Protocole de routage externe type BGP dans un environnement réseaux tactiques adhoc mobiles : faisabilité et performances

Florian Grandhomme

2017

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale MathSTIC

présentée par

Florian Grandhomme

préparée à l'unité de recherche n^o 6074
Institut de Recherche en Informatique et Systèmes Aléatoires
Université Rennes 1

Études de protocoles de routage dynamique externe de type BGP dans un environnement réseaux tactiques ad hoc mobiles : faisabilité et performances

Thèse soutenue à Rennes
le 23 Novembre 2017

devant le jury composé de :

Jalel BEN OTHMAN
Professeur, Université Paris XIII / Rapporteur

Pascale MINET
Chargée de recherche, INRIA Paris / Rapporteur

Hossam AFIFI
Professeur, Télécom Sud Paris / Examinateur

Isabelle CHRISMENT
Professeure, Télécom Nancy / Examinateur

François TAÏANI
Professeur, Université de Rennes 1 / Examinateur

Adlen KSENTINI
Maître de conférences, Eurecom / Directeur de thèse

Gilles GUETTE
Maître de conférences, Université de Rennes 1 / Co-Directeur de thèse

Thierry PLESSE
Ingénieur, DGA - Maîtrise de l'Information / Examinateur

Remerciements

En avant-propos de ce manuscrit, je voulais dans un premier temps adresser une pensée, un merci à tous ceux qui ont été là de près ou de loin pendant cette thèse et qui ont tous contribué à sa réussite aujourd'hui.

Tout dabord, je voudrais remercier l'ensemble des membres du jury qui ont pris le temps détudier mes travaux et de m'apporter des éclairages intéressants de par leurs remarques et questions. Mme. Pascale Minet et M. Jalel Ben-Othman qui ont été les rapporteurs, Mme. Isabelle Chrisment et M. Hossam Afifi qui ont été examinateurs ainsi que M. François Taïani qui a présidé le jury.

Quelques lignes et remerciements particuliers pour les trois autres personnes du jury, qui ont été à mes côtés pendant ces trois années. Adlen Ksentini, merci pour ton aide, tes jugements ainsi que les différentes idées que tu as pu suggérer pour m'aider à avancer. Un grand merci également à Gilles Guette. Au-delà de l'enseignant que j'ai pu avoir au fil de ma scolarité, j'ai également pu apprécier quelqu'un d'une grande aide, toujours plein d'idées et d'énergie à revendre. Enfin, je voudrais remercier Thierry Plesse, qui m'encadrait en tant que représentant DGA. Tu as toujours fait en sorte que les travaux soient mis en valeur et appréciés à Bruz ou dans des collaborations internationales.

Je voudrais adresser des remerciements à l'équipe CIDRE pour les moments vécus pendant ces trois années. Passer de l'autre côté de la barrière et vivre au quotidien de l'équipe m'ont énormément appris que ce soit par le biais de discussions aux pauses café ou bien d'évènements plus importants. Dans ces échanges, je pense également à tous les non-permanents que j'ai pu rencontrer. Qu'ils soient stagiaires, docteurs ou futurs docteurs, je n'oublierai pas les moments scientifiques et de détente passés avec eux.

Enfin, un très grand merci à toute ma famille, mes proches et ma compagne qui m'ont toujours soutenu et encouragé dans la voie que je choisissais. Cette réussite ne serait pas là aujourd'hui s'ils ne m'avaient pas fait confiance et s'ils nétaient pas à mes côtés.

Résumé

Les théâtres d'opérations militaires s'organisent aujourd'hui sous la forme de coalitions. Les forces armées qui sont déployées possèdent des moyens technologiques (communication, routage) et créent des réseaux sans fil. Le mouvement de ces forces sur le terrain donne au réseau une topologie fortement dynamique. Il se caractérise comme un réseau sans fil mobile, plus communément appelé MANET (Mobile Ad hoc NETwork). Pour l'efficacité de la mission et des communications, il est intéressant d'interconnecter facilement les différents groupes participant à la coalition. Cependant, chaque membre de la coalition possède sa propre architecture et souhaite rester autonome, sans s'adapter aux autres. Comme les opérateurs ont pu le faire lors de la création de l'Internet avec le protocole BGP (Border Gateway Protocol), il est nécessaire de créer un protocole qui connecte tous ces groupes participant à la coalition. Ce protocole doit permettre de créer des connexions inter-groupes, supporter facilement les changements de topologies et appliquer des politiques de routage qui permettent d'indiquer des préférences de groupes à emprunter, par exemple. Dans cette thèse, nous allons dans un premier temps étudier la non-adaptabilité de BGP sur les réseaux MANET. Puis, nous étudierons les différentes propositions de la littérature. Ensuite, nous présenterons notre solution ITMAN (Inter Tactical Mobile Ad hoc Network) dans sa première version et les améliorations que nous avons pu y apporter. Enfin, nous terminerons ce manuscrit par les perspectives qui peuvent être menées suite à cette thèse.

Abstract

Nowadays, military operations are organized as coalitions. The armed forces that are deployed have technological features (communication, routing) that create wireless networks. The mobility of these forces on the ground means that the network has a highly dynamic topology. This is a mobile wireless network, more commonly called MANET (Mobile Ad hoc NETwork). For mission and communication efficencies, interconnection between the various groups participating in the coalition is necessary. However, each member of the coalition has its own architecture and wants to keep independancy from the other groups technologies. This situation is similar as the Internet construction, where operators were able to interconnect their infrastructures with BGP (Border Gateway Protocol). It is important to create a protocol that can connect all the groups involved in the coalition. This protocol should allow inter-group connections, easily support topology changes and apply routing policies that specifies groups on the route preferences, for example. In this thesis, we first study the adaptability issues of BGP on MANET. Then, we study the proposals that have been made in the literature. Next, we present ITMAN (Inter Tactical Mobile Ad Hoc Network) in its first version and the improvements that have been made. Finally, we will conclude this manuscript with the perspectives that can be highlighted following this thesis.

Table des figures

Liste des tableaux

Table des matières

Chapitre 1

Introduction

1.1 Cadre et problématique

Les opérations militaires sur les théâtres d'opération sont aujourd'hui de plus en plus organisées sous forme de coalitions. Ces dernières reflètent la collaboration et l'échange entre différents pays alliés qui s'unissent pour atteindre un objectif commun. Pour réussir ces opérations, le matériel de communications utilisé sur le terrain par les forces armées comporte un nombre important de technologies (radiologicielles, antennes UHF/VHF, routeurs IP, outils de visioconférence, communication voix...). L'apport d'équipement numérique pour les différents participants sur théâtre d'opération (fantassins, véhicules, drones, hélicoptères) s'inscrit dans un cadre où chaque pion constitue un nœud de réseau à part entière qui participe à la transmission des données. Tous ces nœuds mobiles participent équitablement à la transmission des données et n'ont pas besoin d'un élément central pour gérer et router les communications. Nous sommes en présence d'un réseau ad hoc mobile (plus communément appelé Mobile Ad hoc NETwork - MANET).

Chaque entité présente dans la coalition gère de manière indépendante son groupe d'unités. Cependant, pour que la coalition fonctionne de son mieux, chacun de ses groupes s'interconnecte avec les autres pour échanger des informations et contribuer à la réussite de la mission. Nous sommes donc face à une situation où plusieurs groupes autonomes veulent se connecter et échanger des données, sans changer leur structure ou technologie interne. Cette situation est identique à la construction de l'Internet. En effet, des fournisseurs d'accès, des états ou des organisations ont voulu échanger des informations et créer un réseau interconnecté. Cependant, chaque membre a voulu garder la souveraineté de son réseau et pouvoir contrôler ses échanges (nature du flux, route à emprunter) par des politiques de routage.

Le protocole de routage externe BGP (Border Gateway Protocol) a donc été créé pour permettre aux différents opérateurs de s'interconnecter, garder la souveraineté de leur réseau et appliquer des politiques de routage. Mais son utilisation stricte sur les réseaux tactiques de terrain n'est pas réalisable. En effet, BGP est un protocole administré car il demande une configuration statique des routes et des politiques par l'administrateur. Sur un réseau hautement dynamique comme une coalition militaire, il paraît peu efficace d'utiliser ce protocole tel quel aujourd'hui. Il y a donc besoin d'un protocole alternatif qui puisse prendre en compte toutes les caractéristiques spécifiques que sont les réseaux ad hoc mobiles comme la topologie très changeante et la difficulté de maintenir des routes à jour.

1.2 Déroulement de la thèse

Pour débuter ce manuscrit, nous plaçons le cadre militaire des travaux. Ces éléments de contexte vont nous permettre de décrire en détail le déploiement d'un théâtre d'opération et les technologies de communication utilisées. Pour comprendre ces éléments technologies, nous expliquerons les différents types de réseaux sans fil qui existent aujourd'hui, la manière dont le routage des données fonctionne, leurs avantages ainsi que leurs inconvénients. Toujours dans le chapitre contexte de cette thèse, nous verrons comment le réseau s'organise sur les aspects de coalition tels que les accords politiques, le déploiement d'outils de sécurité et son organisation de manière générale. Pour conclure ce chapitre, nous présenterons en détails le protocole BGP qui constitue notre modèle de fonctionnement. À travers ses algorithmes et des travaux de littérature, nous expliciterons les problématiques existantes sur l'adaptation stricte du protocole BGP aux réseaux ad hoc mobiles, dont la principale se situe dans la gestion de l'adressage et des préfixes IP du réseau.

Plusieurs travaux de la littérature existent pour créer un protocole inter-domaine sur un réseau ad hoc mobile. Dans l'état de l'art, nous étudierons toutes les propositions qui ont pu être faites dans ce domaine de recherche. Nous expliquerons des algorithmes permettant de répondre à des problématiques isolées mais également des protocoles complets qui ont chacun des spécificités propres. Tous ces éléments seront détaillés dans leur fonctionnement pour que nous puissions voir leurs avantages et inconvénients dans notre cas d'usage. Au terme de cet état de l'art, nous établirons une classification de tous ces protocoles basée sur des recommandations existantes et des critères supplémentaires que nous aurons définis pour correspondre à nos objectifs. Dans la littérature, nous verrons que le filtre de Bloom se dégage nettement comme solution pour résoudre la problématique majeure du routage inter-MANET. Cependant, la philosophie du réseau ad hoc est de permettre à des nœuds possédant les mêmes technologies de communication de pouvoir échanger des données, sans cloisonnement par l'adressage IP.

Pour valider ce fonctionnement des réseaux ad hoc, nous avons mis en place une étude comparative. Cette première contribution met en concurrence NS3 (Network Simulator 3) et CORE (Common Open Research Emulator), majoritairement utilisés dans la littérature comme outil de validation, avec une plateforme expérimentale. Le but est de vérifier que les pré requis pour réussir un test de connectivité soient strictement identiques quelque soit l'outil utilisé. Nous utilisons deux groupes de 5 nœuds qui sont chacun configurés sur deux préfixes IP différents. L'expérience nous montrera que les outils logiciels adoptent le même comportement, c'est-à-dire que seuls les nœuds appartenant au même préfixe IP peuvent communiquer, comme un réseau filaire. Cependant, la plateforme validera notre hypothèse ad hoc puisque chaque nœud est capable de joindre les autres, indépendamment des adresses et préfixes IP utilisés. Nous pousserons cette étude au niveau Liaison de données et ferons le même type de constatation. Les logiciels n'implémentent pas ou mal les mécanismes de signalisation de niveau 2 alors que la plateforme expérimentale montrera que le niveau 2 est bien un pré-requis de communication, et cela même avant l'adressage IP. À partir de toutes ces observations, nous établirons une liste de problématiques observées par l'expérience pour concevoir un protocole inter-MANET.

Nous expliquerons alors notre deuxième contribution qui est ITMAN. Ce protocole utilise le chiffrement des paquets de routage OLSR pour créer des groupes de communications distincts, les protéger de l'écoute extérieure et s'interconnecter avec d'autres groupes de la coalition par le biais d'un système de *beaconing*. De plus, des politiques de routage simples sont appliquées pour permettre au commandement des forces armées de diriger ses communications. Les politiques sont au nombre de trois : *Deny* (refus de communication), *Link* (autorisation limitée) et *Merge* (autorisation complète). Pour évaluer l'efficacité de ITMAN, nous mesurerons le délai qu'il créé dans le réseau par le chiffrement des paquets. Ce délai se montrera relativement faible, d'autant plus au vu de l'implémentation qui en est faite puisque notre preuve de concept n'est pas conçue au niveau du système d'exploitation. Bien que les performances d'ITMAN s'affichent comme correctes, nous ferons l'observation de cas politiques qui sont problématiques pour satisfaire les communications de la coalition.

Le premier cas problématique recensé est le blocage multi-saut. Si deux groupes non-voisins possèdent sur la route une politique *Deny* entre deux voisins, alors, leur communication est bloquée. Le deuxième cas que nous avons relevé est la divulgation non-contrôlée. Dans le cas de deux groupes qui ont une politique *Deny*, si un groupe tiers a une politique avec des groupes concernés par la politique *Deny*, alors, il peut diffuser les informations de l'un vers l'autre et réciproquement. Pour confirmer la pertinence de la résolution de ces problèmes, nous avons mené une étude théorique pour évaluer la fréquence d'apparition de ces cas bloquants. Il va s'avérer qu'à partir de coalitions d'une taille de quatre groupes, il existe une probabilité supérieure à 50% qu'un cas bloquant apparaisse dans le réseau. Sur ce constat, nous avons proposé dans une troisième contribution deux solutions distinctes pour résoudre chacun des problèmes. D'un côté, une solution de tunnel groupe-à-groupe et de l'autre, du filtrage d'annonce. Ces alternatives ont été testées sur un scénario expérimental comportant une mobilité des nœuds. Cette expérience nous a montré que, comparé à un réseau ad hoc de référence, la mise en place de tunnels ou de filtrage ne dégrade les performances globales du réseau.

Pour conclure ce rapport, nous ferons un rappel des différentes contributions qui ont été présentées. Suite à cela, nous proposerons des pistes de travail supplémentaires comme perspectives d'avenir dans le domaine des communications tactiques inter-domaine. Après avoir introduit le manuscrit et présenté le contenu des chapitres, nous commençons par le premier d'entre eux, le contexte d'étude de la thèse.

Chapitre 2

Contexte de la thèse

Dans ce chapitre, nous expliquons l'environnement de travail de cette thèse lié au contexte militaire. La particularité des environnements militaires va nous permettre de poser des hypothèses de travail que nous n'aurions pas obligatoirement pu avoir dans un contexte plus générique.

2.1 Dispositif des forces armées

Aujourd'hui, les communications sur les théâtres d'opération sont essentielles afin de pouvoir observer, évaluer et prendre des décisions face à une situation. Dans le cadre des programmes de défense dont s'occupe la DGA (Direction Générale de l'Armement), les forces armées sont équipées de matériel de communication radio numérique leur permettant de dialoguer avec des voisins proches sur des technologies principalement TDMA [NK85] (Time Division Multiple Access) et sur différentes bandes de fréquences telles que la VHF (Very High Frequency - de 30 à 300 MHz) ou l'UHF (Ultra High Frequency - de 300 à 3 000 MHz). De plus, si un soldat est connecté sur plusieurs fréquences, il doit être capable de router ses données pour les envoyer d'une interface à une autre. Ainsi, les forces armées doivent pouvoir former un réseau global, éventuellement composé de plusieurs réseaux ad hoc.

Les soldats qui partent en mission sur un théâtre d'opération, dépendent d'une base qui est déployée afin de coordonner les opérations et de donner des ordres en boucles courtes. Un commandement est également présent en métropole et doit pouvoir communiquer avec les forces déployées sur le terrain à l'aide de liaisons satellitaires. L'ensemble de ces éléments et communications présents lors d'une opération est illustré par la Figure 2.1.

Nous allons voir dans la partie suivante ce que sont les réseaux ad hoc, les différents modes associés ainsi que le routage des données dans ce type de réseau.

2.2 Les architectures de réseaux sans fil

Dans cette partie, nous nous intéressons aux réseaux sans fil IEEE 802.11 plus communément appelés réseaux WiFi. Nous présentons les architectures existantes, leurs avantages et inconvénients ainsi que les différentes familles de protocoles de routage utilisées.

2.2.1 Les réseaux sans fil - mode infrastructure

Le réseau sans fil le plus utilisé dans notre quotidien est le réseau sans fil en mode infrastructure. Dans ce mode, tous les nœuds du réseau sans fil sont connectés à un point d'accès qui distribue les communications entre les nœuds et vers l'extérieur s'il y est connecté. Le routage des données n'est pas effectué par les nœuds du réseau mais par le point d'accès. Un exemple d'usage, comme le montre la Figure 2.2, est l'accès à Internet à notre domicile. Le fournisseur d'accès à Internet (FAI) met à notre disposition une box, qui constitue le point d'accès, et permet de distribuer

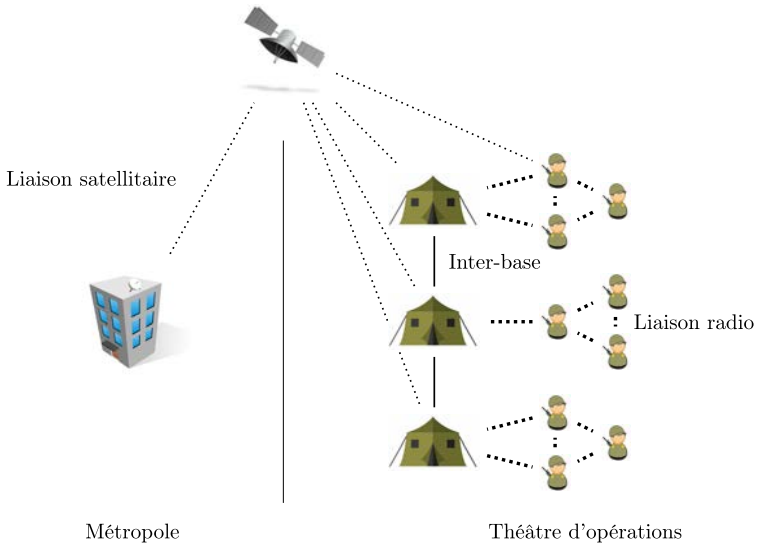


FIGURE 2.1 – Déploiement sur un théâtre d'opérations

aux nœuds (PC, téléphone, tablette) se trouvant au domicile les communications, quelles soient de l'extérieur ou d'une autre machine du réseau. Ce point d'accès constitue un point central de défaillance (également appelé Single Point of Failure - SPOF) sur ce type d'architecture puisque si le point d'accès vient à tomber en panne, les nœuds se trouvent dans l'incapacité de communiquer.

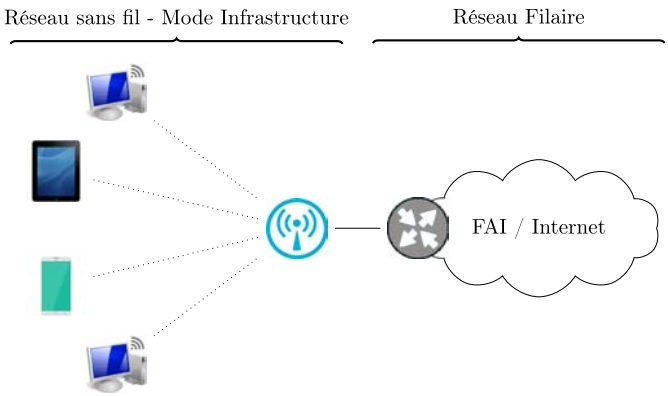


FIGURE 2.2 – Illustration d'un réseau d'infrastructure

2.2.2 Les réseaux sans fil - mode ad hoc

Les réseaux ad hoc sont une autre architecture spécifique des réseaux sans fil. Ils sont auto-organisés et n'ont pas besoin d'élément central pour assurer des communications. Chaque nœud participe au routage des communications dans le réseau comme illustré sur la Figure 2.3. Cette architecture s'oppose au mode infrastructure où toutes les communications sont gérées par un point d'accès.

2.2.3 Les réseaux sans fil - mode ad hoc mobile

Lors d'une mission sur un théâtre d'opérations, l'une des premières tâches effectuée est le déploiement d'un dispositif de communications. Son objectif est de relier les différents groupements d'unités que l'on peut avoir sur le terrain. Cela peut aller du commandement sur une base, au

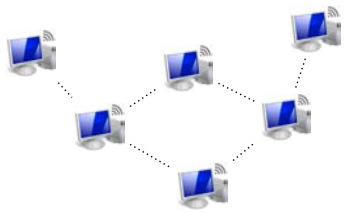


FIGURE 2.3 – Illustration d'un réseau ad hoc

déploiement de drones, de véhicules, d'hélicoptères ou de fantassins. Ce sont ces derniers plus particulièrement qui sont au cœur de l'action et qui échangent des informations entre eux et avec leur hiérarchie. Ils forment des réseaux ad hoc sur le terrain. Ces différentes entités combattantes, possédant chacune l'équipement nécessaire (radios, antennes, systèmes d'information) pour communiquer sans fil, sont assimilés à des nœuds du réseau. Ils sont mobiles et vont donc modifier constamment la topologie du réseau. Cette famille de réseau se nomme Mobile Ad hoc NETwork (MANET). Dans ce type d'architecture, tout comme pour le mode ad hoc, tous les nœuds participent au routage. Cependant, la difficulté est de maintenir en quasi temps réel (ou dans un temps optimal) les routes à jour (ou de les découvrir rapidement) et d'avoir en permanence une vision actualisée de la disponibilité des nœuds.

2.3 Routage dans les réseaux ad hoc

Dans un réseau ad hoc, chaque nœud participe au routage. Chaque nœud doit donc exécuter un protocole de routage ad hoc. Il existe deux familles principales de protocoles de routage dans les réseaux ad hoc. Les protocoles proactifs et les protocoles réactifs.

2.3.1 Les protocoles proactifs

Dans cette première famille, le protocole cherche à connaître tous les nœuds accessibles dans le réseau et toutes les routes associées. Ainsi, il connaît à chaque instant la topologie du réseau et les destinations possibles. Pour cela, un nœud informe à intervalle régulier ses voisins directs de sa présence ainsi que des nœuds auxquels il a accès soit directement, soit indirectement. À réception de toutes ces informations, chaque nœud calcule les routes. À la manière d'OSPF[Moy91] (Open Shortest Path First), le protocole de routage ad hoc OLSR[CDJH14] (Open Link State Routing) calcule le plus court chemin pour accéder à toutes les destinations possibles. OLSR fonctionne à l'aide de deux paquets différents :

— HELLO : Ce message permet à un nœud de connaître son voisinnage à un saut. Lorsque le nœud connaît ses voisins à un saut et qu'il a reçu leurs messages HELLO, il possède alors l'ensemble des nœuds accessibles à un saut et deux sauts. Parmi tous ses voisins symétriques à un saut, le nœud choisit un ou plusieurs MPR (Multi Point Relay). Le but est d'avoir le plus petit sous-ensemble existant permettant d'atteindre toutes les destinations à deux sauts. L'utilisation de ces MPR, au lieu d'un *broadcast*, permet de réduire l'envoi d'informations dans le réseau.

— TC (Topology Control) : Ces messages, générés uniquement par les MPR, servent à construire la table de routage. Chaque MPR envoie dans son TC l'ensemble de ses voisins qui l'ont choisi comme MPR pour établir une vision proche. Ils sont en plus échangés entre les MPR pour construire l'ensemble de la topologie.

Dans OLSR, le lien radio est évalué avec deux composantes : le LQ (Link Quality) et le NLQ (Neighbor Link Quality). Le coût de la route est alors calculé avec ces deux éléments pour obtenir une métrique ETX (Expected Transmission Count) qui vaut $1/(LQ * NLQ)$.

L'avantage de cette famille de protocoles est que lorsqu'un nœud veut envoyer un message, si la route existe elle est disponible immédiatement dans sa table de routage. Cependant, cela nécessite d'utiliser de la bande passante en permanence pour établir et maintenir les tables de routage (envoi

périodique des messages de contrôle HELLO et TC), alors que les réseaux ad hoc possèdent des capacités plus faibles que les réseaux filaires.

2.3.2 Les protocoles réactifs

La deuxième grande famille est celle des protocoles réactifs. Ces protocoles s'exécutent uniquement en cas d'absence de route lors de l'envoi d'un paquet. Lorsqu'un paquet doit être envoyé, la source envoie une requête de route qui se propage dans tout le réseau, en ajoutant au fur et à mesure les nœuds traversés. La requête de route possède une durée de vie limitée pour ne pas engorger indéfiniment le réseau. Si aucune réponse n'est revenue suite à la requête, la source peut renvoyer la requête en indiquant un délai d'attente plus long. Une fois que la requête atteint la destination, comme tous les nœuds traversés ont été indiqué au fur et à mesure dans la requête, la destination peut envoyer une réponse de route en utilisant la suite de nœuds en sens inverse. Une fois la réponse revenue à la source, la source, la destination et tous les nœuds intermédiaires sur le chemin connaissent la route. Le protocole le plus représentatif des protocoles réactifs est AODV (Ad hoc On-demand Distance Vector) [PBRD03] qui utilise les paquets RREQ (Route Request) pour la découverte de route et RREP (Route Response) pour transmettre la route valide.

Contrairement aux protocoles proactifs, leur surcoût en message de signalisation dans le réseau est faible puisque ceux-ci ne sont envoyés que s'il y a besoin d'une route. Cela induit une latence à l'initiation d'une communication car il faut au préalable créer la route.

2.3.3 Les autres familles de protocoles de routage

En dehors des protocoles proactifs et réactifs, il existe d'autres types de protocoles qui ont été proposés pour le routage dans les réseaux ad hoc. L'étude de [BTA+11] présente ces différentes familles de protocoles. En voici quelques exemples :

- Hybride : un mélange entre proactif et réactif. Le mode choisi dépend de la mobilité des nœuds. En forte mobilité, le routage sera réactif alors que sur des réseaux assez peu mobiles, le routage proactif sera choisi (ex : ZRP[HPS02], DST[RRS+99]).
- Géolocalisation : les routes sont calculées par la position des autres nœuds qui est connue en temps réel (ex : LAR[SD17], GPSR[KK00]).
- Énergétique : les routes privilégient les nœuds qui possèdent le plus d'énergie (ex : MEHDSR [TT09], DEAR [ALF03]).
- Multicast : comme pour les réseaux filaires, il est possible de diffuser du contenu d'une source vers un groupe d'abonnés (ex : DCMP[DMRM02], ADMR[JJ01]).

Dans le cadre des réseaux militaires, chaque nœud est assimilé à un soldat ou à un véhicule (voir un hélicoptère ou un drone). Sur le théâtre d'opérations, ce nœud est mobile et le réseau voit sa topologie évoluer constamment. Le réseau est alors considéré comme un Mobile Ad hoc NETwork, plus communément appelé MANET. Ce MANET va donc être un réseau unique sous l'autorité et l'administration d'une seule entité. Dans notre cas d'étude, nous voulons connecter différents réseaux entre eux, bien qu'ils ne soient pas dirigés par une même entité et qu'ils n'adoptent pas tous le même protocole de routage.

2.4 Environnement militaire et planification de mission

Sur un théâtre d'opérations, les opérationnels français n'agissent pas forcément seuls et plusieurs nations ou entités peuvent se rassembler sous un objectif commun dans une coalition (OTAN, UE). Avant de démarrer l'opération, les membres de la coalition s'entendent pour organiser la mission. Lors de cette étape, les hiérarchies alliées qui constituent la coalition s'entendent sur le déploiement du réseau pour qu'il soit fonctionnel. Plusieurs éléments vont donc être organisés voire figés au moment de la préparation de la mission et n'auront pas d'influence ou de modification lors du fonctionnement du réseau. Cette partie présente les hypothèses de travail, des définitions et caractéristiques des éléments du réseau tactique.

2.4.1 Définition d'un groupe

Un groupe se définit comme un ensemble de nœuds capable de communiquer entre eux. Cela se fait selon deux conditions :

i) Selon les technologies utilisées

ii) Selon des contraintes opérationnelles telles que des communications chiffrées

2.4.2 Organisation du réseau

Comme l'indique le premier point de la définition d'un groupe, pour établir des communications, les technologies embarquées dans les nœuds doivent être compatibles. Cela signifie une homogénéité au niveau 3 (couche Réseau) comme par exemple l'utilisation d'un adressage IPv4 (resp. IPv6). Il est également important d'avoir une homogénéité au niveau 2 (couche Liaison) sur des technologies comme le CSMA/Wifi ou le TDMA. Enfin, il est important que les nœuds dans un même groupe utilisent le même protocole de routage pour pouvoir acheminer les paquets. Dans notre cas, nous nous baserons sur des nœuds qui utilisent une technologie WiFi/CSMA et le protocole de routage intradomaine OLSR.

Le fait de parler de technologies homogènes dans le réseau ne signifie pas qu'il ne doit y en avoir qu'une seule et certains groupes doivent s'aligner sur la technologie d'autres groupes. En effet, un nœud peut embarquer plusieurs technologies, s'interfacer avec différents groupes de nœuds et agir comme passerelle. Par exemple, un groupe peut utiliser une technologie TDMA et un autre une technologie CSMA. Cependant, pour qu'ils puissent s'échanger des informations, il faut que des nœuds passerelles dans chaque groupe supportent les deux technologies. Dans les réseaux ad hoc opérationnels, si deux nœuds utilisent les mêmes technologies de niveau 2 et 3 et possèdent le même protocole de routage, ils sont capables de s'échanger des informations.

En planification de mission, les unités et groupes déployés sont connus de chaque partie. Ils sont donc capables d'organiser un plan d'adressage pour que chaque membre de la coalition ait un préfixe IP cohérent et qui ne recouvre pas le domaine d'un autre allié. De plus, deux nœuds n'utilisent pas la même adresse IP en départ de mission.

Sur le terrain, nous recherchons à faire fonctionner en totale autonomie les nœuds de terrain pour organiser le réseau en groupes de communications et transmettre les données. Nous considérons qu'il existe quasiment toujours une communication satellite permettant de recevoir les ordres du commandement en base arrière. Ces ordres peuvent être la fusion ou la séparation de domaines, la modification d'accords politiques ou bien la révocation de certificats suite à une compromission. Dans notre travail, cette liaison satellite ne sera pas utilisée, mais nous permet de justifier certains objectifs qui doivent être accomplis sur ordre.

2.4.3 Sécurité

Nous faisons l'hypothèse que chaque nœud du réseau tactique supporte une ou plusieurs applications qui peuvent posséder différents niveaux de classification. De manière générale, un nœud embarque du matériel cryptographique pour pouvoir réaliser ces opérations de sécurité qui permettent de mettre en place cette classification. Nous faisons l'hypothèse qu'en préparation de mission, chaque nœud se voit attribuer sa paire de clé privée/publique certifiée par l'autorité de certification de la coalition qui est de confiance. Bien que le réseau soit une coalition unie, chaque membre y participant a ses propres infrastructure de distribution et de gestion du matériel cryptographique. Chaque nœud est donc capable d'envoyer des messages chiffrés et de s'authentifier auprès des autres membres. En cas de compromission, la base est également capable de supprimer les droits d'accès à un nœud en révoquant ces mêmes éléments.

Lorsque nous aborderons ces questions dans le manuscrit, nous nous pencherons sur l'exploitation de ce matériel existant et non sur sa création, sa gestion ou bien sa révocation.

2.4.4 Politique et accords de coalition

Lorsque la mission s'organise, chaque membre a un objectif précis dans le but de faire réussir la coalition dans sa mission. Cependant, la nature de ces objectifs et la sensibilité des données récoltées par les différents participants n'est pas la même. Ainsi, nous prenons comme hypothèse que des accords de confiance sont passés entre les différents groupes et caractérisent les niveaux de données que chacun tolère d'envoyer sur le réseau d'un autre. Quelques soient ces accords passés, nous considérons qu'ils sont toujours symétriques. Si A a une haute confiance en B, alors la réciproque est vraie. Cette hypothèse nous permettra de travailler plus simplement puisque nous considérons que les politiques de routage à appliquer sont symétriques. Comme nous avons pu l'évoquer précédemment, bien que ces politiques soient définies en préparation de mission, il est possible de les modifier à tout moment et pouvoir continuer à assurer les services de communication.

