

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE MOHAMED KHIDER DE BISKRA

NUMÉRO D'ORDRE :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



T H È S E

Pour obtenir le titre de

Docteur en Sciences
Spécialité : INFORMATIQUE

Présentée par :

Soheyb AYAD

Une approche pour la découverte sémantique des services Web dans les réseaux mobiles ad-hoc

Thèse dirigée par :

Pr. Okba KAZAR

Soutenue devant le jury composé de :

Président :	Pr. Mohamed BENMOHAMED	Professeur	Université de Constantine 2
Rapporteur :	Pr. Okba KAZAR	Professeur	Université de Biskra
Examineurs :	Pr. Azeddine BILAMI	Professeur	Université Batna 2
	Dr. Hammadi BENNOUI	M.C.A	Université de Biskra
	Dr. Labib-Sadek TERISSA	M.C.A	Université de Biskra
	Dr. Nabila-Aicha BENHARKAT	M.C.A	INSA de Lyon – France

Année universitaire : **2015 – 2016**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

*Je tiens tout d'abord à remercier vivement mon encadreur **Pr. Kazar Okba**, ainsi que **Dr. Benharkat Aicha-Nabila** pour leurs aides et soutien et aussi pour leurs précieux conseils.*

Je veux également remercier tous les membres de jury :

***Pr. Benmohamed Mohamed**, Professeur à l'Université de Constantine 2, pour m'avoir fait l'honneur de présider mon jury de thèse, ainsi que **Pr. Bilami Azeddine**, Professeur à l'Université Batna 2, **Dr. Benharkat Nabila-Aicha**, Maître de conférences à l'INSA de Lyon - France, **Dr. Terrissa Sadek Labib**, Maître de conférences à l'Université de Biskra, **Dr. Bennoui Hammadi**, Maître de conférences à l'Université de Biskra, pour m'avoir fait l'honneur d'accepter de juger ce travail.*

Enfin je tiens à remercier tous ceux et celles qui m'ont aidé et soutenu de près ou de loin pour l'accomplissement de cette thèse.

Dédicaces

Je dédie ce modeste travail :

*À mon défunt frère **Ayad Mohamed Bachir**, que dieu lui accorde sa miséricorde.*

*À ceux que j'ai de plus cher au monde, **mes parents**,*

À mon épouse, mes frères et sœurs, ainsi que mes neveux.

À tous mes amis.

ملخص

عملية اكتشاف خدمات الويب في الشبكات مانات، يعتبر تحديا صعبا للغاية بسبب التغير المستمر في مخطط الشبكة وأيضا عدم وجود مستودع مركزي ثابت لنشر خدمات الويب، وقد اقترحت عدة طرق التي تستند على إما الكلمات الرئيسية و معرفات في عملية الاكتشاف والتي تمثل الخدمة التي سيتم البحث عنها أو باستخدام سيناريو معين من الاكتشاف، كل تلك الحلول المقترحة تحاول احترام قيود مانات مثل الطاقة، وعرض النطاق الترددي وسرعة التدفق، الخ. في هذه الرسالة نقدم نموذجا الجديد المقترح لقياس تكلفة استهلاك الطاقة الكلي في مانات اعتمادا على بروتوكولات اكتشاف خدمات الويب كما نقدم بروتوكول أمثل جديد لاكتشاف خدمات الويب في مانات استنادا على تقنية بين الطبقات مع نشر في عملية اكتشاف الطرق في الوقت نفسه المعلومات عن خدمات الويب الدلالي و قطر الاكتشاف الذي اقترحنه للحد من منطقة الاكتشاف في الشبكة. و في الأخير، نقدم نتائج المحاكاة والتي تظهر التحسين الكبير في مستوى استهلاك الطاقة ومتوسط التدفق.

Ⓜ **كلمات** مفاتيح: أدهوك، استهلاك الطاقة، مانات، اكتشاف خدمات الويب الدلالي، التدفق، بروتوكولات التوجيه

Résumé

Le processus de découverte des services dans les réseaux mobiles ad hoc est considéré comme un défi très difficile en raison de changement continu de la topologie du réseau ainsi que l'absence d'un répertoire central fixe pour la publication des services Web. Plusieurs approches ont été proposées qui sont basées soit sur des mots-clés ou des identificateurs représentant le service à rechercher ou à l'aide d'un scénario spécifique de découverte. Toutes ces solutions proposées tentent de respecter les contraintes de réseaux mobiles ad hoc (MANET) tels que l'énergie, la bande passante, le débit, etc. Dans cette thèse, nous présentons notre nouveau modèle proposé pour mesurer le coût de la consommation totale d'énergie dans les réseaux ad hoc en fonction des protocoles de découverte des services web. Nous présentons également un nouveau protocole de découverte de services web optimisé dans MANET basé sur des techniques de routage cross_layer avec la diffusion dans le processus de routage en même temps l'information sémantique concernant le service web recherché et un paramètre DISCOVERY_DIAMETER que nous avons proposé pour limiter la zone de découverte dans le réseau. Enfin, nous présentons les résultats de simulation de notre approche définie montrant une optimisation significative au niveau de consommation d'énergie ainsi le débit moyen.

Mots-clés: ad hoc, la consommation d'énergie, MANET, découverte sémantique, débit moyen, protocoles de routage.

Abstract

The web services discovery process in mobile ad hoc networks is considered as a very difficult challenge due to the continuous change in the topology of the network and also the lack of a fixed central directory for publishing web services. Several approaches have been proposed which are based on either keywords or identifiers representing the service to be searched or by using a specific scenario of discovery. All of those proposed solutions try to respect the constraints of mobile ad hoc networks MANET such as energy, bandwidth, throughput, etc. In this thesis we present our new proposed model for measuring the cost of the overall energy consumption in ad hoc networks depending on the web services discovery protocols. We also present a new optimized web services discovery protocol in MANET based on cross_layer routing techniques with the dissemination in the routing process at the same time the semantic web services information and a DISCOVERY_DIAMETER parameter that we have proposed to limit the area of discovery in the network. Finally, we present simulation results of our defined approach showing a significant optimization of the energy consumption level and the average throughput.

Keywords: Ad hoc, Energy consumption, MANET, Semantic web services discovery, Average Throughput, Routing protocols.

Table des matières

Introduction générale	14
1. Motivations et objectifs.....	14
2. Organisation de la thèse	16
Chapitre I: Etat de l'art.....	18
I.1 Introduction	18
I.2 Généralités sur MANET.....	19
I.2.1 Les réseaux mobiles ad hoc (MANET).....	19
I.2.2 Caractéristiques des MANETs.....	19
I.2.3 Contraintes liées aux MANETs	20
I.2.4 Domaines d'application des réseaux ad hoc	22
I.2.5 MANET et Internet.....	24
I.2.6 Les protocoles de routage dans MANET.....	25
I.2.6.1 Les protocoles proactifs	26
I.2.6.1.1 DSDV	26
I.2.6.1.2 OLSR	27
I.2.6.2 Protocoles Réactifs.....	28
I.2.6.2.1 AODV	29
I.2.6.2.2 DSR.....	31
I.2.6.3 Protocoles Hybrides	32
I.2.6.3.1 ZRP	33
I.3 Architectures Cross-Layer pour les réseaux sans fil.....	35
I.3.1 Le concept du Cross-Layer	35
I.3.2 La communication dans les architectures Cross-layer.....	36
I.3.2.1 Communication directe entre les couches.....	37
I.3.2.2 Communication via une base de données partagée.....	37
I.3.3 Les approches Cross-layer dans les réseaux sans fil.....	38
I.3.3.1 Les projets traitant la problématique Cross-layer.....	39
I.3.3.1.1 Le projet 4MORE.....	39
I.3.3.1.2 Le projet PHOENIX.....	39
I.3.3.1.3 Le projet ENTHRONE II	40
I.3.4. Discussion.....	40
I.4 Service Web et Web sémantique.....	41
I.4.1 Les services Web	41
I.4.1.1 Définition	41
I.4.1.2 L'architecture orientée service.....	42
I.4.1.3 Architecture des services Web.....	43
I.4.1.4 Technologies de base des services Web.....	44
I.4.1.5 Discussion	44
I.4.2 Le Web sémantique.....	45
I.4.2.1 Naissance de Web sémantique	45
I.4.2.2 Architecture du Web sémantique.....	45
I.4.2.3 Principe du Web sémantique.....	46
I.4.3 Les services web sémantiques	48

I.4.3.1	Présentation	48
I.4.3.2	Langages de modélisation des services Web	49
I.4.3.3	Approches existantes de découverte sémantique des services web	49
I.4.3.4	Approche de Paolucci	50
I.4.3.4.1	Présentation	50
I.4.3.4.2	Adaptation avec le standard UDDI	50
I.4.3.4.3	Architecture de découverte combinée	51
I.4.3.4.4	Le module correspondance.....	52
I.4.4	Découverte des services dans MANET	54
I.5	Conclusion	56
Chapitre II: Synthèse des travaux existants.....		57
II.1	Introduction	57
II.2	Travaux existants.....	57
II.2.1	Approches adaptées dans MANET	57
II.2.1.1	SLP (Service Location Protocol).....	57
II.2.1.2	Salutation.....	58
II.2.1.3	JINI.....	58
II.2.1.4	UPnP (Universal Plug and Play)	59
II.2.2	Approches destinées pour MANET	60
II.2.2.1	La description du service.....	60
II.2.2.1.1	DReggie.....	60
II.2.2.1.2	Konark.....	61
II.2.2.1.3	WSDRS (Web Service Discovery Replication and Synchronization)	61
II.2.2.2	La technique de découverte.....	62
II.2.2.2.1	SEDIRAN.....	62
II.2.2.2.2	The need for cross-layer service discovery in MANETs	63
II.2.2.2.3	PDP (Pervasive Discovery Protocol)	63
II.2.2.2.4	GSD (Group based Service Discovery).....	64
II.2.2.3	La qualité de service QoS.....	65
II.2.2.3.1	TrustSD: Trust based service discovery in mobile ad-hoc networks	65
II.2.2.3.2	Ant based resource discovery and mobility aware trust management for mobile grid systems.....	65
II.2.2.3.3	Trust based security mechanism for service discovery in MANET	66
II.2.2.4	Le type du réseau.....	66
II.2.2.4.1	Efficient integration of web services in ambient-aware sensor network applications.....	66
II.2.2.4.2	An evaluation of mobility effect on tiny service discovery protocol for wireless sensor networks	67
II.2.2.4.3	Autres travaux sur les Sensor Web Services	68
II.3	Conclusion.....	68
Chapitre III: Découverte des services web dans MANET		69
III.1	Introduction.....	69
III.2	Contribution 1 : Evaluation du coût de l'énergie	69
III.3	Contribution 2 : Une nouvelle approche de découverte optimisée.....	74
III.3.1	Motivations.....	74
III.3.2	Architecture proposée	75
III.3.3	Le paquet de découverte RREQSSD.....	76
III.3.4	Le paquet de réponse RREPSSD.....	79

III.3.5 Un routage et une découverte de service web optimisée.....	80
III.3.6 Validation de l'approche proposée.....	83
III.4 Conclusion.....	85
Chapitre IV: Simulations et résultats	87
IV.1 Introduction.....	87
IV.2 Les différents simulateurs réseau.....	87
IV.2.1 Network Simulator NS2.....	87
IV.2.2 OMNET++	88
IV.2.3 OPNET.....	89
IV.2.4 GloMoSim.....	90
IV.2.5 J-Sim	90
IV.2.6 SENS.....	90
IV.2.7 PAWiS.....	91
IV.2.8 SENSE.....	91
IV.2.9 IDEA1	91
IV.3 Implémentation	92
IV.3.1 Mesure des performances.....	93
IV.3.1.1 Premier scenario (Exemple d'application).....	95
IV.3.1.1.1 Présentation	95
IV.3.1.1.2 Simulation et évaluation avec NS2.....	96
IV.3.1.1.3 Impact de nombre des Inputs/Outputs sur l'énergie globale	98
IV.3.1.1.4 Les paquets de découverte.....	99
IV.3.1.2 Deuxième scenario (Expérimentations avec 100 nœuds).....	100
IV.3.1.2.1 L'énergie globale.....	101
IV.3.1.2.2 Quantité de paquets générés dans le réseau.....	104
IV.3.1.2.3 La mobilité	105
IV.3.1.2.4 Le débit moyen.....	107
IV.3.1.2.5 Diamètre de Découverte (Discovery_Diameter)	108
IV.4 Conclusion	111
Conclusion Générale	112
1. Conclusion.....	112
2. Perspectives.....	113
Liste des publications.....	114
Références bibliographiques	115

Liste des figures

Figure 1. Evolution de l'informatique	15
Figure 2. De l'informatique à l'intelligence ambiante	15
Figure 3. MANET et Internet	24
Figure 4. Type de protocoles de routage dans les réseaux ad hoc	25
Figure 5. Les modèles de communication Cross-layer	37
Figure 6. Les approches du Cross-layer	38
Figure 7. Les protocoles des services web	42
Figure 8. Architecture des services web	43
Figure 9. Les niveaux du Web sémantique	46
Figure 10. Approche combinée DAML-S/UDDI	51
Figure 11. Niveaux de correspondances	52
Figure 12. Architectures de découverte de services dans MANET	55
Figure 13. Comparaison entre approches de découverte en matière de coût d'énergie	73
Figure 14. Architecture du système	75
Figure 15. Le paquet RREQSSD	76
Figure 16. Organigramme de réception du paquet RREQSSD	77
Figure 17. Structure du nouveau paquet RREPSSD	79
Figure 18. Organigramme de réception du paquet RREPSSD	80
Figure 19. Processus de routage et de découverte des services web	82
Figure 20. Ontologie globale utilisée	83
Figure 21. Défis de l'approche proposée en regard du coût	85
Figure 22. Modèle de nœuds mobile sous NS2 [JOSE 12]	88
Figure 23. Interface de configuration du modèle de nœud dans OMNET++	88
Figure 24. Architecture d'un nœud mobile sur OMNET++ [NITI 11]	89
Figure 25. Modèle d'un nœud sur SENS [SUND 04]	91
Figure 26. Modèle du nœud dans IDEA1 [NAVA 11]	92
Figure 27. L'architecture de nœud proposée	92
Figure 28. Ontologie du domaine médicale	96
Figure 29. Le premier scenario sous NAM	97
Figure 30. Energie consommée par rapport au nombre d'informations sémantiques dans les paquets de routage	98
Figure 31. Nombre de paquets de découverte dans T_{disc}	99
Figure 32. Scenario durant le temps de simulation	100
Figure 33. Résultat d'exécution du deuxième scenario sous NAM	101
Figure 34. La consommation de l'énergie dans le temps	102
Figure 35. Energie globale consommée par chaque type de paquet	103
Figure 36. Quantité et types de paquets générés dans le réseau	104
Figure 37. Quantité des paquets générés durant le scenario	105
Figure 38. Impact de la mobilité des nœuds sur l'énergie globale consommée	106
Figure 39. Impact de la mobilité des nœuds sur les paquets de découverte générés	107
Figure 40. Le débit moyen	108

Figure 41. L'énergie consommée en utilisant DD	108
Figure 42. Le débit moyen en utilisant DD	109
Figure 43. Impact de Diamètre de Découverte sur le débit moyen par rapport aux paquets générés	110
Figure 44. Comparaison entre la quantité de paquets générés avec et sans DD	110

Liste des tableaux

Tableau 1. Comparaison entre les approches existantes	70
Tableau 2. Les services web offerts	81
Tableau 3. Résultat de matching	82
Tableau 4. Défi de l'approche proposée	84
Tableau 5. Informations concernant le service web recherché	95
Tableau 6. Les services web offerts	95
Tableau 7. Scénario à 30 nœuds	96
Tableau 8. Résultats de Matching	98
Tableau 9. Paramètres de simulation deuxième scénario	100

Introduction générale

1. Motivations et objectifs

Le concept d'informatique ubiquitaire a été introduit par Marc Weiser [WEIS 91] pour désigner sa vision futuriste de l'informatique. Il imaginait un monde peuplé d'objets informatiques et numériques qui seraient reliés en réseaux à très grande échelle et interagiraient de manière autonome et transparente afin d'accomplir diverses tâches de la vie quotidienne. L'idée est de permettre aux utilisateurs d'accéder aux différents services offerts par ces objets communicants, le plus naturellement possible, n'importe où, à tout instant et à partir de divers dispositifs.

La vision de Marc Weiser est devenue aujourd'hui possible grâce à la combinaison de trois facteurs principaux : la miniaturisation et la puissance des composants électroniques, la chute des coûts de production et l'omniprésence des technologies réseaux sans fil [WANT 05]. En effet, ces dernières années ont vu la miniaturisation des dispositifs informatiques et électroniques dotés de capacités de capture, de traitement et de communication, pendant que leur puissance de calcul continuait d'augmenter. Cette situation, associée à la chute des coûts de production, a favorisé la démocratisation de ces technologies et leur intégration dans de nombreux objets du quotidien. Comme le remarque Gérard Berry [BERR 07], «les téléphones, appareils photos et caméras vidéo, les montres, les lecteurs et instruments de musique, les contrôleurs enfouis dans les avions, les voitures ou encore l'électroménager deviennent des ordinateurs habillés autrement». En outre, le développement des technologies réseaux sans fil (e.g., Wi-Fi, NFC, 3G, 4G et 5G) a permis à ces nouveaux types d'objets communicants d'interagir spontanément avec l'utilisateur ou avec d'autres objets, aussi bien localement qu'à distance, et de favoriser leur mobilité. Les figures 1 et 2 illustrent cette évolution, en montrant respectivement comment la diminution de la taille et des coûts des calculateurs a permis de passer d'un ordinateur partagé par tous à de nombreux ordinateurs simultanément utilisables par une même personne, et l'introduction progressive de l'ordinateur dans les activités humaines [JULI 11].

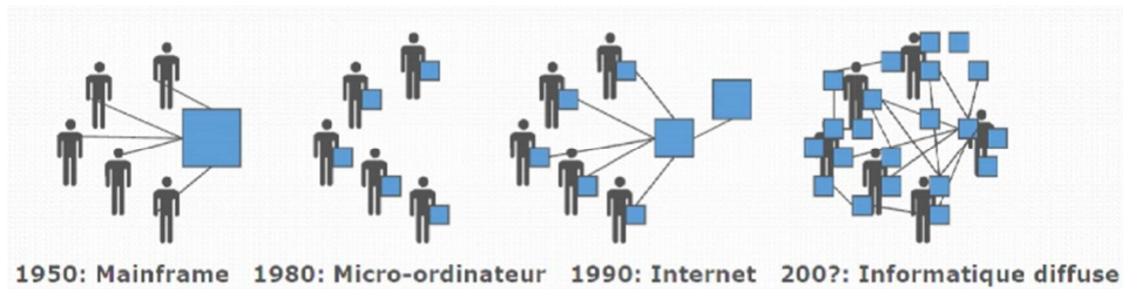


Figure 1. Evolution de l'informatique.

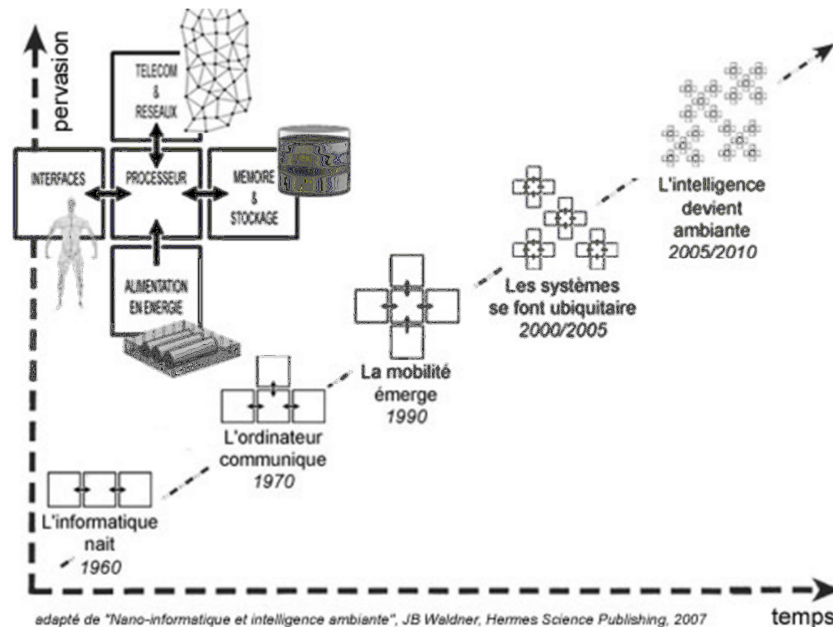


Figure 2. De l'informatique à l'intelligence ambiante.

Les réseaux mobiles ad hoc représentent une des technologies de communication les plus importantes dans l'informatique ubiquitaire, parce que dans de nombreux contextes d'échange d'informations entre les unités mobiles, ils ne peuvent pas compter sur une infrastructure de réseau fixe, mais sur la configuration rapide des connexions sans fil à la volée.

Dans une autre dimension, au niveau applicatif, les techniques actuelles de découverte des services web, effectuent seulement une recherche syntaxique basée sur les mots clés

avec une intervention humaine dans presque chaque étape de cycle de vie des services web. L'apparition de web sémantique a mené plusieurs chercheurs à l'exploiter pour pallier les limites actuelles de découverte des services web dans l'UDDI afin de rendre possible la recherche automatique (sémantique) des services Web en se basant sur leurs capacités fonctionnelles et sémantiques dans leurs descriptions telles que leurs inputs, outputs, préconditions, effets, . . .etc.

Dans notre travail nous avons en premier lieu proposé un modèle d'évaluation des protocoles de découverte de service web. Deuxièmement, nous avons exploité le web sémantique afin de proposer une approche optimisée de découverte de service web par rapport à la consommation d'énergie dans un réseau mobile ad hoc.

2. Organisation de la thèse

Cette thèse est constituée de quatre chapitres.

Le premier chapitre contient un état de l'art sur les réseaux de type MANET et les technologies des services web, où nous allons commencer par présenter des généralités sur les réseaux MANET, ensuite les principes des techniques d'optimisation Cross-Layer, leurs modes de fonctionnement ainsi que leurs types d'approches, et nous terminerons par la présentations des technologies des services web et du web sémantique et surtout leurs relations avec les réseaux MANET.

Dans le deuxième chapitre nous allons présenter les différentes approches de découverte des services web dans MANET en utilisant une classification que nous avons adoptée selon notre étude détaillée.

Le troisième chapitre est consacré à nos contributions où nous allons en premier lieu présenter notre première contribution publiée dans [AYAD 14] qui est la proposition d'un modèle d'évaluation des protocoles de découverte dans MANET par rapport à la consommation d'énergie, ensuite nous présenterons notre deuxième contribution qui est le nouveau protocole optimisé de découverte des services web dans MANET, que nous avons publié dans [AYAD 16].

Dans le dernier chapitre nous illustrerons notre étude de simulation du protocole proposé où nous allons présenter les simulateurs étudiés ainsi que les résultats d'implémentation avec le simulateur NS2 et nous terminerons par une discussion et évaluation des résultats finaux par rapport à un ensemble de paramètres tel que l'énergie consommée, la mobilité, le débit moyen ainsi que le paramètre Discovery_Diameter.

À la fin, nous clôturons cette thèse par une conclusion générale et des perspectives sur des travaux futurs.

Chapitre I :

État de l'art sur les technologies

MANET et Services Web

I.1 Introduction

Les réseaux sont aujourd'hui devenus des enjeux majeurs pour l'avenir. L'internet et son succès et le développement technologique avec l'apparition d'ordinateurs portables, tablettes, téléphones intelligents, objets intelligents connectés, véhicules connectés, connexion 3G, 4G et 5G ...etc., sont autant de facteurs qui changent notre mode de vie. Ce changement est dû aux moyens de communication ainsi qu'à la mobilité des équipements. Cette mobilité est un enjeu important des réseaux d'aujourd'hui et de demain puisque parallèlement à ces changements, de nouvelles utilisations sont apparues. Actuellement, tout réseau sans fil doit, en plus d'offrir un moyen de communication, respecter les contraintes qui lui sont imposées par les applications. Il doit offrir un débit suffisamment élevé et doit pouvoir être facilement et rapidement déployé.

Dans ce chapitre, nous allons en premier lieu présenter des généralités sur les réseaux MANETs ensuite nous allons donner des définitions sur le principe de Cross-Layer et ses avantages en terme d'optimisation et enfin nous terminerons ce chapitre par une présentation des services web sémantiques et leur relation avec les réseaux MANETs.

I.2 Généralités sur MANET

Dans cette partie nous allons présenter des généralités sur les réseaux de type MANET (Mobile Adhoc NETWORK).

I.2.1 Les réseaux mobiles ad hoc (MANET)

Les réseaux ad-hoc [PERK 00] [SIVA 04] sont apparus dans les années soixante-dix. Ce sont des réseaux qui peuvent fonctionner sans infrastructure fixe en établissant des communications directes ou multi-sauts entre les équipements. Un réseau mobile ad-hoc peut être défini comme suit.

***Définition** Un réseau mobile ad-hoc (MANET, Mobile Ad-Hoc Network) est un réseau auto organisé formé spontanément à partir d'un ensemble d'entités mobiles communicantes (ordinateurs portables, téléphones mobiles,...etc.) sans nécessiter d'infrastructure fixe préexistante.*

Contrairement aux réseaux cellulaires qui requièrent un effort important de planification pour leurs déploiements, les réseaux ad-hoc peuvent être déployés facilement et rapidement en permettant des échanges directs entre stations mobiles. Ainsi, le fonctionnement du réseau repose sur les stations mobiles elles-mêmes : celles-ci interviennent à la fois comme terminaux pour communiquer avec les autres usagers et comme routeurs afin de relayer le trafic pour le compte d'autres utilisateurs. En particulier, lorsque deux stations ne peuvent communiquer directement parce qu'elles ne se trouvent pas dans le même voisinage, une communication multi-sauts est initialisée en utilisant les nœuds intermédiaires : les paquets sont transmis de station en station avant d'atteindre la destination souhaitée [REMI 06].

I.2.2 Caractéristiques des MANETs

Un réseau mobile ad hoc (réseau MANET) possède des caractéristiques particulières [CORS 99] comparé aux autres réseaux sans fil [DAVI 08] :

- *Mobile* : Les stations ne sont pas fixes dans les réseaux MANETs. Elles peuvent se déplacer et sont entièrement indépendantes. A tout moment, de nouvelles stations peuvent rejoindre le réseau ou le quitter. Le changement de la topologie d'un réseau MANET dans le temps est un élément primordial.

- *Sans fil* : Les stations d'un réseau MANET utilisent un support sans fil pour communiquer entre elles. Elles partagent le même média lors des échanges d'informations. De fait, ce partage et ses conséquences (collisions, réservation de ressources...) sont autant d'éléments à prendre en compte.
- *Sans infrastructure* : Par nature, les réseaux MANETs ne dépendent pas d'une architecture fixe. Ils peuvent donc être facilement déployés.
- *Auto-organisé et distribué* : Les réseaux MANETs ne disposent pas de point central pour coordonner ou centraliser les échanges. De fait, ces réseaux doivent s'auto-organiser afin d'opérer. De plus, l'absence de centralisation demande à chaque acteur du réseau de participer au bon fonctionnement du réseau (distribution).
- *Multi-saut* : Comme la portée des stations est limitée, il peut s'avérer nécessaire que des stations agissent en tant que pont intermédiaire pour transmettre un paquet d'une source vers une destination. Par conséquent, les nœuds d'un réseau MANET agissent en tant que routeur et relayent les paquets qu'ils reçoivent pour participer au routage multi-saut.
- *Ressources limitées* : Les ressources limitées touchent toute la chaîne de communication d'un réseau MANET en commençant par les nœuds jusqu'aux liens de communication. Les terminaux étant mobiles, ils fonctionnent principalement sur batterie. La mobilité contraint également la puissance embarquée. La capacité des liens sans fil s'avère aussi limitée comparativement aux réseaux filaires. De même, le taux d'erreur est bien plus élevé que dans un réseau filaire.
- *Temporaire et rapidement déployable* : Ce type de réseau est intrinsèquement temporaire et rapidement déployable. Il n'a pas pour but de remplacer un réseau à infrastructure mais de le compléter ou de le remplacer lorsque nécessaire.

I.2.3 Contraintes liées aux MANETs

Les caractéristiques des réseaux ad hoc impliquent des contraintes spécifiques [CHEN 05] sur leur fonctionnement et donc leurs performances. Les principaux problèmes susceptibles d'être rencontrés dans un environnement mobile et ad-hoc peuvent être regroupés en deux catégories selon la source du problème [DAVI 08]:

a) Limitations dues au support de transmission

- *Partage du support de transmission* : Les stations mobiles opèrent sur la même bande de fréquence. Le partage du support de transmission peut engendrer des collisions. Ce problème est lié également à la diffusion des signaux.
- *Taux d'erreur élevé* : Les réseaux sans fil utilisent des ondes radio pour communiquer. Ces ondes ne peuvent pour autant s'affranchir des contraintes liées à leur medium de transmission, l'air. Les perturbations électromagnétiques, solaires ou les obstacles affectent les signaux transmis et sont de fait source de taux d'erreur en bit particulièrement élevés.
- *Capacité des liens variables* : Les réseaux sans fil doivent aussi faire face à la variabilité de leur support de transmission. De fait, les caractéristiques et performances des liens entre deux stations varient constamment. Ainsi, certains liens bidirectionnels, peuvent voir leurs performances chuter et devenir des liens unidirectionnels.
- *Faible débit* : La modestie des débits des réseaux sans fil est un élément souvent mis en avant. Comparés à certains réseaux filaires, les débits peuvent paraître faibles. Et dans le cadre de transferts multimédia nécessitant des échanges de données soutenus, ces débits peuvent ainsi poser problème.
- *Variation de la qualité du signal* : Le canal ne cesse de changer avec le temps. En effet, les conditions extérieures peuvent modifier les caractéristiques de ce canal, par exemple la pluie peut accroître le taux d'affaiblissement de la liaison sans fil. De même, l'apparition d'obstacles peut modifier le canal augmentant le nombre de trajets entre une source et une destination.
- *Sécurité* : Les signaux étant diffusés, ils peuvent être écoutés par toute station mobile se trouvant dans la même zone de couverture. La confidentialité de certaines informations nécessite l'utilisation de mécanismes de sécurité adéquats.

b) Limitations dues aux stations mobiles

- *Faible puissance* : Les stations mobiles sont la plupart du temps conçues pour une utilisation mobile. De fait, elles se doivent d'être légères, de petite taille et surtout doivent être capables de fonctionner de manière autonome (sur batterie). La prise en compte de tous ces éléments participe à la faible puissance de l'électronique embarquée.

- *Durée d'utilisation restreinte* : Les batteries ont une durée de vie limitée. De fait, le temps d'utilisation nomade d'une station est contraint par la capacité de sa batterie mais aussi par la puissance demandée (ressources processeur ou transmissions sans fil). Il est nécessaire de trouver un juste milieu entre ces composantes.
- *Rayon d'action* : La zone de couverture est fonction de la puissance d'émission que peut fournir une station. Le standard IEEE 802.11 définit la puissance maximale à 100mW. Réduire la puissance d'émission, pour notamment économiser de l'énergie, peut engendrer des liens unidirectionnels.
- *Modification de la topologie du réseau avec le temps* : Les stations pouvant être en constant déplacement, la topologie du réseau évolue également. Le voisinage d'un nœud peut varier continuellement : à tout moment des stations peuvent rejoindre ou quitter le réseau. La modification de la topologie est directement fonction de la vitesse de déplacement des stations et du rayon d'action du réseau. Avec un déplacement rapide et soutenu de l'ensemble des stations, la topologie ne cesse d'évoluer.
- *Altération des signaux* : Le déplacement des stations modifie la fréquence des signaux reçus par effet Doppler. Ainsi, à haute vitesse les signaux peuvent s'avérer incompréhensibles. Le contexte de notre étude permet la cohabitation d'applications multimédia ou temps-réel fortement consommatrices en bande passante avec un environnement mobile ad hoc. Nous investiguons, par conséquent, dans la suite de nos travaux les contraintes suivantes : le faible débit, la capacité des liens variables et la modification de la topologie du réseau avec le temps.

I.2.4 Domaines d'application des réseaux ad hoc

Les domaines d'applications des réseaux sans fil ad hoc sont nombreux et nous pouvons citer les suivants [DAVI 08]:

- *Applications collaboratives* : Les utilisateurs professionnels ont besoin d'applications particulières lors d'échanges entre collaborateurs. Ainsi, au cours de réunions ou de conférences, ces utilisateurs peuvent ressentir le besoin de former dans n'importe quel lieu un réseau pour s'échanger des informations. Les réseaux ad hoc sont bien appropriés à ces besoins.

