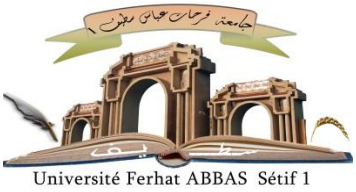
الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITÉ FERHAT ABBAS - SETIF1

FACULTÉ DE TECHNOLOGIE

THÈSE

Présentée au Département d'électrotechnique

Pour l'obtention du diplôme de

DOCTORAT LMD

Domaine : Sciences et Technologie

Filière: Automatique **Option: Automatique**

Par

KACEM Ibrahim

THÈME

Diagnostic des systèmes à évènements discrets complexes par réseaux de Petri

Soutenue le 17/11/2020 devant le Jury:

Mr KHEMLICHE Mabrouk	**Professeur**	**Univ. Ferhat Abbas Sétif 1**	**Président**
Mr SAIT Belkacem	**Professeur**	**Univ. Ferhat Abbas Sétif 1**	**Directeur de thèse**
Mr ABDELAZIZ Mourad	**Professeur**	**Univ. Ferhat Abbas Sétif 1**	**Examinateur**
Mr BOUDEN Toufik	**Professeur**	**Uni. Mohamed Seddik Ben Yahia Jijel**	**Examinateur**
Mr ACHOUR Abd Yazid	**MCA**	**Univ. Abderrahmane Mira Bejaia**	**Examinateur**

A mes très chers Parents Salima et Rachid

A ma Femme chérie Soumia A mes enfants Mounib et Dhouha

A mes très chers Frères et Sœurs Sara et sa famille Nacer et Rassil, à Hasna, Moncef, et Youssra.

A toute ma Famille et Belle Famille

A tous mes Amis

Remerciements

Cette thèse de doctorat était une aventure unique et passionnante. Une expérience enrichissante qui m'a beaucoup apporté sur tous les plans. Elle n'aurait cependant pas été vécue ainsi sans les nombreuses personnes que j'ai eu l'honneur et le plaisir de côtoyer et qui m'ont aidé, de près ou de loin, à atteindre mes objectifs.

Ainsi, je tiens à exprimer ma forte reconnaissance à mes directeurs de thèse : Pr.Dr. Sait Belkacem qui m'accompagné durant toute cette période par ces conseils et orientations. Je tiens aussi à remercier Pr. ABDELAZIZ Mourad, Pr. BOUDEN Toufik, et Dr ACHOUR Abd Yazid pour avoir accepté d'être les examinateurs de ma thèse et pour le temps qu'ils ont consacré, malgré leurs charges et leurs responsabilités, à l'évaluation de mes travaux avec soin. Je remercie également Pr. KHEMLICHE Mabrouk pour avoir acceptée de présider ma soutenance de thèse.

Mes sincères remerciements vont aussi à Pr. Saad Mekhilef et au Dr Sabeur Nacer Eddine qui m'ont accueilli et poussé au monde de la recherche scientifique durant mon stage au laboratoire PEARL de l'université de Malaya. Merci pour ton amitié, tes conseils et ton soutien permanent.

Je ne peux terminer sans adresser mes plus profonds et sincères remerciements aux êtres qui me sont le plus chers. Je pense bien évidement à mes très chers parents à qui je dois tout et à mes frères et sœurs et leurs familles qui m'ont toujours soutenu. Je leur dédie ce mémoire de thèse de doctorat et leur exprime tout mon amour. Je n'aurais jamais pu aller aussi loin sans vous. Je vous remercie infiniment et j'espère vous rendre fiers. A ma tendre femme qui m'a toujours soutenu.

Sommaire

Liste des Figures

Liste des abréviations

QoS : Qualité de service.

DSR: Dynamic Source Routing.

AODV: Ad-hoc On-demand Distance Vector.

OLSR: Optimized Link State Routing Protocol.

SED : Système à evenement discret.

RdP : Réseau de Petri.

RREQ : Paquet Route Request.

CRA : Algorithme de raisonnement simultané.

RdPDF : Réseau de Petri dynamique Floue.

Dot : degré de vérité.

RdPSyncF : Réseaux de Petri Synchronisé Floue.

Chapitre 1

Introduction générale

Résumé

Dans ce chapitre, nous présentons d'abord un aperçu sur le sujet du diagnostic et surveillance dans les systèmes à évènements discrets, où nous choisirons les systèmes de communication sans fil Ad-hoc comme exemple. Nous allons donner une explication simple de ces systèmes. Nous concentrons sur les deux axes principaux la modélisation des protocoles de routage et les méthodes de surveillance des réseaux Ad-hoc avec les réseaux de Pétri, et nous motivons ces études. Deuxièmement, nous décrivons l'organisation de la thèse et les contenus de chaque chapitre. Enfin, nous discutons de la contribution de cette thèse.

1.1. Introduction

Durant les phases de conception et de développement des systèmes, un certain nombre de vérifications et de tests de plus en plus performants sont entrepris afin de garantir au mieux leur sûreté de fonctionnement. Cependant, et malgré toutes les mesures qui peuvent être prises pour effectuer ces tests, il subsistera toujours le risque d'un comportement imprévu qui peut remettre en cause l'intégrité de ces systèmes. Ceci est dû à leur complexité croissante mais aussi à leur environnement de fonctionnement ou tout simplement aux phénomènes de vieillissement et d'obsolescence. L'apparition de ces comportements peut engendrer des conséquences plus ou moins graves, allant de l'indisponibilité partielle du système (et ainsi la baisse de la rentabilité) jusqu'aux catastrophes humaines, matérielles et écologiques. Il apparaît donc de plus en plus nécessaire de disposer de méthodes et d'outils particulièrement efficaces pour la surveillance de l'état de santé du système. Ces outils sont déployés durant son exploitation et relèvent du domaine général de la sûreté de fonctionnement. Notre travail s'articule autour de la surveillance et concerne la catégorie des systèmes dits à mission prolongée (systèmes de production, de communication…). Le but de la surveillance de l'état de santé de ce type de systèmes est de garantir leur sécurité, leur disponibilité ainsi que le maintien de leur qualité dans le temps.

Au cours de cette thèse, nous nous s'intéresserons à la catégorie des Systèmes à Événements Discrets (SED). Un SED est un système dont l'espace d'état est discret. Les transitions entre les états se produisent suite aux occurrences d'événements asynchrones [1, 2]. Les systèmes étant souvent de nature continus, leur modélisation avec des SED nécessite un certain niveau d'abstraction. On peut par exemple caractériser par un événement le dépassement d'un seuil de remplissage d'un contenant même si le niveau de remplissage est une variable continue. D'autres systèmes relèvent directement de cette classe de comportement. C'est pourquoi, de nombreuses méthodologies ont été développées pour la modélisation, la conception, l'analyse de performance ou encore de la surveillance de ce type de systèmes. Les SED peuvent être classés selon le type d'application considérée [2, 3]. Nous allons, dans ce qui suit, donner quelques-unes de ces classes avec des exemples de problématiques qui relèvent du contexte de notre étude. Dans notre thèse, nous concentrons et expliquerons l'un des systèmes à évènement discret les plus répandus actuellement est celui des systèmes de communication sans fil. La communication par les réseaux sans fil et mobiles est aujourd'hui au cœur des rapports entre

les hommes quel que soit leurs types de machines. Pour répondre à cette demande, tout en tenant compte des limites des applications informatiques, chercheures et industriels se sont concentrés sur de nouveaux standards offrant chacun un confort de communication différent. Ainsi sont apparus des concepts nouveaux créés dans le but d'accroître la mobilité des utilisateurs. Mais ce gain de mobilité ne va pas sans entacher d'autres avantages comme la rapidité d'une communication ou sa qualité de service. En effet, comparé aux interfaces filaires, peu nombreuse sont les interfaces sans fil qui offrent un débit rapide. Nous pouvons citer, les ondes hertziennes, l'infrarouge, Bluetooth, etc. Ces interfaces sont partagées et ne permettent pas de réserver des ressources de manière exclusive à une application. De même, l'interférence et le taux d'erreur importants réduisent la capacité de ces interfaces.

L'un des outils utilisés dans la modélisation et le diagnostic des système a évènement discret est les réseaux de Petri (RdP), qui ont été introduits au début des années 1960 par Carl Adam Petri dans sa thèse de doctorat [4]. Au fil des années, ils ont été étendus dans de nombreuses directions, notamment le temps, les données et la hiérarchie. Les RdP sont particulièrement utiles pour modéliser le comportement simultané, distribué et asynchrone dans un système et offrent un bon compromis entre puissance de modélisation et capacité de traitement analytique. Aujourd'hui, ils sont considérés comme l'un des principaux formalismes pour la modélisation, l'analyse et le contrôle de systèmes à événements discrets (SED), avec les automates. Même si les RdP ont été largement étudiés et utilisés dans de nombreux domaines d'application, tels que la fabrication, les transports et les communications, plusieurs problèmes demeurent non résolus. Dans cette thèse, nous nous concentrons sur deux problèmes liés, à savoir la modélisation et le développement d'un algorithme de routage pour les systèmes de communication et spécifiquement les réseaux ad-hoc et le diagnostic de défaut dans ces systèmes.

1.2. Problème de diagnostic

La détection et l'isolement des défaillances dans les systèmes de communication Ad-hoc est un sujet qui a fait l'objet de beaucoup d'attention au cours des dernières décennies. Une défaillance est définie comme toute déviation d'un système par rapport à son comportement normal ou prévu. La surveillance et le diagnostic est le processus consistant à détecter une anomalie dans le comportement du système et à isoler la cause ou la source de cette anomalie. Les défaillances des systèmes de communication peuvent provenir de plusieurs sources, telles que des erreurs de conception, des dysfonctionnements de l'équipement, etc. Le diagnostic des tous les

systèmes à évènement discret inclus les systèmes de communication, est devenu plus difficile et ne peut pas être effectué manuellement à partir d'informations empiriques. Des approches systématiques du problème du diagnostic sont nécessaires de toute urgence.

Les échecs sont inévitables dans l'environnement de technologie de communication d'aujourd'hui. Au fur et à mesure que la technologie avance, nous construisons des systèmes de plus en plus grands et fonctionnels et que nous continuons à exiger de plus en plus de la performance de ces systèmes, nous en augmentons la complexité. Par conséquent (et malheureusement), nous améliorons le potentiel de défaillance des systèmes. Quelle que soit la sécurité de nos conceptions, l'amélioration de nos techniques de contrôle de la qualité et la formation des opérateurs, les défaillances du système deviennent inévitables. Étant donné que les défaillances sont inévitables, la nécessité de moyens efficaces pour les détecter est tout à fait évidente si nous considérons leurs conséquences et leurs impacts non seulement sur les systèmes en cause, mais sur la société dans son ensemble. De plus, nous notons que des méthodes efficaces de diagnostic des défaillances peuvent non seulement aider à éviter les effets indésirables des défaillances, mais peuvent également améliorer les objectifs opérationnels. L'amélioration de la qualité des performances, de l'intégrité et de la fiabilité des services et la réduction des coûts de maintenance et d'entretien des équipements sont des avantages majeurs que peuvent offrir des schémas de diagnostic précis. Ainsi, nous voyons que des méthodes précises et opportunes de diagnostic des défaillances peuvent améliorer la sécurité, la fiabilité, la disponibilité, la qualité et l'économie des processus industriels inclus les systèmes de communication sans fil.

La nécessité de mécanismes automatisés pour le diagnostic rapide et précis des défaillances est bien comprise et appréciée à la fois par l'industrie et par le monde universitaire. De nombreux efforts de recherche ont été et sont consacrés à la conception et au développement de systèmes de diagnostic automatisés, et une variété de schémas, différant à la fois par leur cadre théorique et par leur philosophie de conception et de mise en œuvre, ont été proposés. Du point de vue conceptuel, la plupart des méthodes existantes de diagnostic des défaillances peuvent être classées comme :

(i) des méthodes basées sur un arbre de défaillances [5-8];
(ii) ii) méthodes quantitatives, fondées sur des modèles analytiques [9-11];
(iii) systèmes experts et autres méthodes fondées sur la connaissance [12];

(iv) méthodes de raisonnement basées sur des modèles [13, 14];

(v) les méthodes basées sur le SED [15-29]

1.3. Contributions

Les travaux menés dans le cadre de cette thèse consistent à proposer un protocole de routage de réseau ad hoc avec des contraintes de communication et à étudier simultanément les mécanismes de diagnostic et de réparation de la connectivité. Le protocole SynFAnt est basé sur la logique floue et le système de colonies de fourmis. Notre contribution est résumée dans les points suivants :

- ✓ Définir les types de nœuds situés sur le réseau au moment *t* et les modifications de chaque nœud au fil du temps en utilisant le raisonnement flou sur le réseau.
- ✓ Définir plusieurs paramètres ou indicateurs (par exemple, nombre de sauts, paquet, énergie résiduelle, délai et distance).
- ✓ Choix du meilleur nœud proche en fonction de la valeur de confiance fournie par les nœuds voisins, qui dépend de la logique floue dans le calcul.
- ✓ Proposant une nouvelle procédure de détection basée sur les RdP.
- ✓ L'intégration de la procédure de diagnostic / détection dans le protocole de routage a permis de réduire l'inférence mutuelle entre les nœuds et de réduire les itinéraires interrompus.
- ✓ Enfin, en utilisant plusieurs métriques du réseau, le rapport de livraison de paquet (PDR), la durée de vie du réseau (LT) sont augmentés et les délais de bout en bout sont réduits.

1.4. Structure de la thèse

La thèse est divisée en trois parties. La première partie, du chapitre 1 au chapitre 4, est une partie introductive. La deuxième partie, du chapitre 6 au chapitre 7, concerne la description de notre protocole de routage et la méthode de diagnostic d'un système de communication mobile Ad-hoc MANET utilisant des RdP. Enfin, la troisième partie est consacrée à la conclusion.

Au chapitre 2, nous présentons une revue de la littérature basée sur les résultats les plus importants du diagnostic des systèmes à évènement discret.

Au chapitre 3, nous allons abordé la nature des réseaux mobiles Ad hoc et ses limites pour la mise en œuvre des protocoles de routage classiques. Dans ce chapitre, nous vous présentons les travaux récents et antérieurs sur les problèmes de qualité et de confiance dans le domaine des réseaux mobiles Ad hoc. Tels que le débit, l'énergie, la mobilité, la fiabilité, le temps de retard des paquets, le rapport de livraison des paquets. L'extension qualité de service (Quality of service QoS) de différents types de protocoles de routage de base tels que DSR (routage source dynamique), AODV (vecteur distance Adhoc sur demande), TORA (algorithme de routage commandé temporairement) et OLSR (routage d'état de lien optimal) est expliquée. Les méthodes de gestion de la confiance dans l'environnement MANET sont abordées, telles que la définition, les métriques et les propriétés de la confiance.

Au chapitre 4, nous donnons un aperçu sur les réseaux de Petri les déférents type de cette outil et nous introduisons des notations que nous utiliserons dans la suite de la thèse,

Nous définissons par la suite dans le chapitre 5 un autre outil mathématique qui sera utilisé dans la partie de modélisation de notre approche de routage de système de communication ad-hoc qui est les Ant-system.

Le chapitre 6 présente notre approche de détection de pannes pour le SED, où nous avons pris les systèmes de communication ad-hoc comme exemple, utilisant des RdP. Nous supposons que certaines des transitions du réseau ne sont pas observables, y compris toutes les transitions qui modélisent des comportements défectueux. Notre approche de diagnostic repose sur la notion de marquage de base et de justification, qui nous permet de caractériser l'ensemble des marquages compatibles avec l'observation réelle, ainsi que l'ensemble des transitions non observables dont le déclenchement le permet. Quatre états de diagnostic sont définis, chacun correspondant à un degré d'alarme différent. Cette approche s'applique à tous les systèmes de réseau dont le sous-réseau non observable est acyclique. La logique floue et la synchronisation des transitions ont également étés intégrées dans cette méthode de diagnostic pour déterminer quelle partie ou quel élément influence le plus sur le système en cas de découverte d'une défaillance.

Au chapitre 7 nous décrivons toutes les étapes en détail du protocole de routage que nous proposons. Ce protocole inclus les deux partis commande et surveillance, Le chapitre 7 propose un routage de confiance QoS à l'aide de Réseau de Petri synchronisé Floue RdPSyncF, dans ce système, le MANET est modélisé comme un réseau de Petri dynamique Floue (RdPDF). Le

RdPDF est une structure associée à un graphe combinant des Places et des transitions. Chaque place à un jeton, associé à une valeur de confiance. Lors de la transition d'activation, le jeton est copié d'un emplacement à un autre avec une nouvelle valeur de confiance. Chaque transition est associée à un facteur de certitude (μ) et à une valeur seuil. La valeur est calculée en fonction des paramètres de qualité de la place de sortie. Si la valeur de confiance du jeton de place d'entrée est supérieure à la valeur (c'est-à-dire que la transition est déclenchée), le jeton est transféré à la place de sortie avec la nouvelle valeur de confiance. Dans le modèle MANET, les nœuds sont comme des places et les liaisons sans fil sont comme des transitions. Le nœud source envoie le paquet Route Request (RREQ) (un jeton) avec une valeur de confiance aux nœuds voisins. Après avoir reçu les paquets de requête, le nœud de destination évalue la valeur de confiance du meilleur chemin d'accès au nœud source. Ici, l'algorithme de raisonnement simultané (ARS) est utilisé pour automatiser la procédure de travail de RdPsyncF. Avec l'aide de l'ARS, la méthode proposée SyncFAnt peut sélectionner le chemin digne de confiance du nœud source et peut identifier le chemin alternatif en cas de défaillance du chemin. La performance du SyncFAnt est mesurée théoriquement et pratiquement.

Chapitre 2

Revue de littérature sur le diagnostic des systèmes à évènement discret

Résumé

Le diagnostic des pannes et la surveillance des systèmes a fait l'objet d'une attention considérable au cours des dernières décennies. Dans ce chapitre, nous présentons une étude de l'art bien détaillé sur ce sujet dans le cadre de systèmes à événements discrets en général et dans le domaine de communications et réseaux Ad-hoc en particulier.

2.1. Etat de l'art pour le diagnostic

Le diagnostic des systèmes à événements discrets (SED) est un domaine de recherche qui a fait l'objet d'une attention soutenue au cours des dernières années et qui a été motivé par la nécessité pratique d'assurer le fonctionnement correct et sûr des systèmes complexes de grande taille.

2.2.Méthodes basées sur l'arbre de défaillances

Le schéma le plus largement utilisé pour l'analyse des alarmes, en particulier dans l'industrie du contrôle des processus, repose sur des arbres de défaillances [5, 6, 30, 31]. Les arbres de défaillances fournissent une représentation graphique des relations de cause à effet des fautes dans un système. À partir d'une violation d'objectif ou d'un événement de défaillance du système indiqué par une condition d'alarme, un arbre de défaillances est créé en raisonnant en arrière depuis la défaillance du système en défaillances de base ou primales qui représentent la cause première de la défaillance.

Le principal inconvénient de cette approche est que la construction des arbres de défaillances nécessite beaucoup d'efforts. De plus, ils posent des difficultés pour la gestion des systèmes de retour d'information.

2.3.Méthodes analytiques de redondance

Une grande majorité des approches de diagnostic des défaillances proposées dans la littérature des systèmes de contrôle sont basées sur la redondance analytique (voir [10, 11]). La méthode de redondance analytique, adressée aux systèmes continus, peut être divisée en deux étapes principales:

(i) génération de résidus et (ii) isolation de décision et de panne. Le processus de génération de résidus consiste généralement à générer des signaux résiduels en comparant les valeurs prédites de variables système (à partir de modèles mathématiques du système) avec les valeurs observées réelles. Ces signaux sont nominalement proches de zéro et s'accentuent en cas de défaillance. Au stade de la décision et de l'isolement des défauts, les résidus sont examinés pour déterminer la probabilité de défauts. Un avantage majeur de cette approche est la capacité de détecter, non seulement les défauts brusques (ou graves), mais également les défauts à développement lent (ou débutants) via une analyse de tendance. Les principaux inconvénients

de cette approche sont les dépenses de calcul pour la modélisation détaillée en ligne du processus et, plus important encore, la sensibilité du processus de détection aux erreurs de modélisation et au bruit de mesure. La question de la détection robuste des défaillances à l'aide de modèles analytiques a été et est étudiée en détail.

2.4.Systèmes experts

Le diagnostic des défaillances par des systèmes experts est une approche parfaitement adaptée aux systèmes difficiles à modéliser, c'est-à-dire aux systèmes impliquant des interactions subtiles et complexes (entre composants et au sein de composants) dont les résultats sont difficiles à prévoir (voir [12] et les références qu'il contient). Le principal inconvénient des systèmes experts est qu'il faut beaucoup de temps avant d'acquérir suffisamment de connaissances pour développer l'ensemble de règles heuristiques nécessaires à un diagnostic fiable, associé au fait que cette approche est très dépendante du domaine, c'est-à-dire que les systèmes experts ne sont pas facilement transférables d'un système à l'autre. De plus, il est difficile de valider un système expert.

2.5.Méthodes de raisonnement basées sur un modèle

Une autre approche du diagnostic d'échec qui a été étudiée dans la littérature sur l'intelligence artificielle (IA) est celle du raisonnement basé sur un modèle ([13, 14, 32]). Le paradigme fondamental de cette approche, tout comme les méthodes de redondance analytique, est celui de l'observation et de la prédiction. Ces méthodes basées sur des modèles utilisent un modèle à usage général de la structure et du comportement du système, construit à l'aide de la technologie d'intelligence artificielle standard, telle que la logique des prédicats, les cadres, les contraintes et les règles. Les algorithmes de diagnostic reposent également sur des techniques standard en IA, telles que la recherche heuristique, la satisfaction de contraintes et la simulation qualitative. En général, les méthodes basées sur des modèles ne traitent que des modèles de comportement correct. Il n'y a pas de spécification a priori sur la manière dont les composants peuvent échouer, et une défaillance est considérée comme une anomalie par rapport au comportement normal.

Une autre approche du diagnostic d'échec qui a été étudiée dans la littérature sur l'intelligence artificielle (IA) est celle du raisonnement basé sur un modèle [13, 14, 32]. Le paradigme fondamental de cette approche, tout comme les méthodes de redondance analytique, est celui

de l'observation et de la prédiction. Ces méthodes basées sur des modèles utilisent un modèle à usage général de la structure et du comportement du système, construit à l'aide de la technologie d'intelligence artificielle standard, telle que la logique des prédicats, les cadres, les contraintes et les règles. Les algorithmes de diagnostic reposent également sur des techniques standard en IA, telles que la démonstration de théorèmes, la recherche heuristique, la satisfaction de contraintes et la simulation qualitative. En général, les méthodes basées sur des modèles ne traitent que des modèles de comportement correct. Il n'y a pas de spécification a priori sur la manière dont les composants peuvent échouer, et une défaillance est considérée comme une anomalie par rapport au comportement normal.

Plus récemment, les auteurs de [47] ont proposé une méthode pour l'analyse de la diagnosticabilité et le diagnostic en ligne des RdP labélisés, dite à la volée. Celle-ci consiste à analyser la propriété au cours de la construction du diagnostiqueur. Ainsi, l'analyse de seulement une partie du graphe des marquages permet en général d'obtenir un résultat de diagnostic. Cette approche a été combinée avec d'autres méthodes basées sur les T-invariants [48] ou encore les vérificateurs. D'autres méthodes se rapprochent de la méthode par diagnostiqueur mais visent à construire une partie seulement de l'espace d'état, celle qui est pertinente pour la détection des défauts et/ou la prise de décision quant à la diagnosticabilité du système. Dans [49], les auteurs utilisent une technique dite d'accessibilité de base qui s'appuie sur des RdP. Cette méthode a été d'abord introduite dans [50, 51] pour la détection de défauts. Elle vise à construire une partie réduite du graphe des marquages dite graphe d'accessibilité de base (Basis Reachability Graph ou BRG) qui est un graphe déterministe. Une notion importante pour la construction de ce graphe est celle des explications minimales. Ces dernières correspondent aux séquences d'événements non mesurés de longueur minimale qui expliquent le franchissement d'événements mesurés noter ici que notre protocole et méthode de diagnostique sera basée sur ces deux travaux avec intégration de la logique floue dans la partie de diagnostic et l'utilisation des Ant-système dans le protocole de routage/surveillance dans un système de communication Ad-hoc. Cette notion est inspirée des travaux de [52]. Ces derniers utilisent dans [53] un modèle réduit appelé automate-ROF (automate ne contenant que les événements mesurés et les événements de faute) pour tester la diagnosticabilité des RdP bornés en supposant que la partie non mesurée est acyclique. Les auteurs de [54] analysent la diagnosticabilité des RdP étiquetés en construisant deux graphes : le graphe d'accessibilité de base modifié (MBRG) et le diagnostiqueur d'accessibilité de base (BRD). Bien que la taille du

MBRG soit inférieure à celle du graphe des marquages en pratique, sa construction peut nécessiter un nombre d'étapes égal à la taille du graphe des marquages. Dans [55], l'auteur propose la construction d'un diagnostiqueur semi-symbolique. Il compare ensuite cette approche dans [56] avec deux autres approches du diagnostiqueur [47, 54]. La méthode de dépliage (Unfolding) des RdP utilisée dans [17] est appliquée au cas d'une architecture distribuée de capteurs. Cette technique permet de décrire une succession d'exécutions du système sans entrelacer les événements concurrents. Ainsi, il est supposé que le modèle RdP est composé de plusieurs sous modèles et que toutes les transitions sont étiquetées. Le superviseur collecte une séquence d'événements non synchronisés. La méthode de dépliage met donc à profit la nature distribuée du modèle et permet de construire en ligne un graphe de diagnostic sous forme d'un arbre mis à jour à chaque arrivée d'une étiquette. Ceci évite les problèmes de complexité dus à l'énumération de l'espace d'état. Cette approche a été étendue plus tard à la présence d'événements silencieux [57].

2.6.Méthodes basées sur les automates

Dans le contexte du SED, plusieurs approches théoriques originales ont été proposées en utilisant des automates. Dans [33] et [34] les auteurs proposent une approche pour les SED basée sur l'état pour le diagnostic d'échec.

Les problèmes de diagnostic hors ligne et en ligne sont abordés séparément et les notions de diagnosticabilité dans ces deux cas sont présentées. Les auteurs proposent un algorithme pour le calcul d'une commande de diagnostic, c'est-à-dire une séquence de commandes de test permettant de diagnostiquer les défaillances du système. La convergence de cet algorithme est garantie si le système remplit les conditions pour le diagnostic en ligne.

Dans [35] et [36] Sampath et al. proposent une approche du diagnostic des défaillances dans laquelle le système est modélisé comme un SED dans lequel les défaillances sont traitées comme des événements non observables ; le diagnostic est le processus de détection des occurrences de ces événements à partir des séquences d'événements observés. Le niveau de détail dans un modèle à événements discrets semble être tout à fait adéquat pour une grande classe de systèmes et pour diagnostiquer une grande variété de défaillances. Cette approche est applicable chaque fois que des défaillances entraînent une modification distincte de l'état du système, mais ne l'arrêtent pas nécessairement. Dans [35], les auteurs fournissent une définition de la possibilité de diagnostic dans le cadre de langages formels et établissent les conditions

nécessaires et suffisantes pour la possibilité de diagnostic des systèmes. On présente également dans [35] une approche systématique pour résoudre le problème du diagnostic à l'aide de diagnostiqueurs. Dans [37], Sampath et al. présentent une approche intégrée du contrôle et du diagnostic. Plus spécifiquement, les auteurs présentent une approche pour la conception de systèmes pouvant être diagnostiqués par une conception appropriée du contrôleur de système. Cette approche est appelée diagnostic actif. Ils formulent le problème de diagnostic actif en tant que problème de contrôle de supervision. La procédure adoptée pour résoudre le problème de diagnostic actif est la suivante : étant donné le langage non identifiable généré par le système d'intérêt, ils choisissent d'abord un sous-langage «approprié» de ce langage comme langage juridique. Le choix du langage juridique est un problème de conception et dépend généralement de considérations telles que le comportement acceptable du système (ce qui garantit que le comportement du système n'est pas limité au strict nécessaire pour le rendre éventuellement diagnostiquable) et le délai de détection des défaillances. Une fois que le langage juridique approprié est choisi, ils conçoivent ensuite un contrôleur (contrôleur de diagnostic), qui réalise un langage en boucle fermée qui est contenu dans le langage juridique et qui peut être diagnostiqué. Ce contrôleur est conçu sur la base du cadre formel et des techniques de synthèse fournies par la théorie du contrôle de supervision, avec la contrainte supplémentaire de la diagnosticabilité.

Dans [20] Debouk et al. traitent le problème du diagnostic d'échec dans le SED avec des informations décentralisées. Ils proposent une architecture décentralisée coordonnée comprenant deux sites locaux communiquant avec un coordinateur chargé de diagnostiquer les défaillances survenant dans le système. Ils étendent la notion de possibilité de diagnostic, initialement introduite dans [35] pour les systèmes centralisés, à l'architecture décentralisée coordonnée proposée. En particulier, ils spécifient trois protocoles qui réalisent l'architecture proposée et analysent les propriétés de diagnostic de ces protocoles.

Dans [18], Boel et van Schuppen abordent le problème de la synthèse des protocoles de communication et des algorithmes de diagnostic de défaillance pour le diagnostic de défaillance décentralisé du système avec une communication coûteuse entre les diagnostiqueurs. Les coûts sur les canaux de communication peuvent être décrits en termes de bits et de complexité. Les coûts de communication et de calcul imposent un compromis entre l'objectif de contrôle du diagnostic de défaillance et celui de la minimisation des coûts de communication et de calcul.

Le résultat de cet article est un algorithme de diagnostic d'échec décentralisé de SED dans le cas particulier de deux diagnostiqueurs.

Dans [38], Zad et al. présentent une approche basée sur l'état pour le diagnostic passif en ligne de défaut. Dans ce cadre, le système et le diagnostiqueur (le système de détection de pannes) ne doivent pas être initialisés en même temps. De plus, aucune information sur l'état ni même sur la condition (état de panne) du système avant le lancement du diagnostic n'est requise. La conception du système de détection de pannes, dans le pire des cas, présente une complexité exponentielle. Un schéma de réduction de modèle avec une complexité temporelle polynomiale est introduit pour réduire la complexité de calcul de la conception. La diagnosticabilité des défaillances est étudiée et les conditions nécessaires et suffisantes pour la diagnosticabilité des défaillances sont dérivées.

Dans [26], Jiang et Kumar présentent une méthode de diagnostic d'échec du SED avec des spécifications de logique temporelle linéaire (LTL). Les formules LTL sont utilisées pour spécifier les échecs dans le système. Les spécifications basées sur le LTL rendent le processus de spécification de spécification plus facile et plus convivial que les spécifications basées sur un langage formel / un automate. Ils peuvent capturer les défaillances représentant la violation des propriétés de sécurité et d'activité, alors que les spécifications de langage formel / automates antérieurs peuvent capturer les défaillances représentant la violation des seules propriétés de sécurité (telles que la survenue d'un événement défectueux ou l'arrivée à une État défaillant). La prédiagnosabilité et la diagnosticabilité du SED dans le cadre de la logique temporelle sont définies. Le problème des tests de prédiagnosabilité et de diagnostic est réduit au problème de la vérification des modèles. Un algorithme pour le test de prédiagnosabilité et de diagnosticabilité, et la synthèse d'un diagnostiqueur est obtenu. La complexité de l'algorithme est exponentielle dans la longueur de chaque formule LTL de spécification et polynomiale dans le nombre d'états du système et le nombre de spécifications.

Dans [28] Lunze et al. décrient une méthode pour détecter et identifier les défauts qui se produisent dans les capteurs ou dans les actionneurs de systèmes dynamiques avec des entrées et des sorties discrètes. Le modèle utilisé dans le diagnostic est un automate stochastique. Le schéma d'observateur généralisé (SOG), qui a été proposé pour les systèmes avec des entrées et des sorties à variation continue, est développé pour les systèmes discrets. Ce schéma résout le

problème de diagnostic en tant que problème d'observation. Le système considéré étant décrit par un automate stochastique plutôt que par une équation différentielle, le contexte mathématique et les algorithmes de diagnostic obtenus sont complètement différents des observateurs bien connus développés pour les systèmes à variable continue. Le SOG est complété ici par un module de détection de défaut pour faire face aux défauts de l'installation qui sont différents des défauts d'actionneur ou de capteur. L'algorithme de diagnostic comprend deux étapes, la première détectant l'existence d'un défaut et le second isolant les défauts possibles du capteur ou de l'actionneur ou identifiant les défauts de l'installation.

2.7.Méthodes basées sur les réseaux de Petri

Bien que les modèles d'automates conviennent à la description du SED, l'utilisation des réseaux de Petri (RdP) offre des avantages significatifs en raison de leur double représentation : graphique et mathématique. De plus, la nature intrinsèquement distribuée des RdP lorsque la notion d'état (c'est-à-dire de marquage) et d'action (c'est-à-dire de transition) est locale réduit la complexité informatique impliquée dans la résolution d'un problème de diagnostic. Parmi les premiers travaux pionniers traitant des RdP, nous rappelons le travail de Prock dans [39] qui propose une technique en ligne de détection des fautes basée sur la surveillance du nombre de jetons résidant dans des invariants P: lorsque le nombre de jetons dans P- invariants change, alors une erreur est détectée. Dans [40], Sreenivas et Jafari utilisent des RdP de temps pour modéliser les transitions de contrôleur d'un SED et de renversement afin de déterminer si un état donné est invalide. Plus tard, les RdP sont utilisés par Ghazel et al. [24] une approche de surveillance pour le RdP avec des événements non observables, représenter le comportement «a priori» connu du système et suivre en ligne son état pour identifier les événements survenus. Hadjicostis et Veghese dans [25] utilisent des modèles RdP pour introduire la redondance dans le système et des invariants P supplémentaires permettent la détection et l'isolement des marquages défectueux. La redondance dans une RdP donnée est utilisée par Wu et Hadjicostis [41] pour permettre la détection et l'identification de fautes à l'aide de techniques de décodage algébrique. Ils considèrent deux types de défauts : les défauts de place qui corrompent le marquage de réseau et les défauts de transition qui entraînent une mise à jour incorrecte du marquage après l'occurrence de l'événement. Bien que cette approche soit générale, le marquage net doit être observable périodiquement, même si des événements non observables se produisent. De manière analogue, Lefebvre et Delherm [27] étudient la détermination de l'ensemble des lieux à observer pour l'estimation exacte et immédiate de l'occurrence de failles.

Ramirez-Trevino et al. [29] utilisent des RdP interprétées pour modéliser le comportement du système qui inclut à la fois des événements et des états partiellement observables. Sur la base du modèle RdP interprété dérivé d'une méthodologie en ligne, un schéma utilisant une solution d'un problème de programmation est proposé pour résoudre le problème du diagnostic. Dans [22] Dotoli et al. présentent une nouvelle approche basée sur les événements pour la surveillance en ligne des SED, assurant une détection rapide et précise des défaillances du système. Le modèle de contrôle est basé sur des RdP hybrides du premier ordre, c'est-à-dire des réseaux utilisant l'approximation du premier ordre [42]. La technique d'analyse des pannes proposée repose sur un cadre modulaire, de sorte que des moniteurs élémentaires puissent être connectés à d'autres moniteurs pour vérifier des systèmes plus complexes tout en évitant le problème de l'explosion de l'espace d'état. En outre, le moniteur présenté détecte les défaillances du système dès que possible, avant le temps d'exécution maximal attribué à chaque tâche.

Notez que tous les articles de cette rubrique supposent que les erreurs sont modélisées par des transitions non observables. Cependant, alors que les documents susmentionnés supposent que le marquage de certains places peut être respecté, une série de documents ont été récemment présentés et reposent sur l'hypothèse qu'aucun place n'est observable [17, 23, 43].

En particulier, Genc et Lafortune [23] proposent un diagnostiqueur sur la base d'une approche modulaire permettant de diagnostiquer les défauts de chaque module. Ensuite, les diagnostiqueurs récupèrent les informations de diagnostic monolithiques obtenues lorsque tous les modules sont combinés en un seul module préservant le comportement du système modulaire sous-jacent. Un système de communication connecte les différents modules et met à jour les informations de diagnostic. Même si cette approche n'évite pas le problème de l'explosion d'état, une amélioration est obtenue lorsque le système peut être modélisé comme une collection de modules de RdP couplés via des places communs.

Le principal avantage des approches de Genc et Lafortune [23] réside dans le fait que, si le réseau est borné, le diagnostiqueur peut être construit hors ligne, ce qui permet de déplacer hors ligne la partie la plus fastidieuse de la procédure. Néanmoins, une caractérisation de l'ensemble de marquages cohérente avec l'observation réelle est nécessaire. Ainsi, une grande mémoire peut être nécessaire.

En particulier, Benveniste et al. [17] utilisent une approche de déploiement du réseau pour concevoir un diagnostiqueur asynchrone en ligne. L'explosion d'état est évitée mais le calcul en ligne peut être élevé en raison de la construction en ligne des structures RdP par le déploiement.

Dotoli et al. dans [21], afin d'éviter la refonte et la redéfinition du diagnostiqueur lorsque la structure du système change, a proposé un diagnostiqueur fonctionnant en ligne. En particulier, il attend un événement observable et l'algorithme décide si le comportement du système est normal ou s'il peut présenter des défauts éventuels. À cette fin, certains problèmes ILP sont définis et fournissent finalement les séquences minimales de transitions non observables contenant les erreurs éventuellement survenues. L'approche proposée est une technique générale puisqu'aucune hypothèse n'est imposée sur l'ensemble d'états accessibles qui puisse être illimité, et seules quelques propriétés doivent être remplies par la structure du RdP qui modélise le comportement de défaut du système.

Il est à noter qu'aucun des documents susmentionnés concernant les RdP ne traite de la diagnostiquabilité, c'est-à-dire qu'aucun d'eux ne fournit une procédure permettant de déterminer a priori si un système est diagnostiquable, c'est-à-dire s'il est possible de reconstituer l'occurrence d'événements de faute observant des mots de longueur finie.

En fait, alors que ce problème a fait l'objet d'une étude approfondie dans le cadre des automates, comme indiqué ci-dessus, très peu de résultats ont été présentés dans le cadre des RdP.

La première contribution sur la possibilité de diagnostic des RdPs a été donnée par Ushio et al. [44]. Ils étendent une condition nécessaire et suffisante pour la diagnosticabilité donnée par Sampath et al. [35, 36] supposent que l'ensemble des places est partitionné en places observables et inobservables, alors que toutes les transitions sont non observables en ce sens que leurs occurrences ne peuvent pas être observées. À partir de RdP, ils construisent un diagnostiqueur appelé ω diagnostiqueur simple qui leur fournit les conditions suffisantes pour la diagnostiquabilité des RdP sans bornes.

Chung in [19], contrairement à l'article de Ushio, suppose qu'une partie des transitions de la modélisation RdP est observable et montre que les informations supplémentaires issues des transitions observées ajoutent en général une possibilité de diagnostic au système analysé. De plus, à partir du diagnostiqueur, il propose un automate appelé vérificateur qui permet un

mécanisme de contrôle polynomial sur la possibilité de diagnostic, mais pour les modèles de réseaux de Petri à états finis.

Dans [45], Wen et Jeng ont proposé une approche pour tester la possibilité de diagnostic en vérifiant la propriété de structure de T-invariant des réseaux. Ils utilisent le diagnostiqueur Ushio pour prouver que leur méthode est correcte, mais ils ne construisent pas de diagnostiqueur pour que le système puisse effectuer le diagnostic. Dans [46] Wen et al. ont présenté un algorithme, basé sur un problème de programmation linéaire, de complexité polynomiale en nombre de nœuds pour le calcul d'une condition suffisante de diagnosticabilité du SED modélisé par RdP.

Les recherches futures dans le domaine du diagnostic des pannes pourraient suivre plusieurs directions. Tout d'abord, les conditions de diagnosticabilité ne font pas l'objet d'une enquête complète lors de l'utilisation de RdP. Plus précisément, il est nécessaire d'établir s'il est possible de détecter avec un retard fini les défaillances en utilisant un enregistrement des événements observés. Deuxièmement, les procédures proposées ne sont pas applicables aux RdP étiquetés qui présentent une forme de non-déterminisme, tels que deux transitions ou plus qui partagent la même étiquette. Troisièmement, la complexité informatique des approches en ligne doit être améliorée afin d'appliquer la technique aux systèmes à grande échelle.