Après avoir vu les hypothèses organisationnelles, réseaux et de sécurité sur les groupes constituant la coalition, nous allons définir les objectifs de la thèse, qui concernent l'interconnexion de ces MANET. Pour cela, nous utilisons comme modèle un protocole filaire qui a été créé et qui est encore utilisé de nos jours : le protocole BGP[RL95] (Border Gateway Protocol).

2.5 Le routage inter-domaine et les contraintes d'application aux MANET

Dans le cadre d'une coalition militaire, plusieurs pays sont amenés à collaborer pour réussir une mission. Cependant, chaque pays n'utilise pas obligatoirement les mêmes protocoles de routage et souhaite garder son réseau autonome et sous son commandement. Cette situation est similaire à la construction de l'Internet, où différents opérateurs et pays devaient s'associer pour construire un réseau unique, tout en utilisant des moyens techniques différents et en assurant la souveraineté de chaque membre.

2.5.1 Fonctionnement du routage inter-domaine dans les réseaux filaires

Le protocole BGP a été conçu pour faire communiquer différents réseaux entre eux afin de former l' Internet. En effet, si les réseaux constituant Internet étaient directement interconnectés par un protocole de routage interne comme OSPF ou RIP, les matériels d'interconnexion devraient gérer des tables de routage contenant toutes les destinations possibles sur Internet (des centaines de milliers). Les technologies actuelles ne permettent pas en termes d'efficacité de stockage et de vitesse d'exécution d'utiliser les protocoles de routage classiques. Il est donc important de créer une identification alternative de tous ces réseaux et de les interconnecter par un adressage plus léger. Ainsi, chaque réseau constitutif d'Internet, qui peut être détenu par un Fournisseur d'Accès à Internet (FAI), une organisation ou un état par exemple, est considéré comme un sous-ensemble. Ces domaines sont définis par le protocole BGP comme des *Autonomous System* (AS)[HB96] et identifiés de manière unique par un nombre de 32 bits à la création du protocole. Aujourd'hui, ce nombre est passé à 64 bits pour faire face à l'augmentation du nombre d'AS existants. Dans un AS, l'administrateur désigne de manière statique ses *passerelles* qui seront en interaction avec les autres AS et qui seront en charge d'échanger les informations de routage et faire les calculs de chemins. En identifiant chaque réseau par un nombre unique, l'espace d'adressage est confiné et maîtrisé pour router efficacement les données.

Lors d'une communication inter-AS (ou inter-domaine), le protocole BGP agit à deux niveaux pour construire la topologie du réseau grâce à deux sous-protocoles e-BGP et i-BGP qui fonctionnent au dessus de TCP :

— e-BGP (external BGP) : communication entre deux passerelles d'AS différents
— i-BGP (internal BGP) : communication entre deux passerelles d'un même AS

Nous utilisons l'exemple de la Figure 2.4 pour illustrer une communication inter-domaine entre les AS 10 et 30. Chacun de leur côté, les routeurs A et B ainsi que les routeurs D et E vont prendre connaissance de leur lien et s'ajouter mutuellement comme voisin par e-BGP. Cependant, pour que l'AS 10 puisse joindre l'AS 30, il faut traverser l'AS 20. Ainsi, les routeurs B, C et D vont s'envoyer

au sein de leur AS via i-BGP les informations qu'ils ont récoltées par e-BGP.

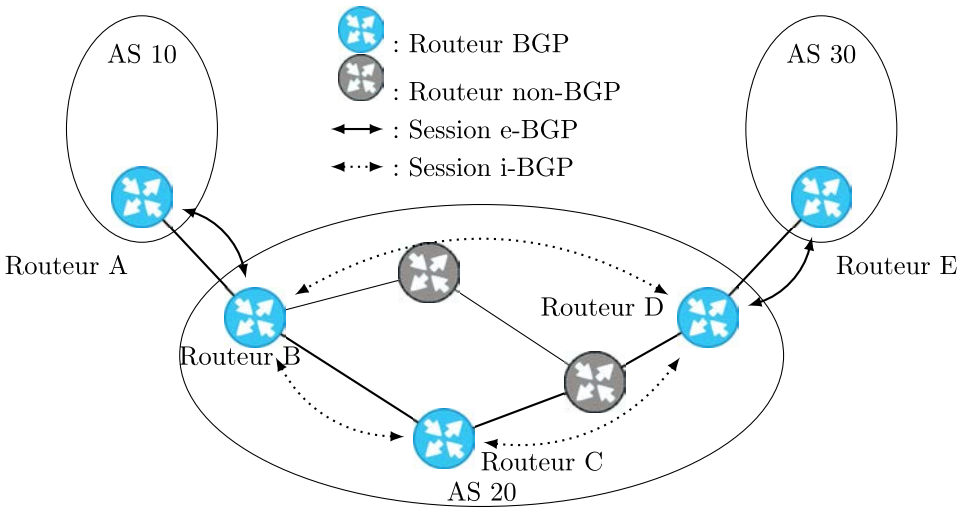


FIGURE 2.4 – Illustration des échanges BGP entre deux AS distants

Les échanges entre routeurs BGP se font via 3 messages principaux :

— OPEN : lorsque la session est établie, le routeur diffuse ses informations (AS joignables, routes)
— UPDATE : si des modifications de topologie arrivent, les mises à jour sont faites à travers ce type de message
— KEEPALIVE : il s'agit d'un message régulier qui permet de faire savoir que le routeur est toujours en état de fonctionnement. Dans la norme BGP, ce message est envoyé toutes les 30 secondes. Si trois KEEPALIVE consécutifs sont manqués, le routeur est considéré en panne.

L'accès d'un point A vers un point B ne se fait quasiment jamais par un seul chemin. Ainsi, BGP offre des options et un algorithme de priorité de route afin d'orienter le trafic lorsque plusieurs routes existent entre A et B. Cela est utile dans le cadre d'ingénierie de trafic ou bien de politique de routage. Grâce à certains de ces paramètres, l'administrateur peut préférer ou exclure des routes. Derrière ces aspects réseaux, il existe des aspects économiques puisqu'un opérateur peut avoir des accords avec d'autres acteurs et donc préférer utiliser leurs AS comme route. C'est pour cela que BGP est un protocole administré manuellement et statiquement. Voici les différentes options proposées par le protocole, triées par ordre de priorité :

— Métrique interne : on cherche à passer par la route la plus courte au sein de notre AS
— Longueur de l'AS path : on cherche à passer par le moins d'AS possible. Dans le cadre d'ingénierie de trafic, on peut utiliser cet élément pour favoriser des AS, en utilisant le bourrage pour allonger le chemin le plus court. Sur un principe similaire, on peut également écarter des AS de la route en allongeant artificiellement l'AS path. Cet élément permet également la gestion des boucles BGP. Si un routeur BGP reçoit un AS path qui contient son numéro d'AS, alors, une boucle s'est créée et la route est rejetée
— Multi-Exit Discriminator (MED) : lors de multi-homing (plusieurs liens qui connectent deux AS), chacun des liens se voit attribuer un MED. BGP emprunte alors le lien qui possède la plus petite valeur de MED
— Préférence sur le niveau de lien : on peut favoriser majoritairement l'i-BGP ou l'e-BGP
— Pérennité des routes : BGP sélectionne les routes les plus anciennes
— ID du routeur : il s'agit du dernier critère évalué. On sélectionne le plus petit ID du routeur.

Après avoir vu en détail, le protocole BGP qui semble répondre à des problématiques similaires aux nôtres mais dans des réseaux filaires administrés, nous allons voir que sa transposition dans les réseaux MANET est loin d'être triviale.

2.5.2 Non-adaptabilité de BGP aux réseaux MANET

Comme souligné précédemment, BGP répond à beaucoup de besoins de fonctionnement pour des communications tactiques. Il n'est cependant pas adaptable tel quel aux réseaux MANET. En effet, dans [BCHK06], les auteurs se placent dans le contexte d'un théâtre d'opération tel que l'on en trouve aujourd'hui. Dans cet environnement ad hoc mobile tactique, les auteurs indiquent les contraintes d'un tel réseau : faible qualité de communication, gêne de l'environnement tel que des montagnes, mobilité des troupes, risque d'interception... (Figure 2.5 extraite de [BCHK06]).

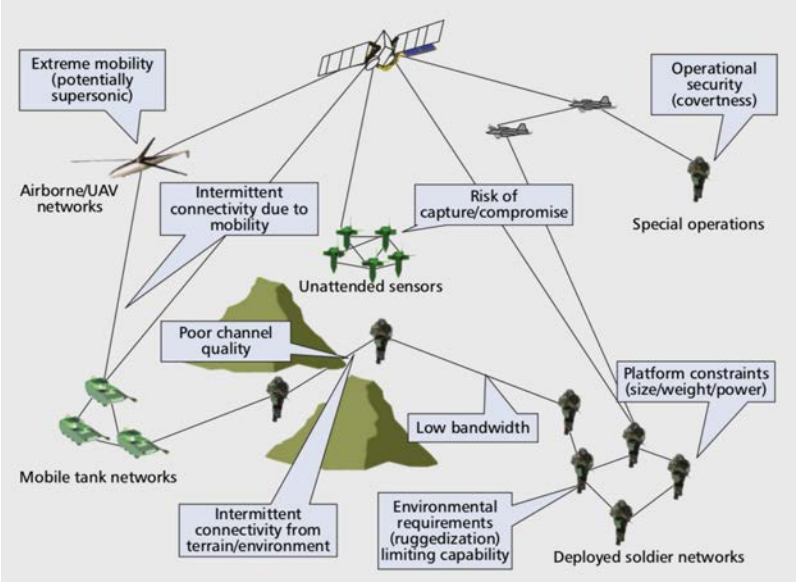


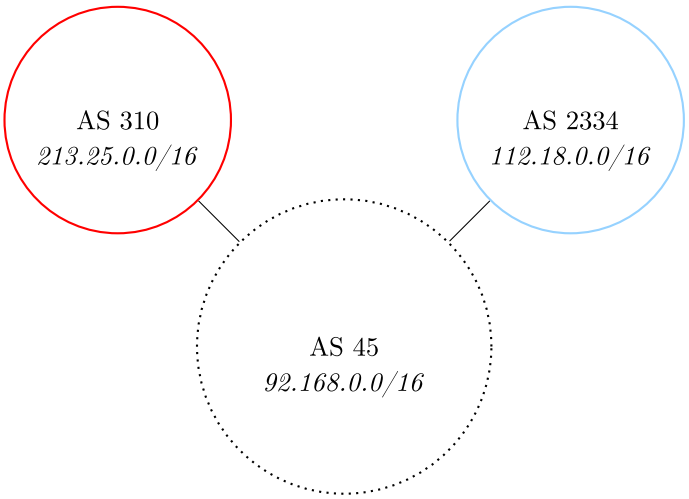
FIGURE 2.5 – Dégradations de performances sur un réseau tactique opérationnel

Ils montrent ainsi que les AS, tels qu'ils existent dans BGP, ne peuvent pas être efficaces dans un tel contexte. En effet, dans BGP, ce sont des groupes de communication créés manuellement par l'administrateur réseau et qui n'ont pas vocation à être modifiés fréquemment. La création ou suppression d'un AS dans BGP est souvent la raison d'une décision politique, économique ou stratégique (ingénierie de trafic pour les accords de peering) mais est très rarement le résultat d'une cause technique. De plus, BGP ne gère pas dynamiquement l'arrivée de nouveaux routeurs dans son réseau. Si un nouveau routeur BGP est introduit dans le réseau, l'administrateur doit l'ajouter comme "*neighbor*" dans la configuration de tous ses autres routeurs BGP. La majorité des éléments constitutifs du réseau (numéro d'AS, voisin au sens de routage BGP, communications autorisées) est saisie par l'opérateur dans les fichiers de configurations des équipements concernés. Le protocole BGP est conçu pour des réseaux filaires stables, fiables et statiques. Les changements de topologie et les incidents sont rares. Sa conception et ses objectifs font que ses réactions sont lentes. Lors d'un incident sur le réseau, comme par exemple une perte de préfixe (routeur e-BGP d'un AS injoignable), BGP détecte et reconstruit une nouvelle route en 90 secondes, ce qui est relativement lent dans notre contexte d'utilisation. Pour rappel, nous travaillons dans le cadre de réseaux tactiques militaires où les MANET ont une très grande mobilité, la topologie est très dynamique et il y a un changement fréquent des routes. Ainsi, la nature statique de BGP fait qu'il n'est pas adaptable dans un environnement dynamique tel qu'un réseau tactique militaire. Avec cette constatation, Burbank *et al.* préconisent de travailler sur les cinq critères suivants (Table 1 de [BCHK06]) pour réaliser un protocole Inter-MANET efficace :

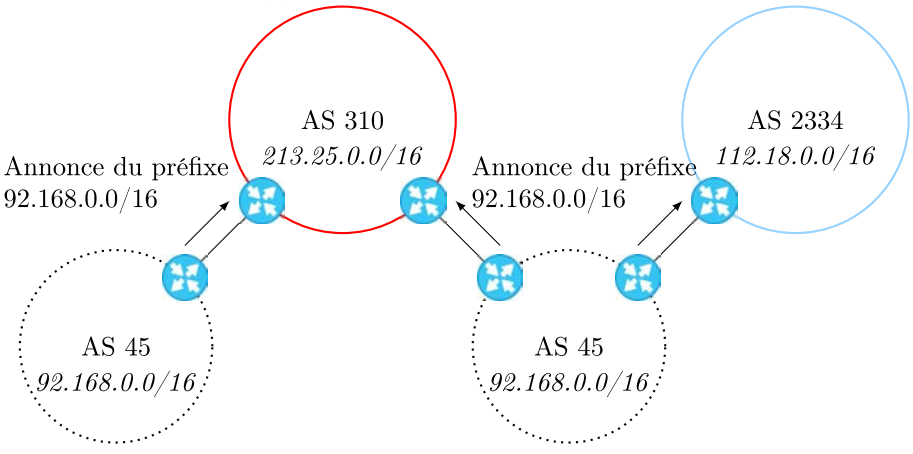
1. Découverte, authentification et connexion rapide aux voisins proches
2. Capacités d'auto-configuration des nœuds
3. Confiance dans les tables de routage *i.e.* gérer efficacement les pertes et retours de connexion
4. Protection de l'information et authentification forte, notamment pour lutter contrer des attaques byzantines

5. Possibilité de gérer plusieurs liens parallèles (*multihoming*) pour un couple source-destination donné

En plus de la mobilité permanente et de la lenteur de réaction du protocole BGP, [CCLW08] met en avant un autre problème qui est le problème de la duplication. Cette duplication peut être présente à 2 niveaux : sur le préfixe IP ou sur le numéro d'AS (son identifiant). Dans le premier cas illustré par la Figure 2.6a, trois AS sont interconnectés. L'AS 45 peut directement joindre les AS 310 et 2334. Si l'AS 310 veut joindre l'AS 2334, il utilise l'AS path 45-2334, et réciproquement lorsque l'AS 2334 veut joindre l'AS 310. Dans le cadre des réseaux MANET, les réseaux peuvent être amenés à se diviser. Ainsi, une division de l'AS 45 peut survenir et donner une nouvelle topologie illustrée sur la Figure 2.6b. Cependant, les adresses IP annoncées par les routeurs de bordure BGP ne sont pas modifiées puisque ces annonces sont paramétrées de manière statique par l'administrateur. Il y a donc une duplication sur les préfixes IP annoncés. Lorsque l'AS 310 cherche à joindre l'AS 2334, il sait que le réseau passerelle est 92.168.0.0/16. Cependant, tous les nœuds n'enverront pas les paquets de la même manière. Ceux qui auront une distance plus courte, au sens du protocole de routage intra-domaine, par rapport à l'AS 45 gauche enverront leurs paquets vers cet AS qui n'a plus d'accès à l'AS 2334. La mobilité des AS crée donc un premier problème qui se situe au niveau de la diffusion des préfixes IP.



(a) Situation initiale, 3 AS distincts



(b) Scission de l'AS 45

FIGURE 2.6 – Duplication d'annonce de préfixe IP en cas de scission d'un AS

Chau *et al.* [CCLW08] se sont alors interrogés sur le fonctionnement du réseau si les routeurs étaient capables de reconfigurer leurs annonces de préfixes suite à une scission. Dans l'exemple de la Figure 2.7, les deux sous-AS se sont reconfigurés sur des préfixes plus longs afin d'éviter la

duplication IP. Cependant, il existe une recherche de chemin au niveau BGP également à travers l'AS path. Les problèmes de communications peuvent alors survenir de deux manières différentes. Tout d'abord, pour joindre l'AS 2334, les membres de l'AS 45 de préfixe 92.168.1.0/24 ont l'AS path, calculé par le protocole BGP, suivant : 45-310-45-2334. Le protocole BGP supprimera automatiquement cette route puisqu'il considérera qu'une boucle de routage BGP est présente car il y a deux fois l'AS 45 sur la route. Ensuite, l'AS 310 peut se tromper dans l'envoi des paquets, tout comme pour la duplication de préfixes IP. Dans sa table de routage, il sait que pour aller à l'AS 2334, il faut passer par l'AS 45. Cependant, il peut envoyer ses paquets vers l'AS 45 de préfixe 92.168.1.0/24. Les paquets n'arrivent donc pas à destination.

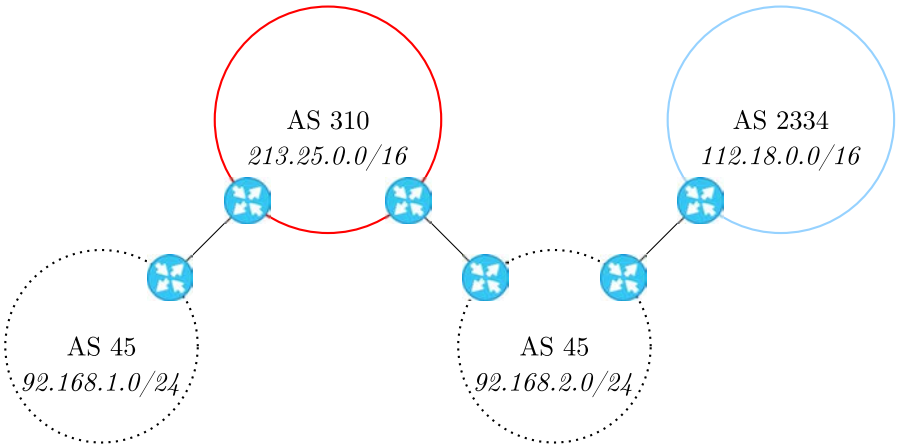


FIGURE 2.7 – Duplication d'identifiant d'AS en cas de scission d'un AS et de reconfiguration des préfixes IP

Après avoir présenté les réseaux ad hoc, le fonctionnement des communications inter-domaine dans les réseaux filaires et les problèmes majeurs dans leur adaptation directe à l'environnement réseau tactique militaire, nous étudierons l'état de l'art des protocoles de routage Inter-MANET ayant été proposés dans la littérature. Nous présenterons les problèmes adressés par ces protocoles et nous mettrons en lumière ceux qu'ils restent à traiter. L'état de l'art se composera de deux parties. Tout d'abord, nous verrons des algorithmes et mécanismes qui proposent de résoudre les problèmes d'adressage (duplication, reconfiguration) puis, des protocoles mettant certains de ces mécanismes en œuvre pour proposer des communications Inter-MANET fonctionnelles et efficaces.

Chapitre 3

État de l'art

Dans ce chapitre, nous présentons l'état de l'art de la littérature concernant le domaine d'étude de la thèse. Nous présentons un ensemble de travaux qui ont été effectués dans ce domaine afin de répondre à la problématique du routage inter-domaine dans les réseaux ad hoc.

3.1 État de l'art : algorithmes et protocoles

Après avoir présenté les réseaux ad hoc, le protocole BGP et ses problèmes d'adaptation aux réseaux MANET, nous allons voir les différentes propositions qui ont été faites dans la littérature pour assurer des communications inter-MANET. Il existe d'abord des outils permettant de résoudre les problèmes d'adressage IP et d'identifiant d'AS dupliqués, mais également des protocoles complets permettant d'établir des échanges Inter-MANET.

3.1.1 DAD : Duplicate address detection

Comme expliqué dans la partie 2.5.2, l'un des problèmes majeurs du routage Inter-MANET est le risque d'annonce d'adresses IP dupliquées au sein du réseau. Dans [BLMA05], les auteurs traitent de la détection de duplication d'adresse ainsi que d'autoconfiguration sur des réseaux mobiles routés par le protocole proactif OLSR. Dans un premier temps, Boudjit *et al.* présentent les cas où des nœuds d'un réseau mobile doivent s'autoconfigurer, et donc les cas où une duplication d'adresse IP peut potentiellement arriver :

- Le nœud arrive dans le réseau et se configure avec une adresse IP déjà utilisée
- Dans une séparation de MANET, si des nœuds isolés viennent à s'approcher d'un autre réseau, il peut y avoir un risque que les plages d'adresses utilisées soient les mêmes (dans le cas où les deux MANET ne se sont jamais entendus sur un plan d'adressage)
- De manière similaire, si deux MANET viennent à fusionner, ils peuvent entrer en conflit d'adresse s'il n'y a pas eu d'entente préalable

Deux manières de détecter la duplication d'adresse IP sur un réseau ad hoc sont citées dans cet article :

1. A l'aide de protocole de distribution d'IP (DDHCP[NP02] ou DCDP[MDMD01]) qui veillent en permanence sur les adresses IP allouées, disponibles ou bien libérées
2. Avec un mécanisme réactif, proposé par les auteurs, appelé DAD (Duplication Address Detection). Une fois un nœud arrivé et configuré, il envoie ce message avec son adresse IP. Si un nœud avec la même adresse IP reçoit ce message, la duplication est détectée. Les nœuds se mettent donc d'accord sur celui qui changera son IP (généralement la plus petite adresse MAC). Une fois qu'un des nœuds a changé d'adresse IP, il relance le DAD pour s'assurer une nouvelle fois de la non-duplication.

Les auteurs se sont intéressés à ces mécanismes de détection de duplication sur les réseaux OLSR spécifiquement. Ils travaillent en particulier sur les aspects d'optimisation et de résolution de duplication. Le but final est d'économiser la bande passante du réseau qui est assez limitée dans

les réseaux sans fil. Leur proposition est d'utiliser le réseau des MPR (MultiPoint Relay) OLSR qui connaissent parfaitement la topologie du réseau et l'emplacement des différents nœuds. Cela évite d'envoyer les paquets MAD (Multiple Address Declaration) en broadcast et de surcharger inutilement le réseau. L'évaluation de l'overhead généré par DAD est entièrement théorique. Le scénario utilisé comprend un réseau de 10.000 nœuds et un nombre d'adresses encore disponibles de 2^{48}. Le résultat de leurs calculs indique que l'utilisation de DAD entraîne une charge en octets de 33% supplémentaires (160 bits) par paquets.

Cette proposition nous semble très intéressante sur deux aspects. Tout d'abord, les auteurs travaillent sur le protocole de routage OLSR qui constitue notre cadre de travail. Ensuite, la surcharge utilisée pour mettre en place la détection de duplication est faible en nombre de bits. Le message MAD étant envoyé avec les messages TC, si l'on suit la norme OLSR qui conseille un intervalle d'émission de 5 secondes, la bande passante nécessaire est seulement de 320 bits/s.

3.1.2 BGP Dynamic AS reconfiguration

L'adressage IP n'est pas le seul élément qui peut causer des problèmes de boucle et de duplication. C'est pour cela qu'un autre type de reconfiguration a été proposé dans [HB07]. Cet article propose un mécanisme supplémentaire au protocole BGP afin qu'il puisse se reconfigurer seul, notamment au niveau des numéros d'*Autonomous System* (AS) en cas de séparation ou de fusion. Hares *et al.* présentent dans un premier temps les problématiques engendrées par la mobilité des AS, dans l'implémentation actuelle de BGP. Le fait d'annoncer les routes passant par le même AS plusieurs fois est considéré comme un bouclage et est détruit par le protocole. De plus, les AS se situant entre deux identifiants identiques envoient leurs annonces sur la mauvaise route. Cependant, BGP a une option permettant d'autoriser les boucles. Mais sur ce principe, les annonces de routes peuvent devenir très longues s'il n'y a aucune analyse des numéros d'AS. Face à ces constatations, les auteurs mettent en avant l'incapacité de BGP à détecter la séparation et la fusion d'AS. Le soucis réside dans le fait que le nœud devrait connaître tous les membres de son AS ainsi que ceux des autres AS. Cela nécessite donc une intervention manuelle avec des informations statiques, ce qui est à l'opposé de ce que nous souhaitons faire dans un réseau ad hoc tactique.

La solution proposée est donc amenée et décrite comme suit. Dans un premier temps, chaque nœud, avec l'aide de son protocole de routage interne, est capable de déterminer les nœuds qui appartiennent à son groupe et donc ceux avec qui il forme un AS (puisque sous un même protocole de routage interne). Si par le protocole interne, le nœud détecte une rupture de liens avec ses voisins, il sait que son AS a été divisé. Un paramétrage amont dans chaque nœud intervient alors, il s'agit d'un ID d'AS unique préalablement configuré. Si un nœud se retrouve seul, il se reconfigure sur ce numéro d'AS. Pour éviter d'avoir autant de numéros d'AS que de nœuds, on agrège si possible le numéro d'AS. L'exemple donné dans [HB07] est le suivant : 4 nœuds A, B, C et D ont les ID unique d'AS respectifs 65001, 65002, 65003, et 65004. S'ils constituent chacun un AS, ils auront ces identifiants. S'ils viennent à fusionner à quatre, ils se regroupent sous un numéro d'AS 65000, agrégat de leurs quatre AS uniques. Le changement d'AS est aussitôt indiqué aux voisins connus des nœuds intéressés. Dans le cas où la séparation venait à être sur le modèle A-B-C/D ou bien A-B/C-D, les nœuds encore en communication se mettent d'accord sur l'AS à prendre (aucun modèle algorithmique n'est proposé dans le papier).

L'identifiant de l'AS est un problème si on veut adapter tel quel BGP pour les réseaux MANET. Avec cette méthode, les nœuds sont capables localement de se reconfigurer dans le cas d'une mobilité, contrairement au protocole BGP original où l'intervention d'un administrateur réseau est nécessaire. Cependant, l'agrégation proposée semble être assez rigide puisqu'elle porte sur des numéros d'AS consécutifs. Ainsi, on peut se retrouver avec des négociations d'identifiants fréquentes qui encombrent le réseau.

3.1.3 Filtre de Bloom

Après avoir vu les mécaniques pour éviter les problèmes d'adressage IP et d'identifiant d'AS dupliqués, nous étudions la proposition faite dans [Blo70, BM04] d'un système d'adressage alternatif à base de filtre de Bloom. Le filtre de Bloom est une structure de taille fixe, matérialisée sous forme d'un tableau de bits. Son but est d'optimiser l'espace utilisé pour représenter la plus grande quantité d'information possible. Généralement, il s'agit d'inscrire des informations dans le filtre, l'envoyer aux membres du groupe pour qu'ils puissent tester la présence d'informations ou y inscrire des nouvelles. C'est une structure probabiliste qui maximise le compromis entre l'utilisation de l'espace mémoire, la vérification d'appartenance d'un ou plusieurs éléments et la probabilité d'erreur (*c-à-d* les faux-positifs : croire que le message est présent dans le filtre alors qu'il n'y est pas).

Ce filtre est généralement utilisé sous la forme d'un vecteur de bits, initialisé à 0. Voici les étapes à suivre, résumées sur la Figure 3.1, si l'on veut inscrire un message :

— on choisit plusieurs fonctions de hachage (numérotées de 1 à k) dont l'espace d'arrivée est de taille égale au vecteur de bits (pour un filtre d'un taille de $2^{10} = 1024$ bits, la fonction de hachage doit donner une valeur de 10 bits),
— lorsque l'on veut insérer un message, on hache le message m par toutes les fonctions de 1 à k,
— chaque hach indique la position du bit à mettre à 1 dans le vecteur (k bits sont donc mis à 1). Si la position contient déjà un 1, il n'y a pas de modification.

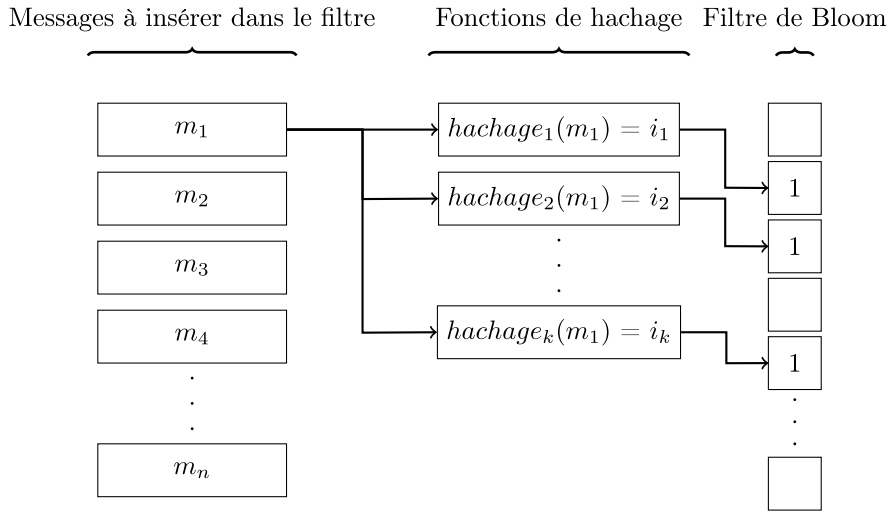


FIGURE 3.1 – Fonctionnement du filtre de Bloom

Lorsque l'on veut vérifier la présence d'un message dans le filtre, les opérations à effectuer sont similaires et sont illustrées par la Figure 3.2.Il faut hacher le message à travers un nombre défini de fonctions. Une fois tous les hachés obtenus, on teste la présence de la valeur 1 à chacune des positions obtenues.

Le filtre de Bloom possède la propriété de ne pas avoir de faux-négatifs *c-à-d* que la non-présence du message est sûre s'il n'y a pas tous les bits correspondants à 1. Cependant, il est possible d'avoir des faux-positifs. Ce type de cas, représenté par la Figure 3.3, se présente lorsque le test d'un message est réussi, alors que un ou plusieurs autres messages ont contribué ensemble au passage à 1 des bits vérifiés.