- *Jeux Vidéo* : Les réseaux sans fil sont bien adaptés pour permettre l'échange d'informations entre applications personnelles. Ainsi, pour les utilisateurs voulant jouer en réseau, il est facile et à faible coût de déployer un réseau ad hoc.
- *Urgences* : Lors de catastrophes d'origine naturelles (comme les tremblements de terre, les tsunamis, les feux de forêt ou d'habitations...) ou non, les infrastructures préexistantes peuvent ne pas être opérationnelles compliquant d'autant plus les besoins de communications des moyens de secours. Les réseaux sans fil, par leur compacité et leur rapidité de déploiement, permettent aux différentes équipes de secours d'établir rapidement des liaisons et d'échanger des informations.
- *Militaires* : Lors d'interventions en milieu hostile, il peut être difficile ou trop encombrant d'utiliser un réseau à infrastructure. Les réseaux sans fil sont parfaitement bien adaptés à ce type d'environnement où les déplacements restent peu rapides et peu soutenus. A titre d'exemple, le département militaire Américain a développé le projet SLICE (Soldier Level Integrated Communications Environment) pour permettre la communication des soldats lors d'interventions militaires. L'idée sous-jacente est que chaque soldat soit équipé d'un ordinateur portable relié à un casque et un microphone. Le projet SLICE est censé créer un réseau sans fil ad hoc permettant le transfert de la voix entre différents soldats.
- *Systèmes embarqués* : Un bon exemple, pour l'utilisation des réseaux sans fil dans les applications embarquées est le projet V2V (Vehicle to Vehicle). En effet, un consortium de constructeurs automobiles s'est focalisé sur l'échange d'informations entre véhicules automobiles afin d'améliorer la sécurité des usagers de la route. Le but de ce projet est de faire communiquer, entre eux, au moyen d'un réseau et ce de manière transparente, plusieurs véhicules proches. En cas de danger (accident, bouchon, brusque ralentissement...), le premier véhicule détectant ce danger prévient les autres véhicules. Chacun de ces derniers véhicules informant à leur tour d'autres véhicules que les conditions de circulation ont évolué. Le conducteur est à ce moment-là prévenu par un voyant lumineux ou sonore du danger et de la conduite à adopter.
- *Réseaux de capteurs* : Les réseaux de capteurs sont des réseaux sans fil dont les équipements se déplacent très peu, et dont la durée de vie des batteries est limitée. Ces équipements peu coûteux permettent de fournir par exemple des informations sur la température à différents endroits d'une chambre froide, le niveau d'ensoleillement d'une pièce, la santé des animaux dans un zoo...

I.2.5 MANET et Internet

Bien qu'ils ne nécessitent pas d'infrastructure fixe, les réseaux ad-hoc sont souvent liés dans la pratique à une passerelle permettant l'interconnexion à l'Internet filaire, comme décrit à la Figure 3. Si l'on considère le scénario d'un opérateur de télécommunications fournissant un accès à l'Internet à travers un ou plusieurs points d'accès sans fil, le déploiement d'un réseau ad-hoc peut être justifié typiquement dans les situations suivantes

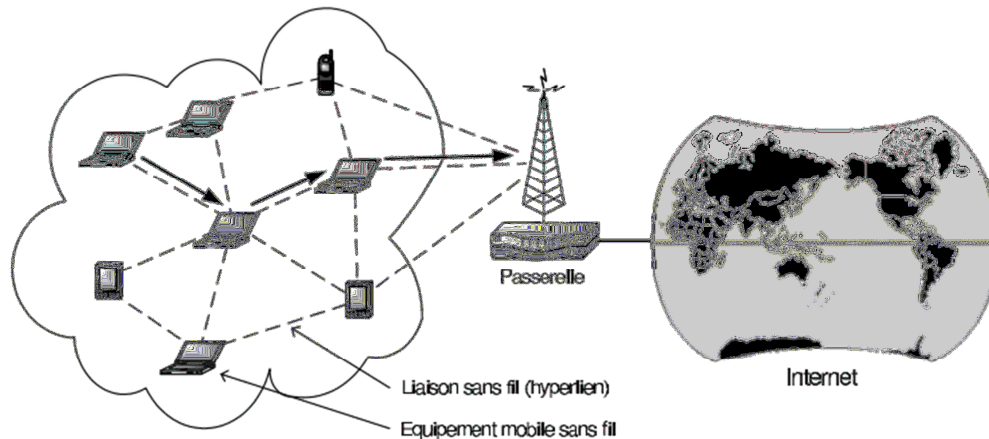


Figure 3. MANET et Internet.

- l'opérateur doit couvrir une zone de très faible densité (le coût d'installation de points d'accès s'avère élevé par rapport à la population qui est prête à souscrire au service) ou de très forte densité (le coût du câblage de l'ensemble de la population est important),
- l'opérateur doit fournir un service dans un environnement où le déploiement des points d'accès est techniquement difficile à réaliser (installation de points d'accès en régions montagneuses),
- le réseau fixe est endommagé (panne d'un point d'accès, catastrophes naturelles), les clients souhaitent continuer à disposer du service.

Dans les situations énoncées, les terminaux des utilisateurs passent en mode ad-hoc afin de communiquer entre eux par communications directes et multi-sauts : les nœuds du réseau interviennent en tant que routeurs pour (1) assurer la liaison entre des nœuds qui ne sont pas voisins directs et (2) permettre aux nœuds, qui n'ont pas de points d'accès (fonctionnels) dans leur voisinage, d'accéder à l'Internet en passant par des nœuds intermédiaires [REMI 06].

I.2.6 Les protocoles de routage dans MANET

Le routage dans les réseaux Ad-Hoc est assez délicat étant donnée la nature changeante de la topologie de ce type de réseaux. Les protocoles de routage pour les réseaux mobiles sont généralement classés d'abord en fonction de leur manière de découvrir le réseau, et après dans leur façon d'établir leurs tables de routage. Ces protocoles peuvent être classés suivant trois catégories : (Les protocoles proactifs; Les protocoles réactifs; Les protocoles hybrides).

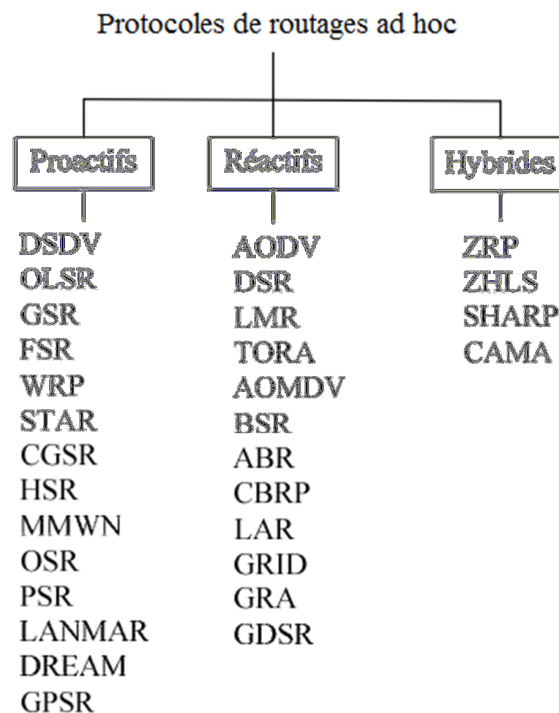


Figure 4. Type de protocoles de routage dans les réseaux adhoc.

Pour maintenir leur table de routage, les protocoles proactifs recherchent à intervalle régulier les différentes routes disponibles dans le réseau. Quand un paquet doit être transmis, sa route est donc connue à l'avance et peut ainsi être immédiatement utilisée. Les protocoles réactifs entreprennent la recherche d'une route uniquement avant de transmettre un paquet. Il existe par ailleurs une approche de routage hybride qui fait la combinaison entre les deux types précédents [ERME 04].

I.2.6.1 Les protocoles proactifs

Les protocoles de routage proactifs reprennent le principe du routage des réseaux filaires. Ils sont basés sur l'existence de tables de routage au niveau de chacun des nœuds. Lorsqu'un nœud du réseau souhaite envoyer un message, il consulte sa table de routage pour connaître la route à suivre jusqu'au destinataire du message. Le principal avantage de ces protocoles est leur réactivité [CHAN 07]. En effet, à tout moment chaque élément du réseau connaît un moyen d'atteindre les autres membres du réseau.

En revanche, il faut être capable d'actualiser les tables de routage en permanence pour tenir compte de la mobilité des nœuds, cela entraîne la diffusion de nombreux messages de contrôle qui engendrent un trafic important dans le réseau, réduisant ainsi la bande passante disponible pour envoyer des données. En plus, cette émission permanente de messages entraîne une consommation énergétique plus importante au niveau des nœuds du réseau. Les différents protocoles réactifs se différencient principalement par le mode de mise à jour des tables de routage.

Nous allons présenter les protocoles DSDV et OLSR. Ces protocoles sont les protocoles proactifs les plus répandus dans la littérature. Le protocole DSDV est une référence par son ancienneté, alors que le protocole OLSR est le seul représentant standardisé des protocoles proactifs.

I.2.6.1.1 DSDV

Il est basé sur l'algorithme du vecteur de distance utilisé dans RIP [MALK 98]. Le protocole à vecteur de distance permet de limiter l'échange des messages de contrôle de la topologie uniquement aux voisins d'un nœud. Ce point est extrêmement important pour préserver la bande passante disponible sur le réseau.

Le protocole DSDV utilise les propriétés de la diffusion pour transmettre les informations de routage. En effet, le grand avantage de la diffusion est qu'une trame émise par une station est entendue par l'ensemble de ses voisins. Périodiquement, chaque station diffuse l'ensemble de sa table de routage suivie d'un numéro pour dater l'information. Ce numéro est appelé numéro de séquence. A partir de deux numéros de séquence, il est possible de déterminer quelle information est la plus récente. La table de routage d'un nœud contient les informations liées à chaque route (adresse de destination, nombre de

nœuds pour joindre cette destination et numéro de séquence de la destination). A la réception de ces informations, les voisins mettent à jour leur table de routage en suivant un schéma bien précis. Toute entrée de la table de routage est mise à jour, seulement, si l'information reçue est plus récente, ou si elle a le même âge mais possède un nombre de nœuds plus faible.

A terme, le protocole DSDV fournit pour chaque destination, la route qui possède le plus faible nombre de nœuds. Pour être un protocole de routage complet, le protocole DSDV doit maintenir l'état des chemins. Pour cela, les nœuds détectent les ruptures de lien. Chaque nœud émet, périodiquement, ses informations de routage à l'ensemble de ses voisins. Si pendant un certain temps, un nœud ne reçoit plus les informations de routage d'un nœud voisin c'est que ce dernier ne fait plus partie de son voisinage. Un lien coupé affecte l'ensemble des routes utilisant ce lien. Un nœud, décelant une coupure, diffuse un paquet contenant l'ensemble des destinations ne pouvant plus être atteint à travers ce lien. Tout nœud, recevant un tel paquet, le propage immédiatement pour faire connaître au plus vite le changement de topologie. Un des problèmes de cet algorithme est qu'il réagit trop lentement aux mauvaises nouvelles. La destination doit prendre connaissance d'une coupure pour transmettre une mise à jour de la topologie [DAVI 08].

I.2.6.1.2 OLSR

Le protocole Optimized Link State Routing (OLSR) a été standardisé en 2003 [CLAU 03]. Son fonctionnement est basé sur l'algorithme à état de liens [MOY 98]. Un nœud du protocole à état de liens diffuse sa connaissance des voisins à l'ensemble de la topologie. De nombreux changements ont dû y être apportés pour être exploitable dans un réseau ad hoc. La bande passante étant limitée la diffusion de ses voisins à l'ensemble des nœuds du réseau est bien trop coûteuse. Le protocole OLSR prend en compte les spécificités de la diffusion (un paquet émis est reçu par l'ensemble des nœuds dans son voisinage immédiat) pour réduire le nombre de paquets nécessaires à l'échange de la topologie.

Chaque nœud doit déterminer l'ensemble de ses voisins. Pour cela périodiquement, ils transmettent des paquets, dits Hello, pour se faire connaître. Ce type de paquet comprend la totalité de la base de liens connue par l'émetteur du paquet. La base de liens d'un nœud regroupe l'ensemble des nœuds lui ayant transmis un paquet Hello. A la

réception des paquets Hello, chaque nœud dans le réseau connaît les nœuds situés dans son voisinage immédiat mais également à deux sauts.

Une fois les voisins découverts, les nœuds peuvent échanger les informations sur leur voisinage pour former la topologie du réseau. Cette fonction est attribuée à des nœuds particuliers sélectionnés parmi ses voisins à un saut. Ces nœuds sont appelés relais multipoints (MPRs) et sont les seuls capables de transmettre les informations de routage. Chaque nœud sélectionne un ensemble de MPRs relayant les informations de routage à l'ensemble des nœuds situés à deux sauts. Chaque MPR transmet, périodiquement, la liste des nœuds qui l'ont choisi comme MPR. Un tel paquet est, seulement, relayé par les nœuds sélectionnés en tant que MPRs.

Une fois la topologie connue par l'ensemble des nœuds du réseau, il suffit d'appliquer un algorithme, de type Dijkstra, pour déterminer les routes vers l'ensemble des nœuds distants. Chaque nœud connaît, ainsi, les routes les plus courtes vers les autres nœuds du réseau [DAVI 08].

I.2.6.2 Protocoles Réactifs

Les protocoles de routage réactifs (ou sur demande) ne maintiennent une route que si elle est utilisée. Lorsqu'un nœud source a besoin de transmettre des données vers un nœud destination, il doit au préalable déterminer une route. Pour cela, des informations de contrôle sont transmises sur le réseau. Comparés aux protocoles proactifs qui conservent les routes vers l'ensemble des stations du réseau dans leur table de routage, les protocoles réactifs ne conservent que les routes qui ont une utilité. Par conséquent, la taille des tables de routage contenues en mémoire est moins importante que pour les protocoles proactifs.

Trouver une route, lorsque la source en a besoin, crée une latence avant de l'obtenir. Pour certaines applications nécessitant un minimum de réactivité, ce délai peut être problématique. Ce délai n'est seulement occasionné au début de l'échange d'information (lors de l'établissement de la route) ou lorsqu'une route est rompue (lors du déplacement d'un nœud par exemple). Durant le laps de temps où le protocole de routage détermine la route nécessaire, les paquets provenant de la couche supérieure sont conservés en mémoire. Nous allons présenter les protocoles réactifs AODV et DSR. Ces

protocoles sont les protocoles réactifs ayant le plus de chances d'être utilisés dans les réseaux MANET du fait qu'ils sont standardisés [DAVI 08].

I.2.6.2.1 AODV

Le protocole de routage Ad hoc On-Demand distance Vector (AODV) [PERK 03] permet le maintien des routes utilisées. En fait, si le changement de statut d'un lien n'affecte pas une communication, aucun échange entre les nœuds n'est donc nécessaire. Les effets des changements de topologie sont ainsi localisés seulement aux routes rencontrant ces modifications et non à la globalité du réseau. Ce protocole est opérationnel seulement dans un environnement où les liens sont symétriques. Ce protocole met en œuvre différentes opérations pour réaliser et maintenir le routage : gestion de la connectivité locale, phase de découverte des routes, maintenance des routes.

La fonctionnalité de gestion de la connectivité locale est appliquée par les nœuds de la manière suivante. Chaque nœud émet périodiquement un paquet, nommé Hello. A la réception de ce paquet, les nœuds apprennent la présence des nœuds voisins. La connectivité locale est modifiée dans les cas suivants : un nœud reçoit un paquet Hello transmis par un nouveau voisin ou un nœud ne reçoit plus de paquets Hello durant un laps de temps défini.

La phase de découverte des routes par le protocole AODV est la suivante : A la réception d'un paquet de données par la source, elle vérifie dans sa table de routage si une route existe jusqu'à la destination. Si elle existe, le paquet est transmis vers le prochain nœud sinon la phase de découverte des routes est engagée. Le paquet est mis en file d'attente le temps d'obtenir une route puis la source diffuse une requête de création de routes, nommée RREQ. A la réception d'un paquet RREQ, un nœud met à jour la route inverse en direction de la source. Le nœud vérifie, ensuite, s'il connaît une route vers la destination. S'il en possède une, il envoie une requête de réponse, nommée RREP, en direction de la source. Sinon, il diffuse la requête RREQ à ses voisins. Lorsque la requête RREP transite vers la source, chaque nœud sur le chemin inverse met à jour sa table de routage avec, comme prochain nœud, l'adresse du nœud qui a émis la requête RREP. Le temporisateur de cette entrée dans la table de routage est mis à jour. Ce temporisateur indique qu'une route est toujours active s'il est non nul.

Pour chaque destination d'intérêt, un nœud maintient une unique entrée dans sa table de routage qui contient les champs suivants : adresse de la destination, numéro de séquence de la destination, prochain nœud sur le chemin vers la destination, nombre de sauts et d'autres paramètres relatifs à la route. L'utilisation du numéro de séquence permet de « dater » la route et d'éviter la présence de boucles. Si deux routes existent entre un nœud et la destination, le nœud conserve la route la plus récente. Si les deux routes sont découvertes simultanément, la route avec le plus faible nombre de sauts est conservée.

La phase de découverte des routes peut, aussi, être réalisée en utilisant une recherche de parcours en largeur. La source positionne, lors de la première tentative de recherche de route, le champ TTL de la requête RREQ à la valeur TTL_DEBUT. Elle positionne le temporisateur, d'attente d'un paquet RREP, à la valeur TEMPS_TRAVERSE millisecondes. La valeur TEMPS_TRAVERSE est calculée avec la formule suivante :

$$TEMPS_TRAVERSE = 2 \times (TTL + TEMPS_CONGESTION) \times TEMPS_TRAVERSE_NOEUD$$

Où TEMPS_TRAVERSE_NOEUD est le temps moyen estimé de la traversée d'un saut par un paquet et TEMPS_CONGESTION est un temps supplémentaire au cas où la requête RREP est retardée à cause d'une congestion.

Si le temporisateur initialisé à TEMPS_TRAVERSE arrive à échéance, c'est-à-dire que la source n'a pas reçu de réponse RREP dans le temps imparti, la source diffuse une nouvelle requête RREQ avec le champ TTL incrémenté avec la valeur TTL_INCREMENT. Ceci est réalisé jusqu'à ce que le TTL atteigne un certain seuil. Dans un tel cas, le TTL est positionné avec le diamètre du réseau, et la phase de découverte des routes devient identique à celle présentée précédemment. Après chaque tentative, le temporisateur d'attente d'une réponse RREP est positionné avec une valeur de TEMPS_TRAVERSE millisecondes.

Le nombre de sauts conservé dans une entrée invalidée de la table de routage (à cause d'un temps d'inactivité trop élevé par exemple, de perte de route...) indique le dernier nombre de sauts connu pour atteindre cette destination dans la table de routage. Lorsqu'une nouvelle route vers cette destination est requise, le TTL dans la requête RREQ est initialisé à ce nombre de sauts plus TTL_INCREMENT.

La phase de maintenance des chemins est réalisée en plusieurs étapes. La première étape consiste en la détection de la perte d'un chemin. Quand un nœud sur un chemin établi se déplace, les routes passant par ce nœud peuvent être rompues. Les nœuds en amont, détectant la perte de connectivité, préviennent les sources affectées en émettant une requête d'erreur, notée RERR. A la réception de ce paquet, le nœud source engage la deuxième étape de la maintenance des routes. Il entame une nouvelle phase de découverte des routes, si un chemin est toujours nécessaire [DAVI 08].

I.2.6.2.2 DSR

Le protocole Dynamic Source Routing (DSR) a été standardisé en 2007 [JOHN 07]. Son fonctionnement est très proche du protocole AODV à la grande différence qu'il fournit dans les paquets de données l'ensemble des nœuds permettant d'atteindre une destination (routage par la source). Cet ajout dans les paquets de données accroît le surcoût et consomme un peu plus de bande passante. A contrario, ces informations lui permettent de gérer l'asymétrie des liens présents dans le réseau. En effet, un paquet de données peut prendre une route différente de son acquittement. Le fonctionnement basique de DSR s'avère assez simple à mettre en œuvre. Il met en place uniquement deux phases : la phase de découverte des routes, et la phase de maintenance de ces mêmes chemins.

Le fonctionnement de la découverte des routes est le suivant. Un nœud source initie une requête de découverte des routes (Route Request) lorsqu'un paquet de la couche supérieure lui provient et qu'il ne possède pas de route vers sa destination. Le nœud source avant de transmettre la requête de route ajoute son adresse dans le champ route du paquet ainsi qu'un identifiant, l'adresse source et l'adresse de destination. Lorsqu'un nœud intermédiaire reçoit une requête de route, il vérifie tout d'abord s'il a déjà reçu la requête. Pour cela, il utilise les champs adresse source, adresse destination et identifiant qui permettent d'identifier de manière unique une requête de route. Si une telle requête a déjà été reçue, elle est supprimée. Dans le cas où la requête lui est destinée, il l'acquitte en envoyant une requête de réponse (Route Reply) confirmant le chemin « source-destination », sinon il la propage en ajoutant, dans le champ chemin, son identifiant.

Le protocole DSR prend en compte les liens unidirectionnels. Par conséquent, le chemin « destination-source » peut être différent du chemin « source-destination ». A la réception d'une requête de découverte des routes, le nœud de destination vérifie s'il possède déjà une route en direction de la source. S'il en connaît une, il transmet la réponse sur cette route.

Dans le cas contraire, il doit en déterminer une. Pour cela, il réutilise le fonctionnement de la découverte des routes énoncé plus haut. A la seule différence qu'il intègre le paquet de réponse (contenant la route entre la source et la destination) à sa propre requête de route. Une fois que la source reçoit la requête de route, elle extrait le chemin pour joindre la destination et l'ajoute dans sa table de routage. Elle envoie un paquet de réponse à la destination sur ce chemin, confirmant le chemin « destination-source ».

L'opération de maintenance consiste dans un premier temps à déterminer si un lien est rompu. Cette opération peut être réalisée par la sous-couche MAC. Si au bout d'un certain nombre d'émissions aucun acquittement n'est reçu, le lien peut être considéré comme coupé. Un nœud détectant la rupture prévient l'ensemble des sources avec un paquet d'erreur (Route Error). A la réception d'un tel paquet, les sources déterminent une nouvelle route si aucune autre n'est connue [DAVI 08].

I.2.6.3 Protocoles Hybrides

Dans un souci de préserver la bande passante, les protocoles de routage hybrides combinent les avantages des protocoles proactifs et réactifs. Lorsqu'il faut traverser un grand nombre de nœuds, les protocoles réactifs deviennent plus intéressants au niveau de la consommation en bande passante. Excepté la latence qui augmente, ce type de protocoles fournit de nombreux avantages pour les topologies avec un nombre élevé de nœuds. En effet, l'entretien des routes est beaucoup plus facile, car seulement les routes utilisées ont besoin d'être mises à jour lors d'une modification de la topologie.

Les protocoles proactifs sont plus performants dans des réseaux ayant un faible nombre de nœuds. En effet, ils connaissent à tout moment au moins une topologie partielle du réseau, et donc peuvent déterminer immédiatement le prochain nœud en direction de la destination. Aucune latence au niveau de l'émetteur ne se fait donc ressentir. La consommation de bande passante est dans ce cas relativement minime car peu de stations sont présentes dans le réseau.

Les protocoles hybrides vont donc tirer avantage de ces deux protocoles. Un nœud va utiliser, dans son proche entourage, un algorithme de routage proactif. Ainsi, chaque nœud a une connaissance globale de son voisinage. Puis à l'extérieur de son entourage immédiat, il va utiliser un algorithme de routage réactif. Ce type d'algorithme s'inspire du comportement humain, c'est-à-dire que nous avons une bonne connaissance du quartier où

l'on habite, mais plus on s'en éloigne, plus on ne connaît que les axes pour atteindre notre lieu de destination, et pas ce qui l'entoure.

Nous allons présenter le protocole hybride ZRP qui est le protocole hybride le plus référencé dans la littérature [DAVI 08].

1.2.6.3.1 ZRP

Le protocole de routage hybride le plus répandu est le protocole Zone Routing Protocol (ZRP) [HASS 99]. Ce protocole découpe la topologie du réseau en deux zones. La première zone est celle dans le voisinage de chaque nœud, elle est appelée Intrazone. En fait, c'est l'ensemble des nœuds qui se trouvent à un nombre de sauts inférieur ou égal à H_{max} . La seconde zone est la zone extérieure à un nœud, appelée Interzone, c'est-à-dire l'ensemble des nœuds qui se trouvent à un nombre de sauts supérieur à H_{max} .

Pour déterminer le chemin pour joindre une destination, deux protocoles de routage vont être employés suivant la zone dans laquelle se trouve la destination. Ainsi, si la destination se situe dans l'Intrazone, le protocole de routage proactif Intrazone Routing Protocol (IARP) est utilisé. Si la destination est extérieure à cette zone, le protocole de routage réactif Interzone Routing Protocol (IERP) est employé.

Le protocole de routage IARP est basé sur un protocole à état de liens. Chaque nœud diffuse, périodiquement, sa connaissance de ses voisins. A l'aide des informations diffusées, les nœuds construisent la topologie et déterminent les routes vers les nœuds situés à proximité. Pour éviter que la diffusion des paquets de contrôle se propage sur la totalité du réseau, la source met le champ TTL à la valeur de H_{max} , le nombre de saut maximum auquel se limite l'Intrazone. Chaque fois qu'un nœud reçoit un tel paquet, il met à jour sa table de routage puis décrémente de 1 le champ TTL du paquet. Si ce champ est égal à 0 le paquet est supprimé sinon il est propagé.

Lorsque le nœud source ne connaît pas de chemin vers la destination, c'est qu'elle ne se trouve pas dans l'Intrazone. Il utilise le protocole IERP pour déterminer un chemin jusqu'à elle. Le protocole IERP est responsable uniquement des communications entre les différentes zones. La source détermine un ensemble de nœuds frontières à son Intrazone. Elle utilise ces nœuds pour déterminer un chemin jusqu'à la destination, tout en réduisant le délai et le surcoût pris par la recherche. Lors de la réception de la requête de demande de

création de route, les nœuds frontières ajoutent leur identifiant dans l'entête de la requête. Ensuite, deux procédures sont appliquées selon que ces nœuds connaissent une route vers la destination ou pas [DAVI 08] :

- La destination est dans l'Intrazone d'un nœud frontière : une réponse est envoyée à la destination en prenant le chemin inverse contenu dans l'entête de la requête.
- La destination ne se situe pas dans l'Intrazone d'un nœud frontière : la requête est propagée à l'ensemble de ses nœuds frontières et l'opération recommence jusqu'à déterminer un chemin.

I.3 Architectures Cross-Layer pour les réseaux sans fil

La conception Cross-Layer est un sujet de recherche très intéressant qui exploite activement la dépendance entre les différentes couches de protocole pour obtenir des gains de performance. Dans cette partie nous allons présenter ce concept ses types ainsi que son principe de fonctionnement dans les réseaux sans-fil.

I.3.1 Le concept Cross-Layer

La multiplication des mécanismes de QoS (Quality of Service) et des techniques d'adaptation sur les différentes couches du modèle TCP/IP a engendré plusieurs problématiques causées principalement par l'isolation des couches. Ces problématiques peuvent être regroupées en trois grandes classes listées ci-dessous :

- *La redondance* : Elle est causée par la duplication d'un mécanisme sur plusieurs couches. Par exemple, la retransmission dans la couche transport en utilisant TCP et la retransmission dans la couche MAC 802.11.
- *L'annulation* : Les avantages introduits par certains mécanismes de QoS sur les couches supérieures ne sont pas respectés par des couches inférieures ce qui provoque leur annulation. Par exemple, la priorité introduite au niveau IP grâce aux classes de service n'est pas forcément assurée au niveau de la couche MAC 802.11.
- *La contradiction* : Dans certains cas extrêmes, l'effet de deux mécanismes présents sur deux couches distinctes est contradictoire. Par exemple, l'utilisation d'UDP au niveau transport pour éviter les retransmissions et l'utilisation de la retransmission au niveau de la couche MAC 802.11.

Afin d'apporter une solution à toutes ces problématiques et d'optimiser les performances des systèmes communicants, nous avons assisté ces dernières années à l'émergence d'un nouveau concept sous l'appellation de Cross-Layer. Ce dernier autorise la violation de la structure protocolaire en couches dans le but d'améliorer les performances de transmission dans les réseaux sans fil et d'assurer une meilleure QoS.

Comme tous les nouveaux concepts, il est très difficile de trouver ou de proposer une définition exacte pour le Cross-layer. Même au niveau de la terminologie, nous trouvons dans la littérature plusieurs variantes : la conception Cross-layer, l'adaptation Cross-layer, l'optimisation Cross-Layer, le retour d'information Cross-Layer. Dans [SRIV 05] [SCHA

05], les auteurs ont proposé une définition générique qui peut englober toutes les techniques et tous les mécanismes Cross-Layer qui existent actuellement. Ainsi, le design Cross-Layer est défini comme suit : « la conception d'un protocole en violation avec l'architecture en couches de référence est une conception Cross-Layer à l'égard de cette architecture ». Le terme violation englobe :

- La définition de nouvelles interfaces entre les couches.
- La redéfinition des limites des couches.
- La conception d'un protocole sur une couche en se basant sur la conception d'un autre protocole sur une autre couche.
- La configuration commune des paramètres à travers les couches.

Le concept Cross-Layer permet la définition de protocoles ou de mécanismes qui ne respectent pas l'isolation des couches du modèle OSI (respectivement TCP/IP). Ainsi, il autorise la communication entre deux, ou plusieurs, couches adjacentes, ou non adjacentes, dans le but d'améliorer les performances globales du système. Ceci peut être réalisé par la définition de nouvelles interfaces au niveau des couches qui permettent de récupérer leurs paramètres de performance mais aussi de configurer certains paramètres d'une manière dynamique. Ces paramètres peuvent être utilisés par les protocoles et/ou les mécanismes d'adaptation pour améliorer la performance globale de la communication en se basant sur des politiques d'adaptation [ISMA 08].

I.3.2 La communication dans les architectures Cross-layer

Le principe de base du concept Cross-layer est de permettre l'échange d'informations entre les couches adjacentes et non adjacentes afin d'améliorer les performances de transmission. Cet échange d'informations peut être mis en œuvre suivant différents schémas. Parmi toutes les architectures Cross-layer proposées dans la littérature, deux modèles de communication peuvent être distingués [SRIV 05] [NASS 06]: La communication directe entre les couches et une base de données partagée entre les couches. Nous présentons ci-dessous une description de ces deux modèles représentés dans la Figure 5 [ISMA 08].

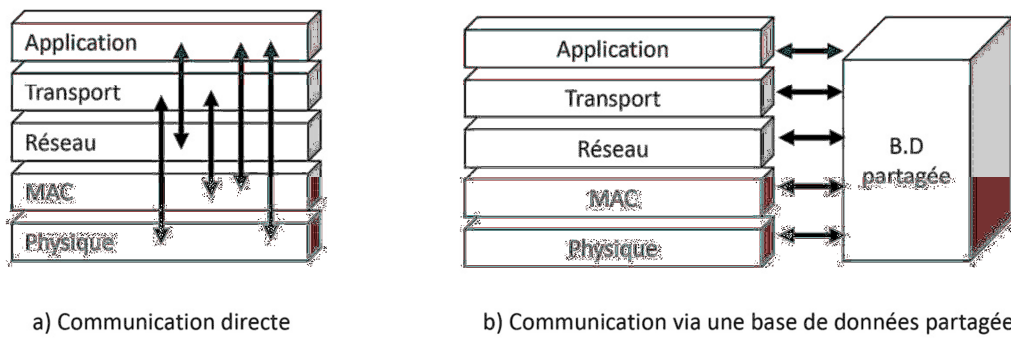


Figure 5. Les modèles de communication Cross-layer.

I.3.2.1 Communication directe entre les couches

La communication directe entre les couches est le modèle le plus utilisé par les architectures Cross-layer. Il permet à une couche d'accéder directement aux paramètres et aux variables d'une autre couche sans passer par un intermédiaire. Cette communication peut être « in-band » en utilisant les en-têtes des protocoles qui sont déployés actuellement, par exemple, la couche IP accède aux champs de l'en-tête TCP « ECN » pour indiquer une congestion dans le réseau. Dans certain cas, des extensions d'en-tête sont nécessaires pour faire passer des informations supplémentaires. La communication directe peut être « out-of-band » en utilisant un nouveau protocole dédié. Le protocole de signalisation CLASS (Cross-Layer Signaling Shortcuts), présenté dans [WANG 03], est un parfait exemple de ce type de protocole. La communication « out-of-band » peut s'effectuer aussi en définissant de nouvelles interfaces (API : Application Programming Interface) au niveau d'une couche qui seront utilisées directement par d'autres couches pour récupérer et configurer des paramètres de fonctionnement. Un exemple de cette API est proposé dans [RAIS 06].

I.3.2.2 Communication via une base de données partagée

Plusieurs architectures Cross-layer [WINT 06] [KHAN 06] proposent l'utilisation d'une base de données partagée afin de stocker et de récupérer des paramètres. Cette base est accessible par toutes les couches qui peuvent, ainsi, s'informer de l'état des autres couches ou récupérer des paramètres de configuration nécessaire à leur fonctionnement interne. La base de données est aussi accessible par un système d'optimisation responsable d'initialiser les paramètres avec les valeurs adéquates. La Figure 5 (b) illustre ce type de communication, la base de données est considérée comme une nouvelle couche en

parallèle à toutes les autres. Cependant, pour mettre en œuvre cette base de données, il faut répondre à plusieurs contraintes conceptuelles, comme la localisation de la base de données (sur les nœuds communicants ou sur un nœud indépendant), le type de communication entre les couches et la base de données partagée, ainsi que le protocole de communication utilisé [ISMA 08].

I.3.3 Les approches Cross-layer dans les réseaux sans fil

Dans la littérature, plusieurs techniques Cross-layer ont été proposées pour améliorer les performances des transmissions sans fil. Au début, ces mécanismes étaient limités à l'interaction entre la couche physique et la couche liaison de données. De plus, les mécanismes proposés étaient indépendants et visaient l'amélioration d'une imperfection précise. Par la suite, nous avons assisté à l'apparition de plusieurs travaux proposant des interactions avec les couches supérieures et aussi à l'apparition des architectures Cross-layer qui tentent de faire collaborer plusieurs couches, prenant en charge plusieurs paramètres, pour une optimisation globale [ISMA 08].

