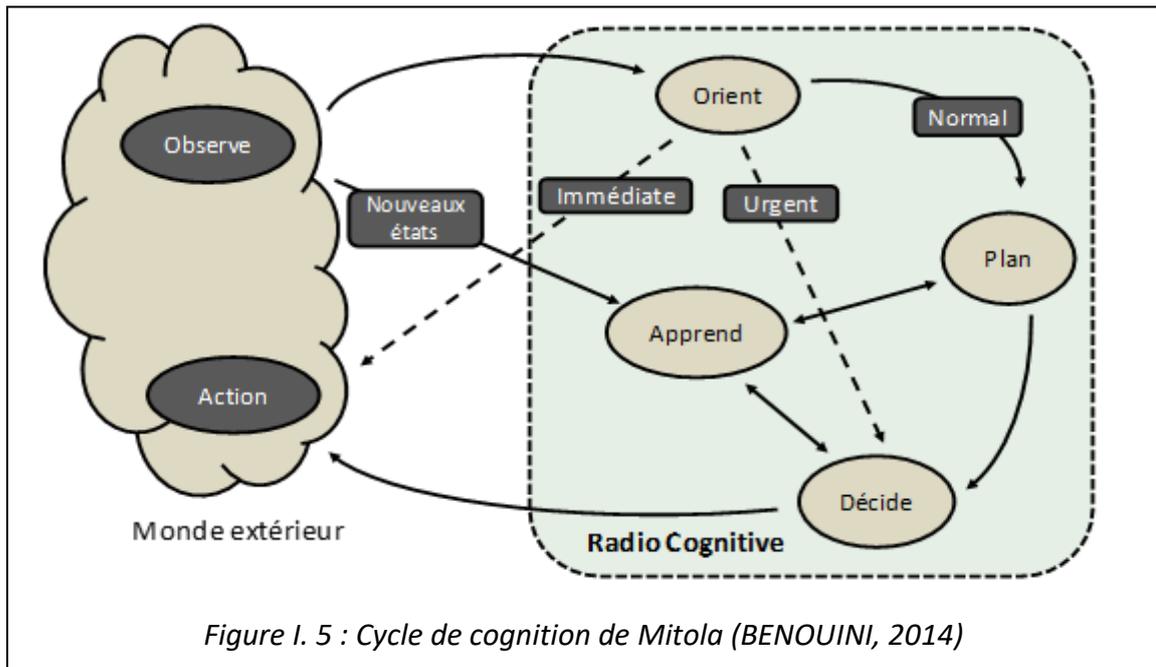
La figure I.4 présente les différentes fonctions de base pour la gestion du spectre dans le cycle de la radio cognitive ainsi que les différentes transitions entre elles.



Après ces trois étapes, la RC sélectionne la bande de fréquence appropriée selon les caractéristiques et les besoins de l'utilisateur. Ces étapes sont connues dans la littérature comme étant le cycle cognitif (BEN DHAOU, 2011).

VII. Cycle cognitif

Un cycle cognitif par lequel une radio cognitive peut interagir avec l'environnement est illustré dans la figure I.5. Les stimuli entrent dans la radio cognitive comme des interruptions sensorielles, envoyées sur le cycle de la cognition pour une réponse. Une telle radio cognitive observe l'environnement, s'oriente, crée des plans, décide, et puis agit (Mitola, 2000).



VII.1 Phase d'observation (détecter et percevoir)

La radio cognitive observe son environnement en analysant les informations. Dans cette phase, lie ces stimuli à des expériences antérieures pour discerner les modèles au fil du temps et déduire le contexte de communication de l'utilisateur.

VII.2 Phase d'orientation

La radio cognitive s'oriente elle-même en déterminant la priorité associée aux stimuli. Une coupure de courant peut directement invoquer un acte (chemin «Immédiate» sur la figure). Une perte de signal sur un réseau peut provoquer une réallocation de ressources. Ceci est réalisé via le chemin étiqueté "Urgent" dans la figure. Cependant, un message de réseau entrant devrait normalement être traité en générant un plan (chemin «Normal»).

VII.3 Phase de planification

La planification comprend la génération de plans. En tant que modèles de causalité sont intégrés dans des outils de planification.

VII.4 Phase de décision

La phase «Décider» sélectionne un plan parmi les plans candidats. La radio peut avoir le choix pour alerter l'utilisateur d'un message entrant ou report et l'interruption jusqu'à plus tard en fonction des niveaux de QoI statués dans cette phase.

VII.5 Phase d'action

"Agir" initie les processus sélectionnés à l'aide de modules effecteurs sélectionnés qui accèdent au monde extérieur ou aux états internes de la radio cognitive. L'accès au monde extérieur consiste principalement à composer des messages qui doivent être envoyés dans l'environnement en audio ou exprimés dans différents langages appropriés. L'accès aux états internes de la radio cognitive, par exemple, l'ajout de nouveaux modèles aux modèles internes existants. L'acquisition de connaissances pourrait être achevée par une action qui crée les structures de données appropriées.

VII.6 Phase d'apprentissage

L'apprentissage est une fonction des observations et des décisions. L'apprentissage initial est réalisé à travers la phase d'observation dans laquelle toutes les perceptions sensorielles sont continuellement comparées à l'ensemble de l'expérience antérieure pour continuellement compter les événements et se souvenir du temps écoulé depuis le dernier événement.

L'apprentissage peut se produire quand un nouveau modèle est créé en réponse à une action. Par exemple, les états internes antérieurs et actuels peuvent être par rapport aux attentes pour apprendre davantage sur l'efficacité d'un mode de communication.

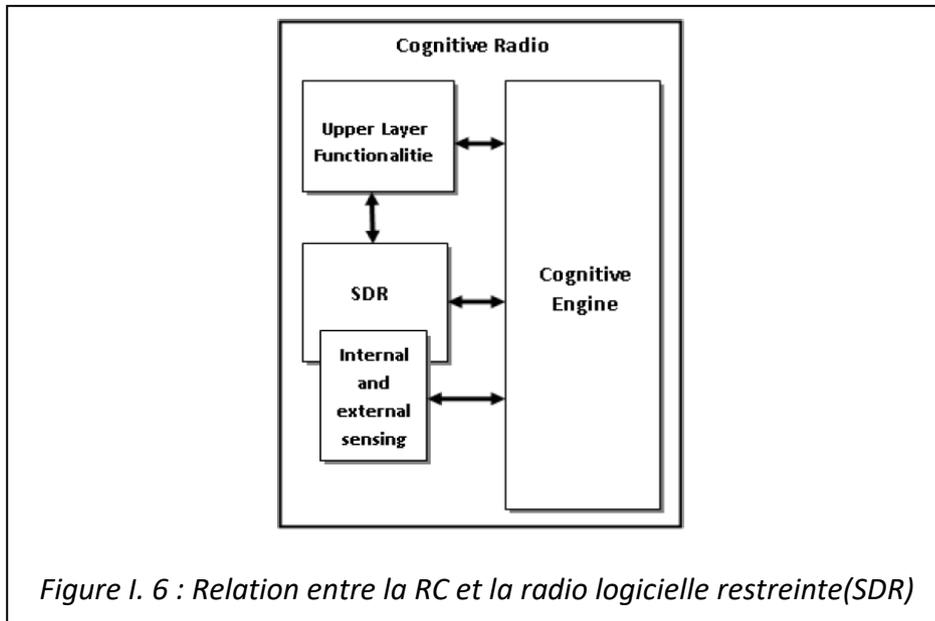
VIII. Relation entre la RC et la radio logicielle restreinte(SDR)

L'une des principales caractéristiques de la RC est la capacité d'adaptation où les paramètres de la radio (fréquence porteuse, puissance, modulation, bande passante) peuvent être modifiés en fonction de : l'environnement radio, la situation, les besoins de l'utilisateur, l'état du réseau, la géolocalisation,...etc (Benmammar, 2014).

La radio logicielle est capable d'offrir les fonctionnalités de flexibilité, de reconfigurabilité et de portabilité inhérentes à l'aspect d'adaptation de la radio

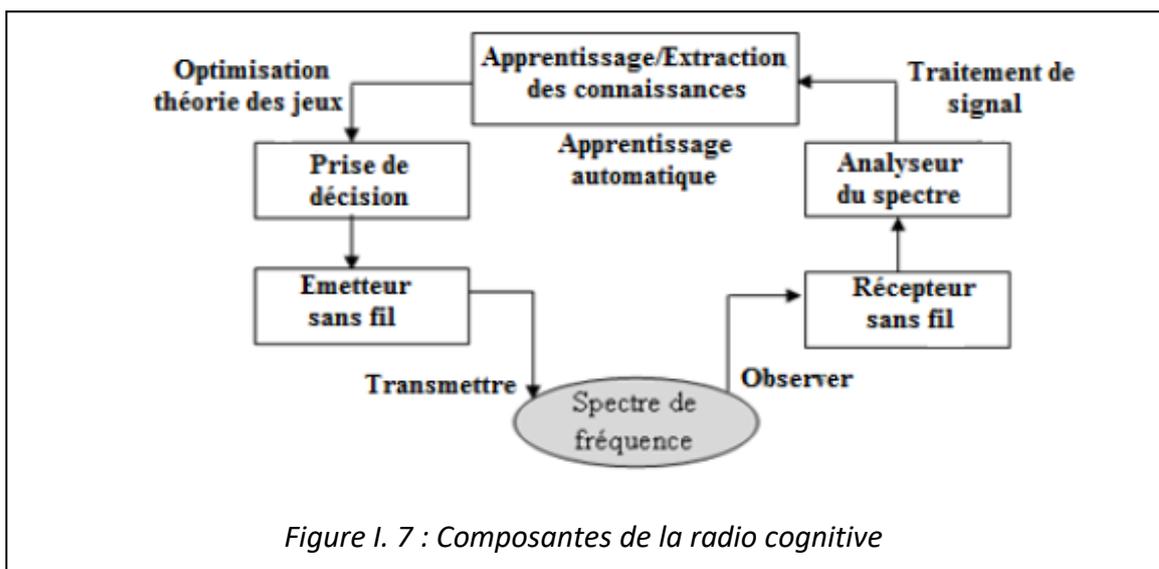
cognitive. Par conséquent, cette dernière doit être mise en œuvre autour d'une radio logicielle. En d'autres termes, la radio logicielle est une "technologie habilitante" pour la radio cognitive.

La figure I.6 montre la relation entre la RC et la radio logicielle restreinte.



IX. Composantes de la RC

Les différentes composantes d'un émetteur/récepteur radio cognitive qui mettent en œuvre ces fonctionnalités sont présentées dans la figure ci-dessous (Benmammar, 2014).



IX.1 Émetteur / Récepteur

Un émetteur/récepteur SDR sans fil est le composant majeur avec les fonctions du signal de transmission de données et de réception. En outre, un récepteur sans fil est également utilisé pour observer l'activité sur le spectre de fréquence (spectre de détection). Les paramètres émetteur/récepteur dans le nœud de la radio cognitive peuvent être modifiés dynamiquement comme dicté par les protocoles de couche supérieure.

IX.2 Analyseur de spectre (Spectrum analyser)

L'analyseur de spectre utilise les signaux mesurés pour analyser l'utilisation du spectre et doit s'assurer que la transmission d'un utilisateur primaire n'est pas perturbée si un utilisateur secondaire décide d'accéder au spectre.

IX.3 Extraction de connaissances et apprentissage (Knowledge extraction/learning)

L'apprentissage et l'extraction de connaissances utilisent les informations sur l'utilisation du spectre pour comprendre l'environnement ambiant RF. Une base de connaissances de l'environnement d'accès au spectre est construite et entretenue, qui est ensuite utilisée pour optimiser et adapter les paramètres de transmission pour atteindre l'objectif désiré sous diverses contraintes.

IX.4 Prise de décision (Decision making)

Après que la connaissance de l'utilisation du spectre soit disponible, la décision sur l'accès au spectre doit être faite. La décision optimale dépend du milieu ambiant, elle dépend du comportement coopératif ou compétitif des utilisateurs secondaires.

X. Langage pour la radio cognitive

Pour qu'une radio cognitive puisse poser des questions et accéder aux informations il est plus important d'avoir des langages standards (BENOUINI, 2014):

- **SDL**: exprime aisément l'état des machines radio, les diagrammes d'ordre de message, et les dictionnaires des données relatifs.

- **UML**: exprime aisément un logiciel objet, y compris des procédures, des cas d'utilisation, etc.
- **IDL** : définit par CORBA pour les Interfaces et l'encapsulation des objets.
- **KQML**: pour l'échange des connaissance par deux primitives **ask**, **tell**.
- **KIF**: pour la traitement axiomatique des ensembles, relations, frames, ontologies.

XI. Domaines d'application

Le concept de la radio cognitive peut être appliqué à une variété de scénarios de communication sans fil, nous allons décrire quelques-uns (BENOUINI, 2014):

XI.1 Les réseaux sans fil de prochaine génération

La radio cognitive devrait être une technologie clé pour la prochaine génération de réseaux sans fil hétérogènes. La radio cognitive fournira des renseignements intelligents à la fois pour l'utilisateur et pour le fournisseur d'équipements.

XI.2 Coexistence de différentes technologies sans fil

Les nouvelles technologies sans fil (IEEE 802.22) sont en cours d'élaboration pour la réutilisation des fréquences radio allouées à d'autres services sans fil (service TV). La RC est une solution qui fournit la coexistence de ces différentes technologies et services sans fil.

XI.3 Services de cyber santé (eHealth services)

La plupart des dispositifs médicaux de soins utilisés sont sans fil et sensibles aux EMI (interférences électromagnétiques) et EMC (compatibilité électromagnétique).

En outre, différents dispositifs biomédicaux (équipement et appareils chirurgicaux, de diagnostic et de suivi) utilisent la transmission RF.

