RÉPUBLIQUE ALGÉRIEN DÉMOCRATIQUE ET POPULAIRE
Ministére de l'Enseignement Supérieur et de la Recherche Scientifique

Université Hadj Lakhdar - Batna
Faculté des Sciences de l'Ingénieur
Département d'Informatique

Mémoire

Pour l'obtention du diplôme de **Magister**
Spécialité : SYSTÈMES INFORMATIQUES DE COMMUNICATION

Sous le titre

Adaptation de TCP aux réseaux sans fil

Présenté par Abderrezak BENYAHIA

Soutenus publiquement le 09/12/2012 devant le jury formé de :

Dr. BELATTAR Brahim (MC-A)	**Président**	*Université de Batna*
Pr. Azeddine BILAMI (Professeur)	**Raporteur**	*Université de Batna*
Dr. ZIDANI Abdelmajid (MC-A)	**Examinateur**	*Université de Batna*
Dr. MAMRI Ramdane (MC-A)	**Examinateur**	*Université de Constantine*

ملخص

طبيعة الحركة لعقد الشبكة الحرة (ad hoc) والوصلات اللاسلكية الغير مؤكدة تؤثر على استقبال البيانات بطريقة صحيحة.
برتوكولات النقل مثل بروتوكول مراقبة الانتقال (TCP) المستخدم لضمان انتقال معتمد في الشبكات السلكية حيث فقدان حزم المعومات يرجح أساسا إلى ازدحام في الشبكات قد يساء تفسيرها و تخلق رد فعل سيء للبروتوكول TCP.
لمعالجة هذه الظاهرة و ضمان اتصال موثوق أكثر، البروتوكول TCP يحاول أن يقدم أفضل الحلول، ولكنه يعامل جميع الخسائر بنفس الطريقة من خلال تذرع آلية التحكم في الازدحام. ففي حالة وجود نسبة عالية من الخسائر فإن أداء البروتوكول TCP في الشبكات الحرة يتدهور.
لتحسين أداء البروتوكول TCP فإن الهدف من هذه المذكرة هو: أولا تقديم الأعمال المتعلقة بالبروتوكول TCP في الشبكات الحرة لتحديد أوجه التقصير فيها و القضايا التي تحتاج إلى مزيد من التحسن أو لم تتم معالجتها حتى الآن.
أما ثانيا هو تقديم مقترح يتمثل في تغيير جديد للبرتوكول TCP يسمى TCP-MANet الذي يهدف إلى التمييز بين الخسائر الناجمة عن انكسار الروابط وظاهرة الازدحام من أجل تحسين أداء البروتوكول TCP.
هذا الحل يستخدم نفس الآلية التي تستخدم في TCP-BuS و لتحقيق استخدام أمثل للمخازن في العقد الوسيطة، والفصل بين طوابير الازدحام و الطوابير المخصصة لاحتواء حزم المعلومات بسبب الروابط السيئة التي ترسل فيما بعد عند استعادة الرابط.
أجريت دراسة على أداء الحل TCP-MANet مقارنة مع البروتوكول TCP التقليدي عن طريق برنامج محاكات الشبكات نس-٢.

الكلمات الرئيسية: TCP، الشبكة الحرة (ad hoc)، الازدحام، TCP-MANet، TCP-BuS, NS-2.

Résumé

La nature mobile des nœuds des réseaux ad hoc et les liens radios qui ne sont pas toujours fiables peuvent avoir un impact sur la bonne réception des données.
Les protocoles de transport tels que TCP prévus pour assurer une transmission fiable dans les réseaux filaires où la perte des paquets est principalement due à des congestions dans les réseaux, peuvent mal interpréter ces événements et engendrent une mauvaise réaction de TCP.
Pour remédier à ce phénomène et fiabiliser les communications, le protocole de transport TCP tente d'apporter les meilleures solutions, mais ce dernier traite toutes les pertes de la même manière en invoquant un mécanisme de contrôle de congestion. Avec un taux élevé des pertes, les performances du protocole TCP dans les réseaux ad hoc se dégradent.
Pour améliorer la performance de TCP, l'objectif de ce mémoire est en premier lieu de dresser un bilan sur les travaux ayant trait à TCP dans les réseaux ad hoc afin d'identifier leurs carences et les points qu'il faut encore améliorer ou ceux qui ne sont pas encore traités.
Dans un second temps, on a proposé une nouvelle variante pour TCP qu'on baptise TCP-MANet dont l'objectif est de différencier entre les pertes dues à la rupture des liens et le phénomène de congestion dans le souci est d'optimiser les performances de TCP.
Cette solution utilise le même mécanisme que TCP-BuS en optimisant l'utilisation des tampons au niveau des nœuds intermédiaires et la séparation entre files liées à la congestion et celles réservées pour contenir les paquets des liens erronés qu'on devra acheminer une fois les liens seront rétablis.
L'étude de performance de la solution TCP-MANet par rapport à TCP traditionnel a été faite par simulation sous ns-2.

Mot clés : TCP, ad hoc, congestion, TCP-MANet, TCP-BuS, ns-2.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Avec l'émergence de la technologie mobile, le développement de réseaux sans fil connait un véritable essor. L'installation de réseaux avec une infrastructure nécessite un cout plus élevé que la catégorie des réseaux ad hoc. Ces réseaux sont constitués d'appareils mobiles autonomes qui peuvent communiquer de façon fiable, et à faible coût nécessitant une installation facile. Le grand degré de liberté et les capacités d'auto-organisation ont rendu les réseaux mobiles ad hoc complètement différents des autres classes de réseaux. Pour la première fois, les utilisateurs ont la possibilité d'installer facilement leurs propres réseaux à moindre coût. Cependant, ce gain de coût a engendré de nouvelles solutions technologiques complexes, qui sont nécessaires à toutes les couches et aussi à travers plusieurs couches.
Pour toutes ces raisons, le domaine des réseaux mobiles ad hoc est l'un des secteurs des réseaux sans fil le plus innovant et stimulant, et cette technologie promet de devenir de plus en plus présente dans la vie de chacun de nous. Les réseaux ad hoc sont une étape clé dans l'évolution des réseaux sans fil, ils héritent des problèmes traditionnels de la communication sans fil et mobile, telles que l'optimisation de la bande passante, le contrôle de l'énergie et l'amélioration de la qualité de transmission. En outre, la nature multi saut et l'absence de l'infrastructure entraine de nouveaux problèmes de recherches tels que la configuration du réseau, la détection des périphériques et le maintien de la topologie, ainsi que l'adressage ad hoc et l'auto-routage.
Les utilisateurs des réseaux ad hoc souhaitent exécuter pendant leurs déplacements des applications populaires telles que ftp, Telnet, http, ...etc., sur des liaisons sans fil. La plupart de ces applications utilisent le protocole TCP au niveau de la couche transport, cependant des travaux de recherche ont montré que ce protocole ne peut pas être directement appliqué aux réseaux sans fil en raison de la mobilité des nœuds et l'incertitude des liens qui varient dans le temps. Ces caractéristiques produisent des erreurs de transmission fréquentes, ce qui interrompe les connexions TCP. Si un expéditeur TCP ne reçoit pas des acquittements du récepteur d'une manière régulière, les événements d'expiration du délai d'attente de transmission des segments se déclenchent, et dans ce cas TCP interprète ces événements et il les considère comme congestion du réseau, en conséquence il exécute un contrôle de congestion dans des situations inappropriées.
Notre contribution dans ce mémoire porte sur le protocole TCP et spécialement l'étude des problèmes liés à ce protocole dans les réseaux ad hoc avec une proposition d'une nouvelle solution baptisée TCP-MANet. Cette nouvelle solution se base sur l'utilisation des informations des couches inferieures tel que la puissance de transmission du signal et la détection des liens erronés dans le but d'améliorer les performances du protocole TCP dans les réseaux ad hoc.
Cette solution vise à distinguer entre les erreurs de transmission causées par les liens erronées et ceux dues à une congestion du réseau. Pour le faire, TCP-MANet utilise l'interaction entre les couches liaison, réseau et transport en se basant sur un mécanisme de notification via un message spécifique.