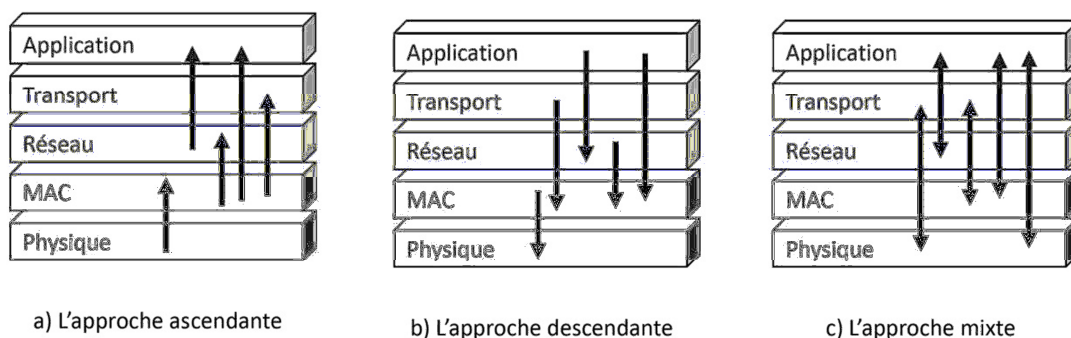


Figure 6. Les approches du Cross-layer.

Tous les travaux se basant sur Cross-Layer sont classés en trois grandes approches identifiées dans [SRIV 05] [SCHA 05] et représentées dans la Figure 6.

- *L'approche ascendante (Bottom-up) :* Les couches supérieures optimisent leurs mécanismes en fonctions des paramètres (conditions) des couches inférieures.
- *L'approche descendante (Top-down) :* Les couches supérieures décident des paramètres de configuration des couches inférieures. Les couches inférieures peuvent aussi considérer certaines spécificités du niveau applicatif pour exécuter leurs traitements.

- *L'approche mixte (Integrated)*: Cette approche exploite les deux approches précédentes dans une même architecture afin de trouver la meilleure configuration inter-couches pour un fonctionnement optimal du système.

I.3.3.1 Les projets traitant la problématique Cross-layer

Plusieurs projets financés par la commission Européenne ont traité, ou traitent, la problématique du Cross-layer en étudiant ce nouveau concept et en proposant de nouvelles interactions Cross-layer afin d'améliorer les performances des transmissions. Nous présentons, ci-dessous, une brève description de quelques projets [ISMA08].

I.3.3.1.1 Le projet 4MORE

Ce projet [4MORE] s'inscrit dans le développement de systèmes 4G, la nouvelle génération des communications mobiles. La vision européenne de la 4G est un nouveau système basé sur IP et offrant tous les services, à n'importe quel moment en utilisant n'importe quel terminal. Pour cela, la 4G doit offrir des débits variant entre 2 et 100 Mbps couvrant plusieurs environnements mobiles (véhicules, piétons) et fixes (intérieures et extérieures) dans une plage de fréquences de 50 – 100 MHz. La technologie de transmission utilisée pour atteindre cet objectif est le MC-CDMA (MultiCarrier-CDMA) qui est déjà adopté au Japon. L'objectif du projet 4MORE est d'étudier, développer, intégrer et valider un système sur puce, rentable et de faible consommation, pour les terminaux mobiles utilisant un système de transmission multi-antennes basé sur MC-CDMA. Ce système doit intégrer de nouvelles interactions entre la couche physique (MC-CDMA) et les algorithmes de la couche MAC.

I.3.3.1.2 Le projet PHOENIX

L'objectif de ce projet [PHON] est le développement d'un lien de communication de bout-en-bout basé sur un réseau sans fil optimisé. Pour cela, le projet propose une communication entre le monde applicatif (le codage source, le cryptage, etc.) et le monde des transmissions (le codage canal, la modulation) à travers le monde réseau (le protocole IPv6). Pour atteindre cet objectif, le projet se focalise sur trois axes principaux. Le premier axe développe un schéma innovant pour une optimisation conjointe du codage source et canal (JSCC : Joint Source Channel Coding). Ce premier axe inclut la définition de nouveaux codages et l'adaptation des codages qui existent actuellement. Le deuxième axe définit de nouvelles stratégies d'adaptation efficaces qui prennent en considération des

paramètres réels (présence de cryptage, le nombre de saut sans fil, etc.). Enfin, le troisième axe définit une nouvelle architecture réseau pour les futurs systèmes de communication sans fil. Cette nouvelle architecture se base sur les optimisations étudiées dans les autres axes.

I.3.3.1.3 Le projet ENTHRONE II

Le projet ENTHRONE II [ENTH] propose une solution globale pour gérer la chaîne de distribution des services audio/vidéo. Cette solution englobe la protection du contenu, la distribution à travers les NGNs et la réception au niveau de l'utilisateur final. Le but de ce projet n'est pas d'unifier ou d'imposer une stratégie mais plutôt d'harmoniser les différentes fonctionnalités afin de supporter une QoS de bout-en-bout à travers des réseaux hétérogènes distribuant des services IP multimédia pour divers utilisateurs. Pour atteindre cet objectif, le projet se base une architecture de gestion ouverte et décentralisée pour la distribution de bout-en-bout. Le modèle MPEG-21 est utilisé comme un support commun pour l'implémentation et la gestion des fonctionnalités de l'architecture. L'adaptation MPEG-21 Cross-layer, qui a pour objectif d'adapter les contenus multimédia pour maintenir la QoS, est considérée comme un élément principal de l'architecture ENTHRONE II. Cette adaptation prend en considération plusieurs paramètres allant de la couche application (qualité objective du flux multimédia) jusqu'à la couche physique (débit physique dans les réseaux sans fil, niveau du signal) en passant par la couche réseau (latence, taux de perte).

I.3.4 Discussion

Le concept du Cross-layer est considéré comme une amélioration de la structure en couche qui existe actuellement. Cette amélioration n'est pas gratuite, elle possède un coût en terme de complexité de conception. En effet, les nouveaux protocoles et les nouvelles applications qui exploiteront ce nouveau paradigme seront plus difficiles à concevoir puisqu'ils doivent prendre en considération plusieurs paramètres présents sur différentes couches. Cette complexité est accentuée lorsque les mécanismes Cross-layer sont conçus d'une manière distribuée sans aucun contrôle centralisé [ISMA 08]. Dans notre travail nous avons adopté l'approche mixte qui est plus attractive puisqu'elle englobe, en quelque sorte, les deux autres approches (ascendantes et descendantes) dans une seule architecture.

I.4 Service Web et Web sémantique

Cette section est consacrée à la présentation de la technologie des Services web ainsi qu'aux principes de base du web sémantique.

I.4.1 Les services Web

I.4.1.1 Définition

A l'origine, la technologie des services web a été initiée par IBM et Microsoft, puis en partie normalisée sous le W3C (World Wide Web Consortium), l'organisme chargé de standardiser les évolutions du web. Les services Web sont des composants logiciels encapsulant des fonctionnalités métier de l'entreprise. Ils peuvent être accessibles depuis une autre application (un client, un serveur ou un autre service Web) à travers le réseau Internet en utilisant les protocoles standard du Web.

Techniquement, Jusqu'ici l'accès via Internet à une ressource applicative ou à une base de données s'effectuait par l'envoi d'une requête s'appuyant sur des langages de script (PHP, JSP, ...). Il s'agissait donc d'un dialogue entre une couche de présentation reposant sur HTML (protocole HTTP) et des applications installées sur un serveur distant. Avec les services Web, un dialogue est désormais instauré entre applications qui peuvent être installées sur des machines distantes. En effet, afin de dialoguer via Internet, ces applications doivent parler le même langage, langage basé sur le XML.

Un service Web est décrit dans un document WSDL (Web Services Description Language), précisant les méthodes pouvant être invoquées, leur signature et les points d'accès du service (URL, Port., etc.). Ces méthodes sont accessibles via le protocole SOAP (Simple Object Access Protocol). Un service Web est une technologie permettant à des applications de dialoguer à distance via Internet par le mécanisme requête/réponse, qui sont des messages XML transportés par HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol) ou SMTP (Simple Mail Transfer Protocol). Les services Web sont centralisés (leurs publications) dans un référentiel commun UDDI (Universal Description Discovery and Integration) afin de faciliter leurs recherches [ALEX 02].

Comme le montre la Figure 7, les services Web présentent au réseau une sorte d'interface avec un arrière-plan des logiciels, tels que les systèmes de gestion de bases de

données, .NET, J2EE (Java2 Platform, Enterprise Edition), ou CORBA (commune Object Request Broker Architecture), des objets, des adaptateurs d'Enterprise Resource Planning (ERP) et d'autres. L'interface de service Web reçoit un message XML de l'environnement de réseau, ensuite elle transforme les données XML en un format compréhensible par les logiciels et éventuellement retourne un message de réponse. Le logiciel sous-jacent un service Web peut être créé en utilisant n'importe quel langage de programmation, système d'exploitation ou système de middleware [ERIC 02].

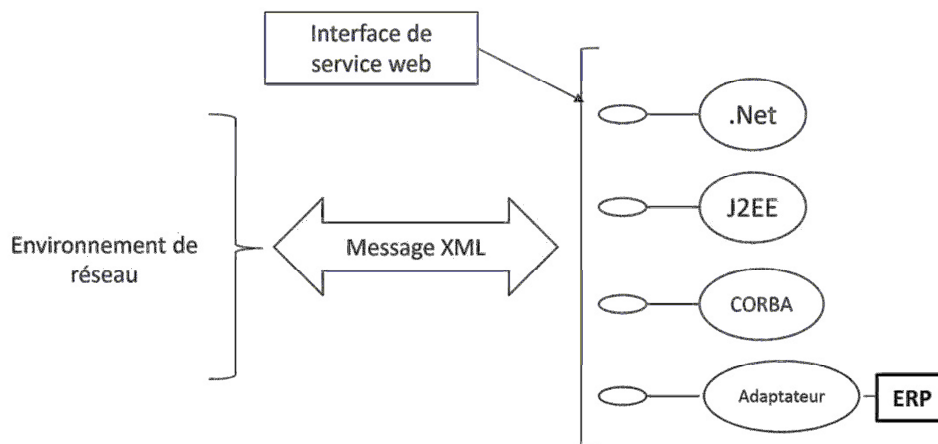


Figure 7. Les protocoles des services web.

I.4.1.2 L'architecture orientée service

Les architectures en informatique relèvent plus de la philosophie que de la spécification. Elles proposent une vision sur le fonctionnement et la dynamique des systèmes.

L'architecture orientée service (SOA), est un modèle abstrait qui définit un système par un ensemble de composants logiciels distribués qui fonctionnent afin de réaliser une fonctionnalité globale préalablement établie [HEAT 01].

Dans une SOA, les services peuvent communiquer entre eux. Cette communication peut soit consister en un simple passage de données ou impliquer la coordination de deux ou plusieurs composants pour l'accomplissement d'une activité. Le choix d'une architecture SOA entre dans la perspective de transformer le Web en une énorme plate-forme à composants (services) faiblement couplés et automatiquement intégrables.

I.4.1.3 Architecture des services Web

L'architecture des services Web Figure 8 est une instance d'architecture orientée service SOA. La définition d'architecture pour les services Web (WSA : Web Services Architecture), consiste à mettre en évidence les concepts, les relations entre ces concepts ainsi qu'un ensemble de contraintes qui assurent l'objectif premier des services Web à savoir l'interopérabilité. Ces éléments sont structurés à travers un modèle de déploiement et de fonctionnement des services Web. Les concepts de l'architecture ne sont pas supposés avoir une existence dans la réalisation informatique, ils désignent des personnes, des organisations ou encore une entité logicielle ou matérielle. La définition ou la description des concepts de l'architecture se présente sous forme d'une identification nominale du concept ainsi que les relations qu'il entretient avec les autres concepts. Les principaux concepts intervenant dans l'architecture des services Web sont :

- *Le fournisseur du service* : d'un point de vue conceptuel, il désigne la personne ou l'organisation responsable juridiquement de l'agent logiciel (i.e. le service). D'un point de vue opérationnel, les fournisseurs de services peuvent également désigner le serveur qui héberge les services déployés.
- *Le client du service* : comme pour le fournisseur, il représente une personne ou une organisation, client potentiel des services. D'un point de vue opérationnel il désigne, l'application cliente qui invoque le service.
- *L'annuaire des services* : il représente la personne ou l'organisation responsable de l'hébergement ou la publication des services. D'un point de vue fonctionnel, il représente l'entité logicielle qui joue le rôle de l'intermédiaire entre les clients et les fournisseurs de services. Le concept registre ou annuaire de service est essentiel dans l'architecture services Web.

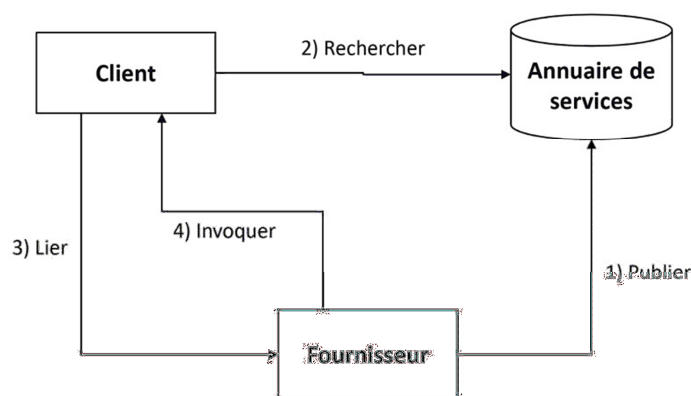


Figure 8. Architecture des services web.

I.4.1.4 Technologies de base des services Web

Pour garantir l'interopérabilité des trois concepts précédents (Fournisseurs de services, Client du service, Annuaire de publication), des propositions de standards (technologie pour les services Web) ont été élaborées : XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) et UDDI (Universal Description, Discovery and Integration).

XML, est aujourd'hui un standard qui permet de décrire des documents structurés transportables sur les protocoles communs d'Internet. XML constitue la technologie de base des architectures services web. En effet, il apporte à l'architecture l'extensibilité et la neutralité vis à vis des plates-formes et des langages de développement. De plus, grâce à la structuration, XML permet la distinction entre les données des applications et les données des protocoles permettant ainsi une correspondance facile entre les différents protocoles. L'interopérabilité entre les systèmes hétérogènes demande des mécanismes puissants de correspondance et de gestion des types de données des messages entre les fournisseurs et les clients. C'est une tâche où les schémas de type de données XML s'avèrent bien adaptés.

SOAP, c'est un protocole de transport réseaux, soumis à un format de message XML et permettant à un client d'invoquer un service distant.

WSDL, c'est un langage de description basé sur XML et utilisé pour décrire l'interface d'utilisation d'un service Web. Le fournisseur de service utilise un document WSDL afin d'indiquer les opérations fournies par un service Web ainsi que les paramètres et les types de données associés à ces opérations.

UDDI, c'est un protocole d'annuaire basé sur XML qui permet de stocker ou de rechercher des informations sur des services Web. Il permet aussi d'automatiser la découverte des services Web.

I.4.1.5 Discussion

Malgré les bénéfices et avantages apportés par l'utilisation des services Web, cette technologie reste encore limitée. Les services Web ne sont qu'un mécanisme de transfert de données d'un système à l'autre, ils n'apportent, en aucun cas, plus de valeurs à l'information

déjà possédée. Ils permettent juste une meilleure diffusion auprès des clients et des fournisseurs.

L'utilisation d'XML dans l'architecture des services Web, permet de structurer la construction d'un document. Il ne permet pas de spécifier le sens à donner au document. Concernant la découverte des services Web dans le registre UDDI, ce dernier permet une recherche sous différentes entrées au registre. Les services peuvent être recherchés par exemple par nom, par emplacement ou par tModels. Cependant, cette recherche est restreinte à la détection de mots-clés et ne supporte aucune inférence des informations présentes dans l'UDDI. Ce problème est causé par l'absence de la sémantique dans les technologies de base des services Web. Pour le résoudre, les recherches faites sur le Web sémantique ont été exploitées dans le domaine des services Web.

I.4.2 Le Web sémantique

I.4.2.1 Naissance de Web sémantique

Tim Berners Lee, le créateur du World Wide Web, a proclamé que le Web sémantique est la prochaine évolution du Web. Les informations présentes sur le Web ont été initialement des documents enregistrés pour être consultés par des utilisateurs humains. Mais, les nouveaux systèmes conçus et développés pour fonctionner sur le Web tendent à décharger l'utilisateur de quelques, voire de beaucoup de tâches en déléguant celles-ci à des agents logiciels. Le Web Sémantique est une vision du futur Web dans lequel l'information est menée un sens explicite facilitant ainsi aux machines le traitement et l'intégration des informations sur le Web.

Les recherches actuelles sur le Web Sémantique proposent de s'appuyer sur des techniques de représentation de connaissances (formalisme et raisonnement) pour munir l'information contenue dans les ressources Web d'une sémantique.

I.4.2.2 Architecture du Web sémantique

Le Web sémantique nécessite une architecture partagée, pour échanger des ressources sur Internet. Il se compose de trois niveaux importants Figure 9 : Le niveau d'adressage, le niveau syntaxique et le niveau sémantique.

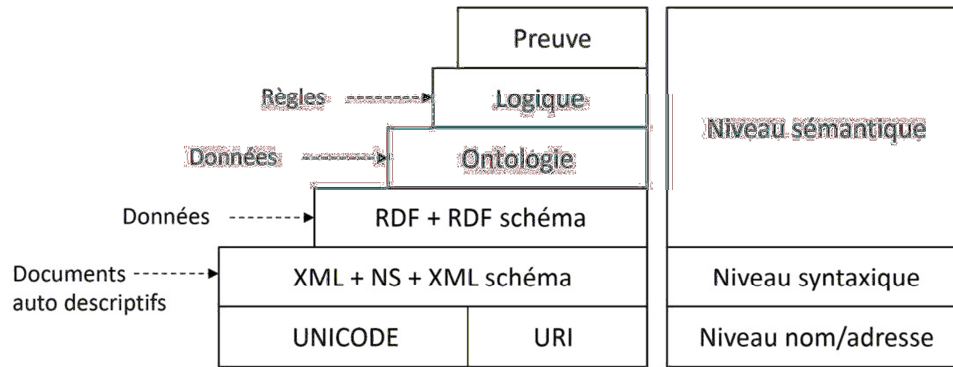


Figure 9. Les niveaux du Web sémantique.

URI: le concept d'URI (Universal Resource Identifier) regroupe toutes les formes syntaxiques permettant de désigner une ressource Internet. L'URI la plus courante est l'URL (Uniform Resource Locator). Une URL est une chaîne de caractère indiquant l'emplacement où se trouve une ressource.

XML: fournit une surface syntaxique pour les documents structurés mais ne fournit aucune contrainte sémantique sur le sens de ces documents.

XML Schéma: c'est un langage pour restreindre la structure des documents XML et aussi étendre XML avec des types de données.

RDF: RDF (Resource Description Framework), est un modèle de données pour les objets ("ressources") et les relations entre eux, il fournit une sémantique simple pour ce modèle de données qui peuvent être représentés en XML.

RDF Schéma: c'est un vocabulaire pour décrire les propriétés et les classes des ressources RDF.

OWL: (Ontologie Web Language) ajoute plus de vocabulaire pour décrire les propriétés et les classes entre autres, les relations entre les classes, cardinalité, égalité, typage de propriétés plus riche, caractéristiques des propriétés et les hiérarchies des propriétés et des classes.

I.4.2.3 Principe du Web sémantique

La vision du Web sémantique s'appuie sur l'utilisation des annotations et les méta-données ainsi et surtout les ontologies :

- Les Annotations et les métadonnées sont, des informations associées à une ressource du web, permettant son exploitation par des agents logiciel.
- Les métadonnées, sont des données sur des données, ils sont aussi des données et peuvent donc être structurées afin de décrire une ressource quelconque. Par contre les annotations représentent des notes critiques ou explicatives accompagnant un texte [YANI 03]. Un des grands principes du Web sémantique est qu'il est nécessaire d'associer aux ressources du Web (sources d'informations et services) des informations exploitable par des agents logiciels afin de favoriser l'exploitation de ces ressources. Pour pouvoir exprimer les métadonnées et les annotations avec les informations sur le Web, beaucoup de langages ont été élaborés comme RDF et RDFS.
- Les ontologies : Pour être susceptibles d'être exploitées automatiquement, les annotations et métadonnées doivent être entièrement explicites, c'est-à-dire suivre un modèle et être exprimées dans un vocabulaire clairement et formellement définis. Les ontologies, deuxième pilier du Web sémantique, sont le réceptacle de ces définitions. l'ontologie a été abordée pour la première fois par *John McCarthy* en 1980 [PSYC 04], puis, *Gruber* [GRUB 93] a proposé sa première définition sur l'ontologie : "*une ontologie est une spécification d'une conceptualisation*". Il considère que toute représentation de connaissance est basée sur une conceptualisation. La conceptualisation représente la collection des objets, de concepts et des autres entités qui sont supposés exister dans un certain domaine d'intérêt ainsi que les relations qui les relie. Une représentation formelle de cette conceptualisation s'appelle une ontologie. Une ontologie possède une syntaxe qui est celle de langage utilisé par la machine, et une sémantique qui est celle de domaine de connaissance. Dans le paradigme du web sémantique le langage OWL¹ est utilisé pour l'implémentation des ontologies.

¹ OWL (Ontologie Web Language), est un langage ontologique du Web qui est une extension de RDFS pour définir et instancier les ontologies du Web. OWL a été conçu pour être utilisé par les applications qui traitent le contenu de l'information au lieu de la présenter seulement aux êtres humains. OWL facilite grandement l'interopérabilité au niveau machine du contenu du Web plus que ce qui est déjà supporté par les langages XML, RDF et RDFS en fournissant du vocabulaire supplémentaire avec des sémantiques formelles. OWL possède des sous-langages de plus en plus expressifs *OWL Lite*, *OWL DL* et *OWL Full*.

I.4.3 Les services web sémantiques

I.4.3.1 Présentation

Les services Web sémantiques se situent à l'intersection de deux domaines de recherche importants qui concernent les technologies de l'Internet : le Web sémantique et les services Web. Le Web sémantique s'intéresse principalement aux informations statiques disponibles sur Internet (pages Web) et à la manière de les décrire d'une façon compréhensible par les machines. D'autre part, les standards permettant la description des services Web tels que WSDL et UDDI qui utilisent des structures XML pour la description de la fonctionnalité des services et des données qu'ils échangent. De telles structures sont syntaxiques et engendrent une forme d'ambiguïté. Ainsi la même structure XML peut être interprétée différemment d'un utilisateur à un autre. Cette contrainte empêche l'utilisation des services Web par des agents logiciels. Le même problème s'était posé quelques années plus tôt pour les pages Web, et avait engendré l'apparition des solutions pour l'annotation sémantique des pages Web. De nombreux travaux ont tentés d'adapter ces solutions pour l'annotation sémantique des services Web.

Un service Web sémantique est un service Web décrit en utilisant des annotations sémantiques dans un langage bien défini (exemple : les ontologies) qui permettent au service d'avoir une interface interprétable sans ambiguïté, facilitant l'automatisation de certaines tâches telles que leurs découverte, sélection, invocation et leurs composition.

- *Découverte automatique de services Web* : actuellement cette tâche doit être réalisée par un utilisateur humain qui doit utiliser un moteur de recherche ou un annuaire pour trouver le service, lire la page Web qui décrit l'utilité et l'utilisation du service, puis l'exécuter manuellement pour vérifier que celui-ci correspond bien aux attentes de l'utilisateur.
- *Invocation automatique de services Web* : l'invocation automatique d'un service signifie l'exécution du service par un programme informatique ou un agent logiciel. Cet agent doit être capable d'interpréter les descriptions de services afin de délivrer les données nécessaires à l'exécution du service Web.
- *Composition automatique de services Web* : l'objectif que l'utilisateur veut atteindre nécessite souvent l'utilisation de plusieurs services Web. L'agent logiciel chargé d'atteindre cet objectif doit disposer de suffisamment de données afin de pouvoir sélectionner, composer et interopérer automatiquement ces services web.

I.4.3.2 Langages de modélisation des services Web

Plusieurs langages de description des ontologies pour les services Web sémantique ont été élaborés, les plus connus sont OWL-S [OWLS 04] (que nous allons exploiter pour l'adaptation de notre approche), et le langage WSMO [WSMO 05].

OWL-S : est un langage de description des ontologies pour les services web. Il est basé sur le langage OWL. Le langage OWL-S est un langage basé sur XML, il a pour objectif d'ajouter des descriptions sémantiques aux services Web (en plus de leur description syntaxique WSDL). Il a été conçu pour faciliter l'automatisation des tâches relatives aux Web Services et plus particulièrement la découverte automatique des Web Services, leurs exécutions, ou bien encore leurs compositions et leurs interopérabilités.

I.4.3.3 Approches existantes de découverte sémantique des services web

Différentes approches ont été proposées pour la résolution du problème de découverte sémantique des services web, toutes ces approches implantent en fait une découverte approximative car il n'est pas réaliste d'imaginer qu'il y a toujours un service qui correspond exactement aux besoins spécifiés, ces approches diffèrent par le langage de description de service utilisé (DAML-S, OWL-S, logique de description...etc.) et/ou par les algorithmes de découverte utilisés comme (l'algorithme de *Paolucci* [PAOL 02][SYCA 02][SYCA 03][PAOL 04], le test de subsumption dans [GONZ 01] [TRAS 01], l'utilisation des filtres [SYCA 98], la logique de réécriture [BENA 02]), et/ou par la méthode utilisée (ajouter une couche sémantique au-dessus de WSDL et UDDI comme le travail de [VERM 03]). On peut dire que, bien que le principe proposé par *Paolucci* et al dans [PAOL 02], se limite à l'utilisation des attributs fonctionnels inputs/outputs pour l'adaptation de leur système de découverte sémantique des services web dans un environnement avec infrastructure, elle reste l'approche la plus simple et efficace pour résoudre ce problème, vu que les auteurs se basent sur le langage de modélisation DAML-S qui offre une description plus riche, simple et détaillée des services web et aussi le fait que les auteurs ont intégré leur solution au-dessus des standards connus tel que UDDI et WSDL, ceci nous mène à exploiter le principe proposé par *Paolucci*.

I.4.3.4 Approche de Paolucci

I.4.3.4.1 Présentation

Dans cette approche [PAOL 02], les informations considérées dans la description des services Web pour la publication sont les entrées/sorties (inputs/outputs) offertes par le service. La requête est également spécifiée sous forme d'un ensemble des inputs/outputs requises. Dans l'algorithme proposé par les auteurs, un service web S peut répondre à la requête R si les entrées requises par S sont équivalentes ou englobent (selon une ontologie partagée) celles offertes par R , et si les sorties offertes par S sont équivalentes ou englobent celles requises par R . Ce critère garantit que le service trouvé satisfait les besoins de la requête. Les auteurs se basent sur une description DAML-S des services Web, les entrées sorties sont ainsi définies en utilisant l'ontologie DAML+OIL publiée sur le Web, les relations de subsomption peuvent donc être facilement déduites.

I.4.3.4.2 Adaptation avec le standard UDDI

L'annuaire UDDI, est une solution qui permet aux entreprises d'enregistrer leurs services web pour que ces derniers soient recherchés et invoqués facilement. Dans UDDI, les services web peuvent être cherchés par : emplacement, par nom d'entreprise, par bindings ou par tModels ce qui implique que l'UDDI offre une recherche pauvre car elle n'est basée que sur les mots clés et quelques valeurs associées aux tModels, il n'y a pas d'inférence, c-à-d, qu'il ne peut produire aucun support pour trouver des services dans le biais de ce que ces services produit. Par exemple, un service de vente de voiture peut être publié comme "Concessionnaires de voitures neuves", mais une recherche sur UDDI pour un service "Concessionnaires d'automobiles" ne va pas identifier le service publié bien que le service "Concessionnaires de voitures neuves" est sous-type de "Concessionnaires d'automobiles" selon l'inférence sur l'ontologie globale.

Le deuxième problème avec UDDI c'est qu'il peut produire beaucoup de résultats qui n'ont pas d'intérêt. Par exemple, lors de la recherche pour "Concessionnaire automobile", peut être que le client n'est pas intéressé par les concessionnaires qui n'acceptent pas des cartes de crédit comme méthode de paiement, ceci implique que pour résoudre ce problème il faut aussi mettre en considération les inputs/outputs des services web afin de produire une recherche plus précise [SYCA 04].

L'approche de *Paolucci* résout ce problème en ajoutant une couche sémantique pour permettre une inférence sémantique, en se basant sur la découverte des capacités des services et par utilisation de DAML-S comme langage de description des capacités (inputs/outputs). L'adaptation de cette approche avec UDDI permettra aux services d'être recherchés par des mots clés UDDI et aussi par inférence sémantique.

I.4.3.4.3 Architecture de découverte combinée

La figure ci-dessous Figure 10 illustre l'architecture combinée DAML-S/UDDI, proposé par les auteurs dans [PAOL O2].

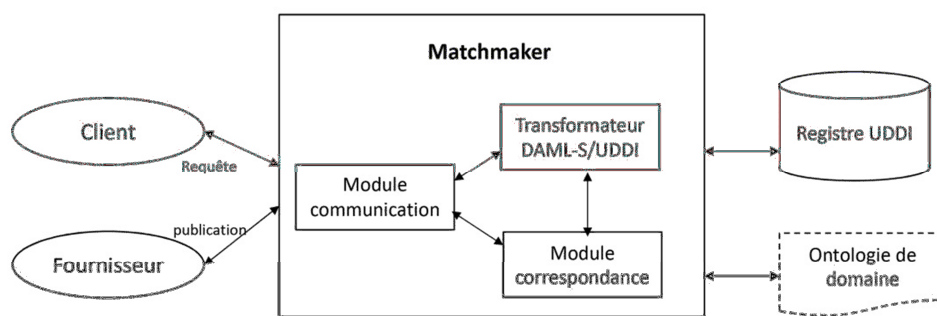


Figure 10. Approche combinée DAML-S/UDDI.

Dans l'architecture combinée DAML-S/UDDI ci-dessus, la première tâche de service web est de publier ces capacités avec le Matchmaker (le Matchmaker représente la couche ou l'algorithme injecté par les auteurs dans l'architecture classique des services web afin d'aboutir à une nouvelle architecture basée sur le raisonnement sémantique). L'enregistrement des capacités permet aux services web d'être découverts et d'agir comme fournisseur [SYCA 03].

Le Matchmaker reçoit les publications des services et les requêtes dans le format DAML-S de l'extérieur à travers le module communication après avoir reconnu que le message est une publication, le module communication l'envoie au module transformateur qui construit une description UDDI à partir de la description DAML-S, ce dernier produit une clé de référence ID de service. Cet ID combiné avec la description des capacités publiées sont envoyés au module de correspondance, qui se charge de sauvegarder les publications dans une base de données de publications.

Si une requête est envoyée au Matchmaker, le module communication l'envoie au module de correspondance qui établit une correspondance basée sur les capacités, le

résultat de cette opération, sont des publications sélectionnées et des références IDs au service enregistré dans l'UDDI.

I.4.3.4.4 Le module correspondance

le module correspondance réalise plusieurs actions, il prélève les inputs/outputs dans les descriptions sémantiques des services qu'offre le fournisseur ensuite à l'aide d'une ontologie globale qui décrit des relations ontologiques entre tous les concepts qui existent (par définition théorique des ontologies) en utilisant le langage OWL le *module correspondance* calcul et sauvegarde les degrés de correspondance entre les inputs/outputs de la requête de l'utilisateur avec ceux des services proposés par le fournisseur.

Concernant le calcul des degrés ou niveau de correspondance, il existe quatre niveaux de correspondance [PAOL 02] illustrés dans la Figure 11 ces niveaux dépendent de la relation entre les concepts associés à chaque input/output de la requête R et la publication S.

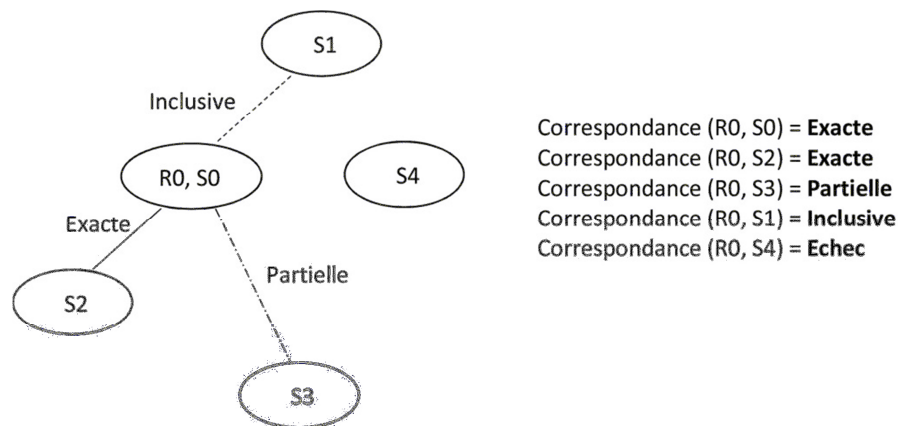


Figure 11. Niveaux de correspondances.

- *La correspondance Exacte* : Le module correspondance reconnaît une correspondance exacte entre le service S et la requête R, lorsque $outR = outS$. Un tel service répond exactement aux besoins de la requête. Aussi la correspondance peut être exacte, lorsque $outR$ est une sous-classe directe d' $outS$.
- *La correspondance Inclusive* : Le module correspondance reconnaît une correspondance inclusive (Plugin), lorsque $outS$ super-classe-de $outR$. Cette règle reconnaît qu'il y a une faible relation entre $outR$ et $outS$, dans ce cas par exemple, on peut s'attendre à ce qu'un service qui a comme output véhicule offre un certain

type de voitures, mais nous ne pouvons pas attendre qu'il offre chaque type de sa sous-classe directe.

- *La correspondance Partielle* : Dans le cas de correspondance partielle (subsumes), la requête est plus générique que le service trouvé. En d'autres termes : *outR* est une super-classe de *outS*. Dans ce cas, le service *S* peut être retenu, mais ce service peut ou ne peut pas satisfaire la requête, il se peut qu'un autre service puisse être recherché pour compléter le manque (composition des services Web).
- *Cas d'échec* : Si aucune des relations de correspondance précédentes n'a été identifiée on dit que le service *S* et la requête *R* sont incompatibles.