Dans ce cas, la RC peut être appliquée pour l'utilisation du spectre de ces dispositifs mais doit éviter toute interférence avec l'autre. Coexistence de différentes technologies sans fil:

XI.3.1 Réseaux d'urgence

Les réseaux de sécurité publique et d'urgence peuvent profiter des concepts de la radio cognitive pour fournir la fiabilité et la flexibilité de communication sans fil.

Ce genre de réseau peut utiliser le concept de la radio cognitive pour permettre la transmission sans fil et la réception sur une large gamme du spectre radio.

XI.3.2 Réseaux militaires

Avec la RC, les paramètres de la communication sans fil peuvent être adaptés de manière dynamique en fonction du temps et de l'emplacement ainsi que de la mission des soldats.

XII. Conclusion

Dans ce chapitre, nous avons présenté des notions importantes concernant la radio cognitive, ainsi que ses principes, en passant par son architecture, fonctions de la RC, cycle de cognition et les composantes de la RC jusqu'aux domaines d'application de la radio cognitive.

Dans le prochain chapitre, nous présenterons d'une façon générale le concept d'exclusion mutuelle distribuée, en énonçant les différentes classes d'algorithmes qui la garantit.

Chapitre II
Exclusion Mutuelle
Distribuée

I. Introduction

Un système distribué est un système constitué d'un ensemble de processus qui accèdent simultanément à certaines ressources. L'accès concurrent a posé de nouveaux conflits qui ont été résolus au moyen de l'exclusion mutuelle.

L'exclusion mutuelle définie par Dijkstra est l'un des paradigmes fondamentaux des systèmes distribués pour garantir des accès cohérents aux ressources partagées. Elle doit assurer qu'au plus un processus peut exécuter une portion de code manipulant une ressource partagée, appelée section critique (SC), et de satisfaire toute demande d'accès à la SC au bout d'un temps fini (Lejeune, 2013).

Plusieurs algorithmes d'exclusion mutuelle existent dans la littérature. Ils peuvent être divisés en deux catégories : les algorithmes à permissions et les algorithmes à jeton. Dans ce mémoire nous nous intéresserons à l'exclusion mutuelle distribuée à jeton.

Dans ce chapitre, nous commençons néanmoins par une comparaison entre un système distribué et un système centralisé. Ensuite, nous donnons une définition formelle de l'exclusion mutuelle (les différents états des processus, les principales propriétés de l'exclusion mutuelle, ...). Nous finissons respectivement par une classification des différentes classes des principaux algorithmes d'exclusion mutuelle distribuée et un état de l'art ainsi qu'une conclusion.

II. Différence entre un système distribué et un système centralisé

Dans les systèmes informatiques, on distingue deux grands modèles de systèmes : système distribué et système centralisé. Un système distribué est un système qui s'exécute sur un ensemble de machines sans mémoire partagée (chacun a sa mémoire locale) et sans horloge globale, mais qui apparaît à l'utilisateur comme un seul système cohérent. Un système centralisé en opposition à ce dernier est un système qui s'exécute sur un ensemble de composants partageant une mémoire et une horloge, ces composants accèdent aux ressources informatiques résidant dans un serveur central qui est responsable de traitement des requêtes.

Le tableau suivant présente une comparaison entre les différentes caractéristiques des deux modèles.

Systèmes centralisés	Systèmes distribués
Traitement des tâches et données centralisé.	Traitement des tâches et données distribué.
Difficile à réaliser la tolérance aux pannes.	Tolérance aux pannes.
Si le système principal échoue, tout s'arrête.	La redondance des données et des applications est probable.
Ils ont des composantes non autonomes.	Ils ont des composantes autonomes malgré qu'elles pourraient être multiples.
Ils sont souvent construits en utilisant une technologie homogène.	Ils peuvent être construits en utilisant une technologie hétérogène.
Les utilisateurs multiples partagent en permanence les ressources d'un système centralisé.	Les composants du système distribué peuvent être utilisés exclusivement.
Ils ont un seul point de contrôle.	Ils sont exécutés dans des processus simultanés.
Ils ont un seul point d'échec.	Ils ont plusieurs points d'échec.

Tableau II. 1 : Comparaison entre le système distribué et le système centralisé

III. Définition de l'exclusion mutuelle

D'après la définition du problème de l'exclusion mutuelle donnée par Dijkstra en 1965 (Dijkstra, 1965). On retrouve les principales propriétés qui doivent être vérifiées par tous les algorithmes d'exclusion mutuelle à savoir : les états des processus, les propriétés fondamentales définissant l'exclusion mutuelle, l'ordonnancement des requêtes et les métriques d'évaluation de performances.

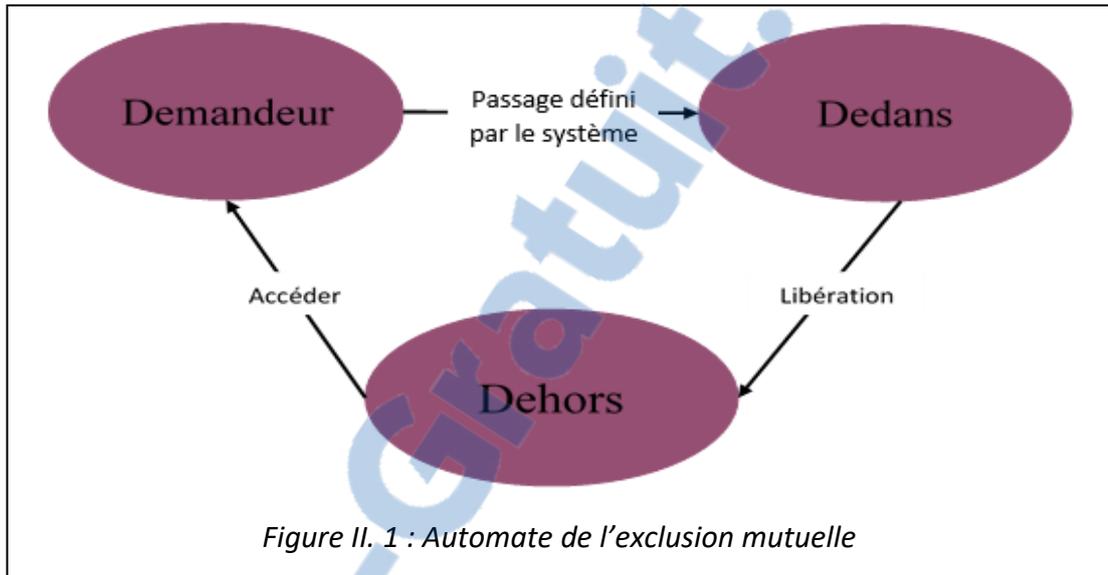
III.1 États des processus

La figure ci-dessous représente l'automate de l'exclusion mutuelle constituée des trois états suivants :

- Dehors : état dans lequel se trouve un processus quand il exécute du code ne

nécessitant pas l'accès à la SC.

- Demandeur : état dans lequel se trouve un processus quand il attend de pouvoir exécuter une SC.
- Dedans : état dans lequel se trouve un processus quand il exécute une section critique (Sopena, 2008).



A partir des états définis précédemment, on peut effectuer trois changements circulairement de la façon suivante :

- De dehors à demandeur : événement provoqué par un processus lorsque celui-ci souhaite entrer en SC en faisant appel à un message Request.
- De demandeur à dedans : événement indiquant qu'un processus peut entrer en SC.
- De dedans à dehors : événement provoqué par un processus lorsque celui-ci sort de la SC en faisant appel à un message Release (Lejeune, 2014).

III.2 Propriétés

Tout algorithme réparti résout le paradigme de l'exclusion mutuelle si et seulement s'il respecte les deux propriétés suivantes :

- Sûreté : au plus un processus est à la fois dans la section critique (dans l'état dedans).
- Vivacité : tout processus demandant à entrer dans la section critique (à passer

dans l'état dedans) y entre en un temps fini.

Autrement dit, le premier invariant implique que l'exclusion mutuelle à une seule ressource est garantie à tout moment. Tandis que le second indique qu'une situation d'interblocage ne se présente jamais (Thiare, 2007b).

III.3 Ordonnement

L'ordonnement consiste à étudier les méthodes d'organiser et de planifier l'exécution d'opérations au cours du temps de façon à respecter un ensemble de contraintes.

Le problème de la famine apparaît lorsqu'une requête de haute priorité peut empêcher en permanence les autres requêtes de plus basses priorités d'exécuter la SC. Cependant, ce problème est fortement lié à la définition de l'ordre total dans lequel les requêtes ont lieu; ces dernières peuvent être pénalisés par un temps d'attente infini pour accéder à la SC, ce qui viole ainsi la propriété de vivacité (Lejeune, 2013). Afin d'éviter le problème précité, une stratégie a été proposée qui exige que toute requête doit devenir en un temps fini la plus prioritaire de cet ordre (Lejeune, 2014).

Dans la majorité des algorithmes, les requêtes sont satisfaites selon une politique d'ordonnement "First In First Out" (FIFO) grâce à une horloge logique sur les requêtes ou bien grâce à l'horloge physique du détenteur du jeton (Lejeune, 2013).

Cette politique d'ordonnement permet d'assurer la vivacité, ainsi toute requête deviendra en un temps fini la plus prioritaire (Lejeune, 2014).

III.4 Extensions de l'exclusion mutuelle

Le problème de l'exclusion mutuelle classique permet de gérer un accès FIFO sur une ressource partagée. Cependant, diverses extensions de ce problème fondamental peuvent être trouvées à savoir :

III.4.1 Exclusion mutuelle à priorité

L'exclusion mutuelle à priorité permet de définir un ordre d'accès en SC en associant à chaque requête un niveau de priorité. Cependant, la prise en compte des priorités peut conduire à des famines dans le cas où des demandes de SC les plus prioritaires empêchent la satisfaction des moins prioritaires (Lejeune, 2013).

III.4.2 Exclusion mutuelle à contraintes de temps

Dans ce cas, le critère de priorité repose sur le principe de réservation d'un délai d'échéance requis, dont l'objectif est de satisfaire toutes les requêtes avant la fin de ce délai (Lejeune, 2014).

III.4.3 Exclusion mutuelle généralisée

Il existe plusieurs généralisations de problème d'exclusion mutuelle initial. La généralisation peut se faire soit en augmentant le nombre d'exemplaires de la ressource soit en considérant plusieurs types de ressource en un seul exemplaire ou bien l'union des deux extensions précédentes en considérant plusieurs ressources en plusieurs exemplaires.

III.4.4 Exclusion mutuelle de groupe

L'exclusion mutuelle de groupe se produit naturellement dans une situation où une ressource peut être partagée par des processus du même groupe, mais pas par des processus de différents groupes. Il est aussi connu sous le nom de la "conversation des philosophes".

Toutes ces extensions peuvent être combinées. À titre d'exemple, on peut citer l'algorithme de Swaroop-Singh qui combine l'exclusion mutuelle de groupe avec l'exclusion mutuelle à priorité.

III.5 Métriques d'évaluation de performances

La performance des algorithmes d'exclusion mutuelle est généralement mesurée par les quatre mesures suivantes:

- Complexité des messages: ceci est le nombre de messages requis pour chaque demande d'entrée en SC.
- Délai de synchronisation: c'est le temps nécessaire après qu'un site libère la SC et avant que le site suivant n'entre à la SC. Notez qu'habituellement un ou plusieurs échanges de messages séquentiels peuvent être nécessaires entre la libération du site de la SC et l'entrée du site suivant dans la SC.

- Temps de réponse (temps d'attente): c'est l'intervalle de temps pendant lequel un site attend que son exécution de SC soit autorisée après que ses messages de requête aient été envoyés.
- Débit système: c'est la vitesse à laquelle le système exécute les requêtes pour la SC (Kshemkalyani, 2011).

III.6 Mécanismes d'exclusion mutuelle

En système distribué, deux approches peuvent être utilisées pour implémenter un mécanisme d'exclusion mutuelle. Dans une approche centralisée, l'un des nœuds fonctionne comme un coordinateur central. Les processus demandent seulement au coordinateur d'obtenir l'autorisation d'entrer dans leur SC. Ce n'est que lorsqu'un processus demandeur reçoit l'autorisation du coordinateur qu'il peut procéder à entrer dans sa SC. Le coordinateur central est responsable d'avoir toutes les informations du système et d'accorder l'autorisation d'utiliser une ressource partagée (Velazques, 1993).

Dans une approche distribuée, la prise de décision est répartie sur l'ensemble du système et se fait via des messages échangés entre les processus.

Les algorithmes d'exclusion mutuelle distribués peuvent être classifiés en deux catégories. Dans la première, les solutions sont basées sur les jetons, l'exclusion mutuelle est un message spécial entre les processus, connu comme un jeton. La seconde contient des algorithmes basées sur la permission, dans ce cas l'accès à la source nécessite d'avoir l'autorisation de tous les autres processus.

Le tableau suivant représente une comparaison entre les deux mécanismes prédéfinis.

Mécanismes d'exclusion mutuelle distribué	Mécanismes d'exclusion mutuelle centralisé
Processus ne communiquent que par échange de messages.	Si coordinateur a un problème, tout le système s'effondre.
Chaque processus connaît l'ensemble des processus du système.	Si processus bloqués lors de demande d'accès à une SC occupée : impossibilité de détecter la panne du coordinateur.

Chaque processus a ses propres variables, pas de partage de variables.	Pas de supposition sur l'ordre des messages.
Les échanges de messages sont FIFO: les messages ne se doublent pas.	Pas de famine (aucun processus ne reste bloqué).

Tableau II. 2 : Comparaison entre l'approche distribuée et l'approche centralisée

IV. Taxonomie des algorithmes distribués

Les algorithmes d'exclusion mutuelle se classent usuellement en deux grandes familles : les algorithmes basés sur les permissions et ceux basés sur la circulation d'un jeton. Cette classification a été introduite en 1991 par Michel Raynal.