Ce mémoire est structuré en cinq chapitres. Les réseaux ad hoc et les différents problèmes liés (rencontrés) pour cette catégorie sont décrits dans le premier chapitre. Le second chapitre est consacré aux problèmes liés au protocole de contrôle de transport TCP dans les réseaux ad hoc. Le troisième chapitre est consacré à la description de la solution proposée dans ce mémoire. Le quatrième chapitre décrit une variété de simulateurs utilisés dans le monde des réseaux informatiques avec une intention particulière pour le simulateur NS-2 sur lequel on a réalisé nos simulations. La validation des différents résultats a fait l'objet du cinquième chapitre. Finalement, nous terminons par une conclusion sur le travail présenté dans ce mémoire ainsi que les perspectives et les orientations pour la poursuite de ce travail de recherches.

Chapitre 1

Les réseaux sans fil

1.1 Introduction

Le domaine sans fil a connu une croissance exponentielle durant cette dernière décennie. On constate un grand progrès dans les infrastructures réseau, la disponibilité des applications sans fil et l'émergence des appareils sans fil tel que les ordinateurs portables ou les ordinateurs de poches, les PDA et les téléphones cellulaires. Ces appareils jouent un rôle important dans notre vie. Ils ne sont pas seulement plus petits, moins chers, plus pratiques et plus puissants, mais ils exécutent aussi plusieurs applications et services réseau. Avec la myriade d'applications et de services gérés par les appareils mobiles, les services réseau affichent aussi une demande croissante. Actuellement la majorité des connexions entre les appareils sans fil est basée sur les fournisseurs de service fixes reposant sur des infrastructures ou des réseaux privés. Bien que les infrastructures des réseaux offrent une excellente façon aux appareils mobiles pour obtenir des services réseau, l'installation des infrastructures du réseau nécessite beaucoup de temps et un coût plus élevé. Parfois et dans pas mal de situations, l'utilisateur sollicite une infrastructure qui n'est pas disponible ou elle ne peut pas être installée. Pour fournir une connectivité ou des services réseau dans ces situations, il est nécessaire d'utiliser un réseau ad hoc.
L'émergence de la technologie sans fil et les difficultés liées à la mise en place des infrastructures réseau ont suscité l'attention des chercheurs à développer des réseaux qu'on peut facilement déployer sans aucune infrastructure. Cette connexion d'appareils mobiles entre eux n'est autre qu'un réseau ad hoc mobile qui est à la fois souple et puissant. De cette façon les nœuds mobiles peuvent non seulement communiquer les uns avec les autres, mais aussi recevoir des services Internet à travers un nœud passerelle Internet dans les zones sans infrastructure. Comme le réseau sans fil continue à évoluer, cette capacité ad hoc va devenir plus importante et les solutions technologiques utilisées pour supporter ces réseaux plus critiques, entrainant un grand nombre de projets de recherche et de développement dans les milieux de l'industrie et universitaire[1].
Dans le reste de ce chapitre, on présentera les éléments fondamentaux des réseaux sans fil, puis on décrira les réseaux ad hoc tout en présentant leurs défis et problèmes.

1.2 Les éléments fondamentaux des réseaux sans fil

La communication entre une variété d'appareils nous offrira la possibilité de fournir une variété de services. Bien que cette communication (entre appareils) est un mécanisme très puissant, il est aussi compliqué et maladroit, il mène à une complexité dans les systèmes actuels. Ceci ne rend pas uniquement le réseau difficile mais limite aussi sa flexibilité. Il existe aujourd'hui plusieurs standards qui assurent l'homogénéité entre une variété d'appareils. Pour assurer cette homogénéité, chaque appareil doit supporter plusieurs standards qui leurs permet de communiquer avec d'autres appareils. Si on prend l'exemple d'un réseau au niveau d'un département universitaire. Pour relier les labos et les centres de calcul, on a besoin d'une longueur des câbles de quelques kilomètres par des conduits dans les mûrs, et les plafonds du département.
Durant ces dernières années, plusieurs standards et technologies de connectivité sans fil ont vu le jour. Ces technologies permettent aux utilisateurs de connecter une large gamme d'appareils de façon très simple et sans nécessité de câblage pour les relier. Ces technologies fournissent des connexions automatiques plus rapides sans créer de problèmes d'incompatibilité entre les appareils. Elles éliminent entièrement la nécessité d'achat de câble pour connecter les appareils individuels. De ce fait on crée une possibilité d'utiliser les données dans n'importe quel endroit où on se trouve. Les réseaux locaux sans fil (WLAN) ont enregistré un développement important ces dernières années. Actuellement, avec l'émergence des technologies sans fil, les WLAN sont de plus en plus utilisés dans les réseaux informatiques comme solutions alternatives plus puissantes et plus flexibles que les LAN filaires. Auparavant, la vitesse des WLAN a été limitée à 2Mbps, mais avec l'introduction de nouveaux standards, ces WLAN peuvent supporter plus de 11Mbps dans les domaines industriels, scientifiques et médicaux (ISM).
Plusieurs technologies et standards existent actuellement, et parmi eux on trouve : le Bluetooth, l'association infrarouge de données (IrDA), HomeRF, et les standards 802.11 de l'institut de l'ingénierie électronique et électrique (IEEE). Ces technologies sont concurrentes dans certains secteurs comme ils sont complémentaires dans d'autres. Du fait de cette diversité dans la technologie existante, la question qui se pose est qu'elle est la meilleure technologie, ou qu'elle est la solution qu'on doit choisir pour une application spécifique. Pour bien répondre à cette question, on essaye dans le paragraphe suivant d'exposer les points forts et les points faibles de chaque technologie ainsi que leurs domaines d'application. Le but est d'utiliser des technologies fondamentales par radio pour permettre des transmissions de données sans fil, et pour fournir un support permettant la construction des réseaux et la gestion de divers appareils au moyen de logiciels de haut niveau[2].

1.2.1 Le Bluetooth

Le Bluetooth est la technologie de lien sans fil micro-onde la plus rapide et la moins couteuse en terme d'énergie, il a été conçu pour connecter des téléphones mobiles, des ordinateurs portables, assistant numérique personnelle (PDA) et d'autres équipements portables. Le Bluetooth est différent de l'infrarouge, donc il ne nécessite pas une position de champ de vision des unités connectées. Cette technologie utilise les modifications des techniques existantes du réseau local sans fil (WLAN) mais elle prend en considération la petite taille et le moindre coût des unités. Chaque fois qu'un Bluetooth est activé, tous les dispositifs qui appartiennent a sa portée peuvent transférer immédiatement les informations de localisation (adresse IP,

MAC,...etc.) et établir des petits réseaux entres eux, sans aucune intervention de l'utilisateur.
Les caractéristiques de la technologie du Bluetooth sont :

- Il opère dans une bande passante de 2.56Ghz, qui est globalement disponible (aucune nécessité de licence).
- Il utilise la fréquence de spectre de diffusion d'hop (FHSS).
- Il peut supporter plus que huit dispositifs dans un petit réseau connu sous le nom « PICONet ».
- Omnidirectionnel, pas de transmission dans le champ de vision à travers des murs.
- Un rayon de portée variant de 10m jusqu'à 100m.
- Moins couteux.
- Il consomme une énergie d'1Mw.
- Un rayon de portée étendu avec une énergie amplifiée (100 mètres).