L'Algorithme 1 suivant représente l'algorithme qu'utilise le module correspondance pour calculer les degrés de correspondances entre les couples des concepts associés à la requête et le service.

Algorithme *DegréDeCorrespondance (outR,outS)*
si outS = outR alors retourner Exacte
si outR sous_classe_directe outS alors retourner Exacte
si outS super_classe_de outR alors retourner Inclusive
si outR super_classe_de outS alors retourner Partielle
autrement retourner Echec

Algorithme 1. Algorithme degré de correspondances [PAOL 02].

Une fois que le module correspondance a terminé la phase de calcul des correspondances existantes il commence à réaliser un tri des résultats trouvés, sachant que la correspondance *Exacte* est la préférée, en deuxième niveau vient la correspondance *Inclusive* et en troisième niveau, la correspondance *Partielle*. Le principal critère de tri est de sélectionner la correspondance avec le plus haut degré dans la correspondance des outputs. La correspondance des inputs est utilisée seulement en deuxième lieu pour séparer les correspondances des outputs avec le même degré.

L'algorithme 2 suivant représente le principe de tri des correspondances.

Algorithme *TriDeCorrespondances(match1, match2)*
si match1.output > match2.output alors
match1 > match2
si match1.output = match2.output & match1.input > match2.input alors
match1 > match2
si match1.output = match2.output & match1.input = match2.input alors
match1 = match2

Algorithme 2. Algorithme de tri des correspondances [PAOL 02].

L'Algorithme 3 suivant représente l'Algorithme de correspondance des outputs par le module correspondance.

Algorithme *CorrespondanceDesOutputs* (*outputsReq*, *outputsServ*)
DegreDeCorrespondanceGlobal = *Exacte*
pourtout *outR* **dans** *outputsReq* **faire**
 chercher *outs* **dans** *outputsServ* **tel que**
 DegreDeCorrespondance = *MaxDegreDeCorrespondance* (*outR*, *outs*)
 si (*DegreDeCorrespondance* = *Echec*) **retourner** *Echec*
 si (*DegreDeCorrespondance* < *DegreDeCorrespondanceGlobal*)
 DegreDeCorrespondanceGlobal = *DegreDeCorrespondance*
retourner *Trier* (*EnregistrerCorrespondance*)

Algorithme 3. Algorithme de correspondance des outputs [PAOL 02].

I.4.4 Découverte des services dans MANET

Les nœuds dans MANET peuvent avoir peu ou pas de connaissances sur les services offerts par chacun d'entre eux. Par conséquent, les mécanismes de découverte des services et des ressources qui permettent aux périphériques de localiser automatiquement les services de réseau et de publier leurs propres services pour le reste du réseau, sont un aspect important de réseau auto-configurable.

Il existe deux type de mécanisme de découverte de services et de ressources dans MANET, premièrement, sans annuaires, où les nœuds demandent de façon réactive des services en cas de besoin et / ou des nœuds annoncent de manière proactive leurs services à d'autres (Figure 12 (a)).

Le schéma alternatif est basé sur un répertoire et qui implique des agents de répertoire où les services sont enregistrés et des requêtes de services sont traitées (Figure 12 (b)) [JERO 04].

Nombreux sont les protocoles proposés pour pallier le problème de découverte des services dans les réseaux de type MANET, ceci fera l'objet du prochain chapitre.

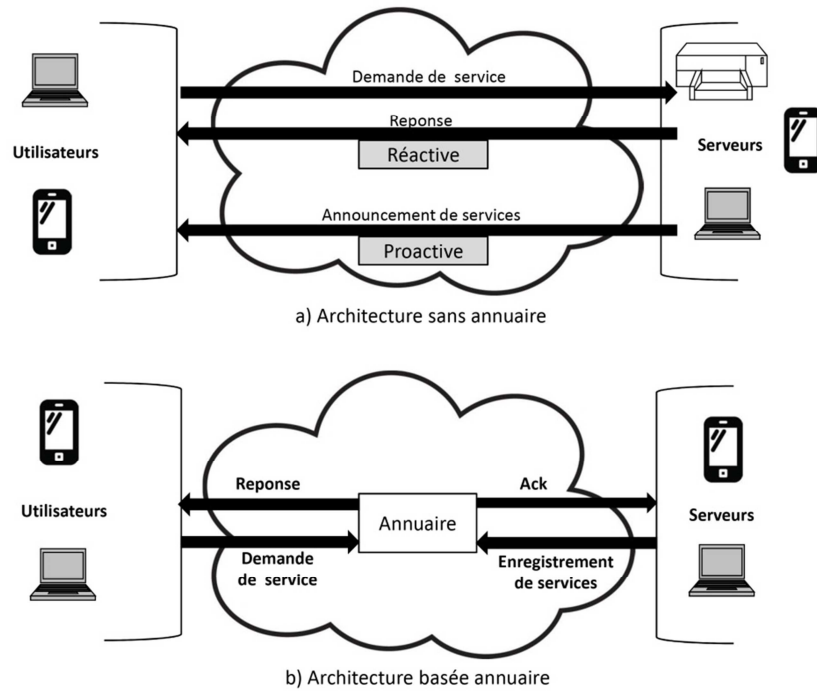


Figure 12. Architectures de découverte de services dans MANET [JERO 04].

I.5 Conclusion

Comme toute innovation, l'apparition des services a donné lieu à un ensemble d'opportunités et d'applications nouvelles. Aujourd'hui, les services Web sont partout présents en nombre. Qu'il s'agisse de services de réservations en ligne ou bien de gestion de comptes bancaires et même d'applications métiers, tous ces services partagent en commun le fait d'être accessibles sous forme de services Web. Comme nous avons vu dans ce chapitre, les services web parlent XML, ils sont décrits dans un format WSDL, précisant les méthodes pouvant être invoquées, leurs signatures et les points d'accès du service (URL, Port., etc.). Ces méthodes sont accessibles via le protocole SOAP. Les services Web sont centralisés (leurs publications) dans un référentiel commun UDDI afin de faciliter leur recherche. Mais à cause du changement fréquent dans la topologie des réseaux dans un environnement mobile ad-hoc, l'absence d'un annuaire fixe et centralisé, la capacité de la connexion sans fils et les contraintes énergétiques des équipements mobiles rend la réalisation des protocoles de découverte des services web conçu pour les réseaux ad-hoc un challenge très sollicité, plusieurs approches et protocoles ont été élaborés dans ce sens.

Dans le chapitre suivant nous allons présenter les différentes approches existantes que nous avons étudiées.

Chapitre II :

Synthèse des travaux existants

II.1 Introduction

La recherche dans le domaine des réseaux ad hoc s'est initialement concentré sur le routage, et plusieurs protocoles de routage tels que DSR, AODV, TORA, OLSR ... etc. ont émergé. Cependant, l'hypothèse de départ de ces protocoles de routage c'est que l'adresse de destination du paquet est connue d'avance au niveau du nœud source.

Les applications dans les réseaux ad hoc utilisent souvent des ressources ou des services qui sont présents dans un autre nœud mobile, il est important pour les applications de découvrir d'une manière transparente les services distants. Les protocoles de routage existants ne sont pas adaptés pour les processus de découverte de services, puisque dans ce dernier cas, l'adresse de destination n'est pas initialement connue. Pour remédier à cette situation, plusieurs approches et protocoles ont été proposés. Dans ce chapitre nous allons présenter plusieurs travaux existants qui traitent la problématique mentionnée.

II.2 Travaux existants

D'après notre étude nous avons classé les travaux existants sur le domaine de découverte de service dans MANET comme suit: Les approches adaptées dans MANET et les approches destinées pour MANET. Les éléments clés de classification des approches destinées pour MANET sont: la Description du Service, la Technique de Découverte, la Qualité de Service QoS et le Type de Réseau.

II.2.1 Approches adaptées dans MANET

Dans cette catégorie de classification on trouve les travaux suivants :

II.2.1.1 SLP (Service Location Protocol)

Le protocole SLP (Service Location Protocol) [VEIZ 97] est un standard d'Internet (IETF), il définit trois types d'agents selon les principes SOA, qui sont l'agent utilisateur (User Agent), l'agent de service (Service Agent), et l'agent d'annuaire (Directory Agent). Son principe de fonctionnement est comme suit, l'agent utilisateur communique avec l'agent

Annuaire pour rechercher et récupérer des informations sur un service souhaité. L'agent de service représente un ou plusieurs services offerts par un nœud. L'agent annuaire collecte toutes les annonces à partir des agents de services. En l'absence de l'Agent Annuaire, l'Agent utilisateur envoie des demandes de service en mode multicast. Les agents de Services écoutent sur l'adresse de multicast SLP et répondent avec un message en unicast s'ils fournissent le service demandé. Tous les services sont représentés par une URL. L'agent utilisateur découvre la présence de l'agent Annuaire au démarrage en envoyant une requête de découverte d'agents Annuaire. Ce dernier périodiquement annonce sa présence dans l'adresse de multicast.

Le protocole SLP est destiné aux réseaux locaux il n'est pas adapté aux réseaux ad hoc avec des nœuds mobiles de plus il est limité puisqu'il utilise seulement une URL contenant le nom de service pour représenter un service.

II.2.1.2 Salutation

Salutation [SALU 99] est une autre architecture de coopération, qui est développée par le Consortium Salutation, pour résoudre les problèmes de découverte de service. L'architecture Salutation est composée de deux éléments importants : Salutation_Manager et Transport_Manager. Le Salutation_Manager est le cœur de l'architecture, similaire au Lookup_Service dans Jini. Un fournisseur de service enregistre ses services avec le Salutation_Manager. Quand un client demande au Salutation_Manager local un service, la recherche sera effectuée par coordination entre tous les Salutations_Managers dans le réseau. Ensuite le client pourra utiliser le résultat retourné. Les Salutations_Managers utilisent RPC (Remote Procedure Call) pour communiquer entre eux. Salutation ne possède pas de réel système de notifications d'évènements. En outre, on ne peut pas s'abonner au Manager pour connaître périodiquement la disponibilité des services. Le Salutation_Manager se situe au-dessus de Transport_Manager qui offre un canal de communication indépendamment de couche de transport réseau.

Le consortium Salutation n'existe plus. Il est donc probable que Salutation soit de moins en moins utilisée.

II.2.1.3 JINI

JINI [JINI 98] est un produit de Sun Microsystems basé sur JAVA. En plus de la découverte de service, Jini fournit aussi l'invocation des services et la notification des

événements. Il fonctionne comme suit : Quand un fournisseur de service rejoint le réseau il lance en multicast dans le réseau local une découverte d'un *Lookup_Service* pour enregistrer ses services. Après la découverte, il télécharge les objets de ses services dans le *Lookup_Service*, l'objet contient une interface de langage de programmation java du service, elle inclut les méthodes que les utilisateurs ou leurs applications vont invoquer pour exécuter les services. Quand un client veut un service il recherche d'abord un *Lookup_Service* de la même manière que le fournisseur de service. Ensuite il recherche le service souhaité par son interface et ses attributs descriptifs. Par la suite, l'objet de service est téléchargé vers le client. Ce dernier pourra donc interagir avec le fournisseur de service à travers l'objet de service qui fonctionne comme un proxy. Dans le cas où il n'y a aucun *Lookup_Service*, le client prétend qu'il est lui-même un *Lookup_service* de manière à ce que les fournisseurs enregistrent leurs services chez lui. Il pourra ensuite prélever les services désirés localement et supprimer le reste.

Ce protocole à l'inconvénient d'être trop gourmand en matière de bande passante, et plus important encore, il n'est pas adapté aux réseaux de type MANET.

II.2.1.4 UPnP (Universal Plug and Play)

UPnP [UPNP 15] est dit être destiné au réseau ad hoc, réseau à domicile, petites entreprises, lieux publics...etc. Avec UPnP le client est considéré comme un point de contrôle et quand un point de contrôle découvre un service, il peut obtenir plus d'informations sur le périphérique offrant le service en téléchargeant une description XML fourni via une URL. Cette description permet au point de contrôle de contrôler le service, de recevoir des notifications d'événements, ou peut-être mettre en place une interface utilisateur pour le service dans un navigateur.

UPnP utilise le protocole Simple Service Discovery (SSDP) pour la découverte de service. Ce protocole est utilisé pour annoncer la présence d'un équipement aux autres ainsi qu'à la découverte d'autres équipements ou services.

Quand un équipement UPnP est ajouté au réseau il envoie en multicast une annonce de chacun de ses services offerts. Ce message doit contenir une validité de service de 30 minutes minimum. Si un équipement quitte le réseau ou révoque un service avant l'heure d'expiration, il doit envoyer en multicast un message correspondant à chacune de ses annonces informant que le service n'est plus disponible. Les points de contrôle enregistrent

les informations sur les annonces de services dans une mémoire cache. Quand un point de contrôle rejoint le réseau il recherche des équipements qui l'intéresse en multicast. Les équipements qui offrent le service répondent en unicast. Avec ce fonctionnement tous les points de contrôles auront des informations sur tous les services disponibles dans le réseau à un instant donné.

Dans UPnP, il est supposé qu'un nœud reste à jour concernant les services disponibles dans le réseau. Ceci est réalisé en demandant des informations sur les services disponibles en rejoignant le réseau, puis en mettant en cache ses informations. Toutefois, l'hypothèse selon laquelle un nœud reste à jour en utilisant une telle méthode ne peut pas être maintenu tactiquement dans un réseau ad hoc. Cela est dû au fait que si deux réseaux ad hoc (MANET) sont interconnectés après avoir été séparés physiquement, les nœuds dans une partition aura raté les annonces des nœuds dans l'autre partition.

II.2.2 Approches destinées pour MANET

II.2.2.1 La description du service

Dans cette deuxième classification des travaux on trouve ceux qui se basent sur la description des services offerts dans le réseau comme XML (Extensible Markup Language) et DAML (DARPA Agent Markup Language). Ces approches incluent :

II.2.2.1.1 DReggie

Le projet DReggie [DIPA 01] est une tentative pour améliorer les mécanismes de matching dans JINI et d'autres systèmes de découverte de services. L'idée principale dans DReggie est d'offrir une découverte sémantique. Dans cette approche, une requête de découverte de service contient la description d'un service «idéal» = celui dont les capacités correspondent exactement aux exigences. Ainsi, le matching implique désormais la comparaison des exigences spécifiées avec les capacités des services existants. Selon les besoins, un match peut se produire même si une ou plusieurs capacités ne correspondent pas exactement. Les descriptions des services dans le système DReggie sont décrites avec DAML. Le processus de matching sémantique qui utilise ces descriptions est effectué par un moteur de raisonnement. Au cœur de DReggie il y a un JINI Lookup_Service (JLS) amélioré qui offre une découverte intelligente des services JINI. DReggie conserve le mécanisme de matching actuellement utilisé par l'infrastructure de recherche et Discovery sous JINI. Les

clients qui tentent de découvrir les services ont la possibilité d'utiliser l'un des deux mécanismes de matching.

Quand un service s'enregistre avec le JLS amélioré, il enregistre une description DAML ainsi que son interface. La description DAML stockée dans le Lookup_Server et mise à jour chaque fois que le service s'enregistre à nouveau. Un client qui tente une découverte de services crée une description DAML du service souhaité comprenant diverses contraintes. Le système DReggie suppose que la requête émise par un client utilise la même ontologie qu'un service utilise pour se décrire. Le matching DAML dans JLS gère des contraintes telles que les exigences du client, le coût, la mobilité, etc. Il assure que le client ne découvre que les services qu'il est capable d'exécuter en termes d'exigences matérielles ou logicielles. [DIPA 01]

II.2.2.1.2 Konark

Dans *KONARK* [SUMI 03], qui est un protocole de découverte des services web pour les réseaux ad-hoc, qui vise les applications de e-commerce mobile, chaque nœud dans le réseau joue le rôle d'annuaire des services disponibles dans le réseau, les auteurs utilisent un arbre dans chaque nœuds pour classifier les services disponibles dans le réseau, la classification est générique au plus haut niveau (racine) et devient de plus en plus précise en descendant l'arbre. Les auteurs ont proposé un mécanisme de découverte et publication des services web, pour la Publication, chaque fournisseur diffuse ses nouveaux services dans le réseau pour que chaque nœud l'intègre dans son arbre (réplication des informations sur les services disponibles dans le réseau). Pour la phase Découverte, si un client cherche un service il diffuse une demande basée sur un nom ou un type de services, les fournisseurs concernés renvoient leurs réponses contenant les services souhaités ainsi que l'URL où se trouvent leurs descriptions WSDL, pour que le client l'enregistre dans son arbre.

L'inconvénient majeur du protocole *KONARK* c'est le gaspillage important des ressources à cause des sauvegardes des mêmes informations dans chaque nœud du réseau, même si l'information ne sera pas utilisée par le nœud.

II.2.2.1.3 WSDRS (Web Service Discovery Replication and Synchronization)

Dans l'approche *WSDRS* [LUKA 06], les auteurs ont segmenté le réseau en zones pour diminuer le trafic, car il n'est pas évident que chaque nœud joue le rôle de registre de ses propres services encore moins une réplication des mêmes informations dans tous les nœuds

du réseau, donc ils supposent que dans chaque zone il y a un Master-Host qui joue le rôle de registre de publication UDDI, il peut aussi reconnaître tous les nœuds qui se connectent dans sa zone, la sélection des groupes au niveau client se fait d'une manière automatique ou manuelle en choisissant son Master-Host lui-même. Ils ont proposés également un mécanisme de réplication et synchronisation des registres UDDI pour assurer la fiabilité et la cohérence des informations disponible dans le réseau (consommation de la bande passante).

L'inconvénient majeur du travail proposé dans [LUKA 06] est l'utilisation du principe de Master-Host comme point d'accès c.-à-d. s'ils supposent qu'il est fixe donc on revient à un réseau avec infrastructure sinon dans le cas contraire on peut se poser la question suivante : que se passera-t-il si le Master-Host se déconnecte de la zone ?

II.2.2.2 La technique de découverte

Dans cette catégorie on trouve les travaux qui ont montré qu'en utilisant les techniques basées sur la communication inter-couche du modèle OSI, on aboutit à une optimisation importante des performances du réseau, comme cela est présenté dans les travaux suivants :

II.2.2.2.1 SEDIRAN

SEDIRAN [ABDE 07] est un protocole de découverte et de publication des services web qui interagit avec les protocoles de routage afin d'optimiser la recherche des services, ses auteurs ont supposé deux types de services 'ordinaires et spéciaux', chaque fournisseur diffuse des annonces des services spéciaux qu'il offre, et dans le cas où un client désire rechercher un service ordinaire il envoie une requête basée sur la distance 'saut' et le type de service qu'il recherche, ce mécanisme nécessite l'interaction avec le client pour la sélection des services souhaités et se répète avec l'augmentation du 'saut' jusqu'à satisfaction du client, les nœuds intermédiaires utilisent un cache pour la sauvegarde des services ordinaires trouvés, l'inconvénient dans cette approche c'est la réplication des mêmes informations des services spéciaux dans tous les nœuds du réseau et une consommation importante de la bande passante par les messages de la recherche ainsi que l'utilisation supplémentaire de la ressource de stockage pour la sauvegarde intermédiaire des services trouvés même si ces derniers ne sont pas utilisés.

II.2.2.2 The need for cross-layer service discovery in MANETs

Dans [VARS 05] les auteurs ont présenté une nouvelle architecture basée inter-couche pour la découverte, la sélection et la redécouverte des services dans MANET, qui intègre la découverte de service avec les mécanismes de routage ad hoc. Ils ont montré analytiquement la supériorité de l'approche inter-couche sur les approches traditionnelles en termes des messages dans le réseau. En outre, de nombreux résultats de simulation pour DSR et DSDV (deux protocoles de MANET) montrent que l'approche basée inter-couche atteint un débit du réseau jusqu'à cinq fois plus grande qu'une implémentation traditionnelle basée sur la couche application. En misant sur le trafic de protocole de routage existant, l'architecture basée inter-couche permet aux clients de réévaluer leur sélection de service, et de passer à un meilleur serveur pour compenser les effets des changements de topologie de réseau.

II.2.2.3 PDP (Pervasive Discovery Protocol)

PDP [CELE 06] a été spécialement conçu pour les réseaux ad hoc; Le protocole n'a pas besoin d'un serveur central, donc aucune infrastructure n'est nécessaire. L'un des objectifs clés du PDP est de minimiser l'utilisation de la batterie dans tous les dispositifs, en particulier dans les plus limitées. Cela signifie que le nombre de transmissions nécessaires pour découvrir des services est réduit autant que possible. Les fournisseurs n'annoncent leurs services que lorsque d'autres clients demandent le service. Les annonces de services sont diffusées à tous les dispositifs dans sa portée, les nœuds ont un cache où ces annonces sont stockées. Les annonces de services comprennent non seulement les services offerts par le nœud lui-même, mais tous les services connus (demandés). PDP prend en compte l'hétérogénéité des nœuds dans le réseau, la réduction de la consommation des plus limités en privilégiant la réponse des dispositifs moins limités.

Chaque dispositif a un agent PDP User Agent (PDP-UA) et un Service Agent (PDP-SA). L'agent PDP-UA est un processus qui travaille pour le compte de l'utilisateur pour rechercher des informations sur les services offerts dans le réseau. Le PDP-SA est un processus de travail pour annoncer les services offerts par le dispositif. Le PDP-SA inclut toujours le temps de disponibilité T de dispositif dans ses annonces. Chaque dispositif a un cache contenant une liste des services qui ont été entendus à partir du réseau. Chaque élément du cache a deux champs: La description de service et le temps d'expiration du service. Le temps d'expiration du service est le temps estimé pour que le service demeure

disponible. Ce temps est calculé comme étant le minimum de deux valeurs: Le temps de disponibilité du nœud local, et la durée de vie de service annoncé. Les entrées sont supprimées de la mémoire cache quand ils expirent.

PDP a deux messages obligatoires: PDP_Service_Request, qui est utilisé pour envoyer des demandes de services, et PDP_Service_Reply, qui est utilisé pour répondre à un PDP_Service_Request annonçant les services disponibles. En outre, PDP a un message facultatif: PDP_Service_Deregister, qui est utilisé pour informer que le service n'est plus disponible. PDP prend en charge deux types de requête,; une requête/une réponse (1/1, le client est intéressé au service, et non pas dans quel dispositif il est offert) et une requête/réponses multiples (1/n, le client veut découvrir tous les périphériques du réseau offrant le service).

II.2.2.2.4 GSD (Group based Service Discovery)

GSD (Group-based Service Discovery protocol) a été initialement introduit dans [CHAK 02], ensuite il a été décrit en détail dans [CHAK 06]. GSD a trois opérations basiques : (1) le déploiement des paquets d'annonce de services. (2) la diffusion des paquets de recherche de services. (3) le routage des paquets de réponse de services, GSD introduit deux mécanismes intéressants : (1) la mise en cache des annonces de services. (2) la diffusion intelligente des paquets de demande de services basée sur le principe de groupe.

Un serveur doit générer des paquets d'annonce de services périodiquement, qui contiennent les descriptions des services fournis par le serveur. Les paquets d'annonce de services sont mis en cache et transmis en mode diffusion par les récepteurs. Le cache est appelé SIC (Service Information Cache). Lors de la génération des paquets d'annonce de services, si l'information dans la mémoire cache du nœud indique qu'il y a des serveurs dans son voisinage, les informations du groupe des services fournis par ces serveurs devraient également être incluses dans le paquet d'annonce de service. Le nombre maximal de sauts que les paquets d'annonce peuvent parcourir, notée ASB, est limité par un paramètre défini par l'utilisateur.

Selon le principe de diffusion basé groupe des paquets de demande de services, quand une requête de demande de service n'est pas matché par un nœud, ce dernier essayera de sélectionner des nœuds dans son cache pour leurs diffuser ce paquet, ces nœuds sélectionnés doivent avoir des voisins qui fournissent des services qui appartiennent au

même groupe que le service demandé. Si le nœud qui reçoit une nouvelle requête trouve un matching avec ses services, il envoie en mode unicast un paquet de réponse au nœud demandeur. [ZHEN 11]

II.2.2.3 La qualité de service QoS

Dans la catégorie des travaux qui se basent sur la qualité de service dans le processus de découverte on trouve:

II.2.2.3.1 TrustSD: Trust based service discovery in mobile ad-hoc networks

Les auteurs dans TrustSD [BHUM 15] ont utilisés le principe des systèmes de gestion de confiance dans un réseau présenté dans [PIRZ 06], [SILV 12], [CHEN 14] pour proposer un protocole où ils ont ajouté un paramètre de confiance qui permet de sélectionner un fournisseur parmi les fournisseurs qui ont répondu à la requête de client. Ce paramètre de confiance est calculé en utilisant un ensemble de paramètres qu'ils ont définis (Energie de nœud, Temps de réponse, La mobilité et La disponibilité du serveur). Ils ont utilisés une approche inter-couche afin d'ajouter ses paramètres dans les paquets de routage. Ils ont aussi ajouté une table pour enregistrer les informations sur les services découverts.

Le problème dans cette approche c'est que les auteurs font une découverte classique basée sur un nom de service et qui sélectionne automatiquement un service (fournisseur) en se basant sur le paramètre de confiance calculé à travers les informations mappés sur les paquets de routage, mais le nom de service souhaité par le client peut ne pas correspondre au service offert par le fournisseur puisque il représente une information non fonctionnelle. Ceci peut engendrer une contradiction dans la découverte ainsi que ce type de découverte peut rater un service qui correspond à la demande de client.

II.2.2.3.2 Ant based resource discovery and mobility aware trust management for mobile grid systems

Dans [SING 13] les auteurs disent que la plupart des mécanismes de sécurité existants sur les systèmes de grille prennent rarement en considération la mobilité des nœuds. Afin de surmonter ces problèmes, ils ont proposé une approche de découverte de ressource basée sur les colonies de fourmis et la gestion de la confiance de la mobilité pour les systèmes de grille mobiles. Au départ, les nœuds dits *Super-grille* sont sélectionnés dans le réseau en utilisant l'optimisation de colonie de fourmis sur la base des paramètres tels que la distance, la vitesse de CPU, la bande passante disponible et de la puissance résiduelle de la

batterie. Ces nœuds sélectionnés sont utilisés dans le mécanisme de découverte de ressources. Afin de maintenir une forte sécurité avec le système de gestion de la mobilité, ils ont adopté une méthode pour la collecte de réputation de confiance.

II.2.2.3.3 Trust based security mechanism for service discovery in MANET

Dans [PARI 13] ils concèdent que dans les réseaux MANET les différents services disponible, nécessitent des niveaux de sécurité différente. Par exemple les applications commerciales nécessitent moins de sécurité comparée aux applications militaires. Ainsi, le développement d'un mécanisme de découverte de service efficace compte tenu des critères ci-dessus est un travail compliqué. Dans leur travail, ils ont proposé un mécanisme de confiance basé sur la sécurité pour la découverte de services dans MANET. Ce mécanisme calcule le degré de confiance pour chaque nœud en prenant en considération son interaction et son comportement dans le réseau.

II.2.2.4 Le type du réseau

Dans cette catégorie on trouve les travaux ceux qui s'intéressent à l'utilisation des services web dans les réseaux de capteurs sans fils (SWS : Sensor Web Services) qui représente un domaine qui a émergé du besoin d'offrir les données capturé par les WSN (Wireless Sensor Networks) via le web, ces travaux sont :

II.2.2.4.1 Efficient integration of web services in ambient-aware sensor network applications

Dans [AMUN 06] les auteurs considèrent les réseaux de capteurs comme dispositifs intelligents qui collectent, détectent, calculent et interagissent avec plusieurs types d'environnements. Certaines applications telles que l'intervention d'urgence, la sécurité intérieure ...etc., ont un réel besoin d'interconnecter ces réseaux divers et accéder à des informations en temps réel. Alors que les protocoles Internet et les standards du Web fournissent des mécanismes bien développés pour accéder à cette information, relier ces mécanismes avec les réseaux de capteurs à ressources limitées est très difficile en raison de la volatilité des liens de communication. Dans leur travail les auteurs ont présenté un modèle de programmation orientée services pour les réseaux de capteurs qui permettent la découverte et l'accès aux services Web. Les Applications de réseaux de capteurs sont réalisées sous forme de graphes de services modulaires et autonomes avec des interfaces bien définies qui leur permettent d'être décrits, publiés, découverts et invoqués dans le réseau en offrant un moyen pratique d'intégration des services à partir de systèmes de

capteurs hétérogènes. Cette approche fournit une découverte dynamique, composition et liaison de services basés sur un algorithme efficace de localisation et de satisfaction de contraintes qui peut être utilisé pour développer des applications qui s'adaptent aux changements dans l'environnement.

II.2.2.4.2 An evaluation of mobility effect on tiny service discovery protocol for wireless sensor networks

Dans [AHME 14] les auteurs ont réalisé une étude comparative sur l'impact de la mobilité dans les protocoles de découverte des services tel que PDP, SDP et UPnP ...etc, dans leur étude ils se sont focalisés sur les performances du protocole TinySDP [KAUR 08]. Ils estiment que la sensibilité des réseaux de capteurs mobiles en matière de limitation des ressources rend le processus de découverte des services très délicat. Un grand temps d'attente des applications des capteurs sans fils va être passé dans seulement la découverte des nœuds en mouvement. Comme les nœuds doivent annoncer périodiquement leurs présences et être éveillé pour découvrir d'autres nœuds pour les services. Dans leur article, les auteurs ont mesuré l'effet de la mobilité des nœuds sur les performances de Tiny Service Discovery Protocol (TinySDP) dans les mobileWSNs (Mobile Wireless Sensor Networks). Ils ont réalisé leur étude d'évaluation selon trois principaux indicateurs qui sont : Le rapport de Succès, le nombre de messages transmis et le temps d'attente moyen. Les résultats de simulation qu'ils ont obtenu montrent que le taux de réussite et le temps d'attente moyen de TinySDP dans les réseaux de capteurs sans fils mobiles (mobileWSN) ont diminué par contre le nombre de message transmis a augmenté.

Le protocole TinySDP utilise une approche distribuée dans laquelle l'information est mise en cache dans le réseau plutôt que dans un serveur central. Toute annonce ou demande est envoyée sur le réseau plutôt qu'un nœud particulier agissant comme un serveur. Les publications sont diffusées pour tous les nœuds dans le réseau. Chaque nœud recevant le message le sauvegarde dans son cache. Avec TinySDP si un client recherche un service il envoie une demande, lorsqu'un nœud récepteur dans la trajectoire de la requête reçoit cette demande il vérifie s'il offre le service demandé ou non, si oui il renvoie une réponse de service à la source/client, et s'il ne peut pas répondre aux besoins des clients complètement (exigences de client), il transmet la demande de service le long de sa trajectoire, mais offre également de fournir son service limité. Et si dans le cas où le nœud récepteur ne fournit pas du service, il vérifie si le service est déjà annoncé et s'il possède une entrée dans son cache si oui il envoie la requête vers le fournisseur de service.

Le protocole utilise une normalisation des identifiants des services où chaque identifiant de service doit être unique pour tous les nœuds fournisseurs, ce qui est pratiquement non faisable vu la diversification des nœuds et de type de services offerts.

II.2.2.4.3 Autres travaux sur les Sensor Web Services

Une approche basé sur les SWS a été proposé dans [ASAD 11] pour les véhicules intelligentes où des services web ont été implémenté dans des capteurs intégré qui permet à la fois de recueillir des informations sur l'état du carburant par exemple et de rechercher le service le plus approprié par rapport aux prix et heures de pointes.

Dans [ASAD 13] les auteurs ont proposé une étude intéressante sur l'utilisation des Sensor Web services (découverte, composition et invocation), ils ont décrit des techniques de personnalisation de services Web (tel que la personnalisation du fichier WSDL) pour s'adapter aux capacités de nœuds de capteurs en prenant en considération les capacités de ses nœuds dans la découverte des services.

Parmi les travaux intéressants aussi qui utilisent cette technologie on trouve les travaux dans [ULLA 10] et [CHEN 10] où un ensemble de nœuds de capteurs ont été déployés dans le corps des patients pour surveiller les activités psychologiques et des problèmes de santé, tels que la température du corps, mouvement, et les fréquences cardiaques. Ici, chaque capteur peut accueillir un WS qui apporte certaines informations au serveur principal.

II.3 Conclusion

Ce chapitre a été consacré à la présentation de différentes approches de découverte de services dans les MANETs, nous avons classé ces approches en deux grandes catégories, Approches adaptées dans MANET et Approches destinées pour MANET. L'étude comparative de ses travaux est la base de notre contribution présentée dans le prochain chapitre où nous allons présenter une comparaison détaillée entre différents approches de différentes classes afin de proposer nos contributions.

Chapitre III :

Découverte des services web dans MANET

III.1 Introduction

Dans ce travail, nous proposons un modèle pour calculer le coût de l'énergie consommée et valider les protocoles de découverte de services Web dans les réseaux mobiles ad hoc. Nous proposons également une approche optimisée, qui a deux objectifs principaux: Le premier est de créer un nouveau mécanisme de découverte sémantique des services web dans les réseaux ad hoc mobiles, c'est à dire, effectuer une découverte sémantique basée sur les paramètres fonctionnels (entrées et sorties) présents dans les descriptions sémantiques OWL-S des services Web sans l'intervention du client dans la phase de sélection des services découverts ce qui signifie réaliser une découverte automatique et précise des services souhaités. Et le deuxième objectif c'est de développer une nouvelle stratégie de découverte, ce qui minimise le nombre de paquets générés dans le réseau durant le processus de découverte des services Web.