2.8.Méthodes basées sur les propriétés structurelles

Prock [39] est l'un des premiers auteurs à exploiter les propriétés structurelles des RdP pour le diagnostic des SED. Des RdP représentant le comportement normal du système sont utilisés et les fautes sont détectées suite à une variation du nombre de jetons dans un P-invariant. Les auteurs de [46] supposent, quant à eux, une mesure partielle des événements et du marquage. En étudiant la relation entre la diagnosticabilité et les invariants du RdP, ils établissent une condition suffisante pour la diagnosticabilité. Dans [29], les auteurs considèrent des RdP interprétés, saufs et fortement connexes. Le diagnostic est réalisé grâce à un diagnostiqueur basé sur la partie nominale du RdP interprété. L'algorithme compare ainsi la sortie du système avec celle du diagnostiqueur pour la détection d'une déviation du comportement synonyme de l'occurrence d'une faute. Dans [58], les auteurs s'intéressent au cas des RdP non bornés et utilisent une méthode inspirée du diagnostic des SED modélisés par les automates à états finis appelée approche vérificateur (ou produit jumelé) [59, 60]. Le principe consiste à réaliser dans

un premier temps une copie du modèle du RdP en renommant les événements silencieux et en éliminant les événements de faute. Ensuite, l'approche réalise le produit synchronisé entre le modèle et sa copie et vérifie la présence, sur ce nouveau modèle, de cycles ambigus. La présence de ces derniers est synonyme de la non-diagnosticabilité du système. En effet, un cycle ambigu correspond à deux séquences, l'une normale et l'autre fautive, ayant le même comportement mesuré et donc indiscernables. L'approche vérificateur a inspiré plusieurs travaux utilisant le même principe de base [19, 61, 62] et plus récemment [63] où les auteurs s'intéressent à la détection de motifs de faute (suite d'événements traduisant un certain comportement fautif du système) tout en exploitant les avantages de la méthode de dépliage.

2.9. Méthodes basées sur les techniques algébriques

Ces méthodes exploitent la représentation mathématique des RdP. Dans [41], les auteurs se basent sur un modèle représentant uniquement le comportement normal du système. Deux types de fautes sont considérés : des fautes sur les places qui faussent le marquage et des fautes sur les transitions qui engendrent un marquage incorrect. Ils utilisent une méthode d'identification qui s'appuie sur une technique de décodage algébrique et de redondance sur les places. Les auteurs supposent une mesure périodique du marquage du système. Dans [64], les auteurs s'intéressent au cas des RdP partiellement mesurés (mesure partielle des transitions et des marquages) et développent une technique pour le calcul du degré de croyance (Belief) qui représente le ratio, suivant une trajectoire de mesure, entre le nombre de trajectoires compatibles fautives et le nombre total de trajectoires compatibles. Le même formalisme est exploité dans [65] avec une mesure partielle des transitions et des marquages. La méthode est basée sur la résolution de LMI dans le but de déterminer tous les comportements compatibles avec la trajectoire de mesure. Cette méthode a été adaptée au diagnostic en ligne dans [66] en limitant la taille de la trajectoire de mesure. Dans [67], le diagnostic s'appuie sur la représentation mathématique des RdP et une résolution de systèmes linéaires avec contraintes en utilisant la programmation linéaire en nombres entiers (Integer Linear Programming ou ILP) en ligne. Cette technique permet de déterminer l'ensemble des séquences réalisables et compatibles avec les mesures et de tester la présence de fautes dans celles-ci. Cette approche est étendue dans [68] au cas de l'analyse hors ligne de la K diagnosticabilité, c'est-à-dire la capacité à détecter la faute au pire K événements après son occurrence. Dans [69], les auteurs introduisent la notion de gmarking. Il est obtenu lorsque l'on souhaite mettre à jour un marquage avec un événement mesuré en utilisant l'équation d'état. Le marquage obtenu est généralement

caractérisé par des composantes négatives dues à la présence de transitions silencieuses. En effet, l'événement mesuré n'est pas validé au marquage considéré et le franchissement d'une séquence de transitions silencieuses est nécessaire pour sa validation. L'utilisation de la programmation linéaire en nombres entiers permet à partir des g-markings d'obtenir en ligne les vecteurs de franchissement qui expliquent la mesure et donc de détecter l'occurrence d'une faute. Dans [70], ces marquages sont utilisés pour établir deux conditions suffisantes : l'une pour la non diagnosticabilité et l'autre pour la diagnosticabilité. Plus récemment, la programmation linéaire en nombres entiers a été exploitée dans [71]. Dans ce travail, les auteurs s'intéressent au diagnostic en ligne des RdP dans le cadre d'une architecture décentralisée. Le Tableau 2.1 bien que non exhaustif, permet d'avoir un aperçu sur le classement des travaux sur le diagnostic aves les RdP en fonction du type de modèle considéré (incluant ou non le comportement fautif), du type de mesures et de la méthode utilisée. Nous remarquons que la majorité des approches est soit basée sur une mesure partielle des événements, soit sur une mesure partielle des marquages. Dans notre cas, nous nous intéressons au diagnostic des RdP partiellement mesurés incluant le modèle de dysfonctionnement et nous considérons à la fois une mesure partielle sur les transitions et les marquages. Ce formalisme offre une plus grande capacité de modélisation des différents types de capteurs. À titre d'exemple, une transition mesurée est plus adaptée pour modéliser un compteur du nombre de pièces usinées dans un atelier. Tandis qu'une place mesurée sera utilisée pour modéliser un capteur de type caméra fournissant le nombre de pièces en cours de traitement. Les méthodes se focalisant sur un seul type de mesure ne sont donc pas directement exploitables avec ce type de formalisme. Dans [64], les auteurs s'intéressent au diagnostic des RdP partiellement mesurés. Cependant, ils proposent une transformation du RdP partiellement mesuré (pour des cas particuliers de mesure partielle du marquage) en un RdP étiqueté (c.-à-d. que la mesure partielle du marquage est représentée par des transitions mesurées dans le nouveau RdP). Bien que cette transformation ait été étendue au cas général d'une mesure pondérée d'un sous ensemble de places dans [72],

2.10. Conclusion

L'état de l'art dans ce chapitre a montré que la problématique du diagnostic des systèmes à évènement discret inclus le système modélisés par des RdP a été largement abordée au cours des dernières années. Dans notre cas, nous avons considéré la problématique du diagnostic et de surveillance des systèmes de communication mobile ad hoc MANET comme un exemple

d’un système à évènement discret, avec les RdP commme outil de modélisation et de diagnostic, Ces réseaux sont caractérisés par une mesure partielle à la fois des transitions et des marquages.

Tableau 2.1. Classification des travaux sur le diagnostic des RdP

	RdP incluant le modèle de dysfonctionnement	Mesure partielle des…		Technique utilisée					
		Marquages	Transitions	ILP/LMI	Algébrique	Diagnostiqueur & graphe des marq.	Vérificateur	Dépliage	Propriétés structurelles
(Basile et al. 2008; Basile et al. 2009; Dotoli et al. 2009; Basile et al. 2012; Cong et al. 2017)	●		●	●					
(Lefebvre 2014a; Lefebvre 2014c)	●	●	●	●					
(Wu & Hadjicostis 2005)		●			●				
(Lefebvre & Delherm 2007)	●	●			●				
(Ru & Hadjicostis 2009)	●	●	●		●				
(Ushio et al. 1998)	●	●				●			
(Genc & Lafortune 2003; Giua & Seatzu 2005; Genc & Lafortune 2007; Cabasino et al. 2010; Cabasino et al. 2011; Cabasino, Giua, et al. 2014; Li et al. 2015a; Liu et al. 2017; Jiroveanu & Boel 2010)	●		●			●			
(Jiang et al. 2001; Yoo & Lafortune 2002; Chung 2005; Madalinski et al. 2010; Cabasino et al. 2012; Gougam et al. 2014; Gougam et al. 2013; Cabasino, Giua, Lafortune, et al. 2009; Li et al. 2015b)	●		●				●		
(Benveniste et al. 2003; Haar et al. 2013)	●		●					●	
(Ramírez-Treviño et al. 2007; Wen et al. 2005)		●	●						●
(Prock 1991)		●							●

Chapitre 3

Les réseaux mobiles ad hoc (MANET)

Résumé

Dans ce chapitre, les propriétés des réseaux mobiles ad hoc (MANET) sont décrites. Les concepts de qualité de service (QoS) et de confiance sont expliqués dans le contexte MANET.

3.1. Introduction

Au cours des dernières décennies, le secteur des communications sans fil a connu une croissance plus rapide que celle des réseaux câblés traditionnels. Grâce au réseau sans fil, un utilisateur peut accéder aux applications ou partager les informations sans fil. Il permet à l'utilisateur d'étendre le réseau dans le monde entier. Les points d'accès sans fil aux zones locales permettent aux utilisateurs d'utiliser des applications Internet avec des périphériques portables tels que des ordinateurs portables, des assistants numériques personnels (ANP), des téléphones intelligents et des tablettes. Les réseaux locaux sans fil (WLAN) sont généralement classés en deux types de réseaux avec et sans infrastructure. La figure 3.1 illustre le réseau d'infrastructure avec trois réseaux locaux sans fil avec leurs points d'accès, les points d'accès étant connectés au réseau filaire. Dans le réseau local sans fil, la communication entre deux nœuds mobiles se fait uniquement via le point d'accès et les nœuds ne peuvent pas communiquer directement. La limitation du réseau d'infrastructure réside dans le fait que les nœuds dépendent du point d'accès et nécessitent une administration centralisée. Le réseau alternatif est moins d'infrastructure.

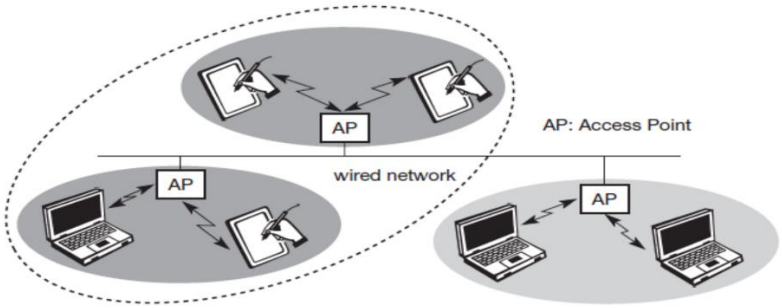


Figure 3.1. Réseau local sans fil basé sur l'infrastructure

La Figure 3.2 présente les réseaux MANET (Mobile Ad hoc Networks), qui permettent aux applications réseau de s'exécuter sans infrastructure fixe. Dans les MANET, plusieurs nœuds forment collectivement des réseaux auto-organisés et auto-administratifs. En raison de la mobilité des nœuds, un nœud peut rejoindre ou quitter le réseau à tout moment. Dans le réseau, un nœud communique directement avec les autres nœuds situés dans sa plage de transmission et communique avec les nœuds éloignés par le biais des nœuds intermédiaires.

3.2. Définition du mot ad hoc

- ✓ en latin : qui va vers ce vers quoi il doit aller, c'est-à-dire « formé dans un but précis »

- ✓ Une personne ad hoc signifie donc que pour un individu donné ayant une connaissance accrue d'une matière, cet individu est parfaitement qualifié pour exécuter la tâche qui lui est confiée
- Ad hoc = Convient, adapté

3.2.1. Les réseaux ad hoc

Un réseau mobile ad hoc, appelé généralement MANET (Mobile Ad hoc network), consiste en une grande population, relativement dense, d'unités mobiles qui se déplacent dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée.

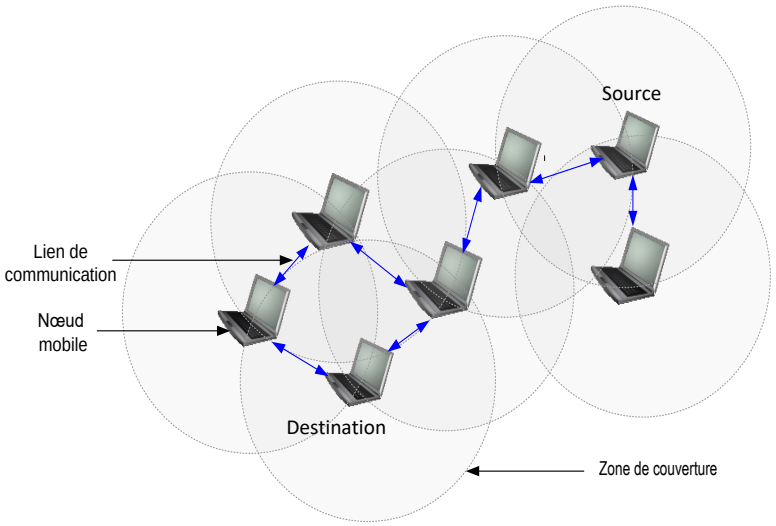


Figure 3.2. Infrastructures d'un réseau mobile ad hoc.

Dans la figure 3.2, le nœud source envoie les données via trois niveaux de nœuds intermédiaires au nœud de destination, qui n'est pas en communication directe. Les nœuds du réseau ne nécessitant aucune administration centralisée, les MANET sont faciles à déployer et conviennent aux applications rapides et urgentes telles que les secours en cas de catastrophe et les applications militaires.

3.3. Propriétés des réseaux mobiles ad hoc

Les MANET sont des réseaux sans infrastructure. Les propriétés des MANET diffèrent de celles des réseaux câblés et sans fil traditionnels.

3.3.1. Opération distribuée

Il n'y a pas d'administration centralisée pour gérer les nœuds du réseau. Le contrôle sur les opérations du réseau est distribué. Chaque nœud agit en tant que routeur et peut prendre ses propres décisions en matière de réseau. Pendant que les applications sont en cours d'exécution,

les nœuds doivent communiquer et coopérer les uns avec les autres. Il est difficile d'implémenter des protocoles basés sur une infrastructure, ce qui nécessite une administration et un contrôle centralisés.

3.3.2. Organisation personnelle

Dans les MANET, un nœud mobile peut rejoindre ou quitter le réseau à tout moment. Par conséquent, le groupe de nœuds mobiles tous ensemble peut former un réseau ou se dissoudre de manière dynamique en fonction de la demande. Un nœud peut conserver toutes les informations du réseau en échangeant des mises à jour avec ses nœuds voisins. En raison de leur capacité d'auto-organisation, les MANET sont adaptables aux changements dynamiques.

3.3.3. Routage multi-sauts

Un nœud peut envoyer les informations directement à ses nœuds voisins à un saut. Mais si le nœud cible est à plusieurs sauts, les paquets de données sont envoyés via la séquence de nœuds intermédiaires. Dans ce routage à sauts multiples, chaque nœud doit gérer la liste des nœuds voisins de contact pour atteindre le nœud de destination.

3.3.4. Terminaux légers

Dans les MANET, les nœuds mobiles sont comme des ordinateurs portables, des capteurs, des téléphones intelligents et des appareils portables. Habituellement, ces terminaux disposent de ressources limitées telles que la taille de la mémoire, la capacité de l'UC, la batterie et la capacité de stockage.

3.3.5. Support physique partagé

Dans les MANET, les nœuds communiquent via le support sans fil. Ainsi, tous les avantages et inconvénients du support physique sans fil sont applicables aux MANET. En raison de la nature ouverte du support sans fil, un nœud peut observer les activités du nœud voisin et peut prendre les contre-mesures nécessaires.

3.4. Défis des réseaux mobiles ad hoc

En raison des changements de topologie dynamiques et du manque d'administration des MANET, la mise en œuvre des protocoles classiques filaires et sans fil pose de nombreux problèmes. Voici les défis dans les MANET.

3.4.1 Topologie dynamique

En raison de la mobilité des nœuds, la topologie du réseau continue à changer. Cela augmente le risque de maintenance du réseau sur un nœud. La nature dynamique des MANET crée la nécessité de modifier tous les protocoles classiques afin de prendre en charge les changements topologiques.

3.4.2. Liens de capacité variable sous contrainte de bande passante

Etant donné que les MANET utilisent un support sans fil, les liaisons sans fil entre les nœuds ont une capacité inférieure à celle des réseaux câblés. Les largeurs de bande des liaisons sans fil sont influencées par les effets des évanouissements, des conditions d'interférence et des accès multiples.

3.4.3. Contraintes de la batterie

Les appareils à MANET utilisent des appareils avec une puissance de batterie limitée. Parfois, le réseau peut être partitionné ou les routes établies sont déconnectées en raison de la défaillance de nœuds avec des contraintes de batterie. Dans le réseau, la puissance de transmission d'un nœud est influencée par le routage, les protocoles MAC et les calculs de la CPU.

3.4.4. La Qualité de Service dans les réseaux mobiles Ad hoc

Dans les MANET, la bande passante, le délai, le retard de transmission, le débit et la fiabilité sont généralement considérés comme des paramètres de qualité de service. La définition des paramètres de qualité de service dépend uniquement du contexte de l'application. En raison des ressources limitées et du manque d'administration, la fourniture de qualité de service n'est pas une tâche triviale dans les MANET.

3.4.5. Menaces à la sécurité

L'utilisation de supports sans fil dans les MANET provoque des menaces pour la sécurité. Un nœud peut entendre les activités de l’autre nœud et créer les problèmes de sécurité. Dans le réseau, les nœuds peuvent devenir égoïstes lorsqu'ils manquent de ressources.

3.5. Protocoles de routage dans les MANET

Le but des protocoles de routage dans le MANET est d'établir la route avec une longueur minimale du nœud source au nœud de destination ; qui sont multi hop loin distance les uns avec

les autres. En raison de modifications dynamiques de la topologie du réseau, les protocoles de routage de réseau câblé existants ne peuvent pas être appliqués au MANET. De nombreux auteurs ont proposé des protocoles de routage pour le MANET en améliorant les protocoles de routage existants afin qu'ils puissent prendre en charge les caractéristiques de MANET.

Sur la base de méthodes de gestion des informations de routage, les protocoles de routage dans les MANET sont classés en trois catégories: protocoles de routage proactifs (pilotés par des tables), protocoles de routage réactif (à la demande) et protocoles de routage hybrides.

3.5.1. Protocoles de routage proactifs ou pilotés par des tables

Les protocoles de routage pilotés par table conservent les informations réseau avant que cela ne soit nécessaire. Ici, chaque nœud conserve les informations de contiguïté pour tous les nœuds du réseau. Les informations de routage sont conservées sous forme de tables et sont partagées entre les nœuds voisins. Les informations contenues dans les tables de routage changent en fonction des modifications de la topologie. Les protocoles de routage pilotés par table ne sont pas recommandés pour les grands réseaux, car ils nécessitent de conserver une entrée pour chaque nœud du réseau. Cela augmente la charge de contrôle des protocoles de routage et consomme des ressources de nœud. Nous allons décrire dans ce qui suit, les protocoles les plus importants de cette classe :

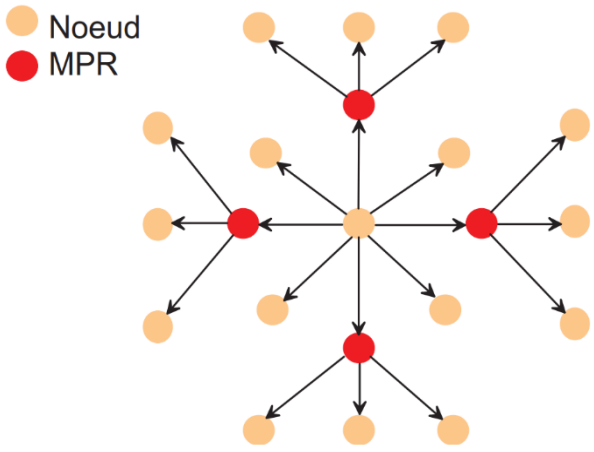


Figure 3.3. Le principe des nœuds MPR

3.5.1.1. Le protocole OLSR

Le protocole OLSR (Optimized Link State Routing) [73]est une version adaptée au cas des réseaux sans fil du protocole LSR (Link State Routing) utilisé pour les réseaux filaires, notamment dans le protocole OSPF (Open Shortest Path First) [74]. Le protocole LSR fonctionne sur le principe d'une inondation globale du réseau par les messages de contrôle :

chaque nœud signale à ses voisins périodiquement son état ; les voisins propagent à leur tour cette information qui envahit le réseau. Cette technique pose malgré tout un problème dans le cas des réseaux sans fil car la bande passante disponible est limitée et un effondrement de la capacité du réseau apparaît rapidement, surtout dans le cas de réseau de grande dimension. Le protocole OLSR améliore le protocole LSR en introduisant la notion de MPR (Multi Point Relay) [75]. Les MPR sont des nœuds élus qui assurent le relais de l'information dans le réseau. Chaque nœud émet la liste de ses voisins mais seuls les nœuds MPR la rediffusent (voir la Figure 3.3). Ce sont les nœuds MPR qui assurent le routage dans le réseau.

Le protocole OLSR étant un protocole proactif, il utilise des messages de contrôle périodiques afin d'actualiser les tables de routage. Les deux principaux messages utilisés sont les paquets "Hello" et les paquets TC (Topologie Control). Les paquets "Hello" sont envoyés par tous les nœuds et contiennent des informations sur le voisinage des nœuds, ils permettent à chaque nœud de calculer ses MPR. Les paquets TC, quant à eux, contiennent l'information sur les MPR et servent à calculer les tables de routage. Le protocole OLSR est performant dans les réseaux denses car les MPR permettent de limiter l'inondation du réseau. De plus il est très réactif, chaque nœud sait à tout moment comment atteindre les autres. Par contre, en termes de consommation énergétique, il sollicite beaucoup les nœuds car ils doivent émettre en permanence des messages et c'est en émission que les supports radio consomment le plus : OLSR est difficilement applicable pour des réseaux de capteurs.

3.5.1.2. Le protocole TBRPF

Le protocole TBRPF (Topology Broadcast based on Reverse Path Forwarding) est basé sur une représentation arborescente du réseau [76]. Chaque nœud construit un arbre lui permettant d'avoir une route vers tous les nœuds du réseau et ce grâce à l'algorithme de Dijkstra afin d'avoir les routes les plus courtes en nombre de sauts. En outre, le protocole TBRPF économise de la bande passante en échangeant uniquement les modifications du réseau et non les tables de routage entière. Les messages "Hello" de mise à jour sont donc différentiels et rapportent uniquement les changements dans les voisinages. Cela permet en outre des économies de ressources.

La figure 3.4 montre un exemple d'arborescence avec un réseau de vingt nœuds : quatre nœuds sont des sommets d'arbre (en rouge) et les autres sont des feuilles. Si le nœud *u* génère un message, seuls les nœuds rouges qui ne sont pas des feuilles propagent l'information.

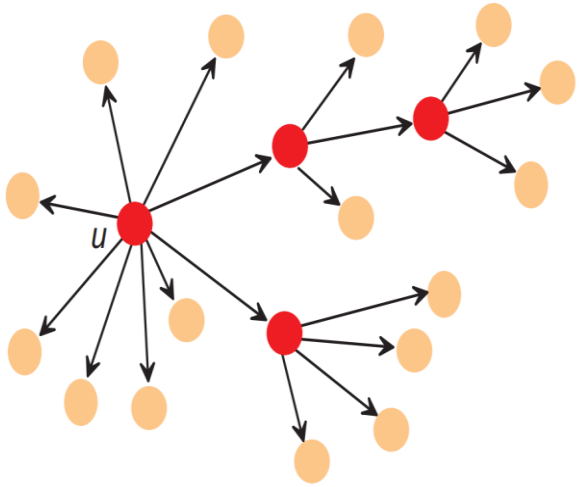


Figure 3.4. Le principe de l'arborescence dans le protocole TBRPF d'après [77]

3.5.1.3. Le protocole de routage « DSDV »

Basé sur l'idée classique de l'algorithme distribué de Bellman-Ford en rajoutant quelques améliorations.

Chaque station mobile maintient une table de routage qui contient :

- ✓ *Toutes les destinations possibles.*
- ✓ *Le nombre de nœud (ou de sauts) nécessaire pour atteindre la destination.*
- ✓ *Le numéro de séquences (SN : sequence number) qui correspond à un nœud destination.*

Le NS est utilisé pour faire la distinction entre les anciennes et les nouvelles routes, ce qui évite la formation des boucles de routage.

La mise à jour dépend donc de deux paramètres : Le temps, c'est à dire la période de transmission, et Les événements

Un paquet de mise à jour contient :

1. *Le nouveau numéro de séquence incrémenté, du nœud émetteur. Et pour chaque nouvelle route :*
2. *L'adresse de la destination.*
3. *Le nombre de nœuds (ou de sauts) séparant le nœud de la destination.*
4. *Le numéro de séquence (des données reçues de la destination) tel qu'il a été estampillé par la destination.*

Le DSDV élimine les deux problèmes de boucle de routage "routing loop", et celui du

"counting to infinity".

Cependant :

Dans ce protocole, une unité mobile doit attendre jusqu'à ce qu'elle reçoive la prochaine mise à jour initiée par la destination, afin de mettre à jour l'entrée associée à cette destination, dans la table de distance. Ce qui fait que le DSDV est lent.

Le DSDV utilise une mise à jour périodique et basée sur les événements, ce qui cause un contrôle excessif dans la communication.

3.5.1.4. Le protocole de routage « FSR »

Le protocole FSR, voir figure 3.5 est basé sur l'utilisation de la technique "œil de poisson" (fisheye), proposée par Kleinrock et Stevens, qui l'ont utilisé dans le but de réduire le volume d'information nécessaire pour représenter les données graphiques.

Dans la pratique, l'œil d'un poisson capture avec précision, les points proches du point focal. La précision diminue quand la distance, séparant le point vu et le point focal, augmente.

La figure suivante illustre la technique FE utilisée par le protocole :

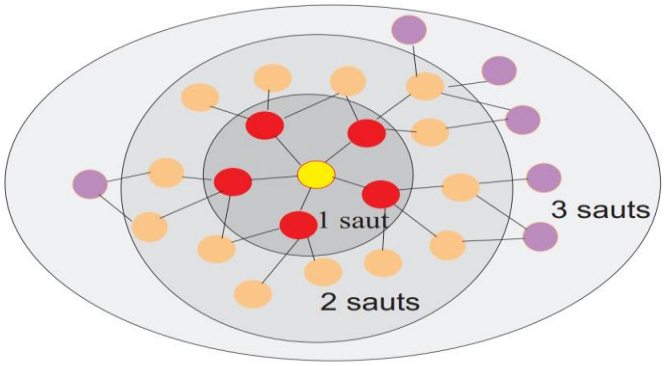


Figure 3.5. Le principe des zones en fonction du nombre de sauts dans le protocole FSR

3.5.1.5.Le protocole DSDV

Le protocole DSDV (Dynamic destination-Sequenced Distance-Vector) [78] est basé sur l'algorithme distribué de Bellman-Ford qui utilise les vecteurs de distance. Dans ce cas, la métrique utilisée est le nombre de sauts. La Figure 3.6 montre un exemple de réseau ayant comme métrique le nombre de sauts. Les routes choisies sont celles présentant le moins de sauts. Chaque nœud du réseau maintient une table de routage à jour comportant tous les nœuds du réseau joignables, le nœud intermédiaire suivant sur la route, le nombre de sauts, et le numéro de séquence de la destination.

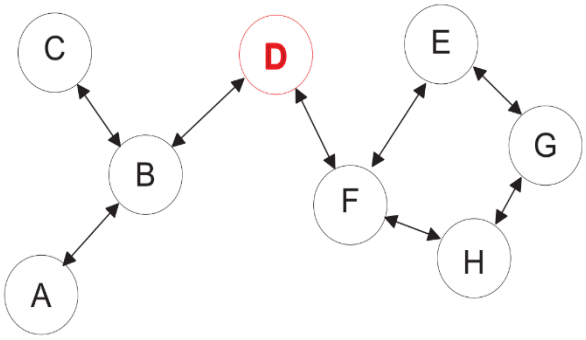


Figure 3.6. Un exemple de réseau utilisant le protocole DSDV d'après [78]

3.5.2. Protocoles de routage à la demande ou réactifs

Les protocoles de routage réactifs ne conservent aucune information de topologie et le processus de routage est lancé lorsqu'il est requis. Ici, la surcharge de routage est moindre, car ces catégories de protocoles n'échangent aucune information de routage. Le processus de routage est la combinaison des phases de découverte et de maintenance de route.

3.5.2.1. Le protocole DSR

Le protocole DSR (Dynamic Routing Source) [79] est un protocole réactif qui s'appuie sur deux sortes de paquets. Lorsqu'un noeud veut émettre un message, il fait une demande de route au moyen d'un paquet RREQ (Route REQuest) envoyé au destinataire. Ce paquet est propagé dans le réseau jusqu'à atteindre le destinataire, chaque station retransmettant le paquet modifié en inscrivant son adresse dans le champ actualisant ainsi la route prise par le paquet. Lorsque le destinataire reçoit le paquet RREQ, il répond à la source avec un paquet RREP (Route REPly) lui indiquant la route pour l'atteindre. La Figure 3.7 illustre le principe de découverte d'une route entre le noeud A et le noeud G. Il peut malgré tout y avoir des problèmes en cas de lien

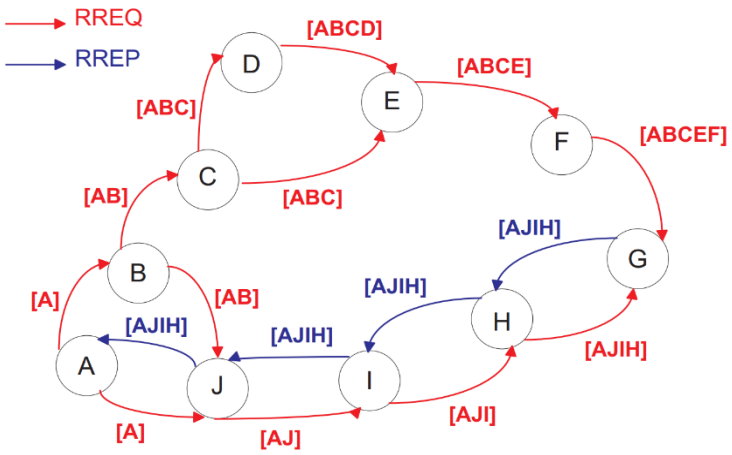


Figure 3.7. Le principe de découverte de route dans le protocole DSR.

asymétrique et des problèmes d'irrégularités au niveau des ondes électromagnétiques, la route de retour n'étant pas forcement possible.

Il existe également une procédure de maintien des routes permettant à chaque nœud découvrant une erreur d'émettre un message. Les routes découvertes sont gardées en mémoire par les stations afin d'avoir une meilleure réactivité, mais au détriment de la validité des routes dans le cas de nœuds très mobiles.

3.5.2.2. Le protocole AODV

Le protocole AODV (Ad hoc On-demand Distance Vector routing) [80] reprend les principes du protocole DSR pour la recherche des routes, mais afin de limiter le trafic, il diminue la taille des paquets en n'incluant pas toute la route au niveau de ceux-ci. Ce sont les nœuds intermédiaires qui stockent les routes au niveau de la table de routage. Lors de la réponse à une demande de route, le paquet utilise ces tables pour revenir à la source de la demande. Ces tables sont en cache et peuvent accélérer ainsi la découverte d'une route. Des messages "Hello" sont également utilisés afin de connaître la validité des liens.

3.5.3. Protocoles de routage hybride

Cette catégorie de protocoles de routage utilise les meilleures fonctionnalités des protocoles de routage réactif et proactif. Dans le réseau, le groupe de nœuds d'une région donnée est appelé zone. Le processus de routage dans la zone est proactif et en dehors de la zone est réactif.

3.5.3.1. Le protocole ZRP

Le protocole ZRP (Zone Routing Protocol) [81] utilise en fait deux protocoles de routage, un proactif et un réactif. Une taille de zone en nombre de sauts est dénie, par exemple 2, comme sur la Figure 3.8. Les nœuds présents dans la zone A à 2 sauts sont gérés suivant un protocole proactif : le protocole IARP (IntrAzone Routing Protocol). Les paquets de contrôle possèdent une durée de vie en nombre de sauts ; lorsqu'un nœud reçoit un paquet de contrôle, il actualise sa table de routage et il retransmet le paquet en décrémentant la durée de vie du paquet. Lorsque la durée de vie du paquet de contrôle est nulle, la bordure de zone est atteinte et le paquet n'est plus retransmis. Les noeuds hors de la zone A sont atteints grâce à un protocole réactif : le protocole IERP (IntErzone Routing Protocol) [Haas 02b]. Un troisième protocole gère les transitions entre les deux précédents : le protocole BRP (Border Resolution Protocol) [82] Le fait de considérer différentes zones dans le réseau est très intéressant, car un nœud peut ainsi

adapter son comportement en fonction de la distance qui le sépare de ses voisins. Un nœud a plus d'interactions avec ses voisins proches qu'avec ceux plus éloignés. En revanche, dans le cas de ZRP, la mise en œuvre au niveau du nœud est plus complexe que pour les autres protocoles : un nœud exécute en fait trois protocoles de routage et cela peut être préjudiciable à ses ressources CPU et mémoire.

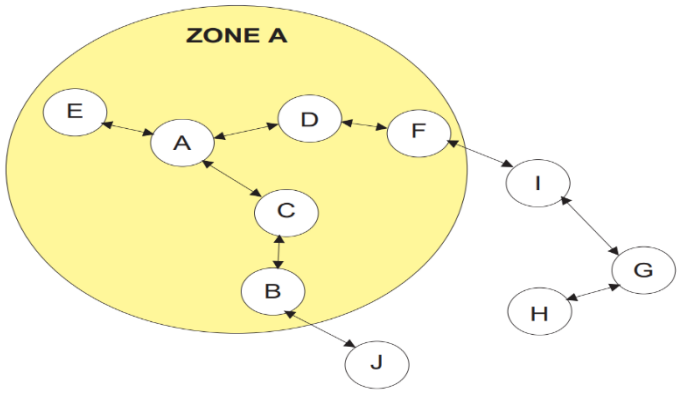


Figure 3.8. Le principe des zones dans le protocole ZRP

Le protocole CBRP Comme son nom l'indique, le protocole CBRP (Cluster Based Routing Protocol) est basé sur la création de groupes (clusters) au sein du réseau [83]. Les nœuds peuvent se voir attribuer des rôles particuliers au sein du réseau grâce à l'échange de message "Hello". Un nœud peut être élu chef de groupe (cluster-head), ou passerelle (gateway) entre groupes (voir la Figure 3.9.) suivant sa situation dans le réseau, principalement sa visibilité des autres nœuds. Les routes sont établies à la demande, comme pour un protocole réactif mais uniquement par les chefs de groupe : si un nœud veut envoyer un message, il demande à son responsable de groupe de lui déterminer la route à suivre. La route trouvée est agrégée au fur et à mesure du message de découverte et le nœud destinataire connaît ainsi le chemin de retour.

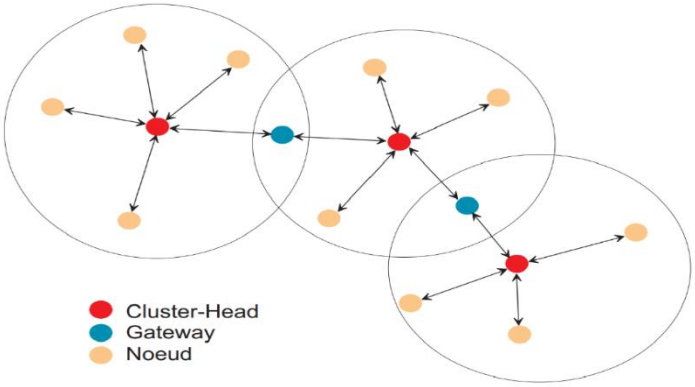


Figure 3.9. Les différents types de nœuds dans le protocole CBRP

L'idée de cluster est pertinente pour les réseaux agri-environnementaux car les différents nœuds peuvent être rassemblés suivant leur fonction ou leur situation géographique (les capteurs d'une

parcelle par exemple). Les passerelles entre clusters doivent être choisies judicieusement car ce rôle est gourmand en ressources et les nœuds assurant cette fonction doivent être correctement dimensionnés.

3.6. Qualité de service (QoS) dans les MANET

La qualité de service dans les MANET est le niveau de service de performance fourni par le réseau à l'utilisateur. L'objectif de la qualité de service est de faire en sorte que le comportement réseau déterministe porte les informations avec le niveau de précision attendu. Les applications multimédia, réelles et sensibles aux erreurs nécessitent une garantie de qualité de service. Atteindre la qualité de service dans les MANET est un problème difficile en raison des changements dynamiques de la topologie du réseau, du manque d'administration centralisée et des ressources disponibles limitées. Ces problèmes ont compliqué le provisionnement de la qualité de service par rapport aux réseaux câblés de contrepartie. Dans la fourniture de la qualité de service, le protocole de routage joue un rôle majeur [84, 85] puisqu'un protocole de routage doit identifier les nœuds intermédiaires potentiels en fonction des besoins de l'application.

Dans les réseaux mobiles ad hoc, la fourniture de la qualité de service tient compte des considérations de conception suivantes.

3.6.1. Réservation de ressources d'état ferme contre d'état matériel

Dans la fourniture de la qualité de service, la réservation de ressources est un mécanisme clé; il réserve les ressources requises sur chaque nœud pour fournir le niveau requis de qualité de service à l'application. Dans les MANET, la réservation de ressources pour la fourniture de la qualité de service est de deux types, à savoir l'état dur et la réservation à l'état souple. A l'état dur, les ressources sont réservées et maintenues au niveau des nœuds jusqu'à la fin de l'application. Dans l'approche de réservation de ressources d'état souple, les ressources sont réservées pour une période limitée ; c'est-à-dire que tous les paquets d'un même flux sont reçus au nœud cible. La période est définie en termes d'heure d'arrivée des paquets. Les ressources sont libérées si aucun paquet de données n'est arrivé dans la période donnée. La méthode de désallocation est utilisée pour libérer les ressources capturées.

3.6.2. Approche par état contre apatride

Dans le procédé avec état, les informations d'état de la topologie du réseau et les informations relatives au flux sont conservées sur chaque nœud. Mais dans la méthode sans état, les nœuds

ne sont pas tenus de conserver de telles informations. Les informations d'état sont de deux types, à savoir les informations d'état locales et les informations d'état globales. Les informations d'état global prennent en charge l'algorithme de routage centralisé, les informations d'état locales prenant en charge les algorithmes de routage distribué. L'approche avec état nécessite un contrôle absolu lors de la collecte et de la maintenance des informations. Cette surcharge est réduite en approche sans état, mais l'assurance de la qualité de service est une tâche difficile.

3.6.3. Approche QoS dure vs QoS souple

La fourniture de la qualité de service est de deux types, à savoir la qualité de service stricte et la qualité de service souple. Dans la fourniture de QoS stricte, les exigences de QoS doivent être satisfaites jusqu'à la fin de l'application. Dans la fourniture de QoS souple, les exigences de QoS ne doivent pas être satisfaites tout au long de la session. En raison de la nature dynamique des réseaux Adhoc, la fourniture d'une qualité de service irréprochable est une tâche ardue, mais peut être soumise à certaines limites.

3.7. Diagnostic et confiance dans les MANET

Dans les réseaux Mobile Adhoc, un nœud doit rejoindre et travailler avec d'autres nœuds sans aucune interaction préalable. Par conséquent, la gestion de la confiance dans les MANET [86, 87] est certainement un avantage pour exécuter des applications en permettant uniquement des nœuds de confiance. Dans les MANET, la gestion de la confiance est la combinaison des phases de calcul, de mise à jour de confiance, de propagation de confiance et de révocation de confiance [88, 89]. En raison de la nature dynamique et distribuée des MANET, la maintenance de la confiance est une tâche ardue. Les contraintes de ressources d'un nœud limitent les informations de confiance à une petite région du réseau. En raison des changements de réseau dynamiques dans les MANET, la confiance peut être considérée comme une variable aléatoire continue, qui évolue dans le temps.

3.7.1 Confiance dans les communications et la mise en réseau

Les concepts de confiance sont également applicables aux réseaux de communication. La confiance est en train d’être identifiée comme l’une des considérations de conception pour les protocoles réseau nouvellement conçus. Pour les réseaux distribués, des relations de confiance entre les éléments sont essentielles pour maximiser les objectifs de l'application, tels que l'évolutivité et la fiabilité. Dans [90], une valeur de confiance de nœud est représentée en termes

de relations avec les autres nœuds du réseau. Ces relations sont établies en fonction des interactions présentes et antérieures. Dans les interactions, chaque élément de preuve positif augmente et un élément de preuve négatif diminue les valeurs de confiance entre les nœuds. Li et Singhal [91] définissent la valeur de confiance du nœud comme étant sa capacité à exécuter des applications avec fiabilité, sûreté de fonctionnement et sécurité.

Dans [92], la composition de service basée sur la confiance dans les MANET est proposée. Où les demandeurs de services qui demandent des services et les fournisseurs qui fournissent des services. Si plusieurs services sont demandés, la composition du service doit être effectuée par plusieurs fournisseurs de services. La méthode proposée trouve le chemin de composition de service fiable pour minimiser l'énergie et la durée totales.

Naimi [93] Introduire un algorithme qui prédit les valeurs métriques quelques secondes à l'avance, afin de compenser le retard impliqué par la mesure de la qualité de la liaison et sa diffusion par le protocole de routage dans un réseau ad hoc.

Dans [94] a proposé un protocole de routage EFMMRP basé sur un raisonnement flou. Dans ce mécanisme, toutes les métriques de réseau des routes sont converties en une seule métrique. Le choix du chemin est basé sur la valeur obtenue par chaque lien, le chemin optimal est celui qui a un coût fuzzy minimum

Kaliappan [95] a utilisé des algorithmes génétiques dynamiques tels que l'algorithme génétique d'immigrants (EIGA) et l'algorithme génétique à mémoire améliorée (MEGA) basés sur l'élitisme pour résoudre l'enracinement dans l'équilibrage de charge dans un réseau ad hoc mobile.