1.2.2 L'IrDA

L'IrDA est une organisation internationale qui crée et favorise les standards infrarouges d'interconnexion de données interopérable à moindre coût. L'IrDA est un ensemble de protocoles qui couvrent toutes les couches de transfert de données et en outre, possède quelque conception de gestion du réseau et d'interopérabilité. Les protocoles IrDA sont l'IrDA DATA qui véhicule les données délivrés et IrDA CONTROL qui envoie les informations de contrôle. Généralement l'IrDA est utilisée dans les technologies de connectivité sans fil ayant des appareils qui utilisent des câbles pour leurs connectivités. IrDA est une norme de transmission de données point-a-point d'angle étroit (cône 30°) conçu pour fonctionner dans une distance au-dessous de zéro à un mètre et à une vitesse de 9600bps à 16Mbps. Maintenant les adaptateurs inclus les mises à niveau traditionnelles aux ports séries et parallèles.
Les caractéristiques de l'IrDA sont :

- Un rayon de porté du contact jusqu'à au moins 1 mètre, et peut être étendue à 2 mètres.
- Une communication bidirectionnelle qui est à la base de toutes les spécifications.
- Une Transmission de données de 9600bps jusqu'au maximum de 4Mbps.
- Les paquets de données sont protégés en utilisant le bit de redondance cyclique (CRC).

1.2.3 HomeRF

HomeRF est un sous-ensemble de l'union international de télécommunication (ITU), il fonctionne principalement au développement d'une norme à fréquence radio (RF) moins couteuse pour la communication des données et des voix. Le groupe de travail HomeRF a développé aussi le protocole à accès sans fil partagé (SWAP). SWAP est une spécification industriel qui permet a des PCs, a des périphériques, a des téléphones sans fil et d'autres dispositifs de communiquer des données ou des voix sans l'utilisation de câble. SWAP est similaire au protocole CSMA/CA (Carrier Sense Multiple Access With Collision Avoidance) de la norme IEEE 802.11 mais avec une extension du trafic de la voix. Le SWAP peut fonctionner dans les réseaux ad hoc ou dans les réseaux avec infrastructure sous le control d'un point d'accès. Dans le réseau ad hoc, toutes les stations sont égales et le contrôle est distribué entre les stations. Dans un réseau avec infrastructure, il

est nécessaire d'installer le point d'accès pour coordonné le système, il fournit une passerelle (gateway) au réseau téléphonique public (PSTN). Les mures et les plafonds ne pose pas de problèmes dans son fonctionnement, aussi une certaine sécurité est fournit à travers l'utilisation d'un identifiant (ID) unique du réseau. Il est robuste est fiable et minimise l'impact des interférences radio.

1.2.4 Le standard 802.11 (WiFi)

Le WiFi est une technologie sans fil commune utilisée par les propriétaires de maison, les petites entreprises, et les services ISP (Internet Service Provider). Les dispositifs WiFi sont disponibles « aux présentoirs »dans les boutiques informatiques, et l'amélioration des dispositifs WiFi sont conçus pour l'utilisation par l'ISP.
Les avantages de WiFi sont :

- Ubiquitaire et fournisseur neutre : communication entre n'importe quel appareil WiFi quelque soit le fabricant.
- Coût abordable et bon marché.
- Hackable, beaucoup de « hacks »existent pour étendre la portée et les performances d'un réseau Wifi.

Mais coté inconvénients on trouve :

- Conçu pour les réseaux locaux (WLAN), pas pour les réseaux étendues (WWAN).
- Utilise le mécanisme CSMA. A un moment donné, une seule station sans fil peut communiquer (i.e. un utilisateur réseau peut monopoliser toutes les ressources du réseau).
- Des applications telles que la vidéoconférence, Voice-Over-Internet-Protocol (VOIP) et le multimédia peuvent altérer les performances du réseau.

1.2.5 Le standard 802.16 (WiMax)

Le WiMAX a d'abord été conçu pour desservir des réseaux pouvant couvrir une municipalité entière MAN « Metropolitan Area Network ». Le WiMAX se trouve à mi-chemin. Il doit à la fois tenir compte d'une importante transmission d'informations mais aussi distribuer ces informations de façon sécurisée à des clients indépendants, tout en comptabilisant les coûts associés à chacun.
Wimax est un acronyme pour Worldwide Interoperability for Microwave Access. Il a été créé pour permettre la convergence et l'interopérabilité entre deux standards de réseaux sans fils auparavant indépendants : Le HiperMAN, propose en Europe par L'ETSI (European Telecommunications Standards Institute) et le standard de transmission radio 802.16, validé en 2001 par l'organisme international de normalisation IEEE (institute of Electrical and Electronics Engineers).
Le WiMAX demeure toutefois une solution de communication à large bande. Il n'est pas nécessairement une solution de rechange pour le WiFi mais plutôt un complément. Il n'est souvent pas toujours désirable d'avoir des données personnelles qui se propagent inutilement sur de longues distances. Les solutions WiFi ont l'avantage de couvrir de courtes distances et de desservir uniquement des zones plus restreintes. Il en va de même pour les ondes à très courte portée, comme Bluetooth, souvent utilisées pour les communications

entre appareils et périphériques. Avec de multiples configurations possibles, chacun pourra adapter cette technologie à ses besoins.
Cette technologie vise donc à introduire une solution complémentaire au DSL (Digital Subscriber Line) et aux réseaux câblés d'une part, et à interconnecter des hotspots WiFi d'autre part. WiMAX est principalement fondé sur une topologie en étoile bien que la topologie maillée soit possible. La communication peut être réalisée en ligne de vue (LOS : Line Of Sight) ou non (NLOS). La dernière mouture du standard qui nous intéresse dans ce dossier est le standard IEEE 802.16e qui couvre les terminaux mobiles et définit des mécanismes évolués de gestion[3].
WiMAX réunit donc plusieurs standards, tous à des états d'avancement différents, qui sont autant d'axes de travail du groupe IEEE 802.16.
Le WiMAX se distingue également grâce au type de modulation des ondes qu'il utilise, l'OFDM « Orthogonal Frequency Division Multiplexing ». Cette modulation de type FDM « Frequency Division Multiplexing »agit sur les fréquences. Cette méthode sépare un signal en plusieurs sous-signaux indépendants (ondes porteuses). Ceci permet de rapprocher et d'augmenter le nombre d'ondes porteuses dans une fréquence sans avoir d'interférences entre elles. Cette modulation permet même le chevauchement des ondes porteuses. La quantité d'information pouvant être transmise est de beaucoup augmentée[4].

1.2.6 HotSpots

HotSpots sont des réseaux sans fil souvent gérer par des entreprises et des particuliers. Ils sont appelés « HotSpots »car ils fournissent une petite zone de couverture aux personnes pour se connecter à des réseaux communautaires et internet. Les endroits populaires pour les réseaux HotSpots sont des zones communales telles que les restaurants et les cafétérias.
HotSpots sont aussi des outils puissants pour soutenir le tourisme. Les visiteurs d'un HotSpots peuvent avoir des informations à propos de la communauté locale, y compris les événements à venir et même des présentations de travaux d'art et d'artisanat locale. La société BC du réseau sans fil de British Columbia, Canada, fournit un service pour la communauté du réseau sans fil HotSpot.

1.3 Limitation de la technologie sans fil

Le spectre radio sans fil est une ressource finie. Beaucoup de gens utilisent le spectre radio dans les réseaux sans fil ce qui augmente le phénomène d'interférences. Dans certains cas, on peut rencontrer des utilisateurs qui interférent avec nous. Il est important d'adopter une politique au début du déploiement du réseau pour travailler avec notre communauté pour la résolution des problèmes d'interférence. Les opérateurs de réseau devraient s'informer mutuellement lors de la mise en place d'un nouveau système sans fil[5].