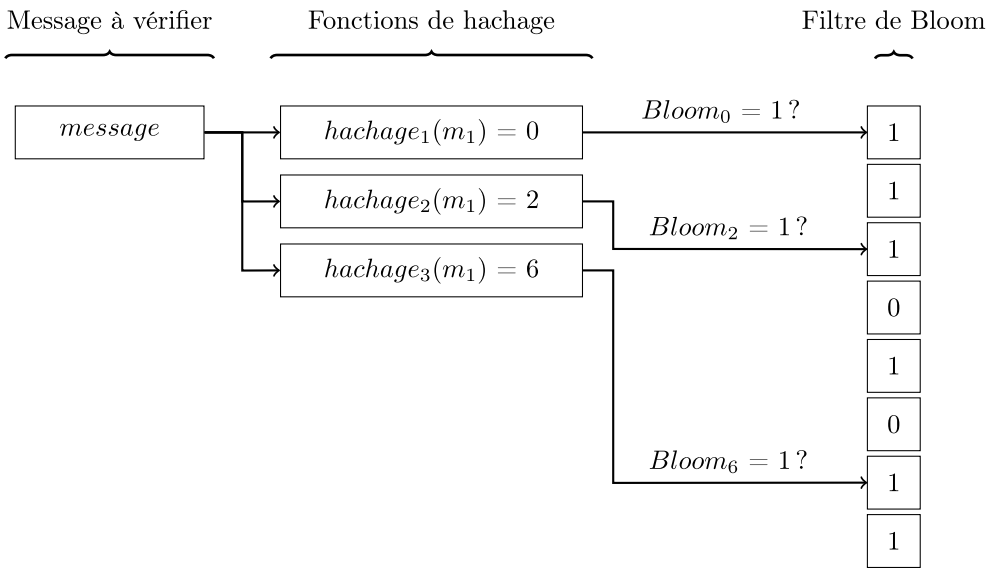


FIGURE 3.2 – Test d'appartenance d'un message dans un filtre de Bloom

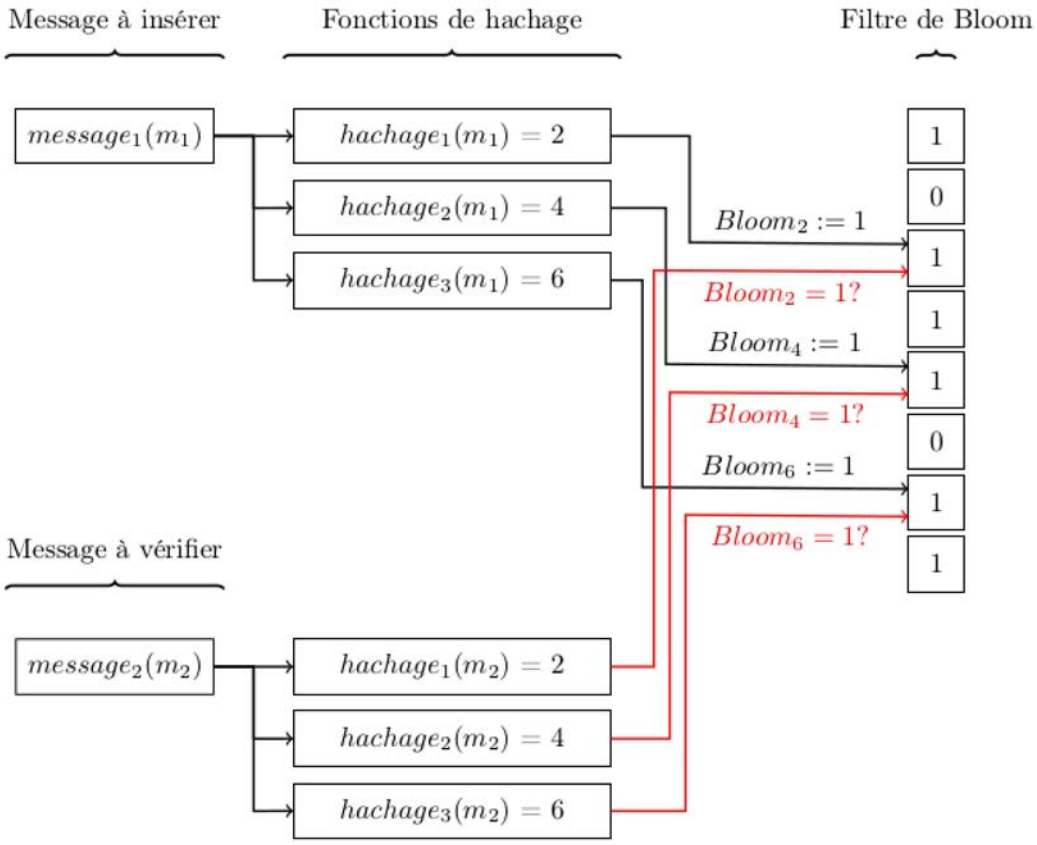


FIGURE 3.3 – Collision de messages dans un filtre de Bloom

En utilisant la structure du Filtre de Bloom comme elle est définie, il est impossible de retirer un message. En effet, le filtre ne cumule pas les messages sur chaque position. Il est donc impossible de savoir si un ou plusieurs messages ont assigné la valeur 1 aux bits du vecteur.

Voici un résumé des propriétés à retenir sur le filtre de Bloom :

— on ne peut pas retirer de message car on ne sait pas si un ou plusieurs messages ont contribué

au passage à 1 d'un bit du vecteur,
— les faux-négatifs ne sont pas possibles. La non-présence du message est sûre s'il n'y a pas tous les bits correspondants à 1,
— cependant, des faux-positifs peuvent survenir. Plusieurs messages peuvent donner les mêmes positions sur le vecteur. Les variables telles que la taille du filtre ou le nombre fonctions de hachage à utiliser déterminent la probabilité d'apparition des faux-positifs.

Dans la littérature, le filtre de Bloom est utilisé pour offrir une solution alternative à l'adressage IP. En effet, chaque groupe possède un filtre de Bloom, qui est rempli par chacun de ses membres avec un attribut unique comme message. Cela peut être son adresse IP (en supposant qu'il n'y ait pas de duplication) ou bien son nom. Ainsi, les filtres sont diffusés sur le réseau. Quand un nœud veut joindre un allié, il cherche parmi les filtres qu'il possède la marque de l'allié. Il sait ainsi dans quel AS il se situe.

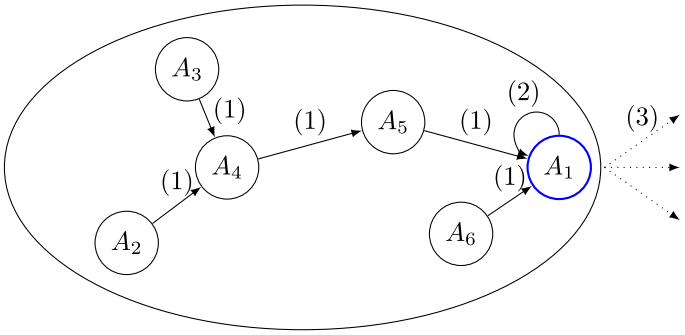
Nous pensons qu'un système alternatif d'adressage est intéressant, notamment en termes de consommation énergétique et de puissance de calcul à utiliser. En associant à chaque nœud un identifiant unique à inscrire dans le filtre de Bloom, il est alors possible d'associer un filtre à un MANET et d'avoir l'ensemble des membres présents dans le groupe. Cependant, quelques problèmes subsistent sur son fonctionnement, notamment dans le cas d'une séparation de MANET. Lorsqu'un nœud quitte un AS, il faut que le filtre de Bloom soit mis à jour. Or, il s'agit d'une structure où on ne peut pas retirer les éléments puisque l'on pourrait effacer l'empreinte d'un autre membre. Le seul moyen de mettre à jour ce filtre, suite à une séparation ou une perte de connexion, serait de le recalculer entièrement avec les nouveaux membres. Cela doit être fait à intervalle régulier pour assurer une validité récente du filtre, ce qui peut au final se révéler un défaut pour maintenir les informations à jour.

Après avoir passé en revue les algorithmes et propositions permettant d'améliorer les problématiques d'adaptation du protocole BGP, nous allons voir les différents protocoles qui ont été publiés sur le thème des communications inter-domaine tactiques.

3.1.4 InterMR : Inter-MANET routing

Dans [LWC+10], les auteurs proposent le protocole Inter-Manet Routing (InterMR). Pour atteindre les groupes voisins et réaliser des échanges inter-MANET, le protocole InterMR utilise des passerelles dans chaque MANET. Au sein d'un même MANET, il existe trois types de nœuds : les nœuds simples sans fonctionnalité spécifique, les passerelles non-actives qui agissent comme les nœuds simples et les passerelles actives qui sont chargées des communications inter-MANET. Ces passerelles actives et non-actives sont choisies en début de mission en fonction des mobilités connues de la mission, pour conserver la connectivité intergroupe au fil du temps. Pour séparer les passerelles actives et non-active, le principe de ratio a été mis en place. Ce nombre est également défini en préparation de mission. Ainsi, dans un MANET de 20 nœuds dont 10 passerelles et un ratio fixé à 0.5, InterMR élit cinq passerelles dans le MANET. Ces passerelles ne sont pas fixes au fil de la mission. Il existe une élection dynamique qui permet de changer les passerelles lorsque la topologie évolue. Le critère utilisé pour élire les passerelles actives est l'accessibilité vers les autres groupes. Plus il y a de destinations accessibles par une passerelle, plus sa probabilité d'être élu active augmente.

L'adressage du réseau, qui est une problématique majeure des communications inter-MANET, repose sur un filtre de Bloom. Chaque nœud du MANET va utiliser un ou plusieurs attributs (*hostname*, adresse MAC, adresse IP, ID unique...) pour les insérer dans ce filtre de Bloom. Cet attribut va être envoyé à la passerelle qui va être en charge de calculer le filtre de Bloom. Cela permet de mettre en évidence une dénomination pour l'ensemble des membres du MANET puisque chacun contribue à la construction du filtre. Le filtre de Bloom agit donc comme un serveur de nommage pour les différents MANET connectés sur le réseau. L'ensemble des étapes de construction du Filtre de Bloom est résumé sur la Figure 3.4.



(1) Envoi à la passerelle de l'identifiant unique
(2) Calcul du Filtre de Bloom représentatif du MANET
(3) Diffusion du Filtre de Bloom aux autres passerelles

FIGURE 3.4 – Construction du Filtre de Bloom d'un MANET

Le routage est fait de manière identique à BGP. D'un côté, le routage interne (i-InterMR) se charge des communications entre les passerelles du MANET. De l'autre, le routage externe (e-InterMR) est utilisé par les passerelles afin de communiquer avec les MANET voisins. Quand un paquet doit être envoyé, le nœud consulte sa table de routage InterMR et adopte la meilleure route possible (MANET à traverser, longueur de chemin) si la destination est connue. Un exemple de réseau InterMR et de propagation de route est illustré par la Figure 3.5.

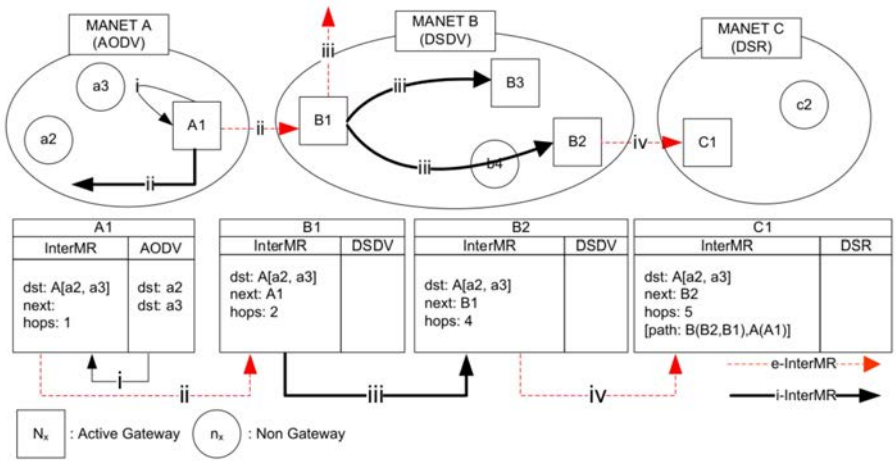


FIGURE 3.5 – Propagation des routes du MANET A à travers le réseau via InterMR

Dans InterMR, le routage réactif est utilisé. Cependant un système de beaconing est créé et des paquets sont envoyés à intervalle régulier. Ils servent à construire le filtre de Bloom, s'échanger les informations de routage entre passerelles et procéder aux élections des passerelles actives et non-actives.

Ces mécanismes permettent d'appliquer des politiques de routage si cela est souhaité par les hiérarchies des différents MANET. InterMR est capable de gérer la mobilité, notamment la division ou fusion de MANET. Ce protocole a été construit à partir de travaux de Master Recherche nommés InterDomain Routing for MANET (IDRM) et créés par les mêmes auteurs[CCLW08].

Comme nous avons pu l'annoncer pour le filtre de Bloom précédemment, nous voyons difficilement comment il est mis à contribution efficacement dans le InterMR. Aucune information n'est donnée concernant ses mises à jour et la suppression de nœuds. De plus, chaque modification de topologie qui survient dans un groupe entraîne aussitôt un nouveau calcul du filtre de Bloom

pour les membres restants, ce qui semble une lourde contrainte sur les ressources à utiliser. Enfin, InterMR fonctionne à l'aide de protocoles réactifs, auquel une routine d'envoi de *beacon* est ajoutée. Enfin, aucune notion de sécurité n'est abordée.

3.1.5 BGP-MX : BGP - Mobility eXtension

Deux évolutions de BGP ont été proposées. Dans la première que l'on peut consulter dans [KTRH11], Kaddoura *et al.* proposent le protocole BGP-MX (BGP Mobility eXtension). Le but est de conserver le protocole BGP tout en lui proposant des améliorations permettant l'utilisation sur des réseaux sujets à la mobilité.

La contribution majeure de cette proposition est un serveur nommé DBPS (Distributed Peer Broker Service) qui est accessible par le réseau. Son rôle va être de superviser, organiser et interconnecter les différents AS du réseau. Pour cela, chaque nœud est doté d'un capteur de position GPS. Il possède également, tout comme le protocole BGP original, un numéro d'AS chargé au lancement du réseau. De manière régulière, chaque nœud envoie sa position GPS ainsi que sa vitesse au DPBS. Ainsi, les auteurs font l'hypothèse que la majeure partie du temps, il existe un chemin entre chaque nœud et le DPBS.

Lorsqu'un nœud arrive dans le réseau, il réalise deux opérations : l'apprentissage de la route vers le DPBS et la découverte de ses voisins par i-BGP. Le DPBS se charge de l'interconnexion des AS. Pour cela, il utilise les positions GPS des nœuds pour déterminer la topologie du réseau. Lorsque deux AS sont à portée, les nœuds informent le DPBS qu'un AS voisin est proche et attendent sa réponse pour connaître les opérations à effectuer. Dans [KTRH11], deux politiques de routage sont expliquées : une connexion entre deux AS avec un seul lien ou avec un nombre non-limité. Si la politique inter-AS à appliquer est la création d'un lien unique et qu'il existe plusieurs liens possibles entre les deux AS concernés, alors, le DBPS choisira la distance la plus courte à l'aide des coordonnées GPS qu'il a pu collecter comme le montre la Figure 3.6. Les passerelles A et C forment le segment le plus court, donc c'est le lien qui sera établi entre les deux AS. Par la réception périodique des positions des nœuds, il est capable de mettre à jour la route à utiliser si deux autres passerelles se rapprochent et créent donc un lien plus court. Lorsque le DPBS devient injoignable, les passerelles passent par un système de découverte automatique, via des beacons.

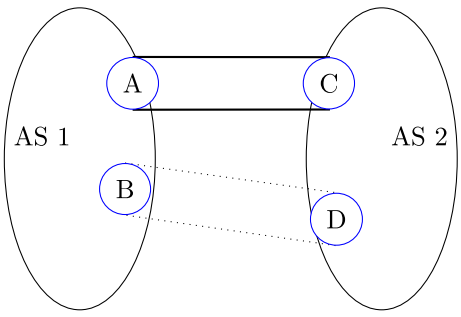


FIGURE 3.6 – Politique de lien unique : détermination des passerelles par la distance la plus courte

Kaddoura *et al.* proposent également un re-routage accéléré qui passe par une modification dans la gestion des routes. Dans la transmission de ces informations, chaque nœud ajoute un index de mobilité. Ainsi, plus l'index indiquera que la route est sujette à la mobilité, moins les passerelles l'utiliseront, bien que ce soit le plus court. BGP-MX favorise la stabilité des routes, au sens mobilité, plutôt que le chemin le plus court. L'expérience pour évaluer cet index de mobilité met en œuvre 3 AS comme indiqué sur la Figure 3.7 qui forment une topologie circulaire. Le DPBS, joignable par l'AS 2, fait communiquer les routeurs 1 et 8, situés dans les AS 1 et 3, par le chemin 1-3-7-8. Le lien 3-7 tombe en panne et le test mesure le délai nécessaire à la reconstruction de la route sur les nœuds 1-2-4-6-7-8.

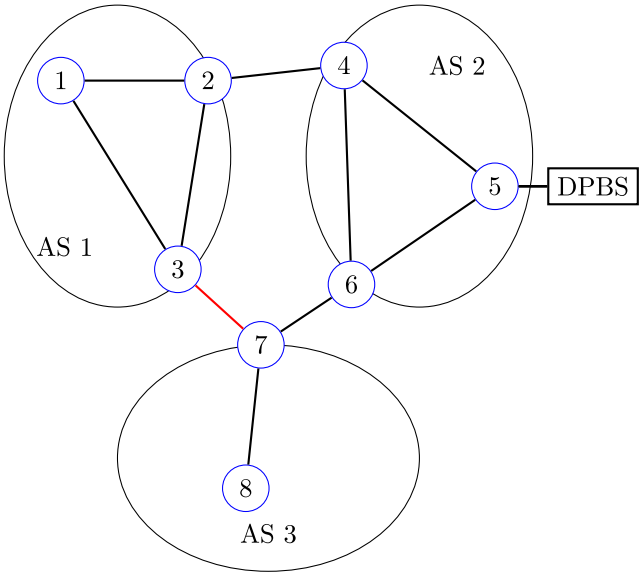


FIGURE 3.7 – Expérience d'évaluation de réparation de route

L'efficacité de ce système est évaluée sur le délai à trouver une route alternative. Ce délai comprend le temps de détection de la panne et la reconstruction de la route. Deux paramètres variables sont utilisés : l'intervalle d'émission des beacons et le nombre de beacons manqués pour considérer la route comme défectueuse. L'ensemble des résultats produits par Kaddoura *et al.* [KTRH11] sont présentés par la Figure 3.8

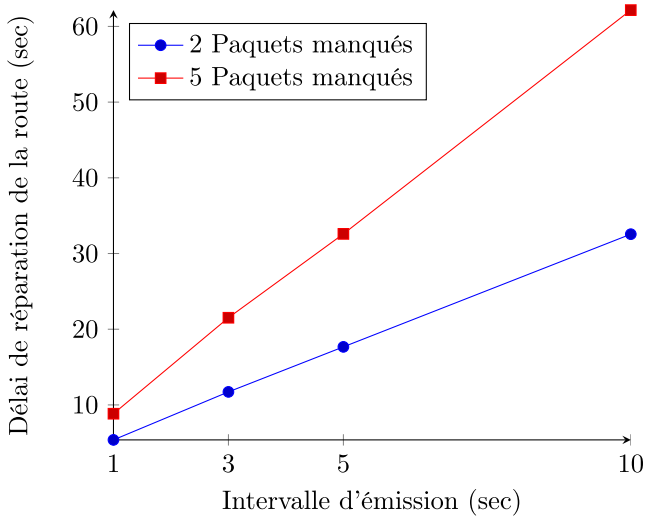


FIGURE 3.8 – Délai de réparation de route en fonction d'intervalle d'émission des beacon de découverte et du nombre de paquets manqué

Le protocole BGP-MX est un protocole rapide pour la détection de voisin et la réaction aux pannes face à son modèle, BGP. En effet, avec des intervalles de découverte qui peuvent sembler lents (10 secondes) ou bien une détection peu rapide d'une rupture de lien (5 paquets manqués), le protocole BGP-MX affiche des performances nettement supérieures au protocole BGP original. BGP met 90 secondes dans la détection de panne, là où BGP-MX est capable de réparer la route en 62 secondes. Cependant, les problèmes de mobilité comme les fusions et divisions d'AS ne sont pas gérées à cause de la nature encore trop statique de BGP-MX. Les nœuds passerelles sont prédéterminés en début d'opération et il est donc très difficile de mélanger les AS. Malgré ses apports et l'amélioration notable des performances en comparaison avec le protocole BGP original,

le DPBS reste une forte hypothèse qui nécessite une connectivité permanente au sein du réseau tactique ad hoc, bien que des mécaniques existent si un nœud n'y a pas accès. Nous n'envisageons donc pas l'utilisation de la solution BGP-MX car il ne possède pas une architecture entre distribuée et autonome. Tout comme pour InterMR, les questions de sécurité ne sont pas abordées.

3.1.6 BGP-MR : BGP - MANET Routing

La deuxième évolution de BGP proposée est le protocole BGP-MR (BGP Manet Routing) décrit dans [OKG14]. BGP-MR fonctionne en collaboration avec le protocole de routage interne OSPF-MDR[OS09], une extension d'OSPFv3 [CFML08]. OSPFv3 permet d'identifier chaque routeur du réseau avec un ID unique, qu'il soit en IPv4 ou en IPv6. Ainsi, deux routeurs peuvent devenir voisins, quelque soit l'adressage employé. Les trois extensions apportées à OSPFv3 par le protocole OSPF-MDR sont les suivantes :

— Les annonces de routes (Link State Advertisement - LSA) sont allégées pour réduire l'encombrement réseau,
— Les domaines sont des Connected Dominating Set (CDS) où sont élus des MANET Designated Router (MDR) et des Backup MANET Designated Router (BMDR). Ce sont ces routeurs qui sont en charge de distribuer les LSA dans le réseau,
— Les CDS sont délimités de telle manière que tous les nœuds du réseau ne soient pas à plus d'un saut d'un MDR ou d'un MBDR.

BGP-MR s'appuie sur les MDR pour joindre les nœuds et réagir rapidement à une modification du réseau. Un pourcentage des nœuds du réseau, défini par un ratio fixe au lancement du réseau, est considéré comme une passerelle et peut potentiellement échanger des informations avec un groupe voisin. Par exemple, si un MANET comporte 100 nœuds et que son ratio est défini sur 0.2, il y a 20 nœuds qui seront désignés comme passerelle. Les auteurs ne donnent pas de recommandation sur le ratio à adopter entre nœuds simples et passerelles. Deux états permettent de différencier les passerelles du réseau : un état actif et un état passif (Figure 3.9).

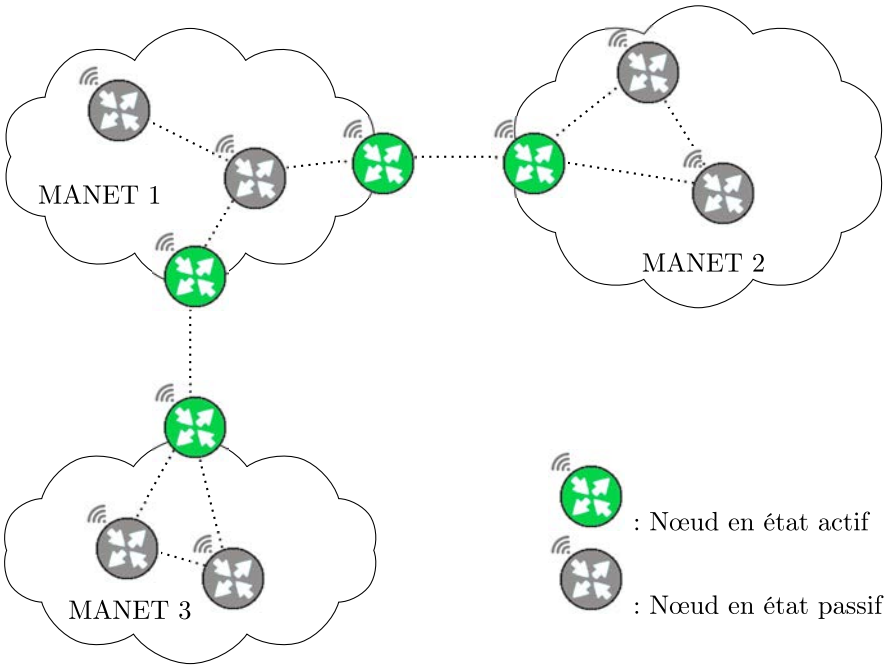


FIGURE 3.9 – Illustration de l'état actif-passif des passerelles BGP-MR

En mode passif, le nœud-passerelle agit comme un simple routeur et se charge de la transmission des paquets. Il reste cependant attentif au beaconing régulier du protocole OSPF-MDR. S'il reçoit

un message d'un nœud appartenant à un autre AS, il passe en état actif et échange les routes avec son nouveau voisin. Lorsque ce lien est rompu à cause de la mobilité (éloignement ou division), la passerelle diffuse un message pour annoncer la suppression des routes fournies par le voisin qui vient de disparaître. Ainsi, BGP-MR évite la duplication des routes et garantit une validité récente de ses entrées. Ceci permet donc à ce protocole de supporter les opérations de fusion et de séparation d'AS. Afin de définir les membres d'un AS, BGP-MR utilise un filtre de Bloom. Il n'est pas indiqué l'élément qui est utilisé pour inscrire chaque membre dans le filtre.

L'évaluation de BGP-MR, réalisée par les auteurs de [OKG14], est basée sur sa capacité à avoir des routes valides. Six MANET restent fixes pendant 20 minutes pour apprendre la topologie réseau. Ensuite, ils entrent en mouvement (aucune spécification ou scénario n'est indiqué) et l'évaluation mesure le taux de routes valides selon l'utilisation de BGP ou BGP-MR. Ainsi, BGP-MR possède au minimum 95% de routes valides au long du scénario, là où BGP ne fait que chuter jusqu'au taux de 80%. Bien que cette performance soit satisfaisante, quelques points restent gênants : BGP-MR fonctionne avec un protocole intra-domaine donné qui est OSPF-MDR, le temps de converge est plutôt long (15 minutes pour un réseau de 200 nœuds) et aucune fonctionnalité n'existe pour la sécurité des échanges.

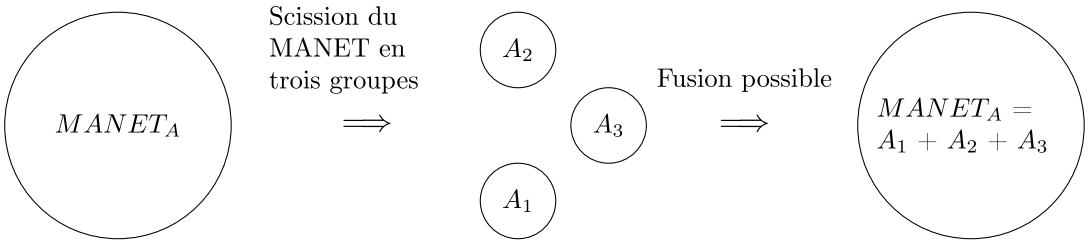
3.1.7 CIDR : Cluster-based Inter-Domain Routing

Dans [ZCG09], le protocole CIDR (Cluster-based Inter-Domain Routing) utilise une approche différente. Chaque AS est conçu comme un Cluster. Un ou plusieurs nœuds sont promus en tant que *Cluster Head* (CH) dont le rôle est de centraliser et de distribuer les informations de routage. Ce nombre est défini par un pourcentage qui définit la proportion de CH dans le groupe. Plusieurs hypothèses sont faites sur les nœuds et les groupes : chaque nœud possède un identifiant unique (l'adresse MAC est proposée), les numéros de groupes sont uniques tout au long du réseau et les liens sont bidirectionnels. Le Cluster Head agit comme un élément hiérarchique pour le domaine. Il agit à la fois comme un serveur DNS pour son groupe, mais également comme une passerelle pour les groupes voisins. Comme pour BGP, il est en charge de l'annonce de son groupe, et les destinations qui sont accessibles, que ce soit en intra-domaine ou en inter-domaine. Il est élu au sein de son groupe en fonction de sa connectivité. Cette connectivité n'est pas définie par les auteurs, il peut s'agir du nombre d'entrées dans la table de routage ou bien le nombre de groupes accessibles dans le voisinage. L'appartenance des membres à un groupe est connue à l'aide d'un Filtre de Bloom. Dans chaque groupe, le Cluster Head insère dans le filtre tous les ID des membres de son groupe. Les auteurs montrent que le Filtre de Bloom permet de réduire le volume de données de contrôle à partir de 25 membres dans un groupe.

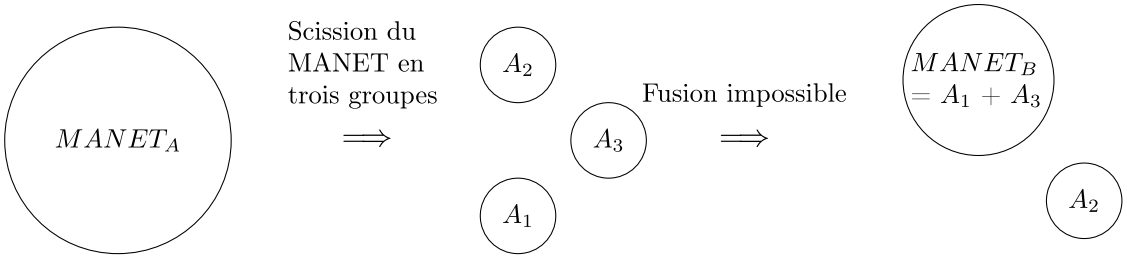
Ce système de Cluster Head fonctionne naturellement sur des réseaux proactifs, où les nœuds s'échangent régulièrement des données. Dans le cadre des réseaux réactifs, les auteurs indiquent qu'il faut mettre en place un système d'envoi de paquets de contrôle contenant les mises à jour des routes et la topologie du réseau. Les nœuds autres que ceux élus en tant que Cluster Head ne sont capables de comprendre que les paquets du ou des Cluster Head auxquels ils sont associés.

CIDR gère partiellement la mobilité de MANET. D'un côté, la division entraîne simplement de nouvelles élections de Cluster Head dans les AS générés. De l'autre, la fusion existe mais selon des critères spécifiques. Dans l'exemple de la Figure 3.10, nous présentons deux cas différents. Le premier cas, représenté sur la Figure 3.10a, montre un cas où la mobilité est pleinement gérée. Un MANET se sépare en 3 sous-MANET que nous avons nommés A_1, A_2 et A_3. CIDR élit dans chacun de ces MANET un nouveau Cluster Head et le réseau continue de fonctionner normalement. Lorsque A_1, A_2 et A_3 décident de fusionner, ils reforment le MANET existant précédemment, utilisent le Cluster Head associé et communiquent normalement. Si deux des trois sous-MANET envisagent de fusionner, comme le montre la Figure 3.10b, le protocole CIDR ne leur permet pas, dans la mesure où A_1 et A_3 ne formaient pas un MANET préalablement existant.

Ce protocole nous montre encore l'utilisation du filtre de Bloom comme adressage alternatif. Ainsi, Zhou *et al.* s'affranchissent du problème de l'adressage IP et se focalisent sur la création du protocole Inter-MANET. Cependant, tous les cas de mobilité ne sont pas gérés, particulièrement



(a) Fusion fonctionnelle de MANET séparés



(b) Fusion impossible de quelques MANET séparés

FIGURE 3.10 – Scénarios possibles de séparation puis fusion avec le protocole CIDR

sur le regroupement de MANET, qui doivent former un groupe préalablement existant.

3.1.8 Synthèse de l'état de l'art

La majorité des travaux existants montrent que la principale problématique des communications inter-MANET est l'adressage IP. Par exemple, citons [LWC+10] (InterMR) qui indique : *"Thus, the prefix based address scheme in BGP is not viable to properly aggregate IP addresses in a MANET"* et [ZCG09] (CIDR) : *"However, mobility and Ad-Hoc deployment in MANET can create arbitrary network partition, which often splits the network into parts that do not have distinct prefix across the network."*. Nous pouvons résumer ces citations ainsi : la mobilité des nœuds entraîne un nombre trop important de combinaisons possibles dans la création d'AS. L'agrégation par IP telle qu'elle existe aujourd'hui dans les réseaux filaires semble donc difficile à réaliser, d'autant plus que le sous-réseau IP ne peut plus permettre l'identification de l'AS. Ainsi, il est nécessaire d'utiliser un autre système d'adressage qui est le filtre de Bloom. Il garantit la non-duplication d'adresse et sa taille fixe permet de gérer facilement la mobilité du réseau. Nous trouvons que l'allègement de ressources qu'apporte cette solution est intéressant. Cependant, l'une des problématiques de l'état de l'art reste l'adressage des nœuds . Or, le filtre de Bloom consiste à proposer un adressage alternatif où chaque nœud utilise un identifiant unique. Il faut donc toujours mettre à jour ses données et actualiser le filtre en cas de mobilité des groupes, ce qui revient au problème initial où un nœud possède une adresse IP qui doit être unique sur le réseau et il faut garder un adressage cohérent et à jour. Il nous est donc difficile aujourd'hui de nous projeter dans l'implémentation d'un de ces protocoles puisque nous ne voyons pas la plus-value avec l'utilisation de ce Filtre de Bloom.