Clausen [96] Étudie le protocole de routage Prise en charge du protocole de routage vectoriel distance-vecteur ad hoc à la demande Lightweight à la demande (LOADng) pour les demandes de routage intelligentes et la recherche en anneau étendue, conçue pour permettre un routage efficace, évolutif et sécurisé dans des réseaux à faible consommation et avec pertes.

3.8. Conclusion

Ce chapitre a abordé les déférents définitions et caractéristiques des réseaux mobiles Ad hoc, ainsi que les protocoles de routage dans MANET, où les solutions existantes pour y résister sont très limitées. Bien que divers travaux de recherche effectués dans le passé aient apporté une contribution majeure, les problèmes d'efficacité de routage et de surveillance de système

de communication restent toutefois non résolus dans de nombreux cas. L'objectif principal de la recherche proposée est de combler les lacunes identifiées dans l'étude. Le prochain chapitre sera une étude théorique sur les outils utilisé dans la modélisation de notre méthode de diagnostic et de routage des systèmes de communication mobile Ad hoc MANET.

Chapitre 4

Les réseaux de Petri pour la modélisation et la surveillance

Résumé

Dans ce chapitre, nous rappelons le formalisme des réseaux de Petri utilisés dans la suite de la thèse.

4. Modèle de réseau de Petri

Les réseaux de Petri (PN) sont un modèle de système à événements discrets développé au début des années 1960 par C.A. Petri dans sa thèse de doctorat [97]. Ils représentent l'une des méthodes les plus efficaces pour analyser des systèmes à événements discrets. Les RdP peuvent être divisés en temps discret et programmé. De plus, les RdP programmés sont divisés en déterministe et stochastique. Dans cette thèse, nous traitons de RdP discrets, c'est un modèle logique qui nous permet de représenter l'ordre des événements d'occurrence, mais pas leur temporisation.

Les raisons pour lesquelles les RdP sont si largement utilisés sont multiples.

- Les RdP sont un formalisme mathématique et graphique permettant de modéliser des systèmes à événements discrets.
- Les RdP donnent une représentation compacte de l'espace d'états. En fait, ils ne nécessitent pas de représenter explicitement tous les états accessibles, mais uniquement les règles d'évolution.
- Les RdP peuvent représenter un système à événements discrets avec un nombre infini d'états via un graphe avec un nombre fini de nœuds.
- Les RdP peuvent représenter le concept de simultanéité.
- Les PN donnent une représentation modulaire. En fait, si un système est composé de plus un sous-système et ces sous-systèmes interagissent les uns avec les autres. Il est ensuite possible de représenter chaque sous-système en tant que RdP, puis, à l'aide de constructions spécifiques, de combiner les unités individuelles pour obtenir le modèle entier.

4.1.Définitions de base

Un réseau Place / Transition (réseau P / T) est une structure $N = (P, T, Pré, Post)$, où P est un ensemble de m places ; T est un ensemble de n transitions ; $Pré$: P ×T → ℕ et Post : P × T→ ℕ sont les fonctions d'incidence pré- et post- qui spécifient le poids des arcs dirigés d'un endroit à l'autre et d'un passage à l'autre ; C = Post - Pré est la matrice d'incidence.

Le préréglage et le post-set d'un nœud $X \in P \cup T$ sont notés ${}^\bullet X$ et $X^\bullet$ tandis que ${}^\bullet X^\bullet = {}^\bullet X \cup X^\bullet$.

Un marquage est un vecteur $M : P \longrightarrow \mathbb{N}$ qui attribue à chaque emplacement d'un réseau P / T un nombre entier non négatif de jetons, représentés par des points noirs. On note $M(p)$ le

marquage de place *p*. Un système P / T ou réseau $\langle N, M_0 \rangle$ est un réseau *N* avec un marquage initial M_0.

Une transition *t* est activée à *M* ssi $M \geq Pre(\cdot, t)$ et peut se déclencher pour donner le marquage $M' = M + C(\cdot, t) = M + C \cdot \vec{t}$ où $\cdot\ \vec{t} \in \mathbb{N}^n$ est un vecteur dont les composantes sont tous égaux à 0 sauf la composante associée à la transition *t* égale à 1.Soit une séquence $\sigma \in T^*$, où T^* est l'ensemble de toutes les séquences finies de transitions dans *T* incluant la chaîne vide ε, on écrit $M\ [\sigma\rangle$ pour indiquer que la séquence de transitions σ est activée en *M*, et nous écrivons $M\ [\sigma\rangle$ pour indiquer que le déclenchement de σ donne M'. Notez que dans cette thèse, nous supposons toujours que deux ou plusieurs transitions ne peuvent pas se déclencher simultanément (non hypothèse de concurrence).

Soit une séquence $\sigma \in T^*$ on appelle $\pi : T^* \rightarrow \mathbf{N}^n$ la fonction qui associe σ à un vecteur $y \in \mathbf{N}n$, nommé vecteur de déclenchement de σ. En particulier, $y = \pi(\sigma)$ est tel que $y\ (t) = k$ si la transition *t* est contenue *k* fois dans σ. On note $|\sigma|$ le nombre d'occurrences de la transition *t* dans la suite σ.

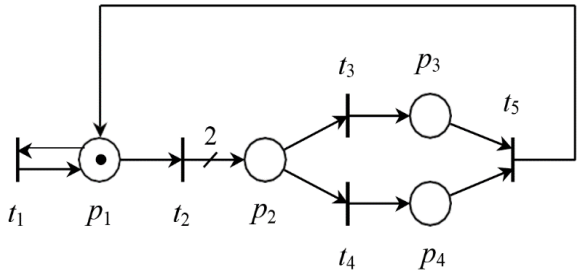


Figure 4.1. Exemple de réseau de Petri

Exemple 4.1. Considérons le système de RdP de la figure 4.1. L'ensemble des places $P = \{p_1, p_2, p_3, p_4\}$

et l'ensemble des transitions est $T = \{t_1, t_2, t_3, t_4, t_5\}$. Les deux matrices Pré et Post sont:

$$Pré = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad Post = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

et la matrice d'incidence est

$$Pré = \begin{bmatrix} 0 & -1 & 0 & 0 & 1 \\ 0 & 2 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Le marquage initial est $M_0 = [1\ 0\ 0\ 0]^T$. Les transitions t_1 et t_2 sont activées à M_0 et leur déclenchement donne respectivement les repères M_0 et $M_1 = [0\ 2\ 0\ 0]^T$. Le préréglage et le postset pour la transition t_2 et la place p_2 sont respectivement $\bullet t_2 = \{p_1\}, t_2^\bullet = \{p_2\}, \bullet p_2 = \{t_2\}$ et $p_2^\bullet = \{t_3, t_4\}$,

Considérons la séquence de tir $\sigma = t_1 t_1 t_2 t_3$ firable dans le RdP considéré à M_0. Le vecteur de déclenchement de σ est $\pi(\sigma) = [2\ 1\ 1\ 0\ 0]^T$, $|\sigma|t_1 = 2, |\sigma|t_2 = |\sigma|t_3 = 1$ et $|\sigma|t_4 = |\sigma|t_5 = 0$.

Un marquage M est accessible en $\langle N , M_0 \rangle$ ssi il existe une séquence de déclenchement σ telle que $M_0 [\sigma\rangle M$. Dans ce cas, l'équation d'état $M = M_0 + C \cdot y$ est vérifiée, où $y = \pi(\sigma)$. L'ensemble de tous les marquages pouvant être atteints à partir de M_0 définit l'ensemble d'accessibilité de $\langle N, M_0 \rangle$ et est noté $R\ (N, M_0)$. Enfin, on note $PR\ (N , M_0)$ l'ensemble potentiellement accessible, c'est-à-dire l'ensemble des marquages $M \in \mathbf{N}^m$ pour lesquels il existe un vecteur $y \in \mathbf{N}^n$ qui vérifie l'équation d'état $M = M_0 + C \cdot y$; C à D, $PR(N, M_0) = \{M \in \mathbf{N}^m \mid \exists\, y \in \mathbf{N}^n : M = M_0 + C \cdot y\}$. Il soutient que $R(N, M_0) \subseteq PR(N, M_0)$.

Un RdP n'ayant pas de circuit dirigé est appelé acyclique. Pour cette sous-classe, le résultat suivant est valable.

Théorème 3.2. [98]Soit N un PN acyclique.

(i) Si le vecteur $y \in \mathbf{N}^n$ vérifie l'équation $M_0 + C \cdot y \geq \vec{0}$, il existe une séquence de déclenchement firable de M_0 dont le vecteur de déclenchement est $\pi(\sigma) = y$.

(ii) Un repère M est accessible à partir de M_0 ssi il existe une solution entière non négative y satisfaisant l'équation d'état $M = M_0 + C \cdot y$, c'est-à-dire, R (N, M_0) = PR (N, M0).

4.2. Langage réseaux

Soit un système de RdP $\langle N, M_0 \rangle$, nous définissons son langage libre comme l'ensemble de ses séquences de tir

$$L(N , M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}.$$

Nous définissons également l'ensemble des séquences de tir de longueur inférieure ou égale à $k \in \mathbf{N}$ comme suit :

$L_k(N,M_0) = \{\sigma \in L\ (N,M_0) \mid |\sigma| \leq k\}$.

On donne enfin un langage $L \subset T^*$ et un vecteur $y \in \mathbf{N}^n$

$L(y) = \{\sigma \in L \mid \pi(\sigma) = y\}$

l'ensemble de toutes les séquences en L dont le vecteur de déclenchement est *y*.

4.3. Règles de production floues (RPF) et les réseaux de Petri floue RdPF

4.3.1. Règles de production floues

Les règles de production floues ont été largement utilisées pour représenter, capturer et stocker de vagues connaissances d'experts dans des systèmes décisionnels. Chaque règle est généralement exprimée sous la forme d'une règle floue si-alors, dans laquelle l'antécédent et le consécutif sont des termes flous exprimés par des ensembles flous. Si un RPF comprend des connecteurs AND ou OR, il est appelé règles de production floues composite ou composé [99].

Pour améliorer les capacités de représentation et de raisonnement des règles de production floues, le paramètre de pondération [100, 101] a été intégré aux règles flou Si-Alors, obtenant les règles de production floues pondérés (RPFP). Soit *R* un ensemble de RPFP, c'est-à-dire, $R = \{R_1, R_2, \ldots, R_n\}$ la forme de la i^{eme} règle peut être présentée comme suit:

R_i: Si *a* Alors *c* (CF = μ), *Th*, *w*

où *a* et *c* sont respectivement les parties antécédentes et conséquentes de la règle, qui comprennent une ou plusieurs propositions avec des variables floues. Le paramètre μ ($\mu \in [0, 1]$) est le facteur de certitude indiquant la force de conviction de la règle, $Th = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ est un ensemble de valeurs de seuil spécifiées pour chacune des propositions dans l'antécédent, et $w = \{w_1, w_2, \ldots, w_m\}$ est un ensemble de pondérations attribuées à toutes les propositions de l'antécédent, montrant l'importance relative de chaque proposition de l'antécédent contribuant à la suite.

En général, les RPF peuvent être divisés en cinq types énumérés ci-dessous [102]:

Type 1. Une règle de production floue simple pondérée

R: Si *a* Alors *c* (μ; λ; *w*)

Type 2. Une règle conjonctive floue pondérée composite dans l'antécédent

*R:*Si a_1 ET a_2 ET…ET a_m Alors c (μ; λ_1, λ_2, ...,λ_m; w_1, w_2, ...,w_m)

Type 3. Une règle conjonctive floue pondérée composite dans la suite

R: Si a Alors c_1 ET c_2 ET…ET c_m (μ; λ; w)

Type 4. Une règle disjonctive floue pondérée composite dans l'antécédent

R: Si a_1 OU a_2 OU…OU a_m Alors c (μ; λ_1, λ_2, ...,λ_m; w_1, w_2, ...,w_m)

Type 5. Une règle disjonctive floue pondérée composite dans la suite

R: Si a Alors c_1 OU c_2 OU…OU c_m (μ; λ; w).

Dans de nombreuses applications pratiques, les règles des types 4 et 5 ne sont pas autorisées dans une base de connaissances, car elles peuvent être transférées dans plusieurs règles de type 1. Les règles suivantes constituent plusieurs exemples typiques de règles RPFP:

R1: SI il les liens sont occupé ALORS que bande passante est basse ($\mu = 0,9$);

R2: SI énergie est grand ET durée de vie de lien est grand ALORS il est fiable ($\mu = 1,0$);

R3: SI la surveillance est élevée ET que le système est fiable ET que système est normale ALORS la stabilité ($\mu = 0,8$);

R4: le semi-conducteur régulateur IF est cassé ALORS l'excitateur n'est pas suffisant ($\mu = 0,9$; $\lambda = 0,2$; $w = 1,0$);

R5: la fréquence SI est supérieure à la valeur normale ET la fréquence double est inférieure à la valeur normale ET l'amplitude change évidemment les charges changent ALORS que le rotor est en flexion à chaud ($\mu = 0,9$; $\lambda_1 = 0,3$, $\lambda_2 = 0,3$, $\lambda_3 = 0,2$; $w_1 = 0,5$, $w_2 = 0,3$, $w_3 = 0,2$)

4.4. Les Réseau de Petri floue RdPF

Pour traiter de l'incertitude dans la représentation et le raisonnement de la connaissance, les RdPF ont été développés à partir de la théorie de RdP, où les jetons représentant l'état des propositions sont marqués par une valeur de vérité comprise entre 0 et 1. En appliquant un formalisme de RdP à des systèmes à base de règles floues, il est capable de visualiser la structure d'un système expert et d'exprimer efficacement son comportement de logique de proposition dynamique. Par exemple, sur la figure 4.2 (b), nous avons

$P = \{p_1, p_2\}, T = \{t_1\}, I(t_1) = \{p_1\}, O(t_1) = \{p_2\}, f(t_1) = \mu_1, \alpha(p_1) = \alpha_1$, et $\alpha(p_2) = 0$.

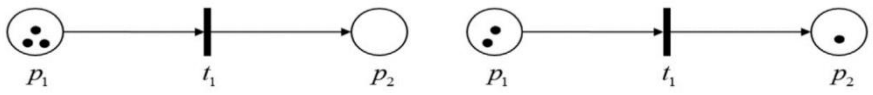
Pour un RdPF, une transition est dite activée si tous ses emplacements d'entrée sont marqués par un jeton et si sa valeur réelle est supérieure ou égale à une valeur seuil. Le processus de raisonnement d'un RdPF est exécuté en activant les règles et en mettant à jour le vecteur de degré de vérité à chaque étape de raisonnement. En raison des caractéristiques des systèmes basés sur des règles floues, les principales différences entre les RdP et les RdPF sont les suivantes :

(1) Dans les RdPF, le nombre de jetons dans un lieu ne peut pas être supérieur à un, car un jeton est associé à une valeur de vérité comprise entre 0 et 1. Un jeton ne représente pas un «objet», alors qu'il est probable qu'il le soit dans les RdP.

(2) Les RdPF sont toujours des réseaux sans conflit, car il n'y a pas de concept de «ressource» dans les RdPF et une proposition peut être partagée par différentes règles en même temps. Par exemple, sur la Figure 4.3., la proposition d_3 est partagée par deux règles R_1 et R_2, qui peuvent utiliser la proposition d_3 simultanément et raisonner en parallèle.

(3) Les jetons ne sont pas supprimés des emplacements d'entrée d'une transition après son déclenchement, car l'évaluation des règles implique uniquement la propagation de la vérité des propositions. C'est-à-dire que la partie antécédente reste vérifiée bien que sa partie résultante puisse déjà être prouvée dans le raisonnement fondé sur la connaissance.

(a) un réseau de Petri



(b) un réseau de Petri floue

avant le tir de transition — après le tir de transition

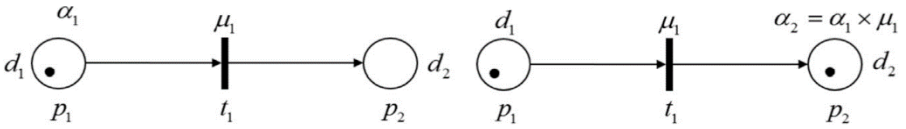


Figure 4.2. Une illustration de PN et FPN.

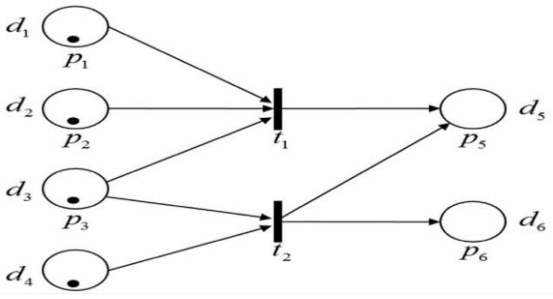


Figure 4.3. Un exemple de FPN avec une proposition partagée.

4.4.1. Définitions des Réseaux de Petri Floue

En 1988, Looney [103] a lancé le concept de RdPF pour représenter les RPF d'un système de prise de décision basé sur des règles. Dans un travail ultérieur, [80]ont proposé un modèle plus générique de RdPF pour modéliser la représentation des connaissances et décrit un algorithme flou pour effectuer automatiquement le raisonnement des connaissances.

Selon [104], une structure de RPF est définie comme un 8 -uple:

$$\text{RdPF} = (P, T, D, I, O, f, \alpha, \beta) \qquad (4.1)$$

où

$P = \{p_1, p_2, \ldots, p_m\}$ est un ensemble fini de places,
$T = \{t_1, t_2, \ldots, t_n\}$ est un ensemble fini de transitions,
$D = \{d_1, d_2, \ldots, d_m\}$ est un ensemble fini de propositions avec $P \cap T \cap D = \emptyset$, $|P| = |D|$,
I: $T \rightarrow P^{\infty}$ désigne la fonction d'entrée, une cartographie des transitions vers les places,
O: $T \rightarrow P^{\infty}$ désigne la fonction de sortie, un mappage de transitions vers les places,
f: $T \rightarrow [0, 1]$ désigne une fonction d'association, une transposition de transitions en valeurs réelles comprises entre 0 et 1,
α: $P \rightarrow [0, 1]$ est une fonction d'association, un mappage d'emplacements à des valeurs réelles comprises entre 0 et 1,
β: $P \rightarrow D$ est une fonction d'association, une cartographie bijective entre des places et des propositions.

4.5. Applications des Réseaux de Petri Floue

En raison de la représentation graphique et de la capacité de traitement dynamique, les réseaux RdPF ont été largement utilisés pour résoudre divers problèmes d'ingénierie au cours des dernières décennies. Par conséquent, les applications pratiques des RdPF ont fait l'objet d'une étude approfondie. Nous décrivons ci-après les résultats en détail.

En raison du grand nombre d'utilisations de cet outil, nous ne mentionnerons que brièvement les travaux liés au domaine des télécommunications et nous évoquerons également les travaux de diagnostic dans les domaines des télécommunications et de l'industrie.

4.5.1. Réseaux de capteurs sans fil

Pour augmenter la fiabilité lors de la sélection du routage, [105]ont proposé un algorithme de routage fiable dans les réseaux ad hoc mobiles (MANET), basé sur les RdPF et leur mécanisme de raisonnement. L'algorithme permet la représentation structurée de la topologie du réseau et

peut calculer l'itinéraire le plus fiable en comparant le degré de fiabilité du routage. Dans [106], les auteurs ont présenté un algorithme de routage à plusieurs niveaux fiable et économe en énergie pour les réseaux de capteurs sans fil utilisant des RdPF. L'algorithme a pris en compte l'énergie résiduelle, le nombre de voisins et la centralité de chaque nœud pour la formation de grappes, ce qui non seulement équilibre la charge énergétique de chaque nœud, mais fournit également une fiabilité globale pour l'ensemble du réseau.

Dans [107] les auteurs ont proposé un modèle, à savoir FuzzyWMN, qui utilise des RdPF pour réaliser l'adaptation de trafic dans des réseaux maillés sans fil caractérisés par une incertitude et une imprécision des informations. Sur la base du vecteur de distance à la demande sécurisé ad hoc (SAODV), [108]ont utilisé des FPN pour proposer un protocole de routage sécurisé dans MANET, appelé FPN-SAODV. Dans le protocole de routage FPN-SAODV, un type de vérification de sécurité floue bidirectionnelle de nœud à nœud a été effectué pour l'envoi et la réception de paquets entre chaque paire de nœuds. Une vérification de la sécurité de l'itinéraire direct a été utilisée pour sélectionner l'itinéraire le plus sûr parmi les chemins candidats à travers les sources jusqu'à la destination. Dans [109], les auteurs se sont concentrés sur une approche d'inférence de connaissances dynamique utilisant des RdPF pour découvrir le meilleur chemin de routage pour les protocoles de routage multidiffusion dans un environnement à bande passante élevée. Les travaux menés dans [110] ont mis en évidence un mécanisme de routage basé sur la confiance permettant de se défendre contre les attaques dans les plans de données et de routage dans un MANET basé sur le routage d'état de liaison optimisé (OLSR), dans lequel un modèle de raisonnement de confiance basé sur les FPN est utilisé. Utilisé pour évaluer les valeurs de confiance des nœuds mobiles et éviter les nœuds malveillants ou compromis, et un algorithme de routage basé sur la confiance est utilisé pour sélectionner un chemin avec la valeur maximale de confiance du chemin parmi tous les chemins possibles.

4.5.2. Diagnostic de défaut et évaluation des risques

Les auteurs dans [111] ont utilisé les RdPF comme technique de modélisation pour construire des modèles de diagnostic de pannes de systèmes de commande, qui visent à diagnostiquer avec précision les pannes lorsqu'une information d'alarme incomplète et incertaine est détectée pour les relais de protection et les disjoncteurs. [112] ont mis en œuvre les réseaux FRPN pour traiter la complexité de l'estimation de la section de pannes des systèmes électriques et ont abordé plusieurs problèmes clés, notamment la conception optimale de la structure des modèles de diagnostic afin d'éviter une taille de matrice importante, l'utilisation de paramètres de

algorithme d'exécution de la matrice pour obtenir une capacité de raisonnement parallèle et intégration de données d'entrée plus fiables pour améliorer la précision de l'estimation. Lui et al [113] ont exposé un modèle de raisonnement de diagnostic de panne dynamique basé sur les RdPF pour résoudre le problème complexe d'estimation de section de panne du système d'alimentation, dans lequel les poids dans le raisonnement flou sont déterminés par les informations d'alarme incomplètes et incertaines des relais de protection et des disjoncteurs.

Dans [114], les auteurs ont étudié la contrainte temporelle entre les occurrences d'événements dans les systèmes d'alimentation et ont introduit une approche RdPF de raisonnement temporel (RdPFT) pour le diagnostic des défaillances. [115] les auteurs ont présenté une méthode de diagnostic des pannes basée sur les RdPF tenant compte de la fonctionnalité de service des dispositifs sources d'information et l'ont appliquée pour diagnostiquer les défauts des dispositifs du système d'alimentation. Pour détecter efficacement les consommations frauduleuses et anormales, [116] ont introduit une approche de la connaissance pour réaliser le plan de rétablissement du service en ligne des systèmes de distribution. Dans leur étude, un modèle de RdPF a été construit pour représenter le schéma de connaissances et d'inférences sur le rétablissement du service et a été testé sur un système de distribution pratique de la Taiwan Power Company. Pour faire face à l'impact des anomalies des panneaux solaires, Wu et al. [117] ont établi un modèle utilisant l'analyse d'arbre de défaillance (AAD) et les RdPF pour effectuer une analyse de fiabilité d'un système mécanique de générateur solaire, qui peut être utilisé pour rechercher les causes fondamentales les plus importantes et proposer des solutions pour améliorer la fiabilité du générateur solaire. Wu et al. [118, 119] ont également mis au point une approche de répartition de la fiabilité associant une évaluation complète floue à des FRPN afin de réaliser la répartition de la fiabilité des panneaux solaires d'engins spatiaux. Dans [95], les auteurs ont exploré une approche FPN en temps réel (RTFPN) permettant de diagnostiquer les défaillances progressives dans les systèmes de fabrication discrets basés sur des automates programmables. Dans cette approche, un modèle PN en temps réel (RTPN) a été utilisé pour surveiller l'état de fonctionnement de l'installation de fabrication et le diagnostiqueur RdPF a été utilisé pour isoler les causes premières de l'erreur lorsqu'une erreur se produit. An et Liang [120] ont proposé un cadre RdPF avec des transitions non observables pour résoudre le problème de diagnostic de pannes des systèmes à événements discrets avec une imprécision et des événements non observables tels que le propulseur Hall. Une stratégie de raisonnement bidirectionnel a été proposée pour calculer certaines valeurs factorielles des résultats de

diagnostic, qui sont un raisonnement en avant et un raisonnement en arrière. Liu et al. [120]ont présenté une approche de diagnostic des fautes et d'analyse des causes basée sur l'approche FER et les DAFPN, capable de capturer tous les types d'informations sur les événements anormaux fournies par les experts, d'identifier les causes premières et de déterminer les conséquences des événements anormaux identifiés en combinant raisonnement en avant et en arrière.

4.6. Réseaux de Petri Synchronisés Flous (RPSyncF)

Nous proposons une variante de réseau de Petri flou inspirée des travaux de Chen [4], adapté au besoin de la modélisation des fonctions de surveillance. Le modèle de Chen est adapté à la modélisation de bases d'informations logiques statiques. Cette approche n'est pas entièrement satisfaisante pour la surveillance dynamique. C'est dans ce sens que nous proposons le réseau de Petri Petri synchronisé flou (RdPSyncF) comme une extension des RdPF. Pour modéliser les fonctions de détection/diagnostic nous définissons un modèle flou capable d'intégrer l'instant d'apparition des défauts du système surveillé.

Les réseaux de Petri synchronisé flou (RdPSyncF) est un outil défini comme étant le n-uplet

RdPSynF **=** < ***P, T, E, I, O, F, Sinc, D, M₀***> avec :

$\boldsymbol{P} = \{p1, p2, \ldots . pn\}$ ensemble fini des places ;

$\boldsymbol{T} = \{t1, t2, \ldots . tn\}$ ensemble fini des transitions;

$\boldsymbol{E} = \{E1, E2, \ldots . En\}$ ensemble fini de évènements externes;

$\boldsymbol{I}: T \rightarrow P$ fonction d'entrée dans les places ;

$\boldsymbol{O}: P \rightarrow T$ fonction de sortie des places ;

F(t) : T **[0,1]** fonction associative qui établi une valeur de crédibilité *F(t)* variable dans le temps pour chaque transition t_i *T*. μ représente le degré de vérité de la proposition correspondant à la transition. L'instant *t* correspond à l'instant temporel *t d* quand l'événement externe E_i sera réceptionné par le système modélisé.

$\boldsymbol{Sync}: \boldsymbol{T} \rightarrow \boldsymbol{E} \cup \boldsymbol{e}$ est une application sur l'ensemble des transitions avec des valeurs sur l'ensemble des événements *E* réuni avec l'événement *e* qui est l'événement avec apparition permanente

D= {d₁, d₂, … dₙ} ensemble de durées associées aux événements externes qui représente la fenêtre temporelle pour leurs réceptions. Ces durées représentent le même temps que les temporisations associées aux transitions synchronisées avec des événements externes.

Chapitre 5

Les Ant-systems et le routage dans les MANET

Résumé

Dans ce chapitre, nous présenterons une brève introduction au monde des fourmis, ensuite nous décrirons en détail chacun des modèles de fourmis artificielles ainsi que les différents algorithmes qui lui sont associés.

5.1. Introduction

Les colonies de fourmis dans la nature montrent une grande capacité à trouver des chemins entre des sources de nourriture et la fourmilière. A l'échelle d'une fourmi, le problème résolu est complexe : la fourmi n'a qu'une vision très locale de son environnement, et elle n'utilise pas de carte ni de GPS. L'intelligence qui permet la résolution du problème est collective et est en fait le résultat émergeant d'un grand nombre d'interactions élémentaires.

Ce mécanisme de résolution collective de problèmes a inspiré une méta-heuristique, c'est-à-dire une méthode générique de résolution de problèmes, appelée « optimisation par colonies de fourmis ».

Dans la suite, nous présenterons une brève introduction au monde des fourmis, ensuite nous décrirons en détail chacun des algorithmes artificielles bases sur les Ant system.

5.2. Généralités sur les fourmis

5.2.1. Les pistes de phéromones

En marchant du nid à la source de nourriture et vice-versa (ce qui dans un premier temps se fait essentiellement de façon aléatoire), les fourmis déposent au passage sur le sol une substance odorante appelée « phéromones ». Cette substance permet ainsi donc de créer une piste chimique, sur laquelle les fourmis s'y retrouvent. En effet, d'autres fourmis peuvent détecter les phéromones grâce à des capteurs sur leurs antennes.

Les phéromones ont un rôle de marqueur de chemin, quand les fourmis choisissent leur chemin, elles ont tendance à choisir la piste qui porte la plus forte concentration de phéromones. Cela leur permet de retrouver le chemin vers leur nid lors du retour. D'autre part, les odeurs peuvent être utilisées par les autres fourmis pour retrouver les sources de nourritures trouvées par leurs congénères.

Ce comportement permet de trouver le chemin le plus court vers la nourriture lorsque les pistes de phéromones sont utilisées par la colonie entière. Autrement dit, lorsque plusieurs chemins marqués sont à la disposition d'une fourmi, cette dernière peut connaitre le chemin le plus court vers sa destination.

5.3. Principe des Ant System

L'idée originale provient de l'observation de l'exploitation des ressources alimentaires chez les fourmis. En effet, celles-ci, bien qu'ayant individuellement des capacités cognitives limitées,

sont capables collectivement de trouver le chemin le plus court entre une source de nourriture et leur nid (voir figure 5.1).

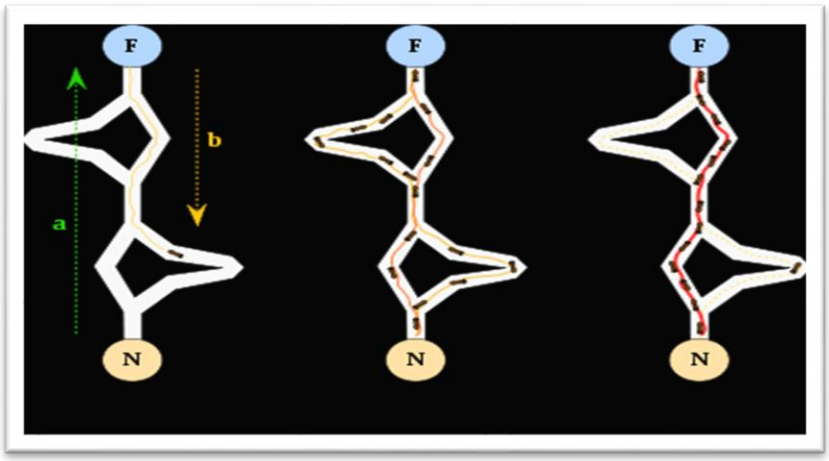


Figure 5.1. Raisonnement des Ant System

1) la première fourmi trouve la source de nourriture (F), via un chemin quelconque (a), puis revient au nid (N) en laissant derrière elle une piste de phéromone (b).

2) les fourmis empruntent indifféremment les quatre chemins possibles, mais le renforcement de la piste rend plus attractif le chemin le plus court.

3) les fourmis empruntent le chemin le plus court, les portions longues des autres chemins perdent leur piste de phéromones.

5.4. Méthode de fonctionnement

Des biologistes ont ainsi observé, dans une série d'expériences menées à partir de 1989, qu'une colonie de fourmis ayant le choix entre deux chemins d'inégale longueur menant à une source de nourriture avait tendance à utiliser le chemin le plus court.

Un modèle expliquant ce comportement est le suivant :

1. Une fourmi (appelée « éclaireuse ») parcourt plus ou moins au hasard l'environnement autour de la colonie ;
2. Si celle-ci découvre une source de nourriture, elle rentre plus ou moins directement au nid, en laissant sur son chemin une piste de phéromones
3. Ces phéromones étant attractives, les fourmis passant à proximité vont avoir tendance à suivre, de façon plus ou moins directe cette piste.

4. En revenant au nid, ces mêmes fourmis vont renforcer la piste.
5. Si deux pistes sont possibles pour atteindre la même source de nourriture, celle étant la plus courte sera, dans le même temps, parcourue par plus de fourmis que la longue piste.
6. La piste courte sera donc de plus en plus renforcée, et donc de plus en plus attractive.
7. La longue piste finira par disparaître, les phéromones étant volatiles.
8. A terme, l'ensemble des fourmis a donc déterminé et choisi la piste la plus courte.

Les fourmis utilisent l'environnement comme support de communication : elles échangent indirectement de l'information en déposant des phéromones, le tout décrivant l'état de leur travail. L'information échangée a une portée locale, seule une fourmi située à l'endroit où les phéromones ont été déposées y a accès. Ce système porte le nom de « stigmergie », et se retrouve chez plusieurs animaux sociaux (il a notamment été étudié dans le cas de la construction de piliers dans les nids de termites). Le mécanisme permettant de résoudre un problème trop complexe pour être abordé par des fourmis seules est un bon exemple de système auto-organisé. Ce système repose sur des rétroactions *positives* (le dépôt de phéromone attire d'autres fourmis qui vont la renforcer à leur tour) et *négatives* (la dissipation de la piste par évaporation empêche le système de s'emballer). Théoriquement, si la quantité de phéromone restait identique au cours du temps sur toutes les branches, aucune piste ne serait choisie. Or, du fait des rétroactions, une faible variation sur une branche va être amplifiée et permettre alors le choix d'une branche. L'algorithme va permettre de passer d'un état instable où aucune branche n'est plus marquée qu'une autre, vers un état stable où l'itinéraire est formé des meilleures branches.

5.5. Quelques concepts de base

Avant de s'intéresser aux algorithmes de fourmis, il convient tout d'abord de présenter quelques concepts de base qui seront utilisées tout au long de cette section.

5.5.1. Problème d'optimisation

Un problème d'optimisation est tout problème définit par un espace de recherche des solutions, une fonction objectif qui associe un coût à chaque solution possible et un ensemble de contraintes. On cherche alors à trouver la solution optimale qui correspond à une solution de coût minimum ou maximum selon qu'il s'agit de minimiser ou de maximiser la fonction objectif.

5.5.2. Problème d'optimisation combinatoire

Un problème d'optimisation combinatoire est tout problème d'optimisation pour lequel il faut trouver une solution optimale avec un espace de recherche de solutions fini mais extrêmement grand. Ce type de problème est dit « difficile ».

5.5.3. Méthodes de résolution

Les méthodes de résolution des problèmes d'optimisation sont de deux types :

- Les méthodes exactes (déterministes): elles fournissent une solution optimale au prix d'un temps de résolution qui risque d'être exponentiel en fonction de la taille des données du problème.
- Les méthodes approchées : pour un problème d'optimisation dit « difficile » aucune méthode exacte n'est capable de le résoudre exactement en un temps raisonnable. Dans ce cas on fait appel à ses méthodes permettant une optimisation approchée. Ce type de méthodes retourne une solution contenue dans un certain intervalle autour de la solution optimum avec un temps de calcul acceptable. Elles représentent un compromis entre la qualité de la solution trouvée et le temps de calcul nécessaire. Parmi les méthodes de résolution approchées, on trouve :

a) Les heuristiques

Une heuristique est une méthode approchée simple, rapide et dédiée à un problème donné. Elle exploite les propriétés structurelles d'une solution et tente de la rendre rapidement une solution admissible par des critères de décision déduits de la connaissance du problème. Aucune garantie quant à l'optimalité de la solution trouvée ne peut être fournie.

b) Les méta-heuristiques

Une méta-heuristique est une méthode approchée générique dont le principe de fonctionnement repose sur des mécanismes généraux indépendants de tout problème. Les méta-heuristiques sont stochastiques et donc peuvent éviter d'être piégés dans des minimums locaux. Elles sont principalement guidées par le hasard (exploration aléatoire de l'espace de recherche), cependant elles sont souvent alliées à d'autres algorithmes afin d'en accélérer la convergence.

5.6. Algorithme Ant System (AS)

Ant System (AS) [127] est le premier algorithme de fourmi reposant sur le comportement des fourmis et appliqué pour la résolution du problème du voyageur de commerce (PVC) (TSP en anglais).

Le problème du voyageur de commerce, consiste à trouver un chemin Hamiltonien dans un graphe complètement connecté. Il s'agit pour un voyageur de commerce de trouver le chemin

le plus court pour visiter une et une seule fois chacune des *n* villes dans lesquelles il doit se rendre. L'espace de recherche est l'ensemble des combinaisons possibles des *n* villes. Il s'agit sans doute du problème d'optimisation combinatoire NP-complet le plus utilisé comme test pour les nouvelles méthodes d'optimisation.

Le PVC est modélisé par un graphe G (*V, E*) où *E* est l'ensemble des nœuds représentant les villes à visiter, et *V* est l'ensemble des arêtes. Une arête existe entre deux nœuds du graphe s'il existe un chemin reliant les deux villes qu'ils représentent. On utilise alors un graphe dont les arêtes sont étiquetées par la distance séparant deux villes.

Initialement, *m* fourmis sont placées aléatoirement sur les nœuds du graphe. Ensuite chacune des fourmis se déplace d'un nœud à un autre en parcourant les arêtes du graphe. Ce déplacement dépend de la liste des villes déjà visitées représentant la mémoire de la fourmi et d'une probabilité fonction de la distance reliant les villes et de la quantité de phéromone présente sur les arêtes du graphe.

L'algorithme AS est constitué d'un nombre d'itération appelée « cycle ». A chaque cycle, chaque fourmi *k* (*k = 1…..m*) parcourt le graphe en se déplaçant d'un nœud vers un autre. Le choix du passage d'un nœud *i* à un nœud *j* se fait aléatoirement en fonction d'une probabilité donnée par l'équation suivante :

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}(t)^{\alpha}.\eta_{ij}^{\beta}}{\sum_{l \in J_i^k} \tau_{il}(t)^{\alpha}.\eta_{il}^{\beta}} & si\ j \in J_i^k \\ 0 & si\ j \notin J_i^k \end{cases} \qquad (5.1)$$

Avec:

- $p_{ij}(t)$: la probabilité qu'une fourmi se situant à la ville i aille à la ville j à l'instant *t*,
- J_i^k : la liste des déplacements possibles pour une fourmi k lorsqu'elle se trouve à une ville i
- η_{ij} : la visibilité, qui est égale à l'inverse de la distance entre les villes *i* et j ($1/d_{ij}$)
- $\tau_{ij}(t)$: la quantité de phéromone sur l'arête *ij* à une itération donnée *t.*
- α et β : sont des constantes qui servent à régler l'importance relative que l'on donne à l'heuristique et à la phéromone.

5.6.1. Mise à jour des phéromones

La mise à jour des phéromones est effectuée à la fin de chaque cycle (ou *run*), c'est-à-dire lorsque toutes les fourmis ont fini de construire leur parcours. Celle-ci est faite en abaissant le taux de phéromone sur toutes les arêtes par un facteur constant (le taux d'évaporation), et puis par l'ajout de phéromones sur les arêtes par lesquelles les fourmis sont passées. Il faut considérer

l'atténuation des phéromones. En effet, ce sont des substances chimiques qui s'évaporent. Pour cela on fait intervenir le taux d'évaporation ρ.

L'évaporation de phéromone est implémentée de la façon suivante :

$$\tau_{ij} \leftarrow \tau_{ij}.\ (1 - \rho) \qquad (5.2)$$

Soient τij le taux de phéromone de l'arête ij et ρ le taux d'évaporation

$$\tau_{ij} \leftarrow \tau_{ij}.\ (1 - \rho) \qquad avec\ 0 < \rho < 1 \qquad (5.3)$$

L'évaporation de phéromone va permettre de faire disparaitre les mauvaises solutions (les autres chemins). Le paramètre est utilisé pour empêcher l'accumulation illimitée de phéromone et permet à l'algorithme de négliger les autres solutions (mauvaises). En effet, si une arête n'est pas choisie par les fourmis, le taux de phéromone qui lui est associé décroit exponentiellement avec le nombre d'itérations. Aussi à la fin de leur parcours, toutes les fourmis déposent des phéromones sur les arêtes par lesquelles elles sont passées durant leur tour; une fourmi *k* dépose une quantité τij de phéromone sur chaque arête de son parcours:

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^k(t)} & si\ (i,j) \in T^k(t) \\ 0 & si\ (i,j) \notin\ T^k(t) \end{cases} \qquad (5.4)$$

T^k : l'hamiltonien réalisé par la fourmi k à l'itération t,

L^k : la longueur du trajet

Q : un paramètre de réglage.

A chaque itération de l'algorithme, on a donc la somme des phéromones qui ne se sont pas évaporées et de celles qui viennent d'être déposées.