1.4 L'internet sans fil

L'internet sans fil est devenu possible grâce à l'évolution des ordinateurs portables et les connexions sans fil sur un réseau téléphonique mobile. Toutefois, la réalisation de l'environnement informatique mobile nécessite une architecture de communication qui n'est pas seulement compatible avec l'architecture actuelle mais aussi prend en compte les caractéristiques spécifiques de mobilité et des liens sans fil.
Durant ces dernières années on a vue une augmentation de l'utilisation des systèmes Internet ainsi qu'une augmentation des communications mobiles. Aujourd'hui plusieurs services important aux utilisateurs se basent sur la technologie Internet. Si les technologies mobiles et internet se rapprochent, donc il serait très bénéfique aux utilisateurs d'économiser le coût des installations réseaux et de s'offrir des plateformes basées sur des services hautement flexibles. Mais pour gérer un réseau internet sans fil fiable, on doit étudier trois types de contraintes :

- L'environnement d'exploitation sans fil
- L'architecture Internet existante
- La limitation des terminaux

Les réseaux sans fil sont plus intéressants pour les raisons suivants :

- La mobilité
- Le temps d'installation réduit
- Une fiabilité accrue
- Economies de coûts à long terme

L'internet est une collection d'exécution collaborative des réseaux informatique qui couvre le monde entier. Elle est aussi une vaste collection de ressources : personnes, informations et multimédias. Le mot « Internet »décrit un certain nombre d'accords, d'arrangements et de connections. En fait, il s'agit d'un réseau de réseaux, plus précisément un réseau de réseaux locaux. Chaque réseau possède son propre domaine et il dispose de ressources et de capacités spécifiques. La figure 1.1 montre une simple connexion Internet.

L'internet offre une variété de services tels que les courriers électroniques, le chat, les communications vidéo et audio en temps réel et le transfert de fichiers. L'internet utilise un système de commutation de paquets pour le transfert de données. L'internet a été conçu pour être très robuste. Dans le cas où une section du réseau deviendra inutilisable, les paquets peuvent être envoyés sur une autre voie pour atteindre leurs destinations. Une partie importante du protocole IP est les normes d'adressage IP qui définissent des mécanismes pour fournir une adresse unique à chaque ordinateur connecté à Internet. Les utilisateurs d'Internet se connectent à un fournisseur de service Internet (ISP) via des modems ou des services de réseaux numériques intégrés (ISDN). L'ISP attribue à l'utilisateur une adresse IP et achemine les paquets entre les utilisateurs d'Internet.
Les caractéristiques des réseaux sans fil montrent que pour gérer le réseau Internet sans fil fiable, il est nécessaire d'examiner les sujets suivants :

- La vitesse des liaisons sans fil
- L'évolutivité
- La mobilité
- La puissance de batteries limitées
- Les déconnexions (volontaires et involontaires)

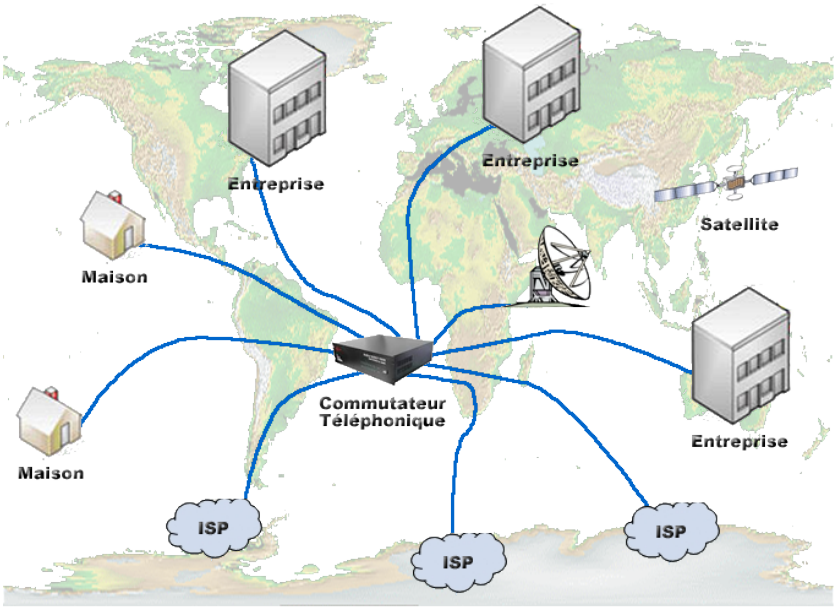

FIGURE 1.1 – Connexion Internet

- La mise en cache de la réplication
- La reprise

1.5 Les limitations de l'IP

Les limitations de l'IP sont dues à ses propres caractéristiques :

- Pour transmettre un paquet dans l'Internet, un ordinateur doit avoir une adresse IP.
- Cette adresse est associée à l'emplacement physique de l'ordinateur.
- Le protocole TCP/IP achemine les paquets vers leur destination en fonction de l'adresse IP.

Ces caractéristiques mènent à une grande limitation. En effet, dans TCP/IP, si l'utilisateur mobile se déplace sans changer l'adresse IP, le routage est perdu mais si l'utilisateur change d'adresse IP, les connexions seront perdues. Dans les deux cas, les paquets sont perdus, Ce qui mène à un réseau non fiable.
Concernant les caractéristiques spécifiques de mobilité et des liaisons sans fil, l'Internet sans fil doit offrir ce qui suit :

- Donner aux utilisateurs mobiles l'expérience Internet complète, et pas seulement un menu limité de services web spécialisés ou seulement la messagerie électronique.
- En effet, la téléphonie vocale devrait migrer vers l'Internet sans fil à n'importe quel moment.
- Etre raisonnablement rapide : au moins avec un débit de 100Mbps par utilisateur.
- Permettre aux utilisateurs fixes et mobiles de travailler à l'intérieur et à l'extérieur.
- Utiliser l'énergie d'une manière efficace, car la plupart des appareils fonctionnent avec des batteries limitées.
- Augmenter le niveau de couverture pour supporter des millions d'appareils actifs, dans une région métropolitaine.

1.6 Les réseaux Ad hoc

Un réseau ad hoc est une collection de nœuds mobiles sans fil qui forment dynamiquement une topologie temporaire sans utiliser une infrastructure ou une administration centralisée Les nœuds mobiles ou routeurs se déplacent librement de façon aléatoire et ils s'organisent arbitrairement ; ainsi, la topologie du réseau ad hoc peut changer rapidement et de manière imprévisible. Un tel réseau peut fonctionner dans un mode autonome ou peut être connecté à l'Internet. Le multi-saut, la mobilité, la grande taille du réseau combinée avec des dispositifs hétérogène, la bande passante et les contraintes de l'énergie rendent la conception des protocoles de routages adéquats pour les réseaux ad hoc est un défis majeur. Une certaine forme de protocole de routage est en générale nécessaire dans un tel environnement, parce que deux hôtes qui souhaiteraient échanger des paquets peuvent ne pas être en mesure de communiquer directement, comme le montre la figure 1.2.