Dans cette section nous décrivons les principaux algorithmes d'exclusion mutuelle distribuée en discutant leurs caractéristiques et leurs performances.

IV.1 Algorithmes à permissions

Le principe des algorithmes à base de permissions est très simple. Lorsqu'un processus désire entrer en SC, il demande à une partie des processus leur permission. Avant d'entrer effectivement en SC, il attend que toutes les permissions demandées arrivent. Lorsqu'un processus reçoit une demande de permission, s'il n'est pas intéressé par la SC, il envoie une permission au processus demandeur (Conan, 2016). Les algorithmes à permissions peuvent être encore subdivisés en trois sous-catégories: permissions individuelles, permissions d'arbitre et permissions généralisées.

IV.1.1 Permissions individuelles

Dans ce type, un site peut donner sa permission à plusieurs autres sites simultanément. S'il n'est pas demandeur, il envoie son autorisation à tous les sites demandeurs, dans le cas contraire, c'est-à-dire s'il est demandeur, il ne donnera son autorisation qu'aux sites dont les requêtes sont plus anciennes (Sopena, 2008).

IV.1.2 Permissions d'arbitre

Contrairement aux algorithmes à permissions individuelles où un site finissait par recevoir l'accord de tous les sites pour chaque entrée en SC, un site ayant reçu plusieurs

demandes, n'accorde son autorisation qu'à l'une d'entre elles à un moment donné ; les autres sont mises en attente.

La deuxième différence des permissions d'arbitres par rapport aux permissions individuelles, est qu'un site arbitre n'envoie qu'une permission à la fois.

IV.1.3 Permissions généralisées

Cette dernière sous-classe d'algorithme utilise, comme pour les permissions d'arbitres, des sous-ensembles pour diffuser ses requêtes diminuant ainsi la complexité en nombre de messages. Mais, à l'instar des algorithmes à permissions individuelles, les processus peuvent ici émettre plusieurs permissions simultanément, ce qui écarte tout risque d'interblocage.

IV.2 Algorithmes à jeton

Dans l'approche basée sur les jetons, un jeton unique est partagé entre les sites. Un site est autorisé à entrer en SC s'il possède le jeton et il continue à tenir le jeton jusqu'à ce que l'exécution du SC soit terminée. Un site contenant le jeton peut entrer en SC à plusieurs reprises jusqu'à ce qu'il envoie le jeton à un autre site. Selon la manière dont un site effectue la recherche du jeton, il existe de nombreux algorithmes basés sur des jetons. Cette abstraction de l'exclusion mutuelle permet d'obtenir facilement la propriété de sûreté grâce à l'unicité du jeton. Pour retrouver le jeton, on distingue deux stratégies d'algorithmes qui sont couramment utilisées : la première utilise des topologies non structurées tandis que la deuxième repose sur des topologies structurées (Kshemkalyani , 2011).

IV.2.1 Algorithmes à jeton structurés

La présente approche possède un aspect structurel: les sites du système sont structurés sous forme d'une configuration logique bien particulière pour retrouver ou acheminer le jeton. Il est à noter, que le choix de la structure logique peut être imposé par l'architecture physique du système (Thiare, 2007b).

Ainsi, les algorithmes de cette approche peuvent être basés soit sur une structure logique statique dans le cas où les liens logiques restent les mêmes durant toute l'exécution et seule leur direction peut changer, soit sur une structure logique

dynamique dans le cas contraire i.e., la topologie peut évoluer au cours de l'exécution de l'algorithme (Lejeune, 2014).

IV.2.1.1 Topologie statique

On distingue trois types de topologies qui n'évoluent pas avec le système, tels que l'arbre, l'anneau et le graphe.

- Anneau : dans ce cas, les sites sont placés sur un anneau unidirectionnel. Cet anneau est utilisé pour véhiculer le jeton. Un site qui désire entrer en SC doit attendre de recevoir le jeton. Cette classe d'algorithmes assure la propriété de vivacité.
- Arbre : en imposant cette structure, le jeton est toujours détenu par le site situé à la racine de l'arbre. Ainsi les requêtes pour obtenir le jeton voyagent toujours des feuilles vers la racine de l'arbre et le nœud ayant obtenu le jeton en dernier devient la racine de l'arbre (Ortiz, 2009).
- Graphe : tous les nœuds sont placés dans une topologie arbitraire. Les requêtes d'un site se propagent sur le réseau par un mécanisme d'inondation. Le jeton est ensuite retransmis de site en site jusqu'à son destinataire (Lejeune, 2014).

IV.2.1.2 Topologie dynamique

Ces algorithmes vont faire évoluer la structure logique en fonction des requêtes. Grâce à cette structure, on peut refléter l'historique des accès à la SC ainsi d'améliorer les performances des sites qui ont un accès fréquent à la SC.

IV.2.2 Algorithmes à jeton non-structurés

Dans ce groupe d'algorithmes, pas de topologie logique imposée sur les sites. Un processus qui veut demander la SC peut envoyer sa requête en parallèle à tous les autres sites. L'ordre des requêtes peut être sauvegardé dans une file d'attente incluse dans le jeton. Ces algorithmes basés sur la diffusion de la demande peuvent être aussi divisés en deux groupes:

IV.2.2.1 Algorithmes statiques

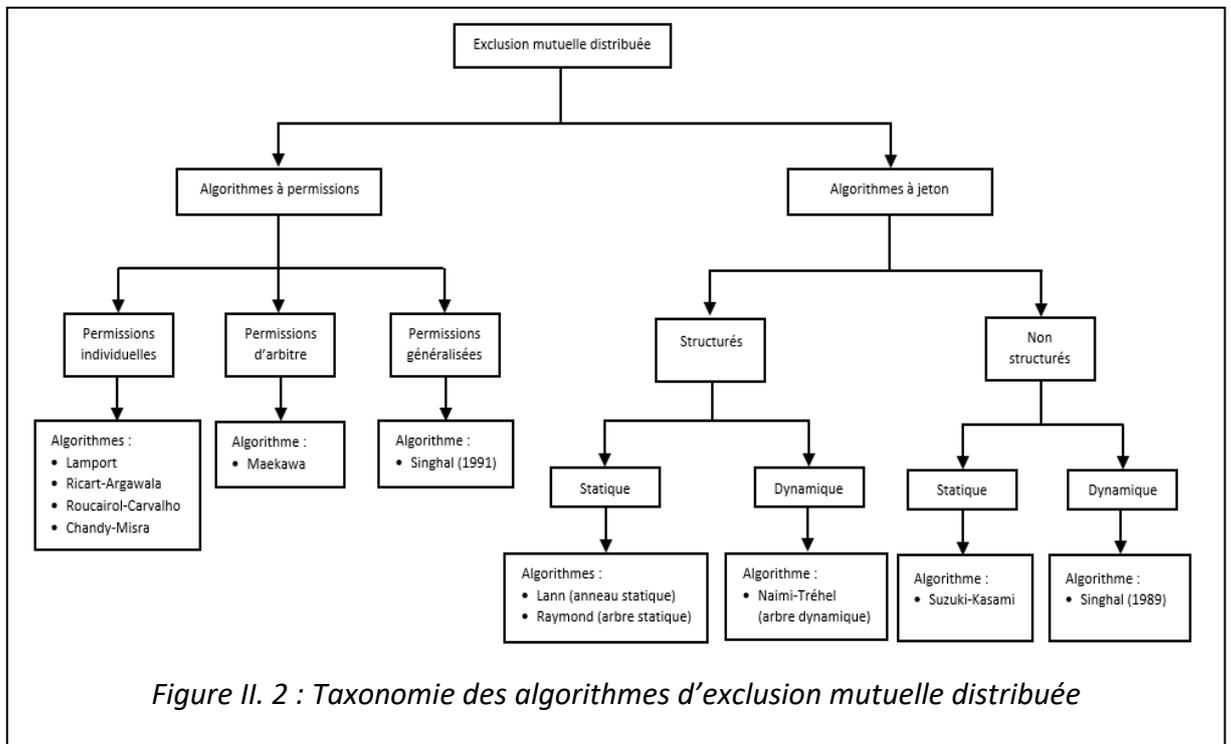
Ces algorithmes ne sauvegardent pas l'historique des différentes exécutions en SC. Un site qui a besoin du jeton pour entrer en SC adresse sa demande à tous les autres sites.

Le site possédant le jeton, une fois sorti de sa SC, le renverra au site le plus ancien dont la requête n’a pas encore été satisfaite.

IV.2.2.2 Algorithmes dynamiques

Dans ce type d’algorithmes, tous les sites sont susceptibles de posséder le jeton, car chacun d’entre eux maintien un historique des derniers demandeurs qui va lui diffuser la requête. Ceci a permis de réduire le nombre de messages et éviter de solliciter les sites qui utilisent peu souvent la SC.

La figure suivante représente un arbre résumant la taxonomie des différents algorithmes d’exclusion mutuelle distribuée.



V. Etat de l’art

L'un des paradigmes les plus classiques de l'informatique distribuée est l'exclusion mutuelle distribuée. Elle consiste à définir un protocole exécuté par un ensemble de processus qui souhaitent se coordonner pour accéder à la SC, qui ne peut être utilisée que par un processus à la fois. Au cours des dernières années, plusieurs algorithmes d'exclusion mutuelle distribuée ont été proposés pour les réseaux ad hoc. Parmi ceux-ci, on trouve ceux proposés dans (Baldoni, 2002), dans cet algorithme le jeton est passé sur un anneau logique créé dynamiquement où chaque fois qu'un processus reçoit le jeton

de son prédécesseur, il décide à la volée en utilisant une politique qui sélectionne comme processus successeur parmi ceux qui n'ont pas encore reçu le jeton. Chaque tour de l'anneau a un coordinateur qui est le seul processus qui peut bloquer le jeton et entrer dans un état inactif. Lorsqu'un nœud souhaite entrer en SC, il envoie une requête au coordinateur, déclenchant ainsi une circulation de jeton. Cela garantit que des scénarios optimaux de circulation de jeton pourraient se produire lorsque la demande de charge est lourde, et en n'émettant aucun message quand aucun processus ne demande la SC. Cependant, un autre algorithme nommé Reverse Link (RL) est basé sur l'échange de jeton a été proposé dans (Walter, 1998). Le système contient un ensemble de nœuds et de ressources partagées, communiquant par un modèle à passage de messages sur un réseau sans fil. RL construit un graphe acyclique direct orienté par jeton. Ce graphe maintient plusieurs chemins menant au nœud contenant le jeton. Cet algorithme suppose que le réseau ne partitionne pas, les nœuds n'échouent pas et la perte de messages ne se produit pas. Dans l'algorithme RL, lorsqu'un nœud souhaite accéder à la ressource partagée, il envoie un message de demande le long d'une des liaisons de communication. Chaque nœud maintient une file d'attente contenant les identifiants des nœuds de voisinage à partir desquels il a reçu la demande pour le jeton. L'algorithme RL a été étendu respectivement en (Walter, 2001) et (Thiare, 2009) pour traiter le problème d'exclusion k-mutuelle. Ce dernier permet à la plupart des processus d'accéder à leurs sections critiques simultanément. Ceci pourrait être la conséquence où plusieurs copies identiques d'une ressource sont rendues disponibles dans le système ou que la ressource par sa nature peut être accédée par un ensemble de processus simultanément.

Dans le problème d'exclusion mutuelle de groupe (GME), plusieurs ressources sont partagées entre les nœuds. Les nœuds demandant d'accéder à la même ressource partagée peuvent le faire simultanément. Cependant, si les nœuds concourent pour accéder à des ressources différentes, un seul d'entre eux peut procéder. Les algorithmes pour résoudre le problème GME pour les réseaux mobiles ad hoc sont donnés dans (Jiang, 2002), (Thiare, 2007a) et (Myoupo, 2009). Ces algorithmes proposés sont adaptés à partir de l'algorithme RL. Dans ces algorithmes basés sur un jeton, le concept de lancer de poids est appliqué pour détecter que tous les nœuds accédant simultanément à la même ressource ont terminé leurs tâches. Ces algorithmes satisfont à l'exclusion mutuelle, au retard délimité et aux propriétés d'entrée simultanées. Ils sont

sensibles à la formation de liens et à la rupture de liaison et convient donc aux réseaux mobiles ad hoc.

VI. Conclusion

Dans ce chapitre, nous avons défini formellement l'exclusion mutuelle suivi d'une présentation des deux familles d'algorithmes qui résolvent le problème d'exclusion mutuelle distribuée. Ces algorithmes diffèrent dans les propriétés des opérations de base sur lesquelles ils s'appuient pour résoudre le problème de l'exclusion mutuelle.

Le concepteur du système, tout en choisissant un algorithme particulier, doit prendre soin des caractéristiques de performance de l'algorithme, de l'exigence de l'application et de l'aspect de la mise en œuvre. Il n'existe donc pas une catégorie meilleure que l'autre mais plutôt des classes adaptées à un contexte et des objectifs de performances précis.

Dans le chapitre suivant, nous allons étudier plus précisément notre algorithme et sa propre version tolérante aux pannes. Dans la première partie nous présenterons la plateforme de simulation qui a servi à la réalisation de notre application. Puis, dans la seconde partie nous étudierons les résultats obtenus sous forme d'une étude comparative.

Chapitre III

*Contribution et
résultats*

I. Introduction

Le problème de l'exclusion mutuelle a été largement étudié dans les systèmes distribués. Les solutions proposées peuvent être classées principalement dans les protocoles basés sur la permission et basés sur un jeton. Certaines des solutions proposées considèrent la topologie physique des réseaux et d'autres, imposent une structure logique sur le réseau comme un anneau ou un arbre. Récemment, ce problème a suscité un intérêt pour les réseaux mobiles ad hoc. Ces réseaux sont connus comme un domaine difficile en raison de leur mobilité, des déconnexions, des pannes des nœuds qui peut entraîner des changements de topologie fréquents et imprévisibles. Ainsi, les solutions d'allocation de ressources (exclusion mutuelle) fournies pour les réseaux fixes ne peuvent pas être appliquées aux réseaux mobiles ad hoc. À notre connaissance, peu d'algorithmes ont été proposés dans la littérature et la plupart sont basés sur une approche à jeton (Baldoni, 2002) et (Walter, 1998).