Afin de nous positionner et choisir comment nous allons travailler au cours de cette thèse, nous proposons dans la partie suivante une comparaison de ces protocoles pour déterminer quels sont les travaux à mener dans le domaine des communications Inter-MANET.

3.2 Positionnement de la thèse

Dans cette thèse, notre objectif est d'établir des communications entre plusieurs groupes de communications. Ces groupes de communications sont des MANET, qui sont sous des autorités différentes. Ils ont pour but d'échanger des données et collaborer dans la réussite d'une mission. Cependant, chacun est gestionnaire de son réseau et souhaite garder sa souveraineté. Ainsi, d'un

groupe à l'autre, il peut y avoir des protocoles de routage interne différents, des recouvrements et duplications d'adresses IP ce qui rend leur interconnexion difficile. De plus, la mobilité de ces réseaux fait que la topologie change fréquemment. Un modèle de fonctionnement similaire à notre situation est le protocole BGP, où différents opérateurs se lient pour construire Internet. Cependant, Burbank *et al.* montrent dans [BCHK06] que l'adaptation de BGP n'est pas possible et qu'il faut travailler avec un nouveau protocole. Les préconisations pour la conception de ce protocole qui ont été apportées sont les suivantes :

1. Découverte, authentification et connexion rapide aux voisins proches
2. Capacités d'auto-configuration des nœuds
3. Confiance dans les tables de routage *c-à-d* gérer efficacement les pertes et retours de connexion
4. Protection de l'information et authentification forte, notamment pour lutter contrer des attaques byzantines
5. Possibilité de gérer plusieurs liens parallèles (*multihoming*) pour un couple source-destination donné

Pour comparer efficacement les travaux précédemment cités, nous nous sommes appuyés sur la liste proposée par [BCHK06]. Nous y avons apporté des modifications pour ajouter des critères qui nous semblaient manquants pour se rapprocher du fonctionnement de BGP. Par exemple, nous avons ajouté la mobilité des AS (fusion/division) et l'interconnexion des domaines indépendamment du protocole de routage interne utilisé. Nous avons regroupé quelques critères que nous avons jugés similaires. Les points 1, 2 et 3 ont été globalisés sous un même critère qui est la maîtrise de l'adressage du réseau *c-à-d* il n'y a pas de risque de duplication d'adresse ou de recouvrement de préfixe. Voici la liste de critères que nous proposons ainsi que l'explication détaillée de chacun des éléments de cette liste.

— Maîtrise de l'adressage IP :
Afin de pouvoir fonctionner correctement, il ne doit pas y avoir de duplication d'adresse IP au sein du réseau. Le protocole doit pouvoir fonctionner et offrir des mécanismes de détection/correction de duplication d'adresse IP dans le cas où de nouveaux arrivants viendraient intégrer le réseau,
— Interfaçage des autres groupes :
Dans le cadre de communications avec d'autres organisations avec qui un plan d'action n'est pas défini au préalable, et en l'absence de gestion partagée, il doit être possible de communiquer même si les protocoles de routage interne sur ces différents réseaux sont différents. Il faut donc être capable de les détecter et de s'interfacer avec eux si besoin,
— Sécurisation des échanges :
Le protocole doit pouvoir proposer des transmissions des informations de routage sécurisées c'est-à-dire que les services d'authentification, d'intégrité et de confidentialité doivent pouvoir être fournis si nécessaire,
— Définition des AS : constitution et passerelles
Afin d'identifier ce que sont les différents AS, il est important d'en définir les nœuds de bordure et leurs propriétés,
— Dynamique des AS :
Que ce soit par mobilité géographique, par volonté ou déclenchement sur ordre, les AS doivent être capables de se séparer et de fusionner de manière rapide et sans coupure notable des services de communication.

Notre classification finale sur les fonctionnalités et problématiques revendiquées comme résolues dans l'état de l'art se résume dans le Tableau 3.1.

	Maîtrise de l'adressage IP	Interfaçage des autres groupes	Échanges sécurisés	Définition des AS	Dynamique des AS
BGP [RL95]	Oui	Oui	Non	Oui	Non
Inter-MR [LWC+10]	Oui	Oui	Non traité	Oui	Oui
BGP-MR [OKG14]	Oui	Non	Non traité	Oui	Oui
CIDR [ZCG09]	Oui	Oui	Non traité	Oui	Partiellement
BGP-MX [KTRH11]	Oui	Oui	Non traité	Oui	Non

TABLE 3.1 – Comparatifs des protocoles pertinents étudiés dans l'état de l'art

Nous avons pu voir que la littérature concernant les protocoles de routage inter-MANET est assez importante et qu'il s'agit d'un enjeu majeur pour les communications tactiques. Bien que le domaine semble bien couvert et résolu, nous constatons que chaque protocole résout un problème différent et surtout dans des conditions propres (InterMR sur des réseaux réactifs, CIDR avec des conditions spéciales de fusion/division, BGP-MR sur de l'OSPF-MDR). Afin de réaliser un protocole tel que nous l'envisageons selon nos critères, nous avons choisi de partir sur la réalisation d'un nouveau protocole. Nos principaux objectifs seront de faire un protocole qui soit capable de supporter les changements de topologie (séparation et fusion) que ce soit par une interruption physique ou par un ordre hiérarchique. Enfin, nous nous intéresserons aux problèmes d'adressage, qui vont s'avérer moins impactant que ce qui a pu être mis en valeur dans l'état de l'art. Dans le cadre de réseaux tactiques militaires, nous ferons également en sorte que ce protocole soit sécurisé, c'est-à-dire qu'il garantisse des propriétés telles que la confidentialité, l'intégrité, l'authentification et la non-répudiation. L'intégration de la sécurité nous fera également travailler sur de nouvelles perspectives et mettra d'autres problématiques en avant.

Chapitre 4

Étude comparative d'outils de simulation dans un réseau ad hoc

Dans l'état de l'art, nous avons pu constater qu'une problématique majeure liée aux communications inter-domaine est l'adressage des nœuds. En effet, dans le routage et les annonces de voisins et de routes, des doublons et boucles sont susceptibles de provoquer des perturbations voire la panne du réseau. Les protocoles de l'état de l'art sont évalués sous simulateur et montrent l'efficacité de solutions telle que le filtre de Bloom en tant que système d'adressage alternatif. La philosophie des réseaux ad hoc est de permettre à n'importe quels nœuds à portée de pouvoir communiquer, quelque soit la configuration IP adoptée (sous réserve qu'ils partagent les mêmes technologies de niveau physique, liaison de données et réseau). À partir de ce constat, il nous semble étrange d'avoir recours à un système d'adressage alternatif puisqu'en ad hoc, tout est conçu pour s'en dispenser. Notre principale motivation à travers cette étude comparative est la vérification des problèmes soulevés en état de l'art. Nous utiliserons pour cela des scénarios simples qui permettront de comprendre facilement les problématiques des communications inter-domaine.

Nous avons choisi NS3 [1] et CORE [2] car ils sont représentatifs des logiciels utilisés aujourd'hui pour simuler des réseaux. BGP-MR [OKG14] a été évalué avec CORE et Inter-MR [LWC+10] a été évalué avec NS2. Nous avons choisi d'évaluer son successeur, NS3, qui aujourd'hui permet la simulation de la couche 2 (Liaison) ainsi que la création de modèles de mobilité.

4.1 Éléments comparés

Avant de présenter les expériences réalisées ainsi que les résultats, nous allons voir comment fonctionnent les logiciels que nous comparons dans cette étude : NS3, CORE et une plateforme expérimentale.

4.1.1 Simulateur NS3

Network Simulator 3 est un logiciel de simulation de réseau. Il fonctionne à partir d'un ou plusieurs scripts d'entrée rédigés en C++. À partir de NS3, il est possible d'extraire plusieurs fichiers de logs utilisés par le logiciel NetAnim :

— un fichier *.xml* qui permet d'avoir une représentation visuelle du scénario ainsi que les échanges de paquets
— un fichier *.flowmon* qui contient tous les échanges effectués pendant la simulation ainsi que leurs performances

1. https ://www.nsnam.org/releases/
2. https ://www.nrl.navy.mil/itd/ncs/products/core

— un fichier *.tr* qui contient l'état des tables de routage des nœuds à chaque moment du scénario
— un fichier *.pcap* qui contient la trace réseau, visualisable avec Wireshark.

Fichier *.xml*

Ce fichier XML va servir dans deux éléments de NetAnim. Tout d'abord, il offre une représentation graphique du réseau simulé. On peut y afficher les adresses IP des nœuds, visualiser les paquets échangés au fil du temps et le type de message envoyé. À partir de ce même fichier, on peut également visualiser sous forme de diagramme d'échanges les paquets envoyés. On peut se focaliser sur un intervalle de temps, sur un type de paquets ou certains nœuds comme le montre la Figure 4.1.

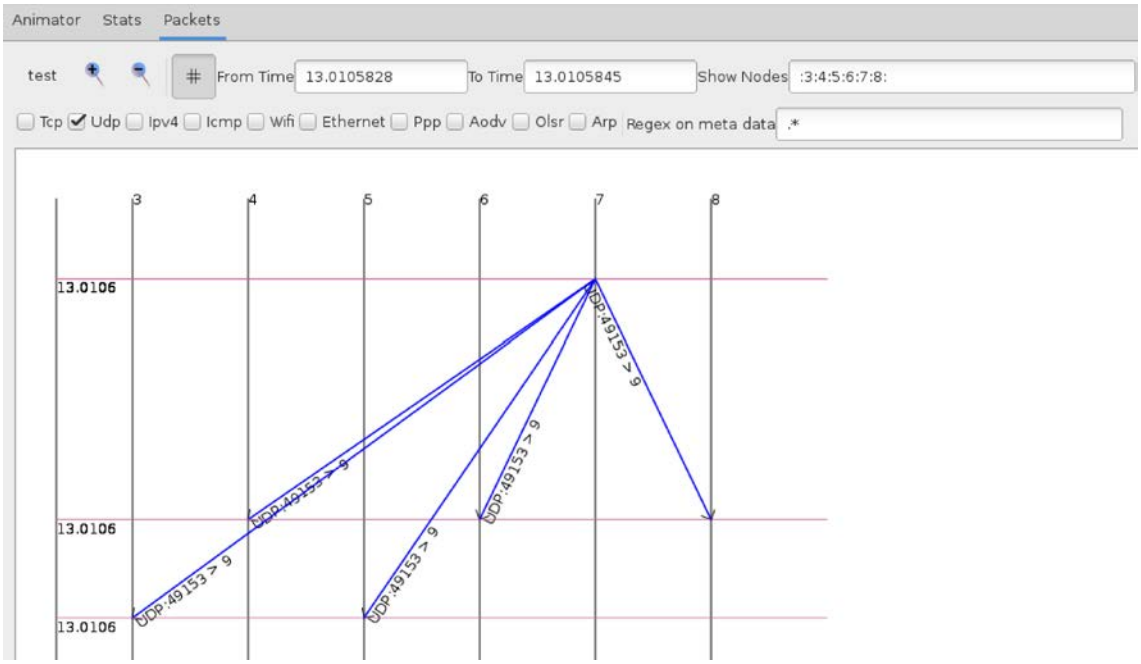


FIGURE 4.1 – Diagramme d'échanges dans NetAnim : envoi par le nœud 7 en diffusion d'un paquet UDP du port 49153 vers le port 9

Fichier *.flowmon*

Ce fichier Flow Monitor relève l'ensemble des paquets qui ont transité dans le réseau. À partir de ces données, il les agrège et les synthétise pour présenter l'ensemble des flux qui ont été échangés entre une source et une destination données. Plusieurs caractéristiques y sont présentées telles que le débit d'envoi/réception des paquets, la gigue ou bien le taux de perte. Dans l'exemple de la Figure 4.2, on peut voir trois flux différents. Tout d'abord, l'émission par 10.0.0.1 des annonces OLSR sur le réseau (Flow ID 2). Ensuite, l'échange UDP de 10.0.0.9 vers 10.0.0.1 qui est décomposé en deux avec l'aller (Flow ID 10) et le retour (Flow ID 11).

Fichier *.tr*

Ce fichier TR possède une architecture XML et contient l'ensemble des tables de routage des nœuds. Comme pour les diagrammes d'échanges, il est possible de voir les ajouts/suppressions d'entrées au fil de la simulation et de se focaliser sur certains nœuds (Figure 4.3).

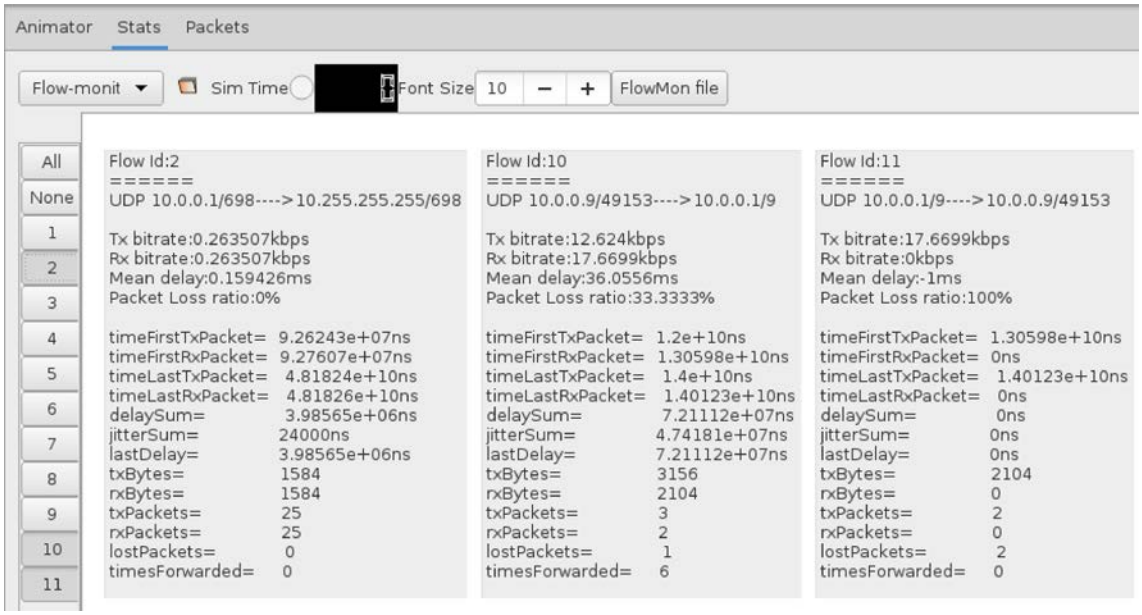


FIGURE 4.2 – Présentation de flux de données

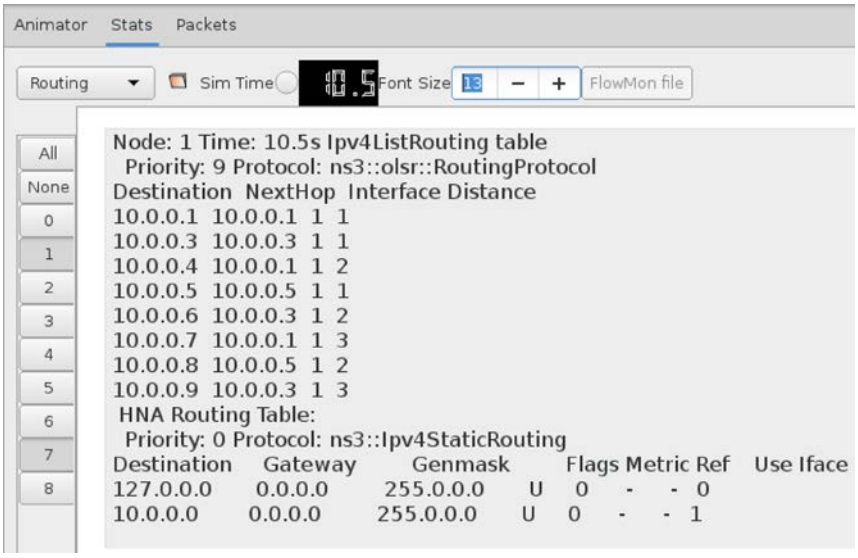


FIGURE 4.3 – Visualisation des tables de routage

4.1.2 Émulateur CORE

Common Open Research Emulator est un logiciel développé par l'US Naval Research Laboratory qui permet l'émulation de réseau. À partir d'une interface graphique, il est possible de définir un ensemble de nœuds et de machines associé à des services. La liste des services proposés par défaut sont illustrés sur la Figure 4.4. Tous ces éléments sont customisables et permettent l'intégration de ses propres services, notamment en faisant appel à des commandes du système ou bien à des scripts en indiquant le chemin de localisation.

CORE est souvent utilisé pour tester les environnements MANET car il permet l'utilisation de scripts de mobilité pour jouer des scénarios. De plus, il est possible d'agir pendant la simulation sur la topologie du réseau. Ainsi, pour vérifier manuellement l'impact du déplacement d'un nœud, il suffit simplement de glisser le nœud à portée d'un autre voisin et de vérifier l'impact. Enfin, les modifications et vérifications peuvent également être faite en pleine simulation, puisque CORE

propose pour chaque nœud l'accès à un terminal pour passer des commandes et lancer des scripts supplémentaires.

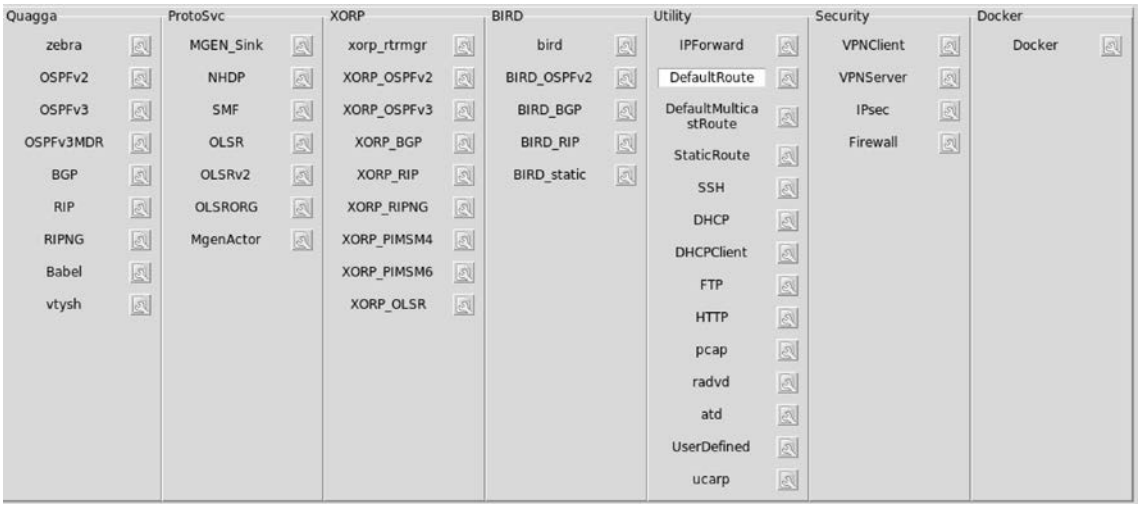


FIGURE 4.4 – Protocoles par défaut proposés par CORE

4.1.3 Plateforme expérimentale

La plateforme est constituée d'un ensemble d'ordinateurs portables. Leurs caractéristiques sont indiquées dans le Tableau 4.5. Les cartes WiFi utilisées sont configurées en mode ad hoc et l'implémentation utilisée de OLSRv1 est *olsrd*[3].

Processeur	Intel Core i3-400 2 x 2,4 GHz
RAM	4 Go
Distribution	Debian 8
Noyau	3.16

FIGURE 4.5 – Caractéristiques techniques de la plateforme expérimentale

4.2 Expériences réalisées

Dans cette section, nous mettons en place trois expériences afin de confronter NS3, CORE et notre plateforme expérimentale pour vérifier si leurs comportements sont identiques face aux réseaux ad hoc. Auparavant, nous présenterons le modèle OSI qui régit les communications réseaux pour comprendre la manière dont les expériences ont été menées.

4.2.1 Rappels sur le modèle OSI

Le modèle OSI[ISO84] (Open Systems Interconnection) est, comme son nom l'indique, une modélisation des communications entre ordinateurs. Il s'agit d'une architecture en couches où chaque niveau va assurer une fonctionnalité précise. Le but est de gérer les communications entre applications distantes à travers un réseau. Ces couches sont au nombre de 7 et sont souvent représentées comme sur la Figure 4.6. Chaque couche est indépendante et ne communique qu'avec une couche adjacente. Par exemple, la couche Réseau n'envoie jamais d'informations directement à la couche Application mais peut le faire par l'intermédiaire de la couche Transport. Lorsqu'une application

3. http ://www.olsr.org/

(couche 7) veut envoyer des données sur un réseau, les informations descendent selon les couches successives et vont être encapsulées par chacune d'entre elles. En réception, l'opération inverse se produit et chaque couche va supprimer son encapsulation au fil de la remontée des couches. Ces couches sont définies comme suit :

- Physique : cette couche fournit le support de communication utilisé, également appelé *medium*, et est en charge de la transmission du signal (ex : fil cuivré, onde radio)
- Liaison de données : cette couche permet à un appareil d'accéder au médium partagé. L'exemple le plus fréquent d'utilisation sur cette couche est l'adressage MAC [4] qui identifie chaque carte réseau du réseau local. Pour faire communiquer les machines, un protocole d'échange est utilisé (ex : Ethernet)
- Réseau : cette couche permet de faire communiquer différentes entités reliées sur un même réseau. IP [5] est aujourd'hui la technologie majoritaire qui assure cette fonctionnalité par un adressage des appareils connectés
- Transport : cette couche permet l'identification du service sur la machine. Cela se fait par un numéro de port. Il existe deux protocoles pour assurer les fonctionnalités de cette couche : TCP [6] (mode "connecté", retransmission du segment en cas d'erreur) qui est privilégié pour des applications qui nécessitent une fiabilité de transmission et UDP [7] (mode "non-connecté", ne garantit pas l'arrivée du segment) qui est privilégié pour des applications orientées temps-réel comme le streaming vidéo.
- Session : si plusieurs sessions d'une même application sont ouvertes, cette couche contient les informations pour diriger les données vers la bonne instance de l'application. Peu voire pas de protocoles sont implémentés pour cette couche. L'essentiel du travail de la couche Session est fait soit au Niveau 4, soit au Niveau 6
- Présentation : cette couche a pour but de faire la transition entre le Niveau 5, où les données sont encore sous forme d'octets, et le Niveau 7 où l'application attend des données mises en forme et prêtes à l'emploi. Il s'agit ici d'encoder les données.
- Application : cette couche correspond à l'application cible, qui est le service de l'utilisateur

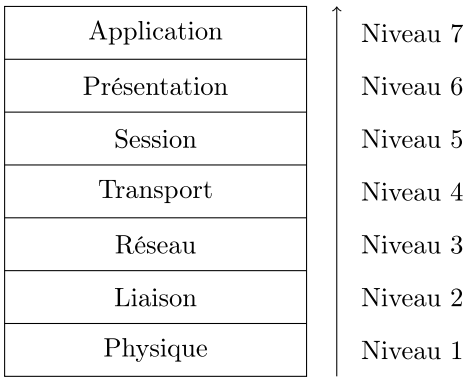


FIGURE 4.6 – Représentation du modèle OSI

Dans la suite de ce chapitre, nous voulons vérifier ce fonctionnement en couches dans un réseau ad hoc que ce soit sous les logiciels utilisés en état de l'art ou sur une plateforme expérimentale. Le but est de vérifier que le comportement est identique entre les trois candidats et particulièrement aux niveaux 2 et 3 qui constituent le problème majeur des communications inter-MANET : l'adressage.

4. Medium Access Control
5. Internet Protocol
6. Transport Control Protocol
7. User Datagram Protocol

4.2.2 Évaluation de la couche Réseau

La première expérience teste la connectivité de deux réseaux ad hoc selon le sous-réseau IP attribué. Les paramètres de la simulation sont les suivants : deux réseaux ad hoc sont composés de 5 nœuds chacun [8]. Ils sont répartis sur deux sous-réseaux IP différents sur une grille de 2 lignes par 5 colonnes. Les nœuds sont espacés de 100 mètres et à portée uniquement de leurs voisins à 1 saut. Le premier sous-réseau, nommé AHN1, a pour préfixe 10.10.10.0/24 tandis que le second sous-réseau, nommé AHN2, a pour préfixe 20.20.20.0/24. Cette expérience est illustrée sur la Figure 4.7.

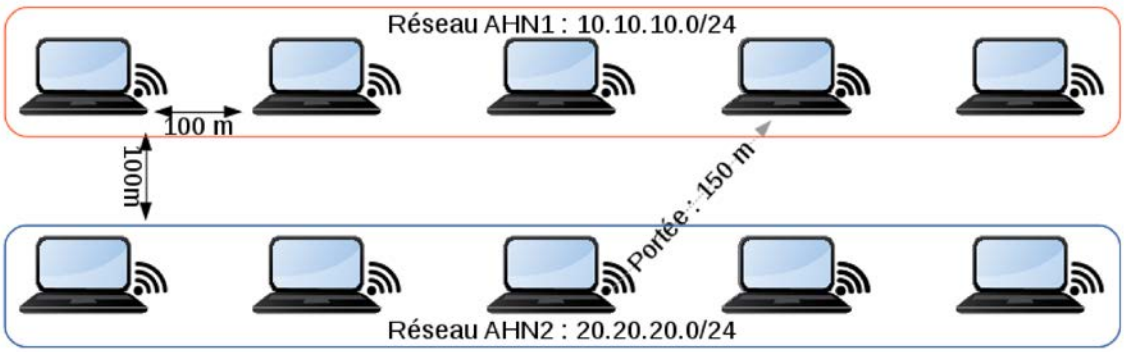


FIGURE 4.7 – Topologie réseau de la première expérience

L'expérience se déroule de la manière suivante : pendant 10 secondes, les nœuds s'échangent des HELLO et des TC OLSR afin de découvrir la topologie du réseau et construire leur table de routage. A partir de la 10^{eme} seconde, un client envoie une requête à un serveur. Dans un premier temps, le client est choisi parmi le sous-réseau AHN1 tandis que le serveur est choisi dans le sous-réseau AHN2. Ensuite, le client et le serveur sont choisis dans le même sous-réseau.

Ce test est réalisé sur les trois éléments : le simulateur NS3 (utilisation du modèle *AdhocWifiMac*, OLSRv1 utilisé), l'émulateur CORE (implémentation NRL-OLSR [9] qui suit la RFC d'OLSRv1) et la plateforme expérimentale (carte sans fil configurée en mode Ad-Hoc et utilisation de l'implémentation OLSRv1 *olsrd*). Les résultats de connectivité sont synthétisés dans le Tableau 4.8.

Nous pouvons remarquer que pour NS3 et CORE, si deux nœuds sont dans des sous-réseaux différents, le client n'a pas de route pour atteindre le serveur et les paquets ne sont pas envoyés. Le comportement observé ici est celui d'un réseau filaire. En effet, si deux sous-réseaux IP définis par leur préfixe en /24 veulent échanger des informations, il est nécessaire qu'un routeur vienne faire la liaison entre les deux sous-réseaux différents. Sur la plateforme, *olsrd* détecte tous les voisins, quelque soit le sous-réseau utilisé. Ainsi, tous les nœuds du réseau sont présents dans les tables de routage et l'échange client-serveur se déroule correctement. Nous constatons donc que le sous-réseau IP des nœuds n'a aucune influence sur la communication. Tant que les deux nœuds sont à portée, ils communiquent. Cette constatation est en adéquation avec la philosophie proposée par les réseaux ad hoc : communiquer avec ses voisins à portée sans avoir besoin d'une infrastructure.

Pour CORE, nous avons décidé de pousser l'investigation en évaluant l'implémentation NRL-OLSR. Elle est disponible est utilisable en dehors du logiciel. Nous l'avons donc déployée sur notre plateforme expérimentale. Nous avons pu constater que NRL-OLSR seul donne des résultats identiques à la plateforme c'est-à-dire que quelque soit les adresses IP source et destination, les nœuds sont capables de se joindre. On peut donc avancer que la conception logicielle bride le fonctionnement ad hoc et fait du cloisonnement en fonction des préfixes IP, à la manière d'un réseau filaire.

8. Pour les résultats de connectivité, le nombre de nœuds n'est pas un critère important
9. https ://downloads.pf.itd.nrl.navy.mil/olsr/

Client \ Serveur	10.10.10.2/24	20.20.20.2/24
10.10.10.1/24	✔	✘
20.20.20.1/24	✘	✔

(a) Connectivité NS3

Client \ Serveur	10.10.10.2/24	20.20.20.2/24
10.10.10.1/24	✔	✘
20.20.20.1/24	✘	✔

(b) Connectivité CORE

Client \ Serveur	10.10.10.2/24	20.20.20.2/24
10.10.10.1/24	✔	✔
20.20.20.1/24	✔	✔

(c) Connectivité Plateforme

FIGURE 4.8 – Connectivité client-serveur selon le choix du sous-réseau

Ce fonctionnement ad hoc où l'adresse du nœud constitue un identifiant unique a pu être observé sur la plateforme. En effet, bien que les nœuds aient été configurés avec des préfixes en /24, les annonces OLSR ne comportent pas de préfixes. Cela montre bien que dans le monde ad hoc, le préfixe IP est ignoré. De plus, lorsqu'un voisin est ajouté dans la table de routage, il est associé à un préfixe en /32. Ce qui valide définitivement le fait que dans un réseau ad hoc déployé, l'adressage ne peut pas créer de problèmes d'agrégation comme nous avons pu le voir dans la Partie 2.5.2. Le préfixe IP ne constituant pas un séparateur, il n'y a pas besoin de système supplémentaire pour faire communiquer 2 nœuds d'espace d'adressage différents.

Sur cette première expérience, nous avons mis en évidence des différences comportementales entre NS3, CORE et l'implémentation *olsrd* dans un réseau ad hoc simple. Côté logiciels, les communications semblent possibles si les conditions sont les mêmes que celles d'un réseau filaire c'est-à-dire si les nœuds appartiennent au même sous-réseau IP. Cependant, nous avons constaté le comportement inverse sur la plateforme. Deux nœuds à portée peuvent communiquer, indépendamment du sous-réseau auquel ils appartiennent. Il semblerait que cette différence soit dû à l'émulateur lui-même pour CORE car l'implémentation NRL-OLSR a montré les résultats attendus sur la plateforme.

Cette expérience nous a montré plusieurs choses. Tout d'abord, l'adressage IP dans les réseaux ad hoc ne fonctionne pas de la même manière que pour les réseaux filaires. Ainsi, chaque nœud est identifié de manière unique en diffusant son adresse avec un préfixe plein. L'utilisation de passerelles pour joindre les autres réseaux est inutile, puisque contrairement aux réseaux filaires, la notion de sous-réseau n'existe pas. Cependant, cela crée une nouvelle problématique qui est la définition des groupes. Bien que nous puissions utiliser des IP fixes pour identifier l'appartenance des nœuds à un groupe, cela ne les empêche pas de communiquer naturellement avec les voisins. Il faut donc travailler sur la création de groupes qui ne s'échangent pas de données tant que les politiques de routage ne les y autorisent pas. C'est pour cela que nous allons travailler sur la couche Liaison et étudier son fonctionnement lors de la deuxième expérience.

4.2.3 Évaluation de la couche Liaison de données

Le découpage en sous-réseaux IP se situe au niveau de la couche Réseau. Cela suppose donc que les couches inférieures, Physique et Liaison, sont autonomes et n'interagissent pas avec le protocole inter-domaine[LWC+10]. Cependant, dans les réseaux sans fil CSMA [Lam80] (Carrier

Sense Multiple Access), un séparateur existe au niveau de la couche Liaison : il s'agit du SSID (Service Set IDentifier) pour les réseaux de la famille 802.11 par exemple (sur les réseaux TDMA [NK85] (Time Division Multiple Access), ce séparateur existe également, il s'agit de la fréquence utilisée). La seconde expérience se déroule de la manière suivante : pour chaque élément de comparaison (NS3, CORE, plateforme), nous testons la connectivité entre deux nœuds en combinant deux paramètres : le sous-réseau IP et le SSID utilisé (nous utiliserons OLSR1 et OLSR2 comme SSID). Pour NS3, nous utilisons l'attribut SSID du modèle *AdhocWifiMac* qui permet de définir la signalisation de Niveau 2. Pour la plateforme expérimentale, nous utilisons la carte WiFi en mode ad hoc.