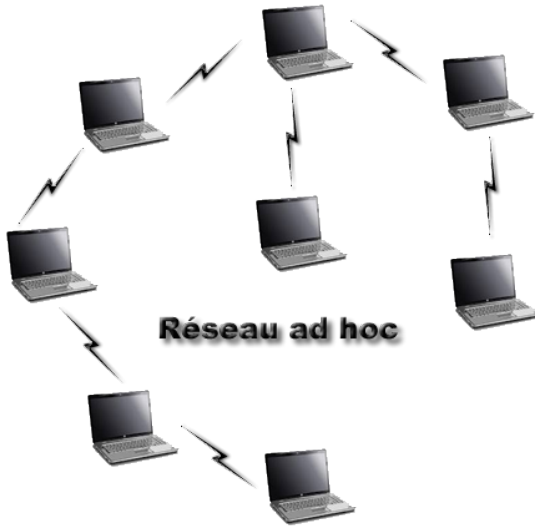

FIGURE 1.2 – Le réseau mobile ad hoc

Les utilisateurs mobiles souhaitent communiquer dans des situations où aucune infrastructure fixe filaire n'est disponible. A titre d'exemple, le cas des pompiers qui souhaitent se connecter à une ambulance en route vers une situation d'urgence et avec l'hôpital pour préparer les actions à entreprendre une fois le blessé ramené. Dans de telles situations, une collection d'hôtes mobiles avec une interface réseau sans fil peuvent former un réseau temporaire sans avoir besoin d'une infrastructure ou d'une administration centralisée, puisque les ordinateurs portables de nos jours sont équipés de processeur puissant, de grand volume de disque dure et une bonne qualité de son et d'image. L'idée de former un réseau entre les membres de l'équipe de sauvetage (Les pompiers) avec l'ambulance, et l'hôpital semble possible. Dernièrement, ces réseaux ont reçus une attention considérable dans les applications commerciales, dues aux propriétés intéressantes de la création d'un réseau naviguant et sans nécessiter une infrastructure telle qu'une station de base ou un contrôleur central[6].
Le groupe du réseau mobile ad hoc (MANet) à été formé au sein de l'IETF. Le principal objectif de ce groupe est de développer et d'évaluer les spécifications des MANets et les conformer aux standards Internet. L'objectif est de supporter les réseaux mobiles ad hoc avec un grand nombre de nœuds mobiles et de résoudre les défis rencontrés. Certains défis auxquels le réseau ad hoc est confronté sont : la portée de transmission

limitée, les problèmes du terminal caché et exposé, les pertes de paquets dus aux erreurs de transmission, les changements de chemin induites par la mobilité et les contraintes de l'énergie[7]

1.6.1 Les réseaux ad hoc et les réseaux cellulaires

Le tableau 1.1 donne les principales différences entre les réseaux ad hoc et les réseaux cellulaires

Les réseaux cellulaires	**Les réseaux ad hoc**
Des réseaux avec infrastructure	Des réseaux sans infrastructure
Les sites des stations de base fixe sont pré-localisés.	Pas de station de base
Topologie de réseau statique	Topologie de réseau très dynamique avec multi-sauts
Environnement relativement occupé et une connectivité stable	Environnement hostile (bruit, pertes) et une connectivité irrégulière
Une planification détaillée avant l'installation de la station de base	Un réseau ad hoc qui se forme automatiquement et s'adapte aux changements
Les coûts d'installation élevés	Moins coûteuse
Nécessite beaucoup de temps pour l'installation	rapidité d'installation

TABLE 1.1 – La différences entre les réseaux ad hoc et les réseaux cellulaires

1.6.2 Les applications des réseaux sans fil ad hoc

Les réseaux sans fil sont apparus avec l'intégration des ordinateurs personnels, de la technologie cellulaire et de l'Internet. Ceci est dû à la croissance des interactions entre la communication et l'informatique. Actuellement, une large variété de réseaux existe, allant des réseaux cellulaires bien connus avec infrastructure jusqu'aux réseaux sans fil ad hoc sans infrastructure. Dans ce qui suit, on donne les différentes applications de réseaux sans fil ad hoc :

- Les réseaux communautaires
- Les réseaux d'entreprises
- Les réseaux de maisons
- Les réseaux des interventions d'urgence
- Les réseaux de véhicules
- Les réseaux de capteurs

Contrairement aux réseaux sans fil avec infrastructure, les réseaux ad hoc ou les réseaux naviguant sont caractérisés par l'absence de l'infrastructure. Les nœuds dans un réseau mobile ad hoc se déplacent librement et s'auto-organisent d'une manière arbitraire. Chaque utilisateur communique librement avec les autres. Les chemins entre chaque paire d'utilisateurs peuvent avoir plusieurs liens, et les fréquences radio entre eux peuvent être hétérogènes, ceci permet une association de différents liens pour faire partie du même réseau. Le réseau mobile ad hoc peut fonctionner dans un mode autonome ou éventuellement être connecté à un réseau plus vaste tel que le réseau Internet.
Les réseaux ad hoc sont utilisés dans des situations où l'infrastructure est non disponible ou pour installer un réseau moins coûteux. L'un des nombreuses utilisations des réseaux ad hoc est dans certains milieux d'affaires où le besoin de l'informatique collaborative pourrait être plus important à l'extérieur qu'a l'intérieur du bureau, tel qu'une réunion d'affaires à l'extérieur avec des clients habituelles.

1.6.3 Les défis techniques dans la recherche

Les réseaux mobiles posent plusieurs défis techniques dans le domaine de la recherche. L'architecture ad hoc présente de nombreux avantages, tel que l'auto-reconfiguration et l'adaptabilité aux caractéristiques mobiles très variables, tels que les conditions de transmission, les contraintes d'énergie et les distributions du trafic. Ces avantages constituent de nouveaux défis qui résident principalement dans l'imprévisibilité de la topologie du réseau à cause de la mobilité des nœuds qui provoque une série de préoccupations dans la conception des systèmes de communication dans les réseaux sans fil ad hoc. Pour faire face à ce problème, plusieurs approches ont été proposées : le routage dynamique et la couche MAC distribuée, le protocole de localisation de service sans fil (WSLP), le protocole de configuration dynamique des hôtes sans fil (WDHCP), un contrôle distribué d'admission d'appel et la qualité de service (QoS) basée sur la technique de routage.

1.6.4 Les problèmes des réseaux sans fil ad hoc

Différents types de terminaux qui forment la plupart des réseaux ad hoc, par exemple, les appareils PDA-like, les téléphones mobiles, les téléavertisseurs bidirectionnelles, les capteurs ou les ordinateurs de bureau, chacun possède des capacités différentes en terme de puissance maximale de transmission, en terme de disponibilité de l'énergie, en terme mobilité et en terme de qualité de service. Les réseaux ad hoc sont en générale hétérogènes en termes de terminaux et en termes de services offerts. Si on prend par exemple l'énergie, on ne doit pas considérer seulement l'hétérogénéité des nœuds en termes de puissance de transmission et de disponibilité d'énergie, mais aussi une variété dans les portées de transmission (tel que les modes actifs ou en veille et l'existence de l'énergie supplémentaire). Les réseaux ad hoc soulèvent aussi des nouvelles questions concernant la sécurité et la confidentialité[8].
Les réseaux ad hoc héritent certains problèmes traditionnels des réseaux sans fil avec infrastructure :

– Le support sans fil n'a pas des frontières appropriées en dehors de laquelle, les nœuds peuvent savoir s'ils sont incapables de recevoir des trames.

– Le canal sans fil et faible, peu fiable, et non protégé contre les signaux étrangers, qui peuvent causer beaucoup de problèmes aux nœuds du réseau.
– Le canal sans fil possède une propagation ayant un temps variable et asymétrique.
– Les problèmes des nœuds cachés et exposés peuvent se produire.

Les problèmes du protocole de contrôle d'accès au media (MAC)

Les accès multiples au support sans fil peuvent être classés en accès aléatoire (ex. CSMA et CSMA avec détection de collision [CSMA/CD]), et les accès contrôlés (ex. TDMA et les méthodes basées sur le jeton). Les méthodes d'accès aléatoire sont les plus adapté aux réseaux ad hoc par l'absence de l'infrastructure. En outre, la norme IEEE 802.11 (WLAN) utilise la méthode CSMA/CA. La technologie du Bluetooth qui est conçus pour supporter des applications sensibles aux délais (ex. audio et vidéo) et plus de trafic de données, adopte la méthode TDMA avec un passage de jeton implicite pour l'attribution des accès. L'utilisation du Bluetooth et la norme IEEE 802.11 ne sont pas optimisés dans un environnement multi saut. Ces technologies sont utilisées pour des réseaux WPAN et WLAN à un seul saut. La conception des protocoles de la couche MAC pour un environnement ad hoc multi-sauts est un sujet de recherche[9].