Les réseaux mobiles sans fil peuvent être classés en deux types : les réseaux avec infrastructure qui utilisent généralement le modèle de la communication cellulaire et les réseaux sans infrastructure ou les réseaux ad hoc. Un réseau ad hoc est un réseau sans fil spontané, auto-organisé, ne reposant sur aucune infrastructure. Les éléments (nœuds) qui le composent sont généralement reliés par des ondes radio (DRID, 2007).

La radio cognitive est une approche novatrice en matière d'accès sans fil, cette technologie de pointe a permis d'utiliser le spectre radio (c'est-à-dire les radiofréquences) de manière entièrement nouvelle et opportuniste.

Dans ce chapitre, nous nous intéressons à établir le lien entre la radio cognitive et les réseaux ad hoc. Pour cela, nous allons présenter une approche qui permet de gérer les éventuels problèmes d'interférences susceptible d'apparaître entre un utilisateur primaire (PU) possédant une licence sur le spectre et un utilisateur secondaire (SU) qui va allouer des canaux sur ce spectre (Bedrane, 2014).

Dans un environnement ad hoc, les mobiles se partagent les ressources présentes dans le réseau. L'utilisation simultanée d'une ressource partagée par plusieurs nœuds peut produire une situation incohérente. L'exclusion mutuelle dans les réseaux ad hoc est considérée comme l'une des solutions de cette problématique. Pour maintenir l'intégrité d'une ressource partagée, un algorithme d'exclusion mutuelle doit vérifier la propriété

de vivacité ainsi que la propriété de la sûreté. Ces deux propriétés garantissent l'absence des situations d'interblocage ou de conflit dans un réseau ad hoc (DRID, 2007).

Dans ce chapitre, nous proposons un algorithme d'exclusion mutuelle basé sur les jetons où il n'y a qu'un seul jeton pour chaque utilisateur primaire à n'importe quel moment donné.

Nous avons choisi d'implémenter un algorithme à base de jetons car il est relativement moins coûteux en termes de messages et plus extensible que les algorithmes à permission. En contrepartie, ils sont particulièrement sensibles aux défaillances, car un nœud peut tomber en panne, quitter ou intégrer le réseau, à tout moment. Les mouvements des nœuds peuvent engendrer des créations et / ou des pertes de liens. Ce qui nous a forcé à traiter dans notre algorithme le cas de la tolérance aux fautes car il présente de très bonnes performances en absence de fautes.

Par rapport aux autres approches, notre algorithme vise, d'une part, à minimiser les nombres de diffusions lors des recouvrements et, d'autre part, à conserver l'équité des requêtes en présence de fautes.

Dans ce chapitre, nous commençons néanmoins par une exposition des différentes suppositions. Ensuite, nous donnons une définition formelle des systèmes multi-agents. Puis, nous présentons respectivement les outils qu'on a utilisé pour réaliser notre application, la topologie de réseau utilisée, le scénario et l'algorithme proposés ainsi que le principe de fonctionnement de ce dernier. Nous finissons par une présentation des résultats obtenus et une comparaison entre ces différents résultats suivi par une conclusion.

II. Suppositions

Dans notre algorithme basé sur les jetons, la possession d'un jeton permet d'accéder aux sections critiques. Nous supposons qu'un réseau non structuré est composé d'un nombre variable d'agents mobiles. Les agents utilisent un support de communication sans fil, les liens physiques de communication sont considérés fiables, sécurisés et bidirectionnels, permettant à n'importe quel agent de communiquer directement avec tout autre par envoi de messages. Nous prétendons que ces hypothèses les rendent bien adaptés à l'environnement radio cognitive sachant qu'ils sont particulièrement sensibles aux pannes.

Nous supposons également que chaque agent PU possède son propre jeton car chaque PU à ses propres ressources pour nous permettre de satisfaire le plus grand nombre de demandes et pour garantir l'unicité de jeton.

III. Système Multi-Agents (SMA)

Un Système Multi-Agents (SMA) est un système distribué composé d'un ensemble d'agents, qui doivent être capables de travailler et d'agir les uns avec les autres dans un environnement commun afin de résoudre les conflits éventuels et pour proposer des solutions réactives et robustes à des problèmes complexes, ce qui réduit le temps de résolution (Larbi , 2014).

Les SMA sont extensibles et adaptatifs, ce qui permet d'ajouter ou d'enlever un agent du système sans causer de problèmes. Dans les réseaux sans fil, nous pouvons modéliser les nœuds RC comme étant des agents. Les SMA sont connus aussi par leur rapidité car les agents peuvent travailler en parallèle pour résoudre leurs problèmes.

Un agent est une entité réelle ou virtuelle, capable de le percevoir et d'agir sur elle-même et sur l'environnement, de communication avec d'autres agents.

IV. Outils de développement

La réalisation de notre application a été faite sous Netbeans en se basant sur la plateforme JADE avec l'utilisation de l'outil JFreeChart pour la création des graphiques et des diagrammes de très bonne qualité.

IV.1 Plateforme JADE

Il existe des plateformes qui permettent la prise en charge des fonctions de base d'un simulateur multi-agents. Parmi ces plateformes on trouve JADE (Java Agent DEvelopment) qui prend en compte nativement le parallélisme, elle est utilisée pour la réalisation de notre application (Benmammar, 2015).

JADE est un Framework de développement de systèmes multi-agents, open-source et basé sur le langage Java. Ce dernier assure une communication transparente par l'échange de messages dans le langage normalisé FIPA-ACL (KDROUCI, 2013).

L'outil possède trois modules principaux (nécessaires aux normes FIPA) à savoir :

- ✓ DF (Director Facilitator) fournit un service de pages jaunes à la plate-forme.
- ✓ ACC (Agent Communication Chanel) gère la communication entre les agents.
- ✓ AMS (Agent Management System) supervise l'enregistrement des agents, leur authentification, leur accès et utilisation du système.

IV.2 JFreeChart

JFreeChart est une librairie open source sous Licence LGPL (Lesser General Public Licence) permettant de créer rapidement des graphiques pour les statistiques (Benmammar, 2015) .

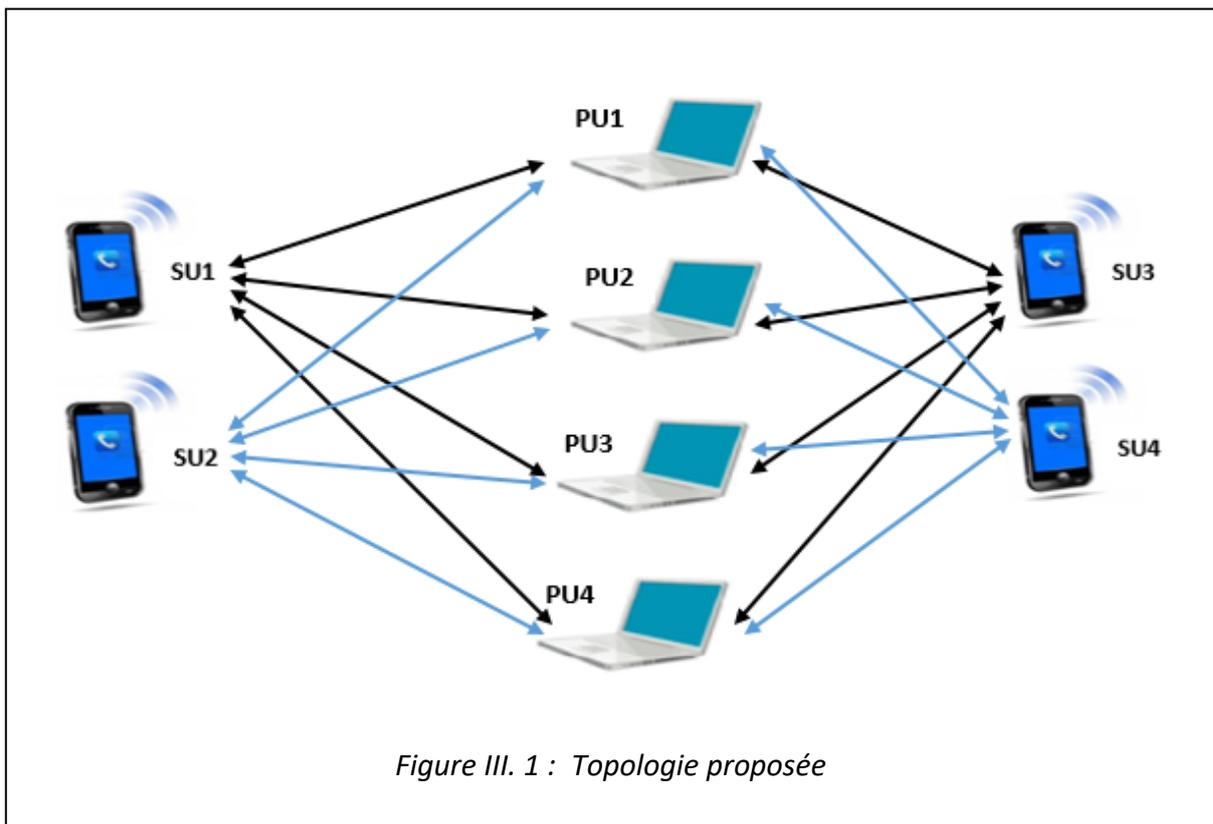
Il est possible de choisir le type de rendu des graphes:

- ✓ Composants Swing.
- ✓ Images (PNG ou JPEG).
- ✓ Fichiers d'images vectorielles PDF, EPS et SVG.

V. Topologie du réseau utilisé

Dans ce qui suit, nous allons utiliser un réseau sans infrastructure ad hoc car il est moins coûteux et s'adapte facilement à n'importe quel endroit. En plus, il est capable de s'organiser d'une manière autonome sans infrastructure fixe. Notre réseau est constitué d'un nombre variable d'entités sous forme de PUs et SUs qui communiquent d'une manière directe entre elles.

La figure suivante présente la topologie de réseau que nous utilisons.

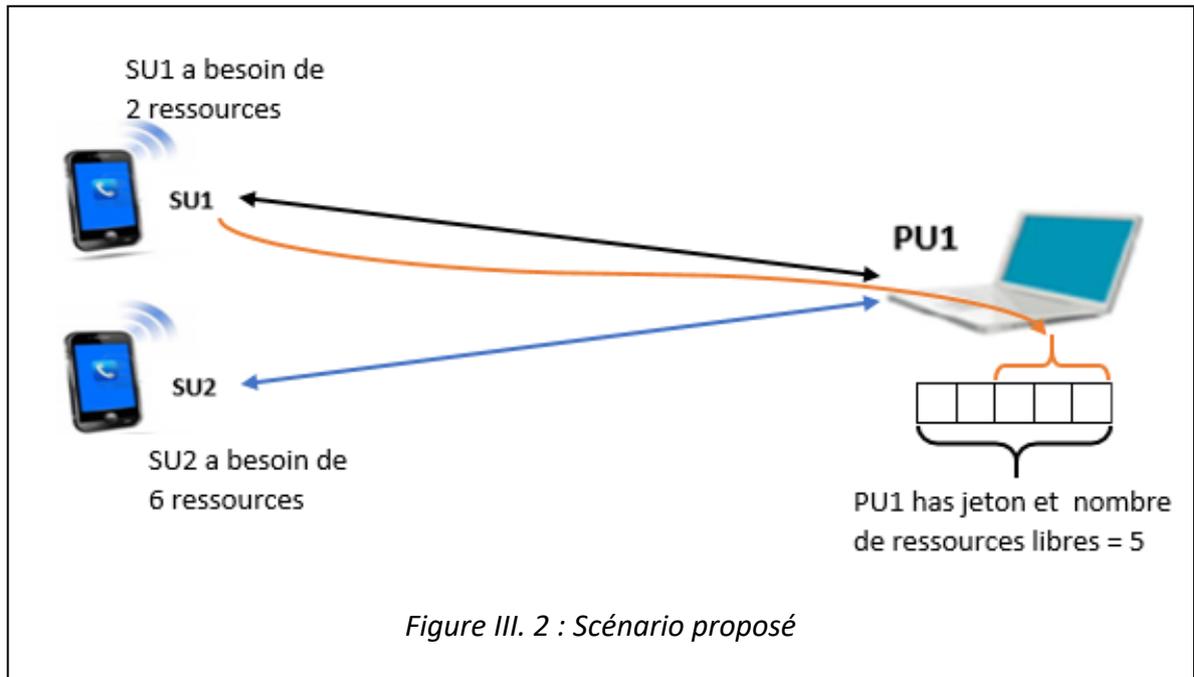


VI. Scénario

Dans ce chapitre, nous proposons une solution basée sur la négociation en SMA dans le cadre de la radio cognitive pour une gestion efficace du spectre. Cette solution est basée sur un type de négociation « multiples à multiples » c'est-à-dire qu'il y a plusieurs PU et plusieurs SU qui veulent communiquer. Sachant que chaque PU possède son propre jeton et pour que le SU communique avec le PU, il a besoin du jeton pour accéder à la SC.

Prenons l'exemple d'un SU qui va entamer une négociation en termes de nombre de canaux. Il va contacter chaque PU possédant le jeton afin de connaître le nombre de canaux disponibles. Un PU privilégie la satisfaction des demandes des SU selon leurs anciennetés, leurs durées et leurs timestamps. En fonction de ce qu'il reçoit de la part de tous les PU, le SU sera en mesure de choisir la meilleure offre pour lui.

La figure III.2 donne une idée sur un éventuel scénario.



VII. Organigramme

Dans la figure III.3, nous donnons un organigramme qui décrit le comportement des PU et des SU afin de présenter comment se déroulent les actions entre eux.