Ce chapitre est organisé comme suit, nous illustrons tout d'abord notre première contribution [AYAD 14] qui représente une étude comparative des travaux existants selon des paramètres que nous avons proposés, ainsi que les formules d'évaluation du coût d'énergie. Ensuite, nous présentons et développons notre approche optimisée que nous avons proposé [AYAD 16], Enfin nous terminons avec une étude comparative de l'approche proposée par rapport aux formules d'évaluation que nous avons proposés.

III.2 Contribution 1 : Evaluation du coût de l'énergie

Nous avons remarqué que chacune des approches étudiées dans le chapitre précédent viole une ou plusieurs contraintes posées dans les réseaux ad-hoc tels que la bande passante et les ressources de stockage ...etc. Ce qui engendrera une consommation importante en matière d'énergie. Nous avons aussi remarqué que les approches de découverte proposées se basent sur soit les IDs des services, leurs noms ou leurs types. Avec une intervention du client pour la sélection de service qui répond à ses besoins, donc une recherche classique.

	BANDE PASSANTE				RESSOURCES			MECANISME DE DECOUVERTE		ENERGIE
	Paquets de publication	Packets de synchronisation	Paquets de découverte	Paquets de réponse	Réplication des informations	Table de routage	Stockage de services	Elements clés de découverte	Recherche et Sélection	Energie consommé
KONARK	Broadcast	Non	Unicast	Unicast	Tous les services entre tous les noeuds	Routes vers tous les fournisseurs du réseau	BD «arbre» de tous les services existants	Nom ou Type de service	Intervention du client	Importante
WSDRS	Broadcast interZones	Entre UDDIs	Unicast	Unicast	Entre UDDIs	Routes vers UDDI de chaque zone	BD UDDI pour chaque zone	Identificateurs des services ou nom du service 'incertaine'	Intervention du client	Importante
SEDIRAN	Broadcast	Non	Broadcast	Unicast	Seulement les services spéciaux entre tous les noeuds	Routes vers tous les fournisseurs des services spéciaux + routes vers les services ordinaires souhaités	BD de tous les services spéciaux + Cache des services ordinaires	Identificateurs des services	Intervention du client	Importante
TrustSD	Non	non	Broadcast	Unicast	non	Routes vers tous les fournisseurs des services	Cache des services découvert	Nom de service (parameter non fonctionnel)	automatique	moyenne
TinySDP	Broadcast	Non	Unicast	Unicast	Tous les services entre tous les noeuds	Routes vers tous les fournisseurs des services	Cache des services dans le réseau	Identificateurs des services + QOS	Intervention du client	Importante

Tableau 1. Comparaison entre les approches existantes.

Dans le tableau 1, nous avons recueilli des contraintes C_i qui influencent d'une manière directe ou indirecte sur les principales contraintes techniques à respecter dans les réseaux mobiles ad hoc, où:

$$C_i = \{Bande\ passante, Ressources\ de\ stockage, Mécanisme\ de\ découverte, Energie\}$$

Nous avons utilisé ces contraintes déduites de notre étude sur les approches existantes de découverte des services web dans les réseaux ad-hoc ainsi que les résultats de travaux dans [FEEN 01] et [GERA 06] pour proposer un modèle présenté au travers de formules pour mesurer le coût énergétique qui nous permettra d'évaluer les protocoles de découverte des services web dans les réseaux ad hoc par rapport à l'énergie globale consommée.

La formule (1) permet le calcul du coût de l'influence sur l'énergie consommée dans chaque nœud par rapport aux contraintes C_i .

$$Cost_{enr} = Cost_{ack} + \prod_{i=1}^3 Cost_{C_i} \quad (1)$$

Où

$Cost_{enr}$: Le coût d'influence sur l'énergie consommée dans chaque nœud.

$Cost_{ack}$: Le coût d'énergie consommée par l'émission des paquets de contrôle.

$Cost_{C_i}$: Le coût d'influence sur l'énergie par rapport à chaque contrainte C_i dans chaque nœud.

Pour calculer $Cost_{C_i}$ qui représente le coût d'influence sur l'énergie consommée dans chaque nœud par rapport au nombre, taille et type de paquets qui circulent dans le réseau, on s'est orienté vers les travaux proposés dans [FEEN 01] qui ont présenté plusieurs résultats de mesure de la consommation d'énergie. Ils ont identifié quatre états de consommation d'énergie

- Emission : envoi des paquets,
- Réception : réception des paquets,
- Repos : en attendant un transfert de paquet
- Endormi : l'état de la plus basse consommation d'énergie où le nœud ne peut ni recevoir ni émettre.

Le coût d'énergie associé à chaque paquet à chaque nœud est représenté par le coût total incrémentiel m proportionnel à la taille du paquet $size$, et un coût b fixe associé au canal d'acquisition. La formule (2) suivante est utilisée pour calculer le coût d'énergie de chaque paquet dans chaque nœud du réseau.

$$Cost = m \times size + b \quad (2)$$

La formule (3) représente la formule que nous avons proposée pour calculer $Cost_{C1}$

$$Cost_{C1} = Cost_{E_{adv}} + Cost_{E_{disc}} + Cost_{E_{sync}} + Cost_{E_{resp}} \quad (3)$$

Où

$Cost_{E_{adv}}$: Le coût d'influence sur l'énergie consommée par l'émission et la réception des paquets de publication des services

$Cost_{E_{disc}}$: Le coût d'influence sur l'énergie consommée par l'émission et la réception des paquets de découverte des services

$Cost_{E_{sync}}$: Le coût d'influence sur l'énergie consommée par l'émission et la réception des paquets de synchronisation entre les nœuds

$Cost_{E_{resp}}$: Le coût d'influence sur l'énergie consommée par l'émission des paquets de réponse des fournisseurs

La formule (4) suivante représente la formule proposée pour calculer $Cost_{C2}$ qui représente le coût d'utilisation des ressources par rapport à l'énergie consommée pour le traitement de l'information stockée.

$$Cost_{C2} = Cost_{R_{repl}} + Cost_{R_{rout}} + Cost_{R_{stor}} \quad (4)$$

Où

$Cost_{R_{repl}}$: représente le coût de répliquon des mêmes informations entre nœuds

$Cost_{R_{rout}}$: représente le coût par rapport aux routes du réseau sauvegardé dans les nœuds

$Cost_{R_{stor}}$: représente le coût par rapport à la structure de sauvegarde des publications des services

La formule (5) représente la formule proposée pour calculer $Cost_{C3}$ qui représente le coût d'influence du mécanisme de découverte proposé par rapport à l'énergie consommée dans ce processus.

$$Cost_{C3} = Cost_{M_{resr}} + Cost_{M_{client}} \quad (5)$$

Où

$Cost_{M_{resr}}$: représente le coût par rapport aux éléments de découverte

$Cost_{M_{client}}$: représente le coût par rapport à l'intervention du client dans la recherche du service

La formule de calcul du coût d'influence sur l'énergie globale dans tout le réseau par rapport à l'approche de découverte est représentée comme suit

$$Cost = \sum_{j=1}^{nouds} \left(Cost_{ack_j} + \prod_{i=1}^3 Cost_{ci_j} \right) \quad (6)$$

La figure 13 représente le graphe de comparaison entre les différentes approches de découverte étudiées en matière de coût d'énergie calculé à partir de la formule proposée, où l'axe *Node* représente le nombre des nœuds disponible dans le réseau et l'axe des *Energy-Cost* représente le coût de l'énergie calculer à partir de la formule (6).

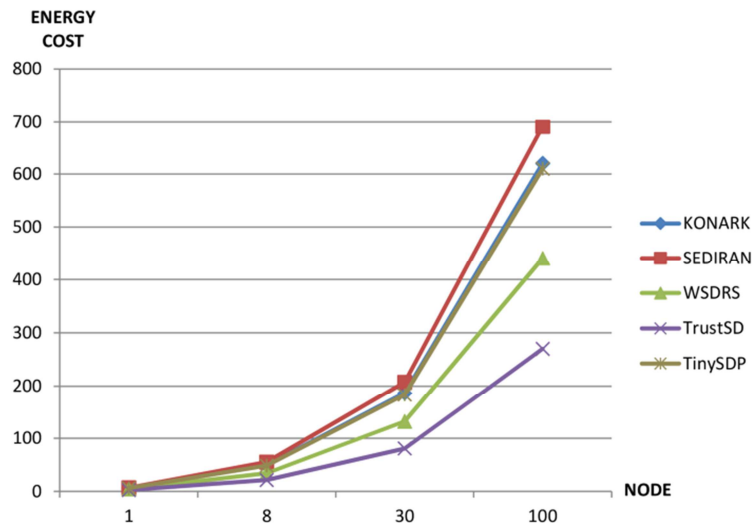


Figure 13. Comparaison entre approches de découverte en matière de coût d'énergie.

Comme le montre la figure 13 après évaluation des approches étudiées, l'approche proposée dans TrustSD (Bhumika et al., 2015) est moins gourmande en termes d'énergie consommée par rapport aux autres approches, cela est dû au mécanisme proposé, parce que les auteurs ont proposé un protocole inter-couche où ils ont utilisé un paramètre de confiance calculé en fonction d'un ensemble de caractéristiques des serveurs (énergie du

serveur, le temps de réponse et la mobilité des serveur) qui sont ajoutés dans le paquet de réponse du fournisseur. Le paramètre de confiance garantit une qualité de service QoS et une sélection automatique des fournisseurs.

III.3 Contribution 2 : Une nouvelle approche de découverte optimisée

L'approche que nous avons proposée s'appuie sur l'étude précédente concernant la consommation d'énergie. En outre, nous avons proposé un nouveau protocole qui vise deux principaux objectifs :

- Réaliser un mécanisme de découverte sémantique des services web dans les réseaux ad-hoc c'est-à-dire effectuer une découverte sémantique en se basant sur les descriptions fonctionnelles des services web tels, les *Inputs* et *Outputs* présentes dans les descriptions sémantique OWL-S des services web et ce, sans intervention du client dans la phase sélection du service. Ceci aboutira à l'automatisation de la phase sélection c'est-à-dire une sélection automatique et précise des services recherchés.
- Élaborer une nouvelle stratégie de découverte qui permet de réduire le nombre de paquets générés dans le processus de découverte de service Web, et également de réduire l'espace de stockage nécessaire afin de réduire la quantité d'énergie utilisée.

III.3.1 Motivations

A cause des limites des réseaux ad hoc nous avons proposé un protocole où nous supposons que chaque nœud fournisseur est l'annuaire de ses propres services afin d'éviter la surcharge du réseau avec des paquets d'annonce des services comme dans les travaux [CHAK 01] [DIPA 02] [LUKA 06] [SUMI 03], [ABDE 07] et (Kaur et al., 2008), et c'est le changement de la topologie permanent qui justifie notre supposition, puisque dans les approches existantes nous avons remarqué qu'à chaque mouvement dans la topologie cela déclenchera soit une redésignation des registres de publication ensuite une réexécution du mécanisme de synchronisation, ou encore une réexécution du mécanisme d'annonce des services dans tous le réseau.

Notre approche est basée sur l'interconnexion de la couche application et la couche de réseau parce que l'utilisation d'une approche inter-couche dans la découverte de service tel que présenté dans (Varshavsky et al., 2005) et (Garcia-Macias, 2005), augmente la

performance du réseau en termes de bande passante et de la consommation d'énergie globale, aussi dans (Laura, 2001) l'auteur a prouvé qu'il existe une relation directe entre le nombre de paquets, la bande passante et l'énergie consommée.

Dans notre protocole, nous avons modifié et adapté le protocole AODV existant afin de satisfaire nos besoins. Le choix de protocole de routage AODV est justifié par sa simplicité de d'implémentation, aussi il ne nécessite pas beaucoup de puissance de calcul, et plus important encore, c'est un protocole de routage réactif, ce qui signifie qu'il ne cherche les routes vers un destinataire que sur demande.

III.3.2 Architecture proposée

L'architecture proposée est illustré dans la figure 14

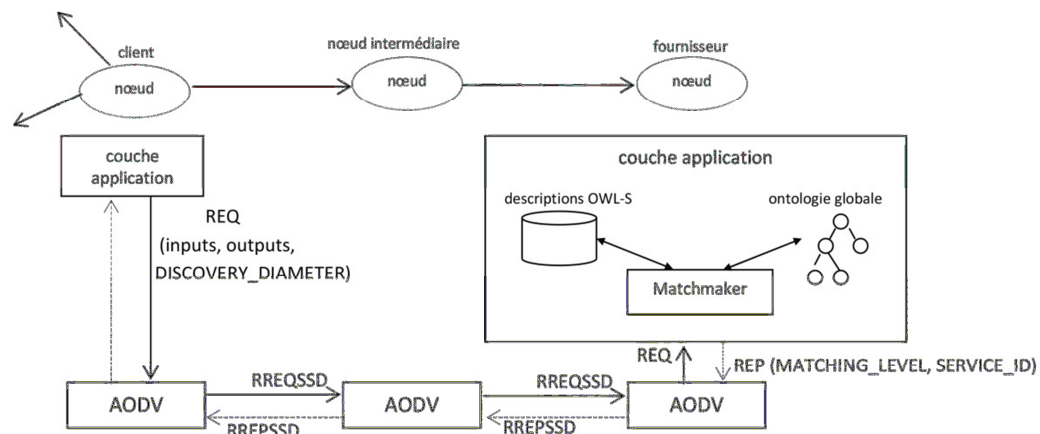


Figure 14. Architecture du système.

Dans notre nouvelle approche, nous avons ajouté deux nouveaux types de paquets au protocole AODV, RREQSSD (Route Request Semantic Services Discovery) et RREPSSD (Route Reply Semantic Services Discovery), ils sont similaires aux paquets RREQ (Route Request) et RREP (Route Reply) mais avec des champs en plus qui correspondent aux messages REQ et REP, où REQ représente la requête contenant les informations nécessaires pour découvrir des routes vers le service souhaité par le client et REP représente la réponse du fournisseur qui offre le service souhaité.

III.3.3 Le paquet de découverte RREQSSD

Dans notre approche pour rechercher un service, au niveau de la couche Application le client saisit toutes les informations nécessaires à la recherche de service souhaité (information sur le client, nom du service, paramètre fonctionnelles du services tels inputs/outputs, pre-conditions et effets ... etc.) ainsi que la distance de recherche *DISCOVERY_DIAMETER* (le *DISCOVERY_DIAMETER* est utilisé pour limiter le champ de recherche selon les besoins de client. Ceci aboutira à une réduction importante du nombre des paquets de recherche qui circulent dans le réseau donc une optimisation importante de l'énergie globale consommé dans ce processus de découverte), ensuite un paquet REQ contenant les concepts fonctionnels INPUTS/OUTPUTS et le diamètre de découverte *DISCOVERY_DIAMETER* va être envoyé vers le protocole AODV de la couche réseau (le choix des paramètres INPUTS/OUTPUTS est justifié dans la section *Le module Matchmaker*), pour que ce dernier l'intègre dans son processus de recherche de route vers la destination souhaitée via le nouveau paquet RREQ (figure 15) du protocole AODV que nous avons proposé.

Type	J	R	G	D	U	Reserved	Hop Count	REQ: <i>DISCOVERY_DIAMETER</i>
REQ : SERVICE_INPUTS								
REQ: SERVICE_OUTPUTS								
RREQSSD ID								
Destination IP Address								
Destination Sequence Number								
Originator IP Address								
Originator Sequence Number								

Figure 15. Le paquet RREQSSD.

Le nouveau paquet RREQSSD va être diffusé en mode *broadcast* vers tous les voisins selon le champ *DISCOVERY_DIAMETER* qui désigne le nombre de nœuds que le message peu traverser.

L'organigramme de la figure 16 montre le comportement des nœuds de réseau lors de la réception du paquet RREQSSD.

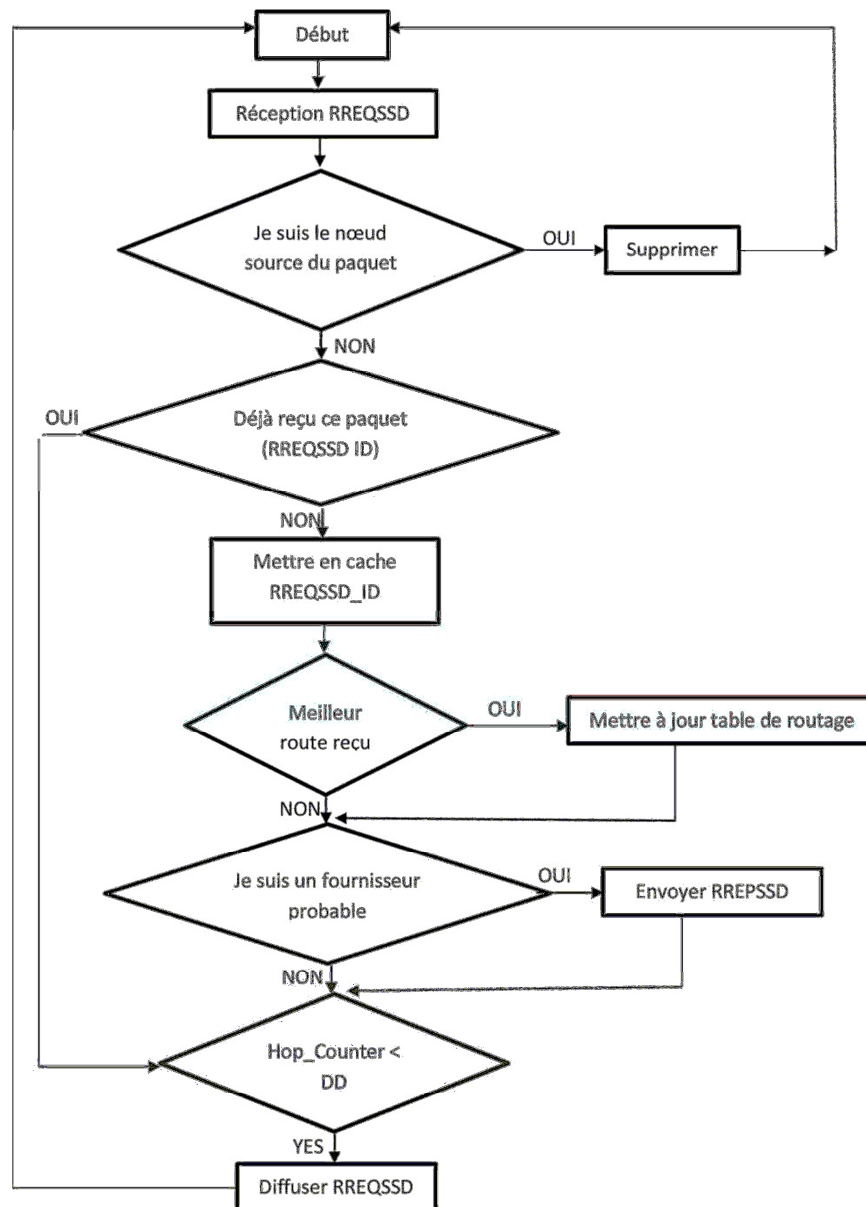


Figure 16. Organigramme de réception du paquet RREQSSD.

Dans le cas où le nœud intermédiaire est un fournisseur de services, et s'il n'a pas déjà reçu ce paquet c'est-à-dire un nouveau RREQSSD_ID, au niveau de la couche réseau le paquet REQ va être envoyé au module Matchmaker au niveau de la couche application. Le Matchmaker va extraire les inputs/outputs dans le paquet REQ, et en utilisant une ontologie globale il va calculer les correspondances avec les inputs/outputs des services web offerts par ce fournisseur selon le principe proposé dans (Paolucci et al. 2002a) présenté dans la partie 3 du chapitre 1. Ensuite il enverra une réponse REP. Dans le cas où le fournisseur ne trouve pas de correspondances avec la requête du nœud source, il va vérifier si le nombre de sauts est inférieur au paramètre Discovery_Diameter, si oui il diffusera le paquet RREQSSD à ses voisins.

L'algorithme 4 représente les nouvelles actions exécutées par le protocole AODV, lors de la réception du nouveau paquet RREQSSD.

```

Algorithme RREQSSD_Reception( )
  si Nouveau_RREQSSD_ID alors
    Sauvgarder_RREQSSD_ID ( );
    Extraction_DISCOVERY_DIAMETER ( );
  si DISCOVERY_DIAMETER >= Hop_Count alors
    Envoyer_Vers_Couche_Application (REQ);
  sinon
    Supprimer_RREQSSD( );
  Finsi

```

Algorithme 4. Actions du protocole AODV à la réception de RREQSSD.

L'algorithme 5 représente les actions effectuées au niveau de la couche application à la réception du paquet REQ

```

Algorithme Couche_Application(REQ)
  si Fournisseur() alors
  si Matchmaker(REQ) alors
    Generer_REP(MATCHING_LEVEL, SERVICE_ID);
    Envoyer_Vers_AODV_Couche_Reseau(REP);
  finsi
finsi

```

Algorithme 5. Actions de la couche application à la réception du paquet REQ.

Le message REP généré au niveau de la couche application contient le degré de correspondance *MATCHING_LEVEL* du service trouvé par le module Matchmaker par rapport aux inputs/outputs de la requête de client et les inputs/outputs des descriptions OWL-S des services offerts par le fournisseur, ainsi que le *SERVICE_ID* du service trouvé.

L'algorithme 6 suivant représente les nouvelles actions exécutées au niveau du la couche réseau (protocole AODV) à la réception de message REP.

```

Algorithme AODV_Couche_Reseau(REP)
  .....
  Generer_nouveau_RREPSSD(REP);
  Envoyer (RREPSSD);

```

Algorithme 6. Actions du protocole AODV à la réception du paquet REP.

III.3.4 Le paquet de réponse RREPSSD

La figure suivante représente la structure du nouveau paquet RREPSSD selon notre approche.

Type	R	A	Reserved	Predix Sz	Hop Count	REP: MATCHING_LEVEL
REP: SERVICE_ID						
Destination IP Address						
Destination Sequence Number						
Originator IP Address						
Lifetime						

Figure 17. Structure du nouveau paquet RREPSSD.

Le protocole AODV au niveau de la couche réseau, après avoir reçu le message REP provenant de la couche application supérieur, il l'encapsule dans le nouveau paquet RREPSSD proposé et l'envoie vers le client. Les nœuds traversés mettent à jour leurs tables de routage. Le protocole AODV du nœud source à la réception du premier paquet RREPSSD il l'enregistre pour une période d'attente *WAITING_TIME* qui correspond à la durée d'attente d'autres paquets RREP provenant d'autres fournisseurs arrivent, une fois tous les paquets reçus il calcule avec le *MATCHING_LEVEL* le chemin le plus court, une fois la route du meilleur match trouvée, la couche réseau envoie vers la couche application le *SERVICE_ID* du service trouvé pour qu'elle commence à émettre les données. Si, ultérieurement, la source reçoit un RREPSSD de la part du même fournisseur contenant un numéro de séquence supérieur ou égale à celui reçu avant, mais avec un nombre de sauts plus petit, elle mettra à jour ses informations de routage vers cette destination et commencera à utiliser la meilleure route. Une route est maintenue aussi longtemps qu'elle continue à être utilisée, c'est-à-dire tant que des données transitent entre la source et la destination.

En cas de coupure de liaison pendant le transfert de données, le nœud d'extrémité envoie un paquet RERR (Route Error) classique au nœud source pour l'informer que la destination est à présent inaccessible. Si le nœud source a toujours besoin d'une route vers cette destination pour terminer le transfert des données il lancera un processus de découverte de route vers une destination spécifique en utilisant uniquement les paquets RREQ et RREP classique.

L'organigramme de la figure 18 montre le comportement des nœuds du réseau lors de la réception d'un paquet RREPSSD.

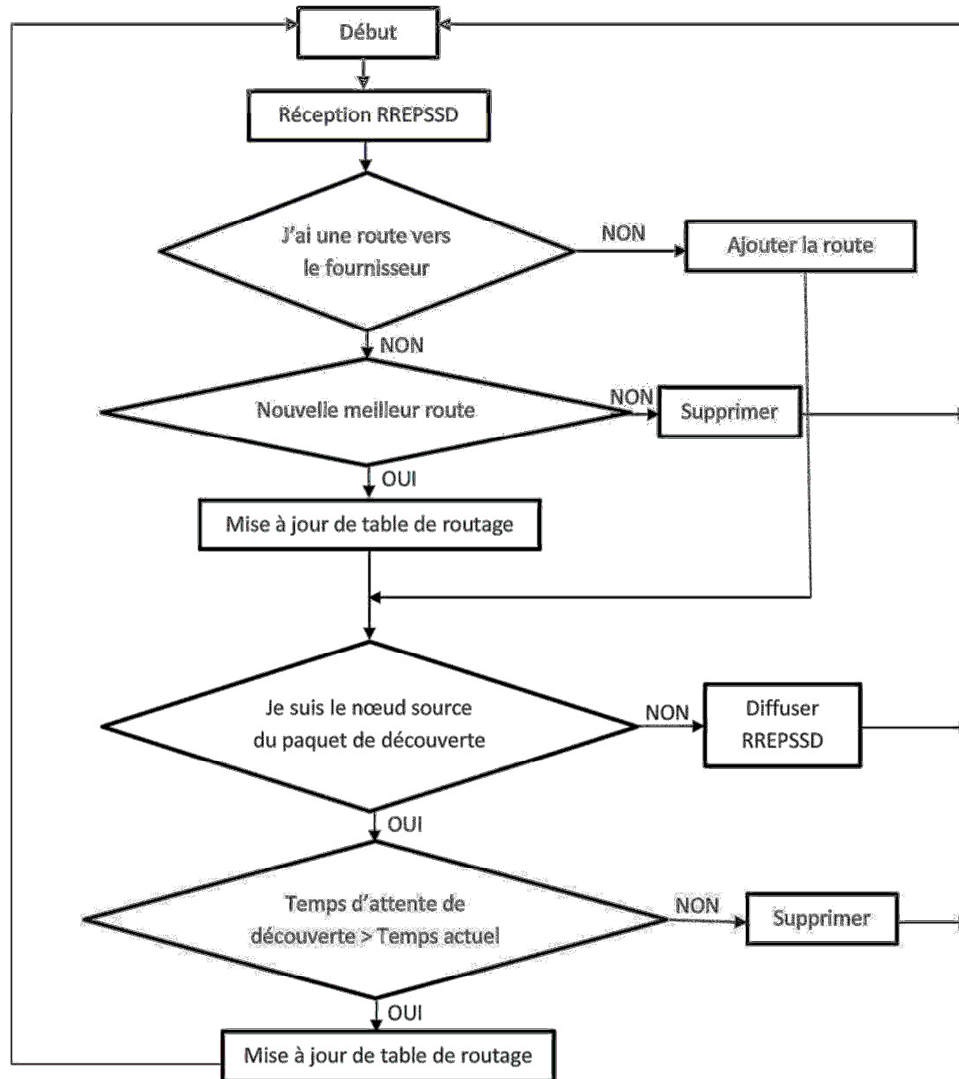


Figure 18. Organigramme de réception du paquet RREPSSD.

III.3.5 Un routage et une découverte de service web optimisée

Afin d'expliquer le fonctionnement de notre approche et illustrer les différentes optimisations apportées par notre protocole, nous supposons un scénario de catastrophe (tremblement de terre), dans lequel est déployé un réseau ad hoc d'urgence, et afin de sauver et donner les premiers soins aux blessés avant de les amener à l'hôpital, l'ambulancier doit d'abord vérifier des informations sur l'état de santé du patient, pour voir s'il a une maladie chronique par exemple, ou s'il est allergique à une injection précise.

Dans ce cas l'ambulancier ira chercher un service web publié par un hôpital ou un centre médical spécifique qui fournit un accès à une base de données qui contient toutes les informations médicales nécessaires sur les patients. Dans notre étude de cas, par exemple, nous supposons que les informations sur le service recherché sont:

- *Nom du service : Maladie_Cardiaque.*
- *Inputs : Nom; Prénom; N° Immatriculation.*
- *Outputs : Hypertention*
- *DISCOVERY_DIAMETER = 2*

Dans notre scénario on suppose qu'il y a Cinq centres médicaux dans les environs, Le tableau suivant représente les informations concernant les services Web offerts, prélevés à partir des descriptions sémantique OWL-S de ces services.

Nom du centre médical	Désignation	Nom du service	Inputs	outputs
Hôpital	H	S1	Nom; Prénom; N° Immatriculation	Maladie
Centre de Pathologies Cardiovasculaires	CPC	S2	Nom; Prénom; N° Immatriculation	Chronique
Clinique de Diabétologie	CD	S3	Nom; Prénom; N° immatriculation	Diabetes
Clinique Pharmaceutique	CP	S4	Médicaments	Quantité
Centre Médical	CM	S5	Nom; Prénom; N° immatriculation	Maladie

Tableau 2. Les services web offerts.

La figure 19 illustre l'exécution de notre approche proposée selon le scénario ci-dessus.

La partie (1) de la figure 19 illustre le résultat de processus de diffusion du paquet RREQSSD dans le réseau. Comme illustré, les fournisseurs qui correspondent au diamètre de découverte ($DD = 2$) sont les nœuds [H, CPC, CP et CM].

Ensuite, comme illustré dans la partie (2), les fournisseurs [H, CPC et CM] envoient des paquets de réponse RREPSSD avec un ML (niveau de correspondance) égal à [2,1 et 2] respectivement, (2: signifie que le degré de correspondance = inclusive et 1: signifie que le degré de correspondance = exacte); le fournisseur CP n'a pas envoyé de paquet de réponse parce qu'il n'a détecté aucun degré de correspondance, ce qui signifie un cas d'échec. Les résultats du processus de matching sont illustrés dans le tableau 3

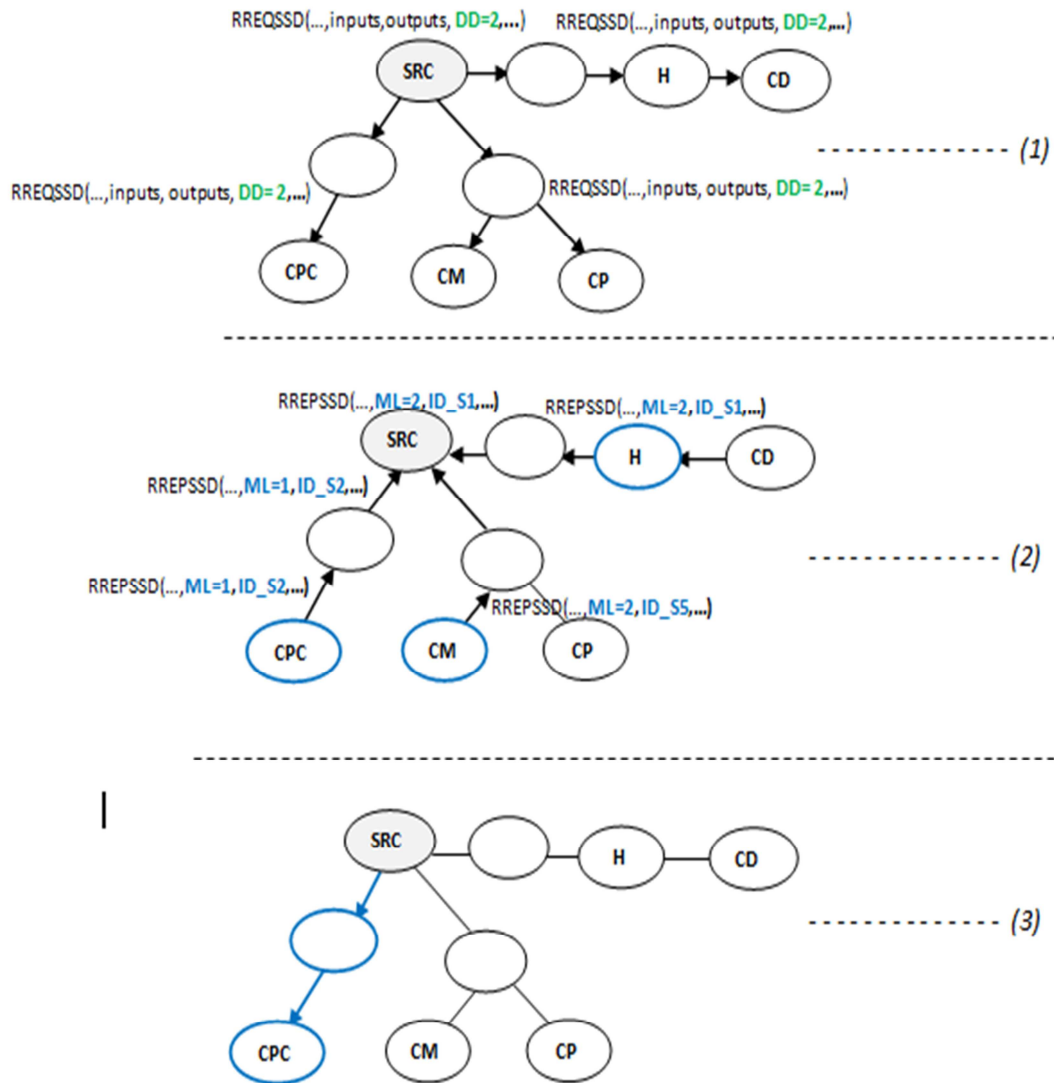


Figure 19. Processus de routage et de découverte des services web.

Matching(Requête ,Service)	Service Provider	Output_Requête	Output_Service	ML
(REQ, S1)	H	Hypertension	Maladie	Inclusive
(REQ, S2)	CPC	Hypertension	Chronique	Exact
(REQ, S4)	CP	Hypertension	Quantité	Echec
(REQ, S5)	CM	Hypertension	Maladie	Inclusive

Tableau 3. Résultat de matching.

Les degrés de correspondances (ML) dans notre étude de cas sont calculés à partir de l'ontologie de domaine suivante.