Cette expérience n'a pas été réalisée pour CORE car les mécaniques de signalisation de niveau Liaison ne sont pas implémentées dans le logiciel [10]. Les résultats des tests de connectivité sont présentés dans le Tableau 4.9.

Client \ Serveur	10.10.10.2/24		
	NS3	CORE	Laptop
10.10.10.1/24	✔	N/A	✔
20.20.20.1/24	✘	N/A	✔

(a) Connectivité client-serveur sur un même SSID

Client \ Serveur	10.10.10.2/24		
	NS3	CORE	Laptop
10.10.10.1/24	✔	N/A	✘
20.20.20.1/24	✘	N/A	✘

(b) Connectivité client-serveur sur différents SSID

FIGURE 4.9 – Connectivité client-serveur en fonction du sous-réseau IP et du SSID

Avec les résultats de cette seconde expérience, nous pouvons faire les constations suivantes :
- Pour NS3, les résultats sont identiques à la première expérience. Le SSID ne semble avoir aucune influence.
- Pour la plateforme, le SSID semble être un critère déterminant de communication. Si les deux nœuds n'ont pas le même SSID alors, ils ne sont pas capables de communiquer.

Nous avons de nouveau montré que le comportement logiciel est différent de celui de la plateforme expérimentale. Le SSID, isolation au niveau Liaison, n'est pas pris en compte dans NS3 en mode ad hoc. N'offrant pas d'implémentation officielle sous NS3, le mode TDMA n'a pas pu être vérifié dans cette expérience.

4.3 Interconnexion de deux MANET sur une infrastructure fixe

Pour terminer sur les expériences réalisées, nous avons voulu observer les communications lorsqu'un ou plusieurs MANET souhaitent se raccrocher à une infrastructure fixe. Ce type de situation se présente souvent en opération puisqu'il permet, une fois les hommes en retour de terrain, d'acheminer plus rapidement leurs informations à la base de commandement. Nous sommes donc dans une situation, représentée sur la Figure 4.10, où deux groupes OLSR sont connectés sur un réseau filaire routé avec OSPF pour notre exemple.

Le processus de test reste inchangé par rapport aux précédentes expériences. Notre but est de tester la connectivité entre les deux MANET, en fonction de l'adressage utilisé. Dans un premier

10. Renseignement directement obtenu par un développeur de CORE. Il nous a été conseillé de créer plusieurs réseaux et d'utiliser les passerelles pour pouvoir simuler des réseaux ad hoc séparés.

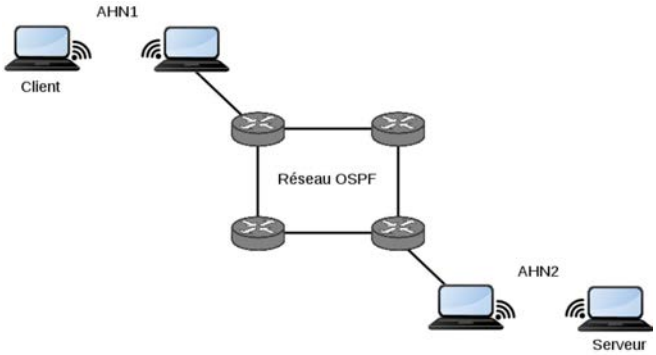


FIGURE 4.10 – Deux groupes mobiles attachés à une infrastructure fixe

test, les MANET seront dans le même sous-réseau. La seconde fois, ils seront séparés. Dans les réseaux ad hoc tels que nos MANET OLSR, une différence se fait entre l'adresse de l'interface sans fil et la détection par le voisin. En effet, si un nœud N1 est configuré avec une adresse 10.10.10.1/24, un voisin N2 le détectera et l'ajoutera dans sa table de routage comme 10.10.10.1/32. Ainsi, dans un réseau ad hoc expérimental, l'adresse IP identifie de manière unique un nœud, ce qui reflète la philosophie du réseau ad hoc. Pour la première partie de ce test, nous utiliserons une configuration avec des préfixes de 24 bits. Les tests de connectivité pour cette expérience sont résumés dans le Table 4.1.

Client \ Serveur	10.10.10.2/24		
	NS3	CORE	Plateforme
10.10.10.1/24	✘	✘	✘
20.20.20.1/24	✔	✔	✔

TABLE 4.1 – Connectivité client-serveur entre deux MANET, configuration par défaut

Nous pouvons constater que les trois outils se comportent de manière similaire. Si le client et le serveur sont dans un même sous-réseau, la connexion est impossible. Nous pouvons expliquer cela de la manière suivante : pour les trois outils, quand une interface est configurée avec un masque /24, une route correspondant à ce préfixe est automatiquement ajoutée dans la table de routage, même si le mode ad hoc est utilisé. Les nœuds qui font la passerelle OLSR/OSPF propagent donc le préfixe 10.10.10.0/24 pour chaque MANET. Lorsqu'un routeur OSPF reçoit ces deux annonces, il ne garde que celle qui lui permet d'accéder le plus rapidement à ce sous-réseau. Ainsi, les passerelles OLSR/OSPF qui émettent dans le réseau filaire voient aussitôt leurs paquets retournés par le routeur, puisqu'ils sont sur la route la plus proche du point de vue du routeur.

Afin d'éviter ce comportement, il y a deux manières de procéder sur les nœuds passerelle :

— Lorsqu'une route du MANET avec un préfixe /24 est enregistrée, elle est supprimée et remplacée par son équivalent avec un masque en /32

— Configurer l'interface sans fil avec un masque en /32.

Avec ces alternatives, nous conservons la philosophie ad hoc de l'identification unique d'un nœud. En conséquence de cela, la passerelle diffuse au réseau OSPF des routes en /32. De nouveaux tests de connectivité ont été réalisés suite à cette modification. Les résultats sont présentés dans le Table 4.2.

Client \ Serveur	10.10.10.2/24		
	NS3	CORE	Plateforme
10.10.10.1/24	✔	✔	✔
20.20.20.1/24	✔	✔	✔

TABLE 4.2 – Connectivité client-serveur entre deux MANET, modification du masque de la passerelle

Avec cette dernière expérience, nous avons montré qu'un MANET partitionné, c'est-à-dire deux groupes sans-fil distincts avec le même préfixe IP visible par les routeurs fixes, peut se connecter sur un réseau filaire et continuer ses communications, sous réserve qu'il n'y ait pas d'adresse dupliquée.

4.4 Résultats

Avec ces expériences, nous avons mis en évidence des différences comportementales entre le simulateur NS3, l'émulateur CORE et notre plateforme expérimentale. Ces différences se situent aux niveaux 2 (Liaison) et 3 (Réseau) du modèle de communication OSI.

Tout d'abord, nous avons étudié la couche Réseau. NS3 et CORE montrent un comportement similaire à celui d'un réseau filaire. Afin de pouvoir communiquer, les nœuds doivent être configurés sur le même sous-réseau IP. En comparaison, notre plateforme expérimentale nous a montré que, quelques soient le sous-réseau IP et le masque utilisés, deux nœuds à portée sont capables de communiquer. Cependant, si l'on extrait l'implémentation NRL-OLSR de CORE, elle montre un comportement identique à celui de la plateforme expérimentale. Nous avons donc ici un élément supplémentaire qui vient conforter le fait que les logiciels brident les communications à un modèle filaire.

Ensuite, nous avons étudié la couche Liaison. L'expérience à ce niveau s'est portée sur le découpage des réseaux en SSID. Cette fonctionnalité n'a pas pu être testée sous CORE puisque le SSID ni aucune autre signalisation Liaison n'est implémentée dans ce logiciel. Le comparatif effectué a montré que la mécanique de SSID, telle qu'elle a été définie lors de sa création, n'est pas respectée sous le simulateur NS3 (Nous avons cependant pu constater que le SSID était fonctionnel sous NS3 mais lorsque celui-ci est utilisé pour un mode Infrastructure). En effet, si deux nœuds appartiennent à des SSID différents, ils sont capables de communiquer (sous réserve d'être dans le même sous-réseau IP). Cependant, la couche Liaison étant évaluée avant la couche Réseau en réception, ces communications ne doivent pas avoir lieu. Pour la plateforme, nous retrouvons un fonctionnement standard où la couche Liaison et ses attributs sont évalués avant la couche réseau en réception. Ainsi, pour que deux nœuds ad hoc communiquent, la condition d'appartenir au même SSID est nécessaire.

Enfin, nous nous sommes intéressés à la communication entre deux MANET via un réseau filaire OSPF. Dans cette expérience, nous avons pu constater deux comportements similaires :

1. Si les MANET appartiennent à des sous-réseaux différents, la communication s'établit
2. S'il s'agit d'un MANET divisé, c'est-à-dire les deux MANET appartiennent au même sous-réseau, la communication ne se met pas en place

Ce comportement peut être expliqué par les éléments suivants :

(i) le nœud qui fait la passerelle OLSR/OSPF

(ii) le traitement des routes dans Linux

En effet, si les deux MANET sont situés sur deux sous-réseaux différents, il n'y a aucun problème dans la diffusion des routes et la communication s'établit nativement. Cependant, si les deux MANET appartiennent au même sous-réseau, la communication est impossible. Ce phénomène se produit car les interfaces réseaux se configurent avec des masques généralement inférieurs ou

égaux à 24. Linux va donc ajouter ce réseau en /24 dans sa table de routage et la diffuser sur le réseau OSPF. L'algorithme du protocole OSPF ne gardant que le chemin le plus court, il conserve uniquement la route de la passerelle la plus proche. Cela empêche donc toute sortie de paquet d'un MANET, puisqu'il sera aussitôt retourné vers cette même passerelle.

Dans un réseau ad hoc, les routes vers d'autres nœuds sont annoncées avec un masque en /32. Nous avons donc rendu fonctionnel la communication entre deux MANET de même sous-réseau en configurant l'interface ad hoc des passerelles avec un masque de 32 bits. Ainsi, les routes créées par Linux en /32 seront diffusées et conservées par le réseau OSPF. Chaque routeur garde donc la trace des deux passerelles et les échanges peuvent avoir lieu.

Avec ces trois expériences, nous avons pu rassembler des preuves que les problématiques du routage inter-MANET ne situent pas obligatoirement dans l'adressage contrairement à ce qui est affirmé dans la littérature. Cela s'explique par l'application d'un comportement filaire aux réseaux ad hoc par les logiciels d'évaluation. À partir des résultats précédemment présentés, nous proposons une nouvelle liste de problématiques sur laquelle nous allons travailler pour établir notre protocole inter-MANET.

4.5 Nouvelles problématiques du routage Inter-MANET

Après avoir mis en évidence des différences comportementales entre un simulateur de réseaux, un émulateur de réseaux et une plateforme expérimentale, nous allons montrer que certaines fonctionnalités des protocoles de l'état de l'art n'ont été créées que pour palier des problématiques de l'environnement logiciel. Si nous nous plaçons dans un cadre réel, nous remarquons que les problématiques à résoudre sont différentes. Dans les logiciels, le routage dans les réseaux ad hoc est géré de la même manière qu'un réseau filaire. Seuls des nœuds appartenant au même sous-réseau IP peuvent communiquer. Cependant, la philosophie des réseaux ad hoc est de pouvoir communiquer avec son voisin, tant qu'il est à portée et quelque soit son adresse IP. La plateforme expérimentale nous a confirmé ce comportement ad hoc, en établissant des communications entre des nœuds à portée, sans contrainte de sous-réseau IP ou de masque. Nous pouvons donc établir que l'ensemble des nœuds dans un MANET ne peut pas être défini par son adressage IP.

Cette constatation faite, nous souhaitons maintenant revenir sur les éléments d'état de l'art que nous jugeons dispensable dans la création d'un protocole de routage inter-MANET.

Tout d'abord, un nœud peut être défini de manière unique grâce à son adresse IP. Son voisinage le percevant comme une adresse en /32, il n'y a aucun risque de recouvrement de masque. Ainsi, une extension de nommage telle que le filtre de Bloom peut s'avérer être une surcharge que ce soit au niveau du nœud (temps de calcul, lecture, écriture) ou au niveau réseau (utilisation de bande passante). Son utilisation en tant qu'adressage alternatif nous semble peu convaincante en terme de réactivité également. Nous pensons qu'il est plus adapté pour des systèmes de broadcast pour éviter l'engorgement du réseau par des retransmissions inutiles.

Ensuite, nous avons fréquemment retrouvé le concept de hiérarchie et d'élection au sein des MANET. Dans le cadre de travaux logiciels, il paraît en effet pertinent de créer des points passerelles afin de pouvoir sortir de notre sous-réseau IP. Cette passerelle peut s'activer avec un système de *beaconing* ou bien par une élection au sein du groupe. Suite à nos expérimentations, nous avons pu constater que la présence de nœuds "chef" est également facultatif. Le réseau ad hoc OLSR est continu, sans distinction de groupe. Ainsi, chaque nœud peut contribuer équitablement à la transmission des données et ce à tout moment. La construction progressive des tables de routage permet à l'ensemble des nœuds de connaître la topologie du réseau et joindre facilement une destination.

Enfin, nous avons mis en évidence une alternative au sous-réseau IP pour séparer les MANET. Ce séparateur agit au niveau de la couche Liaison de données. Cela peut être le SSID (pour les réseaux CSMA) ou bien la fréquence allouée (pour les réseaux TDMA). L'adressage IP ne se révèle finalement pas comme un problème, puisqu'il peut rester identique au fil d'une mission. Ensuite, l'annonce des nœuds en préfixe /32 ne créé pas de soucis d'agrégation tel qu'il existe et est décrit dans la Partie 2.5.2 puisque l'agrégation ne peut pas avoir lieu. Ainsi, cette préoccupation n'est plus à prendre en compte lors de mobilité, séparation ou fusion de MANET. Il suffit de s'assurer qu'il n'y ait pas d'adresse IP dupliquée sur le réseau.

Au cours de ce chapitre valorisé par les publications [GGKP16a] et [GGKP16b], nous nous sommes placés dans un contexte réseau Inter-MANET et avons proposé un comparatif des outils d'évaluation de ces protocoles. Nous avons mis en place trois expériences pour tester des communications dans le cadre de réseaux ad hoc simples, afin de valider cette liste des problématiques. Elles ont été menées sur trois éléments : un simulateur de réseaux (NS3), un émulateur de réseaux (CORE) et une plateforme expérimentale. Nous avons mis en évidence des différences comportementales entre les logiciels et les machines de test. Nous avons pu montrer que les logiciels complexifient l'environnement ad hoc et créent des problématiques qui ne sont pas réalistes sur un déploiement opérationnel. Ces mêmes problématiques ont motivé des travaux qui permettent d'établir des communications inter-MANET sur des simulations et non dans un environnement réaliste. Pour des applications ad hoc inter-domaine, nous pensons donc qu'il est délicat d'utiliser les outils logiciels existant aujourd'hui car ils sont éloignés de la réalité. Nous préconisons des

implémentations directes sur le matériel utilisé pour l'application.

Suite à ces constations, notre travail va consister à répondre à ces problématiques opérationnelles de la communication inter-MANET. La première amélioration que nous envisageons est l'identification unique et certaine d'un MANET. Une fois cet élément créé, nous aurons donc les groupes de communication qui seront explicites. Afin de nous rapprocher au mieux du protocole de référence BGP, nous travaillerons également sur l'intégration des politiques de routage et la sécurisation des échanges.

Chapitre 5

ITMAN - Inter-Tactical Mobile Ad hoc Network

5.1 Motivations

Comme nous avons pu le démontrer dans le chapitre précédent, les problématiques à adresser pour un protocole inter-MANET efficace se révèlent différentes de celles évoquées dans l'état de l'art. À travers ITMAN, nous avons voulu créer un protocole qui réponde à nos objectifs et qui est directement développé sur matériel pour éviter le comportement déviant de logiciels de simulation.

Avant de présenter ITMAN, ses hypothèses de fonctionnement et sa construction, nous allons voir le fonctionnement de certains éléments cryptographiques : le chiffrement et les certificats.

5.2 Cryptographie

5.2.1 Chiffrement

Lorsque des données sont échangées sur un support, que ce soit filaire ou radio, il est possible d'intercepter les communications et de lire ce que les interlocuteurs s'échangent. Pour palier ce problème d'écoute, deux types de chiffrement existent pour assurer la confidentialité : le chiffrement symétrique et le chiffrement asymétrique.

De manière générale, l'algorithme de chiffrement prend en entrée le message d'origine (appelé *message en clair*), y applique une méthode de chiffrement avec une clé de chiffrement et envoie un message (appelé *message chiffré*) visiblement aléatoire et incompréhensible par quiconque ne possède pas la clé de déchiffrement.

Chiffrement symétrique

Les clés utilisées en cryptographie peuvent être de plusieurs catégories. Dans le cadre du chiffrement symétrique, on travaille uniquement avec des clés secrètes qui servent à chiffrer ou déchiffrer. Elles doivent être gardées secrètes par leur propriétaire. Dans le cas de divulgation de cette clé (fuite ou attaque), on parle alors de compromission et la confidentialité des messages est potentiellement perdue si un tiers vient à utiliser la clé pour déchiffrer les messages.

Le processus de chiffrement symétrique est relativement simple. À l'aide d'une fonction et de la clé, le message est chiffré et transmis à son destinataire. En réception, il suffit simplement d'appliquer à ce chiffré la fonction inverse avec la même clé pour retrouver les données en clair. L'un des chiffrements les plus simples qui existe aujourd'hui est le XOR. On applique cette fonction

bit à bit entre le clair et la clé pour obtenir le chiffré. En réception, on effectue l'opération XOR entre le chiffré et la clé pour retrouver le clair (Figure 5.1). Le standard de chiffrement symétrique actuel est AES [BBD+00] pouvant fonctionner avec des clés de 128, 192 ou 256 bits.

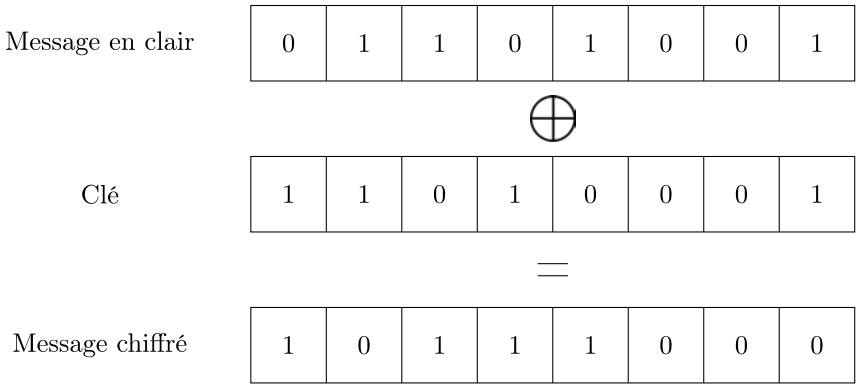


FIGURE 5.1 – Communication chiffrée par une méthode XOR

En chiffrement symétrique, la clé à utiliser est la même en émission ou en réception. Si l'on veut communiquer à distance, il faut au préalable avoir transmis cette clé. Cependant, pour éviter toute compromission de la clé, elle ne peut pas être envoyée en clair au préalable de la discussion. Il y a donc une nécessité que les interlocuteurs trouvent un moyen alternatif pour se mettre d'accord sur la clé à utiliser.

Chiffrement asymétrique

Afin de lever le risque qui pèse sur les clés privées et leur distribution auprès des destinataires, le chiffrement asymétrique propose l'utilisation d'une paire de clés privée/publique. Pour échanger des données, le propriétaire de la paire de clés va distribuer sa clé publique avec ses interlocuteurs. Pour lui envoyer un message, la source doit chiffrer le message avec la clé publique. En réception, le message est déchiffré avec la clé privée pour pouvoir lire les données en clair. Cette situation est illustrée sur le cas simple de la Figure 5.2.

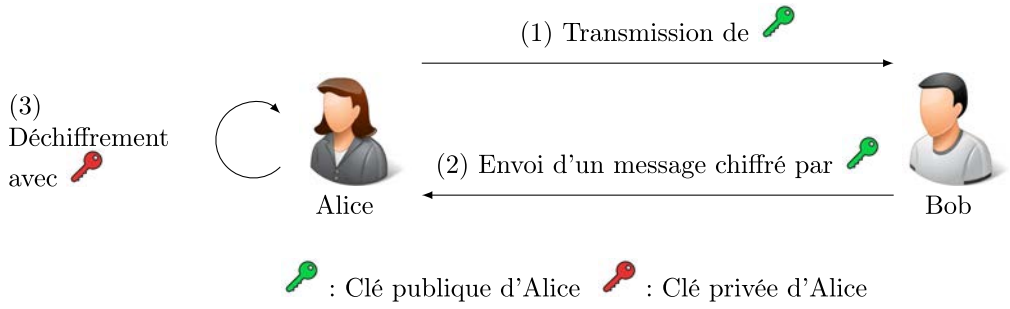


FIGURE 5.2 – Envoi d'un message par chiffrement asymétrique

Plusieurs avantages se dégagent avec cette méthode de chiffrement. Tout d'abord, le matériel cryptographique est créé par un seul élément mais est utilisable par plusieurs en distribuant la clé publique. Ensuite, la partie privée n'étant gérée que par une seule personne, le risque de compromission de la clé est diminué.

Certificats électroniques

Dans un réseau, lorsque une machine transmet sa clé publique en tant qu'élément identifiant, rien n'assure en réception que le message n'ait pas été rejoué par un tiers pour usurper l'identité du réel émetteur. C'est pour cela qu'il existe les certificats électroniques qui permettent d'identifier

de manière sûre un interlocuteur. Le format de certificat le plus couramment utilisé est le format X509 (RFC 4210 [AFKM05]). Dans la vie courante, il est utilisé dans les navigateurs Web pour s'assurer de la connexion HTTPS du site Internet sur lequel nous sommes. Cela permet de valider l'identité de notre destination et s'assurer que l'envoi potentiel de données sensibles, telle que le login de compte bancaire, arrive au bon serveur. Un certificat contient entre autre la clé publique, des éléments d'identification du détenteur et une durée de validité. Tous ces éléments sont délivrés par une autorité tierce et protégés par une signature numérique. Un nouvel échange comprenant la certification de la clé publique d'Alice est illustré sur la Figure 5.3.

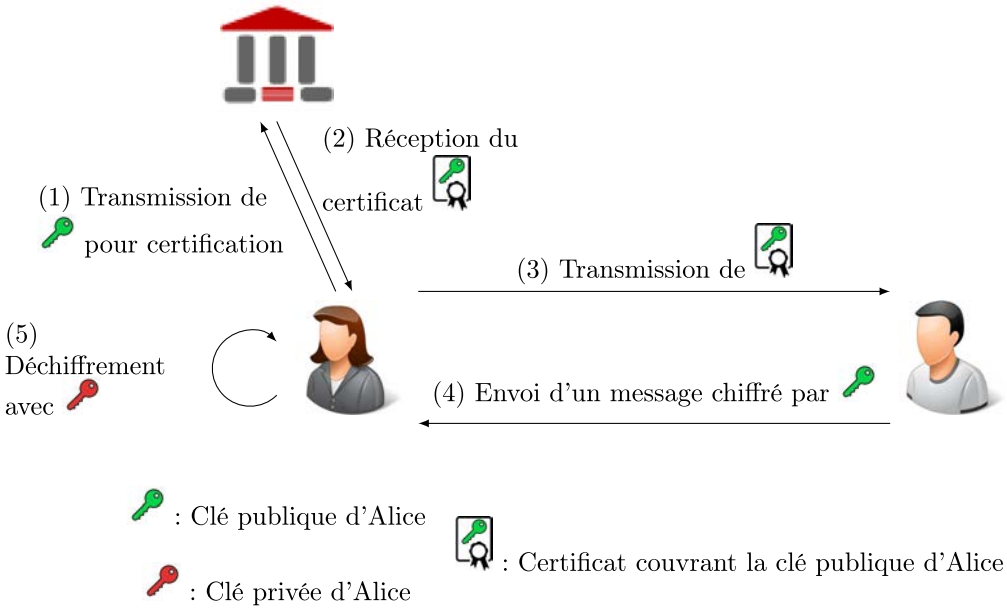


FIGURE 5.3 – Chaîne de certification et envoi d'un message par chiffrement asymétrique

Diffie-Hellman

Lorsque deux parties souhaitent communiquer via un canal chiffré, il est nécessaire qu'elles se mettent d'accord sur la clé de chiffrement à utiliser. Cependant, si les interlocuteurs sont distants, ils ne peuvent pas physiquement se mettre d'accord sur un matériel cryptographique commun à adopter. Diffie et Hellman proposent dans [Res99] un échange permettant, à partir du secret de chacun des participants, de créer une clé secrète commune, sans révéler la clé à un attaquant écoutant simplement les échanges. Ainsi, deux participants souhaitant établir des communications sécurisées peuvent le faire, sans contrainte de distance ou de risque pour sa propre sécurité. Une version authentifiée du protocole existe [BWM98] pour éviter l'attaque de l'homme au milieu.

Après avoir effectué quelques rappels sur des éléments cryptographiques, nous allons voir ITMAN, un protocole de communication inter-MANET.

5.3 Algorithmes de fonctionnement de ITMAN

Cette partie présente les algorithmes qui régissent notre protocole ITMAN ainsi que les évaluations que nous avons faites pour mesurer l'efficacité de notre solution.

5.3.1 Définitions et notations

Comme nous l'avons défini dans la Partie 2.4.3, un nœud embarque du matériel cryptographique. Le Tableau 5.1 détaille les éléments et notations de ce matériel cryptographique.

OGN (Original Group Number)	Groupe d'origine d'un nœud en début de mission
$K_{Priv}(N)/K_{Pub}(N)$	Paire de clé privée/publique du nœud N
$Cert(N)_{CA}$	Certification de l'identité du nœud N et de son OGN, signée par l'autorité CA
$Cert(CA_i)$	Certificat de l'autorité de certification i
$K(OGN)$	Clé de groupe pour chiffrer les messages dans le groupe OGN
$PolicyList$	Politiques à appliquer avec les autres groupes de la coalition

TABLE 5.1 – Éléments embarqués par un nœud N en début de mission

Trois types de communications sont présents sur le réseau :
- Communication Intra-groupe : messages échangés au sein d'un même groupe, chiffrés avec la clé $K(OGN)$
- Communication Inter-groupe : messages échangés entre deux groupes, chiffrés par une clé partagée entre les deux groupes
- Communication Générale : beacons envoyés en clair ou chiffrés par une clé générale de coalition pour la détection des groupes voisins

Dans nos objectifs, il est important que les hiérarchies des forces armées puissent appliquer des politiques de routage sur les communications inter-MANET. Les politiques à appliquer sont indiquées dans la $PolicyList$ sous la forme d'un couple ($politique$; OGN). La valeur de $politique$ peut être :
- Deny : aucune communication autorisée
- Link : communication autorisée au travers d'un lien unique c'est-à-dire via un nœud passerelle dans chaque groupe
- Merge : communication autorisée quelque soit le nombre de liens. Dès que deux nœuds de groupes différents sont à portée, la communication est possible

Nous considérons que la phase de planification de mission, telle que nous l'avons décrite dans la Partie 2.4, permet d'avoir des $PolicyList$ valides. Si le groupe A, identifié par OGN_A, a une politique Link envers le groupe B, identifié par OGN_B, alors, le groupe B a la même politique envers le groupe A.

Maintenant que les éléments présents dans chaque nœud en démarrage de mission sont détaillés, nous allons voir les algorithmes de découverte de groupe et d'établissement des communications.

5.3.2 Algorithmes de découverte des groupes voisins et d'établissement des communications

Lorsque la mission démarre, nous sommes donc dans une situation où chaque groupe chiffre ses informations de routage avec une clé de groupe. Ces informations restent dans un espace défini et ne sont pas transmises à un autre groupe externe ou allié comme le montre la Figure 5.4.

Afin que deux groupes communiquent, ils doivent dans un premier temps être conscient qu'ils sont à portée. Si les communications sont chiffrées séparément dans chaque groupe, il est alors nécessaire d'ajouter des messages envoyés régulièrement de type *beacon* par chaque nœud afin de se faire connaître. Ce *beacon* contient le certificat permettant d'identifier le nœud et son groupe afin de pouvoir prendre les décisions de politique à appliquer. Le paquet est daté par un *TimeStamp* afin d'éviter toute attaque de type rejeu. Ce *beacon* peut être envoyé à l'aide d'une clé de coalition qui sert de canal général à la coalition ou bien en clair, si l'organisation considère que l'identité des nœuds puisse être révélée à tout personne interceptant le paquet. L'algorithme d'envoi et de réception du *beacon* est détaillé dans l'Algorithme 1.

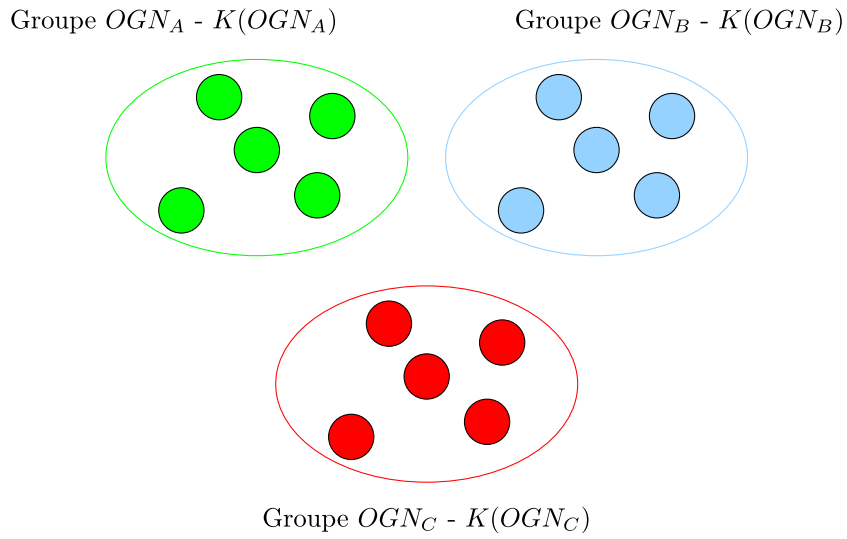


FIGURE 5.4 – Cloisonnement des groupes en départ de mission grâce au chiffrement

Algorithm 1 Routine d'envoi/réception de beacon

```
1: Diffusion périodique de (Cert(A)_CA,Timestamp)_Sig_KPriv(A)
2: while true do
3:   if Réception d'un beacon (Cert(B)_CA,Timestamp)_Sig_KPriv(B) then
4:     if Vérification de(Sig_KPriv(B)), (Timestamp) et (Cert(B)_CA) then
5:       Extraire OGN de Cert(B)_CA
6:       if OGN pas encore accessible then
7:         Vérifier dans (PolicyList) la politique à appliquer
8:       end if
9:     end if
10:  end if
11: end while
```

Une fois que deux groupes ont échangé leurs *beacons*, qu'ils ont vérifié leur validité et que le groupe voisin est nouveau, il faut appliquer la politique de routage associé définie dans *PolicyList*. Dans ITMAN, nous avons défini trois politiques de routage qui sont les suivantes :

— DENY : cette politique indique un refus de communiquer entre deux groupes.
— LINK : cette politique indique la possibilité de communiquer via une passerelle unique.
— MERGE : cette politique indique une communication totale entre deux groupes, quelque soit le nombre de passerelles

Nous allons voir par la suite comment ces politiques sont appliquées techniquement.