On obtient ainsi:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \qquad (5.5)$$

Où *m* est le nombre de fourmis utilisées pour l'itération *t.*

En général, les arêtes parcourues par beaucoup de fourmis et qui font partie d'un tour de courte longueur reçoivent plus de phéromone et seront ainsi choisies plus fortement par les fourmis dans les itérations futures.

Le taux d'évaporation *p* est un paramètre de contrôle, Il va falloir déterminer sa valeur. Si l'on choisit *p* élevé, la longueur du trajet va rapidement converger vers une même valeur et la solution va apparaître plus rapidement, mais celle-ci ne sera pas optimale car on supprime des arêtes qui pourraient être intéressantes. Si on prend *p* petit, beaucoup d'arêtes vont rester marquées et on n'arrivera pas à améliorer les trajets pour pouvoir parvenir à la solution optimale.

Ainsi AS nécessite de nombreux réglages. Suivant les valeurs que l'on attribuera aux différents paramètres, la solution sera trouvée plus ou moins rapidement et sera plus ou moins optimale.

5.7. Algorithme Ant-net

La flexibilité des algorithmes d'optimisation par colonie de fourmis a permis de les appliquer `a des données dynamique. Une de ces possibilités d'adaptation est la gestion des tables de routage. Un réseau peut être vu comme un graphe dont les sommets représentent les routeurs.
Pour exprimer la qualité du réseau, nous disposons de 2 grandeurs :

- La bande passante.
- Le délai moyen.

Pour cet algorithme il existe 2 types de fourmis :

- Les F-ants : ce sont les fourmis-tests, elles parcourent le réseau en enregistrant l'encombrement entre les routeurs et le chemin qu'elle a suivit. Sa fonction décisionnelle est probabiliste et elle peut créer une fourmi B-ant.
- Les B-ants : ce sont les fourmis qui propagent le résultat des 1eres : ells suivent le même parcours que leur créatrice mais en sens inverse afin de pouvoir mettre à jour les tables de routage en fonction de la charge du réseau

L'algorithme AntNet reprend les grandes lignes de AS, les seules différences vraiment notables sont :

- Qu'il n'y a aucune contrainte sur les villes.
- Qu'il y a une piste de phéromone différente pour chaque nœud de destination

L'algorithme peut se décomposer de la façon suivante :

1. Chaque nœud lance une fourmi F-ant vers une destination aléatoire.

2. Chaque fourmi suit sa propre route grâce à une fonction stochastique prenant en paramètre le taux de phéromone de la piste $\tau_{ij}d(t)$ et une valeur proportionnelle à la liste d'attente sur ce segment.

3. Chaque fourmi mémorise son parcours et le délai entre chaque nœud. Elle incrémente également le taux de phéromone des pistes traversées.

4. Une fois arrivé à destination, la F-ant créée une B-ant, celle-ci hérite des informations de la première et emprunte le même chemin en sens inverse.

5. les F-ants sont effacées et on met à jour les $\tau_{ijd}(t)$pour simuler l'évaporation.

6. La B-ant mettre à jour les tables de routage.

5.8. AntHocNet

AntHocNet [128] est un algorithme de routage hybride pour les réseaux mobiles ad hoc construit autour du modèle des ACOs. Il est constitué à la fois d'un mécanisme proactif et d'un mécanisme réactif. Il ne maintient pas les routes entre toutes les paires de stations du réseau à tout instant comme le fait AntNet (dont il est fortement inspiré), mais maintient seulement les routes vers un ensemble de stations quand une demande d'échange de données est initiée. Dans une phase d'initialisation, l'algorithme adopte un comportement réactif où des agents « fourmis réactives » sont propagés dans le réseau à partir des sources, dans le but de trouver des chemins multiples vers les destinations. En accord avec le mécanisme des ACOs, les tables de routage des différentes stations sont considérées comme des pistes de phéromone dont les valeurs quantitatives rendent plus ou moins attractifs des chemins en fonction de leur qualité. Après la phase 'initialisation, les paquets de données sont transférés en choisissant des chemins de manière probabiliste et ce choix influencé par les tables de phéromone. En parallèle du transfert, une autre classe d'agents, les « fourmis proactives » est propagée dans le but de maintenir et de mettre à jour les tables de routage.

5.8.1. L'établissement Réactive des Chemins

Quand un nœud source s commence une session de communication avec un nœud destination *d*, et n'a aucune information disponible sur le nœud destination *d*, il fait une diffusion d'agents appelés 'Reactive Forward Ants'F_d^s. Due à cette diffusion initiale, chaque nœud voisin de *s* reçoit une copie F_d^s(k) de F_d^s. Dans ce qui suit, on va référer à l'ensemble des copies qui sont originaires de la même fourmi originale comme *Ant Generation*. La tache de chaque fourmi F_d^s(k) est de trouver un chemin connectant *s* à *d*. A chaque nœud, une fourmi peut aussi soit envoyée directement (unicasted) ou diffusée (broadcasted), Selon si le nœud courant possède des informations de routage sur *d* ou non. L'information de routage d'un nœud *i* est représentée dans sa table de phéromone T_i. L'entrée $T_{nd}^i \in R$ de cette table est la valeur de phéromone indiquant l'estimation de la bonté du chemin allant de *i* à travers le voisin *n* pour atteindre la destination *d*. Si l'information de phéromone est disponible la fourmi va choisir son prochain saut *n* avec la probabilité P_{nd} :

$$P_{nd} = \frac{(T_{nd}^i)^{\beta_1}}{\sum_{j \epsilon N_d^i} (T_{jd}^i)^{\beta_1}}, \quad \beta_1 \geq 1, \qquad (5.6)$$

Où N_d^i est l'ensemble des voisins de *i* à travers lequel le chemin jusqu'à *d* est connu, et β_1 un

paramètre de réglage peut varier selon le comportement exploratoire des fourmis (bien que dans les expériences courantes β_1 était utilisé à 1)

Si aucune information sur *d* n'est disponible, la fourmi est diffusée (broadcasted). Due a cette diffusion, les fourmis peuvent rapidement proliférer sur le réseau, suivant différents chemins pour atteindre la destination (bien que les fourmis qui ont atteint le nombre maximum de sauts, relié au diamètre du réseau seront éliminées). Quand un nœud reçoit plusieurs fourmis de la même génération, il compare le chemin qu'a fait chacune d'elle : seulement si son nombre de sauts et le temps de voyage s'accommodent avec un facteur d'acceptation α_1 pour que la meilleure d'entre elles continue sa route, et les autres éliminées. Par cette politique, L'Overhead est limité en éliminant les fourmis qui ont suivi des chemins de qualité inférieure, Alors qu'il est possible de trouver d'autres bons chemins. Cependant, il est en effet que la fourmi qui arrive la première à un nœud est laissée à poursuivre son chemin, alors que les autres ultérieurs sont sélectionnées selon les autres critères établis par la fourmi qui les à précédées, qui veut dire qu'elles ont de très grandes chances d'être rejetées. Cela peut conduire à un "schéma de survol" de chemins. Dans le but d'obtenir une grille de multiples chemins suffisante comme le montre la figure 5.2, qui fournit une meilleure protection en cas de défaillances de chemins, on considère dans la politique de sélection le premier saut fait par la fourmi. Si ce premier saut est différent de celui des autres pris par les fourmis qui ont été acceptées, on applique un facteur d'acceptation plus élevé α_2 (moins restrictif) alors dans le cas le premier saut était déjà vu précédemment (dans les expériences α_2 était à 2 par contre α_1 était à 0.9). Une Stratégie similaire peut être retrouvée dans [128]

Chaque 'Forward Ant' retient (mémorise) une liste *P* des nœuds [1,.., n] qu'elle a déjà visité. Une fois arrivée à la destination *d*, Elle est convertie en 'Backward Ant', qui retourne sur ses pas vers la source *s* retraçant ainsi *P* (si le prochain saut n'est pas possible, à cause d'un mouvement d'un nœud, elle sera éliminée) .L'agent 'Backward Ant' calcule en incrémentant un temps estimé $\check{T}_p$ du temps qui va être nécessaire au paquet de données depuis la source *s* vers la destination *d* à travers $\mathcal{P}$, qui est utilisé pour mettre les tables de routage. $\check{T}_p$ est la somme

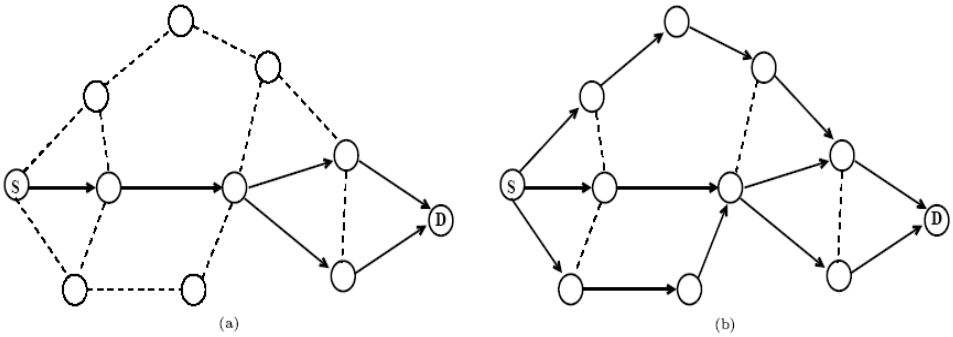


Figure 5.2. Kite shape

des estimations locales $\check{T}_{i+1}^{i}$dans chaque nœud i ϵ *P* du temps pour atteindre le prochain saut i+1 :

$$\check{T}_p = \sum_{i=1}^{n-1} \check{T}_{i+1}^{i} \qquad (5.7)$$

La valeur de $\check{T}_{i+1}^{i}$ est définit comme le produit de l'estimation de la moyenne pour envoyer un seul paquet, $\check{T}_{mac}^{i}$, calcule le temps du nombre actuel des paquets se trouvent en queue dans le MAC Layer, Q_{mac}^{i}:

$$\check{T}_{i+1}^{i} = (Q_{mac}^{i} + 1)\, \check{T}_{mac}^{i} \qquad (5.8)$$

$\check{T}_{mac}^{i}$ est calculé comme une moyenne courante du temps écoulé entre l'arrivée d'un paquet au MAC layer (la couche MAC) et la fin d'une transmission avec succès. Alors si t_{mac}^{i}est le temps nécessaire pour l'envoi d'un seul paquet depuis un nœud *i*, alors le nœud *i* met à jour son estimation comme suit :

$$\check{T}_{mac}^{i} = \alpha \check{T}_{mac}^{i} + (1-\alpha) t_{mac}^{i} \qquad (5.8)$$

Avec α ϵ [1,0]. Depuis $\check{T}_{mac}^{i}$ est calculé au MAC layer qui inclut les activités de l'accès au canal, alors il prend en compte l'embouteillage local du moyen partagé. Les 'Forward Ants' calculent un similaire temps estimé $\check{T}_p$, qui est utilisé pour le filtrage des fourmis, comme il a été mentionné au-dessus.

A chaque nœud intermédiaire *i* ϵ P, l'agent 'Backward Ant' établit virtuellement un chemin vers la destination *d*, créant ou mettant à jour l'entrée de la table de phéromone T_{nd}^{i} ϵ T_i. La valeur de phéromone dans T_{nd}^{i} représente une moyenne courante de l'inverse du coût, en termes de deux valeurs : le temps estimé et le nombre de sauts, pour atteindre *d* en passant par *n*. Si τ_d^i

est le temps de voyage estimé par la fourmi, et *h* est le nombre de sauts, la valeur T_{nd}^{i} utilisée pour mettre à jour la moyenne courante est définit comme :

$$\tau_d^i = \left(\frac{\hat{T}_d^i + hT_{hop}}{2} \right)^{-1},$$

Où T_{hop} est une valeur fixe que prend un saut dans des conditions de décharge. Définissant τ_d^i de cette façon évite la possibilité de larges oscillations dans l'estimation de temps rassemblé par les fourmis (dus aux débordements locaux du trafic) et pour prendre en compte le end-to-end delay et le nombre de sauts. La valeur de T_{nd}^{i} est mise à jour comme suit :

$$T_{nd}^{i} = \gamma T_{nd}^{i} + (1-\gamma)\tau_d^i, \quad \gamma \in [0, 1], \qquad (5.9)$$

Où γ et α étaient tous deux à 0.7 dans les expériences.

5.8.2. Le Routage Stochastique des Données

La phase d'initialisation d'un chemin comme décrit au-dessus crée un bon nombre de chemins reliant la source à la destination, indiqué dans les tables de routage des nœuds. Les données peuvent alors être expédiées (forwarded) entre les nœuds selon les valeurs des entrées de phéromone. Les nœuds dans notre protocole expédient les données d'une manière stochastique. Quand un nœud à plusieurs prochains sauts pour la destination *d*, il sélectionne aléatoirement l'un d'eux, avec la probabilité P_{nd}. P_{nd} est calculée de la même façon que pour les 'Reactive Forward Ants', mais avec un exposant plus élevé (dans les expériences on l'a mis à 2), pour être plus gourmant et respecter les meilleurs chemins :

$$P_{nd} = \frac{(T_{nd}^{i})^{\beta_2}}{\sum_{j \in N_d^i} (T_{jd}^{i})^{\beta_2}}, \quad \beta_2 \geq \beta_1 \qquad (5.10)$$

Selon cette stratégie, on n'a pas à choisir à priori combien de nœuds à choisir : leur nombre sera automatiquement choisi selon leur qualité.

La probable stratégie de routage mène à la diffusion lors du chargement des données selon la qualité estimée des chemins. Si les estimations sont misent à jour, cela mène à un équilibrage automatique de charge '*automatic load balancing*'. Quand un chemin est clairement plus mauvais que les autres, il va être évité, et son embouteillage va être déchargé. Les autres

chemins vont avoir plus de trafic, menant à un embouteillage plus élevé, ce qui va augmenter leur end-to-end delay. En mettant à jour d'une manière continue le trafic des données, les nœuds vont éventuellement de diffuser le chargement des données sur le réseau.

5.8.3. Maintenance et Exploration Proactive des Chemins

Lorsqu'une session de données est en cours, le nœud source envoie à l'extérieur les agents 'Proactive Forward Ants' selon le rythme d'envoi des données (une fourmi avec un paquet de données). Normalement ils sont envoyés directement (unicasted), choisissant le prochain saut selon les valeurs de phéromone en utilisant la même formule que les 'Reactive Forward Ants', mais ont aussi une petite probabilité à chaque nœud d'être diffusées (broadcasted) (cette probabilité était fixée à 0.1 dans toutes les expériences). De cette façon ils servent deux buts essentiels. Si une 'Forward Ant' atteint la destination sans une seule diffusion elle essaye simplement un chemin existant. Elle acquit la mise à jour estimée de ce chemin, est la 'Backward Ant' met à jour la valeur de la phéromone se trouvant sur ce chemin, juste comme le fait la 'Reactive Backward Ant'. En contrepartie la fourmi a été diffusée à chaque nœud, elle quitte les chemins courants pour explorer d'autres.

Après une diffusion la fourmi arrive à chaque nœud voisin du nœud source. Il est tout à fait possible qu'elle ne trouve aucune valeur de phéromone pointant vers la destination, cela induit qu'elle sera une fois de plus diffusée. La fourmi va alors rapidement proliférer et inondé le réseau, comme peuvent faire les 'Forward Reactive Ants'. Pour que cela n'arrive pas on limitera le nombre de diffusions à n_b (n_b était durant les expériences à 2). Si la '*Proactive Ant*' ne trouve aucune information de routage avec un nombre *nb* sauts, elle sera supprimée. L'effet de ce mécanisme est que la recherche de nouveaux chemins est concentrée autour des chemins existants, alors on cherche *les améliorations et les variations des chemins.*

Afin de mieux guider les '*Forward Ants*', on utilise les messages *hello*. Ce sont de courts messages (dans notre cas ils contiennent juste l'adresse de l'expéditeur) chaque quantum de temps t_{Hello} (ex t_{Hello} = 1s) par les nœuds. Si un nœud reçoit un message *hello* d'un nouveau nœud *n*, il rajoute *n* comme une nouvelle destination dans sa table de routage. Après cela il s'attend à recevoir un message *hello* de *n* chaque t_{Hello} secondes. Après n'avoir pas reçu un certain nombre de messages *hello* (*k messages perdus tolérés* = 2 dans notre cas), *n* sera supprimé de la table de routage. En utilisant ces messages, les nœuds connaissent leurs voisins intermédiaires et possèdent des informations de phéromone dans leurs tables de routage. Alors

quand une fourmi arrive à un voisin de la destination, elle peut aller droit vers son but. Si on regarde en arrière à '*Ant Colony*' la source d'inspiration de ce modèle, cela peut être vu comme diffusion de phéromone : la phéromone déposée au sol se diffuse, et peut être détectée par des fourmis se trouvant plus loin. Les messages *hello* peuvent aussi servir un autre but : peuvent détectés les liens cassés. Cela permet aux nœuds de nettoyer les entrées de phéromones inutiles dans leurs tables de routage.

5.8.4. Les panne des liens :

Dans ce protocole, chaque nœud essaye de maintenir une vue à jour de ses voisins à chaque moment. Afin de détecter les défaillances des liens le plus vite possible, avant que ça conduit à des transmissions incorrectes et entraîne la perte de données. La présence d'un nœud voisin se confirme à l'arrivée d'un message *hello*, ou après l'interception ou l'échange avec succès de signaux. La disparition d'un nœud est assumée après qu'un évènement attendu n'a pas eu lieu après un certain temps, définit par t_{Hello} * *k messages perdus tolérés*, ou après l'échec d'un envoi direct vers ce voisin.

Après assumer la disparition d'un nœud, le nœud prend un certain nombre d'actions. En premier lieu, il supprime ce voisin de sa liste de voisins et toutes les entrées associées à ce voisin de sa table de routage. Après il diffuse un message de *notification d'échec de lien.* Comme un message contenant une liste de toutes les destinations pour laquelle le nœud a perdu son meilleur chemin, et le nouveau meilleur chemin estime le end-to-end delay et le nombre de sauts à cette destination (s'il possède encore des entrées pour la destination). Tous ses voisins reçoivent la notification et mettent à jour leurs tables de phéromone en utilisant la nouvelle estimation. Par leur tour si l'un d'eux perd son meilleur ou son unique chemin vers la destination due a cette coupure ou défaillance, ils diffuseront cette notification plus loin, jusqu'à ce que tous les nœuds concernés soient au courant de cette nouvelle situation.

Si la coupure du lien a été reconnue suite à l'échec d'envoi d'un paquet de données, et il n'existe aucun autre chemin disponible pour ce paquet, le nœud va essayer de réparer le chemin localement (et ne va pas inclure ce chemin dans le message de *notification d'échec de lien*). Le nœud diffuse une '*route repair ant'* soit une fourmi de réparation de route qui part à la destination concernée comme une '*Reactive forward Ant*' : qui essaye de suivre l'information de routage disponible tant que possible, autrement elle sera diffusée. Une seule importante différence, c'est qu'elle a un nombre limité (max) de diffusion (qui était à 2 durant les

expériences), Ainsi sa multiplication sera limitée. Le nœud attend un certain temps (empiriquement mis à 5 unités de temps le end-to-end delay du chemin perdu), ensuite si aucune '*backward repair ant*' n'est reçue, il conclut que la réparation du chemin vers la destination était impossible. Entre-temps les paquets qui étaient temporisées pour la destination seront éliminés, et le nœud envoie une notification d'échec de lien sur cette destination perdue.

Les Notification d'échec de lien gardent les tables de routage sur les chemins à jour sue les coupures des liens en amont. Cependant, parfois ils peuvent perdre et suspendre des liens. Un paquet de données suivant un lien arrive à un nœud ou aucune phéromone n'est disponible ni de près ni de loin. Le nœud va alors supprimer le paquet de données et envoyé en retour un message d'avertissement aux sauts précédents, qui peut aider à éliminer les fausses informations de routage.

5.9. Conclusion

Dans ce chapitre, nous avons présenté les algorithmes à base de fourmis inspirés de comportement collectif des fourmis pour la résolution des problèmes d'optimisation. Dans le chapitre suivant nous illustrerons l'application de ce mécanisme dans les réseaux Ad Hoc. Par la suite nous avons parlé du problème d'acheminement des paquets dans les réseaux ad hoc, c'est à dire le problème de routage et ses difficultés. Nous avons donné une présentation synthétique de quelques différentes solutions basés sur les colonies de fourmis qui existent et qui résolvent le problème de routage dans les réseaux mobiles ad hoc, dans le chapitre suivant on va présenter notre propre vision [125] pour le problème de diagnostic des pannes et défaut dans les réseaux MANET. Nous proposons une approche du diagnostic des réseaux de Petri basé sur la notion de marquage de base et la logique floue.

Chapitre 6

Diagnostique de défaut pour les systèmes à événements discrets à l'aide de réseaux de Petri avec des transitions non observables, application à un système de communication ad-hoc

Résumé

Dans ce chapitre, nous présentons une approche de détection de défaut pour les systèmes à événements discrets à l'aide de réseaux de Petri. Nous supposons que certaines des transitions du réseau ne sont pas observables, y compris toutes ces transitions des comportements défectueux. Notre approche de diagnostic est basée sur la notion de base de marquage et la justification, qui nous permettent de caractériser l'ensemble des marquages qui sont compatibles avec l'observation réelle, et l'ensemble des transitions non observables dont le déclenchement le permet. Nous ajoutons à la fin de ce chapitre la notion de la logique floue.

6.1. Defénition de l'outil de diagnostic

Le RdPSyncF modélise l'évolution temporelle des défauts, exprimée par un outil descriptif dynamique. La propagation et l'évolution des défauts sont évaluées par des sections temporelles du marquage du RdPSyncF. Ce modèle est basé sur des informations prédictives qui déclarent a priori certains défauts, comme étant critiques dans l'évolution du système. Ces états critiques identifiés par des places correspondantes du RdPSyncF, définissent le chemin critique des défauts. Les places critiques sont caractérisées par des seuils de criticité des défauts correspondants. Elles matérialisent l'interface qui permet de déclencher l'action de reprise routage / maintenance des liens.

Le RdPSyncF est défini comme étant le n-uplet avec :

$\boldsymbol{P} = \{p_1, p_2, \ldots.. p_n\}$ ensemble fini des places

$T = \{t_1, t_2, \ldots.. t_n\}$ ensemble fini des transitions qui modélisent l'évolution des défauts, en conformité avec un ensemble des règles logiques floues R. Chaque transition est associée à une règle ;

$D = \{d_1, d_2, \ldots..dn\}$ ensemble fini de propositions logiques qui définissent la base des règles R ;

$I : T \rightarrow P$ fonction d'entrée dans les places ;

$o : P \rightarrow T$ fonction de sortie des places ;

$f_i : T \rightarrow F$ fonction qui associe à chaque règle modélisée par une transition, une fonction d'appartenance $\boldsymbol{F}$ qui décrit le degré de crédibilité $\mu_i = F(t)$ de la règle. L'instant t correspond à la détection d'un symptôme de défaut ;

$\alpha_j : P \rightarrow [0,1]$ fonction associative qui établi une valeur de crédibilité α_j (nombre flou) pour les places correspondant aux propositions logiques $d_j \in D$. Ce paramètre représente la possibilité d'apparition du défaut associé à la place p_j ;

$\beta_j : P \rightarrow D$ fonction bijective qui associe les propositions logiques d_j aux places $p_j \in P$;

$\lambda_k : P \rightarrow [0,1]$ fonction qui associe à une place $p_k \in P$, une valeur de seuil λ_k d'acceptabilité/permissivité pour le défaut correspondant, du point de vue de l'action de *reprise*. λ_k correspond au seuil d'avertissement du paramètre α. Le paramètre λ_k est associé uniquement à certaines places symbolisant les défauts dérivés critiques. Le dépassement de ce seuil nécessite une action de reprise;

!?S *= {s_1, s_2,..., s_l}*	ensemble des signaux (flous) des symptômes des défauts envoyés (!) ou reçus (?) par le système surveillé respectivement le système de surveillance. Ils arrivent sur les transitions chargées de l'information temporelle « instant ***t*** de franchissement »;
!R *= {r_1,r_2,..., r_l}*	ensemble des signaux (flous) de reprise émis par le système de surveillance;
$\propto$	est une fonction qui définit la valeur du degré de confiance (DoT) pour un jeton dans la place C à D $\propto: P \rightarrow [0,1]$
M_0	marquage initial.

Par l'intermédiaire des transitions temporelles, une modélisation de *l'évolution dynamique* de la propagation des défauts est mise en œuvre. Les signaux de synchronisation avec le modèle (RdPTO) de détection directe, valident le franchissement des transitions, durant les fenêtres temporelles allouées. Chaque implication élémentaire dispose des paramètres suivants :

- La valeur floue α_j représente la modélisation de la gravité du défaut, en fonction de l'instant de son apparition. Cette valeur est élaborée dans le modèle RdPTO.
- La valeur floue μ_i représente le degré de vérité de l'implication logique $d_i \rightarrow d_k$ en fonction de l'instant d'arrivée du signal de défaut. Cette valeur est élaborée dans le modèle RdPFS et elle représente le degré d'appartenance de la variable floue t – instant d'émission d'un signal de défaut – à la variable linguistique "apparition des défauts d_i, d_k ".
- L'instant t d'injection du signal flou représente l'instant de franchissement de la transition du RdPFS si elle ne se trouve pas à la concurrence des plusieurs places. Dans ce cas, t représente l'instant de franchissement possible de la transition temporelle ou le début de sa sensibilisation.

Dans un système de réseaux mobile Ad-hoc surveillé (modèle *RdPTO*), l'instant de franchissement *t* de la transition associée à une activité a_i surveillée (faisant l'objet d'une détection directe) peut être modélisé par la fonction d'appartenance FN correspondant à une activité normale ou par la fonction d'appartenance F ou F correspondant à l'apparition d'un défaut d'exécution (Figure 6.1.). La détection directe implantée dans le modèle *RdPTO* du système, surveille ainsi l'apparition des défauts par un mécanisme de type chien de garde (CG). Ce mécanisme envoie un signal !si, en utilisant une modélisation floue de l'instant de franchissement de la transition du chien de garde (TG). Le signal si est chargé d'une variable floue $\mu_i=F(t)$ associée au degré de crédibilité de la règle de propagation du défaut dj vers le défaut dérivé d_k. Dans le réseau *RdPFS*, le signal si est réceptionné (?s_i) et une modélisation

floue de l'implication $d_j \rightarrow d_k$ permet de calculer la valeur de vérité de la variable logique d_k induite.

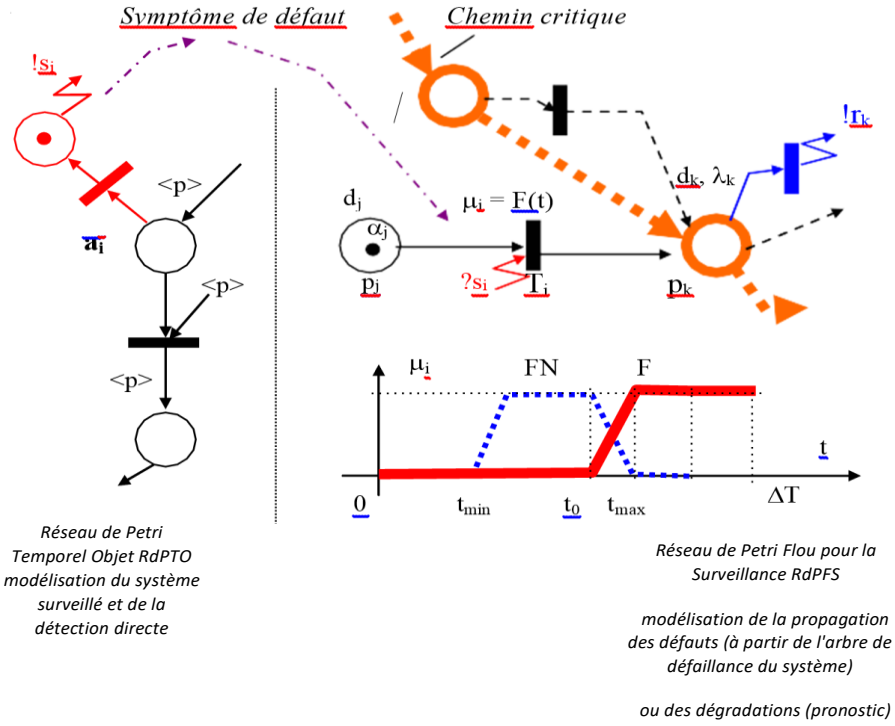


Figure 6.1. Principe de modélisation de la propagation des défauts

Le point de départ de la propagation des défauts dans le modèle de surveillance RdPFS, est constitué par les *défauts de base* (de type d_j). Leur apparition est signalée par l'ensemble des signaux de synchronisation $!s_j$ émis par le modèle RdPTO du système surveillé. Les *défauts dérivés* (de type d_k) sont des combinaisons logiques de défauts de base et/ou de défauts dérivés antécédents.

Chaque transition T_i du RdPFS correspond à une règle floue qui décrit la propagation des défauts. Elle est associée à la fonction $\mu_i = F(t)$ qui exprime le degré de crédibilité de la règle de propagation des défauts modélisée par la transition concernée. Cette fonction associe à chaque règle une *crédibilité dynamique*.

Chaque place du RdPFS correspond à un défaut (ou une dégradation) du système surveillé. Le marquage d'une place p_j correspondant à un défaut de base, est associé à une valeur floue $\alpha_j \in [0, 1]$ de crédibilité (gravité) du défaut de base observé. La valeur floue α_k, associée au marquage de la place p_k correspondant à un défaut dérivé, est calculée en appliquant le modus ponens généralisé (Chen, 1990). Dans notre approche, nous considérons les opérateurs : T(u,v) = min(u,v) - norme triangulaire (t-norme) et ⊥(u,v) =

max(u,v) - conorme triangulaire (t-conorme), ainsi que l'opérateur modus ponens généralisé $T_{probabiliste}(u, v) = u.v$. La valeur de crédibilité α_k de la conclusion sera ainsi :

Quand une place p_k appartenant au chemin critique est marquée, un signal $!r_k$ est instantanément envoyé vers le système de reprise (maintenance). Mis à part sa mission de synchronisation avec le sous-système cible, le signal sera chargé de deux informations : le nombre flou α_k associé au marquage de la place et la valeur du seuil λ_k.

Dans les paragraphes suivants, nous nous concentrerons uniquement sur la méthode de localisation de l'erreur, ce qui suggère une programmation linéaire.

Nous supposerons que le réseau est un RdP simple et ceci afin de faciliter et simplifier l'explication de la méthode proposée, nous expliquerons à la fin du chapitre comment intégrer la programmation linéaire avec le raisonnement Floue.

6.2. Marquages cohérentes

Dans ce chapitre, nous supposons aussi que l'ensemble des transitions T est divisée en deux sous-ensembles T_o *et* T_u, Sois $T = T_o \cup T_u$ *et* $T_o \cap T_u = \emptyset$. L'ensemble T_o comprend toutes les transitions *qui sont observables*, tandis que T_u comprend les transitions non observables ou silencieuses.

On note que n_o (resp., n_u) le cardinal de définir T_0 (resp., T_u), et en tant que C_o (resp., C_u) la restriction de la matrice d'incidence à T_o (T_u).

Définition 6.1. Soit $N = (P, T, Pré, Post)$ soit un réseau avec $T = T_o \cup T_u$. Nous définissons les deux opérateurs suivants.

- *La* projection sur T_o *est* $P_o : T^* \rightarrow T_0^*$ *définie comme suit : (i)* $P_o(\varepsilon) = \varepsilon$*; (ii) pour toute* $\sigma \in T^*$ *et* $t \in T$, $P_o(\sigma t) = P_o(\sigma)\, t$ *si* $t \in T_o$, *et* $P_o(\sigma t) = P_o(\sigma)$ autrement.
- *La* projection sur T_u *est* $P_u : T^* \rightarrow T_u^*$ *défini comme suit : (i)* $P_u(\varepsilon) = \varepsilon$*; (ii) pour toute* $\sigma \in T^*$ *et* $t \in T$, $P_u(\sigma t) = P_u(\sigma) t$ *si* $t \in T_u$ *et* $P_u(\sigma t) = P_u(\sigma)$ *autrement.*

Étant donné une séquence $\sigma \in L(N, M_0)$, on note $w = P_o(\sigma)$, *le mot observée* correspondant.

Définition 6.2. *Soit* $\langle N, M_0 \rangle$ un Réseau de Petri où $N = (P, T, Pre, Post)$ *et* $T = T_o \cup T_u$. *Et* $w \in T_0^*$ *un mot observée. Nous définissons*

$\mathscr{S}(w) = P_0^{-1}(w) \cap L(N, M_0) = \{\sigma \in L(N, M_0) \mid P_o(\Sigma) = w\}$,

l'ensemble des séquences de tir compatible avec $w \in T_0^*$

Définition 6.3. *Soit* $<N, M_0>$ un *RdP où* $N = (P, T, Pré, Post)$ *et* $T = T_o \cup T_u$. *Et* $w \in T_0^*$ *un mot observée. Nous définissons*

$\mathscr{C}(\boldsymbol{w}) = \{M \in R\,(N, M_0) \mid \exists\, \sigma \in \mathscr{S}(w) : M_0\,[\sigma> M\}$,

l'ensemble des marquages compatibles avec $w \in T_0^*$.

Compte tenu de l'observation *w*, $\mathscr{S}(w)$ est l'ensemble des séquences qui peuvent avoir tiré, tandis que $\mathscr{C}(w)$ est l'ensemble des marques dont le système peut être en réalité.

Exemple 6.1. On considère le système de la Figure 6.2 qui représente un protocole de communication simplifié, inspiré de (Ru & Hadjicostis 2009). Le modèle de ce système est représenté sur la Figure 6.3 avec le marquage initial M 1= [1 0 0 1 0 0 0 0 0 0 0 0 1 0] T Les messages sont transmis du client A vers le client B. Dans un premier temps, le message est divisé (transition $T1$) en deux paquets devant être acheminés sur deux canaux différents. À travers le canal 1, le premier paquet est successivement traité par les routeurs $R1$, $R3$ et $R5$ modélisés respectivement par les couples ($P7$, $T7$), ($P8$, $T8$) et ($P10$, $T10$) avant de parvenir au buffer 1 représenté par la place $P11$. Le paquet 2 est quant à lui traité par les routeurs $R2$, $R4$ et $R6$ modélisés respectivement par ($P2$, $T3$), ($P3$, $T4$) et ($P4$, $T5$) et entre dans le buffer 2 représenté par la place $P6$. Enfin, les paquets sont rassemblés pour reconstituer le message original et un accusé de réception est envoyé au client expéditeur (transition $T2$). Ceci représente le fonctionnement normal du système.

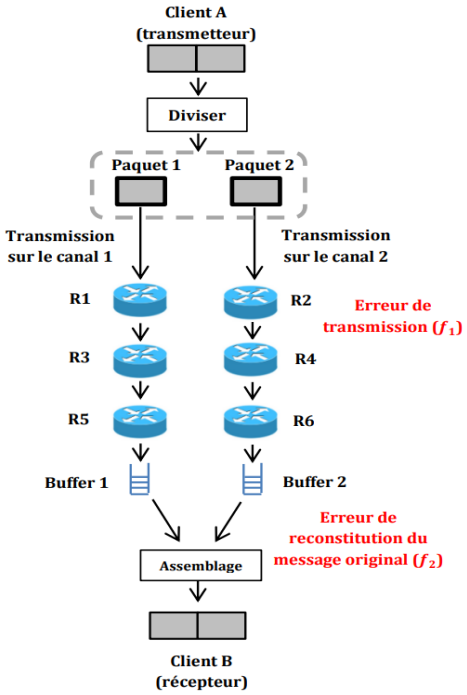


Figure 6.2. Protocole de communication simplifié (Ru & Hadjicostis 2009).

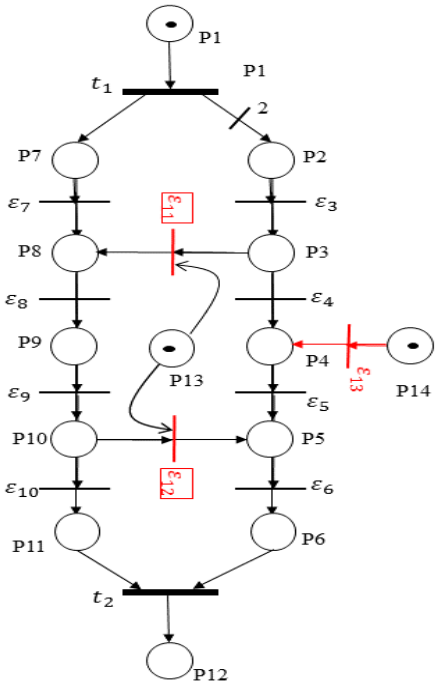


Figure 6.3 Modelé RdP correspondent

Nous supposons que $T_o = \{t_1, t_2\}$ *et* $T_u = \{\varepsilon_3, \varepsilon_4, \ldots, \varepsilon_{13}\}$,où, pour une meilleure compréhension , les transitions non observable sont notées ε_i plutôt que t_i.

Supposons qu'il y'aucun cas observée, à savoir $w = \varepsilon$. Alors $\mathscr{S}(\varepsilon) = \{\varepsilon, \varepsilon_{13}, \varepsilon_{13}\varepsilon_5, \varepsilon_{13}\varepsilon_5\varepsilon_6\}$ et $\mathscr{C}(\varepsilon) = \{M_0, M_1, M_2, M_3\}$, *dans laquelle* M_0 est le marquage initial, $M_1 = [1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$, $M_2 = [1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$ *et* $M_3 = [1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$.

Maintenant, supposons que t_1 est observée. La transition t_1 est activée au marquage initial, donc les tirs de transition non observable est nécessaire pour activer le tir de t_1, plusieurs séquences de transitions non observables sont activés, et plusieurs marquages sont donc compatibles avec le comportement réel. En particulier, toutes les séquences $t_1\varepsilon_3$, $t_1\varepsilon_3\varepsilon_3$, $t_1\varepsilon_3\varepsilon_3\varepsilon_4$, $t_1\varepsilon_3\varepsilon_3\varepsilon_4\varepsilon_4\varepsilon_{13}$, . . ., $t_1\varepsilon_7$, $t_1\varepsilon_7\varepsilon_8$. . ., Etc., peuvent avoir tiré et donné l'observation réelle, $\mathscr{C}(w)$ inclut toutes les marques atteintes lors du tir des séquences ci-dessus.

Maintenant, considérons $w=t_2$. Dans un tel cas, aucune séquence de transitions non observables ne peut l'activer. Donc, $\mathscr{C}(t_2) = \mathscr{S}(t_2) = \emptyset$.

Enfin, considérons $w=t_1t_2$. Dans ce cas, nous obtenons

$$\mathscr{S}(t_1t_2) = \{t_1\varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6\varepsilon_7\varepsilon_8\varepsilon_9\varepsilon_{10}t_2,\ \varepsilon_{13}t_1\varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6\varepsilon_5\varepsilon_6\varepsilon_7\varepsilon_8\varepsilon_9\varepsilon_{10}t_2,$$
$$t_1\varepsilon_3\varepsilon_{11}\varepsilon_8\varepsilon_9\varepsilon_{10}\varepsilon_{13}\varepsilon_5\varepsilon_6\ t_2,\ \ \varepsilon_{13}\ t_1\varepsilon_3\varepsilon_7\varepsilon_8\varepsilon_9\varepsilon_{10}\varepsilon_5\varepsilon_6,\ \ \ldots\},$$

$\mathscr{E}(t_1\ t_{2)}$ = {[0 1 0 0 0 0 0 0 0 0 0 1 1 1] ^{T}, [0 1 0 0 0 1 0 0 0 0 0 1 1 0] ^{T}, [0 1 0 0 0 0 1 0 0 0 0 1 0 0] ^{T,} [1 1 0 0 0 0 0 0 0 0 0 1 1 0] ^{T}, . .},

Où les points désignent toutes les autres séquences qui peuvent avoir tiré et toutes les autres marques compatibles avec t_1t_2, respectivement (qui ne sont pas rapportés ici par souci de concision).

6.3.Explications minimales et e-vecteurs minimaux

Dans cette section, nous présentons quelques définitions de base qui seront utiles par la suite.

Définition 6.4. *Soit un marquage M et une transition observable* $t \in T_o$, Nous définissons

$$\Sigma(M\ ,t\ ) = \{\sigma \in T_u^* | M\ [\sigma\rangle M', M' \geq Pre\ (\cdot,\ t\ )\}$$

l'ensemble des explications *de t à M, et nous définissons*

$$Y(M\ ,\ t\ )=\{e \in \mathbb{N}^{n_u} | \exists \sigma \in \sum (M\ ,\ t\ ) : \pi(\sigma)= e\ \}$$

les vecteurs e (ou vecteurs d'explication), c'est-à-dire les vecteurs de déclenchement associés aux explications.

Ainsi $\Sigma\ (M,\ t)$ est l'ensemble des séquences non observables dont le déclenchement en *M* active et permet *t.*

Parmi les séquences ci-dessus, nous voulons sélectionner celles dont le vecteur de déclenchement est minimal. Le vecteur de déclenchement de ces séquences est appelé e-vecteur minimal.