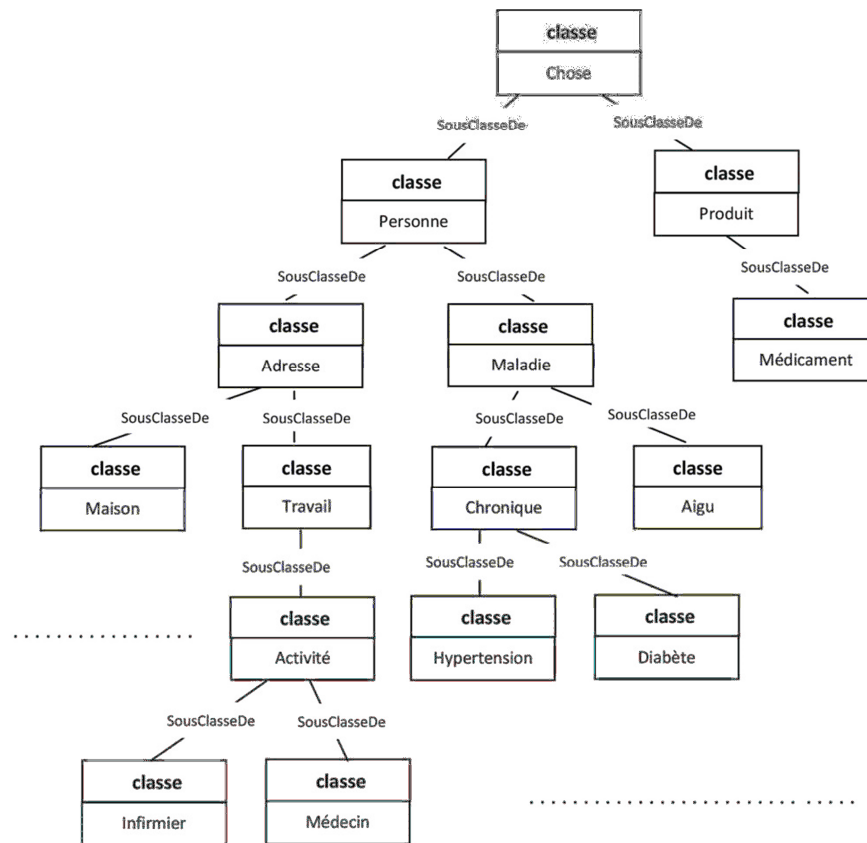


Figure 20. Ontologie globale utilisée.

Enfin, comme l'illustre la figure 19 dans la partie (3), le protocole AODV modifiée au niveau du nœud client, va sélectionner automatiquement la route vers le fournisseur CPC parce qu'il a la meilleure correspondance.

III.3.6 Validation de l'approche proposée

Pour valider notre approche en fonction des défis souhaités présentés dans la partie d'introduction, nous avons utilisé la formule (6) pour comparer notre approche avec d'autres travaux existants et ce en regard du coût de l'énergie totale consommée dans le réseau au cours du processus de découverte.

Le tableau suivant, représente les défis de l'approche proposée par rapport aux contraintes C_i proposés plus haut.

	BANDE PASSANTE				RESSOURCES			MECANISME DE DECOUVERTE		ENERGIE
	Paquets de publication	Packets de synchronisation	Paquets de découverte	Paquets de réponse	Réplication des informations	Table de routage	Stockage de services	Elements clés de découverte	Recherche et Sélection	Energie consommé
APPROCHE PROPOSEE	non	non	broadcast	unicast	non	Route vers fournisseur sélectionné et seulement durant la période d'activité	Non	Paramètres Fonctionnels (inputs/outputs) 'précise'	Automatique	basse

Tableau 4. Défi de l'approche proposée.

Le graphe dans la figure 21 montre le positionnement de notre approche de découverte de service web proposée par rapport à d'autres approches (KONARK, SEDIRAN, SWRDS, TrustSD et TinySDP) selon le coût de l'énergie calculée par la formule (6) proposée.

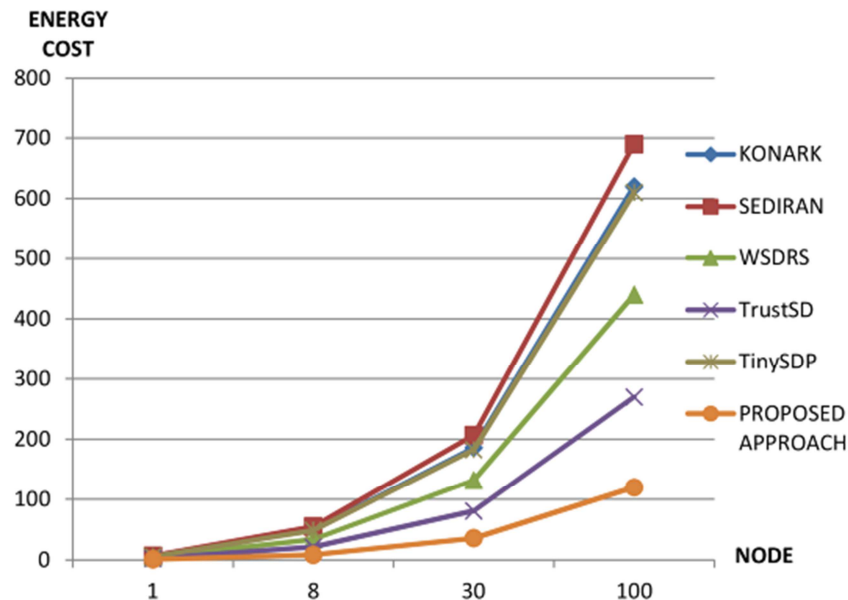


Figure 21. Défis de l'approche proposée en regard du coût.

III.4 Conclusion

Dans ce chapitre nous avons proposé des formules pour calculer le coût de l'énergie consommée dans un environnement ad-hoc par rapport aux approches ou protocoles de découverte des services Web et ceci en se basant sur des paramètres C_i que nous avons déduit par nos études comparatives entre différents approches de découverte existantes. Cette formule permettra ainsi d'évaluer n'importe quel protocole de découverte des services web par rapport à l'énergie consommée et de décider de son applicabilité ou non. Nous avons aussi proposé un nouveau protocole de découverte des services web dans un réseau ad-hoc, ce protocole, présente les caractéristiques suivantes:

- Il est basé sur une architecture décentralisée: chaque nœud dans le réseau gère ses propres services et ne dépend pas d'un annuaire central.
- Un routage basé sur la diffusion des paramètres fonctionnels des services tel les inputs/outputs.

- la découverte des services web est effectuée via les paramètres fonctionnels des services tel les inputs/outputs au lieu de leur identifiants.
- Optimisation de la bande passante au niveau des messages échangés dans le réseau.
- Réduction importante en matière de l'énergie consommée dans le processus de découverte.

Enfin pour valider notre approche nous avons utilisé les formules que nous avons proposées, pour calculer le coût d'énergie consommée, nous avons remarqué une importante réduction en matière de l'énergie dans l'approche proposée par rapport aux autres approches étudiées.

Chapitre IV :

Simulations et résultats

IV.1 Introduction

Afin d'évaluer et valider l'approche que nous avons proposé dans le chapitre précédent nous avons étudié plusieurs outils de simulation et on a fini par choisir NS2 (Network Simulator 2). Dans ce chapitre nous allons présenter les différents outils de simulations ensuite nous allons illustrer et discuter nos expérimentations et les résultats obtenu.

IV.2 Les différents simulateurs réseau

Il existe plusieurs simulateurs de modèles de nœuds ou de réseaux virtuels. Nous allons présenter et étudier les plus connu et les plus utilisés.

IV.2.1 Network Simulator NS2

Network Simulator (NS2) est un simulateur à événements discrets orienté objet, écrit en C++ avec une interface qui utilise le langage OTcl (Object Tool Command Langage). A travers ces deux langages il est possible de modéliser tout type de réseau et de décrire les conditions de simulation : La topologie réseau, le type du trafic qui circule, les protocoles utilisés, les communications qui ont lieu....etc. Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation. Quant au langage OTcl, il fournit un moyen flexible et puissant de contrôle de la simulation comme le déclenchement d'événements, la configuration du réseau, la collecte de statistiques, etc. NS2 ne possède nativement aucune interface graphique. Ainsi, toutes les simulations sont réalisées en ligne de commande. Par ailleurs, grâce à l'extension ".nam" (Network Animator), on peut visualiser les résultats d'une simulation une fois achevée.

L'un des modèles de nœuds mobile sous NS2 est présenté dans la figure 22. Les utilisateurs peuvent proposer des améliorations sur chacun des modules qui composent le modèle de nœuds dans la figure 22. Une nouvelle version du simulateur Network Simulator, nommée NS3 a été développée depuis 2014.

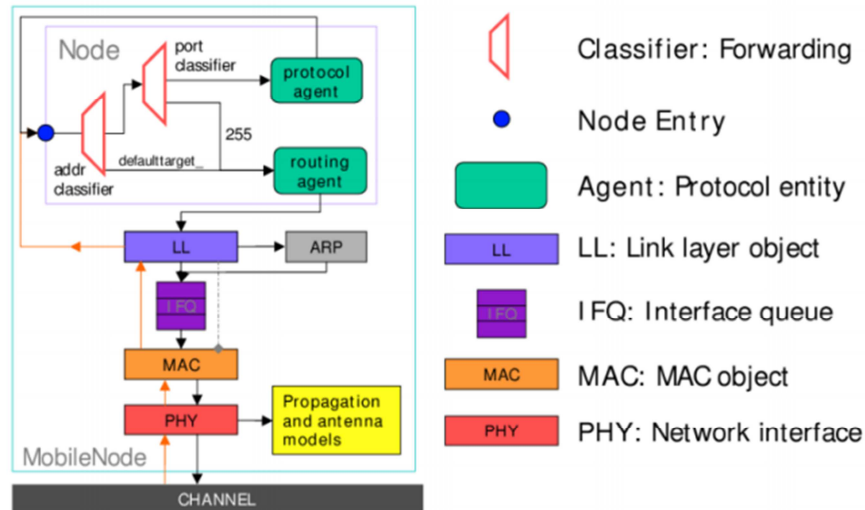


Figure 22. Modèle de nœuds mobile sous NS2 [JOSE 12].

IV.2.2 OMNET++

OMNET ++ est un environnement de simulation open source. Ce simulateur offre une interface graphique solide, et un noyau de simulation intégré. Il a principalement pour but de simuler des communications réseaux mais aussi des systèmes des technologies de l'information. En effet, grâce à son architecture de base flexible, il est capable de simuler des architectures matérielles. C'est ainsi que cette plateforme est devenue connue non seulement au sein de la communauté scientifique mais aussi dans le monde industriel. Et c'est grâce à cette architecture modulaire qu'il est plus facile d'y implémenter de nouveaux protocoles. Le simulateur OMNET fournit à la fois l'estimation de l'énergie consommée dans l'unité de communication et dans l'unité de traitement. La configuration de chaque nœud se fait par une interface graphique comme la montre la figure 23.

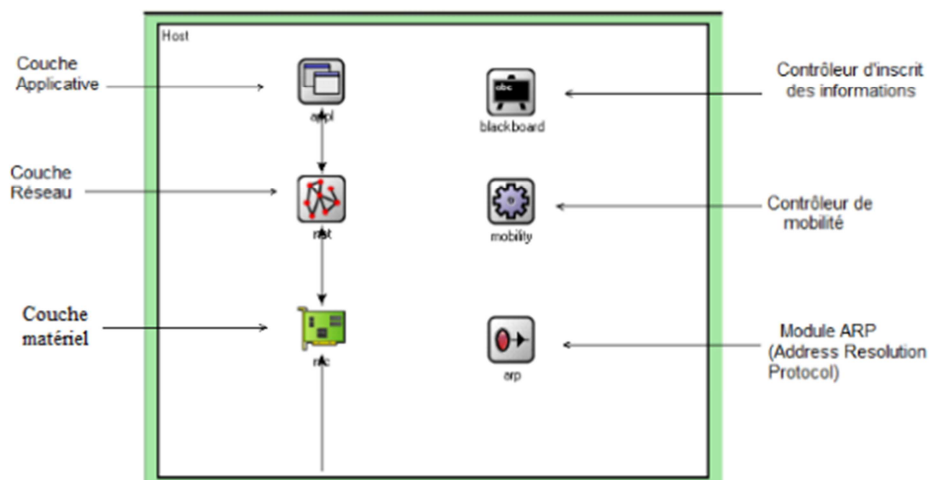


Figure 23. Interface de configuration du modèle de nœud dans OMNET++.

L'architecture du modèle de nœud dans OMNET est montrée par la figure 24:

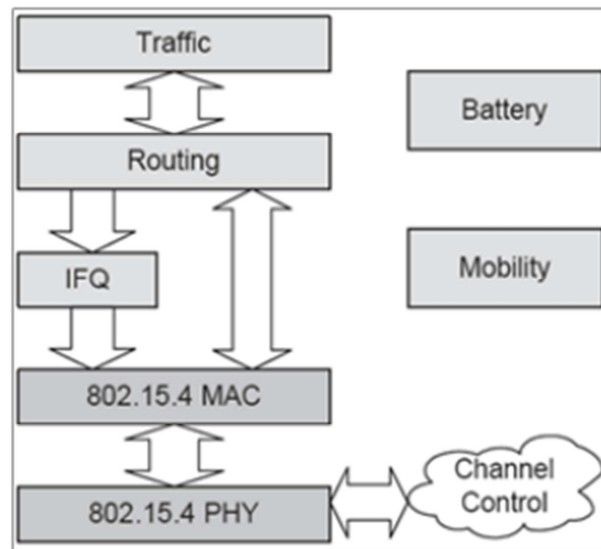


Figure 24. Architecture d'un nœud mobile sur OMNET++ [NITI 11].

Dans OMNET++, l'architecture d'un modèle est hiérarchique, et composée de modules. Il peut y avoir un module simple ou un module composé. Un module simple correspond à un fichier .cc et un fichier .h. Un module composé comprend de simples modules ou d'autres modules composés connectés entre eux. Les paramètres, les sous modules et les ports de chaque module sont spécifiés dans un fichier .ned. Ainsi, l'utilisateur peut intervenir sur chacun des modules [AINA 15].

IV.2.3 OPNET

Le simulateur OPNET [MEEN 14] est développé avec le langage C++. C'est un simulateur de réseau basé sur une interface graphique intuitive. Sa prise en main et son utilisation est relativement aisée. Le simulateur OPNET dispose de trois niveaux de domaine hiérarchique, respectivement de niveau plus haut au niveau plus bas : network domain, node domain et process domain. La simulation sous OPNET fournit en standard une liste d'implémentations de routeurs, de stations de travail, de switches, etc.

L'utilisateur peut construire un modèle de réseau en utilisant les modèles de nœuds prédéfinis et fournis par la librairie d'OPNET. Autrement, l'utilisateur peut définir lui-même le modèle des liens, le modèle de nœud en tant que routeur ou hôte, etc. Ce simulateur n'est gratuit que pour les Universités [AINA 15].

IV.2.4 GloMoSim

GloMoSim [ZENG 98] permet la simulation d'environnement à grande échelle pour des réseaux sans fil et filaires. Il a été développé selon les capacités du langage Parsec. Il est capable de simuler un réseau purement sans fil, avec tous les protocoles de routage que cela implique (AODV, DSR, algorithme de Bellman-Ford (routage par vecteur de distance), ODMRP, WRP, FSR, ...). A l'avenir, des nouvelles versions pourront simuler à la fois un réseau filaire et un réseau hybride (filaire et sans fil). La plupart des systèmes réseaux de GloMoSim sont construits en utilisant une approche basée sur l'architecture à sept couches du modèle OSI.

De plus, l'intégration de modules supplémentaires ne nécessite pas la compréhension du fonctionnement du noyau. L'utilisation précise de Parsec s'avère suffisante. GloMoSim est un simulateur caractérisé par une grande portabilité (Sun Solaris, Linux, Windows) et sa licence est gratuite pour les universitaires. Cependant, le simulateur GloMoSim ne modélise pas la consommation d'énergie au niveau des nœuds [AINA 15].

IV.2.5 J-Sim

Le simulateur J-Sim [SOBE 06] permet de simuler des réseaux de l'ordre de 1000 nœuds. Le simulateur utilise deux langages, Java et TCL, quasi indifféremment. L'architecture et le code sont suffisamment bien structurés pour une prise en main relativement rapide. De plus il permet d'utiliser n'importe quelle application Java en tant que générateur de trafic. Par ailleurs, 802.11 est la seule norme modélisée dans J-Sim, notamment au niveau couche MAC. De plus, il est difficilement maniable [AINA 15].

IV.2.6 SENS

Le simulateur SENS [SUND 04] considère la consommation d'énergie liée à l'acquisition des données environnementales. Le modèle d'environnement, avec lequel les nœuds capteurs interagissent, est fortement considéré dans ce simulateur. Le modèle de nœud dans SENS est montré par la figure 25.

Les protocoles MAC et routage ne sont pas modifiables dans le simulateur SENS. De plus, les applications que l'on peut simuler sont limitées.

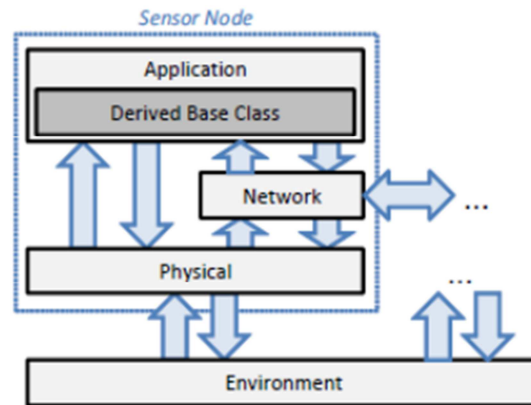


Figure 25. Modèle d'un nœud sur SENS [SUND 04].

IV.2.7 PAWiS

PAWiS [WEBE 08] est un environnement de modélisation pour le simulateur OMNET++. Il modélise la puissance consommée par les unités de traitement sur chaque nœud. Le canal de communication est également modélisé de façon à pouvoir prendre en compte les bruits, les interférences et la perte de puissance de signal. Plusieurs méthodes d'accès au canal sont également fournies avec PAWiS. En outre, PAWiS n'est qu'un environnement de modélisation et son utilisation doit être maintenue dans le simulateur OMNET++ [AINA 15].

IV.2.8 SENSE

Le simulateur SENSE [CHEN 05] a été développé pour simuler des réseaux de grande taille. Il permet de simuler la consommation des différents composants du nœud. Il intègre deux modèles de la couche MAC. Par ailleurs, ce simulateur est difficile à manipuler. De plus, les modèles de propagation radio et de capture de données environnementales ne sont pas implémentés dans ce simulateur [AINA 15].

IV.2.9 IDEA1

Un autre simulateur "IDEA1" est proposé dans [NAVA 11], il permet l'exploration de l'espace de conception. L'architecture des nœuds dans IDEA1 est définie avec un bas niveau d'abstraction comme celle dans NS3 qui est une extension de NS2.

D'après l'architecture du modèle du nœud dans IDEA1 montrée par la figure 26, l'estimation d'énergie est basée sur les différents modules matériels implémentés dans le modèle de nœud.

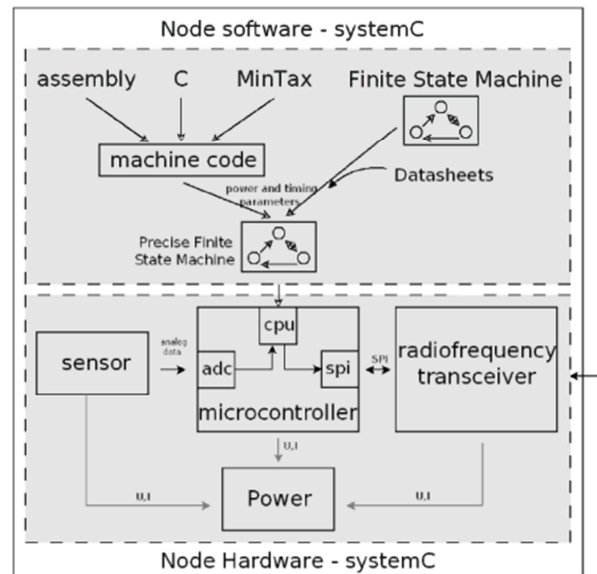


Figure 26. Modèle du nœud dans IDEA1 [NAVA 11].

IV.3 Implémentation

Après avoir étudié les environnements de simulation réseaux les plus utilisés nous avons choisi le simulateur NS2 afin d'implémenter et analyser notre protocole.

Pour cela nous avons modifié l'architecture des nœuds illustrée dans la figure 22 afin d'implémenter notre approche inter-couche de découverte sémantique des services web. La nouvelle architecture de nœud implémenter est présenté dans la figure suivante :

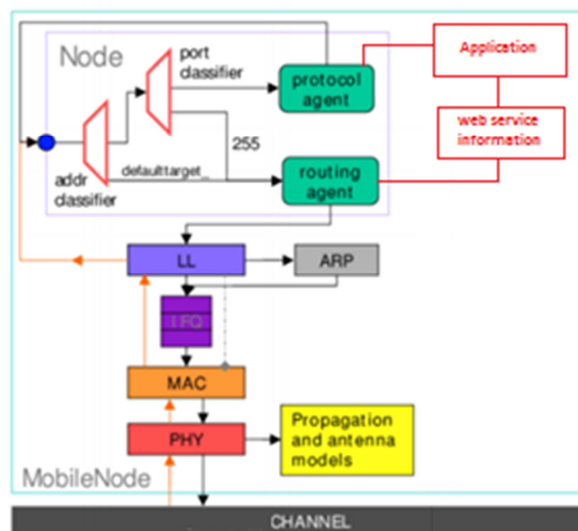


Figure 27. L'architecture de nœud proposée.

Comme illustré dans la figure précédente nous avons modifié l'architecture existante d'un nœud mobile sous NS2 afin de permettre la communication entre la couche application et l'agent de routage où ce dernier récupérera les données sémantiques concernant le service souhaité à partir de la couche application avant d'entamer le processus de découverte de route, ses données sémantiques correspondent aux inputs/outputs du service web à rechercher.

Au niveau de la couche réseau (routage) comme présenté précédemment nous avons choisi de modifier le protocole AODV afin d'implémenter notre approche. Les principaux fichiers modifiés dans la plateforme NS2 afin d'implémenter notre protocole sont les suivants :

```
~/ns-allinone-2.35/ns-2.35/aodv/aodv.cc
~/ns-allinone-2.35/ns-2.35/aodv/aodv.h
~/ns-allinone-2.35/ns-2.35/aodv/aodv_packet.h
~/ns-allinone-2.35/ns-2.35/aodv/aodv_rtable.cc
~/ns-allinone-2.35/ns-2.35/aodv/aodv_rtable.h
~/ns-allinone-2.35/ns-2.35/common/mobilenode.cc
~/ns-allinone-2.35/ns-2.35/common/mobilenode.h
```

IV.3.1 Mesure des performances

Tout d'abord pour analyser en détails nos résultats d'expérimentations et mesurer les performances de notre protocole, nous définissons d'abord quelques paramètres.

On note :

- F : Tous les fournisseurs disponibles dans le réseau.
- F_p : Les fournisseurs probables par rapport à la requête du client, défini par :

$$\{F_p \in F \mid \exists F_i \in F (F_p = F_i \text{ et } F_i \text{ satisfait } REQ)\}$$
- T_{disc} : Le temps maximum de réception des réponses REP, défini par :

$$T_{disc} = \max(t_1, t_2, \dots, t_n)$$
 Où t_i : temps de réponse de chaque fournisseur probable F_p .

- $N_{req}(T_i)$: Le nombre de requêtes RREQSSD a un instant T_i .
- $E_{req}(T_i)$: L'énergie consommée par les requêtes de découverte a un instant T_i .
- $N_{rep}(T_i)$: Le nombre de paquets RREPSSD a un instant T_i .
- $E_{rep}(T_i)$: L'énergie consommée par les requêtes de réponses a un instant T_i .
- $N_{total}(T_i)$: Le nombre total des requêtes impliqué dans le processus de découverte à un instant T_i , défini par :

$$N_{total}(T_i) = N_{req}(T_i) + N_{rep}(T_i)$$

Afin d'évaluer les performances de notre nouveau protocole nous allons étudier deux scenarios, le premier scenario illustre un exemple d'application du protocole proposé, et un deuxième scenario afin d'évaluer les performances du protocole proposé par rapport à un ensemble de facteurs qui influencent la consommation d'énergie.

Nous avons aussi utilisé quelque outils pour calculer, analyser et dessiner les graphes de mesures souhaité, ces outils sont : AWK et Gnuplot

- *AWK* : Le nom awk vient des initiales de ses concepteurs: Alfred V. Aho, Peter J. Weinberger, et Brian W. Kernighan. La version originale d'awk a été écrite en 1977 dans les laboratoires d'AT & T Bell. En 1985, une nouvelle version a rendu le langage de programmation plus puissant, introduisant la définition des fonctions par l'utilisateur, l'utilisation de multiple flux d'entrée et les expressions régulières calculées [ARNO 97]. disponible sur la plupart des systèmes Unix et sous Windows avec Cygwin ou Gawk. Il est principalement utilisé pour la manipulation de fichiers textuels pour des opérations de recherches, de remplacement et de transformations complexes.
- *Gnuplot* : est un utilitaire qui sert à produire des représentations graphique, basé sur ligne de commande, il fonctionne sous Linux, OS/2, MS Windows, OSX, VMS, et de nombreuses autres plates-formes. Il est distribué sous une licence de logiciel libre. Il a été créé à l'origine pour permettre aux scientifiques et aux étudiants de visualiser les fonctions et les données mathématiques de manière interactive, mais il a évolué pour supporter de nombreuses utilisations non-interactives telles que des scripts web. Il est également utilisé comme un moteur de

traçage par des applications tierces comme Octave. Gnuplot a été utilisé et en cours de développement depuis 1986. [GNUP]

IV.3.1.1 Premier scenario (Exemple d'application)

IV.3.1.1.1 Présentation

Pour le premier scenario nous avons implémenté le scenario présenté dans le chapitre précédent mais sans utilisation du paramètre DD (Discovery_Diameter) c'est-à-dire, nous avons supposé un scenario de désastre (séisme), où est déployé un réseau ad hoc d'urgence et afin de porter secours et donner les premiers soins aux blessés avant de les emmener à l'hôpital, les ambulanciers doivent d'abord vérifier des informations médicales concernant le patient c.-à-d. vérifier s'il est atteint d'une maladie chronique par exemple ou s'il est sensible à telle ou telle injection afin de pouvoir réagir de la manière la plus sûre est rapide possible. Dans ce cas les ambulanciers vont chercher un service web offert par un hôpital ou un centre médical spécifique qui permet l'accès à une base de données médicale qui comporte toutes les informations médicales nécessaires sur les blessés. Dans notre étude de cas par exemple on suppose que les informations sur le service recherché sont:

Paramètre	Valeur
Nom du service	Maladie_Cardiaque
Inputs	Nom; Prénom; N° Immatriculation
Outputs	Hypertension

Tableau 5. Informations concernant le service web recherché.

Dans notre scenario on suppose qu'il y a cinq fournisseurs F avec trois fournisseurs probables F_p de services médicaux dans les environs. Le tableau 6 suivant représente les informations concernant les services Web offerts.

Nom du Fournisseur	Désignation	Nom du service	Inputs	outputs
Hôpital	H	S1	Nom; Prénom; N° Immatriculation	Maladie
Centre de Pathologies Cardiovasculaires	CPC	S2	Nom; Prénom; N° Immatriculation	Chronique
Clinique de Diabétologie	CD	S3	Nom; Prénom; N° immatriculation	Diabète
Centre Pharmaceutique	CP	S4	Médicaments	Quantité
Centre Médical	CM	S5	Nom; Prénom; N° Identifiant	Maladie

Tableau 6. Les services web offerts.

L'ontologie de domaine utilisé dans ce scénario pour le matching est illustrée dans la figure 28 suivante

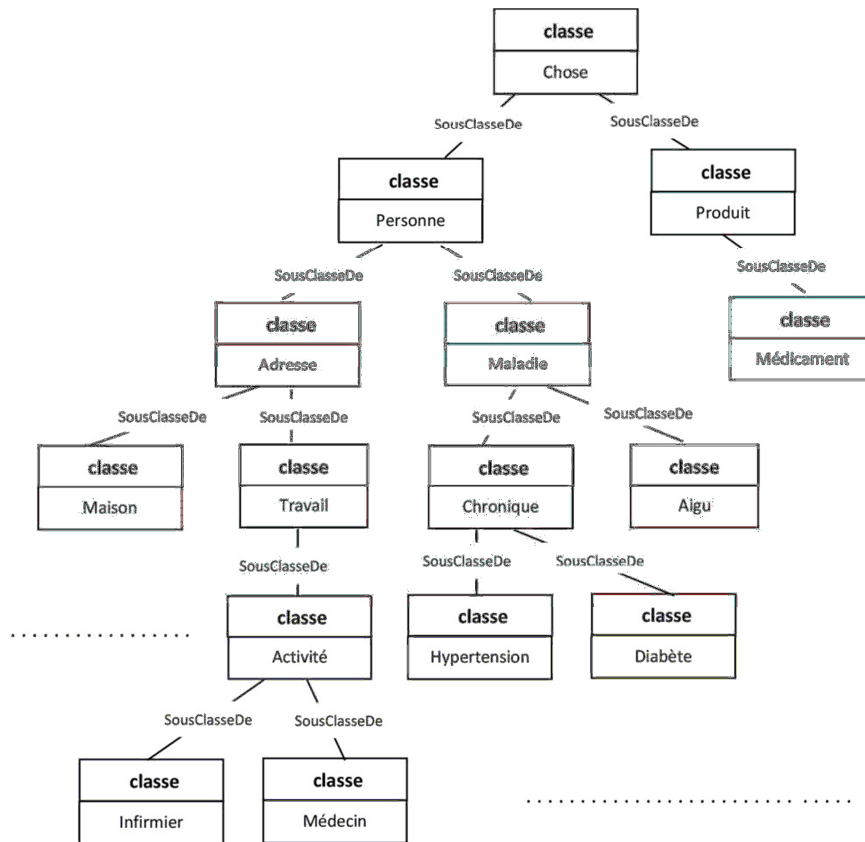


Figure 28. Ontologie du domaine médicale.

IV.3.1.1.2 Simulation et évaluation avec NS2

La configuration détaillée du premier scénario sous NS2 est présentée dans le tableau 7 suivant :

Paramètre	Valeur
Simulateur	NS2.35
Temps de simulation	1 secondes (1)
Nombre de nœuds	30
Terrain de simulation	850*850 mètres (m)
Nombre de clients	1
Nombre de fournisseurs	5
Nombre de fournisseurs probable	3
Modèle de propagation	Tworayground
Energie du nœud	3.4 joules (j)
Energie consommée lors de transmission	0.33 joules (j)
Energie consommée lors de réception	0.1 joules (j)
Energie consommée dans le repos	0.05 joules (j)
Energie consommée en mode veille	0.03 joules (j)
Mobilité	100 mètres/secondes (m/s)

Tableau 7. Scénario à 30 nœuds.

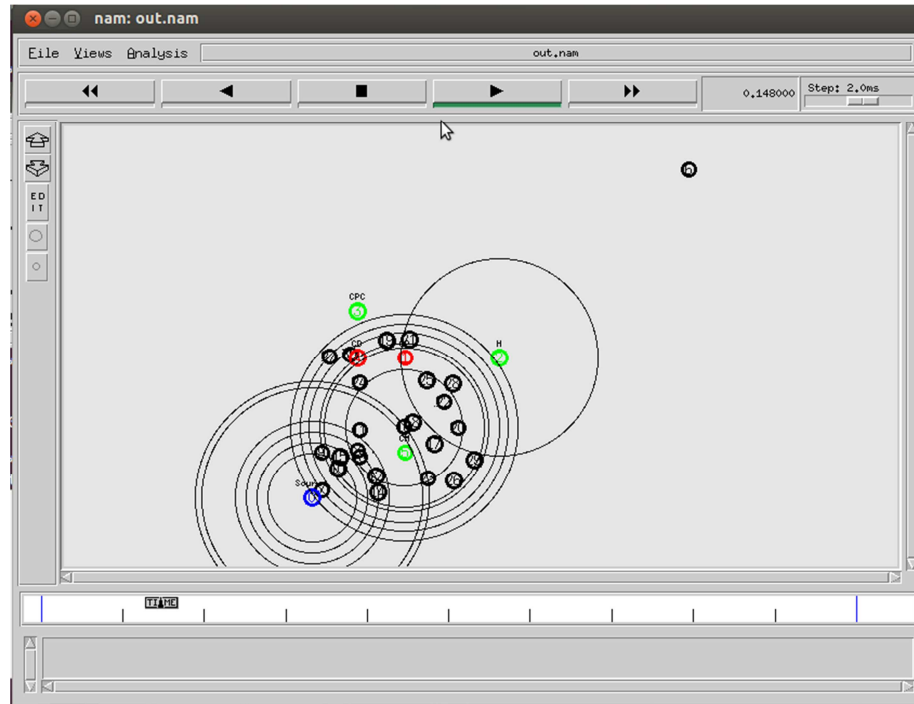


Figure 29. Le premier scenario sous NAM.

La figure 29 représente le résultat d'exécution du premier scenario sous NAM après le lancement d'exécution du script Tcl avec la configuration présente dans le tableau 7. Comme illustré, le client est représenté en bleu, les fournisseurs probables F_p sont représentés en vert, et les fournisseurs en rouge ce sont les fournisseurs qui ont reconnu un Echec lors du matching entre la requête du client et les descriptions des services qu'ils offrent. Sachant que dans le script de scenario que nous avons utilisé, nous avons supposé qu'un Client mobile représenté en bleu avec le Label « source » lance une recherche de son service à l'instant 0.10 secondes et débute l'invocation et la transmission des données en utilisant un flux de données FTP avec agent de transport TCP avec acquittement dès que le service est découvert après un temps T_{disc} .