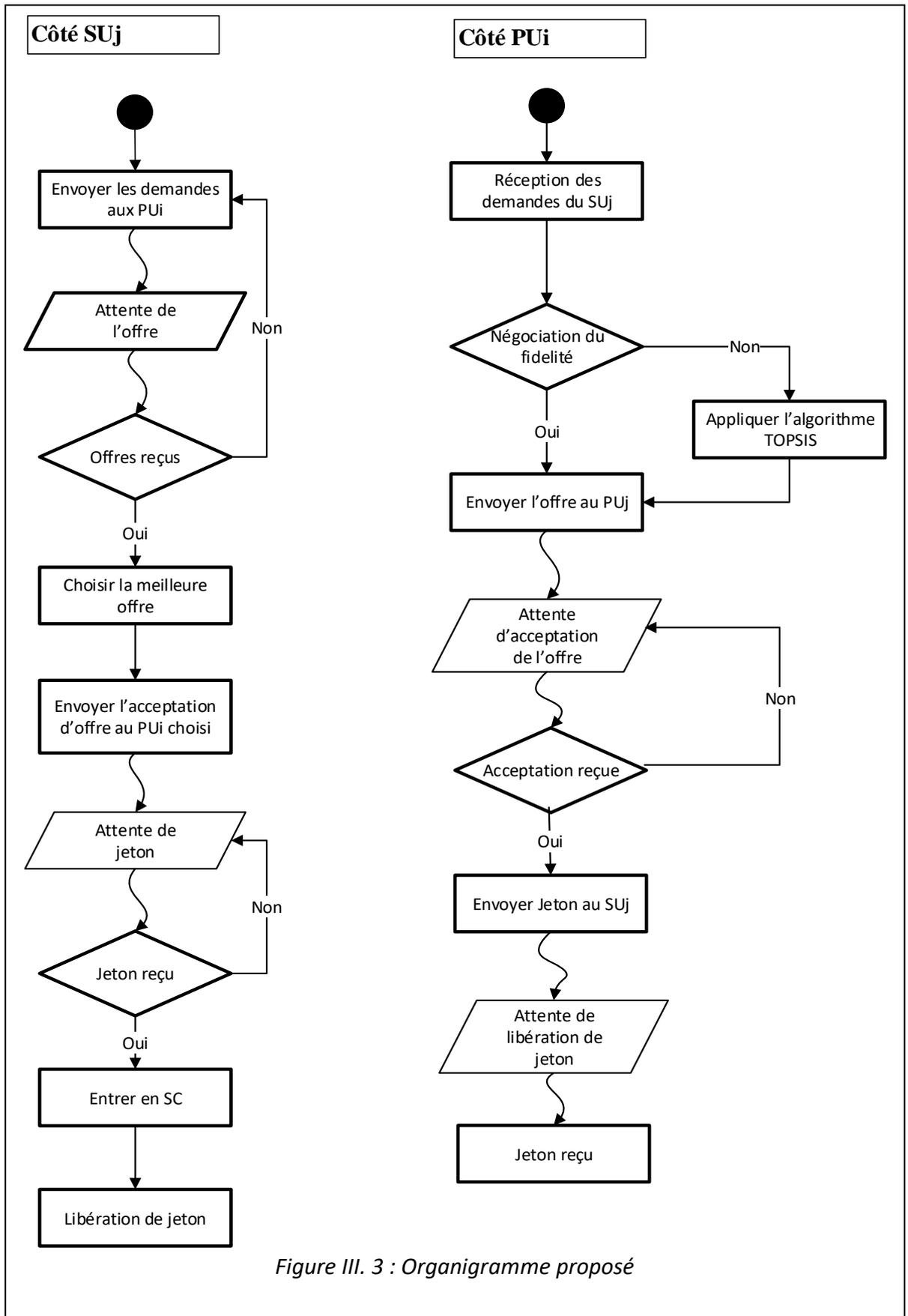


Figure III. 3 : Organigramme proposé

VIII. Algorithme proposé

Notre objectif est de proposer un algorithme d'exclusion mutuelle distribuée basé sur les jetons et tolérant aux pannes et permettant une mise en œuvre sur un réseau radio cognitif. Cet algorithme devra, entre autres, limiter tant que possible l'utilisation de la diffusion de messages.

Dans cette section nous présentons notre algorithme. Dans un premier temps nous présenterons la partie consacrée à la gestion du spectre. Puis, nous ferons de même pour la tolérance aux fautes.

VIII.1 Pour la gestion de spectre

Un agent peut communiquer avec un autre agent par passage de message. Un agent SU qui veut demander le jeton doit d'abord envoyer un message aux agents PU cible. Un message comporte un nombre d'arguments. L'agent récepteur détermine le plan d'action en vérifiant le type de message reçu et obtient ces paramètres.

Afin d'illustrer la solution proposée pour partager le spectre, nous présentons dans les figures III.4 et III.5 notre algorithme proposé.

```
for i=0 to PUi.length do
    if (PUi has jeton) then
        Envoyer un message REQUEST au PUi

if ( Message = PROPOSE) then
    SUj choisit la meilleure offre
    SUj envoie un message d'acceptation d'offre au PUi sélectionné

if (Message = INFORM) then
    SUj reçoit jeton
    T = Duree
    while ( T n'a pas expiré) do
        SUj fait son traitement
    if( T a expiré) then
        SUj libère jeton
```

Figure III. 4 : Une partie d'algorithme proposé pour la gestion de spectre côté SU

```
if (Message = REQUEST) do
  if (nbcandisp > nbcandem) do
    for i=0 to PUi.length do
      for j=0 to SUj.length do
        if (id(tabjeton) = id(req)) then
          Négociation_du_fidelité()
          if (la négociation du fidelité est réussie) then
            Envoyer l'offre au SUj sélectionné
          else Appliquer l'algorithme TOPSIS
              et envoyer l'offre au SUj sélectionné

if ( Message = ACCEPT_PROPOSAL) then
  Envoyer jeton SUj
  Jeton_present = faux
  fidelite = fidelite + 1

if ( Message = INFORM) then
  PUj reçoit jeton
  Jeton_present = vrai
```

Figure III. 5 : Une partie d'algorithme proposé pour la gestion de spectre coté PU

VIII.2 Pour la tolérance aux pannes

Les systèmes actuels sont soumis à une charge permanente et sont supposés être hautement disponibles. De plus, l'usage d'un nombre de plus en plus important de machines augmente la probabilité de production d'une panne. Dans ces circonstances, il est nécessaire de concevoir les systèmes de façon à ce qu'ils soient capables de survivre à l'occurrence d'une panne (Sicard, 2005).

Les figures III.6 et III.7 donnent les grandes lignes sur la partie d'algorithme qui traite la tolérance aux fautes. La panne apparaît si la durée de traitement « T » d'un SU a expiré et celui-ci ne libère pas le jeton. Pour faire ceci, nous avons testé côté SU, si ce SU a envoyé un message de libération « INFORM » au PU désiré, si ce n'est pas le cas alors on considère que ce SU est en panne, et coté PU, nous avons testé ce cas par la réception de message de libération, s'il a bien reçu ce message alors le SU fonctionne correctement sinon il est en panne.

```

if (Message = INFORM) then
  SUj reçoit jeton
  T = Duree
  while ( T n'a pas expiré) do
    SUj fait son traitement
  if( T a expiré) then
    if (SUj n'a pas libérer jeton) then
      SUj en panne

```

Figure III. 6 : Une partie d'algorithme proposé pour la tolérance aux fautes coté SU

```

if ( T a expiré) then
  if ( Message = INFORM) then
    PUi reçoit jeton
    Jeton_present = vrai
  else SUj en panne

```

Figure III. 7 : Une partie d'algorithme proposé pour la tolérance aux fautes coté PU

IX. Présentation du fonctionnement

Notre démarche consiste à créer un modèle typique de négociation: « de plusieurs à plusieurs».

Initialement, chaque agent PU possède son propre jeton. Lorsqu'un agent SU demande le jeton, sa requête est transmise aux PU possédant le jeton.

Notre proposition dans le cadre de ce travail est basée sur une négociation multicritères, car nous nous sommes centrés sur plusieurs critères de négociation, cette multitude de paramètres nous conduit à introduire une méthode adéquate qui est TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) [Annexe A], qui mène à une négociation optimisée et à un résultat efficace.

Chaque PU a un ensemble de ressources sous forme de canaux libres et les SU vont demander ces ressources à travers un message « REQUEST », ce dernier est envoyé à

tous les PU possédant le jeton. Ce message a comme paramètres le nombre de canaux demandés, la durée de traitement.

À chaque réception d'une requête, les PU qui ont reçu ce message, vont calculer la valeur de fidélité maximale de tous ses SU demandeurs. Après le calcul de la fidélité, le PU envoie un message « PROPOSE » au SU qui a la plus grande valeur de fidélité. Par contre, lors du calcul de fidélité, si un PU a trouvé deux SU ou plus qui ont la même valeur maximale de fidélité alors il applique l'algorithme TOPSIS pour choisir le SU le mieux adapté à ses besoins et puis envoie à ce dernier l'offre désignée.

Une fois que le SU reçoit l'offre, il envoie un message d'acceptation d'offre au PU désigné dans le cas où il a reçu un seul offre. Par contre dans le cas où le SU reçoit plusieurs offres alors il choisit le PU qui a le plus proche nombre de canaux disponibles à son nombre de canaux demandés, sinon si plusieurs PU ont le même nombre de canaux alors le SU envoie le message d'acceptation d'offre au premier PU qui a lui envoyé l'offre.

A la réception d'un message d'acceptation d'offre, le PU transmet le jeton au SU ayant émis la requête sous forme d'un message « INFORM » et met son état à false.

La réception du jeton permet au SU d'accéder aux ressources du PU pendant la durée demandée. Après l'achèvement de cette durée, le SU retransmet le jeton à son PU propriétaire sous forme d'un message « INFORM ». Et pour terminer, ce PU remet son état de possession de jeton à true dès la réception d'un message de ce type et il aura la capacité d'accepter d'autres messages « REQUEST ».

X. Présentation de l'application

La figure suivante montre l'interface d'accueil de notre application.



Lorsqu'on clique sur le bouton « Démarrer » l'ensemble des agents est démarré et l'interface suivante apparaît. Dans cette interface, nous utilisons deux champs de saisie de type JTextField, le premier pour saisir le nombre de canaux demandés par le SU et le deuxième pour saisir sa durée de traitement. Un bouton « Valider » de type JButton est aussi utilisé pour lancer les interactions entre les agents.



Figure III. 9 : Interface principale

XI. Simulation

Dans ce qui suit, nous allons présenter un scénario de simulation dans lequel on utilise 4 SU et 4 PU. Dans cette simulation PU1, PU2 et PU4 possèdent le jeton et ont respectivement 20,40,80 canaux. Dans la table jeton, chaque PU connaît la fidélité de tous les SU. La figure III.10 donne une idée sur notre tableau jeton.

PU1		PU2		PU3		PU4	
Nom SU	fidélité						
SU1	4	SU1	8	SU1	0	SU1	12
SU2	3	SU2	6	SU2	11	SU2	13
SU3	2	SU3	5	SU3	10	SU3	14
SU4	1	SU4	8	SU4	9	SU4	15

Figure III. 10 : Un exemple de tableau jeton

Au départ, les 4 SU envoient un message « REQUEST » aux 3 PU qui possèdent le jeton. Par contre, ils n'envoient pas ce message au PU3 car il ne possède pas le jeton. Ce message prend comme paramètre le nombre de canaux demandés ainsi que la durée de traitement.

Dans notre cas, le nombre de canaux demandés par SU1, SU2, SU3 et SU4 respectivement sont 13, 11, 12, 10. 23, 21, 22, 20 représentent la durée de traitement de chaque SU respectivement.

Les PU reçoivent en premier lieu les demandes des SU, ensuite ils voient s'ils peuvent satisfaire leur demandes. Dans cet exemple, après le calcul de la valeur maximale de fidélité, l'agent PU1 et PU4 envoient un message d'acceptation d'offre au SU1 et SU4 car ces derniers sont les plus fidèles. Par contre, l'agent PU2 a trouvé SU1 et SU4 qui ont la même valeur maximale, dans ce cas il fait appel à la méthode TOPSIS pour calculer la solution idéale des deux agents et envoyer le message au SU1 qui a la meilleure solution.

A la réception d'offre, SU4 envoie un message d'acceptation d'offre « ACCEPT_PROPOSAL » au PU4 car il a reçu une seule offre. Par contre, dans le cas de SU1 où il a reçu deux offres alors il choisit le PU1 qui a le plus proche nombre de canaux disponibles à son nombre de canaux demandés pour lui envoyer le message.

A la suite, PU1 et PU4 transmettent le jeton aux SU1 et SU4 respectivement sous forme d'un message « INFORM » et met son état à false et incrémente sa valeur de fidélité.

Finalement, après l'achèvement de la durée de traitement, SU1 et SU4 retransmettent le jeton aux PU1 et PU4 sous forme d'un message « INFORM ». Et pour terminer, ce PU1 et PU4 remet son état de possession de jeton à true.

La figure III.11 montre le résultat des différentes interactions possibles entre les agents lors du déroulement de l'exemple en utilisant l'agent sniffer.