Définition 6.6. *Étant donné un marquage M et une transition* $t \in T_o$,Nous définissons

$$\Sigma_{min}(M,t) = \{\sigma \in \Sigma(M\ ,\ t\ ) | \nexists \sigma' \in \Sigma(M\ ,t\ ) : \pi(\sigma')\ \pi(\sigma)\}$$

l'ensemble des explications *minimales de t à M, et nous définissons*

$$Y_{\min}\ (M,\ t) = \{e \in \mathbb{N}_u^n\ \ |\ \exists\ \sigma \in \Sigma_{\min}\ (M,\ t):\ \pi\ (\sigma) = e\}$$

l'ensemble correspondant de vecteurs e-minimaux.

Exemple 6.2. Considérons le système de réseau de la figure 6.3 déjà pris en compte dans l'exemple 6.1. Nous calculons l'ensemble des e-vecteurs minimaux pour certains marquages. Les e-vecteurs correspondent aux transitions non observables de $\varepsilon_{\ 3\ \text{à}\ \varepsilon\ 13.}$

il tient que $\Sigma(M_{0,}\ t_{1)} = \mathscr{L}(t_{1)\}} = \{\varepsilon\}$. Ensuite, $\Sigma(M_{0,}\ t_{2)} = \mathscr{L}(t_{\ 2)} = \emptyset$. Enfin, *Soit* $M_{\ 1} = M_{\ 0} + C\ (\cdot,\ t_{\ 1)} = [2\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1]\ ^T$. alors $\Sigma(M_{\ 1,}\ t_{\ 2)} = \mathscr{L}(t_1t_{\ 2)}$ (voir l'exemple 6,4 et figure 6.3) et

$$Y_{\min}(M_1, t_2) = \{[1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0]^T, [2\,1\,1\,1\,0\,1\,1\,1\,1\,0\,0]^T,$$
$$[1\,0\,1\,1\,0\,1\,1\,1\,1\,0\,1]T, [0\,0\,1\,1\,1\,1\,1\,1\,1\,0\,0\,1]\,T\},$$

c'est-à-dire les vecteurs de déclenchement relatifs aux séquences dans $\mathcal{L}(t_1 t_2)$.

Dans [31] Corona *et al.* Prouvé le résultat important suivant. On dit qu'un reseau P / T est sans conflit en arrière si $\forall\, p \in P \mid p^{\bullet} \mid \leq 1$, c'est à dire, si chaque place possède un seul arc d'entrée.

Théorème 6.1. *[31] Soit N = (P, T, Pré, Post) un réseau de Petri avec $T = T_o \cup T_u$. Si le sous-réseau induit par T_u est sans conflit en arrière, alors | Y min (M, t) | =1.*

Différentes approches peuvent être utilisées pour calculer $Y_{\min}(M, t)$, par exemple, voir [122, 123].

Dans cette thèse, nous proposons une approche qui trouvent tous les vecteurs de $Y_{\min}(M, t)$ si elle est appliquée à des réseaux dont le sous-réseau induit par des transitions non observable T_u est acyclique (mais pas nécessairement en arrière sans conflit). Notre méthode est inspiré par la procédure proposée par Martinez Silva et [66] pour le calcul de P-invariants minimaux. Il peut se résumer par l'algorithme suivant.

Notez que l'approche proposée peut également être appliquée aux sous-réseaux induits par Tu qui ne sont pas acycliques. Cependant, dans ce cas, l'algorithme peut entrer dans une boucle : pour garantir la fin d'un nombre fini d'étapes, nous devons ajouter des critères de terminaison appropriés.

Algorithme 6.1. ***[Calcul du*** $Y_{\min}(M, t)$***]***

Entrée:

- *un réseau de Petri N dont le sous-réseau non observable est acyclique,*
- *l'ensemble des transitions non observables T_u*
- *un marquage M,*
- *une transition observable t.*

Sortie: $Y_{\min}(M, t)$.

C_U^T	$I_{n_u * n_u}$
A	B

1. ***Soit*** Γ : *où: A: = (M - Pre (·, t))^T, B :=* $\vec{0}_{n_u}^T$

2. *Tanque* $A \geq 0^T$

2.1. Choisissez un élément $A(i^*, j^*) < 0$.

2.2. ***Soit*** $\mathcal{L}^+ = \{i \mid C_u^T(i, j^*) > 0\}$.

2.3. Pour tout $i \in \mathcal{L}^+$

2.3.1. ajouter à $[A \mid B]$ *une nouvelle ligne*
$[A(i^*, \cdot) + C_u^T(i, \cdot) \mid B(i^*, \cdot) + e_i^T]$ *où* e_i *est le i ième vecteur de base canonique.*

2.4. Retirer la ligne $[A(i^*, \cdot) \mid B(i^*, \cdot)]$ *de la table.*

Fintanque

3. *Retirer de B toute ligne qui couvre les autres lignes.*

4. Chaque ligne de B est un vecteur de $Y_{min}(M, t)$.

L'algorithme ci-dessus peut être expliqué comme suit.

Étant donné un marquage *M* et une transition *t*, l'algorithme 6.9 calcule les e-vecteurs minimaux, c'est-à-dire les vecteurs de déclenchement de séquences non observables dont le déclenchement en *M* est nécessaire pour permettre à *t*.

A l'étape 1, un vecteur de ligne est défini, A = A (1, ·), qui a un nombre de colonnes égal au nombre de places du réseau. Ce vecteur contient un élément négatif *A (1, j)* si la place p_j n'active pas *t* en *M*, et la valeur | *A (i, j)* | indique le nombre de jetons manquants dans pj pour activer *t*. *B* est initialement un vecteur de déclenchement nul.

À l'étape 2.1, nous vérifions s'il existe une transition non observable dont le déclenchement peut augmenter le nombre de jetons dans p_j. Dans l'affirmative, nous considérons tous les déclenchements possibles (d'une seule transition) calculant les marquages atteints par chacun de ces déclenchements et le vecteur B, dans la partie droite du tableau, désigne le vecteur de déclenchement correspondant. Ces nouvelles marques et les vecteurs de déclenchement correspondants constitueront les nouvelles lignes de la matrice A, tandis que la ligne précédente sera supprimée.

La boucle *Tant que* est répétée jusqu'à ce que tous les marquages représentés par la matrice *A* aient des composants non négatifs

Notez que lors de l'étape 2.3, il est possible que la nouvelle ligne $[A (i^{*}, \cdot) + C_u{}^T (i, \cdot) \mid B (i^{*}, \cdot) + e_i{}^T]$ est identique à une ligne déjà présente dans la table: si tel est le cas, il est inutile de l'ajouter.

Un exemple détaillé d'application de l'algorithme 6.2 est donné à l'exemple 6.3.

Exemple 6.3. Considérons le réseau de la figure 6.1. Soit $M = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1]^T$ et $t = t_2$. Nous avons :

$$C_u=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \text{Pre}(.,t2)=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

nous supposons d'abord

Γ =

$$\left[\begin{array}{cccccccccccc|cccccccccc} 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

donc il y a deux éléments de A, à savoir *A (1, 6) et A (1,* 11), qui sont négatifs. Nous considérons *A(1,* 11) < 0. Dans un tel cas $\mathcal{F}^+$ = {8}, donc par étape 2.3.1 de l'algorithme 6.9 nous ajoutons la ligne suivante à Γ:

$$\left[\begin{array}{cccccccccccc|cccccccccc} 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

Obtenu à partir de la première ligne de A en ajoutant la 8ème ligne de Γ.

Maintenant, nous supprimons la ligne Γ (12, ·) de la table (étape 2.4) et on recommence la procédure parce que *la matrice A* contient encore des éléments négatifs, à savoir *A (1,* 6) *et A* (1, 10). Considérons *A (1,*10). Dans ce cas, $\mathcal{F}$+ = {7}, on ajoute donc la ligne suivante à Γ:

$$\left|\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 & 0 & 1 & 0 & 1 \end{array}\right| \left|\begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}\right|$$

obtenu à partir de la première ligne de A en ajoutant la 7ème ligne de Γ.

Maintenant, on enlève Γ (12, ·) de la table et on recommence la procédure, car il y a encore des éléments négatifs dans *A: A (1*,6) *et A* (1, 9). Nous nous concentrons sur *A (1*, 6) donc $\mathscr{S}^+$= {4}. Par étape 2.3.1 nous ajoutons la ligne suivante pour Γ:

$$\left|\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 1 & 0 & 1 \end{array}\right| \left|\begin{array}{cccccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}\right|$$

Enlever Γ (12, ·) maintenant de la table. Nous avons encore deux éléments négatifs dans *A: A (1, 5) et un* (1, 9). nous sélectionnons *A (1,* 9) ainsi $\mathscr{S}^+$= {6}. La nouvelle ligne à ajouter à Γ est:

$$\left|\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right| \left|\begin{array}{cccccccccc} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{array}\right|$$

obtenu à partir de la première *ligne de A* en ajoutant la 6ème ligne de Γ. Maintenant nous enlevons Γ (12, ·) de la table et parce qu'il ya encore des éléments négatifs dans *A*, à savoir *A (1*, 5) *et A* (1, 8), on recommence la procédure. Nous considérons A (1, 8) ainsi $\mathscr{S}^+$= {5, 9} et nous ajoutons deux nouvelles lignes pour Γ:

$$\left|\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{array}\right| \left|\begin{array}{cccccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{array}\right|$$

obtenu à partir de la première *ligne de A* en ajoutant le 5e et le 9e rang de Γ, respectivement. Maintenant nous enlevons Γ (12, ·) de la table et on recommence la procédure parce que A contient encore des éléments négatifs. Ceux-ci sont les suivants: *A (1, 5), A (2, 3), A (2, 5), A (2,* 13). Nous choisissons A (2, 13) ainsi $\mathscr{S}^+$ =∅. et nous devons enlever la deuxième ligne de *A. Or,* dans A, il n'ya qu'un seul élément négatif, c'est-à-dire A (1,5),. Dans ce cas, $\mathscr{S}^+$= {3, 10} et nous ajoutons deux nouvelles lignes pour Γ:

$$\left|\begin{array}{cccccccccccc} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & 1 \end{array}\right| \left|\begin{array}{cccccccccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \end{array}\right|$$

obtenu à partir de la première *ligne de A* en ajoutant le 3e et le 10e rang de Γ, respectivement. Maintenant nous enlevons Γ (12, ·) de la table et on recommence la procédure. *Une* matrice contient les éléments suivants:*Un* négatives (1, *4), A (2, 10), A*

(2, 13). *Nous choisissons un* (2, 13) ainsi $\mathcal{J}^+=\emptyset$, et nous devons supprimer la deuxième ligne de *A. Maintenant, dans une* il y a un seul élément négatif, à savoir, *(1,* 4), et $\mathcal{J}^+= \{2, 11\}$. Par étape 2.3.1 de l'algorithme 6.9 nous ajoutons les deux nouvelles lignes suivantes à Γ:

$$\left|\begin{array}{cccccccccccc|cccccccccc} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right|$$

obtenu à partir de la première *ligne de A* en ajoutant le 2e et le 11e rang de Γ, respectivement. Maintenant nous enlevons Γ (12, ·) de la table et on recommence la procédure, car il est un élément négatif dans A, à savoir A (1, 3). En outre, il détient $\mathcal{J}^+ = \{1\}$, et la nouvelle ligne à ajouter à Γ est:

$$\left|\begin{array}{cccccccccccc|cccccccccc} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array}\right|$$

obtenu à partir de la première *ligne de A* en ajoutant la 1ère ligne de Γ. Nous enlevons Γ (12, ·) et observe que *A (2,* 2) est négatif, et $\mathcal{J}^+$est vide. Ainsi, on enlève la deuxième ligne de *A.* Maintenant, nous nous arrêtons parce que tous les éléments *de A sont* non négatifs.

Comme nous n'avons qu'une ligne, aucune ligne ne couvre évidemment les autres, et nous concluons que la ligne de B, à savoir

$$\left|\begin{array}{cccccccccccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right|$$

est le seul élément de Y_{min} *(M,t).*

6.4.Marquages de Base et J-vecteurs

Dans cette section, nous présentons deux des concepts les plus importants pour notre approche: marquages de base et *J-vecteurs*. Un marquage de base M_b est un marquage obtenu à partir du marquage initial M_0 avec le déclenchement du mot observé *w* et de toutes les transitions non observables, dont le déclenchement est nécessaire pour permettre *w*. Le j-vecteur $y \in J_{min}(M_0, w)$ est le vecteur de tir d'une séquence de transitions non observables, dont le déclenchement est nécessaire pour atteindre Mb.

Définition 6.5. *Soit* $<N , M_0>$ *un système de RdP où* $N = (P,T, Pre, Post)$ *et* $T = T_o \cup T_u$.

Soit $\sigma \in \mathcal{L}(N, M_0)$ *une séquence de tir et* $w = P_o(\Sigma)$ *le mot observée correspondant. Nous définissons l'ensemble des* justifications de *w comme*

$$J(w) = \{\sigma_u \in T_u^* \mid [\exists \sigma \in \mathcal{L}(w) : \sigma_u = P_u(\sigma)] \wedge \nexists \sigma' \in \mathcal{L}(w) : \sigma'_u = P_u(\sigma') \wedge \pi(\sigma'_u) \leq \pi(\sigma_u)]\}$$

De plus, nous définissons

$$J_{\min}(M_0, w) = \{y \in \mathbb{N}_u^n \mid \exists \sigma_u \in \mathcal{J}(w) : \pi(\sigma_u) = y\}$$

L'ensemble correspondant de vecteurs j.

En termes simples, $\mathcal{J}(w)$ est l'ensemble des séquences de transitions non observables entrelacés avec *w* dont l'activation de déclenchement est *w* et dont le vecteur de déclenchement est minimal. Les vecteurs de déclenchement de ces séquences sont appelés *j-vecteurs*

Définition 6.6. *Soit* $\langle N, M_0 \rangle$ *un système de RdP où* $N = (P, T, Pré, Post)$ *et* $T = T_o \cup T_u$. Soit *w une observation donnée et* $\sigma_u \in \mathcal{J}(w)$ *l'une de ses justifications. le marquage*

$$M_b = M_0 + C_u \cdot C \cdot y + y', \text{ où } y = \pi(\sigma_u),\ y' = \pi(w),$$

C'est-à-dire que le marquage a atteint le tir w entrelacé avec la justification σ_u*, est appelée base de marquage et y est appelée sa* j-vecteur *(ou vecteur de justification).*

Évidemment, comme il existe en général plus d'une justification pour un mot *w* (l'ensemble $\mathcal{J}(w$ n'est généralement pas un singleton), le marquage de base peut ne pas être unique. En outre, deux vecteurs j ou plus peuvent correspondre au même marquage de base.

Notez cependant que, dans des hypothèses appropriées sur le sous-réseau induit par T_u, l'unicité de M_b peut être assurée par l'unicité du vecteur-j. En particulier, dans [31] Corona et al. a prouvé que cela est vrai si le sous-réseau induit par T_u est sans conflit en arrière, car dans ce cas, pour chaque observation *w*, il n'y a qu'une seule justification.

Définition 6.7. *Soit* $\langle N, M_0 \rangle$ *un système de RdP où* $N = (P, T, Pré, Post)$ *et* $T = T_o \cup T_u$ *. Soit* $w \in T_o^*$ *être un mot observé. Nous définissons*

$$\mathcal{M}(w) = (M, y) \mid \exists \sigma \in \mathcal{L}(w): M_0 [\sigma > M \ \wedge M \in M_b \wedge y \in J_{\min}(M_0, w)$$

l'ensemble des couples (base de marquage - par rapport j-vecteur) qui sont compatibles avec $w \in T_o^*$.

Notez que l'ensemble $\mathscr{M}(w)$ ne tient pas compte des séquences de transitions observables qui peuvent avoir réellement déclenchées. Il ne garde que la trace des marquages de base qui peut être atteint et des vecteurs de tir des séquences de transitions non observables qui ont déclenchés pour les atteindre. En effet, ce sont les informations réellement significatives et importantes lors du diagnostic. La notion de $\mathscr{M}(w)$ est fondamentale pour fournir un moyen récursif de calculer l'ensemble des j-vecteurs.

Proposition 6.1. *Etant donné un système de RdP* $<N, M_0>$ *où* $N = (P, T, Pré, Post)$ *et* $T = T_o \cup T_u$ *, dont le sous-réseau induit par* T_u *est acyclique. Soit* $w = w't$ *une observation donnée. L'ensemble* $J_{min}(M_0, w)$ *est défini comme suit :*

$$J_{min}(M_0, w) \subseteq \{y \in \mathbb{N}_u^n | \ y = y' + e \ : \ y' \in J_{min}(M_0, w'), e \in J_{min}(M_b', t)\}$$
$$où\ M_b' = M_0 + C_u \cdot y' + C_o \cdot \pi(w').$$

Preuve: Il résulte de Définitions 6.4, 6.6 et 6.7. En particulier, il résulte du fait que, dans les réseaux de Petri où le sous-réseau est inobservable marquages de base acycliques complètement caractère l'ensemble des marques compatibles, comme il sera montré dans le théorème 6.20.

En termes simples, l'ensemble des j-vecteurs $J_{min}(M_0, w)$ est un sous-ensemble de l'ensemble obtenu en additionnant les j-vecteurs *par* $J_{min}(M_0, w')$ qui conduisent à une *base de marquage* M_b qui soit activer t ou activer une séquence de transitions non observables que soit activer t, ainsi que les vecteurs de $J_{min}(M_b', t)$. C'est un sous-ensemble depuis un vecteur de mise à feu $y \in J_{min}(M_0, w)$ obtenu en sommant les deux vecteurs $y' \in J_{min}(M_0, w')$ *et* $e \in J_{min}(Mb', t)$ pourrait ne pas être minimal, comme le montre l'exemple suivant

Exemple 6.4. Considérons le réseau de Petri de la figure 6.2. Soit $M_0 = [0\ 0]^T$.
Considérons l'observation $w = t_1$. L'ensemble de la justification est $\mathscr{J}(t_1) = \{\varepsilon_1, \varepsilon_2\}$.
Considérons maintenant $w = t_1t_2$. L'ensemble de la justification est $\mathscr{J}(t_1t_2) = \{\varepsilon_2\}$. La séquence de tir $\varepsilon_1\varepsilon_2$ n'est pas une justification, parce qu'il n'est pas minimal puisqu'il couvre ε_2, C'est à dire $\pi(\varepsilon_2) < \pi(\varepsilon_1\varepsilon_2)$.

L'exemple suivant permettra de clarifier les concepts introduits précédemment.

Exemple 6.5. Considérons le réseau de Petri dans la figure 6.4. Ce qui représente une étape très simple dans le processus de transfert d'informations dans les reseau MANET est la trasmission des paquets, Soit M_0 le marquage indiqué sur la

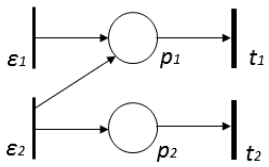


Figure 6.4. Un exemple de réseaux de Petri

Prenons l'observation $w = t_1$. Il détient J (t_1) = {ε} and $J_{\min}(M_0, t_1)$ = {0}, donc la base marquage associé à $w = t_1$ est

$M_b = M_0 + C(\cdot, t_1) = [0\ 2\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1]^T$

et son j-vecteur est $\vec{0}$. *Ainsi M* $(t_1) = \{(M_b, \vec{0})\}$

Maintenant, considérons $w = t_1t_2$. Dans un tel cas, l'ensemble des justifications sont

$$\mathcal{J}(t_1t_2) = \{\varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6\varepsilon_7\varepsilon_8\varepsilon_9\varepsilon_{10},\ \varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6\varepsilon_3\varepsilon_{11}\varepsilon_8\varepsilon_9\varepsilon_{10},$$
$$\varepsilon_8\varepsilon_9\varepsilon_{10}\varepsilon_{13}\varepsilon_5\varepsilon_6,\ \ldots\ldots\}$$

où tous les points dénotent d'autres séquences (qui ne sont pas rapportés ici par souci de concision) qui sont activées au M_b et qui ont le même vecteur de tir des précédents. L'ensemble de j-vecteurs est :

$$J_{\min}(M_1, t_2) = \{\ [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0]^T, [2\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0]^T,$$
$[1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1]^T$, $[0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1]^T$ }.

Maintenant, Soit

$$e_1 = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0]^T,\quad e_2 = [2\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0]^T,$$
$$e_3 = [1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1]^T,\quad e_4 = [0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1]^T\},$$

Les marquages de base atteint après le tir de *w* sont les suivants :

$$M_b^1 = M_b + C_u \cdot e_1 + C(\cdot, t2) = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1]^T$$
$$M_b^2 = M_b + C_u \cdot e_2 + C(\cdot, t2) = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1]^T$$
$$M_b^3 = M_b + C_u \cdot e_3 + C(\cdot, t2) = [0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0]^T$$
$$M_b^4 = M_b + C_u \cdot e_3 + C(\cdot, t2) = [0\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0]^T$$

Ansi $\mathcal{M}(t_1t_2)=\{(M_b^1, e_1), (M_b^2, e_2), (M_b^3, e_3), (M_b^4, e_4)\}$ and $J_{min}(M_0,w)$

et $J_{\min}(M_0, w)J_{\min}(M_1, t_2)$ étant $J_{\min}(M_0, t_1) = \{0\}$.

Maintenant, considérons $w = t_1t_2t_2$. Il est facile de vérifier que

$J_{\min}(M_b^1, t_2)=\{e_3\}$, $J_{\min}(M_b^2, t_2)=\{e_4\}$,

$J_{\min}(M_b^3, t_2) = \{e_1\}$, $J_{\min}(M_b^4, t_2)=\{e_2\}$.

En conséquence, à $w = t_1t_2t_2$ nous n'avons qu'un seul vecteur j, car

$$Mb5=M_b^1+C_u\cdot e_3+C(\cdot, t_2)$$
$$=M_b+C_u\cdot(e_1+ e_3)+2C(\cdot, t_2),$$
$$M_b^6=M_b^2+C_u\cdot e_4+C(\cdot, t_2)$$
$$=M_b+C_u\cdot(e_2+ e_4)+2C(\cdot, t_2),$$
$$M_b^7=M_b^3+C_u\cdot e_1+C(\cdot, t_2)$$
$$=M_b+C_u\cdot(e_3+ e_1)+2C(\cdot, t_2),$$
$$M_b^8=M_b^4+C_u\cdot e_2+C(\cdot, t_2)$$
$$=M_b+C_u\cdot(e_4+ e_2)+2C(\cdot, t_2).$$

En particulier, on estime que $y_1 = e_1 + e_3 = e_2 + e_4$. Ont outre,

$M_b^5= M_b^6= M_b^7= M_b^8= [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0]^T$,

il n'y a donc qu'un seul marquage de base au $w = t_1t_2t_2$.

Enfin, $\mathcal{M}(t_1\ t_2\ t_2) = \{(M_b^5, y_1)\}$ et $J_{\min}(M_0, w) = \{y_1\}$.

L'ensemble $M(w)$, qui inclut tous les couples (marquage de base – *vecteur-j* relatif) qui sont compatibles avec une observation w, peut facilement être construit en utilisant la procédure informelle présentée dans l'exemple précédent. L'algorithme suivant formalise cette procédure. Notez que, comme pour l'algorithme 6.1 le réseau de Petri utilisé comme entrée doit avoir le sous-réseau acyclique induit par T_u.

Algorithme 6,2. [Calcul des marquages de base et J-vecteurs]

***Entrée**: - un ⟨N , M_0⟩ réseau de Petri, dont le sous-réseau inobservable est acyclique,*
- l'ensemble des transitions non observables T_u,
-le mot observée w.

***Sortie**: M (w).*

1. Soit $w = \varepsilon$.
2. Soit $M(w) = \{(M_0, 0)\}$.
3. Attendez jusqu'à ce qu'une nouvelle transition t se déclenche.

4. Soit $w' = w$ et $w = w't$.
5. Soit M (w) =ø ;
6. ***Pour*** *tous M' tel que $(M', y') \in M(w')$,* ***faire***
6.1. ***Pour*** *tout $e \in J_{min}(M', t)$* ***faire***
6.1.1. Soit $M = M' + C_u \cdot e + C(\cdot, t)$,
6.1.2. ***Pour*** *tous y' tel que $(M', y') \in M(w')$* ***faire***
6.1.2.1. Soit $y = y' + e$,
6.1.2.2. Soit $M(w) = M(w) \cup \{(M, y)\}$.
Fin pour
Fin Pour
Fin pour
7. Retirer de M (w) tout couple (M, y) pour lesquels il existe une autre paire (M', y') tel que y couvre y'.
8. Aller à l'étape 3.

En termes simples, l'algorithme ci-dessus qui est basé sur la proposition 6.14 peut être expliqué comme suit. Nous supposons qu'un certain mot *w* (qui est égale à la chaîne vide à l'étape initiale) a été observé. Ensuite, une nouvelle transition t se tir. Nous considérons toutes les marques de base à l'observation *w'* avant la mise à feu de *t*, et nous sélectionnons parmi eux ceux qui peuvent avoir autorisé le tir de *t*, en tenant également compte du fait que cela peut avoir besoin des tirs de séquences appropriées de transitions non observables. En particulier, nous nous concentrons sur les explications minimales, et donc sur les e-vecteurs minimaux correspondant (étape 6.1). Enfin, nous mettons à jour l'ensemble *M(w)*, y compris tous les couples de nouveaux marquages de base et J-vecteurs, en tenant compte du fait que pour chaque base de marquage au *w'* il peut correspondre plus d'un j-vecteur.

Introduisons maintenant le résultat suivant qui sera utile dans le reste du these.

Définition 6.8. Soit $\langle N, M_0 \rangle$, un système de RdP où $N = (P, T, Pré, Post)$ et $T = T_o \cup T_u$. Soit $w \in T_o^*$ un mot observée. Nous noterons

$$M_{base}(w) = \{M \in \mathbb{N}^m \mid \exists y \in \mathbb{N}_u^n \text{ and } (M, y) \in M(w)\}$$

l'ensemble des marquages de base à *w*. En outre, on note que

$$M_{base} = \bigcup_{w \in T_0^*} M_{base}(w)$$

l'ensemble de toutes les marquages de base pour toute observation *w* possible.

Fait 6.1. Compte tenu d'un système RdP borné $\langle N, M_0 \rangle$ avec $N= (P, T, Pré, Post)$ et $T=T_o \cup T_u$, l'ensemble de M_{base} est fini.

Preuve: Il résulte de ce fait que l'ensemble des marquages de base est un sous-ensemble de l'ensemble de l'accessibilité, ce qui est fini parce que le système réseau est borné.

Nous concluons cette section, en montre que notre approche fondée sur les marquages de base est capable de caractériser complètement l'ensemble de l'accessibilité sous observation partielle.

Nous commençons avec un résultat qui caractérise les séquences de tir. Dans le théorème suivant, nous montrons qu'une séquence $\sigma\tilde{}$ est cohérente avec l'observation w si et seulement s'il existe une séquence équivalente (par exemple, une séquence ayant la même séquence observée) qui est la concaténation de deux séquences : une première atteint la base de marquage $M(w)$ et le second contient uniquement les transitions non observables.

Théorème 6.2

Considérons un système de RdP $\langle N, M_0 \rangle$ dont le sous-réseau non observable est acyclique.

Il existe une suite $\tilde{\sigma} \in T^*$ telle que $M_0[\tilde{\sigma}\rangle \tilde{M}$ à projection observable $P_o(\tilde{\sigma})=w$ et vecteur de déclenchement $\pi(\tilde{\sigma}) = \tilde{y}$ si et seulement s'il existe aussi un couple $(M,y) \in M(w)$et une séquence inobservable $\sigma'' \in T_u^*$ telle que $M[\sigma''\rangle \tilde{M}$ et $\tilde{y} = \pi(w)+y+\pi(\sigma'')$.

Preuve : Nous prouvons ce résultat par induction sur la longueur sur le mot observée w.

(Étape de base) Pour $w = \varepsilon$, le résultat est évidemment valable.

(Étape inductive) Supposons que le résultat soit valable pour $v \in T_o^*$. Nous prouvons qu'il est valable pour $w=vt$ où $t \in T_o$.

(Seulement si). S'il existe une séquence $\tilde{\sigma} \in T^*$ telle que M_0 [$\sigma\tilde{}\rangle$M alors il existe des séquences σ' et σ'' telles que.

$$M_0[\sigma'\rangle M'[t\rangle M''[\sigma''\rangle \tilde{M}$$

Où $P_0(\sigma')= v$, $\pi(\sigma'' \epsilon T_u^*$. Par induction, il existe $(M, y) \in M(v)$ tel que

$$M_0[\sigma_a'\rangle \mathrm{M}[\sigma_b'\rangle M'[t\rangle M''[\sigma''\rangle \tilde{M}$$

6.4 États de diagnostic

Considérons un système modélisé comme un réseau P / T dont l'ensemble de transitions est partitionné en l'ensemble des transitions observables et inobservables, c'est-à-dire que $T=T_o \cup T_u$.

Supposons qu'un certain nombre de comportements anormaux (ou de faute) peuvent se produire dans le système. L'apparition d'un comportement de défaut correspond au déclenchement d'une transition non observable, mais il peut également y avoir d'autres transitions non observables, mais dont le déclenchement correspond à des comportements réguliers.

Ensuite, supposons que les comportements de défaut peuvent être divisés en *r* principales classes (classes de défaut), et nous ne sommes pas intéressés à distinguer les événements de défaut dans la même classe.

Ceci peut être facilement modélisé en termes de réseau de Petri en supposant que l'ensemble des transitions non observables est partitionné en deux sous-ensembles, à savoir:

$$T_u = T_f \cup T_{reg}$$

Où T_f comprend toutes les transitions de défaut et T_{reg} inclut toutes les transitions relatives à des événements non observables, mais réguliers. L'ensemble T_f est en outre partitionné en sous-ensembles de *r*, à savoir,

$$T_f = T_f^1 \cup T_f^2 \cup T_f^3 \cup \ldots\ldots \cup T_f^r$$

où toutes les transitions du même sous-ensemble correspondent à la même classe de défauts. Nous dirons que la i-ème faute s'est produite lorsqu'une transition dans T_f^i s'est déclenchée.

Dans la sous-section suivante, nous introduisons la définition du diagnostiqueur et son état.

6.4.1 Définitions basiques

Définition 6.9. *Un* diagnostiqueur *est une fonction* $\Delta: T_0^* \times \{T_f^1, T_f^2, \cdots, T_f^r, \} \rightarrow \{0, 1, 2, 3\}$ *qui associe à chaque observation w et à chaque classe de défauts* T^i, $i = 1,\ldots, r$, *un* état de diagnostic .

$\Delta\ (w, T_f^i,) = 0$ *si pour tout* $\sigma \in \mathscr{L}(w)$ *et pour tout* $t_f \in T_f^i$ *il retient* $t_f \notin \sigma$.

Dans un tel cas, le $i^{ème}$ *défaut ne peut pas s'être produit, car aucune des séquences de déclenchement cohérentes avec l'observation ne contient de transitions d'erreur de classe i.*

$\Delta\ (w, T_f^i) = 1$ *si:*

(i) il existe $\sigma \in \mathcal{S}(w)$ *et* $t_f \in T_f^i$ *tel que* $t_f \in \sigma$ *mais*

(ii) pour tout $\sigma \in \mathcal{L}(w)$ *et pour tout* $t_f \in T_f^i$, *on dit que* $t_f \notin \sigma$.

Dans un tel cas, une transition d'erreur de classe i peut s'être produite mais ne figure dans aucune justification de w.

$\Delta\ (w, T_f^i) = 2$ *s'il existe* $\sigma, \sigma' \in \mathcal{L}(w)$ *tel que:*

(i) il existe $t_f \in T_f^i$ *tel cette* $t_F \in \sigma$ *;*

(ii) pour tout $t_f \in T_f^i$, $t_f \notin \sigma'$.

Dans un tel cas, une transition d'erreur de classe i est contenue dans une justification (mais pas dans toutes) de w.

$\Delta\ (w, T_f^i) = 3$ *si pour tout* $\sigma \in \mathcal{S}(w)$ *il existe* $t_f \in T_f^i$ *tel que* $t_f \in \sigma$.

Dans ce cas, il faut que la itième faute se soit produite, car toutes les séquences indésirables compatibles avec l'observation contiennent au moins une transition d'erreur de classe i.

Les états de diagnostic 1 et 2 correspondent tous deux à des cas dans lesquels une défaillance peut être survenue mais ne s'est pas nécessairement produite. La principale raison de les distinguer est la suivante. Dans l'état 1, le comportement observé ne suggère pas qu'une erreur s'est produite, car toutes les séquences minimales conduisant à w sont exemptes d'erreur. Au contraire, dans l'état 2, au moins une des justifications du comportement observé contient une transition dans la classe.

Notez que dans la pratique, l'état de diagnostic 1 représente une situation courante dans de nombreuses applications réelles. Par exemple, une soupape dans une usine chimique peut être brisée à tout moment. Ainsi, tous les états atteints sans défaut ne tombent jamais dans l'état de diagnostic 0 mais dans l'état de diagnostic 1.

Exemple 6.6.

Considérez le système de réseau illustré à la figure 6.3. Supposons que deux comportements d'erreur différents (classes d'erreur) peuvent survenir durant l'envoi du message.: (1) un paquet

est déplacée vers le sens inverse ou malencontreusement transféré sur le premier canal par le routeur $R2$ (T_f^1 = { ε_{11} , ε_{12} }); (2) un paquet d'un type différent (par exemple, une autre sources) entre dans le canal 2 (T_f^2 = { ε_{13} }).

Enfin, que toutes les autres transitions non observables appartiennent à l'ensemble T_{reg} , c'est T_{reg} = { ε_3, ε_4,. . . , ε_{10} }.

Considérons $\omega = \varepsilon$. Comme indiqué dans l'exemple 6.16, il est vrai que $\mathcal{J}(\varepsilon) = \mathcal{S}(\varepsilon) = \{\varepsilon\}$. Ensuite, nous pouvons observer que la transition $\varepsilon_{13} \in T_f^2$ peut se déclencher à M_0, tandis que les autres transitions de défaut ne sont pas activées à M_0. Donc, Par conséquent, nous concluons que $\Delta(\varepsilon, T_f^1) = 0$ et $\Delta(\varepsilon, T_f^2) = 1$.

Maintenant, considérons $\omega = t_1$. Comme cela a déjà été discuté dans l'exemple 6.16 , il s'agit de $\mathcal{J}(t_1) = \{ \varepsilon \}$ et aucune transition de faute ne peut donc être contenue dans une justification de w . Au contraire, toutes les transitions de faute sont contenues dans au moins une séquence dans $\mathcal{S}(t_1)$. Ainsi, $\Delta(t_1, T_f^1) = \Delta(t_1, T_f^2) = 1$.

Concentrons-nous maintenant sur l'observation $\omega = t_1 t_2$. En regardant l'exemple 6.4, il est facile de conclure que $\Delta(t_1 t_2, T_f^1) = \Delta(t_1 t_2, T_f^2) = 2$. En fait, toutes les transitions de défaut sont contenues dans au moins une séquence en $\mathcal{J}(t_1 t_2)$, mais il existe aussi des justifications de t_1 t_2 qui ne contiennent pas de transitions de faute.

Finalement, soit $\omega = t_1 t_2 t_2$. Dans un tel cas, il tient $\Delta(\omega, T_f^1) = \Delta(\omega, T_f^2) = 3$ car comme il peut être facilement vérifié, toutes les justifications de w contiennent des transitions des deux classes.

Le calcul en ligne des ensembles $\mathcal{S}(w)$ et $\mathcal{J}(w)$ peut être très exigeant en calcul dans les systèmes à grande échelle. Nous proposons donc ci-après deux procédures alternatives pour calculer les états de diagnostic. Ces procédures sont basées sur les notions d'explications minimales, de vecteurs minimaux et de marquages de base, présentées dans les sections suivantes. Cette procédure s'applique aux systèmes de réseau dont le sous-réseau non observable est acyclique.

Définition 6.9. Soit $\langle N, M_0 \rangle$un système de réseau avec une fonction de marquage: $T \rightarrow L \cup \{\varepsilon\}$, où N = (P, T, Pré, Post) et $T = T_o \cup T_u$. Soit $w \in L$ un mot observé. Nous définissons

$$\mathcal{M}(w) = \{(M, y) \mid (\exists \sigma \in \mathcal{S}(w) : M_0[\sigma\rangle M) \wedge (\exists(\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w) : \sigma_o = P_o(\sigma), \sigma_u = P_u(\sigma), y = \pi(\sigma_u))\}$$

L'ensemble des couples (marquage de base ; vecteur j relatif) compatibles avec w L^*

Notez que l'ensemble M (w) ne tient pas compte des séquences de transitions observables qui peuvent avoir réellement été déclenchées. Il ne garde en mémoire que les repères de base pouvant être atteints et les séquences de transitions non observables qui ont été déclenchées pour les atteindre. En effet, ce sont les informations vraiment importantes lors du diagnostic.

6.4.2 Caractérisation des états de diagnostic

Dans cette sous-section, nous fournissons des résultats qui nous permettent de caractériser les états de diagnostic à partir de la connaissance de l'ensemble $\mathscr{M}(w)$. La proposition suivante nous permet d'estimer la valeur d'un état de diagnostic, atteinte après l'observation d'un mot *w*, à partir de l'analyse de l'ensemble $\mathscr{M}(w)$ défini dans la définition 6.23.

Proposition 6.1. *Considérons un mot observé* $w \in T_o^*$.

$\Delta(w, T_f^i) \in \{0, 1\}$ *si pour tous* $(M, y) \in M(w)$ *et pour tous* $t_f \in T_f^i$ *ça tiens* $y(t_f)=0$.

$\Delta(w, T_f^i) = 2$ *si il existe* $(M, y) \in M(w)$ *et* $(M', y') \in M(w)$ *tel que:*

(i) *il existe* $t_f \in T_f^i$ *tel que* $y(t_f) > 0$,

(ii) *pour tout* $t_f \in T_f^i$, $y'(t_f) = 0$.

$\Delta(w, T_f^i) = 3$ si *pour tout* $(M, y) \in M(w)$ *il existe* $t_f \in T_f^i$ *tel que* $y(t_f) > 0$.

Preuve: Cela découle trivialement de la définition des états de diagnostic et du théorème 6.20 .

À partir de la seule analyse de *M* (w), il est possible de déterminer les états 2 et 3, tandis qu'une distinction entre les états 0 et 1 est nécessaire. La proposition suivante montre comment les états 0 et 1 peuvent être distingués en ce qui concerne l'accessibilité du réseau non observable.

Proposition 6.2. *Considérons un mot observé* $w \in T_o^*$ *telle que* $\forall (M, y) \in M(w)$ *et* $\forall t_f \in T_f^i$

ça tiens $y(t_f) = 0$.

$\Delta(w, T_f^i) = 0$ *si* $\forall (M, y) \in M(w)$ *et* $\forall t_f \in T_f^i$ *il n'existe pas de séquence* $\sigma \in T_u^*$ *tel que* $M[\sigma\rangle$ *et* $t_f \in \sigma$.

$\Delta\ (w, T_f^i) = 1$ *si* ∃ *au moins un* $(M, y) \in M(w)$ *et une séquence* $\sigma \in T_u^*$ *tel que pour au moins un* $t_f \in T_f^i$, $M[\sigma\rangle$ *et* $t_f \in \sigma$.

Preuve :Il découle du fait que, selon la proposition 6.24, (w, T_f^i) {0,1} si toutes les justifications minimales de *w* ne contiennent aucune transition de faute de classe *i*. Cependant, selon la définition 6.22 l'état de diagnostic est égal à zéro si, à chaque base, le marquage *M* sur *w* n'est pas activé. Au contraire, l'état de diagnostic est égal à un si au moins une transition d'erreur de classe *i* est activée sur une base en indiquant *M* en *w*.

Si le sous-réseau non observable est acyclique, la proposition suivante nous permet de distinguer les états 0 et 1 en résolvant un problème trivial de programmation linéaire en nombres entiers.

Proposition 6.3. Pour un réseau de Petri dont le sous-réseau inobservable est acyclique, soit w ∈ T_0^* être un mot observé de telle sorte que pour tous (*M, y*) ∈ M (w) ça tiens

$y(t_f) = 0\ \forall\ t_f \in T_f^i$.

Laisser nous considérer la contrainte ensemble

$$l(M) = \begin{cases} M + C_u.z \geq \vec{0} \\ \sum_{t_f \in T_f^i} z(t_f) > 0 \\ z \in \mathbb{N}^{n_u} \end{cases}$$

$\Delta\ (w, T_f^i) = 0$ *si* ∀ $(M, y) \in M(w)$ *l'ensemble de contraintes (6.3) n'est pas réalisable*

$\Delta\ (w, T_f^i) = 1$ *si* ∃ $(M, y) \in M(w)$ tel que l'ensemble de contraintes (6.3) soit réalisable

Preuve: découle de la proposition 6.25 et du fait que si le sous-réseau non observable est acyclique, l'ensemble de contraintes (6.3) caractérise l'ensemble d'accessibilité du réseau non observable. Ainsi, il existe une séquence contenant une transition $t_f \in$

T_f^i firable en *M* sur le sous-réseau non observable si et seulement si *T (M)* est réalisable.