Comme présenté dans la figure 29 les nœuds fournisseurs de services qui correspondent à la requête du Client sont l'hôpital (H), le Centre de pathologie Cardiovasculaire (CPC) et le Centre Médical (CM) représenté en vert au final c'est le CPC qui va être sélectionné par rapport au niveau de correspondance comme illustré dans le tableau 8.

Matching (Requête, Service)	Fournisseur de Service	Output_Requête	Output_Service	Degré de Matching
(REQ, S1)	H	Hypertension	Maladie	Inclusive
(REQ, S2)	CPC	Hypertension	Chronique	Exacte
(REQ, S3)	CD	Hypertension	Diabète	Echec
(REQ, S4)	CP	Hypertension	Quantité	Echec
(REQ, S5)	CM	Hypertension	Maladie	Inclusive

Tableau 8. Résultats de Matching.

IV.3.1.1.3 Impact de nombre des Inputs/Outputs sur l'énergie globale

Comme illustré dans la figure 30 où nous avons pratiqué une série de tests pour valider notre théorie de l'ajout des informations sémantiques concernant les services web dans des paquets de routage au niveau de la couche réseau, nous avons calculé l'influence du nombre des paramètres fonctionnels sémantique sur l'énergie globale consommée dans le processus de découverte, où nous avons alloué pour chaque information sémantique un Octet pour représenter l'identifiant de chaque paramètres. D'après les résultats obtenus l'ajout des informations sémantiques avec de 2, 5 et 10 octets aux paquets de découverte n'affecte pas d'une manière importante l'énergie globale consommée dans le processus de découverte des services web, où nous remarquons qu'entre l'intervalle du temps [0.10 et 0.14 secondes] qui est équivalente à la période d'exécution de processus de découverte des services disponible. Ceci motive notre approche de l'ajout des informations sémantiques dans le processus de routage.

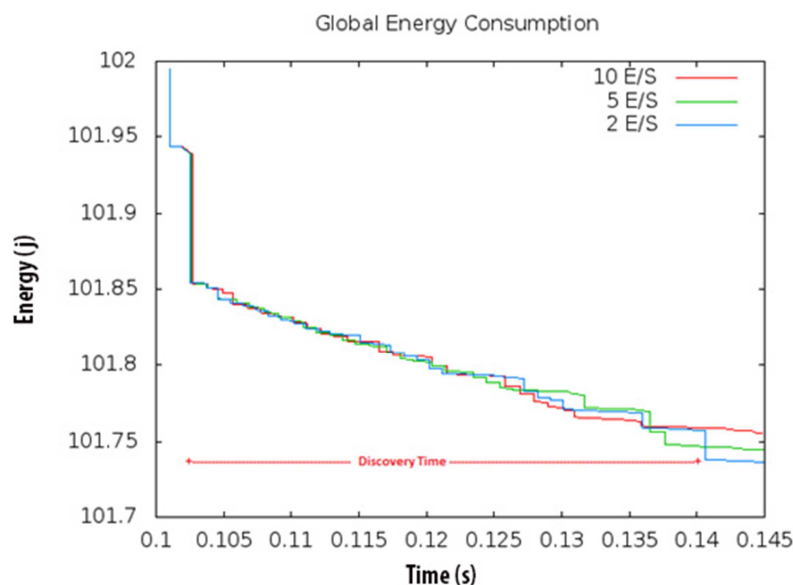


Figure 30. Energie consommée par rapport au nombre d'informations sémantiques dans les paquets de routage.

La petite variation de l'énergie peu importante consommée à partir de l'instant 0.125 s, est expliquée par la mobilité variable des nœuds ce qui influe sur les paquets HELLO de reconnaissance des voisins ainsi que le nombre des voisins à chaque instant t .

IV.3.1.1.4 Les paquets de découverte

La figure 31 représente le nombre des paquets de découverte et réponses des fournisseurs qui circule dans le réseau durant le temps de découverte T_{disc} . La figure 31 est composée de 3 parties, la partie -1- représente l'évolution des paquets de découverte des services RREQSSD par rapport au temps de la simulation. La partie -2- représente l'évolution des paquets de réponses RREPSSD des fournisseurs probable F_p par rapport au temps de la simulation. Et la partie -3- illustre l'évolution des $N_{total}(i)$ durant le temps de la simulation.

Depuis les résultats obtenus nous remarquons un $Max(N_{total}(i))$ de 40 paquets entre l'intervalle de temps [0.125 et 0.13 s], aussi que N_{total} dépend principalement de N_{req} vu le principe de diffusion sans limite des paquet de découverte RREQSSD.

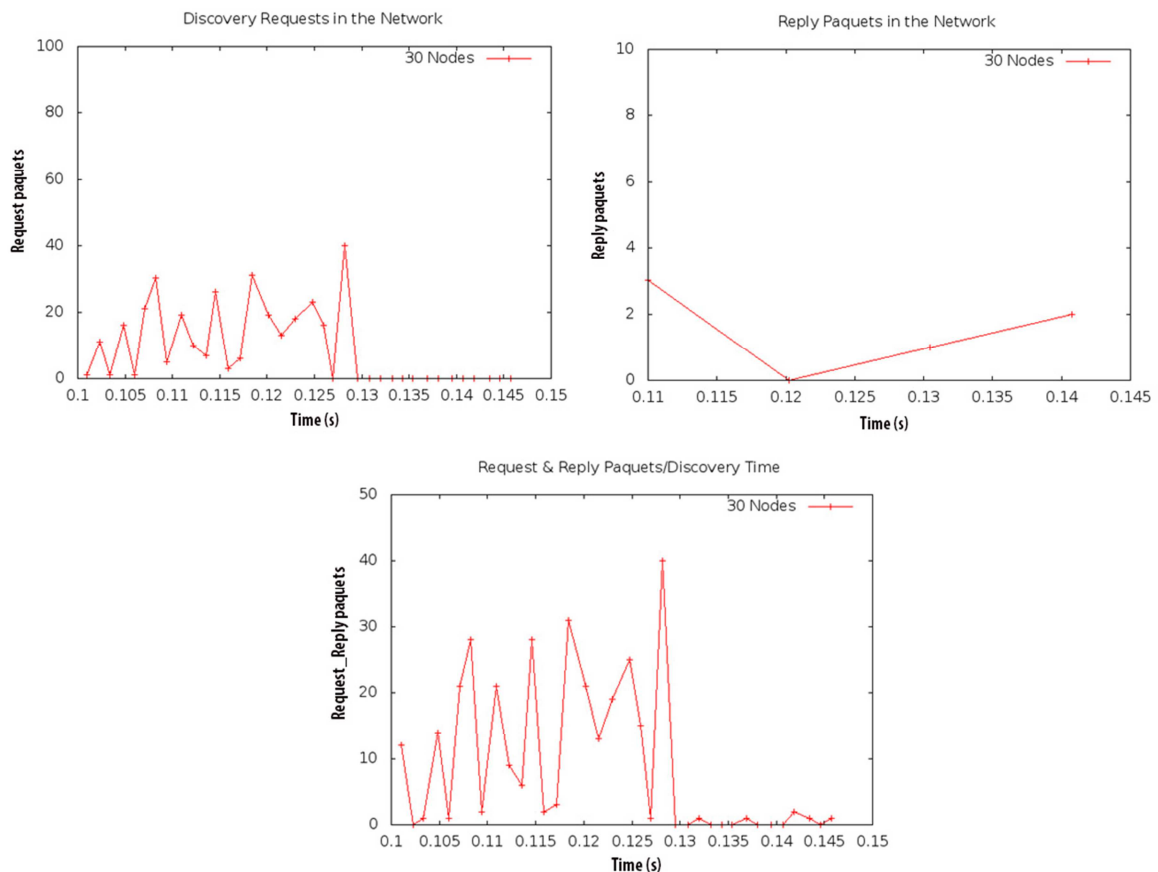


Figure 31. Nombre de paquets de découverte dans T_{disc} .

IV.3.1.2 Deuxième scénario (Expérimentations avec 100 nœuds)

Le tableau ci-après résume les paramètres de notre deuxième expérimentation:

Paramètre	Valeur
Simulateur	NS2.35
Temps de simulation	10 secondes (s)
Nombre de nœuds	100
Terrain de simulation	850*850 mètres (m)
Nombre de clients	5
Nombre de fournisseurs	20
Nombre de fournisseurs probable	10
Nombre des Inputs/Outputs	2
Modèle de propagation	Tworayground
Energie du nœud	3.4 joules (j)
Energie consommée lors de transmission	0.33 joules (j)
Energie consommée lors de réception	0.1 joules (j)
Energie consommée lors dans le repos	0.05 joules (j)
Energie consommée lors en mode veille	0.03 joules (j)
Mobilité	100 mètres/secondes (m/s)

Tableau 9. Paramètres de simulation deuxième scénario.

Pour cette partie d'expérimentation et afin de mieux étudier le comportement de notre protocole par rapport à plusieurs paramètres C_i définis dans le chapitre précédent, nous avons varié les demandes des clients sur la période d'exécution du scénario comme illustré dans la figure suivante.

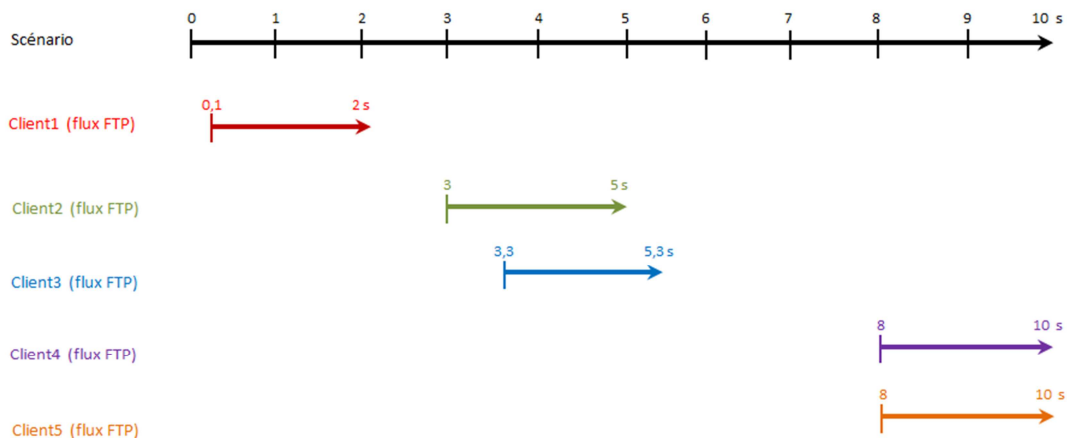


Figure 32. Scénario durant le temps de simulation.

La figure 33 suivante illustre une capture d'écran de l'exécution du scénario proposé sous NAM à l'instant $t=8,1$ s. Les nœuds en bleu représentent les nœuds clients demandeurs de services, en vert les nœuds probables F_p qui répondent aux demandes des

clients et en rouge sont les fournisseurs qui ont reconnu un échec lors de matching avec les demandes des clients.

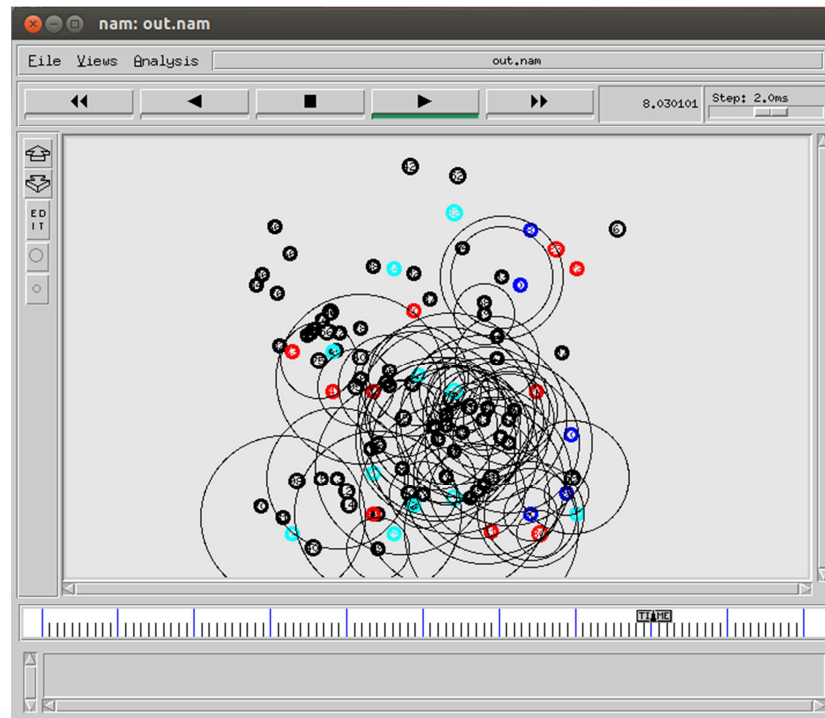


Figure 33. Résultat d'exécution du deuxième scenario sous NAM.

IV.3.1.2.1 L'énergie globale

La figure 34 représente la courbe de consommation de l'énergie. Nous avons utilisé un script AWK afin de parser et calculer la consommation de l'énergie au fil du temps, l'algorithme 7 suivant représente le traitement que nous avons utilisé pour parvenir à dessiner la courbe dans la figure 34, où

N_i : Représente les identificateurs des nœuds participants

$Init_E$: l'énergie courante dans chaque nœud, au début elle contient 3.4 joule.

E_i : l'énergie parsée dans chaque ligne du fichier *Trace*, qui correspond à chaque nœud N_i à l'instant t_i de chaque évènement ou ligne dans le fichier *Trace*;

$Global_E$: représente l'énergie globale dans tout le réseau, initialement elle est égale à 340 joule ($init_E \times nb_nodes = 3.4 \times 100$)

Algorithme calcul de l'énergie dans le temps

Début

Pour (chaque ligne dans le fichier Trace) **faire**

Si ($Init_E(N_i) > E_i$) **alors**

$Global_E = Global_E - (Init_E(N_i) - E_i)$;

$Init_E(N_i) = E_i$;

Fin si

Fin pour

Fin

Algorithme 7. Calcul de consommation d'énergie.

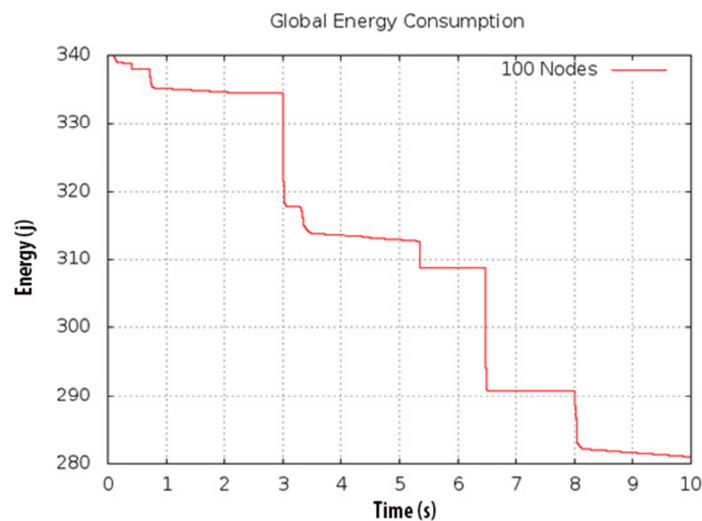


Figure 34. La consommation de l'énergie dans le temps.

Comme illustré dans la figure 34, l'énergie reconnaît une chute importante entre l'intervalle de temps $[3 \text{ s} \rightarrow 8 \text{ s}]$, et ceci est expliqué par la diffusion non limitée des paquets de découverte déclenchés par les clients 1, 2 et 3 ainsi que la transmission de données après découverte de route vers leurs destination.

Afin d'étudier en détail le comportement de notre scénario et expliquer l'évolution de la consommation de l'énergie dans la figure 34, nous avons classifié les types des paquets qui circulent dans le réseau en trois (Données, Paquets de découverte et Paquets de contrôle) ensuite nous avons étudié simultanément chaque type par rapport à l'énergie globale consommée dans le scénario proposé durant la période de simulation. Le résultat est illustré dans la figure 35.

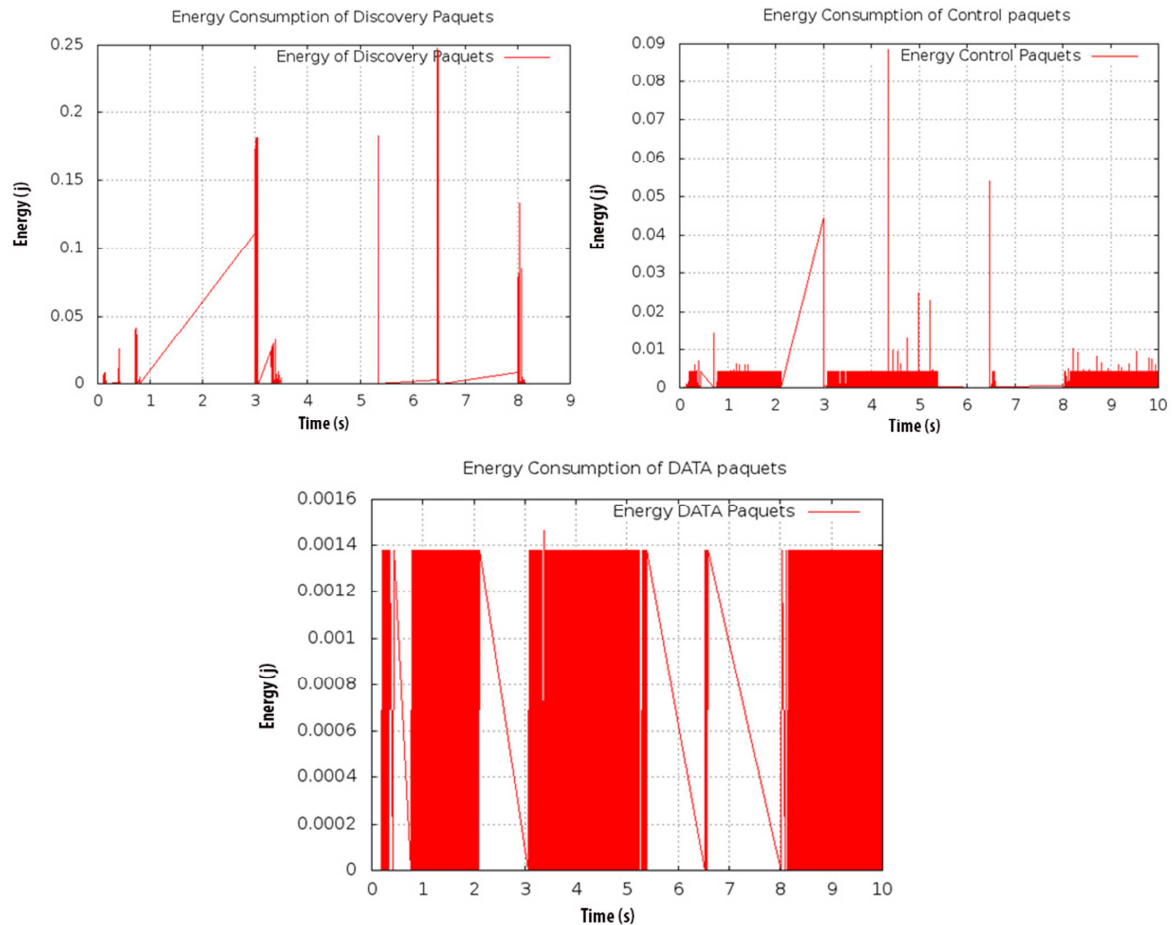


Figure 35. Energie globale consommée par chaque type de paquet.

La partie -1- de la figure 35 représente l'énergie consommée par les paquets de découverte, on remarque une consommation importante à l'instant $t=3s$, qui représente la période où deux clients ont lancé un processus de découverte de service, on remarque aussi une consommation par les paquets de découverte entre l'intervalle $[5.3 s \rightarrow 8 s]$. Pourtant c'est une période neutre par rapport à notre scénario, ceci est expliqué par le processus de diffusion sans limite des paquets de découverte. Enfin la consommation à l'instant $t=8s$ est tout à fait normale, elle représente l'instant où les deux derniers clients ont démarré leurs processus de découverte.

La partie -2- qui représente le taux de la consommation d'énergie au fil du temps par rapport aux paquets de contrôles, l'activité de ces paquets durant tout le scénario est tout à fait normale puisque ce sont des paquets utilisés pour tous type de transmission dans chaque nœud.

Dans la partie -3-, les intervalles $[0 \rightarrow 2]$, $[3 \rightarrow 5.3]$ et $[8 \rightarrow 10]$ illustre clairement la relation entre transmission des données et consommation d'énergie.

IV.3.1.2.2 Quantité de paquets générés dans le réseau

Un autre paramètre a étudié et qui influence sur la quantité de l'énergie consommée et sans aucun doute le nombre des paquets générés sur le réseau, puisque comme c'était présenté dans le chapitre précédent, la consommation d'énergie dépend de principalement de la transmission et réception des paquets par les nœuds. Nous avons calculé le nombre de tous les paquets échangés à travers le réseau par rapport au deuxième scenario, le résultat est représenté dans l'histogramme de la figure 36.

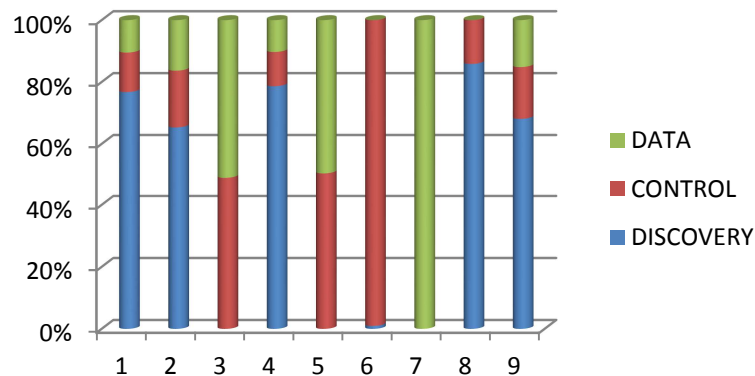


Figure 36. Quantité et types de paquets générés dans le réseau.

La figure 37 suivante représente l'activité détaillée des paquets de découverte de notre nouveau protocole au fil du temps, où la partie -1- représente le nombre de transmissions des paquets de découverte RREQSSD, la partie -2- les transmissions des RREPSSD et la dernière partie la somme de tous les paquets de découverte intervenant durant tout le scenario proposé.

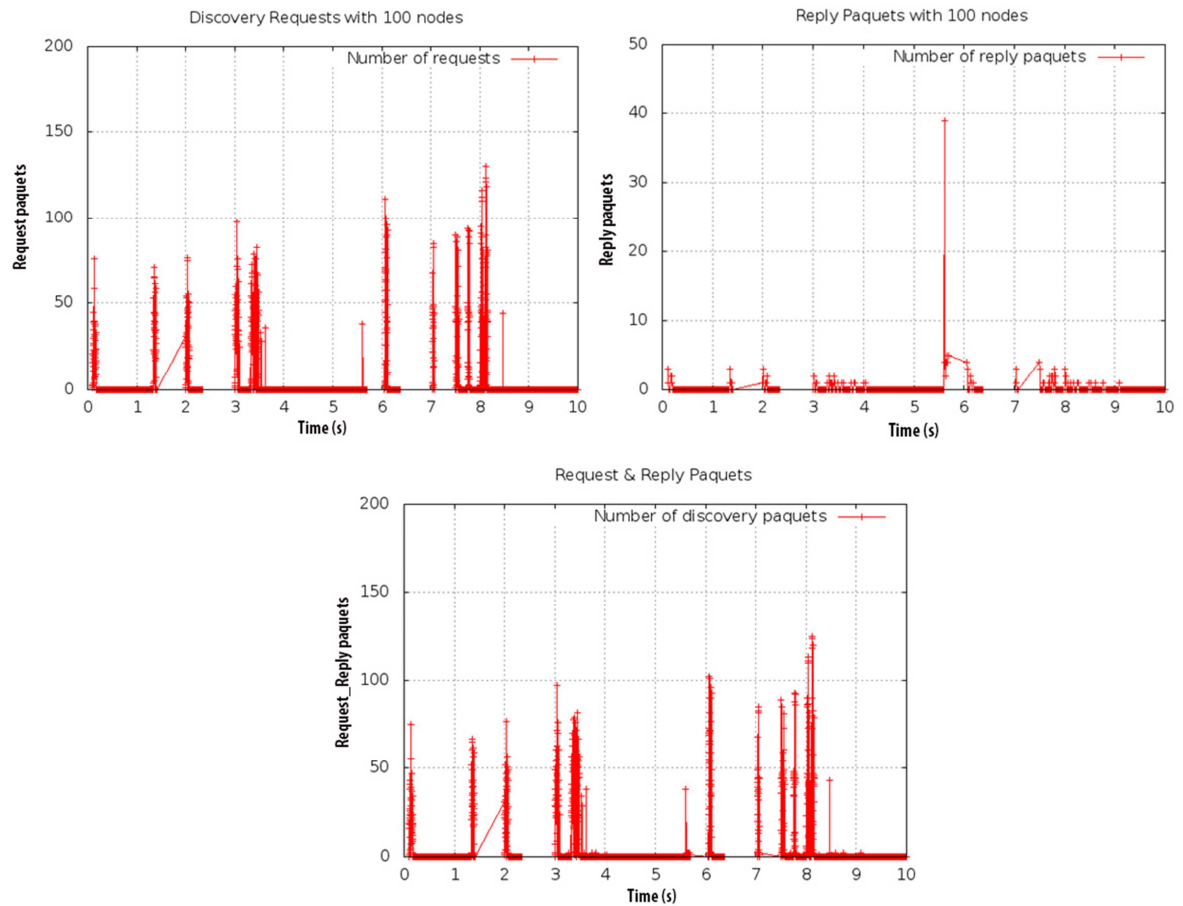


Figure 37. Quantité des paquets générés durant le scénario.

IV.3.1.2.3 La mobilité

Un autre paramètre qui dépend des caractéristiques de réseau et dont nous avons étudié l'impact sur l'énergie consommée par rapport à notre nouveau protocole est la mobilité des nœuds. Pour cela nous avons analysé le même scénario que nous avons proposé pour la deuxième partie de notre étude expérimentale mais en changeant la vitesse de déplacement des nœuds c'est-à-dire nous avons modifié le paramètre de la mobilité $M=20$. Le résultat de comparaison entre les deux expérimentations est illustré dans la figure 38.

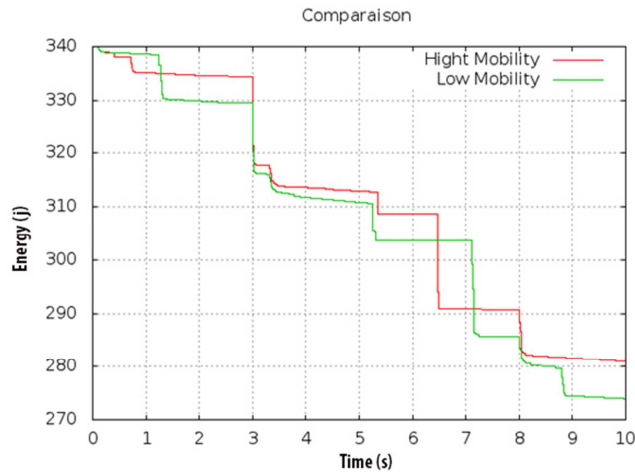


Figure 38. Impact de la mobilité des nœuds sur l'énergie globale consommée.

Nous remarquons que le comportement du réseau par rapport aux deux expérimentations est presque le même, la différence de consommation d'énergie constatée au fil du temps est expliquée par d'autres paramètres qui sont le voisinage et la portée de connexion.

Nous avons aussi réalisé une étude comparative entre le nombre de paquets de découverte générés par notre protocole au fil du temps dans les deux expérimentations, le résultat est illustré dans la figure 39.

La partie -1- de la figure 39 représente une comparaison entre le nombre de paquets de découverte RREQSSD générée dans les deux expérimentations. La partie -2- représente une comparaison entre le nombre de paquets de réponse RREPSSD. Et la partie -3- illustre la comparaison par rapport à la totalité des paquets de découverte générés dans les deux expérimentations où nous remarquons une différence importante est ceci est expliqué par le voisinage et la portée de connexion radio entre les nœuds.

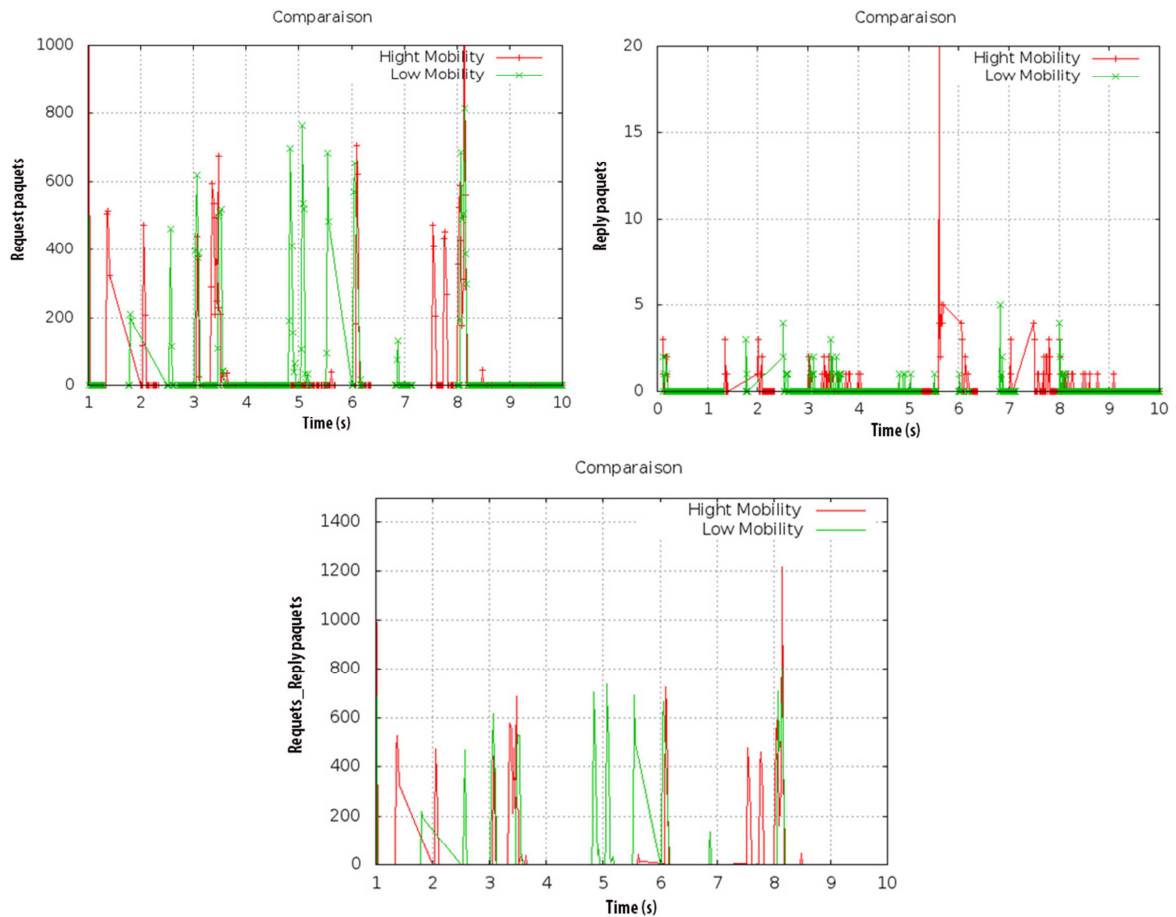


Figure 39. Impact de la mobilité des nœuds sur les paquets de découverte générés.

IV.3.1.2.4 Le débit moyen

Un autre paramètre important qui nous permet d'évaluer notre protocole est le débit moyen dans le réseau. Pour cela nous avons proposé la formule suivante pour cette évaluation, où le débit moyen est calculé par la somme des tailles des paquets de découverte sur le temps moyen de transmission.

$$R = \frac{\sum_{i=1}^{nb_disc} S}{T}$$

Où :

R : Le débit moyen

nb_disc : Le nombre de paquets de découverte

S : La taille du paquet

T : Le temps moyen de transmission des paquets

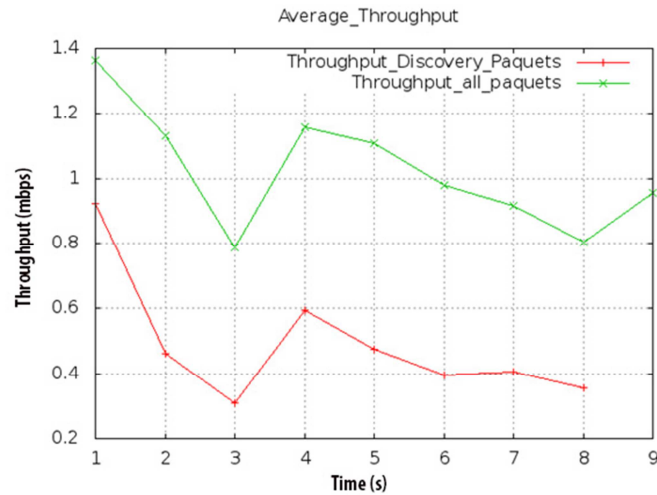


Figure 40. Le débit moyen.

La figure 40 représente le débit moyen des requêtes de découverte dans le réseau (en rouge), par rapport au débit moyen consommé par tous les paquets générés dans notre scénario d'expérimentation (représenté en vert).