Politique *Deny*

Lorsque l'étape de vérification de l'identité est passée, il faut établir la communication. Comme indiqué précédemment, il y a le choix entre trois politiques. Dans le cas de la politique Deny, il n'y a aucune étape supplémentaire. En effet, comme les groupes sont déjà isolés, il n'y a pas de nouveaux éléments à créer pour empêcher les deux groupes de communiquer. Ils sont juste conscients de leur proximité grâce aux *beacons*.

Politique *Link*

Lorsqu'un nœud A reçoit un beacon d'un nœud B d'un autre groupe et qu'ils sont autorisés à communiquer via une politique Link, A et B créent un canal chiffré. La création de ce canal est

réalisée par la génération d'une clé partagée entre les deux nœuds. Cette clé pourra être pré-chargée en début de mission ou bien générée à partir d'un échange Diffie-Hellman. Une fois cela fait, les nœuds peuvent échanger leurs informations de routage via ce canal inter-domaine chiffré par la clé partagée K_{AB}. L'algorithme 2 décrit cette création du point de vue de A et est illustrée par la Figure 5.5. Il s'agit de la suite de l'Algorithme 1 une fois que les nœuds sont authentifiés.

Algorithm 2 Application de la politique Link

1: **if** Vérification dans ($PolicyList$) = Link **then**
2: **if** (Groupe B non présent dans la table de routage) **or** (Groupe B présent dans la table de routage **and** Test de connectivité échoue) **then**
3: Négociation de la clé partagée K_{AB}
4: Envoi des informations de routage au groupe B via le canal chiffré par K_{AB}
5: Réception des informations de routage du groupe B via le canal chiffré par K_{AB}
6: Échange régulier des informations de routage via le canal chiffré par K_{AB}
7: **end if**
8: **end if**

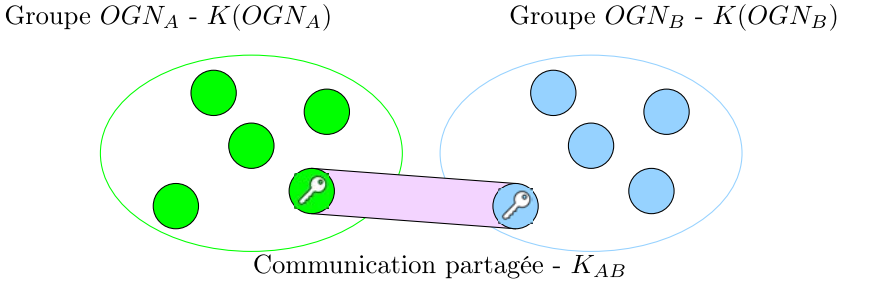


FIGURE 5.5 – Illustration d'une politique de routage Link

L'algorithme que nous avons présenté, et plus particulièrement la condition de la ligne 2, permet d'établir une communication dans le cas d'un groupe B scindé et dont les sous-groupes engendrés souhaitent se connecter au groupe A. En effet, si la situation illustrée sur la Figure 5.6 vient à se produire, c'est-à-dire que le groupe A vient à se scinder, $B2$ refusera la communication à $A2$ puisqu'il possède déjà une route vers OGN_A. Nous avons donc ajouté dans notre algorithme l'ajout d'un test de connectivité pour déterminer s'il s'agit d'un groupe séparé ou non. Voici les cas possibles et la décision prise par le nœud $B2$ en fonction des situations :

1. $A2$ n'est pas dans la table de routage : $A2$ a été supprimé des destinations et a donc été séparé de son groupe, le lien peut être créé
2. Pas de réponse au test : il s'agit d'une séparation car l'ancienne route n'a pas encore été supprimée par le protocole de routage, le lien peut être créé
3. Réponse au test : $A2$ est toujours joignable, ce n'est pas une séparation, le lien n'est pas créé

Une fois le tunnel créé, les routes sont propagées par les passerelles de la manière suivante (Figure 5.7) :

— Un nœud de OGN_A envoie ses informations de routage chiffrées avec $K(OGN_A)$.
— Ces informations arrivent jusqu'à la passerelle de OGN_A. La passerelle transmet ces informations deux fois mais chiffrées différemment. Une fois avec l'autre groupe, via K_{AB} , et une autre fois pour son groupe, avec $K(OGN_A)$.
— Le paquet arrive à la passerelle de OGN_B. Le paquet est déchiffré grâce à K_{AB}, traité puis chiffré avec $K(OGN_B)$ pour les membres de OGN_B

Après la présentation de la politique Link, nous allons maintenant voir la réalisation de la politique Merge.

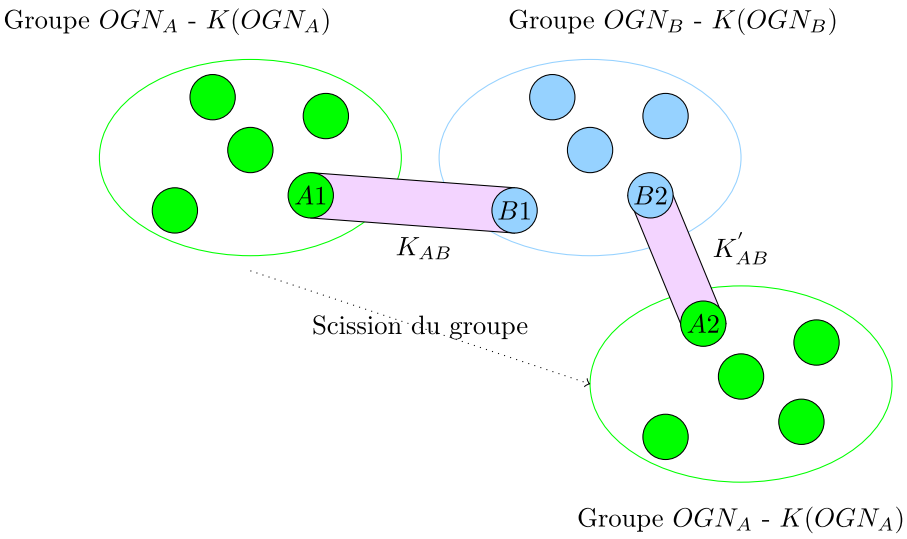
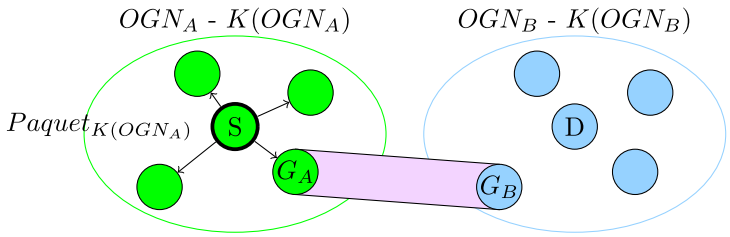
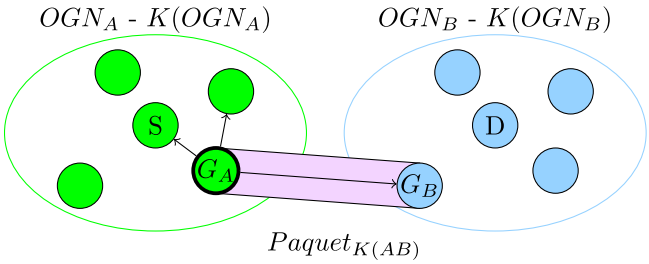


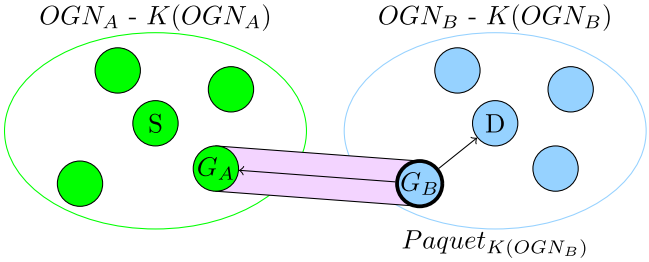
FIGURE 5.6 – Politique Link sur un groupe divisé



(a) Transmission des informations de routage dans OGN_A



(b) Traitement et renvoi par la passerelle



(c) Transmission des informations de routage dans OGN_B

FIGURE 5.7 – Transmission des informations de routage dans le cadre d'une politique Link

Politique *Merge*

La différence entre les politiques Link et Merge se situe au niveau du nombre de passerelles. L'algorithme de création de la clé partagée est donc identique. Cependant, une fois ce processus terminé, l'opération effectuée est différente. En effet, comme chaque nœud peut agir comme passerelle, la clé partagée K_{AB} est diffusée au sein de chaque groupe. Ainsi, chaque nœud possède le matériel cryptographique nécessaire pour communiquer. Cela permet d'éviter que chaque rencontre entraîne une génération de clé. L'algorithme 3 récapitule ce déroulement et est illustré par la Figure 5.8.

Algorithm 3 Application de la politique Merge

1: **if** Verification dans ($PolicyList$) = Merge **then**
2: Négociation de la clé partagée K_{AB}
3: Envoi des informations de routage au groupe B via le canal chiffré par K_{AB}
4: Réception des informations de routage du groupe B via le canal chiffré par K_{AB}
5: Envoi dans le groupe de la clé partagée K_{AB}
6: Échange régulier des informations de routage via le canal chiffré par K_{AB}
7: **end if**

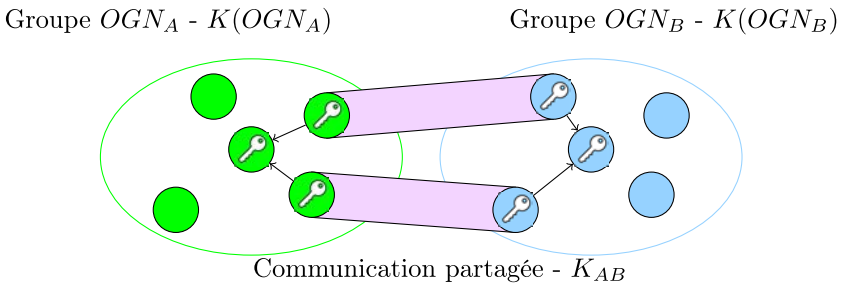


FIGURE 5.8 – Illustration d'une politique de routage Merge

ITMAN est un protocole qui s'appuie sur le protocole de routage intra-domaine et la cryptographie pour découvrir les autres groupes et s'y connecter. Il est également possible de définir des politiques d'échanges avec les autres groupes. La construction topologique du réseau se fait toujours à l'aide d'OLSR. Notre travail s'est articulé autour de la composition de groupes de communications, de leur interconnexion et de l'application de politique de routage. Pour évaluer les performances d'ITMAN, nous avons donc choisi d'évaluer les latences qu'induisent le chiffrement, le beaconing et la création de clé partagée par rapport à un réseau OLSR simple.

5.4 Évaluation expérimentale

Procédé de mesure

La plateforme de test qui nous permet d'évaluer l'impact de ITMAN et ses mécaniques est identique à celle utilisée dans le chapitre 3. La preuve de concept a été conçue de la manière suivante.

Deux scripts écrits en Python sont chargés de l'envoi et la réception des beacons. Le premier, *BroadcastSender.py*, est chargé d'envoyer à intervalle régulier le beacon signalant sa présence aux autres groupes. En face, *BroadcastReceiver.py* se charge de recevoir l'identité du nœud émetteur et d'en vérifier l'authenticité et la validité.

Le chiffrement des paquets OLSR est réalisé en plusieurs étapes. Tout d'abord, nous utilisons des commandes *iptables* associées pour stocker les paquets à destination du port 698 (OLSR)

dans une file. Nous utilisons pour cela NFQUEUE, qui est une extension de QUEUE et permet l'utilisations de files séparées et identifiées de manière unique. Nous faisons la distinction entre les paquets entrants et sortants qui sont stockés dans deux files différentes. En parallèle, deux scripts *ProxyIn.py* et *ProxyOut.py* sont chargés de récupérer dans leurs NFQUEUE respectives les paquets stockés pour les traiter. À l'aide de la bibliothèque *Scapy*[1], nous pouvons facilement manipuler le paquet, extraire les octets de la charge UDP, les remplacer par leur version chiffrée et recalculer les longueurs et checksum des en-têtes.

Afin d'être le plus précis possible dans la mesure du délai d'ajout d'un voisin, nous avons choisi de relever trois valeurs :

— Première valeur : les paquets sont simplement interceptés et renvoyés de manière identique. Il s'agit de notre valeur étalon, le mode non-chiffré.
— Deuxième valeur : les paquets sont interceptés par NFQUEUE et la longueur et le checksum du paquet sont recalculés. Cela nous permet d'évaluer le temps que les proxys mettent pour effectuer les calculs.
— Troisième valeur : les paquets sont interceptés par NFQUEUE, la payload OLSR est chiffrée et la longueur et le checksum du paquet sont recalculés. Ceci est la valeur finale et correspond au paquet utilisé dans ITMAN. Avec la deuxième valeur, on peut évaluer précisément le temps de chiffrement des paquets. Il s'agit du mode chiffré.

La procédure de l'expérience est la suivante : les timer OLSR sont configurés selon les recommandations du RFC 3626 (Hello 2 sec, TC 5 sec) et les paquets sont chiffrés avec de l'AES-128 bits.

Construction du réseau

La première expérience porte sur l'ajout de voisin au sein d'un groupe. Avant de communiquer avec d'autres groupes, il faut d'abord établir les communications intra-domaine. Nous mesurons le temps qui est mis entre la réception d'un premier paquet OLSR HELLO (*beacon* de découverte OLSR) et l'ajout de la route dans la table de routage. Notre ensemble statistique de test est composé de mille valeurs sur chaque valeur étudiée. La Figure 5.9 montre la répartition du délai mesuré pour les deux modes de communication qui sont :

— le mode non-chiffré (Première valeur)
— le mode chiffré (Troisième valeur)

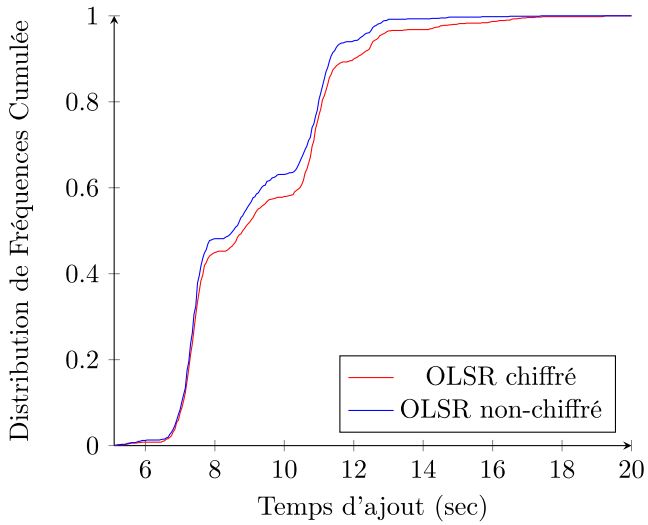


FIGURE 5.9 – Comparaison du délai d'ajout de voisin

1. http ://www.secdev.org/projects/scapy/

	Interception simple	OLSR non-chiffré	OLSR chiffré
Moyenne (sec)	9.01	9.08	9.35
Ecart-type (sec)	2.07	2.53	2.35

TABLE 5.2 – Comparaison du délai d'ajout du voisin selon le test

Le Tableau 5.2 résume les valeurs statistiques afin de pouvoir comparer facilement ce délai. OLSR en mode non-chiffré ajoute plus rapidement le voisin que le mode chiffré, ce qui est normal. Cependant, cet écart est minime, surtout si l'on considère qu'il faut chiffrer trois paquets pour ajouter le voisin car un voisin n'est ajouté par OLSR qu'après réception de 3 HELLO. L'écart induit entre les deux modes est inférieur à 300 ms. La Figure 5.9 nous montre que le délai d'ajout de voisin en mode non-chiffré est plus court, mais que le mode chiffré ne montre pas une dégradation de performance importante (seulement 300 ms en moyenne). Notre preuve de concept basée sur les règles *iptables*, l'interception NFQUEUE et le chiffrement par Python montre que l'ajout d'éléments cryptographiques ne dégrade pas les performances de routage au sein d'un groupe de communication. Le délai peut encore être réduit avec une implémentation à un plus bas niveau, c'est-à-dire directement dans le code OLSR ou le driver de la carte, ou grâce à l'utilisation d'un autre langage plus adapté que Python pour dialoguer avec le matériel ou faire des opérations cryptographiques.

Nous avons également mesuré le délai de création d'une clé partagée entre deux nœuds, dans le cas où elles ne serait pas pré-chargée en début de mission. Sur ce test conduit mille fois, le temps moyen de création est de 0,5 seconde. Nous trouvons ce délai acceptable car cette opération n'est faite qu'une seule fois pour établir un lien inter-groupe. De plus, il est facilement optimisable avec la planification de mission et l'utilisation de clé pré-enregistrées puisque les accords politiques sont connus.

Évaluation de ITMAN selon les critères définis en État de l'art :

Dans cette partie, nous allons revenir sur la liste de critères que nous avons défini dans la Section 3.2 et intégrer ITMAN dans le Tableau 5.3. Voici les éléments qui caractérisent ITMAN pour chacun des éléments de la liste :

— Maîtrise de l'adressage IP : Comme nous avons pu le voir dans le Chapitre 4, il n'y a pas de nécessité à gérer la duplication ou le recouvrement de préfixe. Le seul élément à maîtriser est la non-duplication d'une adresse IP. Pour cela, la préparation de mission en amont du déploiement doit pouvoir éviter ce problème.

— Interfaçage des autres groupes : Dans ITMAN, nous utilisons le protocole OLSR dans nos groupes de communications. ITMAN utilise la cryptographie autour du protocole intra-domaine pour créer les groupes et les interconnecter. Nous pensons que l'utilisation d'autres protocoles proactifs à la place d'OLSR est envisageable. L'interconnexion entre différents protocoles proactifs, sous la condition que les nœuds puisse interpréter chacun des protocoles, est également possible. Concernant les protocoles réactifs, nous pensons qu'il est possible de les utiliser avec ITMAN. De la même manière que les protocoles proactifs, il est nécessaire que les nœuds soient compatibles si plusieurs protocoles sont utilisés. Pour finir, la connexion entre un groupe proactif et un protocole réactif ne nous semble pas possible sans l'ajout de nouvelles fonctionnalités. En effet, un nœud avec un protocole réactif possède une table de routage qui est vide la majorité du temps.

— Sécurisation des échanges : ITMAN utilise le chiffrement symétrique pour réaliser l'envoi d'informations sécurisées au sein d'un gruope. À l'aide de l'algorithme Diffie-Hellman, le protocole est également capable de se connecter à d'autres groupes de manière sécurisée. Aucune information n'est envoyée en clair.

— Définition des groupes : Dans ITMAN, un groupe est défini par un identifiant de groupe *OGN* et sa clé de chiffrement associée. Le groupe est alors défini par l'ensemble des membres possédant la même clé, puisqu'ils sont les seuls à pouvoir déchiffrer les messages.

— Dynamique des groupes : Concernant la séparation des groupes, elle est supportée lorsqu'elle intervient par mobilité. En effet, même si les groupes s'éloignent, ils restent confinés par

leur clé de chiffrement. De plus, ils restent joignables l'un envers l'autre à distance grâce au test de connectivité que nous avons mis en place. Si la séparation vient d'un ordre, il est nécessaire de distribuer de nouvelles clés pour les membres concernés. Pour la fusion des groupes, la politique Merge permettant d'avoir une clé de chiffrement partagée avec un autre groupe. Ainsi, lorsque les membres des deux groupes sont à proximité, ils sont capables de communiquer à l'aide de cette clé partagée.

	Maîtrise de l'adressage IP	Interfaçage des autres groupes	Échanges sécurisés	Définition des AS	Dynamique des AS
BGP [RL95]	Oui	Oui	Non	Oui	Non
Inter-MR [LWC+10]	Oui	Oui	Non traité	Oui	Oui
BGP-MR [OKG14]	Oui	Non	Non traité	Oui	Oui
CIDR [ZCG09]	Oui	Oui	Non traité	Oui	Partiellement
BGP-MX [KTRH11]	Oui	Oui	Non traité	Oui	Non
ITMAN [GGKP16c]	Oui	Partiel	Oui	Oui	Oui

TABLE 5.3 – Comparatif de ITMAN et des protocoles étudiés dans l'état de l'art

Bien que nous ayons fait la proposition d'un protocole permettant l'interconnexion sécurisée de MANET et appliquant des politiques de routage, quelques éléments et cas particuliers doivent être discutés.

5.5 Limites actuelles et pistes d'amélioration

Dans cette première version d'ITMAN, nous avons identifié deux limites qui sont directement liées à la construction du protocole.

La première limite est la connectivité entre des groupes qui ne sont pas directement voisins. En effet, avec l'utilisation de politiques de routage qui restent simple (connexion ou non), des blocages sur des potentielles communications multi-saut se créent. Dans le cas de l'exemple de la Figure 5.10, bien que les groupes A et C soient joignables et autorisés à communiquer, le refus entre B et C empêche l'échange d'informations.

La deuxième limite est la divulgation non contrôlée d'informations. En effet, avec l'utilisation d'un protocole proactif, les informations de routage sont envoyées régulièrement par chaque nœud. Lorsqu'un groupe est interconnecté avec un autre, il lui envoie ses informations de routage qui vont être redistribuées à d'autres groupes éventuels. Dans le cas de l'exemple de la Figure 5.11, une fois que le groupe B a appris les nœuds accessibles de A, le protocole OLSR va automatiquement les envoyer vers C auquel il est également interconnecté. Cependant, A et C ont une politique Deny et souhaitent potentiellement garder vis-à-vis de l'autre leur topologie confidentielle.

Ces constatations nous ont motivés à approfondir le travail sur l'interconnexion des groupes. Le travail proposé remplit les fonctionnalités de cloisonnement de groupes, d'échanges sécurisés et de connexion inter-MANET. La suite de nos travaux va consister en l'amélioration des politiques de routage pour améliorer la connectivité entre les groupes. Ce chapitre a fait l'objet d'une publication [GGKP16c].

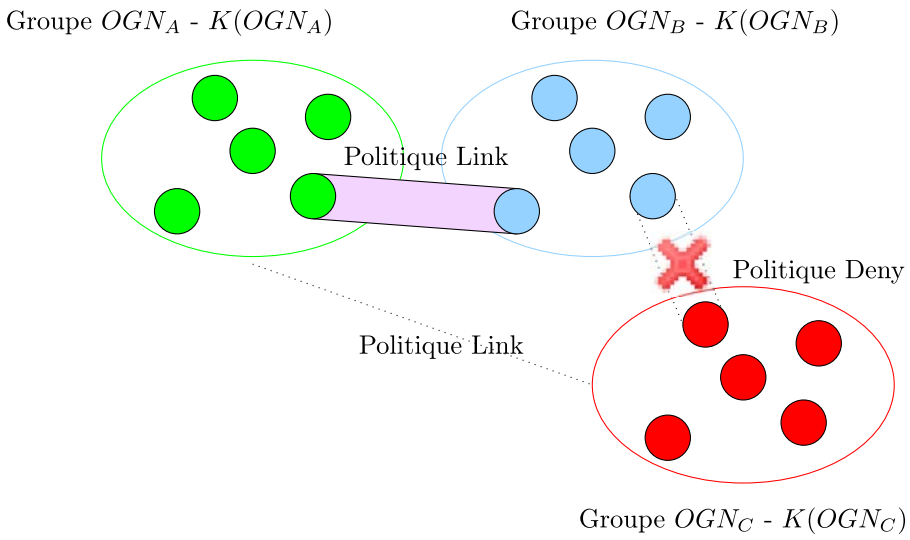


FIGURE 5.10 – Blocage multi-saut

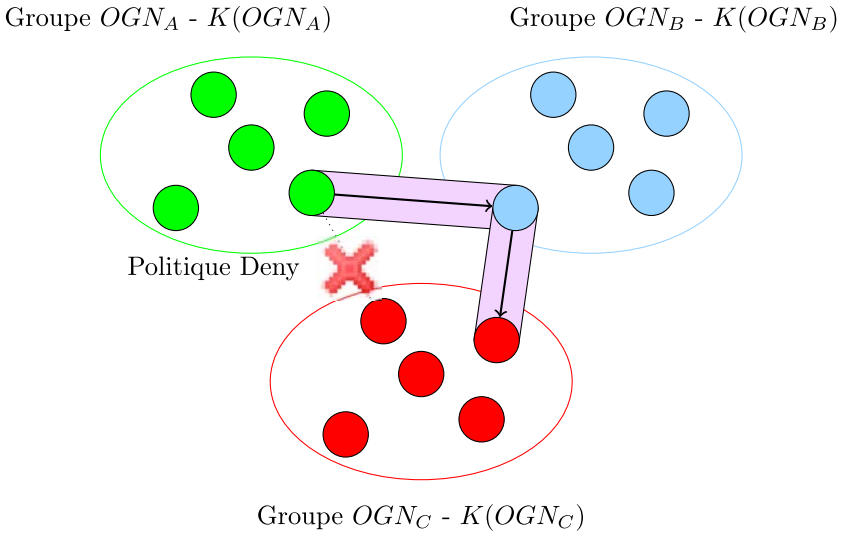


FIGURE 5.11 – Divulgation non contrôlée

Chapitre 6

Politiques avancées

Comme décrit en conclusion du précédent chapitre, le but de ce chapitre est d'améliorer les politiques de routage de ITMAN. Ces politiques, bien qu'autorisant un nombre défini de types de communications, peuvent se résumer au simple fait d'accepter ou non une communication. Il est vite apparu que des situations bloquantes pouvaient apparaître si trois groupes ou plus étaient présents dans la coalition. Deux problématiques ont été mises en évidence. D'un côté, la non-accessibilité de communication à plus d'un saut. De l'autre, il peut y avoir de la diffusion non-contrôlée de l'information. Dans ce chapitre, nous évaluons dans un premier temps les fréquences d'apparition de ces blocages. Ensuite, nous proposerons deux alternatives aux politiques de routage simple que nous mettrons à l'épreuve lors d'une expérience.

6.1 Évaluation théorique d'apparition de cas bloquants

Afin de valider la pertinence de cette étude qui nous est apparu suite à la proposition d'ITMAN, nous avons quantifié les cas bloquants en fonction de la taille de la coalition (nombre de groupes). Cette évaluation se fera sur les deux problèmes évoqués : le blocage multi-saut et la divulgation non-contrôlée.

6.1.1 Notation et modélisation utilisées

Avant de voir l'évaluation et ses résultats, nous définissons dans cette partie le cadre et les éléments qui vont être utilisés. Nous travaillerons pour des coalitions allant de deux à six groupes. Le nombre de nœuds dans chaque groupe n'est pas un élément déterminant. Nous cherchons à évaluer les connexions inter-groupe, qui sont indépendantes du nombre de nœuds les constituant.

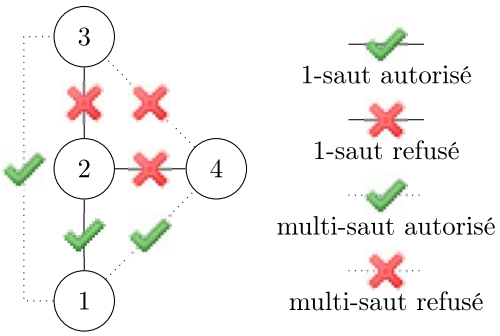
Pour pouvoir appliquer des algorithmes et des calculs sur nos topologies réseaux, nous avons utilisé une représentation matricielle. La connectivité entre les groupes est définie par une matrice M_c (matrice de connectivité) et les politiques appliquées entre les groupes est définie par une matrice M_p (matrice de politique). Ces deux matrices sont carrées et de taille n, où n désigne le nombre de groupes dans la coalition. Chaque groupe est alors désigné par un numéro allant de 1 à n. Pour illustrer les matrices M_c et M_p, nous utiliserons la topologie exemple de la Figure 6.1a.

La représentation des connexions inter-groupes de la matrice M_c, illustrée sur la Figure 6.1b, est construite comme suit :

— $M_c[i][j] = 1$: les groupes i et j sont connectés (pour $i \neq j$),
— $M_c[i][j] = 0$: les groupes i and j ne sont pas connectés (pour $i \neq j$),
— $M_c[i][j] = M_c[j][i]$: les liens sont symétriques,
— $M_c[i][i] = 2$: connexion non valable (pour $i = j$).

La représentation des politiques inter-groupes de la matrice M_p, illustrée sur la Figure 6.1c, est construite comme suit :

— $M_p[i][j] = 1$: les groupes i et j sont autorisés à s'échanger des données (pour $i \neq j$),
— $M_p[i][j] = 0$: les groupes i et j ne sont pas autorisés à s'échanger des données (pour $i \neq j$),
— $M_p[i][j] = M_c[j][i]$: les politiques sont symétriques,
— $M_p[i][i] = 2$: politique non applicable (pour $i = j$).



(a) Exemple de réseau ad hoc et ses politiques associées

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 1 \\ 0 & 1 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}$$

(b) Représentation matricielle de la topologie (Matrice M_c)

$$\begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$$

(c) Représentation matricielle des politiques inter-groupe (Matrice M_p)

FIGURE 6.1 – Utilisation des matrices M_c et M_p pour la représentation de communications et de politiques ad hoc inter-groupe

Le but de l'évaluation est de rechercher des cas problématiques sur toutes les combinaisons topologie/politique possibles. Par exemple, pour une coalition à quatre groupes, il existe 2^6 topologies possibles (nous utilisons des liens symétriques). De même, il existe 2^6 possibilités de politiques au sein du réseau. Nous appliquons chaque possibilité de politique à toutes les topologies existantes. Pour évaluer la proportion de communications bloquées dans une coalition de quatre groupes, nous devons donc étudier 4 096 cas possibles (les 64 politiques différentes appliquées sur les 64 topologies possibles).

Nous procéderons à une évaluation sur deux niveaux :

— Niveau réseau : à partir du moment où un couple de groupes est bloqué, le réseau entier est considéré comme bloqué. Dans l'exemple de la Figure 6.1b, le couple 1-4 est bloqué par le lien 2-4 donc le calcul est arrêté pour cette combinaison topologie/politique.
— Niveau groupe : même si un couple est bloqué, les autres sont testés. Toujours dans le cadre de la Figure 6.1b, les couples 1-4 et 1-3 sont bloqués. Ainsi, le réseau est bloqué à 33,33% pour cette combinaison topologie/politique.

Pour chaque association topologie/politique, nous calculerons pour chaque groupe deux tables de routage. La première, nommée Tr_i, permet de connaître quelle sont les destinations accessibles à partir du groupe i. Cette table est remplie avec l'algorithme de Dijkstra [Dij71]. La deuxième table, nommée Tp_i, tient compte des politiques appliquées dans le réseau. Ainsi, la présence d'un refus sur la route du groupe i vers une destination fait que son coût passera automatiquement à l'infini. La Figure 6.2 représente Tr_1 et Tp_1, qui sont respectivement la table de routage normale et la table de routage politisée du groupe 1.

Après avoir introduit l'utilisation de matrices pour représenter un réseau de coalition (connectivité et politiques appliquées) ainsi que les évaluations que nous allons effectuer, nous allons calculer théoriquement la proportion de cas non-désirés de communications (blocages multi-saut et divulgation non-contrôlée) pour des réseaux tactiques. S'ils se révèlent trop importants, il sera alors nécessaire de proposer des politiques de routage approfondies et qui ne se limitent pas à accepter ou non des liens inter-groupes.

Table de routage Tr_1		
Destination	Prochain Saut	Distance
2	2	1
3	2	2
4	2	2

(a) Table de routage basique, via l'algorithme de Dijkstra

Table de routage politisée Tp_1		
Destination	Prochain Saut	Distance
2	2	1
3	-	∞
4	-	∞

(b) Table de routage politisée

FIGURE 6.2 – Politisation d'une table de routage

6.1.2 Communication multi-saut bloquée

Dans cette partie, nous étudions spécifiquement le premier cas problématique : la communication multi-saut bloquée. Elle est définie par la présence d'un lien *Deny* entre deux groupes voisins sur la route qui relie deux groupes distants autorisés à communiquer. Ce cas est rappelé par la Figure 6.3.