Les problèmes relatifs au réseau

La plupart des fonctions principales des protocoles de réseaux sans fil ont besoin d'être réétudiés. Ces protocoles utilisent des services de transmission à un seul saut fournis par les technologies actuels pour assurer une transmission de bout en bout entre l'expéditeur et le récepteur. L'expéditeur doit localiser le récepteur dans le réseau en utilisant les services de localisation. Le but de ces services est de cartographier dynamiquement son emplacement dans le réseau. Les solutions adoptées pour gérer les terminaux mobiles dans les réseaux avec infrastructure sont insuffisantes, ce qui mène à trouver de nouvelles approches pour la gestion de la mobilité.
Une solution simple pour localiser les nœuds mobiles est basée sur l'inondation de l'emplacement de la requête à travers le réseau. Cette solution est conçue pour les réseaux de taille limitée. Le contrôle de la zone d'inondation peut aider à affiner cette solution, ceci peut être obtenu en augmentant progressivement le nombre de sauts impliqués dans la propagation des inondations, jusqu'à ce que le nœud soit localisé.
L'approche par inondation constitue un service de localisation réactif dans lequel aucune information de positionnement est maintenue à l'intérieure du réseau. Le coût de maintenance du service d'information de localisation est négligeable et toute la complexité est associée aux opérations de requêtes. D'autre part, les services de localisation proactive divisent la complexité en deux phases, la première phase construit et maintien dans le réseau des structures de données qui stockent les informations de localisation de chaque nœud et la deuxième phase exploite les structures de données pour simplifier les requêtes.

Le routage ad hoc

La nature dynamique du réseau mobile ad hoc cause des changements fréquents et imprévisibles de la topologie du réseau, en plus de la difficulté et la complexité du routage entre les nœuds mobiles. L'importance critique du protocole de routage pour établir des communications entre les nœuds mobiles ainsi que les défis et les complexités rencontrés dans ces réseaux ont rendu le problème de routage dans les réseaux ad hoc un domaine de recherche plus actif[10], [11].
De nombreux protocoles de routage ont été proposés ; leurs performances sous différentes conditions de trafic et d'environnements de réseau ont été étudiées. Plusieurs analyses comparatives des protocoles de routage des réseaux mobiles ad hoc ont été publiées. La classification des protocoles de routage peut être faite selon le type de la propriété de distribution des paquets et on trouve les protocoles de routage unicast, multicast, et/ou broadcast.

Les problèmes du protocole de contrôle de transmission

TCP est un protocole de contrôle de transport orienté connexion, efficace. Il fournit un contrôle de flux et un contrôle de congestion qui sont nécessaires pour assurer une fiabilité de livraison de paquets. Ce protocole a été conçu initialement pour fonctionner dans les réseaux fixes. Puisque les taux d'erreurs dans les réseaux fixes sont assez faibles, TCP utilise la perte des paquets comme une indication de congestion du réseau et traite cette congestion en ajustant sa fenêtre de congestion à un taux de transmission correspondant. L'environnement ad hoc mobile multi-saut apporte de nouveaux défis au protocole TCP à cause du changement fréquent de la topologie, des déconnexions, des variations des capacités de liaisons et du taux d'erreur élevé du support sans fil, ce qui entraine l'exécution d'un backoff de façon inapproprié[12], [13], [14], ceci réduit l'utilisation de la bande passante du réseau et augmente le délai de restauration de connexion[15]. En outre, la variation des capacités de liaison pourrait causer des liens asymétriques et des acquittements retardés, ce qui affecte ainsi l'ajustement de la fenêtre de congestion[16], [17]. En conséquence, les mécanismes TCP standard de contrôles de flux et de congestion ne fonctionnent pas correctement dans les réseaux ad hoc.
Face aux questions de la couche physique, un certain nombre d'études ont montrés que la couche MAC et le protocole de la couche réseau ayant un impact significatif sur les performances du protocole TCP. Puisque la fiabilité au niveau du lien est assurée par la couche MAC, le mécanisme de contrôle d'erreurs utilisé par la couche MAC peut affecter les performances du protocole TCP. La synchronisation bien définis est donc nécessaire entre le protocole TCP et les protocoles de la couche MAC afin de réduire l'effet de cette interférence sur les performances du protocole TCP[18].
Finalement, les différentes implémentations du TCP mènent à des résultats de performances différents. Par exemple les conflits entre les paquets de données TCP et les acquittements peuvent dégrader les performances du protocole lorsque la taille de la fenêtre est supérieur à un paquet[19]. En conséquence, le fonctionnement efficace du protocole TCP dans les MANET nécessite des adaptations spécifiques à différentes couches. De nombreuses solutions et optimisations ont été proposées au cours des ces dernières années pour améliorer les performances du protocole TCP. Parmi ces solutions, plusieurs d'entre elles ont été développées spécifiquement pour les environnements de réseaux sans fil cellulaires, où le dernier saut est basé sur un support sans

fil. Bien qu'il existe un certain nombre de différences entre les réseaux cellulaires et les réseaux ad hoc, plusieurs solutions proposées peuvent être utilisées dans les réseaux ad hoc mobile, alors que d'autres solutions peuvent être utilisées après une certaine adaptation.
Face à ces techniques, de nombreux nouveaux mécanismes d'optimisation du protocole TCP ont été proposées dans le but de résoudre les problèmes spécifiques aux MANET, y compris l'adaptation de détection des erreurs et les stratégies de rétablissements des chemins dans l'environnement ad hoc. Par exemple, les méthodes développées pour distinguer entre la perte des paquets causée par la congestion du réseau et la perte des paquets causée par des erreurs de transmissions[20], ce qui permettrait au TCP de prendre les mesures appropriés.

La sécurité du réseau

Sécuriser les réseaux sans fil ad hoc est une question très difficile, il y'a certains attaques spécifiques auxquelles le contexte ad hoc est vulnérable. Effectuer la communication dans l'espace libre expose les réseaux ad hoc aux écoutes et aux injections de messages. Les attaques du réseau ad hoc peuvent être classifiées en attaques actives et passives. L'attaque passive n'injecte pas n'importe quel message, mais elle écoute le canal. Une attaque passive tente de découvrir des informations précieuses et elle ne produit aucun nouveau trafic dans le réseau. Dans le cas d'une attaque active, les messages sont insérer dans le réseau. Ces attaques exécutent dans le réseau des actions telles que la réplication, la modification et la suppression des données échangées. Dans les réseaux ad hoc, les attaques actives sont l'usurpation de l'identité, le déni de service (DOS) et l'attaque de divulgation.

La qualité de service

La capacité du réseau à fournir une qualité de service qui dépend des caractéristiques intrinsèques de tous les composants du réseau, à partir des liens de transmission jusqu'a couche MAC et réseau. Les liaisons sans fil ont une faible capacité très variable et des taux de perte élevés. Les topologies sont très dynamiques. Les protocoles de la couche MAC basés sur l'accès aléatoire ne fournissent aucune qualité de service.
Les protocoles de qualité de service de la couche MAC résout les problèmes de contention moyenne, de maintient des communications unicast fiable et de réservation des ressources en temps réel dans un environnement sans fil distribué. Les nombreux améliorations et protocoles de la couche MAC qui ont été proposés peuvent fournir des garanties de qualité de service au trafic temps réel dans un environnement sans fil distribué comprennent le protocole Group Allocation Multiple Access with Piggyback Reservation (GAMA/PR) et le protocole Black Burst (BB).