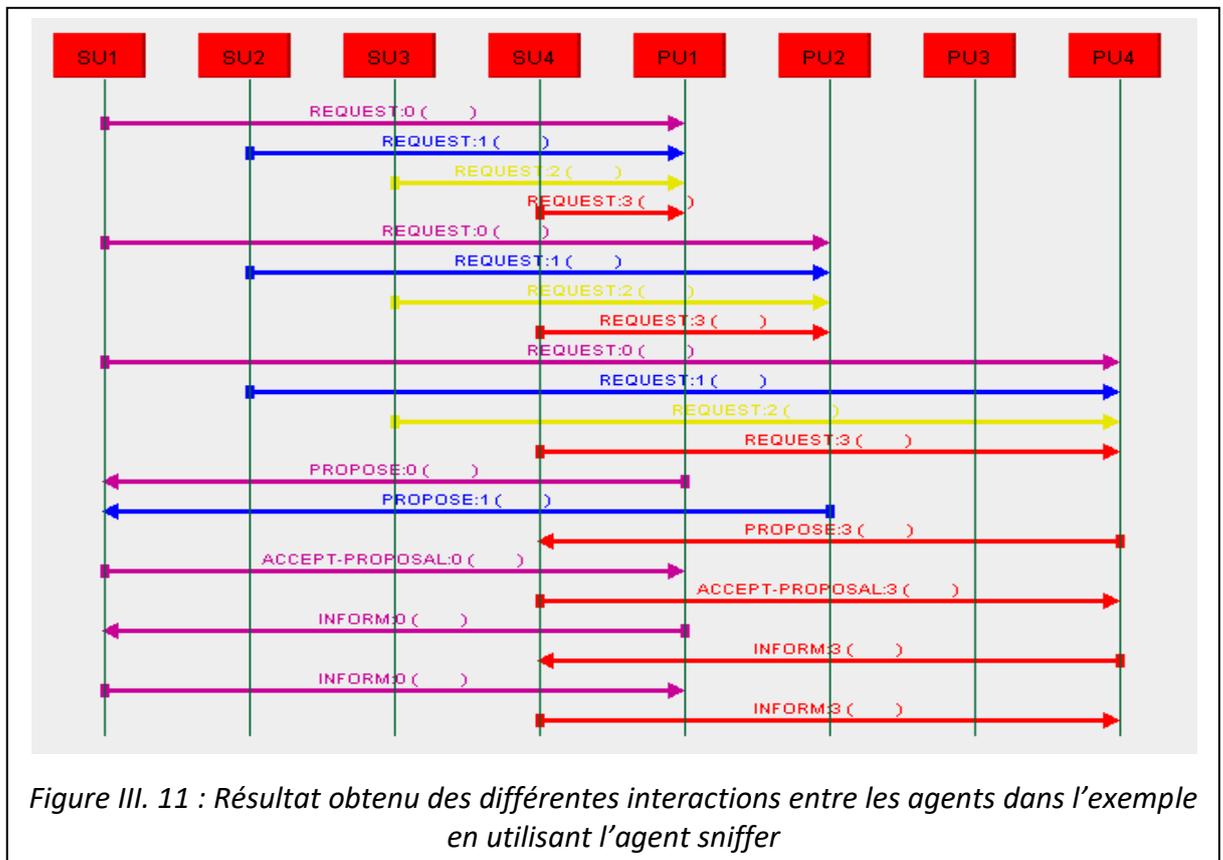


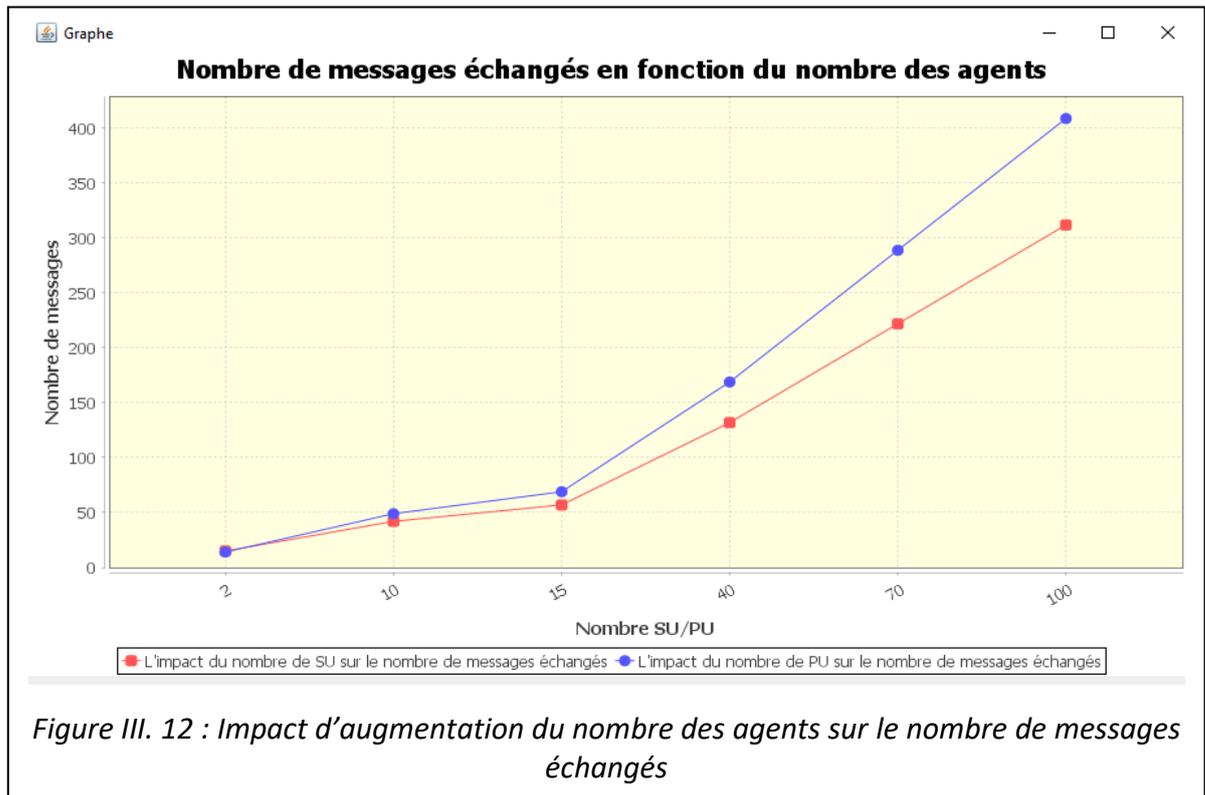
Figure III. 11 : Résultat obtenu des différentes interactions entre les agents dans l'exemple en utilisant l'agent sniffer

XII. Résultats obtenus

Dans ce qui suit, nous souhaitons mesurer l'impact d'augmentation du nombre de PU ainsi que celui du nombre de SU sur le nombre de messages échangés. Pour évaluer l'impact du nombre de PU, nous supposons que le nombre de SU est 3, tous les SU envoient un message « REQUEST » et que tous les PU possèdent le jeton. Par contre, pour arbitrer l'influence du nombre de SU, nous fixons le nombre de PU à 3 et modifions le nombre de SU.

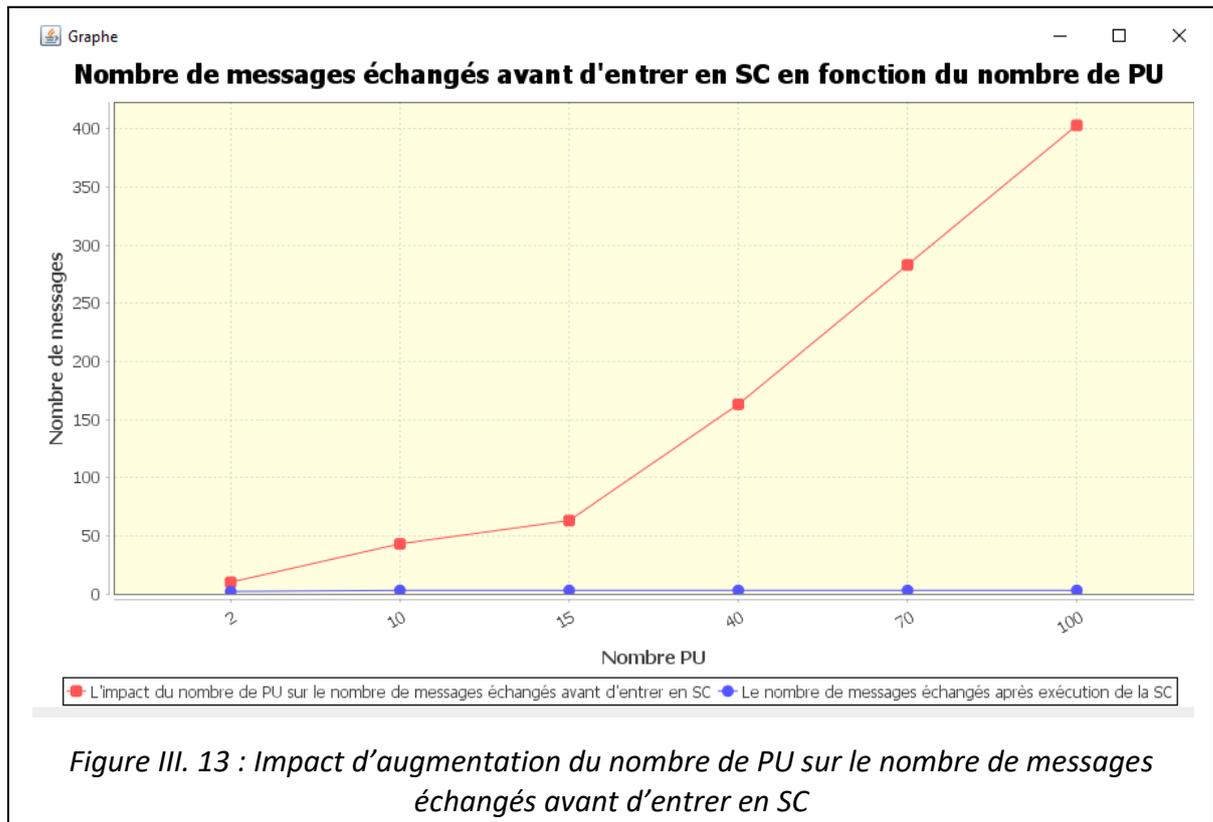
La figure III.12 montre que le nombre de messages échangés augmente que ce soit en augmentant le nombre de PU ou de SU avec lequel on lance la négociation. C'est logique car il y aura plus de messages à traiter dans le système. Mais ce que nous remarquons aussi à partir du graphe c'est que le nombre de messages échangés lors du passage à l'échelle augmente en augmentant le nombre de PU beaucoup plus que le

nombre de SU car lorsqu'on augmente le nombre des agents PU, le nombre de messages « REQUEST » et « PROPOSE » dépend de ce nombre et le reste des messages est constant à 3. Par contre, dans le cas où le nombre des agents SU est dynamique alors seulement le nombre de messages « REQUEST » qui dépend du nombre d'agents SU lancé et le reste des messages est constant est égal à 3 aussi.



Notre objectif maintenant est d'évaluer l'impact du nombre de PU sur le nombre de messages échangés avant d'entrer en SC. Pour cela, nous utilisons le même jeu de données précédent.

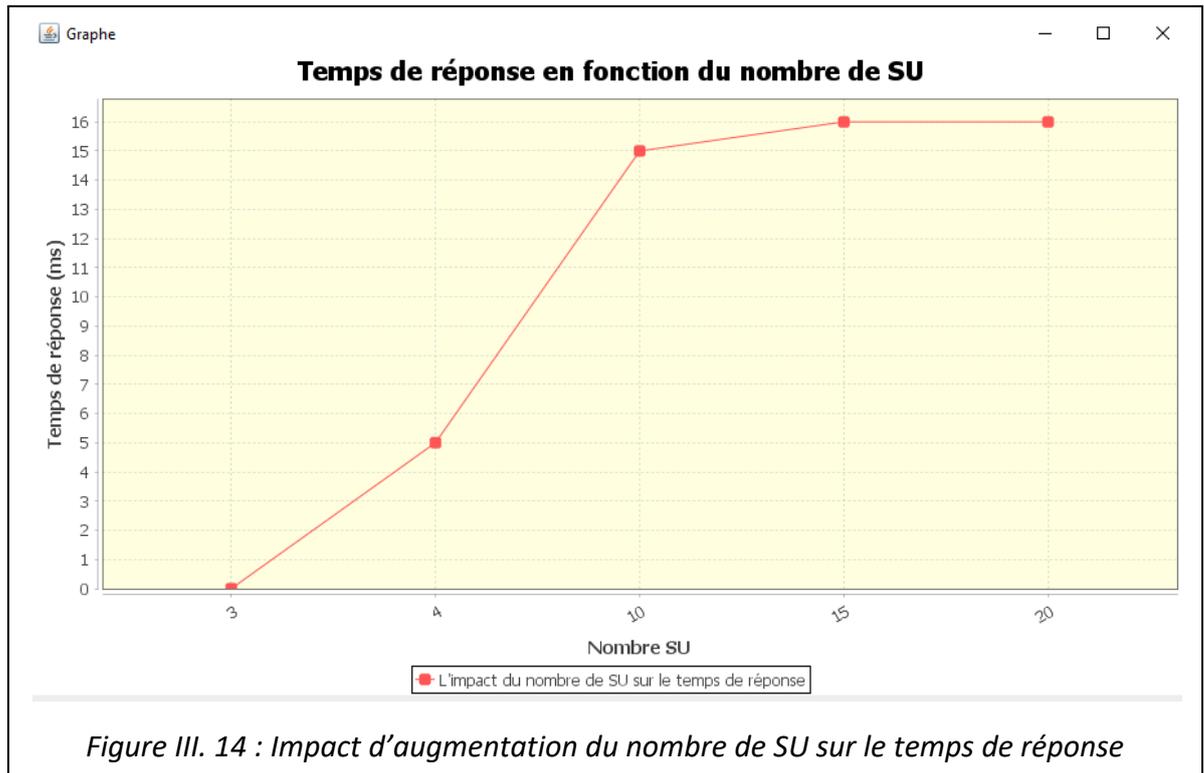
La figure III.13 présente l'impact d'augmentation du nombre de PU sur le nombre de messages échangés avant d'entrer en SC.