Exemple 6.7. Considérons à nouveau le système réseau de la figure 6.3.

Soit *w* = *ε*. Ça tiens *M (ε)* = {(M_0,$\vec{0}$)} donc de la proposition 6.24, (ε, T_f^1) = Δ (ε, T_f^2) = {0,1}.

Pour déterminer complètement les états de diagnostic, nous devons vérifier si le jeu de contraintes entier défini dans la proposition 6.26 admet des solutions. Ce n'est pas vrai dans le cas de la première classe, alors que c'est le cas pour la deuxième classe. Nous concluons donc que (ε, T_f^1) = 0 et (ε, T_f^2) = 1, conformément à l'exemple 6.23.

Nous revenons maintenant au premier paragraphe de ce chapitre, continuons à expliquer comment combiner entre la programmation linéaire que nous utilisons en diagnostic expliqué dans les parties précédentes et la logique floue

Soit maintenant la proposition logique élémentaire : $\boldsymbol{d_i \rightarrow d_k}$ définissant une propagation de défaut.
Nous pouvons énoncer la proposition suivante :

Proposition 6.4 : Si le signal $!s_i$ de type Symptôme, associé à un défaut de base d_i n'est pas envoyé vers le RdPSyncF, la transition temporelle qui modélise l'implication élémentaire $d_i \rightarrow d_k$ n'est pas franchissable.

La transition qui modélise l'implication logique $d_i \rightarrow d_k$ est une transition temporelle. Si le signal $!s_i$ est envoyé à l'instant τ, il sert de signal de sensibilisation pour la transition temporelle. Une transition qui ne reçoit pas le signal $?s_i$ est une *transition inactive*.

La transition correspondant à un ensemble de places (défauts) concurrentes présente les propriétés suivantes :

Proposition : Les formules qui généralisent le calcul de la valeur floue pour la place de sortie et pour la transition équivalente, s'appliquent uniquement pour les couples places/transitions qui reçoivent les signaux de synchronisation.

Proposition : Si une transition n'a pas au moins un défaut de base en amont, elle ne reçoit pas de signal de type symptôme et le degré de vérité associé est la fonction

constante μ=1. Cette transition a le comportement d'une transition ordinaire correspondant à un raisonnement logique classique.

Proposition : Si une transition $t_i \in T$ reçoit en entrée au moins deux signaux si et sj , un coefficient de vérité équivalent ($\mu = \mu_{equiv}$) est associé à la transition des places concurrentes (Figure 6.4). Cette transition matérialise la conjonction : di dj → dk

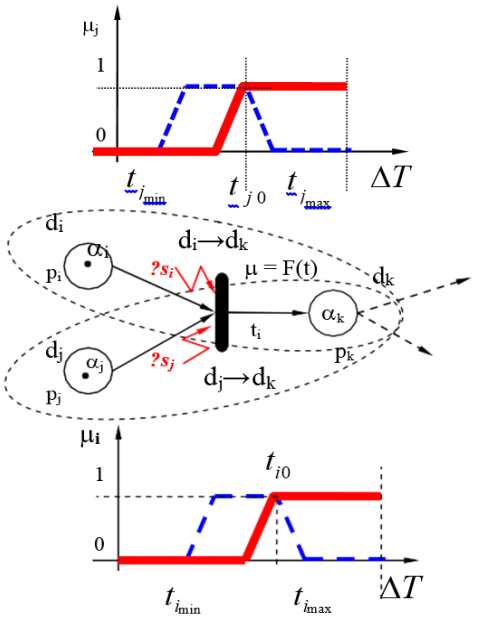


Figure 6.5. Modélisation des raisonnements logiques concurrents $d_i \wedge d_j \rightarrow d_k$ par RdPSynF

6.5.Une approche générale de diagnostic

Sur la base des résultats présentés dans la section précédente 6.4.2, si le réseau induit par T_u est acyclique, le diagnostic peut être effectué simplement en regardant l'ensemble *M (w)* pour tout mot observé *w* et, si l'état du diagnostic soit 0 ou 1, en évaluant en outre si l'ensemble de contraintes entier correspondant (6.3) admet une solution.

L'algorithme proposé pour l'élaboration du *RdPSynF*, outil spécialisé dans l'analyse temporelle floue de la propagation des défauts, fait appel aux étapes suivantes :

Les principales étapes de la procédure sont résumées dans l'algorithme suivant.

Algorithme 6.3. ***[Une approche générale du diagnostic]***

Entrée *:*

- *un réseau de Petri ⟨N, M0> dont le sous-réseau inobservable est acyclique,*
- *Modélisation du système surveillé par RdPSynF*
- *Détection directe modélisée par des mécanismes de chien de garde (CG), extensions du modèle RdPF du système.*
- *l'ensemble des transitions inobservables Tu,*
- *les classes de défauts $\{T_i^f\}$ i = 1, ..., r,*

- *le mot observé w.*

***Sortie** : les états de diagnostic.*

1. *Soit* $w = \varepsilon$.

2 *Soit* $M(w) = \{(M_0, \vec{0})\}$.

3 *Attendez qu'une nouvelle transition t déclenche.*

4 *Soit* $w' = w$ *et* $w = w' t$.

5 *Soit* $M(w) = \emptyset$. ***[Calcul de*** $M(w)$ ***]***

6 *Pour tout* M' *tel que* $(M', y') \in M(w')$ *, faire*

mettent en évidence la relation de causalité temporelle entre les défauts, relation qui permet d'associer à chaque transition qui modélise la règle, un intervalle de sensibilisation correspondant à la fenêtre temporelle $[ti_{\min}, ti_{\max}]$ *pendant laquelle, après l'occurrence d'un défaut, un défaut dérivé est susceptible d'apparaître*

6.1. *pour tout* $e \in J_{\min}(M', t)$ *, faire*

6.1.1. *soit* $M = M' + C_u \cdot e + C(\cdot, t)$,

6.1.1.1. *pour tout* y' *tel que* $(M', y') \in M(w')$ *, faire*

6.1.1.1.1. *Soit* $y = y' + e$

6.1.1.1.2. *Soit* $M(w) = M(w) \cup \{(M, y)\}$.

7. *Pour tout* $i = 1, \ldots, r$ *, faire* ***[Calcul de l'états diagnostic]***

En partant du réseau RdPFS obtenu, on identifie les implications logiques élémentaires, modélisées par des transitions. L'implication logique élémentaire met en évidence la "contribution" de chaque variable logique de l'antécédence dans la valeur de vérité de la conséquence.

7.1. *si* $\forall (M, y) \in M(w)$ *et* $\forall t_F \in T_f^i$ *il retient* $y(t_f) = 0$*, faire*

7.1.1. *si* $\forall (M, y) \in M(w)$ $T(M)$ *n'est pas faisable, faire*

7.1.1.1. *Soit* $\Delta(w, T_f^i) = 0$,

7.1.2. *autre*

7.1.2.1. *Soit* $\Delta(w, T_f^i) = 1$,

7.2. *si* $\exists (M, y) \in M(w)$ *et* $(M', y') \in M(w)$ *tel cette:*

(i) $\exists t_f \in T_f^i$ *tel que* $y(t_f) > 0$*, (ii)* $\forall t_f \in T_f^i$ *,* $y'(t_f) = 0$ *, faire*

7.2.1. *Soit* $\Delta(w, T_f^i) = 2$,

7.3. *si* $\forall (M, y) \in M(w)$ $\exists t_f \in T_f^i$ *tel que* $y(t_f) > 0$*, faire*

7.3.1. *Soit*$\Delta(w, T_f^i) = 3$.

8 *Aller à l'étape 3*

En termes simples, les étapes 1 à 6 coïncident avec celles de l'algorithme 6.3, tandis qu'à l'étape 7, nous calculons les états de diagnostic à l'aide des propositions 6.1 et 6.3.

La dynamique du marquage du réseau RdPSyncF, ne peut pas être évaluée en appliquant l'équation fondamentale du RdP générique. Ceci est du au fait que le réseau RdPSyncF comporte un aspect atypique du point de vue de la propagation des valeurs de crédibilité. Nous nous intéressons donc à l'algorithme d'évolution du marquage du RdPSyncF, en fonction des contraintes temporelles liées à chaque transition, ainsi que des signaux de synchronisation du RdPSyncF.

Les valeurs de crédibilité pour chaque défaut se trouvent en dépendance directe des valeurs (α_i, μ_i) injectées par les signaux ?si.

L'outil RdPSyncF est dédié à la modélisation des implications logiques qui enchaînent des variables logiques par l'intermédiaire des opérateurs logiques ET ou OU. Nous avons montré qu'il n'est pas adapté à la modélisation complète des raisonnements logiques. Il est par contre spécialisé dans la surveillance des systèmes de production et, plus particulièrement, la modélisation des règles logiques qui décrivent la propagation des défauts.

L'algorithme proposé pour l'élaboration du RdPsyncF, outil spécialisé dans l'analyse temporelle floue de la propagation des défauts, fait appel aux étapes suivantes :

- ✓ Modélisation du système surveillé par RdPSyncF.
- ✓ Détection directe modélisée par des transitions observables (capteurs), extensions du modèle RdPSyncF du système.
- ✓ Fuzzyfication de l'instant de franchissement $\tau_{Symptome}$ de la transition observale. Les jetons du cette transition du RdPSyncF deviennent des objets ayant comme attribut le nombre flou μ_d ($\tau_{Symptome}$).
- ✓ L'apparition des symptômes dans les places, implique l'émission des signaux de synchronisation !si. Ces signaux, émis par les places du RdPTO, sont réceptionnés comme signaux duaux ?si par les transitions du RdPSyncF. Les signaux !?si sont chargés des informations $\alpha_i = \mu_d$ ($\tau_{Symptome}$) représentant la gravité du défaut (dégradation) enregistré.
- ✓ Les implications élémentaires mettent en évidence la relation de causalité temporelle entre les deux défauts, relation qui permet d'associer à chaque transition qui modélise la règle, un intervalle de sensibilisation correspondant à la fenêtre temporelle $[ti_{min}, ti_{max}]$ pendant laquelle, après l'occurrence d'un défaut, un défaut dérivé est susceptible d'apparaître.
- ✓ Pour chaque implication logique élémentaire, on propose une modélisation floue de la force de liaison entre les variables impliquées. Cette description floue est en fonction de l'instant d'arrivée du $\tau_{Symptome}$. La règle logique modélisée reçoit la valeur de crédibilité : $\mu_i = \mu_a (\tau_{Symptome})$
- ✓ En appliquant le modus ponens généralisé, on peut calculer la valeur de crédibilité de la conséquence de cette règle : $\alpha_k = (\alpha_j \cdot \alpha i) \cdot \mu_k$

Une simulation d'un exemple sera présentée dans le dernier chapitre de cette thèse.

Nous mentionnons *Diag-R* comme une abréviation de l'algorithme global de diagnostic de routage proposé dans ce chapitre. Cette abréviation sera utilisée dans le chapitre suivant comme appel à la fonction de diagnostic dans le processus de routage.

6.6. Conclusion

Dans ce chapitre nous nous sommes intéressés au diagnostic des SED. Pour cela, les RdP synchronisé sont utilisés pour modéliser le système avec ses capteurs. Le modèle représente à la fois le comportement fautif et normal du système. Comme nous pouvons le voir sur le schéma représenté sur la Figure 6.3, la démarche de diagnostic peut se décomposer en trois parties : les explications minimales et e-vecteurs minimaux pour chaque observation transmettent par l'ensemble des transitions observables, la caractérisation de l'état de système à travers le traitement de contenus des vecteurs d'explication et à la fin l'application de la logique floue afin de préciser la nature de défaut et son degré. Le résultat obtenu dans ce chapitre sera exploité dans le chapitre suivant dans le but de proposé un protocole de routage pour les systèmes de communication ad hoc MANET, et qui résolu à la fois le problème de routage ainsi que la surveillance de l'état des nœuds et des liens de réseau.

Chapitre 7

Un protocole de routage, de diagnostic et surveillance d'un système de communication ad hoc MANET

Résumé

Le réseau de Petri à synchronisation floue (RdPSyncF) est largement adopté dans la modélisation des fonctions de routage et de détection / décision utilisant une approche de transition synchronisée floue, où la methode ant-systèmes est utilisé pour trouver la meilleur solution au problème de routage. La technique proposée au chapitre précèdent est incluse dans le présent protocole afin d'améliorer ses performances en cas d'erreurs et de défauts. Les résultats obtenus montrent l'efficacité du protocole SyncFAnt (Fuzzy Ant System) synchronisé proposé par rapport à quatre protocoles. Le protocole de routage SynFAnt améliore le rapport de livraison des paquets, le débit, le délai de bout en bout et le taux d'acceptation des flux de qualité de service.

7.1. Introduction

Dans ce chapitre, nous présenterons, dans une première partie, notre vision conceptuelle permettant la mise en œuvre d'un protocole de routage, pour cela le travail présenté dans [124] fournira un bon support pour développer notre protocole car il est adapté à la modélisation des bases d'informations statiques. Cette approche est efficace pour le contrôle mais n'est pas satisfaisante pour la surveillance et le contrôle du flux dynamique des données. Ainsi, nous proposons la synchronisation floue optimisée par la methode ant-system, comme une extension du RdPF pour la modélisation et le diagnostic / surveillance des systèmes de communication mobile ad hoc.

Dans les MANET pour exécuter les applications efficacement, tous les nœuds doivent avoir le niveau seuil des paramètres de qualité. La méthode proposée, SyncFAnt, considère la bande passante, la Durée de vie des liens et la fiabilité en tant que paramètres de qualité de service.

Dans ce paragraphe, nous allons utiliser les concepts suivants :

- *W* est une matrice d'entrée, qui contient des informations de pondération des emplacements d'entrée pour une transition. Dans la matrice, un élément $W_{ij} \in [0,1]$ est la pondération de la place P_i à la transition T_j
- *U* est une matrice de sortie, qui contient les informations de facteur de certitude (μ) d'une transition .Un élément de la matrice *U* représente l'influence de la transition t_j sur la valeur de confiance du place de sortie p_i, c'est à dire U : T$*$ $P \rightarrow [0,1]$.
- *Th* est une matrice de sortie contenant des informations sur le seuil de transition. Un élément $\tau_{ij} \in [0,1]$ de la matrice *Th* est la valeur du seuil de transition t_j pour la sortie de place p_i, , c'est-à-dire *Th* : O $\rightarrow [0,1]$.
- *M* est l'état de marquage du DFPN, qui possède les valeurs de confiance dynamiques des places en entrée. $M_0 = (\propto (p_1), \propto (p_2), \propto (p_3), \dots \dots \propto (p_m))^T$ est le marquage initial de RdPSyncF.

7.2. Description du protocole de routage

Cette section présente notre vision conceptuelle de la mise en œuvre d'un protocole de routage basé sur Ant Systems et le réseau de Petri flou. En d'autres termes, il faut déterminer un chemin entre les différents éléments du réseau pour envoyer un message - des paquets - entre deux éléments qui ne communiquent pas directement entre eux. La figure 7.1 présente un réseau ad hoc et la représentation équivalente avec les réseaux de Petri. Chaque hôte du réseau doit avoir les éléments suivants pour mettre en œuvre un réseau ad hoc basé sur un raisonnement flou

- ✓ Agent fournit le meilleur voisin de l'hôte parmi tous ses voisins. Le champ précédent sert à connaître l'hôte qui a envoyé le dernier paquet.

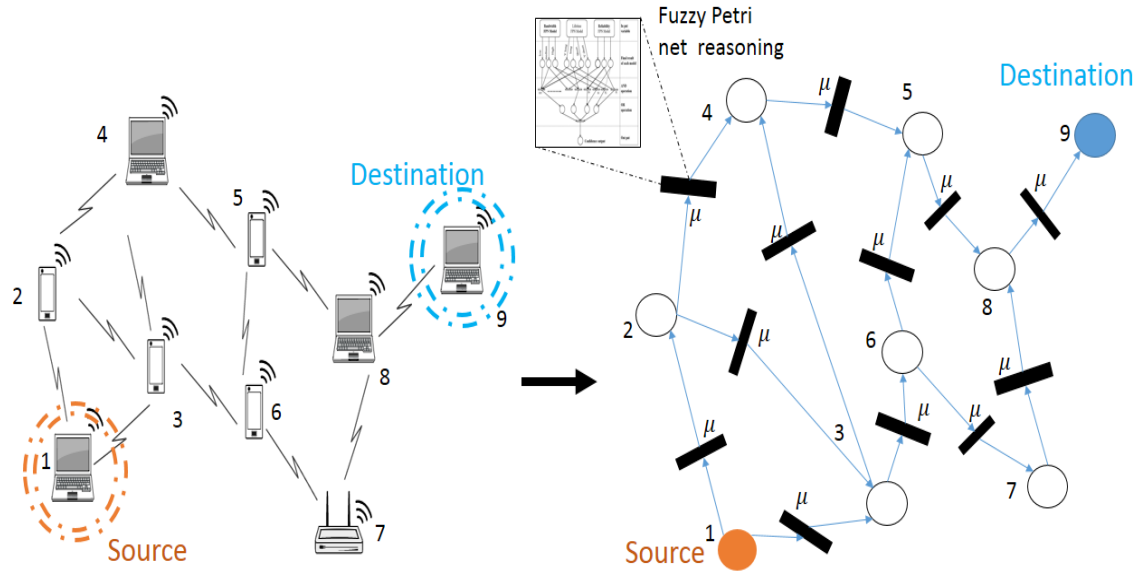


Figure 7.1. Un réseau mobile ad hoc et la représentation équivalente

- ✓ Une liste de tous les hôtes et les distances entre les nœuds voisins. C'est une fonction appelée BestHop (décrite à la Section 7.2.2.1) et son rôle principal est d'utiliser des règles floues pour sélectionner le meilleur nœud voisin S parmi tous ses autres voisins.

Dans le réseau ad hoc, les durées de temporisation dans le model de réseau de Petri correspondant sont différentes pour chaque transition et correspondent à la durée de la communication. Une évolution du marquage sensiblement différente est proposée si les ressources d'un système de communication sont modélisées. Après la prise de vue de la transition, la valeur floue $\alpha_k = \alpha_i . \mu_j$ sera associée à chaque jeton. Chaque jeton aura indépendamment le statut de jeton réservé pendant l'intervalle [*0* d_i] associé à la transition.

7.2.1. RdPSynF pour la modélisation des ressources de systèmes de communication

Dans cette partie, une description des métriques associées au problème de routage dans notre protocole est présentée.

A. modèle de la bande passante

Pour assurer la qualité de service [90-92], nous estimons la bande passante disponible, pouvant les allouer. L'estimation doit être répartie et prendre en compte les différents phénomènes

propres aux réseaux ad hoc (interférences, collisions, etc.). Le principe de base de la plupart des protocoles est que la surveillance des nœuds aide la radio à déterminer le taux d'occupation d'un canal radio. Ensuite, une estimation de la synchronisation du mobile émetteur et récepteur, ainsi que du taux de collision, est effectuée pour déterminer les liaisons à largeur de bande résiduelle. Cependant, l'estimation de ces protocoles ne prend pas en compte le type de transmission (ou la meilleure qualité de service) dans le réseau. Nous l'ajoutons donc au protocole AODV [80], qui permet de différencier les hôtes Hopcount et les hôtes voisins en tant que deux critères du modèle de bande passante.

- ✓ Les hôtes qui ont peu de voisins sont les mieux adaptés car ils ont un débit de communication élevé.
- ✓ Le flux le plus efficace se situe entre les deux nœuds les plus proches, car l'énergie requise pour effectuer une opération entre deux nœuds proches est inférieure à celle d'une autre distance. En conséquence, nous nous référons à la qualité du système et à la conservation de l'énergie.

B. La durée de vie d'un lien

Dans le deuxième bloc de contrôle, nous avons utilisé la logique floue pour estimer la durée de vie (life time) LT pour chaque route. Par conséquent, nous nous concentrons sur deux critères : le nombre de paquets envoyés par la route et la plus faible énergie consommée par les nœuds de route.

- ✓ L'un des critères les plus importants dans l'optimisation de la durée ou le temps d'envoi consiste à contrôler le nombre de paquets envoyés par un nœud à la fois et le nombre de paquets reçus. Ce facteur est important pour évaluer la qualité des liens. Si l'un des nœuds est encombré par un certain nombre de paquets, la bande passante partagée serait réduite de manière à ce que l'unité devienne instable et perde des paquets.
- ✓ Pour la consommation d'énergie, qui est importante dans le domaine du routage, nous considérons le canal de caractéristique énergétique comme moyen d'évaluation. Le taux de consommation d'énergie est le paramètre utilisé pour calculer la condition d'alimentation d'un nœud arbitraire. La méthode utilisée est simple et détaillée dans [125].

C. La fiabilité

Soit M l'ensemble des membres multicast. En supposant que $n \in M$, pour le membre multicast $q(t)$, le routeur n_i ait stocké le nombre de routes p_i (autant de nœuds que possible) jusqu'à le temps t, établi à l'horodatage.

$$\tau s_i, q(1), \tau s_i, q(2), \ \ldots\ , \tau s_i, q\left(p_i, q(t)\right)$$

Dans son cache de route capable de stocker sur la plupart des routes C_i, la fiabilité $rl_i(t)$ du routeur n_i au temps t est donnée par,

$$\mathrm{rl_i}(\mathrm{t}) = \left(\frac{1}{|\mathrm{M}|}\right) \sum_{\mathrm{n_q} \in \mathrm{M}} \left(\frac{\mathrm{P_{i,q}}(\mathrm{t})}{\mathrm{C_i}}\right) \left\{ \prod_{1 \le \phi \le \mathrm{P_{i,q}}(\mathrm{t})+1} (1 - \tau \mathrm{s_{i,q}}(\phi)/\ \ \mathrm{t} \right\}^{\frac{1}{\mathrm{P_{i,q}}(\mathrm{t})+1}}$$

$$n_q \in M \qquad 1 \le \Phi \le p_{1,q}(t) + 1$$

Dans la formulation ci-dessus, on suppose que si n_i n'a pas stocké de route vers un membre de multidiffusion n_q, la valeur sera limitée à 1 uniquement et $\tau s_{i,q}(\Phi)$ sera 0

Par conséquent, si n_i n'a pas déjà de chemin stocké vers un membre de multidiffusion, sa fiabilité sera de 0. Notez que, pour tout routeur n_i , $rl_i(t)$ se situe entre 0 et 1. La fiabilité d'un routeur acquiert une valeur élevée si un grand nombre de routes récentes sont stockées dans le cache de route de n_i à un instant t correspondant à un grand nombre de membres de multidiffusion appartenant au groupe de multidiffusion M.

L'utilité de stocker plusieurs chemins, dans la mesure du possible, pour un membre de multidiffusion, si un chemin se rompt au milieu de la communication de multidiffusion, un autre chemin stocké peut être essayé au lieu d'initier une nouvelle session de découverte de route découvrir un itinéraire vers le membre multicast. Cela permet de réduire le coût des messages sur le réseau.

La figure 7.2 présente un aperçu du processus de modélisation floue. Ce schéma indique les composants qui doivent être définis pour notre protocole de routage, y compris les entrées prises en compte dans le modèle (c'est-à-dire la bande passante, la durée de vie et la fiabilité). En conséquence, la valeur de confiance floue peut être déduite en fonction des degrés suivants : très faible (LV), faible (L), moyen (M), élevé (H) et très élevé (VH).

7.2.2. Évaluation du facteur de certitude basé sur flou (μ)

Dans la figure 7.2, le lien sans fil entre deux nœuds *(p_j, p_k)* est associé à la transition. Le facteur de certitude μ de la transition est évalué à l'aide de la logique floue. Le système flou utilise les paramètres de qualité de service (bande passante-BW, heure d'expiration de liaison-HEL et fiabilité-R) du nœud p_k en tant que variables d'entrée floues pour évaluer le facteur de certitude. Trois ensembles flous, à savoir Très faible, Faible, Moyen et Élevé, sont utilisés pour mesurer chaque paramètre de qualité de service. Les fonctions d'appartenance triangulaire permettent de mesurer des ensembles flous. La transition est déclenchée si $d_j \times w > \tau$ et la valeur de confiance du noeud de sortie est $d_k = d_j \times w \times \mu$.

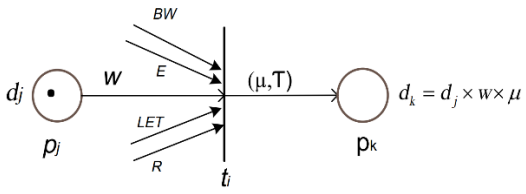


Figure 7.2. Calcule de Facteur de certitude

7.2.3. Évaluation de la valeur seuil (*Th*)

Entre deux nœuds, une transition a une valeur seuil. Cela indique le niveau minimum requis de ressources qualité que chaque nœud devrait avoir. La valeur de seuil est calculée sur la base du taux de perte des paramètres de qualité du nœud. Dans l'équation (7.1), la valeur de seuil est inversement proportionnelle à la fonction du taux de perte de paramètres de qualité de service.

$$Th\infty = \frac{1}{f(\Delta BW, \Delta LT, \Delta R)} \qquad (7.1)$$

Dans l'équation ci-dessus, Δ est un taux de perte de paramètres de qualité de service.

7.2.4. Description du raisonnement flou du protocole de routage SynFAnt

Classiquement, les rouages d'un système flou sont basés sur la structure illustrée à la figure 7.4, qui comprend 5 blocs ou couches :

A. La base de connaissances

Présenté par la couche 1 de la figure 7.4, il contient les définitions des fonctions d'appartenance (formulaires et paramètres) associées aux variables d'entrée / sortie ainsi que l'ensemble des règles floues. Dans le protocole proposé, les hôtes voisins et le débit efficace sont des entrées

pour le modèle de bande passante, et son résultat est exprimé par la valeur de confiance de la bande passante. De même, l'énergie et le nombre de paquets envoyés et reçus sont des entrées pour le modèle de durée de vie et sa sortie est la valeur de confiance de la durée de vie. En outre, les deux valeurs de confiance de bande passante et de durée de vie en sortie sont les entrées du modèle global de structure de raisonnement par inférence floue présenté à la figure 7.3.

B. La fuzzification

La première étape consiste à transformer les variables (entrée et sortie) en variables linguistiques, comme indiqué dans la couche 2 de la figure 7.3. L'univers de la parole (c'est-à-dire la plage de valeurs que peut prendre la variable) est défini. Ensuite, chaque variable est divisée en catégories appelées variables linguistiques. Ces variables sont exprimées par des mots qui leur donnent un sens en utilisant le langage humain (valeurs linguistiques).

La figure 7.5. montre le mécanisme de modification de la variable. Par exemple, les hôtes voisins d'un nœud sont divisés en modalités (un individu avec 4 hôtes voisins est faible à 20%, moyen à 60% et haut à 0%). Ce processus est similaire à la définition des lois a priori dans les statistiques bayésiennes, cet exemple étant une loi antérieure (0.2, 0.6, 0). La différence dans ce cadre est que la somme des vérités ne doit pas nécessairement valoir 1.

De la figure 7.5 :

- Univers vocal : correspond au nombre de nœuds, il peut être représenté par un entier [0-10].

- Variable linguistique : est le nom de la variable de sortie (nombre de nœud).

- Valeurs de langue : "Cat1" (bas), "Cat2" (moyen) et "Cat3" (haut).

C. Inférence floue

- Construire un ensemble de règles

Sur la base des catégories précédemment exécutées, un ensemble de règles est construit. Comme le montre la figure 7.4, chacune des variables *u(t)* et *x(t)* est divisée en 3 catégories (haute, moyenne et basse).

Une véracité pour chacune des règles est ensuite calculée. La construction de ces règles, basée principalement sur les opérateurs logiques "ET" et "OU" est traduite mathématiquement de cette manière.

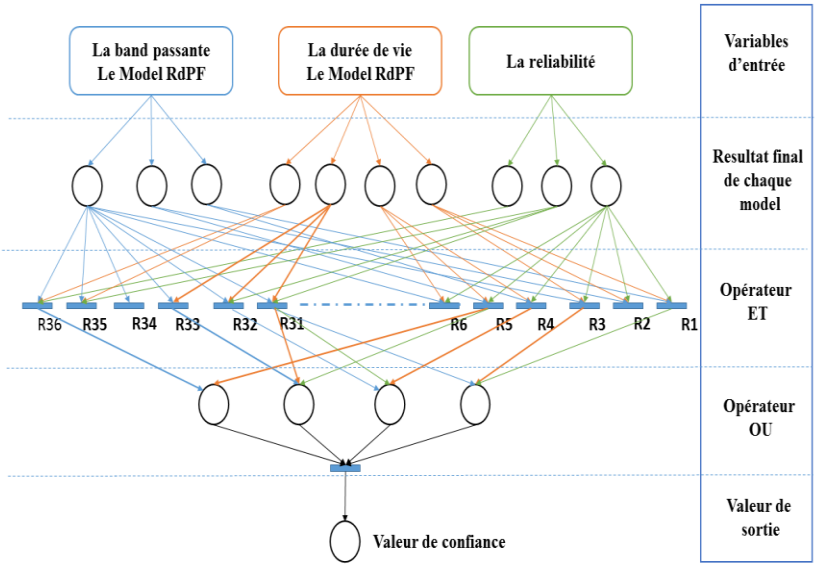


Figure 7.3. Block de la structure de raisonnement d'inférence floue

Le lien de sortie de la couche 2, représenté par la valeur d'appartenance, spécifie le degré d'appartenance de la valeur d'entrée à l'étiquette respective. On peut formuler des règles linguistiques qui relient les étiquettes linguistiques pour *x (t)* et *u (t)* via une partie *Si*, appelée antécédent d'une règle, et la partie *Alors*, également appelée une conséquence de la règle qui détermine le résultat linguistique. Et qui donne une étiquette pour x (t +1). La structure d'une règle unique peut donc être présentée par l'équation 7.2:

$$\text{IF}\left(x(t)\ \text{is}\ Ax(t)\right)\ \text{AND}\left(u(t)\ \text{is}\ Au(t)\right),\quad \text{THEN}\left(x(t+1)\ \text{is}\ Ax(t+1)\right) \qquad (7.2)$$

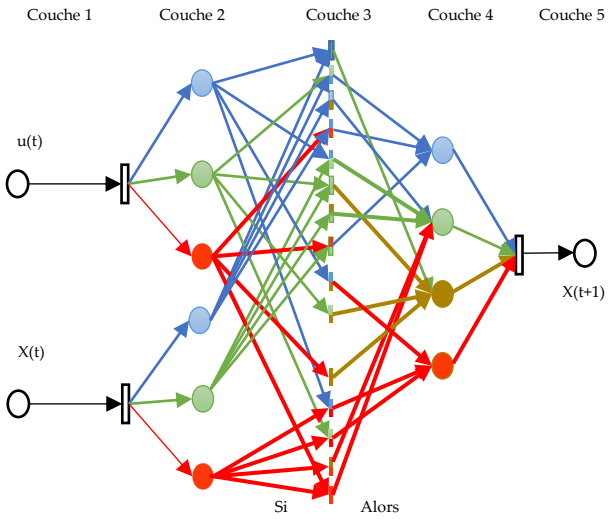


Figure 7.4. Structure de la couche floue des nœuds du réseau de Petri fuzzy de Mamdani.

où *Ax (t), Au (t)* et *Ax (t+1)* sont les étiquettes linguistiques pour *x (t), u (t)* et *x (t + 1),* respectivement, générées pour les points de données.

- Implication: Calcul de la règle d'activation

Il reste à définir une règle d'activation afin d'obtenir une seule réponse. Cette étape s'appelle

l'implication. La méthode Mamdani présentée dans la couche 3 de la figure 7.3 est utilisée dans la méthode proposée.

$$\text{Mamdani}: \mu_{ConclusionRi} y \rightarrow MIN_y \left(\mu_{Ri}(X_0), \mu_{ConclusionRi}(y)\right)$$

Soit x_0 = *(variable1, variable2)* sont les caractéristiques de l'individu

où

- $\mu R_i(x_0)$ est le degré d'activation de la règle;

- $\mu Conclusion(y)$ est la fonction d'appartenance de l'ensemble flou de sortie conformément à la règle de décision.

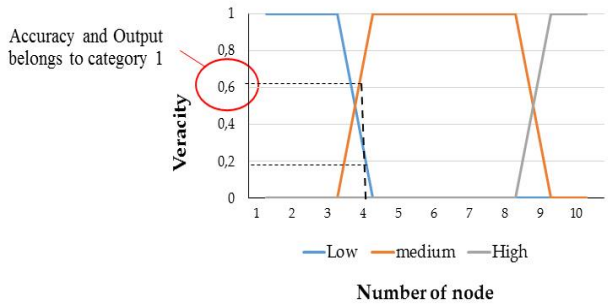


Figure 7.5. Ensembles flous de l'entrée de l'hôte voisin avec un nombre initial de 10

- agrégation

Dans cette étape, toutes les règles sont regroupées. Ce regroupement est donc basé sur l'opérateur logique "OU". Nous utilisons les compositions MAXIMUM pour caractériser l'ensemble des sorties par une fonction d'appartenance égale au maximum des fonctions d'appartenance des sous-ensembles flous. L'étape d'agrégation est présentée par la couche 4 de la figure 7.3.

D. Défuzzification

Cette étape est présentée dans la couche 5 de la figure 7.3 où l'activation finale obtenue lors de l'étape d'agrégation est transformée en une valeur réelle. La méthode du centre de gravité (CdG) est utilisée, car elle est préférable (et plus cohérente avec les principes de la logique floue) en ce sens, qu'elle incorpore le fait qu'un individu peut appartenir à deux catégories à la fois.

$$CoG:\ X_G = \frac{\int_U X\,\mu_x d_x}{\int_U \mu_x d_x} = \frac{\sum_{i=0}^{n} x_i \mu_{xi}}{\sum_{i=0}^{n} \mu_{xi}}$$

où U est l'univers du discours de la variable de sortie.

Cela revient à considérer l'espérance liée à la densité $d = \frac{\mu}{\int_U \mu_x d_x}$ associée à la fonction de véracité. Le modèle de bande passante de la figure 7.3 est pris à titre d'exemple, ce qui constitue une entrée dans le modèle de structure de raisonnement d'inférence floue. Le mécanisme d'estimation de la valeur de confiance de la bande passante est également basé sur le raisonnement flou illustré à la figure 7.6. Il est clair que le raisonnement permettant d'estimer la valeur de confiance d'un tel nœud pour la largeur de bande métrique est un raisonnement flou, c'est-à-dire. Un réseau hiérarchique flou est exécuté au niveau de deux modèles de bande passante et de durée de vie.

La fonction d'appartenance pour les hôtes voisins, le débit, l'énergie, le nombre de paquets envoyés et reçus, la bande passante, la durée de vie et la fiabilité sont présentés dans le tableau 7.1, ainsi que les expressions linguistiques utilisées pour la fuzzification.

D. Analyse de RdPSynFAnt

Pour optimiser de manière optimale l'outil proposé, notre étude repose essentiellement sur une formule probabiliste permettant de calculer la meilleure probabilité de passer d'un nœud à un autre en fonction des paramètres cités précédemment. Notre étude se concentre sur les transitions,

1) fonction de base 'BESTHOP'

Le but de cette fonction est de sélectionner le meilleur nœud voisin N parmi d'autres. Sa fonction essentielle est basée sur une formule probabiliste utilisée pour calculer la meilleure

probabilité de passer d'un nœud à un autre en fonction des paramètres cités dans la section (7.2.1).

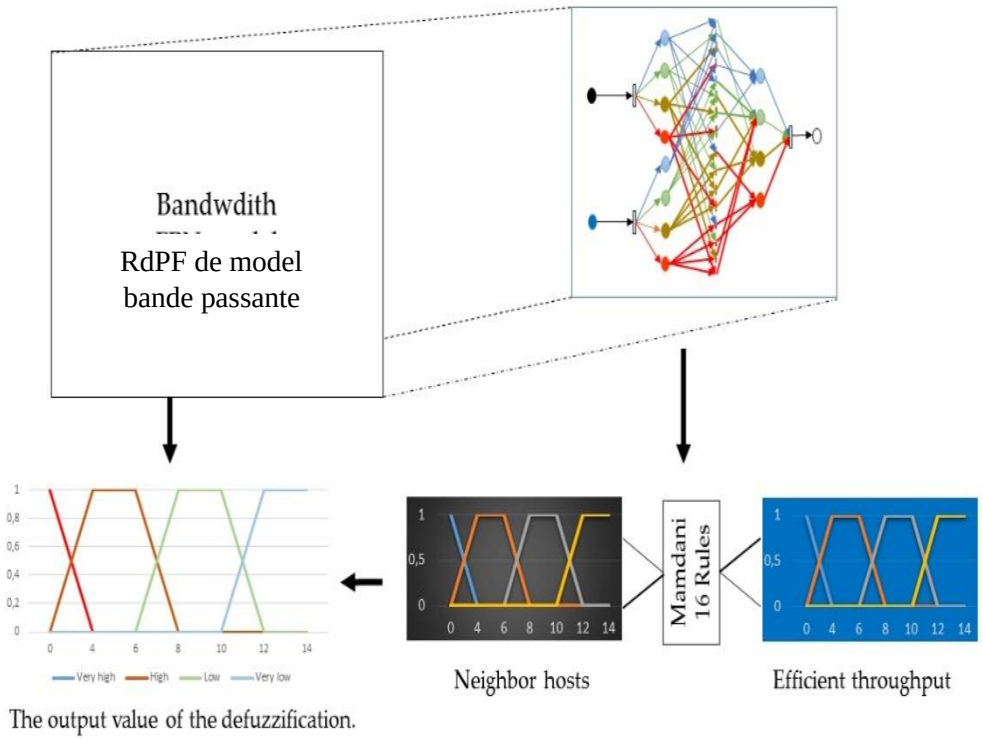


Figure 7.6. Contrôle flou du modèle de bande passante modélisé par le réseau de Petri

L'algorithme de présentation de cette fonction est le suivant.

Algorithme: fonction BestHop

Début

BS: Meilleur Pas

Si *S n'a pas de voisin,* ***alors***

Le noeud S est en dehors de la plage BS ← −1

Retour (BS)

Si non

Sélectionnez le meilleur nœud et ses voisins à l'aide du modèle de règles floues

ÉTAPE 1: Entrez les variables requises (montant, distance, etc.) pour la règle floue dans le modèle pour le connecteur de nœuds.

ÉTAPE 2: Calculez le degré d'adhésion de la proposition des variables à l'aide de la formule trapézoïdale.

ÉTAPE 3: Utilisez l'opérateur composé ET (min) pour calculer la puissance de tir.

ÉTAPE 4: Utilisez l'opérateur de composition OU (MAX) pour définir la force de tir maximale.

La fonction renvoie le noeud qui a la meilleure valeur. / / Retour (BS)

Proposition 1: Une fonction constante F (t) = 1 est associée à chaque transition pour laquelle aucun événement externe n'a été détecté.

Proposition 2: Un marquage instable sera enregistré après le franchissement d'un événement par une transition, qui n'a pas été associée à une temporisation d et qui a une durée égale à 0.

La fonction fait appel à l'algorithme de diagnostic Diag-R pour faire le control de la qualité des liens et nouds

Fin si

Fin

7.2.5. Illustration d'algorithme de routage

L'illustration de l'algorithme de routage du système Ant flou (SynFAnt) peut être présentée à l'aide d'un pseudo-algorithme, dont le but est de montrer le fonctionnement général du protocole.

Algorithme: SynFAnt

S = source;
D = destination
Table de nœuds: contient la stabilité (Stab), les hôtes voisins, la valeur de confiance calculée par la fonction BestHop.

Début

1. *La base de règles R est un ensemble de règles logiques déduites de l'expression logique du raisonnement d'inférence floue et du tableau 1.*
2. *Chaque place P modélise une communication possible et est associé à une proposition logique di: «le signe i est en train de se produire».*
3. *Chaque transition ti modélise une règle logique Ri, qui exprime l'apparition d'une nouvelle communication en combinant les propositions logiques.*
4. *Si une transition modélise l'évolution d'une communication, c'est-à-dire connaissant la durée d de l'activité capable de générer cette communication, alors le coefficient de la vérité de cet événement est flou.*
5. *Une communication peut être décrite comme une conjonction / disjonction du lien déjà produit. Le coefficient de vérité de la variable logique correspondante respecte les règles de réunion ou d'intersection applicables aux ensembles flous.*

Si *(meilleur saut (s) ≠ -1)* ***alors***
// envoie un message entre S et le meilleur voisin
Initialisation: τij ← τ0
(i, j) ∈{1 ... n}, chaque fourmi place au hasard un nœud de direction pour
t= 1 à t = t_{max} do
Pour *chaque fourmi k* ***faire***
Construisez un chemin (t) avec la règle de transition 1. Calculez la longueur Lk (t) de l'itinéraire
Fin pour
Laissons T + trouver le meilleur chemin et la longueur correspondante L +. Mettre à jour la phéromone conformément à la règle 2
Fin si
Retour T + et L +
/ / Destination trouvée, retourne et marque la table de chaque nœud Ant (S, D)
Replay de l'algorithme Sinon
Si *(Meilleur saut (S) = -1)* ***alors***
Si *(NbNear [S] = 0)* ***alors***
Nœud S hors limites Erreur de communication autre
Le nœud D est directement connecté à S Ant (S, D)
Fin si
Si Diag-R<>0
Le protocole détecte un problème ou défaut
Comparer le signe avec les références de diagnostic
Black lister la route
Actualiser la table
Envoie message d'alerte aux nœuds voisins
Appel à l'algorithme de diagnostic
Re-comparer les liens enregistrés dans la table avec la suppression de l'itinéraire actuel.