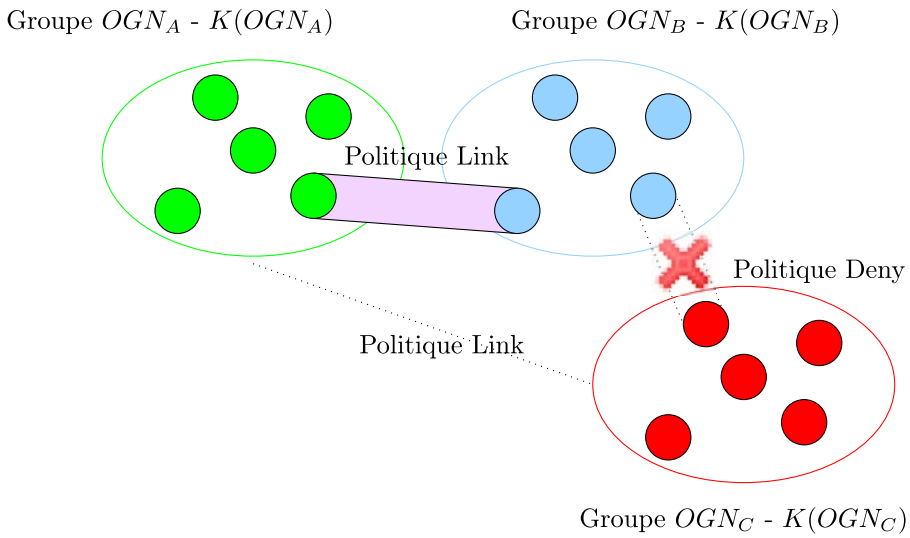


FIGURE 6.3 – Blocage multi-saut

Afin de pouvoir appliquer un algorithme de calcul sur la modélisation que nous avons présentée, nous devons trouver les caractéristiques génériques d'une communication bloquée par multi-saut. Généralement, ce cas arrive lorsque les conditions suivantes sont satisfaites :

— A et B sont autorisés à communiquer (M_p [A][B] = 1),
— A et B ne sont pas directement connectés (M_c [A][B] = 0),
— A et B peuvent se joindre via la table de routage (Tr_A [Distance][B] > 0),
— A et B ne peuvent pas se joindre via la table de routage politisée (Tp_A [Distance][B] = ∞).

À partir de ces critères, nous pouvons mesurer la probabilité d'apparition d'une communication bloquée par multi-saut, en fonction du nombre de groupes présents dans la coalition. Le Tableau 6.1 présente ces valeurs du point de vue groupe, et le Tableau 6.2 les présente du point de vue réseau.

Du point de vue d'un groupe, le taux de blocage montre qu'il y a peu de chance qu'un groupe n'atteigne pas sa destination au sein de la coalition, même en augmentant le nombre de ses alliés, puisqu'il dépasse à peine les 10%. Cependant, cela ne signifie pas que les politiques binaires suffisent

Nombre de groupes	2	3	4	5	6
Cas totaux	4	192	24 576	10 485 760	16 106 127 360
Cas bloquants	0	9	2 238	1 280 250	2 167 642 545
Ratio	0%	4,69%	9,11%	12,21%	13,46%

TABLE 6.1 – Blocage multi-saut du point de vue “groupe”

Nombre de groupes	2	3	4	5	6
Cas totaux	4	64	4 096	1 048 576	1 073 741 824
Cas bloquants	0	9	1 750	747 369	953 953 191
Ratio	0%	14,06%	42,7%	71,3%	88,8%

TABLE 6.2 – Blocage multi-saut du point de vue “réseau”

et n’ont pas besoin d’amélioration. En effet, si nous nous plaçons au niveau réseau, nous voyons que pour des coalitions de quatre groupes ou plus, des informations peuvent être bloquées avec une probabilité supérieure à 40%, voire jusque 90% pour des coalitions de six groupes. Nous avons pu voir que ce pourcentage augmente très rapidement par rapport à la taille de coalition. Pour des coalitions de 7 groupes et plus, il est donc raisonnable de penser que le taux de blocage est compris entre 90% et 100% de blocage. En se basant sur le fait que dans une coalition, chaque membre a une importance et qu’il est nécessaire que les informations transitent, ce taux de blocage n’est pas satisfaisant et il est donc nécessaire de trouver un moyen de réduire voire d’annuler ces blocages. Nous allons maintenant étudier la fréquence d’apparition des cas de divulgation non-contrôlée, après en avoir défini les conditions.

6.1.3 Divulgation non-contrôlée

Dans cette partie, nous étudions le deuxième cas problématique causé par l’utilisation de politiques de routage binaires : la divulgation non-contrôlée. Cette situation arrive lorsque deux groupes ne souhaitent pas communiquer. S’ils peuvent tous les deux joindre un groupe avec lequel ils sont autorisés à communiquer, alors, ce groupe tiers transmettra les informations de routage de l’un à l’autre et réciproquement. Ce cas est rappelé par la Figure 6.4.

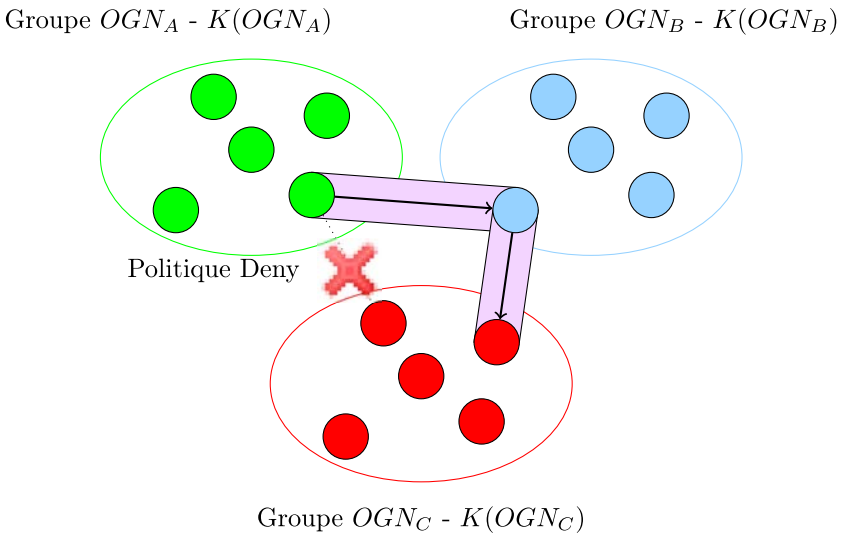


FIGURE 6.4 – Divulgation non-contrôlée

De la même manière que pour le blocage multi-saut, nous évaluons la probabilité d’apparition du cas de diffusion d’information non-contrôlée, en fonction du nombre de groupes présents dans la

coalition. Le Tableau 6.3 présente ces valeurs du point de vue réseau, et le Tableau 6.4 les présente du point de vue groupe.

De manière générique, une divulgation non-contrôlée d'une information du groupe A vers un groupe B se produit lorsque les conditions suivantes sont satisfaites :

— A et B ne sont pas autorisés à communiquer (M_p [A][B] = 0),
— A et B peuvent se joindre via la table de routage (Tr_A [Distance][B] > 0),
— Si au moins un groupe i, où i est accessible par A via la table de routage politisée Tp_A, est capable de joindre B via sa table de routage politisée Tp_i ($\exists$ i $\in$ [1 ; n]$\backslash\{A; B\}$, Tp_A [Distance][i] > 0 et Tp_B [Distance][i] > 0).

Nombre de groupes	2	3	4	5	6
Cas totaux	4	64	4 096	1 048 576	1 073 741 824
Cas bloquants	0	6	1 196	561 524	803 705 184
Ratio	0%	9,38%	29,20%	53,55%	74,85%

TABLE 6.3 – Divulgation non-contrôlée du point de vue "réseau"

Nombre de groupes	2	3	4	5	6
Cas totaux	4	192	24 576	10 485 760	16 106 127 360
Cas bloquants	0	6	2 712	2 685 120	7 811 595 552
Ratio	0%	3,13%	11,04%	25,61%	48,50%

TABLE 6.4 – Divulgation non-contrôlée du point de vue "groupe"

L'étude des deux tableaux nous indique que la diffusion non-contrôlée de route apparaît beaucoup plus fréquemment que le blocage multi-saut. Pour des coalitions de cinq groupes, il existe une probabilité supérieur à 25% que les routes diffusées à un voisin de confiance soient redistribuées à un groupe de moindre confiance. Dans une coalition de même taille, il existe une probabilité supérieure à 50% pour qu'au moins un cas comme celui-ci apparaisse. Il est donc, là aussi, nécessaire de proposer des solutions alternatives aux politiques de routage actuelles pour que les souhaits politiques des groupes de la coalition n'empêchent pas les communications.

6.2 Pistes d'approfondissement des politiques de routage

Après avoir constaté la nécessité d'améliorer les politiques de routage, nous proposons de deux solutions permettant de résoudre chacun des deux problèmes précédemment cité. À partir de maintenant, il n'est plus question d'utiliser des politiques de routage de type *Deny*, *Link* ou *Merge*. Le but est de proposer de nouvelles fonctionnalités permettant de créer des communications inter-domaine et de ne pas les bloquer par des politiques se résumant à autoriser la connexion ou non. Pour le bon fonctionnement de ces solutions, nous considérons que les groupes sont pleinement coopératifs et participent à la construction du réseau, quelque soit le voisin à sa portée. Nous proposons, à partir de cette topologie construite, deux manières d'éviter le blocage multi-saut ou la divulgation non-contrôlée.

6.2.1 Résolution du blocage multi-saut : les tunnels

Lorsque deux groupes sont séparés par plusieurs sauts et souhaitent communiquer, ils utilisent d'autres groupes comme route pour s'échanger des informations. Bien que les groupes distants soient d'accords pour s'échanger des données, il se peut que deux groupes voisins sur la route n'aient pas d'accords politiques pour communiquer. Nous nous retrouvons alors face à une situation où deux groupes sont joignables par une route mais cette route est bloquée car deux groupes sur la route ne partagent pas une politique d'accord. Nous sommes face à un blocage multi-saut. Pour répondre à ce problème, nous proposons d'utiliser des tunnels. À l'instar des clés partagées dans ITMAN, chaque groupe est capable de créer un tunnel bout-en-bout jusqu'à sa destination. Pour cela, deux

modifications sont apportées au protocole ITMAN original. Tout d'abord, les groupes doivent être coopératifs et donc établir les liens inter-domaine. Le fait de communiquer ou non avec un autre groupe ne se décide plus à la connexion, mais à l'établissement du tunnel. Ainsi, il faut modifier le fichier de politique associé à chaque nœud. Il devient alors un ensemble de groupes politiquement alliés, et où chaque groupe est associé à une clé de chiffrement. Comme pour ITMAN, il y a une priorisation des clés de déchiffrement en réception. Il est plus fréquent pour un nœud de recevoir un paquet OLSR de son groupe. Ainsi, la clé de groupe est la première à être utilisée pour le déchiffrement. Ensuite, le choix entre la clé partagée à un saut ou la clé partagée multi-saut est fait selon le nombre détenu pour chaque type de clé (ex : si un nœud possède une clé à un saut et deux multi-saut, les clés multi-saut seront d'abord testées).

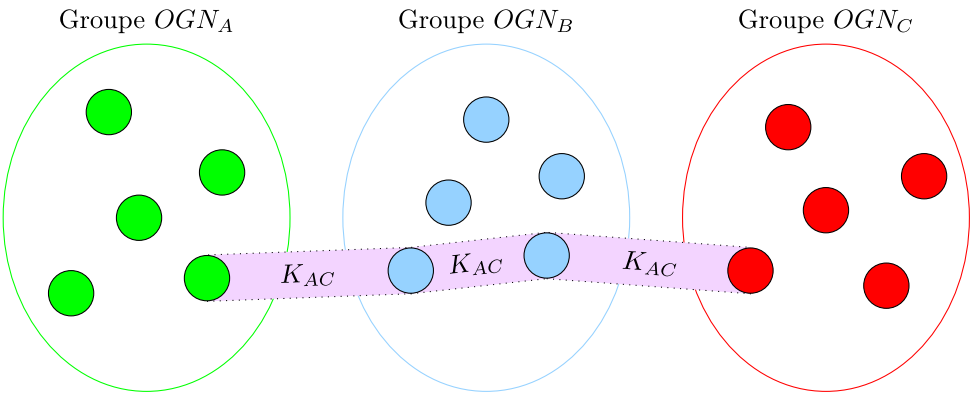


FIGURE 6.5 – Exemple d'utilisation des tunnels avec une clé partagée entre OGN_A et OGN_C

Avec cette piste, nous pensons arriver à un compromis entre transport des communications et respect des politiques. Bien que des groupes peuvent avoir des niveaux de confiance faibles, s'ils sont sur l'unique route possible, il faut pouvoir utiliser ce groupe comme route. Avec les tunnels chiffrant de groupe en groupe les données, les communications ont lieu quelque soit la confiance attachée aux différents membres de la route. Le seul critère obligatoire pour communiquer est la disponibilité de la route. Nous allons maintenant étudier une solution pour la résolution de la divulgation non-contrôlée : le filtrage d'annonces.

6.2.2 Résolution de la divulgation non-contrôlée : le filtrage d'annonces

Lorsqu'un groupe émet ses informations de routage à un groupe de confiance, il n'a plus de contrôle sur ce que arrive à ces données une fois à destination. Dans notre cas d'usage, le protocole OLSR distribue naturellement les routes auprès des groupes alliés. Cependant, il est possible que les politiques partagées avec les groupes à plus d'un saut ne soient pas de confiance. Ainsi, le groupe voisin allié va retransmettre les informations de routage dans un groupe non souhaité. Nous parlons alors de divulgation non-contrôlée. La solution que nous proposons face à la divulgation non-contrôlée des routes est le filtrage. À l'identique du protocole BGP, chaque groupe peut choisir de restreindre l'annonce de nœuds de son réseau et conserver uniquement des nœuds "passerelles" comme le suggère la Figure 6.6. Le fichier de politique devient alors, pour les membres d'un groupe, un ensemble d'adresses IP de son groupe qui sont les nœuds à ne pas annoncer sur les canaux partagés avec les autres groupes. Cette distinction se fait simplement au moment d'envoyer un paquet OLSR. ITMAN chiffre le paquet pour chaque clé différente qu'il possède. Ainsi, au moment de chiffrer avec les clés partagées, ITMAN applique la suppression des adresses contenues dans le paquet OLSR. Ainsi, un groupe peut préserver la diffusion de tous ses nœuds tout en laissant des passerelles disponibles aux autres groupes pour transporter des communications. Un nœud qui est présent dans la liste de filtrage ne peut pas devenir passerelle et suspend donc sa diffusion de beacon. Si par ordre hiérarchique, le nœud est supprimé du filtrage, il est à nouveau capable d'envoyer des beacons pour découvrir, trouver des voisins et établir des liaisons inter-groupes.

Dans la prochaine partie, nous allons voir la construction d'un paquet OLSR afin de pouvoir établir notre algorithme de filtrage et résoudre la divulgation non-contrôlée.

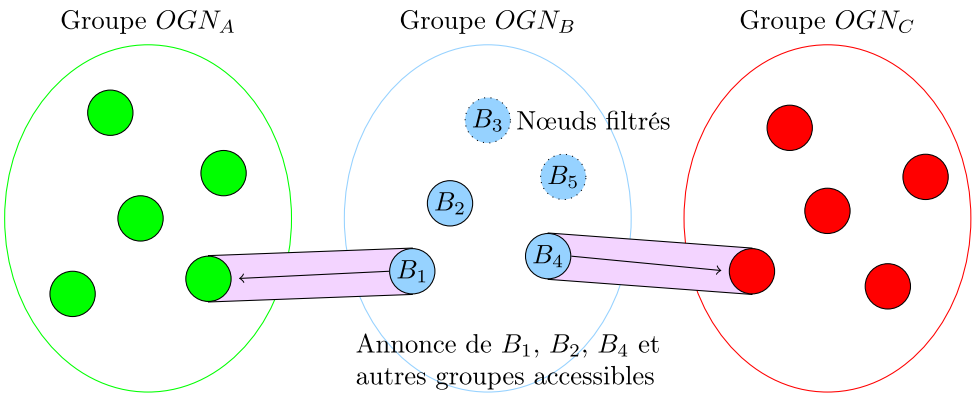


FIGURE 6.6 – Exemple d'utilisation de filtrage d'annonce

Construction d'un paquet OLSR

Pour pouvoir traiter efficacement les annonces OLSR et supprimer les adresses IP qu'un groupe ne souhaite pas divulguer, il faut connaître la structure du paquet. Selon la norme RFC 3626 du protocole OLSR[CJ03], un paquet OLSR est composé de la manière suivante :

— un en-tête de 4 octets, qui comporte la taille du paquet OLSR (2 octets) et un numéro de séquence (2 octets)
— un ou plusieurs messages OLSR. OLSR comporte deux types de message : HELLO et Topology Control (TC)

Bien que les messages soient de deux types différents, ils possèdent le même début de construction, comme illustré sur la Figure 6.7 :

— *Message Type* : indique si le message est un HELLO ou un TC
— *Vtime* : durée de validité du message
— *Message Size* : taille du message
— *Originator Address* : émetteur original du message
— *Time To Live* : nombre de sauts restant avant destruction du paquet
— *Hop count* : nombre de nœuds que le message a traversé
— *Message Sequence Number* : numéro de séquence du message

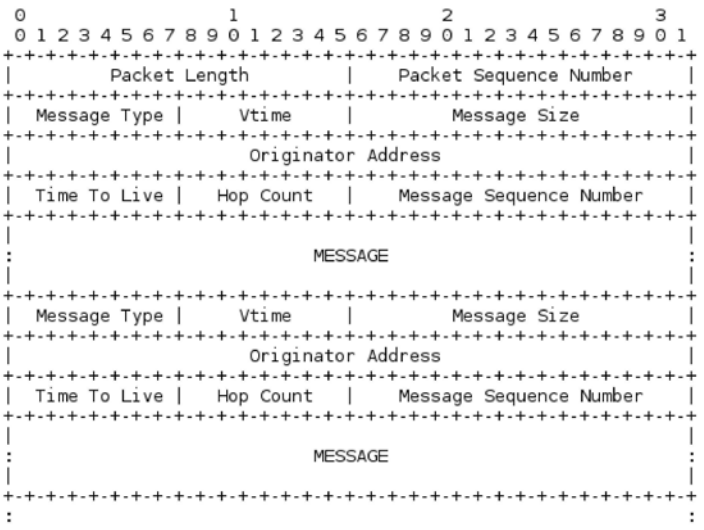


FIGURE 6.7 – Construction globale d'un paquet OLSR

Nous allons maintenant voir comment sont construits ces deux types de message et étudier leurs champs afin d'établir un algorithme de filtrage.

Message HELLO

Le premier message existant, de type HELLO, permet à un nœud OLSR d'avertir ses voisins proches de sa présence. Dans les réseaux sans fil, l'accessibilité d'un nœud varie selon plusieurs facteurs qui sont majoritairement la portée, l'environnement et le niveau de réception. Ainsi, un nœud n'établit pas obligatoirement la même connexion avec tous ses voisins. Dans un message HELLO, quatre types de liens sont utilisés :

— *Symmetric Link* : les deux nœuds envoient et reçoivent correctement les paquets
— *Asymmetric Link* : un des deux nœuds ne reçoit pas les paquets
— *Lost Link* : le lien est perdu
— *Unspecified Link* : aucune information sur le lien

La RFC 3626 du protocole OLSR nous donne la construction d'un message HELLO illustrée par la Figure 6.8. Ainsi, deux champs supplémentaires introduisent le paquet :

— *HTime* : intervalle d'émission des messages HELLO, permet de déterminer l'attente pour considérer un lien perdu qui est de 3 HELLO (1 octet)
— *Willingness* : permet d'influencer l'élection du nœud en tant que MultiPoint Relay (MPR) (1 octet)

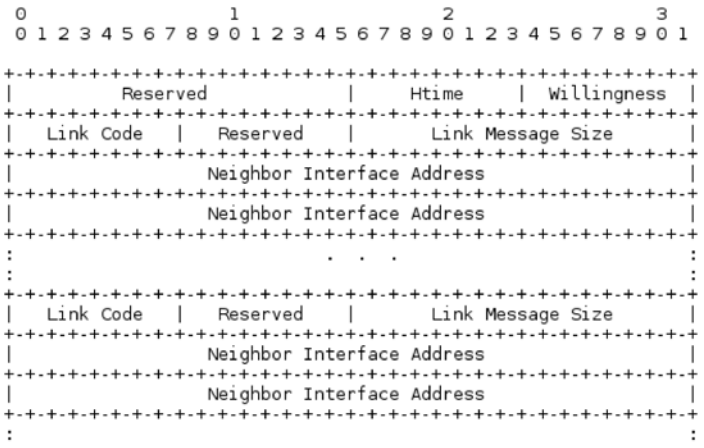


FIGURE 6.8 – Construction d'un message HELLO

Lorsqu'un nœud OLSR arrive dans le réseau, sans aucun voisin, il envoie un paquet HELLO contenant tous les arguments expliqués jusqu'à présent pour se déclarer auprès des nœuds alentours. Une fois qu'il trouve des nœuds proches, il va les annoncer dans son HELLO. Ces voisins vont être ajoutés et classés selon le type de lien qui est établi. Le type de lien est expliqué par un *Link Code* et sa longueur est indiquée par le *Link Message Size*. Les adresses IP des voisins correspondant à ce type de lien sont listées et chacune se voit affecter deux valeurs de 1 octet chacun : un LQ (Link Quality) et un NLQ (Neighbor Link Quality). Ces deux variables permettent alors de calculer le coût de la route par la métrique ETX (Expected Transmission Count) qui vaut $1/(LQ * NLQ)$. La déclaration d'un voisin dans un paquet OLSR utilise 8 octets (4 octets pour l'adresse IP, 2 octets réservés, 1 octet pour le LQ, 1 octet pour le NLQ).

Message TC

Pour commencer à envoyer des paquets TC, il faut qu'un nœud valide au moins un lien de type symétrique avec un autre nœud (trois HELLO successifs échangés) et qu'il soit élu en tant que Multi Point Relay. Seuls les MPR envoient des paquets TC. Le TC permet aux MPR d'informer les nœuds du réseau des destinations accessibles.

Le TC est construit de manière simple. Après avoir déclaré un ANSN (Advertised Neighbor Sequence Number) qui lie le MPR à son nœud voison qui l'a élu, la source du TC liste les adresses IP des voisins accessibles, chacune associée avec un LQ et un NLQ. Ce message est celui qui permet aux récepteurs d'enregistrer les nœuds dans les tables de routage et connaître l'ensemble des MPR

qui ont accès au réseau. Pour construire le réseau sur une grande vision, les MPR se transmettent leurs TC pour étendre l'apprentissage.

Filtrage des paquets OLSR

Après avoir vu la construction d'un paquet OLSR et des messages HELLO et TC, nous mettons en place un algorithme de filtrage qui permet de garder un paquet OLSR cohérent. D'après la description que nous avons pu en faire précédemment, des champs sont sensibles au retrait d'une adresse IP tels que la taille voire la présence d'un message TC s'il n'y a qu'un voisin d'annoncé.

Pour effectuer notre filtrage, nous commençons par séparer les messages que contient le paquet OLSR. Ensuite, nous traitons chaque message individuellement. Voici les différents cas de figure qui se présentent pour le retrait d'une adresse que nous nommerons IP_{filtre} :

- IP_{filtre} est dans un message HELLO et seul dans un type de lien : il faut retirer les 8 octets correspondant à IP_{filtre} (4 d'adresse, 2 réservés, 1 pour le LQ, 1 pour le NLQ) ainsi que les 4 octets servant de préambule de lien
- IP_{filtre} est dans un message HELLO et n'est pas seul dans un type de lien : il faut retirer les 8 octets correspondant à IP_{filtre} (4 d'adresse, 2 réservés, 1 pour le LQ, 1 pour le NLQ)
- IP_{filtre} est dans un message TC et est seul : il ne faut pas émettre le TC
- IP_{filtre} est dans un message TC et n'est pas seul : il faut retirer les 8 octets correspondant à IP_{filtre} (4 d'adresse, 2 réservés, 1 pour le LQ, 1 pour le NLQ)
- IP_{filtre} est l'émetteur d'un TC : il ne faut pas rediffuser le TC

Toutes ces conditions sont testées dans un ordre optimisé. L'algorithme de filtrage, regroupant les étapes qui viennent d'être listées est illustré sur la Figure 6.9.

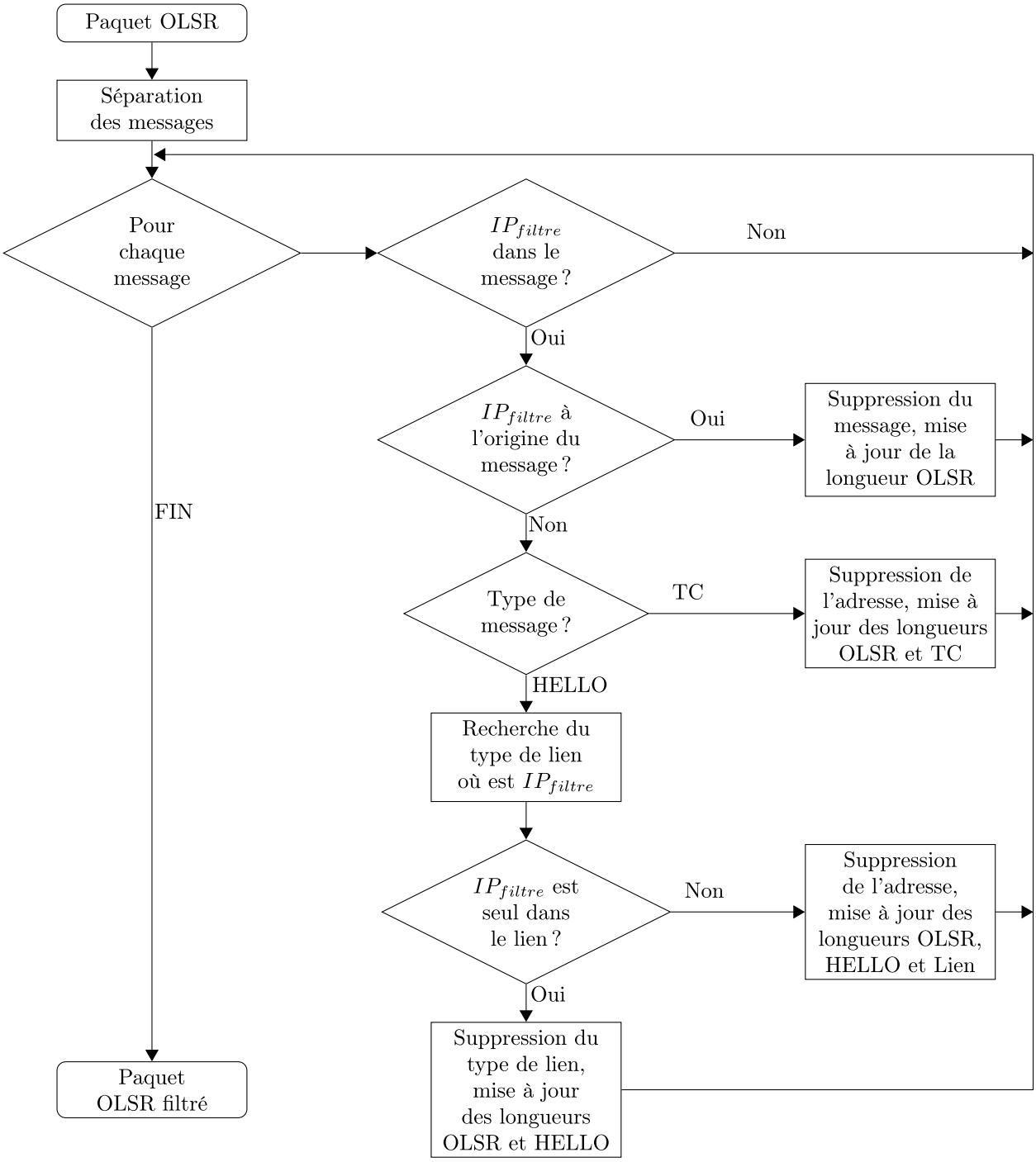


FIGURE 6.9 – Algorithme utilisé pour filtrer une adresse IP d'un paquet OLSR

6.2.3 Évaluation expérimentale des pistes de résolution

Dans cette partie, nous étudions les performances des deux solutions que nous avons proposées lors d'une expérience. Ces performances seront comparées avec un réseau OLSR simple afin de voir, comme pour ITMAN, la surcharge qu'apportent nos propositions sur les critères suivants :

— Débit d'échange : nous mesurons ici la vitesse à laquelle les données sont échangées
— Taux de perte : nous mesurons ici la proportion de paquets qui a été perdue
— Gigue : nous mesurons ici la variation de la latence de réception des paquets
— Délai moyen aller-retour : nous mesurons ici le temps que met un paquet pour faire un aller-retour entre la source et la destination

Scénario de simulation statique

L'expérience a été menée avec l'émulateur CORE sur le scénario illustré en Figure 6.10. Ce dernier se déroule de la manière suivante : trois groupes sont présents dans la coalition. Chaque groupe comporte trois nœuds. Le nœud n1 souhaite échanger des informations pendant 90 secondes avec n9 une fois que la route pour l'atteindre est apprise. Ce premier scénario nous permet d'évaluer, sur les quatre critères précédemment décrits, les altérations qui arrivent lorsque les tunnels groupe-à-groupe ou le filtrage d'annonce sont mis en place. Lorsque le filtrage est mis en place, le groupe bleu empêche l'annonce de n5 aux autres groupes. Lorsque le tunnel est mis en place, il est de bout-en-bout entre les groupes vert et rouge.

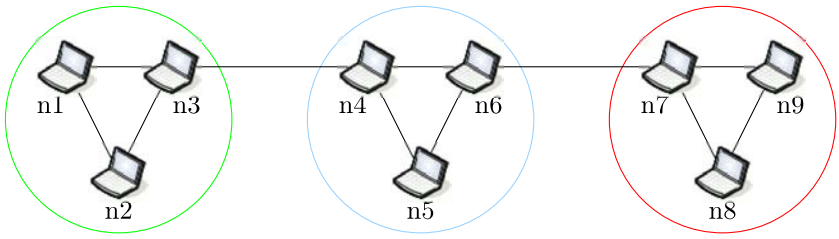


FIGURE 6.10 – Scénario d'évaluation statique

Pour nous aider dans l'évaluation et la mesure, nous avons utilisé l'outil *iperf* en mode UDP pour obtenir les données sur le débit d'échange, le taux de perte et la gigue ainsi qu'un échange de paquet pour mesurer le délai moyen aller-retour. Les caractéristiques des nœuds de l'émulateur ainsi que la configuration par défaut de *iperf* donne u débit d'envoi des données est de 1,05Mb/s. Le filtrage d'annonce et l'envoi des paquets via les tunnels ont été réalisés à l'aide de proxys écrits en Python. De la même manière qu'ITMAN, les paquets sont interceptés par des commandes *iptables* et placés en stockage dans une file d'attente NFQUEUE. Le proxy de filtrage ou de tunnel récupère alors les paquets dans cette file et va effectuer l'algorithme demandé avant d'envoyer le paquet à la carte réseau.