IV.3.1.2.5 Diamètre de Découverte (Discovery_Diameter)

Afin de valider l'importance du paramètre Diamètre de Découverte (DD), nous avons réalisés une série d'expérimentations et de comparaison pour montrer l'optimisation apportée par rapport aux paramètres discutés précédemment (Energie, Débit et Nombre de paquets).

La figure 41 suivante illustre une comparaison entre l'étude de la consommation d'énergie en utilisant le paramètre DD durant le temps de la simulation et les résultats obtenu dans la section *-a-*.

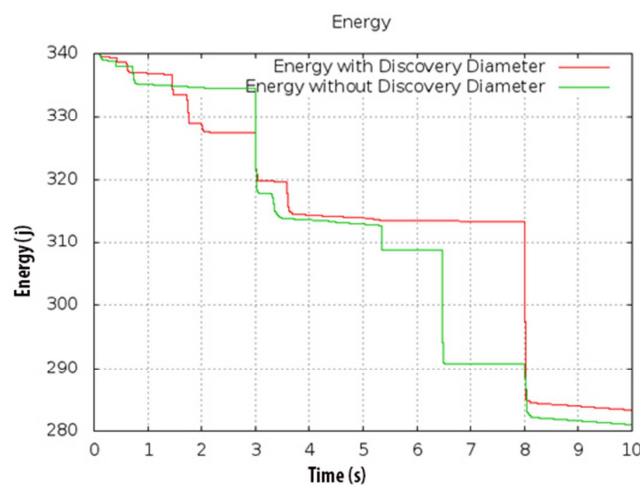


Figure 41. L'énergie consommée en utilisant DD.

Nous remarquons une importante optimisation dans l'énergie due à la limitation du domaine de broadcast. Nous remarquons aussi une logique dans la consommation d'énergie par rapport au scénario proposé où il y a une stabilité dans les intervalles de temps de non activité [2 → 3 s] et [5 → 8 s].

La figure 42 suivante représente notre étude comparative sur l'impact du paramètre DD sur le débit moyen étudié dans la section **-d-**.

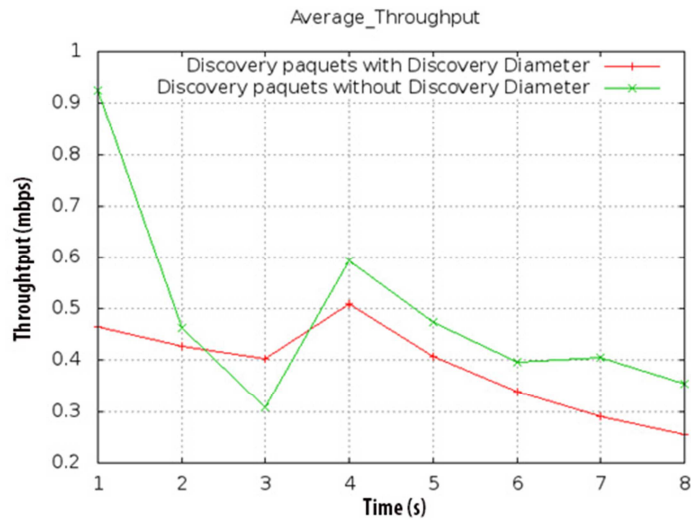


Figure 42. Le débit moyen en utilisant DD.

L'optimisation et la stabilité du débit moyen dans le réseau par rapport aux paquets de découvertes et clairement illustré dans la figure 42, où par exemple dans l'intervalle du temps [0 → 2s] le débit est réduit à la moitié de celui calculé dans l'expérimentation sans utilisation du paramètre Diamètre de Découverte.

La figure 43 représente une comparaison entre le débit calculé par rapport aux paquets de découverte (RREQSSD et RREPSSD) et le débit consommé par tous les paquets générés, ceci nous permet de voir l'impact de Diamètre de Découverte sur la bande passante.

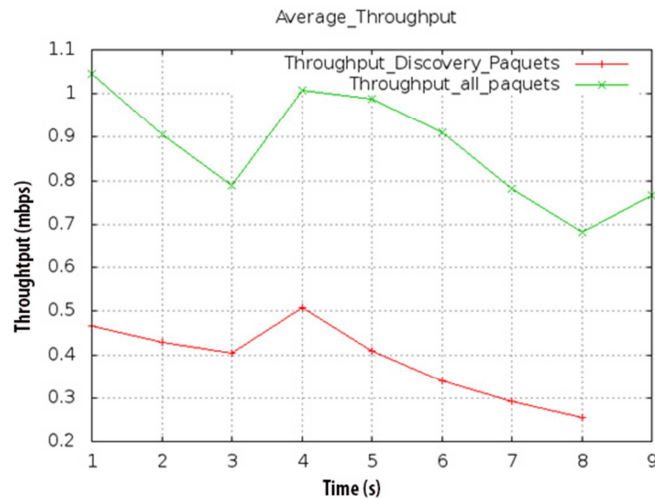


Figure 43. Impact de Diamètre de Découverte sur le débit moyen par rapport aux paquets générés.

Un autre point important dans la comparaison est le calcul du nombre de paquets de découverte générés, puisque ce paramètre comme démontré précédemment a une influence directe sur l'énergie globale consommée.

La figure 44 illustre la comparaison que nous avons effectuée par rapport à la quantité des paquets de découverte générés dans les deux expérimentations c'est-à-dire avec et sans utilisation du Diamètre de Découverte.

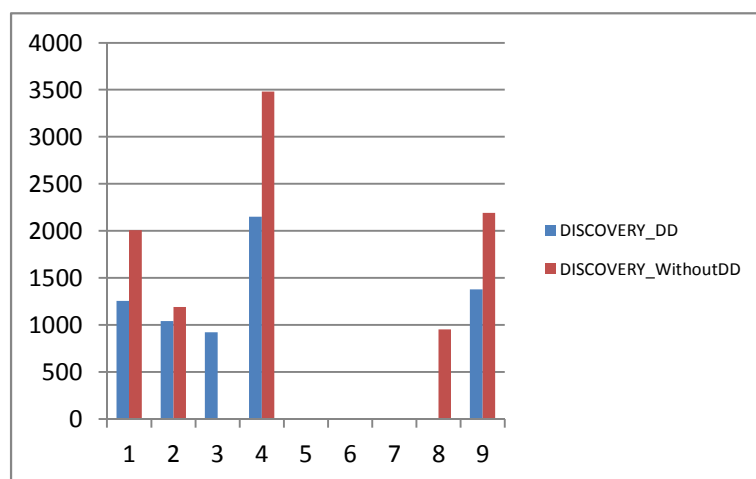


Figure 44. Comparaison entre la quantité de paquets générés avec et sans DD.

Comme illustré l'approche avec utilisation de paramètre DD offre un gain important par rapport au nombre de paquets de découverte générés, ce qui justifie sa proposition pour

l'aboutissement a notre objectif qui est l'optimisation de l'énergie consommée dans le réseau.

IV.4 Conclusion

Dans ce travail, nous avons mis en place un nouveau protocole de découverte de service web basé sur une approche inter-couche et un couplage d'objectifs c'est-à-dire effectuer une découverte de route vers un destinataire et découverte de service web en même temps. Ce nouveau protocole est basé principalement sur la découverte sémantique et la modification du protocole AODV, nous avons étudié ses performance dans le réseau, nous avons aussi validé notre proposition de limitation de champs de recherche par le paramètre `Discovery_Diameter` qui a apporté une optimisation importante en matière d'énergie et bande passante.

Conclusion Générale

1. Conclusion

Comme conclusion finale, nous pouvons dire que l'utilisation des techniques Cross-Layer ainsi que les technologies de Web sémantique pour la découverte combinée de route ainsi que les services web semble une voie prometteuse permettant de mieux optimiser la consommation de l'énergie et la bande passante dans les réseaux MANET.

Dans cette thèse, nous avons commencé par présenter les réseaux MANET, leurs domaines d'application et plus important leurs limites fonctionnelles. Nous avons aussi donné des définitions des technique Cross_Layer et leurs mode de fonctionnement, ainsi que les services web et le web sémantique où nous avons vu comment la découverte actuelle des services web est purement syntaxique et ne permet pas d'exploiter toutes les capacités (dans la publication) des services dans l'annuaire de publication.

Ensuite, et après notre étude détaillée des différents travaux de découverte de service dans les réseaux MANET nous avons proposé un modèle pour l'évaluation du coût de l'énergie globale consommé dans le réseau par rapport au principe de fonctionnement des protocoles de découverte de service web où nous avons identifié un ensemble de paramètre C_i qui influence d'une manière directe ou indirecte le coût de l'énergie consommée.

Comme deuxième objectif dans notre thèse, nous avons proposé un nouveau protocole de découverte de service web optimisé dans les réseau MANET ce protocole offre à la fois une découverte sémantique basé sur les contraintes fonctionnelles des services web tel que les inputs et outputs avec une intégration de ses paramètres dans le processus de découverte de route dans le réseau, aussi il offre une optimisation en matière de consommation d'énergie et bande passante vu la réduction d'utilisation des ressources dans les nœuds ainsi que la réduction des messages dans le réseau.

Ensuite nous avons évalué notre nouveau protocole par rapport au modèle d'évaluation que nous avons proposé où nous avons remarqué que notre protocole offre une optimisation importante par rapport aux travaux étudiés.

Afin de valider notre approche nous avons implémenté notre solution sous le simulateur NS2 où nous avons étudié plusieurs scénarios, ça nous a permis d'évaluer ce nouveau protocole par rapport au paramètre de limitation de champs de recherche que nous avons proposé (Discovery_Diameter) et au différentes contraintes telle que la mobilité et le débit moyen consommé.

2. Perspectives

Les perspectives de ce travail pourraient s'énoncer ainsi :

- Puisque notre supposition du paramètre Discovery_Diameter se base sur la probabilité que des fournisseurs probables peuvent se trouver dans le champ de découverte, chose qui n'est pas toujours satisfaite, dans ce cas une solution pour mesurer taux d'échec est nécessaire dans le processus d'évaluation du protocole de découverte de services dans les réseaux MANET.
- Bien qu'ils ne nécessitent pas d'infrastructure fixe, les réseaux ad-hoc sont souvent liés dans la pratique à une passerelle permettant l'interconnexion à l'Internet filaire dans ce cas une solution pour traiter ce cas d'interconnexion pour permettre la découverte mixte peut être réalisée.
- Etude d'une solution de découverte de service en utilisant les performances d'IPv6 dans ce genre de réseau.
- Etude détaillée du temps d'attente pour la collecte des réponses RREPSSD au niveau du nœud source vu que plusieurs paramètres peuvent entrer en jeu comme les interférences réseau, la congestion, les réponses multiples ...etc.
- Dans le cadre de la QoS une découverte basé sur la capacité du nœud serveur tel que l'énergie restante et son temps de disponibilité dans le réseau peuvent être ajouté dans les paquets de réponse RREPSSD afin d'améliorer les performances de la découverte.
- Etude des solutions proposées dans le cas de composition de services web en matière de modèle de coût d'énergie proposé.

Liste des publications

Revues internationales

Soheyb Ayad, Okba Kazar, Nabila-Aicha Benharkat, Labib-Sadek Terrissa, “Cross-Layer Routing Based on Semantic Web Services Discovery with Energy Evaluation and Optimization in MANET”, *International Journal of Communication Networks and Information Security (IJCNIS)*, Vol 8, No 1, Pages 47-56, 2016.

Conférences internationales

Soheyb Ayad, Okba Kazar, Nabila-Aicha Benharkat, “Evaluating the Energy Consumption of Web Services Protocols in Ad Hoc Networks”, In 2nd AASRI Conference on Power and Energy Systems (PES), *ELSIVIER: AASRI Procedia*, Vol 7, Pages 8–13, 2014.

Références bibliographiques

- [ARNO 97] A. D. Robbins, "Effective AWK Programming," A User's Guide for GNU Awk Edition 1.0.3 February 1997.
- [ALEX 02] A. Mustiere, "Introduction aux services Web," 06 juin 2002.
- [AMUN 06] A. Issac, M. Kushwaha, X. Koutsoukos, S. Neema, J. Sztipanovits, "Efficient Integration of Web Services in Ambient-aware Sensor Network Applications," Published in: 3rd International Conference on Broadband Communications, Networks and Systems, 2006.
- [ABDE 07] A. Obaid, K. Azeddine, M. Hafedh, L. Louise, "A Routing Based Service Discovery Protocol for Ad hoc Networks," Third International Conference on Networking and Services, Athens, pp. 108, 2007.
- [ASAD 11] A. Omar, M. Erol-Kantarci, H. Mouftah, "Management of PHEV Charging from the Smart Grid Using Sensor Web Services," In Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, Niagara Falls, ON, Canada, 2011.
- [ASAD 13] A. Omar, M. Erol-Kantarci, H. T. Mouftah, "A Survey of Sensor Web Services for the Smart Grid," J. Sens. Actuator Netw, pp. 98-108, 2013.
- [AYAD 14] A. Soheyb, O. Kazar, N. A. Benharkat, "Evaluating the Energy Consumption of Web Services Protocols in Ad Hoc Networks," In 2nd AASRI Conference on Power and Energy Systems (PES), ELSIVIER: AASRI Procedia, Vol 7, Pages 8–13, 2014.
- [AHME 14] A. Salim, E. Akram, "An Evaluation of Mobility Effect on Tiny Service Discovery Protocol for Wireless Sensor Networks," INTERNATIONAL JOURNAL OF COMPUTERS TECHNOLOGY, Vol. 14, No. 1, 2014.
- [AINA 15] A. Randrianarisaina, "Modélisation de la consommation d'énergie En vue de la conception conjointe (matériel/logiciel) des applications embarquées," Application aux réseaux de capteurs sans fil (wsn), Engineering Sciences, UNIVERSITE DE NANTES, 2015.
- [AYAD 16] A. Soheyb, O. Kazar, N. A. Benharkat, L. S. Terrissa, "Cross-Layer Routing Based on Semantic Web Services Discovery with Energy Evaluation and Optimization in MANET," International Journal of Communication Networks and Information Security (IJCNIS), Vol 8, No 1, Pages 47-56, 2016.

- [BENA 02] B. Boualem, H. Mohand-Said, L. Alain, R. Christophe, T. Farouk, "On Automating Web Services Discovery," Vol. 14, Issue. 1, pp. 84-96, 2002.
- [BENA 02] B. Benatallah, M. S. Hacid, A. Leger, C. Rey and F. Toumani, "On Automating Web Services Discovery," 2002.
- [BERR 07] B. Gérard, "Pourquoi et comment le monde devient numérique," 2007.
- [BHUM 15] B. Gohil, M. A. Zaveri, H. K. Rath, "Trust Based Service Discovery in Mobile Ad-Hoc Networks," Lecture Notes on Software Engineering, Vol. 3, No. 4, pp. 308-313, 2015.
- [CORS 99] C. Scott and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," January 1999.
- [CHAK 01] C. Dipanjan, F. Perich, S. Avancha, A. Joshi, "DReggie: Semantic Service Discovery for m-commerce Applications," Proceedings of Workshop on Reliable and Secure Applications in Mobile Environment, 2001.
- [CHAK 02] C. Dipanjan, A. Joshi, Y. Yesha, T. Finin, "GSD: A novel group-based service discovery protocol for manets," In Proc.of MWCN, pp. 140-144, 2002.
- [CLAU 03] C. Thomas, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", 2003.
- [CHEN 05] C. GILBERT, J. BRANCH, M. PFLUG, L. ZHU, B. SZYMANSKI, "Sense: A wireless sensor network simulator," Advances in Pervasive Computing and Networking, pp.249–267, 2005.
- [CHEN 05] C. Lei and W. Heinzelman, "QoS-aware routing based on bandwidth estimation for mobile ad hoc networks," IEEE J. Select. Areas Commun., vol. 23, pp.561.572, 2005.
- [CELE 06] C. Campo, C. G. Rubio , A. M. López and F. Almenárez, " PDP: A lightweight discovery protocol for local-scope interactions in wireless ad hoc networks," in Computer Networks Volume 50, Issue 17, pp.3264–3283, 5 December 2006,.
- [CHAK 06] C. Dipanjan, A. Joshi, Y. Yesha and T. Finin, "Towards distributed service discovery in pervasive computing environments," IEEE Transactions on Mobile Computing, 5(2), pp.97–112, 2006.
- [CHAN 07] C. Jean-Pierre, "Algorithme de routage coopératif à qualité de service pour des réseaux ad hoc agri-environnementaux," Thèse doctorat, Université Blaise Pascal - Clermont II, 2007.
- [CHEN 10] C. Min, S. Gonzalez, A. Vasilakos, H. Cao, V. Leung, "Body area networks: A survey," Mobile Netw. Appl. J, Vol. 16, Issue. 2, pp. 171-193, 2010.

- [CHEN 14] C. Ing-Ray, J. Guo, F. Bao, J.H Cho, "Trust management in mobile ad hoc networks for bias minimization and application performance maximization," *Ad Hoc Networks*, Vol. 19, pp. 59-74, 2014.
- [DIPA 01] D. Chakraborty, F. Perich, S. Avancha, A. Joshi, "DReggie: Semantic Service Discovery for m-commerce Applications," *Proceedings of Workshop on Reliable and Secure Applications in Mobile Environment*, 2001.
- [DIPA 02] D. Chakraborty, A. Joshi, Y. Yesha, T. Finin, "GSD: A novel group-based service discovery protocol for manets," *In Proc.of MWCN*, pp. 140-144, 2002.
- [DAVI 08] D. Espès, "Protocoles de routage réactifs pour l'optimisation de bande passante et la garantie de délai dans les réseaux ad hoc mobiles," Thèse de doctorat, Université de Toulouse 3, France, Soutenue le 27 novembre 2008.
- [ENTH] ENTHRONE II project, website <http://www.ist-enthrone.org/>
- [ERIC 02] E. Newcomer, "Understanding Web Services XML, WSDL, SOAP and UDDI," 2002.
- [ERME 04] E. Erwan, "Localisation et Routage géographique dans les réseaux sans fil hétérogènes," Thèse de Doctorat de l'université Paris VI –Pierre et Marie Curie, 2004.
- [FEEN 01] F. Laura-Marie, "An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks," *Mobile Networks and Applications* 6, Netherlands, pp.239-249, 2001.
- [GNUP] <http://www.gnuplot.info/>
- [GRUB 93] G. Thomas, "A translation approach to portable ontology specifications," *Knowledge acquisition*, 5(2), pp.199-220. 1993.
- [GONZ 01] G. C. Javier, T. David and B. Claudio, "A Semantic Web Approach to Service Description for Matchmaking of Services," *Trusted E-Services Laboratory*, 2001.
- [GARC 05] G. J. Antonio, "Service discovery in mobile ad-hoc networks: better at the network layer?," *Parallel Processing, ICPP 2005 Workshops*, pp. 452-457 , 2005.
- [GERA 06] G. Allard, M. Pascale, N. Dang-Quan, S. Nirisha, "Evaluation of the Energy Consumption in MANET," *ADHOC-NOW'06 Proceedings of the 5th international conference on Ad-Hoc, Mobile, and Wireless Networks*, Springer-Verlag Berlin, Heidelberg, pp. 170-183, 2006.
- [GRYA 06] G. A. Eugene, "Service Discovery in Bluetooth," *Group for Robotics and Virtual Reality*, Department of Computer Science, Helsinki University of Technology, Helsinki, Finland, 2006.

- [HASS 99] H. J. Zygmunt, R. Pearlman, "Zone routing protocol for ad-hoc networks," Internet Draft, 1999.
- [HEAT 01] H. Kreger, "Web services conceptual architecture," (wsca 1.0), may 2001.
- [ISMA 08] I. Djama, "Adaptations inter-couches pour la diffusion des services vidéo sans fil," Networking and Internet Architecture, Université Sciences et Technologies - Bordeaux I, French, 2008.
- [JINI 98] JINI, "Jini Architecture Overview," Sun Microsystems, 1998.
- [JERO 04] J. Hoebeke, I. Moerman, B. Dhoedt and P. Demeester, "An Overview of Mobile Ad Hoc Networks: Applications and Challenges," In Journal-Communications Network, 3(3), p.60-66, 2004.
- [JOHN 07] J. Don, Y. Hu, D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," February 2007.
- [JULI 11] J. Mercadal, "Approche langage au développement logiciel : application au domaine des systèmes d'informatique ubiquitaire," Thèse de doctorat, Soutenue le 10 octobre 2011.
- [JOSE 12] J. M. Gummadi, D. Sunil Kumar, P. Suresh Varma and N. Supria, "A Simulation Based Study of AODV, DSR, DSDV, Routing Protocols in MANET Using NS-2," International Journal of Advanced Research in Computer Science and Software Engineering, 2(3), pp.43-51, 2012.
- [KHAN 06] K. Sheraz, Y. Peng, E. Steinbach, M. Sgroi, W. Kellerer, "Application-driven cross-layer optimization for video streaming over wireless networks," In IEEE Communications Magazine, ISSN: 0163-6804, pp.122- 130, 44(1), Jan 2006.
- [KAUR 08] K. Mandeep, S. Bhatt, L. Schwiebert, G. G. Richard, "An Efficient Protocol for Service Discovery in Wireless Sensor Networks," GLOBECOM Workshops, IEEE, pp. 1-6, 2008.
- [LUKA 06] L. Juszczak, L. Jaroslaw, D. Schahram, "Web Service Discovery Replication and Synchronization in Ad-Hoc Networks," ARES'06 Proceedings of the First International Conference on Availability, Reliability and Security, IEEE Computer Society Washington, DC, USA, 2006.
- [MALK 98] Malkin Group, "RIP Version 2", 1998.
- [MOY 98] M. John, "OSPF Version 2," 1998.
- [MICR 00] Microsoft Corporation, "Understanding Universal Plug and Play," A White Paper, 2000.

- [MEEN 14] Meenakshi, Satvika and A. Kaushik, "Comparative analysis of handover and traffic classes in UMTS using OPNET simulator for improving QoS," International Conference on Signal Propagation and Computer Technology (ICSPCT), 2014.
- [NASS 06] N. Sedaghati-Mokhtari, M. N. Bojnordi, N. Yazdani, "Cross-Layer Design: A New Paradigm," Proc In International Symposium on Communications and Information Technologies (ISCIT '06), pp.183-188, Bangkok, Sept 2006.
- [NITI 11] N. G. Palan and A. P. Khadilkar, "Media access control protocol modelling for mobile sensor network-using OMNETT++ -MiXiM network simulator," International Conference on Sustainable Energy and Intelligent Systems (SEISCON), 2011.
- [NAVA 11] N. David, W. Du, F. Mieveville, I. O'connor, "IDEA1: A Validated System C-Based Simulator for Wireless Sensor Networks," Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, 2011.
- [OWLS 04] OWLS, Semantic Markup for Web Services, 22 November 2004. <http://www.w3.org/Submission/OWL-S/>
- [PHON] PHOENIX project, website <http://www.ist-phoenix.org/>
- [PERK 00] P. Charles, "Ad-hoc Networking. Number ISBN 0-201-30976-9," Pearson Education, Addison-Wesley (Eds.), New Jersey, USA, 2000.
- [PAOL 02] P. Massimo, T. Kawamura, T. R. Payne, K. Sycara, "Semantic Matching of Web Services Capabilities," International Semantic Web Conference (ISWC), 9-12 June, Sardinia, Italy, 2002.
- [PERK 03] P. Charles, Belding-Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," July 2003.
- [PAOL 04] P. Massimo, D. Martin, S. McIlraith, M. Burnstein, D. McDermott, D. McGuinness, B. Parsia, T. R. Payne, M. Sabou, M. Solanki, N. Srinivasan, K. Sycara, "Bringing Semantics to Web Services: The OWL-S Approach," First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, CA, 2004.
- [PIRZ 06] P. A. Amir, C. McDonald, "Trust establishment in pure Ad-hoc networks," Wireless Personal Communications, Vol. 37, issue 1-2, pp. 139-168, 2006.
- [PARI 13] Pariselvam S., Parvathi R. M. S. and Franklin D., "Trust based security mechanism for service discovery in MANET," Journal of Theoretical and Applied Information Technology, Vol. 56, No. 2, pp. 226-234, 2013.
- [REMI 06] R. Badonnel, "Supervision des Réseaux et Services Ad-Hoc," Networking and Internet Architecture," Université Henri Poincaré - Nancy I, 2006.

- [RAIS 06] R. Vijay, S. Iyer, "Cross-layer feedback architecture for mobile device protocol stacks," *IEEE Communications Magazine*, 44(1), pp.85-92, Jan 2006.
- [SYCA 98] S. Katia, "Multiagent Systems," *AI Magazine*, (2):79-92, 1998.
- [SALU 99] Salutation Consortium, "Salutation architecture specification version 2.0c," June 1999.
- [SYCA 02] S. Katia, M. Paolucci, T. Kawamura, T. R. Payne, "Importing the Semantic Web in UDDI," *Web Services, E-Business and Semantic Web Workshop, CAiSE*, Toronto, Canada, 2002.
- [SYCA 03] S. Katia, M. Paolucci, A. Ankolekar and N. Srinivasan, "Automated discovery, interaction and composition of Semantic Web services," *Web Semantics: Science, Services and Agents on the World Wide Web 1*, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, pp. 27-46, 2003.
- [SUMI 03] S. Helal, D. Nitin, V. Varun, L. Choonhwa, "Konark: A Service Discovery and Delivery Protocol for Ad-Hoc Networks," *Wireless Communications and Networking, IEEE*, Vol. 3, pp. 2107-2113, 2003.
- [SIVA 04] S. R. Murthy and B. S. Manoj, "Ad-hoc Wireless Networks: Architectures and Protocols," Number ISBN 0-13-147023-X, Prentice Hall (Eds.), New Jersey, USA, 2004.
- [SYCA 04] S. Katia, N. Srinivasan and M. Paolucci, "Adding OWL-S to UDDI, implementation and throughput," *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)* July 6-9, San Diego, California, USA, 2004.
- [SUND 04] S. Sameer, W. Kim, G. Agha, "Sens: A sensor, environment and network simulator," In *Proceedings of the 37th annual symposium on Simulation*, IEEE Computer Society, 2004.
- [SRIV 05] S. Vivek and M. Motani, "Cross-layer design: a survey and the road ahead," *IEEE Communications Magazine*, vol.43, no.12, pp.112-119, December 2005.
- [SCHA 05] S. Mihaela and S. Shankar, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE Wireless Communications Magazine*, vol. 12, no. 4, pp.50-58, August 2005.
- [SOBE 06] S. Ahmed, W. P. Chen, J. C. Hou, L. C. Kung, N. Li, H. Lim, H. Y. Tyan, H. Zhang, "J-Sim : a simulation and emulation environment for wireless sensor networks," *Wireless Communications*, 13(4) :104–119, 2006.
- [SILV 12] S. Icamaan, A. Zisman, "A framework for trusted service," *Service-Oriented Computing*, Heidelberg, Berlin: Springer, pp. 328-343, 2012.

- [SING 13] S. Arjun, P. Chakrabarti, "Ant based resource discovery and mobility aware trust management for Mobile Grid systems," IEEE 3rd International Conference on Advance Computing Conference (IACC), pp. 637-644, 2013.
- [TRAS 01] T. David, C. Bartolini, J. Gonzalez-Castillo, "A Semantic Web Approach to Service Description for Matchmaking of Services," Trusted E-Services Laboratory, July 30th, 2001.
- [ULLA 10] U. Sana, H. Higgins, B. Braem, B. Latre, C. Blondia, I. Moerman, S. Saleem, Z. Rahman, K. Kwak, "A comprehensive survey of wireless body area networks," J. Med. Syst, 2010.
- [UPnP 15] UPnP Forum, "UPnP Device Architecture 2.0," 2015,
- [VEIZ 97] V. John, G. Eric, C. Perkins, M. Day, "RFC 2165: Service Location Protocol," June 1997.
- [VERM 03] V. Kunal, S. Kaarthik, S. Amit and M. John, "Adding Semantics to Web Services Standards," Proceedings of the first IEEE International Conference on Web Services (ICWS'03), Las Vegas, Nevada, pp. 395-401, 2003.
- [VARS 05] V. Alex, B. Reid, E. D. Lara, "The Need for Cross-layer Service Discovery in MANETs," Mobile Adhoc and Sensor Systems Conference, Technical Report, University of Toronto, 2005.
- [WEIS 91] W. Mark, "The Computer for the 21st Century," Scientific American, 265(3): pp.66-75. 1991.
- [WANG 03] W. Qi, M. A. Abu-Rgheff, "Cross-Layer Signalling for Next-Generation Wireless Systems," Proc In IEEE Wireless Communication and Networking, New Orleans, 2003.
- [WSMO 05] WSMO, Web Service Modeling Ontology, 2005. <http://www.w3.org/Submission/WSMO/>
- [WANT 05] W. Roy and T. Pering, "System Challenges for Ubiquitous & Pervasive Computing," Dans Proceedings of the 27th International Conference on Software Engineering (ICSE'05), pages 9-14, St. Louis, MO, USA. ACM Press, 2005.
- [WINT 06] W. Rolf, J. H. Schiller, N. Nikaein, C. Bonnet, "CrossTalk: cross-layer decision support based on global knowledge," in IEEE Communications Magazine, ISSN: 0163-6804, pp.93- 99, 44(1), Jan 2006.
- [WEBE 08] W. Daniel, J. Glaser, S. A. Mdani, S. Mahlkecht, "Power aware simulation framework for wireless sensor networks and nodes," EURASIP Journal on Embedded Systems, 2008.

- [YANI 03] Y. Prié, “Annotations et métadonnées pour le Web sémantique,” LIRIS-Université Lyon 1, Journée Web Sémantique et SHS, Action spécifique WS. 7 mai 2003.
- [ZENG 98] Z. Xiang, R. Bagrodia, M. Gerla, “GloMoSim: a library for parallel simulation of large-scale wireless networks,” Twelfth workshop on Parallel and Distributed Simulation (PADS), 1998.
- [ZHEN 11] Z. Gao, L. Wang, M. Yang and J. Wang, “FNMGSDP: An Optimized Group-Based Service Discovery Protocol for MANETs,” *Wireless Personal Communications*, 57(2), pp.137-162, 2011.
- [4MORE] 4MORE project website <http://4more.av.it.pt/>

ملخص :

عملية اكتشاف خدمات الويب في الشبكات مانات، يعتبر تحديا صعبا للغاية بسبب التغير المستمر في مخطط الشبكة وأيضا عدم وجود مستودع مركزي ثابت لنشر خدمات الويب، وقد اقترحت عدة طرق التي تستند على إما الكلمات الرئيسية و معرفات في عملية الاكتشاف والتي تمثل الخدمة التي سيتم البحث عنها أو باستخدام سيناريو معين من الاكتشاف، كل تلك الحلول المقترحة تحاول احترام قيود مانات مثل الطاقة، وعرض النطاق الترددي وسرعة التدفق، الخ.

في هذه الرسالة نقدم نموذجا الجديد المقترح لقياس تكلفة استهلاك الطاقة الكلي في مانات اعتمادا على بروتوكولات اكتشاف خدمات الويب كما نقدم بروتوكول أمثل جديد لاكتشاف خدمات الويب في مانات استنادا على تقنية بين الطبقات مع نشر في عملية اكتشاف الطرق في الوقت نفسه المعلومات عن خدمات الويب الدلالي و قطر الاكتشاف الذي اقترناه للحد من منطقة الاكتشاف في الشبكة. و في الأخير، نقدم نتائج المحاكاة والتي تظهر التحسين الكبير في مستوى استهلاك الطاقة ومتوسط التدفق.

كلمات مفاتيح: أدهوك، استهلاك الطاقة، مانات، اكتشاف خدمات الويب الدلالي، التدفق، بروتوكولات التوجيه.

Résumé :

Le processus de découverte des services dans les réseaux mobiles ad hoc est considéré comme un défi très difficile en raison de changement continu de la topologie du réseau ainsi que l'absence d'un répertoire central fixe pour la publication des services Web. Plusieurs approches ont été proposées qui sont basées soit sur des mots-clés ou des identificateurs représentant le service à rechercher ou à l'aide d'un scénario spécifique de découverte. Toutes ces solutions proposées tentent de respecter les contraintes de réseaux mobiles ad hoc (MANET) tels que l'énergie, la bande passante, le débit, etc.

Dans cette thèse, nous présentons notre nouveau modèle proposé pour mesurer le coût de la consommation totale d'énergie dans les réseaux ad hoc en fonction des protocoles de découverte des services web. Nous présentons également un nouveau protocole de découverte de services web optimisé dans MANET basé sur des techniques de routage cross_layer avec la diffusion dans le processus de routage en même temps l'information sémantique concernant le service web recherché et un paramètre DISCOVERY_DIAMETER que nous avons proposé pour limiter la zone de découverte dans le réseau. Enfin, nous présentons les résultats de simulation de notre approche définie montrant une optimisation significative au niveau de consommation d'énergie ainsi le débit moyen.

Mots-clés: Ad hoc, la consommation d'énergie, MANET, découverte sémantique, débit moyen, protocoles de routage.

Abstract:

The web services discovery process in mobile ad hoc networks is considered as a very difficult challenge due to the continuous change in the topology of the network and also the lack of a fixed central directory for publishing web services. Several approaches have been proposed which are based on either keywords or identifiers representing the service to be searched or by using a specific scenario of discovery. All of those proposed solutions try to respect the constraints of mobile ad hoc networks MANET such as energy, bandwidth, throughput, etc.

In this thesis we present our new proposed model for measuring the cost of the overall energy consumption in ad hoc networks depending on the web services discovery protocols. We also present a new optimized web services discovery protocol in MANET based on cross_layer routing techniques with the dissemination in the routing process at the same time the semantic web services information and a DISCOVERY_DIAMETER parameter that we have proposed to limit the area of discovery in the network. Finally, we present simulation results of our defined approach showing a significant optimization of the energy consumption level and the average throughput.

Keywords: Ad hoc, Energy consumption, MANET, Semantic web services discovery, Average Throughput, Routing protocols.