Envoyer cette erreur aux nœuds voisins pour refaire l'actualisation de tableau.
Replay étape de calcule de destination
Fin si
Fin

La description détaillée de l'algorithme de routage proposé est donnée par:

Phase 1 **: découverte des voisins**

Tous les nœuds du réseau diffusent un message «Hello» contenant leurs adresses périodiquement. Le nœud qui reçoit ce paquet déclare le nœud émetteur en tant que voisin et l'ajoute à la table du voisin. Ensuite, la fonction BestHop est automatiquement déclenchée par ce nœud pour classer le nouveau nœud dans la table de routage en fonction de la valeur de confiance attribuée.

Tableau 7.1. Échelle d'évaluation des hôtes voisins, efficacité du débit, énergie, nombre de paquets envoyés et reçus, et fiabilité des critères de la COIF (c'est-à-dire L (faible), M (moyen), H (élevé) et VH (très faible) HAUTE))

Parameters	Low	Meduim	High	Verry High
Neighbor hosts	[0, 3]	[2, 6]	[5, 10]	[9, 14]
Efficient throughput	[0, 4]	[3, 8]	[7, 12]	[11, 16]
Energy (joules)	[0, 0.4]	[0.4, 0.7]	[0.5,0.9]	[0.8, 1.5]
Nombre de paquets envoyés et reçus	[0, 3]	[2, 5]	[4, 8]	[7, 10]
Reliability coef	[0, 0.45]	[0.40, 0.7]	[0.65, 1]	/
Durée de vie	[0, 0.55]	[4, 0.70]	[0.63, 1]	
Bandwidth	[0, 0.33]	[0.2, 0.75]	[0.60, 1]	/

Phase 2 **: découverte de l'itinéraire**

Lorsqu'un nœud source *S* souhaite transmettre des données à un nœud de destination *D*, il envoie périodiquement un certain nombre de fourmis de transmission pour trouver le meilleur itinéraire. L'envoi périodique de ces paquets nous permet de trouver de meilleurs itinéraires qui seront utilisés lors de l'envoi de paquets de données, tout en assurant la ***maintenance*** de l'itinéraire. À mesure qu'il avance vers la destination, F-**Ant** enregistre l'adresse des nœuds visités, la bande passante minimale, le délai de la route et sa fiabilité. Chaque F-ant choisit un nouveau nœud à visiter parmi ses voisins en fonction d'une probabilité appelée probabilité de transition définie par l'équation 1 chapitre 5.

Pour chaque nœud *N* choisi, la fourmi suivante vérifiera si ce nœud a déjà été visité. Si c'est le cas, il existe un cycle qui devrait être évité en supprimant tous les nœuds successeurs de *k* dans la liste des noeuds visités (champ de paquet F-Ant). Ensuite, F-Ant choisit un nouveau nœud intermédiaire en fonction de la probabilité de transition.

Si le nœud n'est pas encore visité, l'antenne de transmission vérifie si ce nœud est la destination souhaitée. Si ce n'est pas le cas, il s'agit d'un noeud intermédiaire qui n'a pas encore été visité. Le protocole effectuera les tâches suivantes:

- Ajouter ce nœud à la liste des nœuds visités.
- Vérifiez si le débit de ce lien est inférieur au de taux de champ de F-Ant.
- Basculez cette valeur sur ce champ si c'est le cas.
- Ajouter au champ délai la valeur de délai de ce lien (Délai = Délai + délai du lien).
- Vérifiez si la stabilité de ce lien est inférieure à la stabilité enregistrée dans le tableau des fourmis (Stab).
- Attribuez cette valeur à ce champ si c'est le cas (Stab = stabilité de la liaison).

Si le routage atteint la destination souhaitée, il crée le fichier Backward et transfère toutes les informations qu'il a transmises (vitesse, délai, stabilité, liste des nœuds visités et somme des phéromones de tous les liens). Enfin, cette F-ant sera tuée. La procédure pour trouver la route optimale est décrite à la figure 7.6.

Phase3: **Calcul de la quantité de phéromone à ajouter aux liens**

Le nœud de destination "*D*" détecte un nouveau voisin "*N*" à chaque fois, qui sera ajouté à la table des voisins du nœud. Ensuite, la valeur de la phéromone de liaison (*S, Ni*) sera affectée à une constant.

Ensuite, cette valeur est mise à jour, quand une fourmi en arrière passe. A ce moment, cette

valeur est ajoutée à une quantité de phéromone calculée en fonction de la qualité de la route parcourue par la fourmi avant. Cette quantité notée *(S, D)* est exprimée par l'équation (7.3):

$$\Delta\tau(u_i,u_j)=\frac{B(R)_B^{\beta}+T(R)_T^{\beta}}{D(R)_D^{\beta}} \quad (7.3)$$

où:

- *B (R)* est la bande passante de la route *R*. C'est la bande passante minimale de tous les liens formant *R* car il s'agit d'une métrique concave.

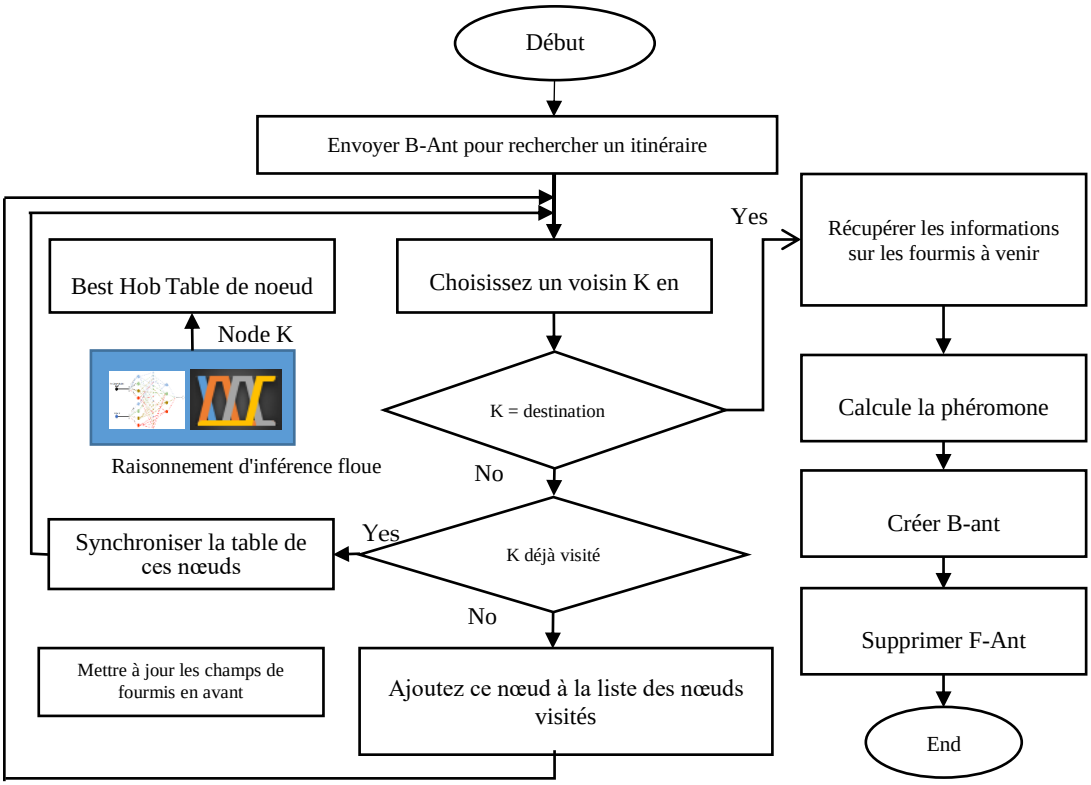


Figure 7.7. Organigramme de recherche d'itinéraire

- *D (R)* est le retard de la route *R*. C'est la somme des retards de tous les liens formant *R* parce que c'est une métrique additive.

- *T (R)* est la stabilité de la route *R*. C'est la stabilité minimale de tous les liens formant *R* car il s'agit d'une métrique concave.

- β_B, β_T et β_D sont des poids, représentant l'importance relative de chacune des métriques lors de la mise à jour de la phéromone sur les liaisons de route *R*.

Pour respecter le principe d'évaporation de la phéromone, la quantité de phéromone sur toutes les liaisons du réseau est périodiquement multipliée par le facteur d'évaporation *p*, sachant que $0 < p < 1$.

La quantité de phéromone spécifiant la qualité de l'itinéraire trouvé est également calculée à l'arrivée de la destination du nœud Ant avant. Cette quantité est stockée dans le champ Phéromone de la table de routage

***Phase 4:* Réponse à la fourmi avant F-Ant**

La B-Ant prend le chemin opposé à celui emprunté par la fourmi avant. Il utilise la liste des

nœuds visités par la F-Ant dans la direction opposée. Lorsque la fourrière approche du nœud source, elle met à jour les tables des nœuds traversés.

7.2.6. Comportement de l'algorithme SyncFAnt aux différents nœuds

Le protocole proposé est implémenté en tant qu'agent de routage s'exécutant sur tous les nœuds du réseau. Plusieurs fonctions sont implémentées comme décrit à la figure 7.7. L'appel de ses fonctions dépend de la nature du nœud (nœud source, nœud de destination ou nœud intermédiaire).

- **Au nœud source**

Selon la figure 7.8, le nœud source *S* pourrait être un demandeur de route pour envoyer des données à un nœud de destination *D*. De plus, le nœud *S* peut également recevoir un message de rappel des autres nœuds préalablement visités par un certificat de transfert. Le comportement du nœud source dans chaque cas est décrit dans les algorithmes suivants.

Fonction d'algorithme SyncFAnt au nœud source

```
   S = source;
   D = destination
Liste de visites: Table
Début
Si S veut envoyer des données à D qui ne se trouve pas dans la table de routage,
alors
   Créez des fourmis et envoyez-les lors de la recherche d'itinéraire.
Fin si
Si le noeud S reçoit une fourmi en arrière envoyée par D, alors
   Mettre à jour les tables (routage et voisins)
   Tuer la fourmi en arrière
   Envoyer des données
Fin si
Fin
```

- **Au nœud intermédiaire**

Le nœud intermédiaire se comporte différemment selon la nature du paquet reçu:

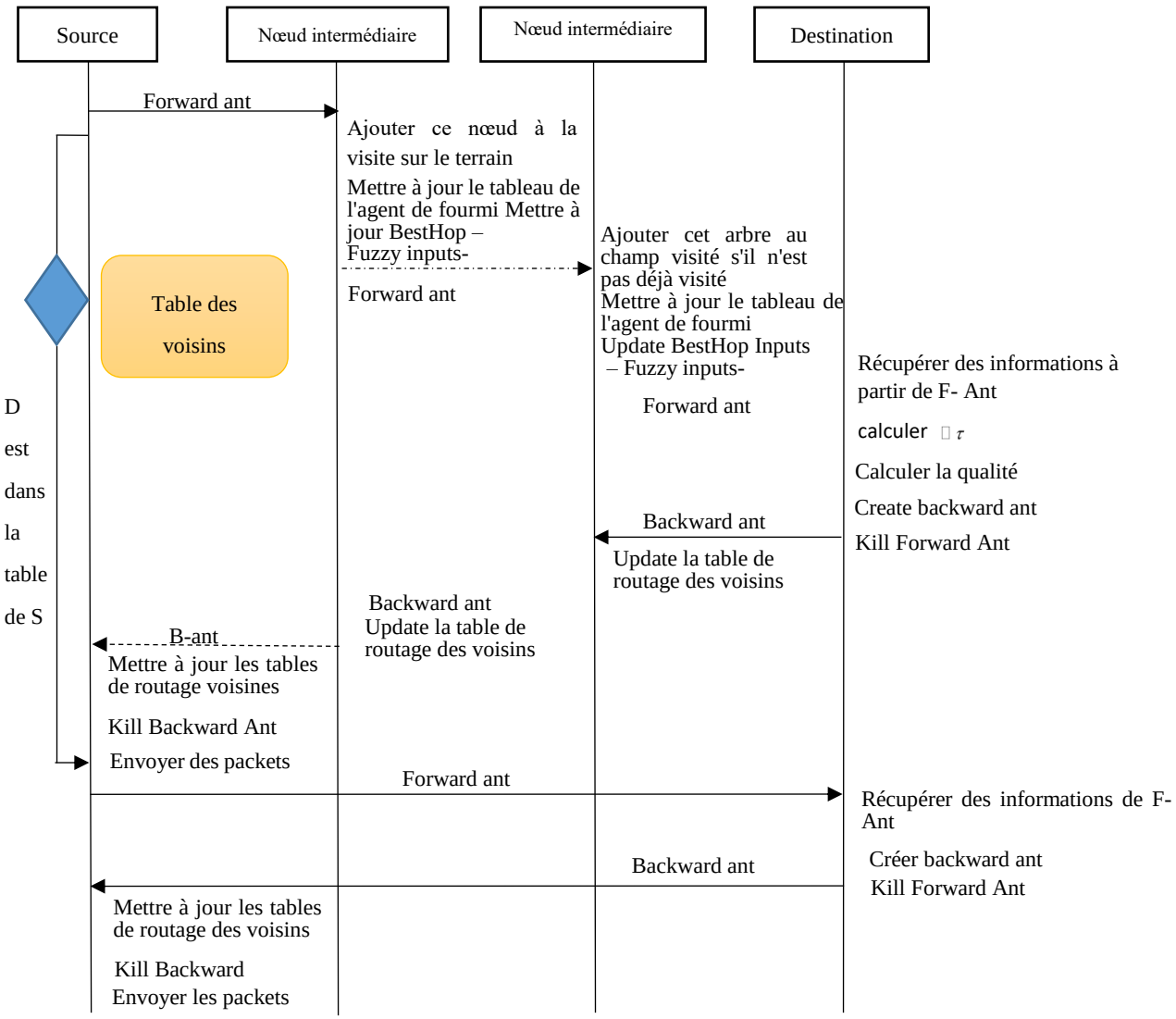


Figure 7.8.Les différentes fonctions de l'algorithme SynFAnt aux différents nœuds

Algorithme SynFAnt Fonction nœuds intermédiaires

Début
Si *le paquet reçu est un ant avant,* ***alors***
Si *ce nœud k a visité (nœud pas encore visité)* ***alors***
Ajouter ce nœud à la liste des nœuds visités
Mettre à jour les champs F-Ant:
Débit min (Débit, débit de liaison)
Retard Retard + retard du lien
Stab min (Stab, stabilité du lien)
Envoyer F-Ant au nœud choisi en fonction de la transition de probabilité (équation 1 chap 5)
Supprimez le cycle et envoyez le Forward à un autre nœud en fonction de la probabilité de transition donnée par BestHop Function.
Fin si
Si non
Si *le paquet reçu est une fourmi arrière,* ***alors***
Mettre à jour les tables (table des voisins et table de routage)
Envoyez la fourrière au nœud suivant dans la liste des visites.
Fin si
Fin si
Fin

- **Au nœud de destination**

Lorsqu'un nœud reçoit un paquet F-Ant, il vérifie s'il s'agit du nœud de destination. Si tel est le cas, la procédure est la suivante:

Algorithme: Fonction SynFAnt au noeud de destination

Début
  Si *le paquet reçu est un ant avant, alors*
    Calculez la quantité de phéromone à ajouter aux arches (équation 2).
    Créer une fourmi en arrière
    Envoi de la version précédente au noeud de destination
    Tuer F-Ant
  Fin si
Fin

7.3. Routage SyncFAnt avec un exemple de topologie

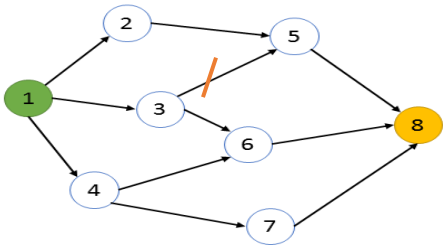


Figure 7.8. Topologie du réseau MANET

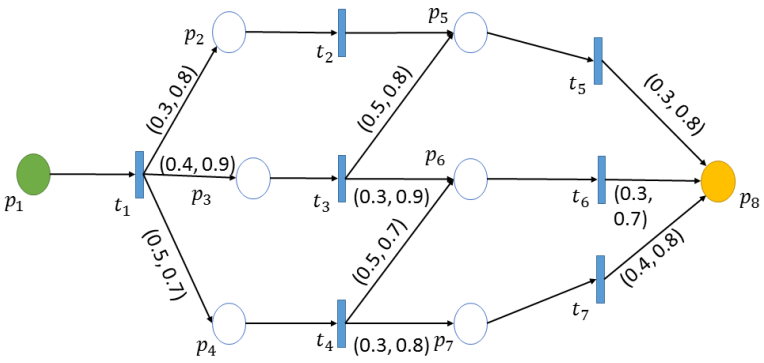


Figure 7.9: Réseau MANET modélisé RdPSyncF

Dans la figure 7.8, la topologie du réseau Mobile Adhoc est représentée avec une taille de 8 nœuds. Ici, les nœuds 1 et 8 sont respectivement des nœuds source et de destination. Dans la figure 7.9, la topologie MANET est changée en RdPSyncF, où les nœuds mobiles sont comme des places et les liaisons sans fil, comme des transitions. Chaque transition comporte une paire de valeurs, à savoir (seuilτ, facteur de certitude μ).

Le nœud source-1 commence le processus de routage en envoyant des paquets RREQ à ses nœuds voisins 2, 3, 4 du voisin d'un saut. Chaque nœud, lors de la réception du paquet de demande, calcule la valeur seuil en utilisant l'équation (7.1) et le facteur de certitude en appliquant une logique floue. Pour la transition entre l'expéditeur du paquet et lui-même. Ces valeurs calculées pour les nœuds 2, 3 et 4 sont respectivement égales à (0,3,0,8), (0,4,0,9) et (0,5,0,7). Chaque nœud intermédiaire inclut ces valeurs dans un paquet RREQ et les envoie aux nœuds voisins du saut suivant. Le nœud 8 de destination collecte l'état du réseau via plusieurs paquets RREQ. En exécutant l'algorithme CRA, le nœud de destination trouve le chemin digne de confiance par lequel il obtient la valeur maximale de DoT de la source. Ici, le chemin est 1-3-5-8 avec le maximum de DoT possible du nœud de destination est 0.576.

Ici, le MANET est considéré comme un réseau RdPSyncF, où les nœuds sont similaires aux places et les liaisons sans fil sont en transition.

1) Un nœud source (S) lance le processus de recherche d'itinéraire vers le nœud de destination ; il prépare le paquet de demande d'itinéraire (RREQ) en ajoutant sa valeur degrés de vérité$\propto (s)$ à 1, c'est-à-dire$\propto (s) = 1$. Le nœud source envoie les paquets RREQ à des nœuds voisins du saut.
2) Après la réception des paquets, un nœud intermédiaire calcule les valeurs de seuil (τ) de Facteur de certitude (μ) pour le nœud émetteur. Un nœud intermédiaire ajoute son ID de nœud, μ, τ et poids (w) au paquet de demande avant de le transmettre au nœud du saut suivant.
3) Le nœud de destination reçoit les multiples paquets RREQ et à travers eux, il peut rassembler les informations réseau telles que les nœuds et les liaisons sans fil (μ, τ et w).
4) Avec les informations disponibles, le nœud de destination peut exécuter l'algorithme SyncFAnt pour calculer les valeurs de DoT de chaque nœud et rechercher le chemin digne de confiance.
5) Le nœud de destination prépare le paquet de réponse de route (RREP) et l'envoie au nœud source via le chemin digne de confiance identifié. Le nœud source définit le chemin et commence le transfert des données vers la destination.

7.3. Simulations et comparaisons

7.3.1. Algorithme SyncFAnt de monodiffusion

On discute ici de l'SyncFAnt pour le MANET modélisé RdPSF, comme illustré à la figure 7.9.

Entrée: *I, O, U, W, Th* sont les matrices (définies dans la section 7.1) avec l'ordre $m \times n$, où les lignes représentent les nœuds (dans la séquence 1, 2, 3, 4, 5, 6, 7, 8) et les colonnes représentent les transitions (t_1, t2, t3, t4, t5, t6, t7), avec un défaut au niveau de la route t_2.M_0 est un vecteur

de marquage initial avec une valeur DoT source égale à 1 et 0 pour les autres nœuds. Dans la figure 7.9, chaque transition a un seul place d'entrée, les poids sont donc égaux à 1.

//Sortie:

M_k est un vecteur de sortie avec les valeurs DoT finales de tous les nœuds.

Pour ce système, le marquage initial est M_O =[1,0,0,0,0,0,0,0] et les autres matrices: I, O, U, W, Th sont comme

$$I=\begin{pmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&1&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\\0&0&0&0&0&0&0\end{pmatrix}\quad O=\begin{pmatrix}0&0&0&0&0&0&0\\1&0&0&0&0&0&0\\1&0&0&0&0&0&0\\1&0&0&0&0&0&0\\0&1&1&0&0&0&0\\0&0&1&1&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&1&1&1\end{pmatrix}\quad W=\begin{pmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&1&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\\0&0&0&0&0&0&0\end{pmatrix}$$

$$Th=\begin{pmatrix}\infty&\infty&\infty&\infty&\infty&\infty&\infty\\0.3&\infty&\infty&\infty&\infty&\infty&\infty\\0.4&\infty&\infty&\infty&\infty&\infty&\infty\\0.5&\infty&\infty&\infty&\infty&\infty&\infty\\\infty&0.4&0.5&\infty&\infty&\infty&\infty\\\infty&\infty&0.3&0.5&\infty&\infty&\infty\\\infty&\infty&\infty&0.3&\infty&\infty&\infty\\\infty&\infty&\infty&\infty&0.3&0.3&0.4\end{pmatrix}\quad U=\begin{pmatrix}0&0&0&0&0&0&0\\0.8&0&0&0&0&0&0\\0.9&0&0&0&0&0&0\\0.7&0&0&0&0&0&0\\0&0.7&0.8&0&0&0&0\\0&0&0.9&0.7&0&0&0\\0&0&0&0.8&0&0&0\\0&0&0&0&0.8&0.7&0.8\end{pmatrix}$$

Étape 1 : Pour la première itération, Soit k = 1.

Étape 2 : Pour chaque transition, calculez les valeurs de degré de confiance pondérées (WDoT) à partir des emplacements en entrée.$\Gamma^{(k)} = k_{(k-1)}W$

$$\Gamma^{(1)} = M_0 W = \begin{bmatrix}1&0&0&0&0&0&0&0\end{bmatrix}\begin{pmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&1&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\\0&0&0&0&0&0&0\end{pmatrix}$$

Et $\Gamma^{(1)} = [1\ 0\ 0\ 0\ 0\ 0\ 0]^T$

Étape 3: Recherchez les arcs de sortie éligibles de chaque transition. Ces valeurs DoT pondérées sont supérieures aux valeurs de seuil.

$$Y^{(k)} = (y_{ij}^{(k)})_{m*n} = (\Gamma^T)^k - Th$$

est $Y^{(k)}$ la matrice de différence résultante, calculée en soustrayant la matrice de seuil de la matrice de valeurs DoT des transitions. Partout où les entrées Y^kde la matrice sont des nombres positifs, la matrice de comparaison ($E^{(k)} = (e_{ij})_{m*n}^{(k)}$) les entrées sont 1 sinon c'est 0.

$$E^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Étape 5 : calculez le nouveau marquageM_k. $M_k = M_{k-1} \oplus (\psi^{(k)} \otimes \Gamma^{(k)})$, où $\psi^{(k)} \otimes \Gamma^{(k)}$ représente la valeur maximale du produit de DoT et μ les valeurs du nœud

$$\psi^{(1)} \otimes \Gamma^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.8 \\ 0.9 \\ 0.7 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

C'est à dir

$\psi^{(1)} \otimes \Gamma^{(1)}$=[0 0.8 0.9 0.7 0 0 0 0]

M_1=M_0+($\psi^{(1)} \otimes \Gamma^{(1)}$) = [1 0.8 0.9 0.7 0 0 0 0]

Si deux états de marquage successifs sont égaux, c'est-à-dire M_k = M_{k-1}, passez à l'étape 6; sinon continuez l'itération suivante (passez à l'étape 2) pour K = K + 1

Étape 6: Terminez le raisonnement

Ici M_2 = (1 0.8 0.9 0.7 0.72 0.81 0.56 0) ,

et M_3 = (1 0.8 0.9 0.7 0.72 0.81 0.56 0.576)

M_4 = (1 0.8 0.9 0.7 0.72 0.81 0.56 0.576)

Puisque les deux dernières itérations sont égales, M3=M_4 terminez le processus.

Ici, le degré de confiance du nœud de destination (nœud -8) est de 0,576, c'est-à-dire que le meilleur chemin (1-3-5-8) de la source à la destination a la valeur de confiance 0,576. Dans la figure 5.9, il existe cinq chemins possibles du nœud source au nœud de destination : 1-2-5-8, 1-3-5-8, 1-3-6-8, 1-4-6-8 et 1-4. -7-8. Mais le chemin 1-4-6-8 est invalidé, car la transition ne peut pas être déclenchée. Pour le premier chemin, DoT est évalué en tant que

$$\underline{1 > 0.3} \Rightarrow 1 \times 0.8 = 0.8$$
$$\underline{0.8 > 0.4} \Rightarrow 0.8 \times 0.7 = 0.56$$
$$\underline{0.56 > 0.3} \Rightarrow 0.56 \times 0.8 = 0.448$$

De même, les valeurs DoT pour les trois chemins restants sont respectivement de 0,576, 0,567 et 0,448.

7.3.1.1. Traçage du chemin

Pour tracer le chemin vers le nœud source,

1) Vérifier la matrice($\psi^{(k)} \otimes \Gamma^{(k)}$) (où les lignes représentent les nœuds et les colonnes représentent transitions).

2) Identifiez le nœud de destination t_i à travers lequel il a obtenu le haut DoT dans la matrice($\psi^{(k)} \otimes \Gamma^{(k)}$).

3) Pour la transition t_i, identifiez le nœud p_i d'entrée en vérifiant dans la matrice d'entrée I.

4) Pour le nœud p_i , trouvez la transition à partir de laquelle il a obtenu la valeur DoT élevée dans la matrice ($\psi^{(k)} \otimes \Gamma^{(k)}$).

5) Répétez cette procédure jusqu'à ce que le nœud source soit identifié.

5.3.1.1. Processus de récupération d'itinéraire

Le RdPSyncF a une meilleure capacité de récupération d'itinéraire en identifiant un autre chemin avec une valeur de DoT élevée possible. Dans la figure 7.10, si la liaison sans fil des nœuds 3 à 5 est interrompue en raison de la mobilité des nœuds, ce qui entraîne la déconnexion du chemin existant, c'est-à-dire 1-3-5-8. Le nœud de destination commence à trouver un autre chemin, avec un meilleur degré de confiance. Ici, le nœud de destination trouve sa deuxième plus haute valeur DoT dans la matrice($\psi^{(k)} \otimes \Gamma^{(k)}$) et continue le traçage comme expliqué dans la section ci-dessus. Le paquet RREP reçoit une réponse du nœud de destination au nœud source via la route récemment découverte. Dans la figure 7.10, le nouvel itinéraire découvert est 1-3-6-8 avec un degré de confiance de 0,567.

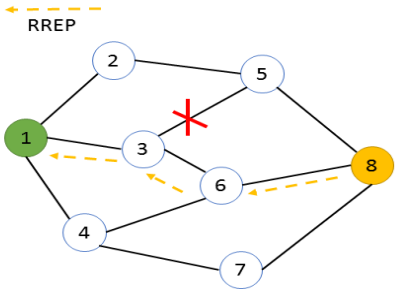


Figure 7.10. Récupération de route à l'aide d'un raisonnement simultané.

7.3.2. Extension RdPSyncF du routage multidiffusion

Dans le routage de multidiffusion, un arbre de multidiffusion est formé, où l'expéditeur est à la racine et tous les nœuds de destination sont des nœuds feuilles. Tous les chemins de la source à la destination doivent satisfaire aux exigences de qualité. Dans RdPSyncF, le routage multidiffusion est l'extension du protocole de routage MAODV Adhoc à la multidiffusion. Ici, chaque fois qu'un nouveau nœud veut partager des informations avec les membres du groupe, il envoie les paquets de jointure RREQ aux nœuds voisins. À la réception du paquet RREQ, un nœud intermédiaire donne la réponse (RREP), s'il fait partie de cet arbre de multidiffusion. Sinon, il transmet la demande de jointure à d'autres nœuds en ajoutant son ID de nœud. Les membres du groupe (y compris la source et les destinataires) lors de la réception de plusieurs paquets de demande, envoient un message de réponse (RREP) au nouveau nœud. Le nouveau nœud reçoit les paquets RREP des membres du groupe, à travers lesquels il collecte les métriques telles que et les valeurs. Il exécute l'algorithme SyncFAnt pour évaluer les valeurs de DoT des membres du groupe, sélectionne le membre avec une valeur de DoT élevée et joint l'arborescence de multidiffusion via ce membre de groupe.

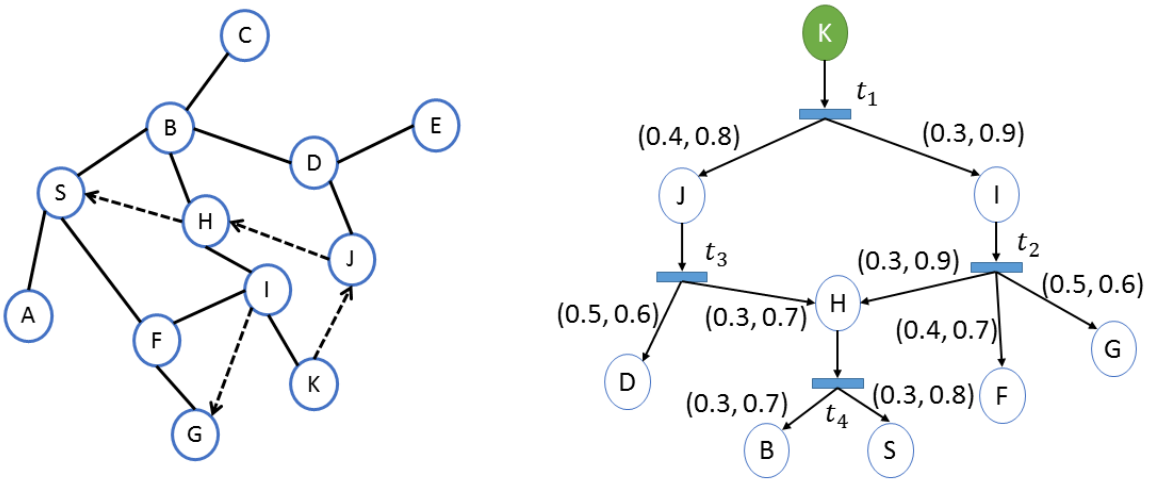
Dans le MANET, comme le montre la figure 7.11 (a), S est le nœud source, (A, C, E, G) sont les nœuds de destination et (B, D, F) sont des nœuds de transfert. Un nouveau nœud K, qui souhaite rejoindre l'arbre de multidiffusion, envoie la demande de participation. En réponse aux paquets de requête reçus, les nœuds S, B, D, F et G répondent au nœud K. Par le biais des paquets de requête et de réponse, le nouveau nœud rassemble les informations de réseau et le modèle.

le RdPF équivalent à celui de la figure 7.11 (b). Le nœud K implémente l'algorithme SyncFAnt pour trouver le membre de groupe potentiel avec une valeur de confiance élevée

7.3.2.1. Algorithme de raisonnement simultané du routage multidiffusion

L'ARS est étendu pour l'algorithme de multidiffusion, comme décrit à la figure 7.11b.

Ici, le vecteur de marquage initial est M_0= [1, 0, 0, 0, 0, 0, 0, 0,0]$^{T.}$ Dans chacune des matrices



(a) (b)

Figure 7.11. Routage multidiffusion modélisé RdpSyncF

(I, O, U, W, Th), les lignes (nœuds / places) sont dans la séquence de [K, I, J, H, G, F, H, D, B, S]

et les colonnes (transitions) sont dans la séquence de (t_1, t_2, t_3, t_4).

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad O = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.9 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.7 & 0 \\ 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 0.7 \end{bmatrix}$$

$$Th = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0.3 & 0.3 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0.3 \end{bmatrix} \quad \Gamma^{(1)} = [1 \quad 0 \quad 0 \quad 0]^T \quad \psi^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.9 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

comme expliqué dans la section 5.3.5, les itérations suivantes marquant les vecteurs sont comme

M_1= [1,0.9, 0.8, 0, 0, 0, 0, 0,0] ^{T}

M_2= [1,0.9, 0.8, 0.54, 0.63, 0.81, 0.48, 0,0] ^{T}

M_3 = [1,0.9, 0.8, 0.54, 0.63, 0.81, 0.48, 0.648,0.561] ^{T}

M_4 = [1,0.9, 0.8, 0.54, 0.63, 0.81, 0.48, 0.648,0.561] ^{T}

Ici, les deux dernières itérations M_3=M_4 sont égales et le nœud B du membre du groupe a la valeur de confiance la plus élevée. Ainsi, par le biais du membre du groupe B, le noeud K rejoint le groupe de multidiffusion.

7.4.1. Simulation numérique

Nous avons évalué les performances de SynFAnt [125] à l'aide du simulateur NS-2. Nous avons comparé notre protocole avec quatre autres modèles récents, qui sont des modèles de confiance fondés sur la réputation, comme moyen de communication: EFMMRP [94], EELB-Mega [95, LOADng [96] et ETX-Ant [97]. Le scénario permettant de tester notre protocole dans le simulateur NS2 est détaillé dans le tableau 7.2. Le nombre de nœuds dans le réseau considéré varie de 10 à 200 et est positionné de manière aléatoire sur un carré de 1 000 m × 1 000 m. La portée de communication est de 250 m, tandis que la portée de détection de porteuse est de 250 m. Un total de 10 flux CBR, dont 5 Best Effort et 5 QoS, est établi entre les nœuds source et de destination choisis au hasard. Chaque simulation dure 100 secondes et les résultats présentés représentent la moyenne de 30 simulations pour un numéro de nœud défini.

Afin de comparer les performances des différentes métriques de routage. Ces critères sont des mesures de performance qui dépendent des critères recherchés par le protocole de routage. Dans cette étude, nous cherchons à maximiser le rapport de livraison de paquets (PDR) ou à minimiser de manière équivalente le taux de perte de paquets, à maximiser la capacité du réseau et à réduire les retards de bout en bout.

Tableau 7.2. Scénario pour la topologie ns-2

Parametres	Valeurs
Nombre de nœuds simulés	20-80-120-160-200
Taille de la zone de la topographie x(m)	1000 m
Taille de la zone de la topographie y(m)	1000 m
Wireless range	250 m
Taille de paquet	512 bytes
Temps de pause (s) à la simulation	0s
Node placement	Random
Mobility model	Random way point
Speed node	[5-10-15-20-25- 30] m/s
Protocoles de routage simulés	EFMMRP,EELB-Mega, LoaDng and EXT

Nous utilisons donc le PDR, la capacité, le délai de bout en bout et le taux d'acceptation des flux de qualité de service du paquet de livraison par paquet comme métriques de performance.

A. Ratio de livraison des paquets

- Le rapport de livraison de paquet (Packet Delivery Ratio PDR) est le rapport entre le nombre de paquets de données reçus par la destination et le nombre de paquets de données envoyés par la source. Le PDR est calculé comme suit:

$$PDR = \frac{\sum paquets\ reçus}{\sum paquets\ envoyés}$$

Les performances de quatre protocoles SynFAnt, EFMMRP, LOAdng et EELB-mega sont illustrées à la figure 7.12. Il est observé que le rapport de distribution de paquets PDR augmente relativement avec le nombre de nœuds du réseau. On peut clairement voir qu'il y a plus de chance d'avoir un routage plus stable avec un réseau qui contient plus de nœuds par rapport à un réseau avec moins de nœuds.

Le protocole SynFAnt dépasse les autres protocoles ; Même si le protocole EFMMRP utilise une logique floue pour le contrôle des incertitudes, le bloc de contrôle néglige la durée de vie des liens qui jouent un rôle très important dans la livraison des paquets. Le protocole LOADng ne prend pas en compte les incertitudes liées à la sélection par trajets multiples pour la transmission de paquets; ce qui explique la dégradation du PDR en fonction du nombre de nœuds de ce protocole.

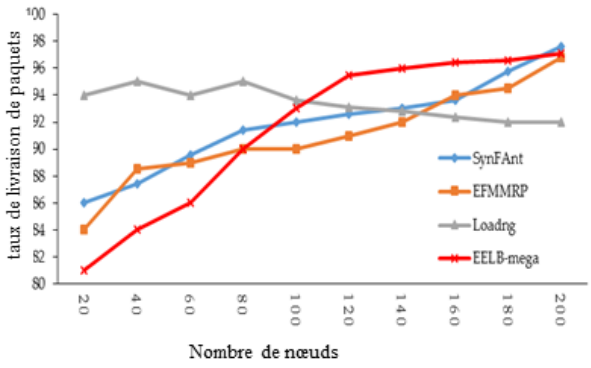


Figure 7.12. Ratio de distribution du paquet par rapport au nombre de nœuds.

La figure 7.13 illustre la PDR pour différents protocoles en faisant varier la vitesse des nœuds. On observe que la valeur de PDR diminue en fonction de la vitesse, principalement en raison du décalage entre les valeurs prises en compte par le protocole de routage et les valeurs réelles des métriques. Les performances du protocole SynFAnt proposé sont meilleures que celles du protocole EFMMRP et Les protocoles ETX, car SynFAnt préfèrent les routes avec des pertes

plus faibles et une capacité élevée. ETX est la pire métrique car il choisit les chemins les plus longs.

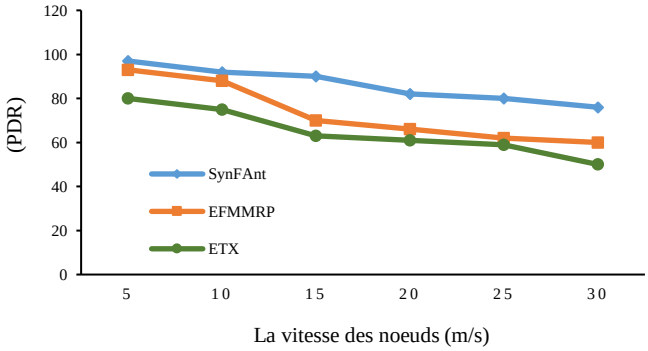


Figure 7.13. Ratio de livraison du paquet par rapport à la mobilité.

B. Capacité

La capacité est la quantité de trafic passé envoyé par tous les nœuds *(N)* du réseau pendant la simulation. Cette métrique représente la quantité maximale de trafic pouvant transiter sur le réseau. La plus grande capacité du réseau offre une meilleure qualité de service à un plus grand nombre d'utilisateurs. Cette métrique est calculée comme suit:

$$Capacité = = \frac{\sum_{n \in N} |paquets\ recevés|}{temps\ de\ routage}$$

Le débit exprimé en *Kb/s* en fonction du débit de la source exprimé en *Kb/s* pour une vitesse fixe de 25 km/h est représenté à la figure 7.14. Cette figure illustre la capacité des métriques SynFAnt, EFFMRP et ETX pour la topologie aléatoire. On constate que les performances de SynFant et de EFFMRP sont meilleures que celles d'ETX, car ils choisissent les itinéraires avec des capacités élevées et moins de pertes. De plus, le débit de SynFAnt dépasse 515 *Kb/s*. De plus, la capacité des métriques est mesurée en fonction du nombre de nœuds. Les résultats obtenus du débit pour les protocoles SynFAnt, EFMMRP, EELB-Mega et ETX sont présentés à la figure 7.15. Dans le protocole proposé, le débit moyen est amélioré à 98,4% par la variation du nombre de nœuds de 50 à 200, à un taux de 33 ± 0,38%, le protocole EELB-Mega atteint un taux de 28% et l'EFMMRP un taux de 31,5%. Pour le protocole ETX, il n'atteint pas un débit remarquable car il ne prend pas la mesure d'énergie dans les critères de l'algorithme de routage.