1.7 Conclusion

Dans ce chapitre on a présenté les éléments fondamentaux de la technologie sans fil. Cette technologie est limitée à cause du media sans fil partagé entre les différents utilisateurs. Le développement de la technologie sans fil a crée de nouvelles recherches dans le domaine des réseaux. Afin que le réseau Internet profite des caractéristiques des appareils sans fil, les recherches ont donnés naissance à une nouvelle classe de réseaux sans fil sans infrastructure ou réseaux ad hocréseau !ad hoc. Les réseaux ad hoc peuvent s'organiser arbitrairement pour former un réseau temporaire sans aucune infrastructure préexistante. Ces réseaux posent plusieurs défis à cause de leurs caractéristiques (tel que la mobilité et l'énergie).
Aussi on a présenté les principaux problèmes des réseaux ad hoc et en particulier ceux associés au protocole de contrôle de transmission (TCP) le plus utilisé sur le réseau Internetréseau !Internet. Le chapitre suivant traitera les défis et les problèmes relatifs à ce protocole dans un réseau mobile ad hoc, comme il décrit aussi les améliorations proposées pour l'adapter dans les réseaux sans fil.

Chapitre 2

Protocoles de transport dans les réseaux ad hoc

2.1 Introduction

Le réseau ad-hoc est un système distribué complexe composé de nœuds sans fil ou filaires capable de s'auto-organiser dynamiquement. De cette manière ils forment temporairement des topologies de réseau aléatoire sans utiliser aucune infrastructure (point d'accès). Récemment, l'introduction de nouveaux protocoles tels que Bluetooth, IEEE 802.11, HyperLAN rendent possible le déploiement commerciale des réseaux ad-hoc.

Le protocole de contrôle de transmission (TCP) a été conçu pour fournir une transmission de données de bout en bout dans les réseaux non fiables. Théoriquement le protocole TCP devrait être indépendant de la technologie utilisée dans le support de transmission. En particulier, le protocole TCP ne devrait pas s'assurer si le protocole internet utilise un support de transmission câblé ou sans fil. Pratiquement la plus part des publications du protocole TCP ont été conçus soigneusement avec des hypothèses spécifiques aux réseaux câblés. Ces publications ne prennent pas en considération la dégradation des performances du protocole TCP dans les transmissions sans fil.

Le principal problème du protocole TCP dans le réseau ad hoc est qui ne fait pas de différence entre la perte du à la congestion, et celle causée par défaillance de liens. Les réseaux câblés sont caractérisés par un faible taux d'erreurs, d'où toutes les versions de TCP supposent que la perte de paquets est causée par la congestion. En conséquence quand le protocole TCP détecte une perte de paquet après expiration du temporisateur ou duplication multiple du paquet d'acquittement (ACK), il ralenti le taux d'émission en ajustant sa fenêtre de congestion. Malheureusement les réseaux sans fil soufrent de plusieurs types de pertes qui ne sont pas liées à la congestion, ce qui ne permet pas au protocole TCP de s'adapter à cet environnement. Les réseaux ad hoc héritent plusieurs caractéristiques telles que le taux d'erreurs et les chemins asymétriques élevés, ce qui pose de nouveaux problèmes dus à la mobilité et les communications multi hop, tels que les partitions du réseau, les échecs de routes et les terminaux cachés.

2.2 TCP traditionnel

Cette section est consacrée aux mécanismes du protocole de contrôle de transmission (TCP)[21] ayant une influence sur l'efficacité du TCP dans un environnement mobile tel que le contrôle de congestion, le démarrage lent, la retransmission rapide et/ou la découverte rapide (fast-restransmit/fast-recovery) et la mobilité.

2.2.1 Contrôle de congestion

Les protocoles de la couche transport tel que TCP sont conçus pour des réseaux câblés composés de nœuds fixes. La transmission de données se fait par le biais des adaptateurs réseaux, de la fibre optique, des câbles en cuivre, des équipements spéciaux pour les routeurs, etc. Ces équipements fonctionnent parfaitement sans erreurs de transmission. Pour des logiciels de la couche application est assez fiable, la perte de paquets lors de sa transmission n'est pas due aux matériels et/ou logiciels, mais la raison probable de la perte de paquet est due à la saturation temporaire au niveau d'un nœud intermédiaire qui fait partie du chemin de transmission, c.-à-d., un état de congestion dans un nœud intermédiaire.
La congestion peut apparaître de temps en temps dans les réseaux soigneusement conçus, les tampons réservés aux paquets au niveau routeur se trouvent remplis, et dans cette situation le routeur n'est pas en mesure d'expédier les paquets assez rapidement parce que le taux de paquets destinés pour une seule destination est plus élevé que la capacité de son tampon. La seule chose que peut faire un routeur est d'enlever des paquets du tampon et de les marquer comme perdus. Les paquets enlevés seront perdus, et le récepteur est notifié par cet événement quand il détecte un trou dans le flux de paquet transmis. Donc le récepteur n'informe pas l'expéditeur du paquet manquant, mais il continue d'acquitter la séquence des paquets jusqu'au paquet perdu. L'expéditeur observe l'absence de l'acquittement d'un paquet déjà transmit et suppose que le paquet est perdu à cause d'une congestion. La retransmission du paquet absent et le régime de transmission en plein taux se trouve ralenti, car ceci pourrait seulement augmenter la congestion. Bien qu'il ne soit pas garanti que tous les paquets de la connexion TCP prennent la même voie dans le réseau, cette hypothèse est valable pour tous les paquets. Pour réduire la congestion, TCP ralenti le taux de transmission de façon dynamique. Toutes les autres connexions éprouvant la même congestion font exactement la même chose. Cette coopération des connexions TCP sur internet est l'une des raisons principales de sa survie jusqu'à ce jour.
Les sections qui suivent introduisent les mécanismes du protocole TCP ayant une influence sur ce dernier dans l'environnement mobile[22].

2.2.2 Démarrage lent

La réaction du TCP à un acquittement manquant est tout à fait efficace, mais il est nécessaire de se débarrasser de la congestion rapidement. Le comportement du TCP déclenché après la détection de la congestion est appelé « démarrage lent »[23].
L'expéditeur toujours calcule la fenêtre de congestion pour un récepteur. La taille de démarrage de la fenêtre

de congestion est d'un segment (paquet TCP). L'expéditeur envoie un paquet et attend un acquittement. Si cet acquittement arrive, il incrémente la fenêtre de congestion par un autre segment. Puis il envoie deux paquet (fenêtre de congestion = 2). Après l'arrivé de deux acquittements correspondants, l'expéditeur ajoute une autre fois deux à la fenêtre de congestion (un pour chaque acquittement). Donc la fenêtre de congestion deviendra quatre. Ce schéma consiste à doubler la fenêtre congestion à chaque réception des acquittements, qui prennent un temps d'aller-retour (RTT). Ceci s'appel la croissance exponentielle de la fenêtre de congestion dans le mécanisme du démarrage lent.
Il est très déconseillé de doubler la fenêtre de congestion à chaque fois parce que cette dernière peut devenir très large ce qui entraine des temps d'attente avant retransmission très grand. La croissance s'arrête au seuil de la congestion. Dès que la fenêtre de congestion atteint le seuil de congestion, l'incrémentation ultérieure du taux de transmission est uniquement linéaire en ajoutant un à la fenêtre de congestion pour chaque arrivé d'un acquittement.
L'incrémentation linéaire continue jusqu'à l'expiration du temps au niveau de l'expéditeur à cause d'un acquittement manquant, ou jusqu'à la réception d'un acquittement dupliqué pour le même paquet. Dans les deux cas l'expéditeur place le seuil de congestion à la moitié de la fenêtre de congestion courante, puis il répète un démarrage lent mais la fenêtre se développe en mode linéaire.
Dans la section suivante, on va décrire deux autres mécanismes utilisés pour améliorer l'efficacité du protocole TCP.