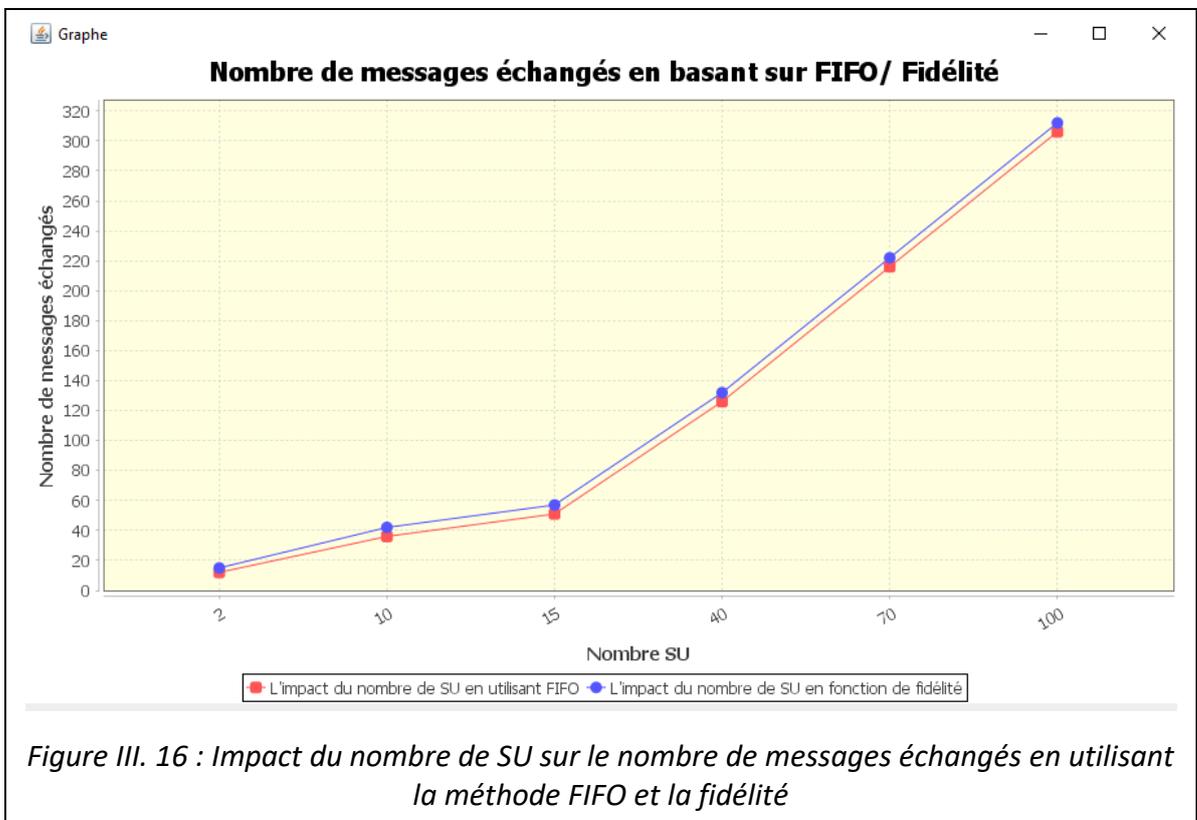
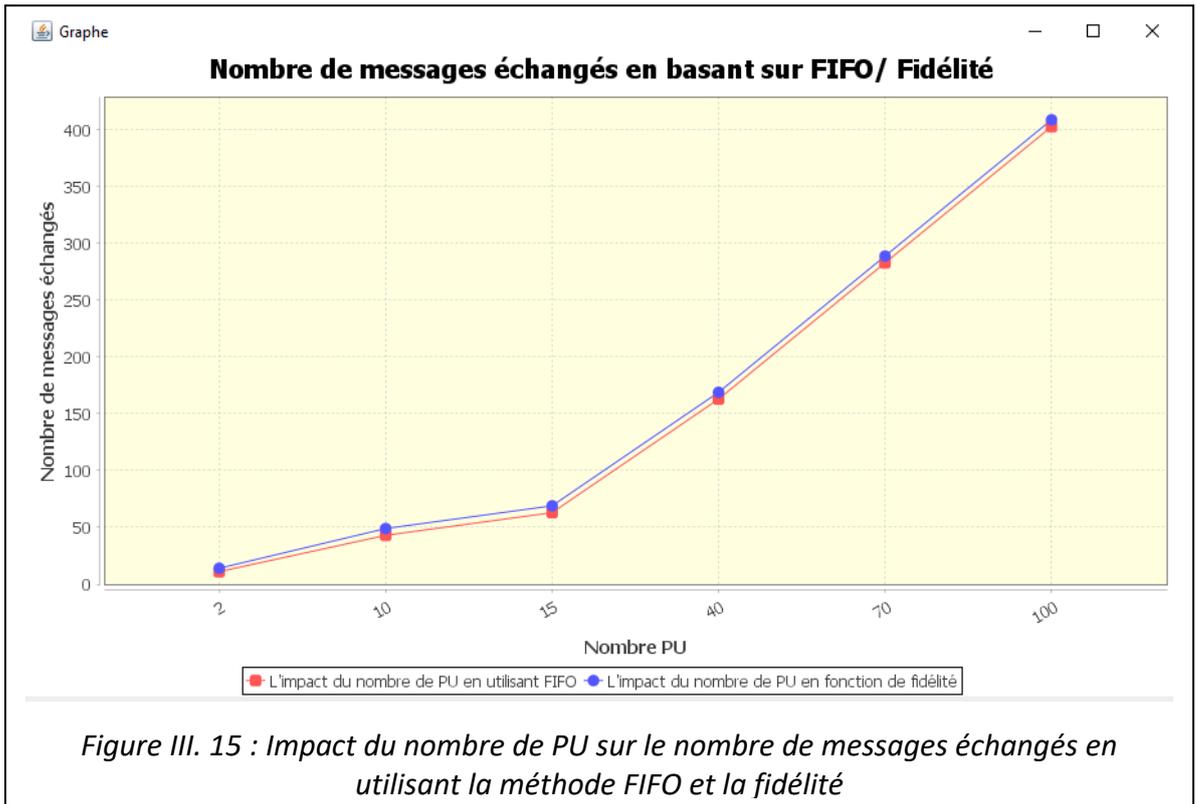
A partir des deux figures précédentes, nous remarquons que le nombre le plus élevé de messages échangés se passe avant d'entrer en SC.

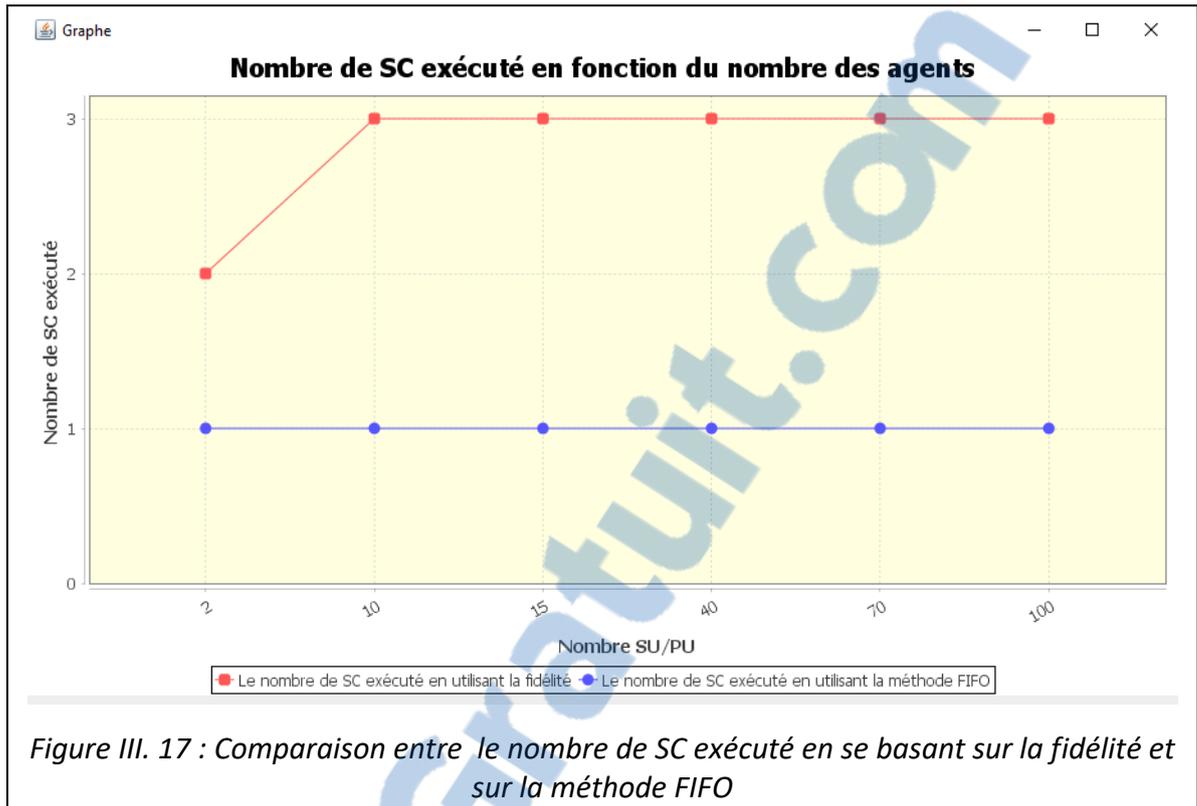
La nécessité de recevoir une réponse rapide à une demande, pousse le SU à considérer le temps de réponse comme étant un paramètre important. Pour étudier l'impact du nombre de SU sur le temps de réponse, nous avons considéré la topologie suivante : 3 PU avec un changement du nombre de SU de 3,4,10,15 et 20 respectivement.

La figure III.14 montre que le temps de réponse dépend du nombre de SU dans le réseau. C'est évident car il y aura plus de réponse à traiter. Par exemple, avec 10 SU, le temps de réponse est de 15 ms par contre avec 4 SU il est uniquement à 5 ms.



Dans un système de communication par messages, un problème important concerne l'accès à une ressource partagée. Cependant, diverses extensions de ce problème fondamental peuvent être trouvées dans la littérature. Parmi celles-ci, on trouve la méthode d'accès FIFO, cette méthode permet de servir les demandes d'accès suivant l'ordre dans lequel elles ont été demandées. Dans les figures III.15 et III.16, nous faisons une petite comparaison entre la méthode FIFO et notre algorithme en terme de nombre de messages échangés. Pour cela, nous reprenons le 1^{er} jeu de données utilisé dans cette section. A partir de ces figures, nous remarquons qu'en utilisant la méthode FIFO le nombre de message échangés que ce soit en modifiant le nombre de SU ou le nombre de PU est inférieur à ce on se basant sur la fidélité car on utilisant la méthode FIFO, il y aura qu'un seul SU qui exécute la SC ce qui réduit le nombre de messages « ACCEPT_PROPOSAL » et « INFORM ». Par contre, on se basant sur la fidélité, on peut réduire le nombre de messages envoyé dès le départ puisque le message « REQUEST » n'est envoyé qu'aux PU possédant le jeton et en plus on peut satisfaire le plus grand nombre possible de demandes c.à.d. le nombre de SC exécuté est plus élevé que lorsqu'on utilise la méthode FIFO, ce qui est présenté dans la figure III.17.





XIII. Évaluation des performances

Dans cette section, nous analysons les performances de notre algorithme. Pour cela, nous vérifions que notre algorithme assure la propriété de sûreté. Cette propriété ne se résume qu'à l'unicité du jeton et la vivacité est respectée car lorsque les PU reçoivent l'ensemble de message « REQUEST » envoyés par les SU, alors chaque PU envoie le jeton à un et un seul SU choisit, soit qu'il soit le plus fidèle ou choisit par la méthode TOPSIS. Ce qui démontre qu'à chaque fois le jeton est détenu par un seul SU et en un temps fini. Pour cela, on peut conclure que notre algorithme vérifie bien la propriété de la sûreté.

De plus, notre algorithme assure l'entrée concurrente (efficacité). Ceci apparaît dans le fait que plusieurs SU peuvent exécuter la SC concurrentement.

- L'algorithme présente une complexité en messages en $O(N)$ dans le pire des cas.

XIV. Conclusion

Dans ce chapitre, nous avons présenté au début les supports nécessaires au fonctionnement de notre application, ensuite nous avons abordé les différentes étapes du développement de l'application sous l'IDE Netbeans en utilisant l'outil JADE pour mettre en œuvre notre système multi-agents.

Pour cela, nous avons proposé un algorithme d'exclusion mutuelle distribuée basé sur la satisfaction des demandes des SU non seulement sur leurs anciennetés mais aussi sur la technique de décisions multicritère TOPSIS.

Les résultats obtenus étaient intéressants car ils nous ont permis de vérifier l'efficacité de notre algorithme en terme du nombre de messages échangés ainsi que le temps de réponse pour l'utilisateur de la RC.

Conclusion générale

Pour partager le spectre d'une manière opportuniste dans les réseaux de radio cognitive et ainsi satisfaire au mieux les besoins des SU, il est nécessaire d'utiliser des algorithmes pour l'accès dynamique au spectre et assurer une communication de bonne qualité aux SU .

Nous pensons que l'utilisation des algorithmes d'exclusion mutuelle distribuée serait une solution efficace pour le partage des ressources spectrales. Ces algorithmes fonctionnent selon les principes du paradigme de transmissions cognitives des utilisateurs primaires et des utilisateurs secondaires transmettent simultanément sur la même bande spectrale.

Dans un premier temps, nous avons effectué une étude sur l'importance de la RC ainsi que son principe de fonctionnement. Après cela, nous avons proposé aussi un état de l'art sur les différents algorithmes d'exclusion mutuelle distribuée où nous décrivons l'ensemble des solutions proposées dans la littérature, pour explorer certains aspects du problème de l'encombrement du spectre pour le résoudre ensuite en introduisant les systèmes multi agents.