C. Délai moyen

Le délai de bout en bout est le temps moyen entre l'envoi d'un paquet et sa réception. Cela inclut le délai d'acheminement et d'autres retards divers, tels que le délai de transmission, le délai de propagation et le délai d'attente. Le délai de bout en bout est calculé comme suit:

$$\text{Retard}=\frac{\sum_{n\in N} Retard}{\sum Packets\ recevés}$$

Les valeurs mesurées du retard sont représentées en fonction du nombre de nœuds dans le réseau. La figure 7.16 montre clairement que le délai varie proportionnellement au nombre de nœuds. En effet, les scénarios étant générés avec la même densité de nœuds, la surface de simulation augmente avec le nombre de nœuds. Par la suite, à mesure que le trafic de routage augmente, les paquets de données sont de plus en plus retardés dans les files d'attente des nœuds intermédiaires, ce qui se traduit directement par une augmentation considérable du temps de réponse. Ces facteurs (le nombre de nœuds et le nombre de paquets envoyés) sont la cause principale du retard important donné par le protocole LOAdng. Étant donné qu'EELB-Mega génère le temps système le plus important, il génère le délai le plus long par rapport aux autres protocoles. D'autre part, la sollicitation de certains nœuds par rapport à d'autres lors du routage de paquets de données peut entraîner des goulots d'étranglement dans les chemins de données. Ainsi, le délai de livraison des colis augmente considérablement. Cet effet explique le temps de réponse significatif généré par le protocole LOAdng. Les protocoles SynFAnt et EFMMRP génèrent des retards proches et beaucoup plus courts que ceux enregistrés par les protocoles LOAdng et EELB-Mega. Grâce à sa surcharge et à la variété de chemins utilisés, SynFAnt produit le délai de bout en bout le plus faible.

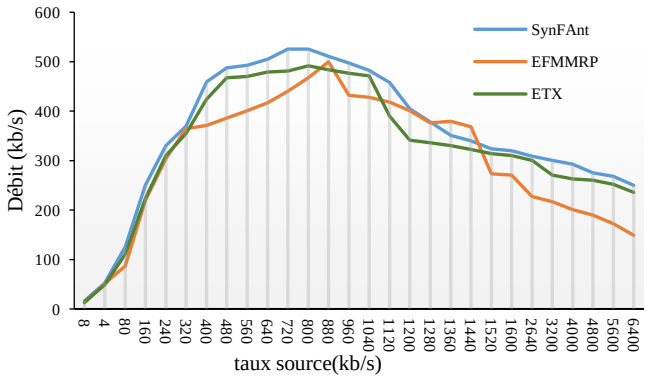


Figure.7.14. Taux de transfert vs débit source.

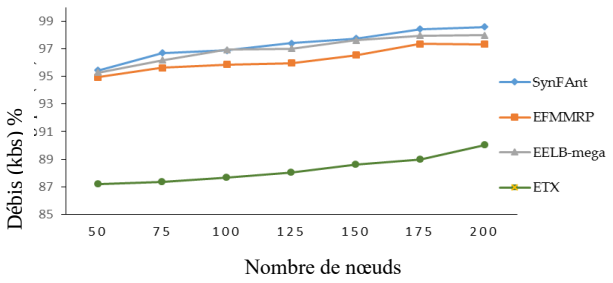


Figure 7.15. Débis par rapport au nombre de nœuds.

La figure 7.17 illustre la simulation du retard de bout en bout en fonction de la variation de vitesse. On voit clairement que le délai de bout en bout causé par EFMMRP et ETX est plus important que le protocole SynFAntPN. Une très faible augmentation du retard pour le protocole et l'ETX entre les deux vitesses 5 et 15 m/s est observée, après ce point, le retard augmente pour les trois protocoles. De plus, la vitesse de la mobilité est proportionnelle au mouvement des nœuds, la destruction fréquente des chemins doit donc être suivie du déclenchement du processus de maintenance de la route. Cela affecte le délai de livraison des paquets transmis car il faudra plus de temps pour arriver à leur destination.

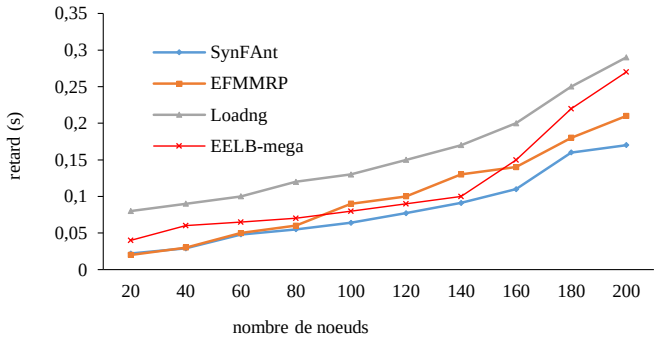


Figure 7.16. Délai de livraison par paquet en fonction du nombre de nœuds.

D. Taux d'acceptation des flux de QoS

Nous introduisons une nouvelle métrique (ϑ), qui représente le taux d'acceptation des flux de qualité de service et est définie comme suit:

$$\vartheta = \frac{nombre\ de\ flux\ Qos\ autorisés\ correctement}{nombre\ total\ de\ flux\ QoS\ dans\ le\ réseau}$$

Un flux de qualité de service correctement accepté est un flux qui n'a pas subi de dégradation de son débit supérieure à 5% au cours de la transmission [24]. Par conséquent, cette situation implique que l'estimation de la larg eur de bande résiduelle différentielle et la phase de contrôle d'admission sont fiables. Cette métrique nous permettra également d'estimer la fiabilité de l'estimation différentielle de la largeur de bande résiduelle. Une estimation erronée conduirait

. irrémédiablement à une dégradation du débit des flux de qualité de service et, par conséquent, à une diminution de la valeur de la variable ϑ.

La figure 7.18 montre la valeur du paramètre de protocole utilisée. Évidemment, plus le réseau est dense, plus le taux d'acceptation φ des flux de QoS est bas car la largeur de bande résiduelle des liaisons devient considérablement réduite (c'est-à-dire que la capacité reste constante alors

que le nombre de liaisons augmente). Lorsque le réseau est de faible densité (c'est-à-dire entre 10 et 20 nœuds), le taux d'acceptation est relativement élevé pour notre protocole (63%), alors que le protocole EFMMRP achemine environ 51% des flux de QoS. Ainsi, les deux mécanismes de différenciation de flux et d'estimation différenciée de la largeur de bande résiduelle permettent une augmentation du taux d'acceptation des flux de qualité de service. Cependant, lorsque le réseau est modérément dense (c'est-à-dire entre 80 et 120 nœuds), le taux d'acceptation des flux de QoS de tous les protocoles commence à diminuer. Cependant, le protocole SynFAnt fournit toujours jusqu'à 50% des flux de QoS présents. Enfin, lorsque le réseau est très dense (c'est-à-dire entre 140 et 180 nœuds), une réduction du débit des flux Best Effort est insuffisante pour garantir des ressources pour le flux de qualité de service compte tenu du fait que la bande passante résiduelle des liaisons devient très faible. . Cependant, avec SynFAnt, 26% des flux de QoS sont toujours acheminés avec les conditions requises, alors que les autres protocoles acheminent au moins 17% de ces flux.

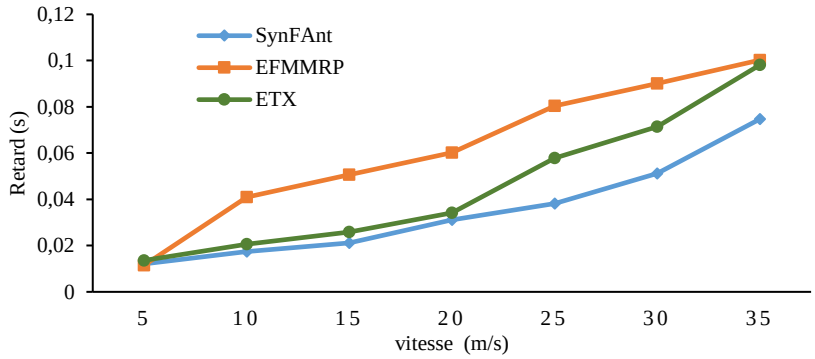


Figure 7.17. Délai de livraison d'un paquet par rapport à la mobilité.

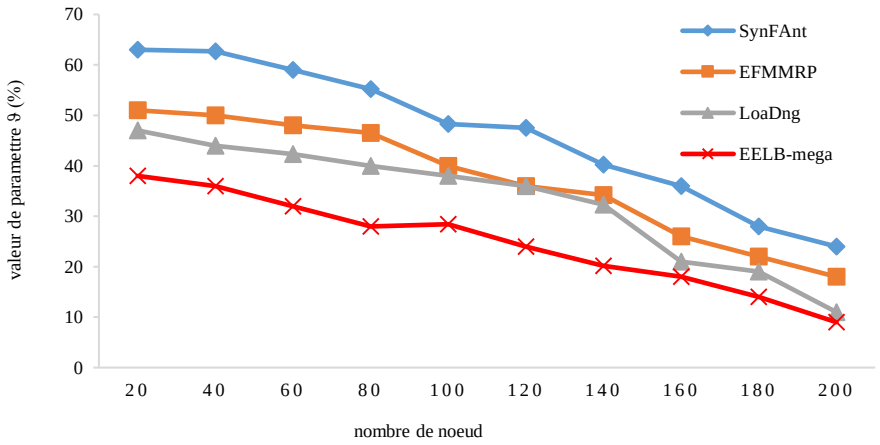


Figure 7.18. Taux d'acceptation des flux de QoS avec les protocoles SynFAnt, EFMMRP, LOADng et EELB-méga

7.5. Conclusion

Dans ce chapitre, un nouveau protocole de routage basé sur le système SyncFAnt pour les réseaux ad hoc, est proposé. L'idée de base est de proposer une solution adaptative permettant de réduire les encombrements et les retards de bout en bout en vérifiant les valeurs de confiance de chaque lien du réseau et d'éviter les routes et les nœuds qui enregistrent des erreurs et des défauts. Le protocole utilise des paramètres tels que la bande passante, la durée de vie et la fiabilité. De plus, une nouvelle formule prenant en compte l'énergie, le nombre d'hôtes voisins, le débit efficace et le nombre de paquets envoyés et reçus pour estimer les valeurs de confiance de chaque nœud est proposée.

Le processus pour trouver les meilleures routes dans le protocole proposé est basé sur la logique floue et le système de colonie de fourmis. Ce dernier est une heuristique inspirée d'un domaine biologique intéressé par l'étude du comportement des fourmis. La solution proposée utilise les deux systèmes en raison de leur intelligence et de leur capacité d'adaptation au changement de l'environnement. Cette solution proposée pour contrôler intelligemment le flux dans les MANET présente les avantages suivants: le contrôle est préventif et rapidement adapté aux changements qui se produisent, et le protocole proposé permet également de détecter les nœuds défectueux et de proposer rapidement de nouvelles tables de routage, afin d'éviter des retards

Conclusion Générale & Perspectives

Le développement de méthodes et d'outils de surveillance des systèmes est une préoccupation majeure des industriels et de la communauté scientifique du domaine. Ce type d'outils permet de surveiller l'état de santé du système et de détecter les éventuels comportements anomaux de ce dernier. Le travail présenté dans cette thèse s'inscrit dans ce contexte. Nous nous sommes intéressés à la surveillance des Systèmes à Événements Discrets ou nous avons pris les réseaux de communication ad hoc MANET comme exemple. Les principales contributions de notre travail sont les suivantes :

- Utilisation des réseaux de Petri Synchronisé flous :

Nous avons développé une approche de surveillance des SED basée sur une extension de RdP détaillé dans le chapitre 6. Le formalisme des RdP offre un grand pouvoir de modélisation des phénomènes complexes liés au fonctionnement du système, et ceci de manière compacte par rapport aux automates à états finis. Ils sont modulaires et donc parfaitement adaptés aux systèmes actuels composés de plusieurs modules qui interagissent entre eux. L'extension synchronisée floue du formalisme considéré permet de prendre en compte les comportements aléatoires du système ainsi que les aspects temporels dès l'étape de la modélisation.

- Protocole de routage d'un système de communication MANET:

L'un des contributions le plus important dans notre travail est l'intégration de la procédure de diagnostic / détection dans le protocole de routage a permis de réduire l'inférence mutuelle entre les nœuds et de réduire les itinéraires interrompus.

Dans SyncFAnt, la confiance QoS d'un nœud est calculée en fonction de paramètres de qualité et d'attitude de nœud tels que l'énergie, la bande passante, la mobilité et la fiabilité. Les paramètres sont agrégés à l'aide de la logique floue dans le calcul de la valeur de confiance QoS. La logique floue suivait la base de règles, où les règles sont dérivées pour refléter les conditions du réseau.

Le processus de routage est proposé pour les protocoles de routage unicast et multicast. Les exigences d'espace et de temps de la méthode proposée sont analysées avec des notations asymptotiques et les résultats sont pris dans un outil de simulation.

Perspectives

Plusieurs perspectives de recherche se dégagent au terme de cette thèse. Elles visent à améliorer et étendre les résultats obtenus. Parmi ces perspectives, on note deux points principaux : le premier point concerne le modèle utilisé. Une des limitations de celui-ci est d'avoir considéré que toutes les dates d'occurrence des événements suivent des lois exponentielles et qu'une faute est forcément liée à l'occurrence d'un événement. Le deuxième point concerne l'étude des deux propriétés de diagnosticabilité et de pronosticabilité.la troisième point est le développement d'un protocole de routage pour les systèmes de communication MANET avec plus de contrainte, comme l'étude la vitesse élever c'est à dire le passage vers le développement des protocole de routage qui inclus l'aspect de surveillance et de diagnostic dans les système VANET

Bibliographie

1. Ramadge, P.J. and W.M. Wonham, *The control of discrete event systems.* Proceedings of the IEEE, 1989. **77**(1): p. 81-98.
2. Cassandras, C.G. and S. Lafortune, *Introduction to discrete event systems*. 2009: Springer Science & Business Media.
3. Alexandre, P., *Contribution au diagnostic décentralisé des systèmes à événements discrets: Application aux systèmes manufacturiers*. 2006.
4. Petri, C.A., *Kommunikation mit Automaten: Institut für Instrumentelle Mathematik.* Schriften des IIM Nr, 1962. **2**.
5. Lapp, S.A. and G.J. Powers, *Computer-aided synthesis of fault-trees.* IEEE Transactions on Reliability, 1977. **26**(1): p. 2-13.
6. Lee, W.-S., et al., *Fault Tree Analysis, Methods, and Applications ð A Review.* IEEE transactions on reliability, 1985. **34**(3): p. 194-203.
7. Viswanadham, N. and T. Johnson. *Fault detection and diagnosis of automated manufacturing systems*. in *Proceedings of the 27th IEEE Conference on Decision and Control*. 1988. IEEE.
8. De Vries, R.C., *An automated methodology for generating a fault tree.* IEEE transactions on reliability, 1990. **39**(1): p. 76-86.
9. Frank, P.M., *Fault diagnosis in dynamic systems using analytic and knowledge-based redundancy-{A} survey and some new results.* 1990.
10. Willsky, A.S., *A survey of design methods for failure detection in dynamic systems.* Automatica, 1976. **12**(6): p. 601-611.
11. Frank, P.M., *Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results.* automatica, 1990. **26**(3): p. 459-474.
12. Scherer, W.T. and C.C. White, *A survey of expert systems for equipment maintenance and diagnostics*, in *Knowledge-Based System Diagnosis, Supervision, and Control*. 1989, Springer. p. 285-300.
13. Dague, P., et al. *Analog systems diagnosis*. in *Readings in model-based diagnosis*. 1992. Morgan Kaufmann Publishers Inc.
14. Dvorak, D.L., *Monitoring and diagnosis of continuous dynamic systems using semiquantitative simulation*. 1992, University of Texas at Austin.
15. Basile, F., P. Chiacchio, and G. De Tommasi. *Online diagnosis of discrete event systems based on Petri nets*. in *2008 9th International Workshop on Discrete Event Systems*. 2008. IEEE.
16. Tommas, F.B.P.C.G.D., *An efficient approach for online diagnosis of discrete event systems.* IEEE Trans. on Automatic Control, 2009. **54**(4): p. 748 - 759.
17. Benveniste, A., et al., *Diagnosis of asynchronous discrete-event systems: a net unfolding approach.* IEEE Transactions on Automatic Control, 2003. **48**(5): p. 714-727.
18. Boel, R.K. and J.H. van Schuppen. *Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers*. in *Sixth International Workshop on Discrete Event Systems, 2002. Proceedings.* 2002. IEEE.
19. Chung, S.-L., *Diagnosing PN-based models with partial observable transitions.* International Journal of Computer Integrated Manufacturing, 2005. **18**(2-3): p. 158-169.
20. Debouk, R., S. Lafortune, and D. Teneketzis, *Coordinated decentralized protocols for failure diagnosis of discrete event systems.* Discrete Event Dynamic Systems, 2000. **10**(1-2): p. 33-86.
21. Dotoli, M., M. Fanti, and A. Mangini, *Fault detection of discrete event systems using Petri nets and integer linear programming.* IFAC Proceedings Volumes, 2008. **41**(2): p. 6554-6559.
22. Dotoli, M., M.P. Fanti, and A.M. Mangini. *Fault monitoring of discrete event systems by first order hybrid Petri nets*. in *Workshop on Petri Nets and Agile Manufacturing, a satellite event*

of the 29th Int. Conf. on Application and Theory of Petri Nets and Other Models of Concurrency. 2008.
23. Genc, S. and S. Lafortune, *Distributed diagnosis of place-bordered Petri nets.* IEEE Transactions on Automation Science and Engineering, 2007. **4**(2): p. 206-219.
24. Ghazel, M., A. Toguyéni, and M. Bigand, *A monitoring approach for discrete event systems based on a time Petri net model.* IFAC Proceedings Volumes, 2005. **38**(1): p. 331-336.
25. Hadjicostis, C.N. and G.C. Verghese. *Monitoring discrete event systems using Petri net embeddings*. in *International Conference on Application and Theory of Petri Nets*. 1999. Springer.
26. Jiang, S. and R. Kumar, *Failure diagnosis of discrete-event systems with linear-time temporal logic specifications.* IEEE Transactions on Automatic Control, 2004. **49**(6): p. 934-945.
27. Lefebvre, D. and C. Delherm, *Diagnosis of DES with Petri net models.* IEEE Transactions on Automation Science and Engineering, 2007. **4**(1): p. 114-118.
28. Lunze, J. and J. Schroder, *Sensor and actuator fault diagnosis of systems with discrete inputs and outputs.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2004. **34**(2): p. 1096-1107.
29. Ramírez-Treviño, A., et al., *Online fault diagnosis of discrete event systems. A Petri net-based approach.* IEEE Transactions on Automation Science and Engineering, 2007. **4**(1): p. 31-39.
30. van Schuppen, J.H., *System theory for system identification.* Journal of Econometrics, 2004. **118**(1-2): p. 313-339.
31. Moreira, M.V., T.C. Jesus, and J.C. Basilio, *Polynomial time verification of decentralized diagnosability of discrete event systems.* IEEE Transactions on Automatic Control, 2011. **56**(7): p. 1679-1684.
32. Dvorak, D. and B. Kuipers, *Process monitoring and diagnosis: A model-based approach.* IEEE expert, 1991. **6**(3): p. 67-74.
33. Lin, F., *Diagnosability of discrete event systems and its applications.* Discrete Event Dynamic Systems, 1994. **4**(2): p. 197-212.
34. Lin, F., J. Markee, and B. Rado. *Design and test of mixed signal circuits: a discrete-event approach*. in *Proceedings of 32nd IEEE Conference on Decision and Control*. 1993. IEEE.
35. Sampath, M., et al., *Diagnosability of discrete-event systems.* IEEE Transactions on automatic control, 1995. **40**(9): p. 1555-1575.
36. Sampath, M., et al., *Failure diagnosis using discrete-event models.* IEEE transactions on control systems technology, 1996. **4**(2): p. 105-124.
37. Sampath, M., S. Lafortune, and D. Teneketzis, *Active diagnosis of discrete-event systems.* IEEE Transactions on Automatic Control, 1998. **43**(7): p. 908-929.
38. Zad, S.H., R.H. Kwong, and W.M. Wonham, *Fault diagnosis in discrete-event systems: Framework and model reduction.* IEEE Transactions on Automatic Control, 2003. **48**(7): p. 1199-1212.
39. Prock, J., *A new technique for fault detection using Petri nets.* Automatica, 1991. **27**(2): p. 239-245.
40. Srinivasan, V. and M.A. Jafari, *Fault detection/monitoring using time Petri nets.* IEEE transactions on systems, man, and cybernetics, 1993. **23**(4): p. 1155-1162.
41. Wu, Y. and C.N. Hadjicostis, *Algebraic approaches for fault identification in discrete-event systems.* IEEE Transactions on Automatic Control, 2005. **50**(12): p. 2048-2055.
42. Balduzzi, F., A. Giua, and G. Menga, *First-order hybrid Petri nets: a model for optimization and control.* IEEE transactions on robotics and automation, 2000. **16**(4): p. 382-399.
43. Basile, F., P. Chiacchio, and G. De Tommasi, *An efficient approach for online diagnosis of discrete event systems.* IEEE Transactions on Automatic Control, 2009. **54**(4): p. 748-759.
44. Ushio, T., I. Onishi, and K. Okuda. *Fault detection based on Petri net models with faulty behaviors*. in *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*. 1998. IEEE.

45. Wen, Y. and M. Jeng. *Diagnosability analysis based on T-invariants of Petri nets*. in *Proceedings. 2005 IEEE Networking, Sensing and Control, 2005.* 2005. IEEE.
46. Wen, Y., C. Li, and M. Jeng. *A polynomial algorithm for checking diagnosability of Petri nets*. in *2005 IEEE International Conference on Systems, Man and Cybernetics*. 2005. IEEE.
47. Liu, B., M. Ghazel, and A. Toguyéni, *On-the-Fly and Incremental Technique for Fault Diagnosis of Discrete Event Systems Modeled by Labeled Petri Nets.* Asian Journal of Control, 2017. **19**(5): p. 1659-1671.
48. Li, B., M. Khlif-Bouassida, and A. Toguyéni, *On-the-fly Diagnosability analysis of labeled Petri nets using T-invariants.* IFAC-PapersOnLine, 2015. **48**(7): p. 64-70.
49. Cabasino, M.P., et al., *Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems.* Control Engineering Practice, 2011. **19**(9): p. 989-1001.
50. Giua, A. and C. Seatzu. *Fault detection for discrete event systems using Petri nets with unobservable transitions*. in *Proceedings of the 44th IEEE Conference on Decision and Control*. 2005. IEEE.
51. Cabasino, M.P., A. Giua, and C. Seatzu, *Fault detection for discrete event systems using Petri nets with unobservable transitions.* Automatica, 2010. **46**(9): p. 1531-1539.
52. Boel, R.K. and G. Jiroveanu, *The on-line diagnosis of time Petri nets*, in *Control of Discrete-Event Systems*. 2013, Springer. p. 343-364.
53. Jiroveanu, G. and R.K. Boel, *The diagnosability of Petri net models using minimal explanations.* IEEE Transactions on Automatic Control, 2010. **55**(7): p. 1663-1668.
54. Cabasino, M.P., C.N. Hadjicostis, and C. Seatzu. *Initial marking estimation in labeled Petri nets in a probabilistic setting*. in *53rd IEEE Conference on Decision and Control*. 2014. IEEE.
55. Boussif, A., *Contributions to model-based diagnosis of discrete-event systems*. 2016.
56. Boussif, A., B. Liu, and M. Ghazel. *An experimental comparison of three diagnosis techniques for discrete event systems*. 2017.
57. Haar, S., C. Rodríguez, and S. Schwoon. *Reveal your faults: It's only fair!* in *2013 13th International Conference on Application of Concurrency to System Design*. 2013. IEEE.
58. Cabasino, M.P., et al. *Diagnosability analysis of unbounded Petri nets*. in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. 2009. IEEE.
59. Jiang, S., et al., *A polynomial algorithm for testing diagnosability of discrete-event systems.* IEEE Transactions on Automatic Control, 2001. **46**(8): p. 1318-1321.
60. Yoo, T.-S. and S. Lafortune, *Polynomial-time verification of diagnosability of partially observed discrete-event systems.* IEEE Transactions on automatic control, 2002. **47**(9): p. 1491-1495.
61. Madalinski, A., F. Nouioua, and P. Dague, *Diagnosability verification with Petri net unfoldings.* International Journal of Knowledge-Based and Intelligent Engineering Systems, 2010. **14**(2): p. 49-55.
62. Cabasino, M.P., et al., *A new approach for diagnosability analysis of Petri nets using verifier nets.* IEEE Transactions on Automatic Control, 2012. **57**(12): p. 3104-3117.
63. Gougam, H.-E., Y. Pencolé, and A. Subias, *Diagnosability analysis of patterns on bounded labeled prioritized Petri nets.* Discrete Event Dynamic Systems, 2017. **27**(1): p. 143-180.
64. Ru, Y. and C.N. Hadjicostis, *Fault diagnosis in discrete event systems modeled by partially observed Petri nets.* Discrete Event Dynamic Systems, 2009. **19**(4): p. 551.
65. Lefebvre, D., *Fault diagnosis and prognosis with partially observed Petri nets.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014. **44**(10): p. 1413-1424.
66. Lefebvre, D., *On-line fault diagnosis with partially observed Petri nets.* IEEE Transactions on Automatic Control, 2013. **59**(7): p. 1919-1924.
67. Dotoli, M., et al., *On-line fault detection in discrete event systems by Petri nets and integer linear programming.* Automatica, 2009. **45**(11): p. 2665-2672.
68. Basile, F., P. Chiacchio, and G. De Tommasi, *On K-diagnosability of Petri nets via integer linear programming.* Automatica, 2012. **48**(9): p. 2047-2058.

69. Basile, F., P. Chiacchio, and G. De Tommasi, *Fault diagnosis and prognosis in Petri Nets by using a single generalized marking estimation.* IFAC Proceedings Volumes, 2009. **42**(8): p. 1396-1401.
70. Basile, F., M.P. Cabasino, and C. Seatzu, *State estimation and fault diagnosis of labeled time petri net systems with unobservable transitions.* IEEE Transactions on Automatic Control, 2014. **60**(4): p. 997-1009.
71. Cong, X., et al., *Decentralized diagnosis by Petri nets and integer linear programming.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017. **48**(10): p. 1689-1700.
72. Tong, Y., Z. Li, and A. Giua, *On the equivalence of observation structures for Petri net generators.* IEEE Transactions on Automatic Control, 2015. **61**(9): p. 2448-2462.
73. Clausen, T., et al., *Optimized link state routing protocol (OLSR).* 2003.
74. Bhatt, M.C., *Survey on Mobile Ad-Hoc Network Protocol.* Imperial Journal of Interdisciplinary Research, 2016. **2**.
75. Qayyum, A., L. Viennot, and A. Laouiti, *Multipoint relaying: An e) cient technique for Cooding in mobile wireless networks.* 2000.
76. Ogier, R., F. Templin, and M. Lewis, *Topology dissemination based on reverse-path forwarding (TBRPF)*. 2004, IETF RFC 3684, February.
77. Ogier, R.G., *Topology broadcast based on reverse-path forwarding (TBRPF).* Internet Engineering Task Force (IETF) draft, draft-ietf-manettbrpf-06. txt, 2002.
78. Perkins, C.E., *Dsdv routing over a multihop wireless network of mobile computers.* Ad hoc networking, 2000: p. 53-74.
79. Johnson, D.B., D.A. Maltz, and J. Broch, *DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks.* Ad hoc networking, 2001. **5**(1): p. 139-172.
80. Das, S.R., E.M. Belding-Royer, and C.E. Perkins, *Ad hoc on-demand distance vector (AODV) routing.* 2003.
81. Ilyas, M., *The handbook of ad hoc wireless networks*. 2002: CRC press.
82. Haas, Z.J., M.R. Pearlman, and P. Samar, *The bordercast resolution protocol (BRP) for ad hoc networks.* IETF, MANET Internet Draft, 2002: p. 13801-14853.
83. Jiang, M., *Cluster based routing protocol (CBRP).* IETF Internet-draft, 1999.
84. Tabatabaei, S. and K. Tabatabaei. *Routing and quality of service support for mobile ad hoc networks*. in *2010 2nd International Conference on Computer Engineering and Technology*. 2010. IEEE.
85. Ge, Y., T. Kunz, and L. Lamont. *Quality of service routing in ad-hoc networks using OLSR*. in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*. 2003. IEEE.
86. Chen, R., et al. *Integrated social and quality of service trust management of mobile groups in ad hoc networks*. in *2013 9th International Conference on Information, Communications & Signal Processing*. 2013. IEEE.
87. Cheng, N., K. Govindan, and P. Mohapatra. *Rendezvous based trust propagation to enhance distributed network security*. in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2011. IEEE.
88. Xia, H., et al., *Applying trust enhancements to reactive routing protocols in mobile ad hoc networks.* Wireless Networks, 2016. **22**(7): p. 2239-2257.
89. Govindan, K. and P. Mohapatra, *Trust computations and trust dynamics in mobile adhoc networks: A survey.* IEEE Communications Surveys & Tutorials, 2011. **14**(2): p. 279-298.
90. Theodorakopoulos, G. and J.S. Baras, *On trust models and trust evaluation metrics for ad hoc networks.* IEEE Journal on selected areas in Communications, 2006. **24**(2): p. 318-328.
91. Ramana, K.S., A. Chari, and N. Kasiviswanth, *A survey on trust management for mobile ad hoc networks.* International Journal of Network Security & Its Applications (IJNSA), 2010. **2**(2): p. 75-85.

92. Lim, K.H. and A. Datta. *Enhancing the TORA protocol using network localization and selective node participation*. in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)*. 2012. IEEE.
93. Naimi, S., et al. *Anticipation of ETX metric to manage mobility in ad hoc wireless networks*. in *International Conference on Ad-Hoc Networks and Wireless*. 2014. Springer.
94. Yadav, A.K., S.K. Das, and S. Tripathi, *EFMMRP: Design of efficient fuzzy based multi-constraint multicast routing protocol for wireless ad-hoc network.* Computer Networks, 2017. **118**: p. 15-23.
95. Kaliappan, M., S. Augustine, and B. Paramasivan, *Enhancing energy efficiency and load balancing in mobile ad hoc network using dynamic genetic algorithms.* Journal of Network and Computer Applications, 2016. **73**: p. 35-43.
96. Clausen, T., J. Yi, and U. Herberg, *Lightweight on-demand ad hoc distance-vector routing-next generation (LOADng): Protocol, extension, and applicability.* Computer Networks, 2017. **126**: p. 125-140.
97. Petri, C.A., *Kommunikationen mit automaten*. 1962, PhD Thesis, University of Bonn, 1962. English translation: Technical Report ….
98. Giua, A., C. Seatzu, and D. Corona, *Marking estimation of Petri nets with silent transitions.* IEEE Transactions on Automatic Control, 2007. **52**(9): p. 1695-1699.
99. Chen, S.-M., *A fuzzy reasoning approach for rule-based systems based on fuzzy logics.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996. **26**(5): p. 769-778.
100. Tsang, E.C., et al., *Refinement of generated fuzzy production rules by using a fuzzy neural network.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2004. **34**(1): p. 409-418.
101. Yeung, D.S. and E.C. Tsang, *Weighted fuzzy production rules.* Fuzzy sets and systems, 1997. **88**(3): p. 299-313.
102. Liu, H.-C., et al., *Dynamic adaptive fuzzy Petri nets for knowledge representation and reasoning.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013. **43**(6): p. 1399-1410.
103. Looney, C.G., *Fuzzy Petri nets for rule-based decisionmaking.* IEEE Transactions on Systems, Man, and Cybernetics, 1988. **18**(1): p. 178-183.
104. Chen, S.-M., J.-S. Ke, and J.-F. Chang, *Knowledge representation using fuzzy Petri nets.* IEEE Transactions on knowledge and data engineering, 1990. **2**(3): p. 311-319.
105. Hu, Z.-g., et al., *A reliable routing algorithm based on fuzzy Petri net in mobile ad hoc networks.* Journal of Central South University of Technology, 2005. **12**(6): p. 714-719.
106. Yu, Z., et al., *A reliable energy-efficient multi-level routing algorithm for wireless sensor networks using fuzzy Petri nets.* Sensors, 2011. **11**(3): p. 3381-3400.
107. Khoukhi, L., et al., *Toward fuzzy traffic adaptation solution in wireless mesh networks.* IEEE Transactions on Computers, 2012. **63**(5): p. 1296-1308.
108. Pouyan, A.A. and M. Yadollahzadeh Tabari, *FPN-SAODV: using fuzzy petri nets for securing AODV routing protocol in mobile Ad hoc network.* International Journal of Communication Systems, 2017. **30**(1): p. e2935.
109. Chiang, T.-C., C.-F. Tai, and T.-W. Hou, *A knowledge-based inference multicast protocol using adaptive fuzzy Petri nets.* Expert Systems with Applications, 2009. **36**(4): p. 8115-8123.
110. Tan, S., X. Li, and Q. Dong, *Trust based routing mechanism for securing OSLR-based MANET.* Ad Hoc Networks, 2015. **30**: p. 84-98.
111. Sun, J., S.-Y. Qin, and Y.-H. Song, *Fault diagnosis of electric power systems based on fuzzy Petri nets.* IEEE Transactions on Power Systems, 2004. **19**(4): p. 2053-2059.
112. Luo, X. and M. Kezunovic, *Implementing fuzzy reasoning Petri-nets for fault section estimation.* IEEE Transactions on Power Delivery, 2008. **23**(2): p. 676-685.
113. Liu, H.-C., Q.-L. Lin, and M.-L. Ren, *Fault diagnosis and cause analysis using fuzzy evidential reasoning approach and dynamic adaptive fuzzy Petri nets.* Computers & Industrial Engineering, 2013. **66**(4): p. 899-908.

114. Zhang, J., et al., *Reconfigurable coordination of distributed discrete event control systems.* IEEE Transactions on Control Systems Technology, 2016. **23**(1): p. 323-330.
115. Cheng, H., et al., *Fault diagnosis method based on Petri nets considering service feature of information source devices.* Computers & Electrical Engineering, 2015. **46**: p. 1-13.
116. Yang, H.-T. and C.-M. Huang, *Distribution system service restoration using fuzzy Petri net models.* International journal of electrical power & energy systems, 2002. **24**(5): p. 395-403.
117. Wu, J., S. Yan, and L. Xie, *Reliability analysis method of a solar array by using fault tree analysis and fuzzy reasoning Petri net.* Acta Astronautica, 2011. **69**(11-12): p. 960-968.
118. Wu, J., et al., *Reliability apportionment approach for spacecraft solar array using fuzzy reasoning Petri net and fuzzy comprehensive evaluation.* Acta Astronautica, 2012. **76**: p. 136-144.
119. Wu, Z. and S.-J. Hsieh, *A realtime fuzzy Petri net diagnoser for detecting progressive faults in PLC based discrete manufacturing system.* The International Journal of Advanced Manufacturing Technology, 2012. **61**(1-4): p. 405-421.
120. An, R. and W. Liang, *Unobservable fuzzy Petri net diagnosis technique.* Aircraft Engineering and Aerospace Technology, 2013. **85**(3): p. 215-221.
121. Liu, H.-C., et al., *Knowledge acquisition and representation using fuzzy evidential reasoning and dynamic adaptive fuzzy Petri nets.* IEEE transactions on cybernetics, 2012. **43**(3): p. 1059-1072.
122. Boel, K.R. and G. Jiroveanu, *Distributed contextual diagnosis for very large systems.* IFAC Proceedings Volumes, 2004. **37**(18): p. 333-338.
123. Jiroveanu, G. and R.K. Boel. *Contextual analysis of Petri nets for distributed applications*. in *16th Int. Symp. on Mathematical Theory of Networks and Systems (Leuven, Belgium)*. 2004.
124. Chettibi, S. and S. Chikhi, *Dynamic fuzzy logic and reinforcement learning for adaptive energy efficient routing in mobile ad-hoc networks.* Applied Soft Computing, 2016. **38**: p. 321-328.
125. Kacem, I., et al., *A new routing approach for mobile ad hoc systems based on fuzzy Petri nets and ant system.* IEEE Access, 2018. **6**: p. 65705-65720.
126. Shah, R.C. and J.M. Rabaey. *Energy aware routing for low energy ad hoc sensor networks*. in *2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002 (Cat. No. 02TH8609)*. 2002. IEEE.
127. Dorigo, M., V. Maniezzo, and A. Colorni, *Ant system: optimization by a colony of cooperating agents.* IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics, 1996. **26**(1): p. 29-41.
128. Di Caro, G., F. Ducatelle, and L.M. Gambardella, *AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks.* European Transactions on Telecommunications, 2005. **16**(5): p. 443-455.

ملخص: بسبب التعقيد الكبير الذي أصبح عليه حال أغلب الأنظمة وقصد التقليل من المستشعرات وجب تطوير طرق مراقبة فعالة. يتناسب موضوع هاته الأطروحة مع هذا الموضوع وأكثر تدقيقا تشخيص الاختلالات في الأنظمة ذات الأحداث المتقطعة. تعتبر شبكات الجوال والاتصال اللاسلكي أي دو استخدام البنية السلكية أو الإدارة من الأمثلة الجد المهمة في هذا السياق. نتناول في هاته الأطروحة موضوع بروتوكولات النقل للمعلومة ضمن هاته الأنظمة. من خلال هذا العمل نقترح بروتوكول فعال جدا يعتمد على نظام شبكة بيتري وكذا التفكير الضبابي بالإضافة إلى الاستعانة بخوارزمية النمل في إيجاد أحس الطرق بين عقدتين في شبكة التواصل. كما أدمج البروتوكول مع تقنية لتشخيص أي اختلال في النظام وهذا لضمان أكبر فعالية وثبات للشبكة. تمت مقارنة البروتوكول المقترح مع أربعة بروتوكولات، أثبت من خلال المقارنات نجاعته في تحسين جودة أداء الشبكة وضمان تدفق ثابت للمعلومة كما أثبت نجاعة في اكتشاف الاختلالات التي من شأنها تعطيل عمل الشبكة.

كلمات استدلالية: نظام بيتري الضبابي، أنظمة الاتصالات اللاسلكية، تشخيص، مراقبة، بروتوكول توجيه.

Résumé: La complexification des systèmes et la réduction du nombre de capteurs nécessitent l'élaboration de méthodes de surveillance de plus en plus efficaces. Le travail de cette thèse s'inscrit dans ce contexte et porte sur le diagnostic et le pronostic des Systèmes à Événements Discrets (SED).Le réseau ad hoc mobile (généralement appelé (MANET) regroupe un nombre important et relativement dense d'unités mobiles qui se déplacent sur n'importe quel territoire. Son seul moyen de communication est l'utilisation d'interfaces sans fil sans utiliser d'infrastructure préexistante ni d'administration centralisée. De plus, le routage devrait fournir une stratégie pour l'envoi de données à tout moment entre une paire de nœuds (c'est-à-dire source et destination) sur un réseau. Cependant, le principal problème consiste à déterminer un routage optimal des paquets sur le réseau. L'objectif principal du protocole proposé est de trouver l'investissement le moins coûteux en capacités nominales qui assure l'acheminement du trafic nominal et garantisse sa capacité de survie en cas de défaillance d'un arc ou d'un nœud et surveillé au même temps l'état de système. Dans ce contexte, le réseau de Petri synchronisé flou est utilisé dans la modélisation des fonctions de routage et de détection / décision utilisant une approche de transition floue synchronisée, où le système de fourmi est utilisé. Les résultats obtenus montrent l'efficacité du protocole SynFAnt (Fuzzy Ant System) synchronisé proposé par rapport à quatre protocoles. Le protocole de routage SynFAnt améliore le rapport de livraison des paquets, le débit, le délai de bout en bout et le taux d'acceptation des flux de qualité de service.

Mots clés : SED ; réseaux MANET ; réseaux de Petri floue ; diagnostic, surveillance ; protocole de routage.

Abstract. The complexification of systems and the reduction in the number of sensors require the development of increasingly effective surveillance methods. The work of this thesis fits in this context and deals with the diagnosis and the prognosis of Discrete Event Systems (SED). The mobile ad hoc network (usually called MANET) includes a large and relatively dense number of mobile units that move on any territory. Its only means of communication is the use of wireless interfaces without the use of pre-existing infrastructure or centralized administration. In addition, routing should provide a strategy for sending data at any time between a pair of nodes (i.e., source and destination) over a network. However, the main problem is to determine an optimal routing of packets on the network. The main objective of the proposed protocol is to find the least expensive investment in nominal capacities that ensures the routing of nominal traffic and guarantees its survivability in case of failure of an arc or node and monitored at the same time. In this context, the fuzzy synchronized Petri net is used in the modeling of routing and detection / decision functions using a synchronized fuzzy transition approach, where the ant system is used. The results obtained show the effectiveness of the synchronized SynFAnt (Fuzzy Ant System) protocol proposed with respect to four protocols. The SynFAnt routing protocol improves packet delivery reporting, throughput, end-to-end delay, and acceptance rate for QoS flows.

Keywords.**:** Discrete event systems; MANET; Routing protocol, Ad hoc, Fuzzy Petri net, Ant system; diagnosis