Le Tableau 6.5 liste les valeurs moyennes et l'écart-type des 4 critères réseau que nous évaluons. Ces données ont été récoltées sur la base de dix expériences pour chaque mode (OLSR, filtrage et tunnel). Sur la mise en place de ces solutions dans un réseau ad hoc statique, nous pouvons faire les constatations suivantes :

— la présence d'un mécanisme comme le filtrage ou le tunnel n'impacte le débit auquel les données sont transmises et aucun paquet n'est perdu
— le délai aller-retour est également peu impacté par la mise en place de filtrage ou de tunnel
— une augmentation de la gigue est présente sur le mode tunnel. L'utilisation de chiffrement et déchiffrement en permanence mobilise des ressources du nœud et entraîne donc cette variation de latence

Les graphiques 6.14 et 6.15 montrent l'évolution moyenne de la gigue et du débit au fil de la simulation. Aucune altération de débit n'est présente puisque les nœuds sont statiques et la vitesse

Configuration	OLSR seul	Filtrage	Tunnel
Débit réception (Mb/s)	1,050	1,050	1,050
$\sigma_{Debitreception}$ (Mb/s)	0,034	0,034	0,019
Gigue moyenne (ms)	0,061	0,073	0,507
$\sigma_{Giguemoyenne}$ (ms)	0,033	0,037	0,112
Délai moyen aller-retour (ms)	200,900	202,837	201,65
$\sigma_{Delaimoyenaller-retour}$ (ms)	0,038	0,786	1,197
Taux de perte	0%	0%	0%

TABLE 6.5 – Comparaison de performances sur quelques critères réseau pour le scénario statique

est de 1,05Mb/s. Aucun élément, en dehors d'une déconnexion ou panne d'un nœud de la route, n'est susceptible de faire varier cet élément. Concernant la gigue, l'effet du mode tunnel s'observe très facilement puisque cette courbe se détache très nettement des deux autres modes.

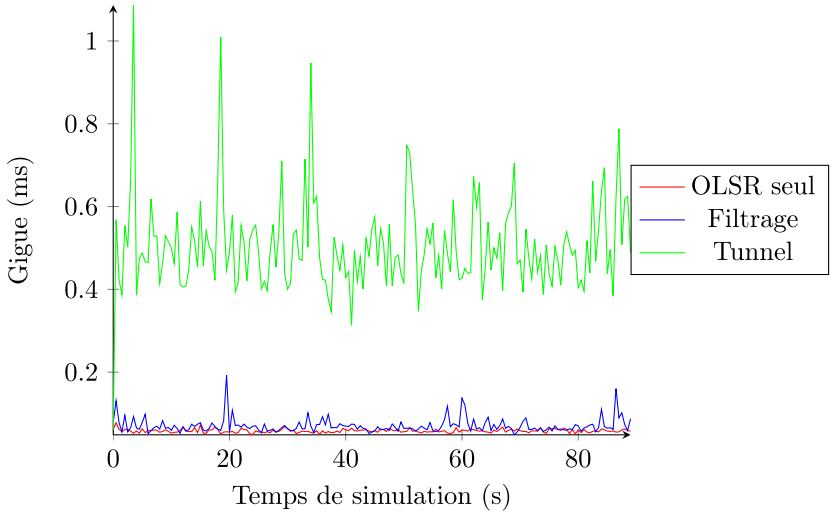


FIGURE 6.11 – Évolution de la gigue en fonction du temps de simulation sur le scénario statique

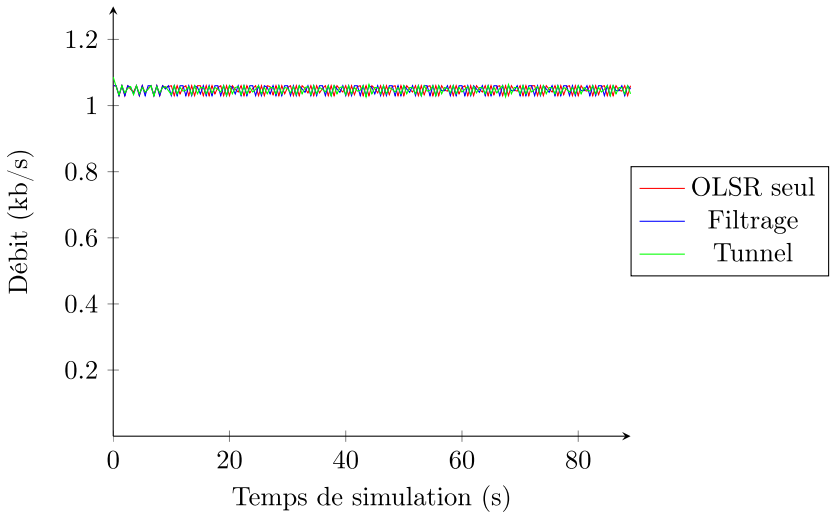


FIGURE 6.12 – Évolution du débit en fonction du temps de simulation sur le scénario statique

Les solutions que nous avons proposées pour répondre aux problèmes de blocage multi-saut et de divulgation non-contrôlée ne dégradent pas de manière importante les performances du réseau. Face à ce premier scénario statique, nous pouvons donc valider l'utilisation de ces deux alternatives. Cependant, pour tester la robustesse de nos solutions et voir leur impact sur les performances d'un réseau ad hoc mobile, nous proposons d'ajouter de la mobilité sur le nœud source n1. Le scénario de mobilité qui va altérer la mesure est expliqué dans la partie suivante.

Scénario de simulation mobile

Dans ce scénario mobile, nous gardons les groupes, leur disposition ainsi que le flux observé entre les nœuds n1 et n9. L'échange réussissant de manière statique et montrant des bonnes performances quelque soit la solution utilisée, nous avons choisi d'inclure de la mobilité afin de comparer la résistance de nos propositions face à une situation de déplacement. La mobilité, illustrée sur la Figure 6.13, se déroule de la manière suivante : de 0 à 25 secondes, n1 va se déplacer en restant à portée de n3 jusqu'à être à son nord. À partir de ce moment, n1 va continuer sa route vers l'est jusqu'à atteindre le nord de n9 à 90 secondes de simulation. Pendant ce temps, il aura subi des connexions et reconnexions à différents nœuds qui sont n4, n6 et n7. Le but est d'évaluer les altérations qu'apportent nos solutions en comparaison d'un protocole OLSR qui agirait seul.

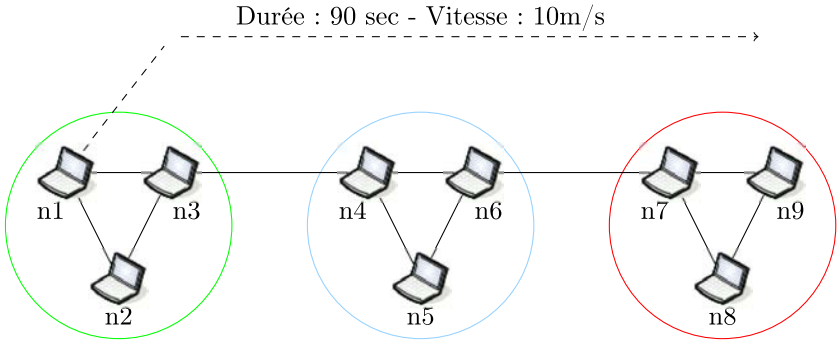


FIGURE 6.13 – Scénario d'évaluation mobile

L'ensemble des mesures de performance résumés dans la Table 6.6 et les Figures 6.14 et 6.15 nous permet de mettre en évidence plusieurs choses.

- Tout d'abord, des altérations de communication sont présentes, que ce soit au niveau du débit de communication ou bien des pertes de paquets. Dans ces deux catégories, il y a une dégradation des performances comprise entre 40% et 45%. Le mode tunnel est le mode qui affiche les moins bonnes performances, bien qu'elles soient très légèrement inférieures
- La gigue reste quasiment identique pour le mode OLSR seul. Une augmentation de 55% environ est observée pour les modes filtrage et tunnel à cause de la mobilité
- Le délai moyen aller-retour est inférieur au scénario statique. Comme n1 se rapproche tout au long du scénario, la distance à parcourir vers sa destination n9 est de plus en plus courte. Ceci est l'explication de la réduction de ce délai
- La mobilité crée une dispersion des valeurs observées. Quelque soit le critère évalué, l'écart-type augmente. Sur chaque élément mesuré, c'est le mode tunnel qui présente la plus grande dispersion de valeurs

La comparaison de nos propositions avec un réseau OLSR simple permet de montrer que ces écarts de performances qui peuvent avoir lieu ne sont pas causés par le filtrage ou la création de tunnel. En effet, ces dégradations sont présentes pour le cas de référence (OLSR seul) donc nous en déduisons que les baisses de performances sont dues à la mobilité.

Configuration	OLSR seul	Filtrage	Tunnel
Débit réception (Mb/s)	0,614	0,612	0,603
$\sigma_{Debitreception}$ (Mb/s)	0,487	0,488	0,486
Gigue moyenne (ms)	0,071	0,101	0,831
$\sigma_{Giguemoyenne}$ (ms)	0,146	0,207	0,235
Délai moyen aller-retour (ms)	166,660	168,127	171,923
$\sigma_{Delaimoyenaller-retour}$ (ms)	3,08	4,80	6,98
Taux de perte	41,7%	41,9%	42,4%

TABLE 6.6 – Comparaison de performances sur quelques critères réseau pour le scénario mobile

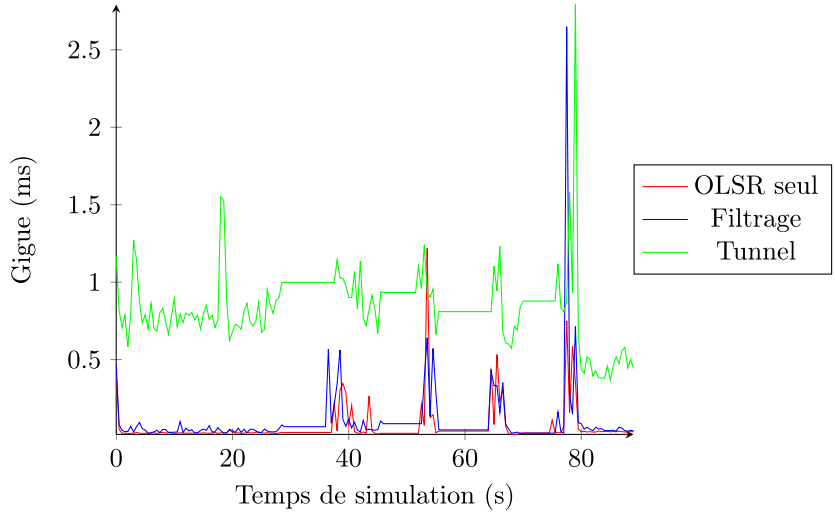


FIGURE 6.14 – Évolution de la gigue en fonction du temps de simulation pour le scénario mobile

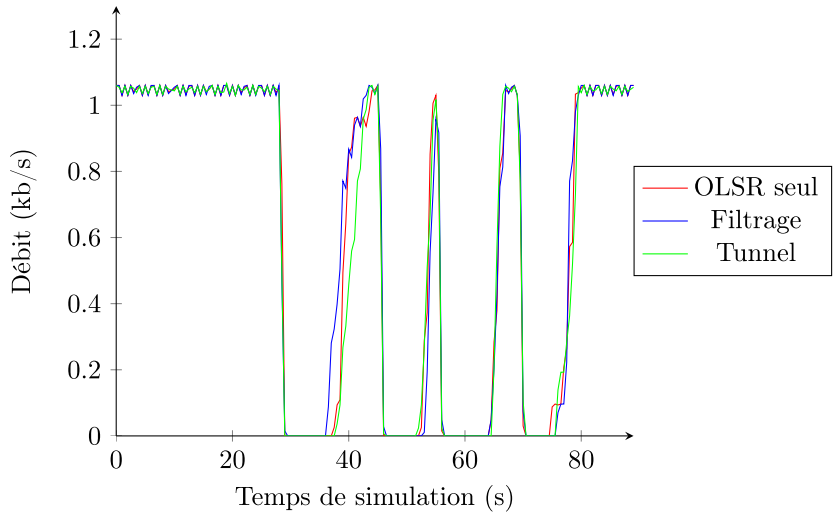


FIGURE 6.15 – Évolution du débit en fonction du temps de simulation pour le scénario mobile

Dans ce chapitre, nous avons été capable d'évaluer les problématiques d'ITMAN et plus particulièrement leur probabilité d'apparition au sein du réseau de coalition. Face à des apparitions très fréquentes, nous avons proposé deux alternatives aux politiques de routage binaires. D'un côté, la mise en place du filtrage d'annonces pour contrer le problème de divulgation non-contrôlée et de l'autre, la mise en place de tunnel groupe-à-groupe pour contrer le problème du blocage multi-saut. À travers deux scénarios (statique et mobile), nous observons qu'il est possible de mettre en place des mécanismes tels que le filtrage d'annonce ou les tunnels groupe-à-groupe pour contourner l'utilisation de politiques de routage binaires. La mobilité est l'élément qui amène les pertes de connexion sur les flux. Comme l'évalution d'ITMAN, notre mode expérimental n'est pas optimal puisque le codage du filtrage et des tunnels n'est pas au niveau matériel. Il y a un gain de performances à faire sur l'implémentation, mais cela ne change rien au fait que nos solutions n'entraînent pas de baisses de performances significatives (en comparaison d'une utilisation d'OLSR seul. Cette expérience montre que le filtrage et les tunnels sont des solutions alternatives à l'utilisation de politiques de routage simples.

Chapitre 7

Conclusion et Futurs travaux

Pour conclure ce manuscrit, nous proposons un récapitulatif de la problématique posée, de nos contributions tout au long de cette thèse ainsi qu'une ouverture avec quelques idées sur les travaux futurs à effectuer dans notre domaine de recherche.

7.1 Contexte et problématique

Au terme de cette thèse, nous avons pu voir le déploiement d'un théâtre d'opérations militaire ainsi que les critères et conditions associés. Ces éléments nous ont donné un premier cadre de travail de la thèse. À partir de là, nous avons vu l'environnement des réseaux sans fil dans les trois modes qui existent aujourd'hui : infrastructure, ad hoc et MANET. L'objectif de cette thèse était de pouvoir réaliser l'interconnexion de différents MANET d'une même coalition. Ainsi, bien que les alliés doivent être coopératifs, ils ont également des souhaits de souveraineté et de niveaux de confidentialité sur les informations échangées dans la coalition. Nous avons alors pris pour modèle le protocole BGP, qui est la référence des communications inter-domaine pour les réseaux filaires. Il permet à différentes entités, comme des fournisseurs d'accès, de s'échanger des données en conservant leur réseau confidentiel et en appliquant des politiques de routage. C'est avec le protocole BGP qu'Internet fonctionne aujourd'hui. Cependant, l'adaptation du protocole BGP aux réseaux MANET n'est pas possible sans modification du protocole original. En effet, ayant été conçu pour les réseaux filaires, le protocole BGP demeure une solution peu réactive et administrée, ce qui est à l'opposé d'un réseau MANET qui est autonome, mobile et à topologie variable.

L'état de l'art dans le domaine de la communication inter-MANET nous a permis de voir une problématique majeure émerger : l'adressage des nœuds du réseau. De par la nature très dynamique de la topologie du réseau, il est très difficile de maintenir des groupes cohérents en adressage et de pouvoir diffuser les routes de manière fiable (*c'est-à-dire* éviter les duplications d'adresse et les recouvrements de préfixe). C'est pour cela que les protocoles de l'état de l'art s'appuient majoritairement sur le Filtre de Bloom comme solution d'adressage alternatif dans les réseaux MANET. Cependant, nous avons soulevé deux problèmes à ces raisonnements. Tout d'abord, les réseaux ad hoc fonctionnent avec un adressage plat (masque de 32 bits) ce qui fait qu'il n'y a pas de problèmes dans la diffusion des routes tels que la duplication ou le recouvrement. De plus, le Filtre de Bloom est une structure qui ne permet pas, tel qu'il nous est présenté dans la littérature, le retrait d'éléments. Il doit donc être réinitialisé et rempli fréquemment pour avoir une topologie à jour ce qui semble très proche d'une reconfiguration. Les protocoles de l'état de l'art ayant été validés sous simulateur, nous avons donc mis en place des tests permettant de vérifier l'existence de cette problématique de l'adressage des nœuds en conditions réelles.

7.2 Contributions

Dans cette première contribution que constitue la vérification des problématiques d'état de l'art, nous avons confronté le logiciel NS3, l'émulateur CORE et une plateforme expérimentale pour confirmer que, quelque soit l'environnement de test utilisé, la dynamicité des MANET créé des problèmes d'adressage au niveau IP. Les observations que nous avons pu faire sont les suivantes : NS3, CORE et la plateforme montrent des comportements différents au niveau Réseau sur des expériences que nous qualifions de basique. En effet, deux nœuds à portée sont capables de se joindre sur notre plateforme, quelques soient leur configuration IP alors que sous NS3 et CORE, un comportement filaire est observé puisqu'il faut absolument que les nœuds soient configurés sur le même sous-réseau IP. De cette observation, nous en avons déduit que sur un réseau ad hoc réel, chaque nœud est identifié de manière unique par son adresse IP et qu'il est donc impossible d'utiliser un préfixe IP pour définir un groupe. Nous avons alors fixé de nouveaux objectifs pour la création d'un protocole inter-MANET. Le premier objectif est la définition d'un groupe et de ses membres. Puisque l'agrégation n'est pas valide, il a fallu trouver un moyen alternatif de créer les différents groupes de communications. Ensuite, pour nous rapprocher au mieux du protocole BGP, il a fallu connecter ces différents groupes, appliquer des politiques de routage et sécuriser les échanges.

Tous ces objectifs ont été pris en compte pour définir le protocole ITMAN. Dans cette deuxième contribution, les groupes participant à la coalition sont définis par un chiffrement de groupe. Les informations de routage sont chiffrées de manière symétrique par une clé connue de tous les membres du groupe. Ainsi, seuls les membres du groupe sont capables de s'échanger des données et toute écoute extérieure est alors impossible. Pour créer des échanges inter-MANET, nous avons mis en place un système de *beaconing* permettant de signifier sa présence à d'éventuels groupe voisins. L'identité des nœuds est garanti par un système de certificats. Trois politiques de routage inter-MANET peuvent s'appliquer : *Deny* (refus de communication), *Link* (autorisation limitée) et *Merge* (autorisation complète). Ces liaisons inter-MANET sont établies par des tunnels chiffrés pour assurer les propriétés de sécurité telles que la confidentialité ou l'intégrité. Pour évaluer ITMAN, nous avons mesuré l'impact du chiffrement sur la création de la topologie du réseau. Le délai induit par le chiffrement sur l'ajout d'un nœud voisin est très faible, ce qui impacte peu la construction du réseau. Cependant, l'utilisation des politiques de routage telles que nous les avons définies bloque des communications entre des voisins à plusieurs sauts ou bien divulgue des informations de manière non-contrôlée.

La troisième partie de cette thèse a donc été consacrée à l'amélioration des politiques de routage pour permettre d'améliorer les communications entre les groupes. La pertinence de cet axe a été vérifiée par une étude théorique sur la fréquence d'apparition des deux types de cas bloquants : le blocage multi-saut et la divulgation non-contrôlée. Nous avons alors mis en place des tunnels groupe-à-groupe et du filtrage d'annonce pour respectivement résoudre les deux problèmes apparus dans la première version d'ITMAN. Les performances ont été mesurées sur un scénario mobile pour évaluer la réactivité des solutions face aux différents pertes et retours de connexions qu'il peut y avoir dans un MANET. La comparaison avec un réseau OLSR classique nous a montré une nouvelle fois que la mise en place de ces deux éléments n'apportent pas de dégradations importantes des performances du réseau.

7.3 Perspectives

Au cours de cette thèse, nous avons pu voir qu'il est important de connaître les outils utilisés et leur comportement pour pouvoir se rapprocher le plus possible d'un comportement réaliste. La création de MANET ainsi que leur interconnexion ont pu être réalisées tout en garantissant des fonctionnalités de politique de routage et de sécurité, sans sacrifice majeur sur les performances. Bien que notre exemple se base sur un réseau OLSR, il est tout à fait envisageable de le transposer sur un autre protocole de routage ad hoc, puisque notre travail s'est majoritairement construit autour du protocole de routage plutôt que sur la création de nouveaux algorithmes de route. La

dernière partie de la thèse montre que les limitations d'utilisation des politiques inter-MANET peuvent être assouplies pour améliorer l'accessibilité du réseau. Cependant, il n'existe pas que la solution technique. En effet, la nécessité de mettre en place de telles mécaniques est principalement liée au fait de respecter des volontés hiérarchiques. La technologie permet d'établir des connexions inter-MANET mais la mise en place de politiques complexifie la création des protocoles car il faut veiller au respect de ces règles supplémentaires entre les acteurs du réseau. C'est notamment ce que nous avons pu voir avec les cas de blocage multi-saut et de divulgation non-contrôlée.

Dans un premier travail futur concernant les communications inter-domaine de terrain, il peut être intéressant d'élargir les fonctionnalités et les hypothèses de fonctionnement de ITMAN. Pour le moment, ITMAN fonctionne au niveau Réseau uniquement et permet l'interconnexion de réseaux OLSR. Cela suppose que tous les membres du réseau utilisent le même protocole de routage interne mais cette hypothèse n'est pas toujours réaliste et peut être forte si on sort du contexte des coalitions militaires. Comme le protocole BGP, il faut qu'ITMAN puisse être totalement indépendant du protocole de routage intra-domaine utilisé. Il doit agir à un niveau supérieur par rapport aux protocoles de routage intra-domaine. Le fait de séparer les niveaux de fonctionnement du protocole intra-domaine et du protocole inter-domaine permet d'appliquer des attributs et des fonctionnalités distincts. Il devient alors possible de se rapprocher d'un modèle similaire au protocole BGP et d'utiliser des fonctionnalités telles que des préférences de groupe, du contrôle d'annonces envers les autres domaines ou des priorisations de passerelles lors d'un *multi-homing*. Cependant, il n'existe pas que des protocoles intra-domaine proactifs dans les réseaux ad hoc. Si un domaine choisit d'utiliser un protocole réactif, les tables de routage des nœuds sont vides la plupart du temps. Le protocole inter-domaine n'a alors aucune information à envoyer aux autres domaines. Nous envisageons alors la mise en place d'un beaconing, comme cela a pu être fait dans InterMR [LWC+10] ou des protocoles hybrides comme ZRP [HPS02], pour garantir la connaissance des nœuds dans les groupes réactifs. Ainsi, le protocole inter-domaine possède des informations de chaque groupe à transmettre.

En second lieu, il faut envisager l'implication de groupes non-militaires tels que des groupes de services secours dans le réseau. Cependant, ils ne participent pas aux préparatifs de mission et n'ont pas obligatoirement les mêmes capacités technologiques et de sécurité à leur disposition. Nous envisageons donc de travailler sur la création d'un réseau comprenant des entités militaires et non-militaires. Deux idées peuvent être approfondies pour cet axe. D'un côté, l'organisation militaire peut anticiper la distribution d'un matériel de sécurité simple aux groupes extérieurs à la coalition pour permettre une interconnexion. Il ne serait pas obligatoire d'aller jusqu'à l'utilisation de politiques de routage ou l'utilisation des certificats de sécurité prouvant la présence du nœud non-militaire dans l'opération. Le but est que ces entités, malgré le fait qu'elles ne participent pas à la préparation de mission, puissent garantir des propriétés de sécurité (notamment la confidentialité) sur l'envoi des informations vers les groupes militaires. La deuxième idée que nous avons est la suivante. Si l'entité non-militaire ne peut assurer par ses technologies la réception ou l'utilisation du matériel de sécurité, il faut alors que la coalition abaisse son niveau d'exigence. Cela peut se traduire par la présence de nœuds transitoires, chargés de faire le relais entre un réseau que l'on considère non-sécurisé (hors coalition) et le réseau militaire.

La troisième piste de travail que nous envisageons suite à cette thèse est l'utilisation d'une solution basée sur le Software-Defined Networking. Le SDN permet sur les équipements d'un réseau d'ajouter une couche dite "d'abstraction". Cette dernière permet de séparer la partie fonctionnelle du réseau de la partie décisionnelle. Sur le terrain, les nœuds conservent une mission exclusive de collecte et de transmission des données (la partie fonctionnelle) alors qu'en base arrière, le commandement possède un rôle de gestionnaire pour faire évoluer le réseau selon les besoins du moment (la partie décisionnelle). Aujourd'hui, notre idéal de fonctionnement, BGP, connaît de nouveaux modèles de transport tel que BGP-NLRI (Network Layer Reachability Information) [MRM+09] pour configurer les politiques de routage sur les routeurs BGP à partir d'un contrôleur SDN. L'idée de transposer ce fonctionnement dans un réseau ad hoc mobile nous semble une piste intéressante et peut permettre, dans le cadre des réseaux tactiques, de simplifier la gestion de ces réseaux autonomes.

Liste des Publications

Conférences Internationales

F. Grandhomme, G. Guette, A. Ksentini, and T. Plesse. *Comparing Inter-Domain Routing Protocol Assessment Tools for MANET.* In International Conference on Communications (ICC). IEEE, 2016.

F. Grandhomme, G. Guette, A. Ksentini, and T. Plesse. *Comparaison d'outils d'évaluation de performance des protocoles de routage inter-MANET.* dans Nouvelles Technologies de la Répartition (NOTERE). IEEE, 2016

F. Grandhomme, G. Guette, A. Ksentini, and T. Plesse. *ITMAN : an Inter Tactical Mobile Adhoc Network Routing Protocol.* In Military Communications Conference (MILCOM). IEEE, 2016

soumission en cours - F. Grandhomme, G. Guette, A. Ksentini, and T. Plesse. *Alternatives to Binary Routing Policies Applied to a Military MANET Coalition.*

Bibliographie

[AFKM05] C. Adams, S. Farrell, T. Kause, and T. Mononen. Internet x. 509 public key infrastructure certificate management protocol (cmp), 2005. RFC 4210.

[ALF03] A. Avudainayagam, W. Lou, and Y. Fang. Dear : a device and energy aware routing protocol for heterogeneous ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(2) :228–236, 2003.

[BBD+00] L. Bassham, W. Burr, Morris D., J. Foti, and E. Roback. Report on the development of the advanced encryption standard (aes). *National Institute of Standards and Technology*, 2000.

[BCHK06] J. L. Burbank, P. F. Chimento, B. K. Haberman, and W. T. Kasch. Key challenges of military tactical networking and the elusive promise of MANET technology. *IEEE Communications Magazine*, 44(11) :39–45, 2006.

[BLMA05] S. Boudjit, A. Laouiti, P. Muhlethaler, and C. Adjih. Duplicate address detection and autoconfiguration in OLSR. In *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005.*, pages 403–410. IEEE, 2005.

[Blo70] B.H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7) :422–426, 1970.

[BM04] A. Broder and M. Mitzenmacher. Network applications of bloom filters : A survey. *Internet mathematics*, 1(4) :485–509, 2004.

[BTA+11] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut. Routing protocols in ad hoc networks : A survey. *Computer networks*, 55(13) :3032–3080, 2011.

[BWM98] S. Blake-Wilson and A. Menezes. Authenticated diffie-hellman key agreement protocols. In *Selected Areas in Cryptography*, volume 1556, pages 339–361. Springer, 1998.

[CCLW08] C.-K. Chau, J. Crowcroft, K.-W. Lee, and S. H.Y. Wong. Inter-domain routing for mobile ad hoc networks. In *3rd International Workshop on Mobility in the Evolving Internet Architecture*, MobiArch '08, pages 61–66. ACM, 2008.

[CDJH14] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg. The Optimized Link State Routing Protocol Version 2, 2014. RFC 7181.

[CFML08] R. Coltun, D. Ferguson, J. Moy, and A. Lindem. OSPF for IPv6, 2008. RFC 5340.

[CJ03] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr), 2003. RFC 3626.

[Dij71] E. W. Dijkstra. *A short introduction to the art of programming*, volume 4. Technische Hogeschool Eindhoven Eindhoven, 1971.

[DMRM02] S. K. Das, B. S. Manoj, and C. S. Ram Murthy. A dynamic core based multicast routing protocol for ad hoc wireless networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 24–35. ACM, 2002.

[GGKP16a] F. Grandhomme, G. Guette, A. Ksentini, and T. Plesse. Comparing Inter-Domain Routing Protocol Assessment Tools for MANET. In *International Conference on Communications (ICC)*. IEEE, 2016.

[GGKP16b] F. Grandhomme, G. Guette, A. Ksentini, and T. Plesse. Comparison of inter-manet routing protocol evaluation tools. In *New Technologies for Distributed Systems (NOTERE), 2016 13th International Conference on*, pages 1–6. IEEE, 2016.

[GGKP16c] F. Grandhomme, G. Guette, A. Ksentini, and T. Plesse. ITMAN : An inter tactical mobile ad hoc network routing protocol. In *MILCOM2016*, 2016.

[HB96] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS), 1996. RFC 1930.

[HB07] S. Hares and P. Bose. BGP dynamic AS renumbering, January 4 2007. US Patent App. 11/354,289.

[HPS02] Zygmunt J Haas, Marc R Pearlman, and Prince Samar. The zone routing protocol (zrp) for ad hoc networks. 2002.

[ISO84] IS ISO. 7498 : Information processing systems, open systems interconnection, basic reference model. *International Standards Organization, Geneva, Switzerland*, 1984.

[JJ01] J. G. Jetcheva and D. B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 33–44. ACM, 2001.

[KK00] B. Karp and H.-T. Kung. Gpsr : Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.

[KTRH11] M. Kaddoura, B. Trent, R. Ramanujan, and G. Hadynski. BGP-MX : Border gateway protocol with mobility extensions. In *Military Communications Conference (MILCOM)*, pages 687–692. IEEE, 2011.

[Lam80] Simon S Lam. A carrier sense multiple access protocol for local networks. *Computer Networks (1976)*, 4(1) :21 – 32, 1980.

[LWC+10] S.-H. Lee, S. H. Wong, C.-K. Chau, K.-W. Lee, J. Crowcroft, and M. Gerla. InterMR : Inter-MANET routing in heterogeneous networks. In *International Conference on Mobile Ad-Hoc and Sensor Systems*, pages 372–381. IEEE, 2010.

[MDMD01] A. Misra, S. Das, A. McAuley, and S. K. Das. Autoconfiguration, registration, and mobility management for pervasive computing. *IEEE Personal Communications*, 8(4) :24–31, 2001.

[Moy91] J. Moy. OSPF version 2, 1991. RFC 1247.

[MRM+09] P. Marques, R. Raszuk, D. McPherson, J. Mauch, B. Greene, and N. Sheth. Dissemination of flow specification rules, 2009. RFC 5575.

[NK85] R. Nelson and L. Kleinrock. Spatial tdma : A collision-free multihop channel access protocol. *IEEE Transactions on communications*, 33(9) :934–944, 1985.

[NP02] S. Nesargi and Ra. Prakash. Manetconf : Configuration of hosts in a mobile ad hoc network. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 1059–1068. IEEE, 2002.

[OKG14] I. Okundaye, T. Kunz, and S. Gulder. Inter-domain routing for tactical mobile ad-hoc networks. In *80th Vehicular Technology Conference (VTC Fall)*, pages 1–6. IEEE, 2014.

[OS09] R. Ogier and P Spagnolo. Mobile ad hoc network (manet) extension of ospf using connected dominating set (cds) flooding. Technical report, 2009.

[PBRD03] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing, 2003. RFC 3561.

[Res99] E. Rescorla. Diffie-hellman key agreement method. 1999.

[RL95] Y. Rekhter and T. Li. A border gateway protocol 4 (BGP-4), 1995. RFC 1771.

[RRS+99] S. Radhakrishnan, G. Racherla, C. N. Sekharan, N. SV Rao, and S. G. Batsell. Dst-a routing protocol for ad hoc networks using distributed spanning trees. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, volume 3, pages 1543–1547. IEEE, 1999.

[SD17] M. Saleh and L. Dong. Secure location-aided routing protocols with wi-fi direct for vehicular ad hoc networks. *arXiv preprint arXiv :1707.00654*, 2017.

[TT09] M. Tarique and K. E. Tepe. Minimum energy hierarchical dynamic source routing for mobile ad hoc networks. *Ad Hoc Networks*, 7(6) :1125–1135, 2009.

[ZCG09] B. Zhou, Z. Cao, and M. Gerla. Cluster-based inter-domain routing (CIDR) protocol for MANETs. In *Sixth International Conference on Wireless On-Demand Network Systems and Services*, pages 19–26. IEEE, 2009.