L'utilisation des SMA combinée avec les algorithmes d'exclusion mutuelle distribuée pour l'attribution de ressources et le partage de spectre a montré son intérêt dans la cadre de la RC. Ce système a pour but de vérifier l'efficacité de notre approche en termes de temps, de qualité des solutions et de respect des contraintes.

Finalement, notre solution proposée dans le cadre de ce mémoire est validée à travers la simulation avec JADE, nous avons montré qu'avec la négociation, plusieurs SU peuvent trouver les offres qui s'adaptent avec leurs besoins tout en réduisant le nombre de message dans le réseau et éviter l'encombrement.

Comme perspective à ce travail, on peut l'étendre dans le cadre de mobilité où les SU vont rencontrer un nombre assez important de PU. Dans ce cas, les SU doivent négocier pendant leur mobilité avec tous les PU rencontrés dans sa zone de mobilité. En plus, nous prévoyons de faire une comparaison avec d'autres approches de la littérature telles que l'algorithme RL(Reverse Link) qui suppose également qu'il existe initialement un jeton unique. Enfin, nous envisageons d'optimiser notre algorithme en se basant sur d'autres paramètres dans le choix de la meilleure offre tels que la distance et le prix.

BIBLIOGRAPHIE

Bibliographie

(Amraoui, 2012) : Amraoui A., [et al.], « Négociations à base d'Enchères dans les Réseaux Radio Cognitive », présenté à NOTERE (Nouvelles Technologies de la Répartition)/CFIP (Colloque francophone sur l'ingénierie des protocoles), 2012.

(Baldoni, 2002) : Baldoni R., Virgillito A., Petrassi R., « A distributed mutual exclusion algorithm for mobile ad-hoc networks », in: Computers and Communications, Proceedings, ISCC 2002, Seventh International Symposium on. IEEE, pp. 539–544, 2002.

(Bedrane, 2014) : Bedrane A., « Vers l'auto-gestion d'un réseau de radio cognitive », Mémoire de fin d'études : Université Abou Bakr Belkaid - Tlemcen, 2014.

(BEN DHAOU, 2011) : BEN DHAOU A., « Allocation dynamique des bandes spectrales dans les réseaux sans-fil à radio cognitive », Mémoire de fin d'études: UNIVERSITÉ DU QUÉBEC À MONTRÉAL, 2011.

(Benmammar, 2014) : Benmammar B. et Amraoui A., « Réseaux de radio cognitive : Allocation des ressources radio et accès dynamique au spectre », 2014.

(Benmammar, 2015) : Benmammar B., « Allocation de ressources dans un réseau de radio cognitive en utilisant JADE », Rapport de recherche, 2015.

(BENOUINI, 2014) : BENOUINI R., « Radio cognitive & Intelligence Artificielle», 2014.

(Conan, 2016) : Conan D., « Initiation à l'algorithmique répartie », Télécom SudParis - France, CSC4509,2016.

(Dijkstra, 1965) : Dijkstra E.W., « Solution of a problem in concurrent programming control », Communications of the ACM, 8(9) :569, 1965.

(DRID , 2007) : DRID H., KHEDDOUCI H., « Algorithme d'Exclusion Mutuelle Distribué pour les Réseaux Ad Hoc », in: The 4th International Conference on Sciences of Electronic, Technologies of Information and Telecommunications SETIT, 2007.

(Jiang, 2002) : Jiang J.R., « A group mutual exclusion algorithm for ad hoc mobile networks », in: Proc. 6th International Conference on Computer Science and Information, pp. 266-272, 2002.

(KDROUCI, 2013) : KDROUCI K., « Négociation de l'accès au spectre dans les réseaux de radio cognitive », Mémoire de fin d'études : Université Abou Bakr Belkaid - Tlemcen, 2013.

(Khattab, 2012) : Khattab A., Perkins D. et Bayoumi M., « Cognitive Radio Networks: From Theory to Practice», Springer Science & Business Media, XXII, p. 110, Hardcover, 2012.

(Kshemkalyani , 2011) : Kshemkalyani A.D., Singhal M., « Distributed computing: principles, algorithms, and systems », Cambridge University Press, 2011.

(Larbi , 2014) : Larbi I., Benmamar B., « Négociation de spectre dans les réseaux de radio cognitive », Rapport de recherche, 2014.

(Lejeune , 2013) : Lejeune J., [et al.], « Un algorithme équitale d'exclusion mutuelle distribuée avec priorité, in: 9ème Conférence Française Sur Les Systèmes d'Exploitation (CFSE'13), Chapitre Français de l'ACM-SIGOPS, GDR ARP.

(Lejeune, 2014) : Lejeune J., « Algorithmique distribuée d'exclusion mutuelle: vers une gestion efficace des ressources », Thèse de doctorat: Université Pierre et Marie Curie - Paris, 2014.

(Mitola, 2000) : Mitola J., « Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio», Ph.D. thesis, KTH Royal Institute of Technology, 2000.

(Mitola, 2006) : Mitola J., « Cognitive Radio Architecture : The Engineering Foundations of Radio XML », Wiley 2006, Hardcover, p. 473.

(Myoupo, 2009) : Myoupo J.F., Naimi M., Thiare O., « A clustering group mutual exclusion algorithm for mobile ad hoc networks », in: Computers and Communications, ISCC 2009. IEEE Symposium on. IEEE, pp. 693–696, 2009.

(Nguyen, 2012) : Nguyen T.D., « Conception d'antenne intelligente reconfigurable pour la radio cognitive », Université Grenoble Alpes - France, 2012.

(*Ortiz, 2009*) : *Ortiz A.*, « Contrôle de la concurrence dans les grilles informatiques. Application au projet ViSaGe ». Thèse de doctorat: Université de Toulouse, Université Toulouse III-Paul Sabatier, 2009.

(*Sicard, 2005*) : *Sicard S.*, « Tolérance aux fautes pour des systèmes autonomes », Rapport de recherche, 2005.

(*Sopena, 2008*) : *Sopena J.*, « Algorithmes d'exclusion mutuelle: tolérance aux fautes et adaptation aux grilles » , Thèse de doctorat: Université Pierre et Marie Curie - Paris,2008.

(*Thiare, 2007a*) : *Thiare O., Naimi M., Gueroui M.*, « A group mutual exclusion algorithm for mobile ad hoc networks », in Proc. Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, pp. 373-377, 2007.

(*Thiare, 2007b*) : *Thiare O.*, « Exclusion mutuelle de groupes dans les systèmes distribués », Thèse de doctorat: Université de Cergy Pontoise - France, 2007.

(*Thiare, 2009*) : *Thiare O., Naimi M.*, « A group k-mutual exclusion algorithm for mobile ad hoc networks », in: International Work-Conference on Artificial Neural Networks, Springer, pp. 58–66,2009.

(*TRIGUI, 2013*) : *TRIGUI E.*, « Gestion multi-agents du spectre pour des terminaux mobiles à radio cognitive », Thèse de doctorat : Université de technologie de Troyes - France, 2013.

(*Velazques, 1993*) : *Velazques M.G.*, « A survey of distributed mutual exclusion algorithms », Colorado State University – USA, 1993.

(*Walter, 1998*) : *Walter J., Welch J., Vaidya N.H.*, « A mutual exclusion algorithm for ad hoc mobile networks », Dial M for Mobility Workshop, Dallas TX, 1998.

(*Walter, 2001*) : *Walter J., Cao G., Mohanty M.*, « A k-mutual exclusion algorithm for wireless ad hoc networks », in Proceedings of the first annual Workshop on Principles of Mobile Computing (POMC 2001), Newport, Rhode Island, USA, 2001.

(*Zayen, 2010*) : *Zayen B.*, « Stratégies d'accès et d'allocation des ressources pour la radio cognitive », Réseaux et télécommunications [cs.NI], Télécom ParisTech, 2010.

Annexe A

Exemple de l'algorithme TOPSIS

Le client désire acheter une voiture et ne sait pas laquelle choisir parmi les marques suivantes (Civic, Saturn, Ford, Mazda). L'algorithme TOPSIS peut aider le client à faire le meilleur choix en se basant sur les critères suivants:

- Style ;
- Fiabilité ;
- Carburant Eco ;
- Coût.

Poids $w_j = \{0.1, 0.4, 0.3, 0.2\}$ respectivement pour les critères cités auparavant.

Matrice de décision :

	Style	Fiabilité	Carburant Eco.	Coût
Civic	7	9	9	8
Saturn	8	7	8	7
Ford	9	6	8	9
Mazda	6	7	8	6

Étape 1: Calcul des préférences normalisées

Pour la construction d'une matrice normalisée, il faut multiplier chaque colonne par $(\sum x_{ij}^2)^{1/2}$ pour avoir r_{ij} .

	Style	Fiabilité	Carburant Eco.	Coût
Civic	0.46	0.61	0.54	0.53
Saturn	0.53	0.48	0.48	0.46
Ford	0.59	0.41	0.48	0.59
Mazda	0.40	0.48	0.48	0.40

Étape 2: Calcul des préférences normalisées avec des poids associés aux critères

Pour la construction d'une matrice normalisée et pondérée, il faut multiplier chaque colonne par w_j pour obtenir V_{ij} .

	Style	Fiabilité	Carburant Eco.	Coût
Civic	0.046	0.244	0.162	0.106
Saturn	0.053	0.192	0.144	0.092
Ford	0.059	0.164	0.144	0.118
Mazda	0.040	0.192	0.144	0.080

Étape 3: Identification des solutions idéales et anti-idéales

Déterminer la solution idéale

$$A^+ = \{0.059, 0.244, 0.162, 0.080\}$$

Déterminer la solution anti-idéale

$$A^- = \{0.040, 0.164, 0.144, 0.118\}$$

Étape 4: Calcul des distances de séparation

$$S_i^* = [\sum (v_j^* - v_{ij})^2]^{1/2}$$

	Style	Fiabilité	Carburant Eco.	Coût
Civic	$(0.046 - 0.059)^2$	$(0.244 - 0.244)^2$	$(0.162 - 0.162)^2$	$(0.106 - 0.080)^2$
Saturn	$(0.053 - 0.059)^2$	$(0.192 - 0.244)^2$	$(0.144 - 0.162)^2$	$(0.092 - 0.080)^2$
Ford	$(0.059 - 0.059)^2$	$(0.164 - 0.244)^2$	$(0.144 - 0.162)^2$	$(0.118 - 0.080)^2$
Mazda	$(0.040 - 0.059)^2$	$(0.192 - 0.244)^2$	$(0.144 - 0.162)^2$	$(0.080 - 0.080)^2$

Distance idéale

Civic	0.000845	0.029
Saturn	0.003208	0.057
Ford	0.008186	0.090
Mazda	0.003389	0.058

Distance anti-idéale

Civic	0.006904	0.083
Saturn	0.001629	0.040
Ford	0.000361	0.019
Mazda	0.002228	0.047

Étape 5: Calcul de l'index de similarité à la solution idéale

$$C_i^* = S_i' / (S_i^* + S_i')$$

Civic	0.083/0.112	0.74	→ Meilleure solution
Saturn	0.040/0.097	0.41	
Ford	0.019/0.109	0.17	
Mazda	0.047/0.105	0.45	

Résumé

La radio cognitive est un concept nouveau pour les communications sans fil futures, elle a suscité un intérêt considérable parmi les universités, l'industrie et les organismes de réglementation. Ce concept est une zone technologique immature mais en pleine expansion. En termes de régulation du spectre, l'avantage principal visé par cette fonctionnalité est de permettre d'utiliser le spectre disponible de la manière la plus efficace, pour assurer une utilisation optimale. L'objectif de ce PFE est d'utiliser les algorithmes d'exclusion mutuelle distribuée pour résoudre le problème de l'encombrement de ressources spectrales dans un réseau de radio cognitive. Pour cela, nous avons choisi d'implémenter un algorithme basé sur la circulation de jeton et notre but a été par la suite de faire une validation à travers l'analyse des résultats obtenus à l'aide d'une implémentation du modèle en utilisant la plateforme multi-agents JADE.

Mots-clés Radio cognitive, systèmes multi-agents, exclusion mutuelle, JADE.

Abstract

Cognitive radio is a new concept for future wireless communications, and it has generated considerable interest among universities, industries and regulators. This concept is an immature but rapidly growing technology area. In terms of spectrum regulation, the main advantage of this feature is to allow the use of the available spectrum in the most efficient way, to ensure optimum use. The objective of this work is to use distributed mutual exclusion algorithms to solve the problem of spectral resources congestion in a cognitive radio network. We chose to implement a token-based algorithm and our goal was subsequently to validate our proposition through the analysis of the obtained results using a model implementation using the JADE multi-agents platform.

Key words: Cognitive radio, multi-agent systems, mutual exclusion, JADE

تلخيص

الراديو الإدراكية هو مفهوم جديد للاتصالات اللاسلكية في المستقبل، و التي ولدت اهتماما كبيرا بين الجامعات والصناعة والهيئات التنظيمية. هذا المفهوم هو منطقة تكنولوجية لا تزال غير ناضجة ولكنها متنامية. ومن حيث تنظيم الطيف، تتمثل الميزة الرئيسية لهذه الوظيفة في استخدام الطيف المتاح بالطريقة الأكثر فعالية لضمان الاستخدام الأمثل. والغرض من هذه المذكرة هو استخدام خوارزمية توزيع الإقصاء المتبادل على أساس مفهوم رمزي لحل مشكلة ازدحام الموارد الطيفية في شبكة الراديو الإدراكية. لهذا، اخترنا لتنفيذ خوارزمية القائمة على الرمز المميز، وكان هدفنا في وقت لاحق للمصادقة من خلال تحليل النتائج التي تم الحصول عليها باستخدام نموذج التنفيذ باستخدام نظام المحاكاة المتعددة JADE.

الكلمات المفتاحية الراديو الإدراكية، الأنظمة متعددة الوكلاء، الإقصاء المتبادل، جاد.