2.2.3 Fast retransmit/fast recovery

Il existe deux cas qui causent une réduction du seuil de congestion. Un de ces cas est quand l'expéditeur reçoit d'une manière continue des acquittements du même paquet, ceci informe l'expéditeur que le récepteur a obtenu tous les paquets jusqu'au dernier paquet reçu dans la séquence. Dans TCP, un récepteur envoie des acquittements uniquement s'il reçoit des paquets de l'expéditeur. La réception des acquittements du récepteur prouve également que le récepteur a reçu d'une manière continue des choses de l'expéditeur. Un trou dans le flux de paquets n'est pas causé par une congestion grave, mais il est causé par une perte de paquet due à une erreur de transmission. Dans ce cas la, l'expéditeur retransmet le paquet manquant après l'expiration de l'horloge. Ce comportement est appelé « Fast Retransmit »[23].
La réception des acquittements prouve qu'il n y'a pas de congestion pour justifier un démarrage lent. L'expéditeur peut continuer avec la fenêtre de congestion courante. L'expéditeur invoque une découverte rapide « Fast Recovery »à partir du paquet perdu. Ce mécanisme peut améliorer l'efficacité du TCP.
L'autre cas de l'activation du démarrage lent est le délai dû à un acquittement manquant. En utilisant Fast retransmit/Fat recovery TCP interprète cette situation comme une congestion dans le réseau et active le mécanisme du démarrage lent.

2.2.4 La mobilité

Le démarrage lent du protocole TCP est l'un des mécanismes les plus utiles dans le réseau câblé, mais l'efficacité du protocole TCP se dégrade dans le cas des nœuds mobiles qui échangent des informations en

utilisant ce protocole. La raison de cette dégradation est l'utilisation du démarrage lent sous des hypothèses erronées. Le protocole TCP détecte une situation de congestion à partir des acquittements reçus. Les situations de congestion peuvent se produire dans des réseaux qui portent des nœuds mobiles, mais la perte de paquets n'est pas la raison principale de cette congestion.
Le taux d'erreurs dans les liens sans fil est plus grand par rapport aux liens câblé de fibre ou de cuivre. La perte de paquet ne peut pas être toujours compensée par les retransmissions de la couche liaison (ARQ) ou la correction d'erreurs (FEC). Essayer de retransmettre au niveau de la couche liaison pourrait, par exemple, déclencher la retransmission du paquet si cela prend beaucoup de temps. Ce qui met la couche liaison face à un problème de transmission du même paquet deux fois sur un mauvais lien. La détection de cette duplication dans la couche liaison est difficile, parce que plusieurs connexions utilisent le chiffrage de bout en bout, ce qui rend impossible d'observer le paquet dupliquer.
La mobilité elle-même peut causée des pertes de paquets. Il y'a plusieurs situations où le déplacement d'un point d'accès à un autre n'est pas possible pour les nœuds mobiles. Par exemple, en utilisant l'IP mobile, pendant le déplacement du nœud mobile d'un agent à un autre, il pourrait y avoir quelques paquets en transit dans l'ancien agent étranger. L'ancien agent étranger n'est pas capable de véhiculer ces paquets au nouvel agent ou même copier les paquets dans un tampon si la déconnexion du nœud mobile prenne beaucoup de temps. Cette perte de paquets n'est pas causée par les liens sans fil, mais elle est causée par les problèmes de routage.
Lorsque le mécanisme de contrôle de congestion du TCP détecte une absence d'acquittements après expiration du délai d'attente, il ne peut pas distinguer entre les différentes causes de la perte, donc il considère que cette dernière est la cause d'une congestion. Dans les deux cas les paquets sont perdus (que ce soit en raison de contrôle de bit de parité ou a une saturation du routeur). Cependant, les raisons sont complètement indépendantes. TCP ne distingue pas entre ces deux raisons. Les mécanismes de notification explicite de congestion (ECN) ont été discutés et quelques recommandations ont été déjà données[24]. Cependant, la RFC 3155[25] a publié que l'ECN ne peut pas être utilisé au tant que substitut pour la notification explicite d'erreur de transmission. Le TCP standard réagit avec un démarrage lent si des acquittements sont absents, ceci influe sur les performances du TCP dans le cas d'une erreur de transmission dans un lien sans fil et ce qui n'aide pas réellement pendant le déplacement. Ce comportement a comme conséquence une grave dégradation des performances du TCP original s'il est utilisé aussi dans les nœuds mobiles.
Cependant, on ne peut pas changer le protocole TCP juste pour supporter les utilisateurs mobiles ou les liens sans fil. Les mêmes arguments qui ont été donnée pour maintenir IP inchangé s'appliquent aussi au TCP. La base d'ordinateurs installés qui utilisent TCP est trop grande pour être changé et plusieurs mécanismes importants tel que le démarrage lent maintiennent Internet fonctionnel. Chaque amélioration au TCP doit, donc, demeurer compatible avec le TCP standard et ne doit pas compromettre le comportement prudent du TCP dans le cas de congestion.

2.3 Issues de conception du protocole TCP dans les réseaux ad hoc

Les performances du protocole TCP se dégradent dans les réseaux ad hoc, parce que le protocole TCP est face à un nouveau défi à cause de plusieurs raisons spécifiques à ces réseaux : La perte dans les canaux (lossy

channel), les stations cachées et exposées, les chemins asymétriques, les partitions de réseau, les échecs du chemin et les contraintes d'énergie.

2.3.1 Les défis

La perte dans les canaux

Les principales causes des erreurs dans les canaux sans fil sont :

L'atténuation du signal : c'est à cause d'une dégradation de l'intensité du signal électromagnétique au niveau du récepteur (peut être due à la distance), cette dégradation produit un rapport signal/bruit (SNR) plus faible.

Effet du doppler : c'est à cause des vitesses relatives de l'émetteur et du récepteur. L'effet du doppler cause des décalages de fréquence dans le signal d'arrivé, compliquant de ce fait la réception réussite du signal.

Effacement multi chemin : Les ondes électromagnétiques reflétant hors objets ou diffractant autour des objets peuvent résulter en signal voyageant sous des chemins multiples de l'émetteur au récepteur. La propagation des chemins multiples peut mener aux fluctuations dans l'amplitude, la phase et l'angle géographique du signal reçu par un récepteur.
Pour augmenter le succès des transmissions, les protocoles de la couche liaison implémentent les techniques ARQ (Automatic Repeate Request), FEC (Forward Error Correction), ou les deux ensembles. Par exemple, IEEE 802.11 implémente la technique ARQ. Donc quand un émetteur détecte une erreur, il retransmettra la trame, et la détection des erreurs est basée sur une horloge. Le Bluetooth implémente les deux techniques ARQ et FEC sur quelques connections synchrones et asynchrones. Notant que les paquets transmis sous un canal d'effacement peuvent être interpréter par le protocole de routage comme une détection d'un nouveau voisin. Ce voisin pourrait fournir un plus court chemin aux nœuds plus loin. Malheureusement, ce nouveau plus court chemin est habituellement incertain. Mais d'autres protocoles comme DSDV (Destination Sequence Distance Vector) et AODV (Ad hoc On-demande Distance Vector) dans un vrai réseau, constatent qu'aucune de ces techniques ne peuvent fournir un chemin multi hop stable en raison des comportements et spécialement de l'effacement physique du canal[2].

Les stations cachées et exposées

Dans les réseaux ad hoc, les stations peuvent s'appuyer sur un mécanisme de détection de porteuse physique afin qu'elles déterminent si le canal est inactif, tel que la norme IEEE 802.11 DCF (Distrubuted Coordination Function). Ce mécanisme de détection ne résout pas complètement les problèmes de la station cachée et la station exposée. Avant d'expliquer ce problème, on doit définir le terme « série de transmission ». La série de transmission est une série de paquets, qui concerne la station émettrice, où un paquet transmis peut être reçu avec succès.

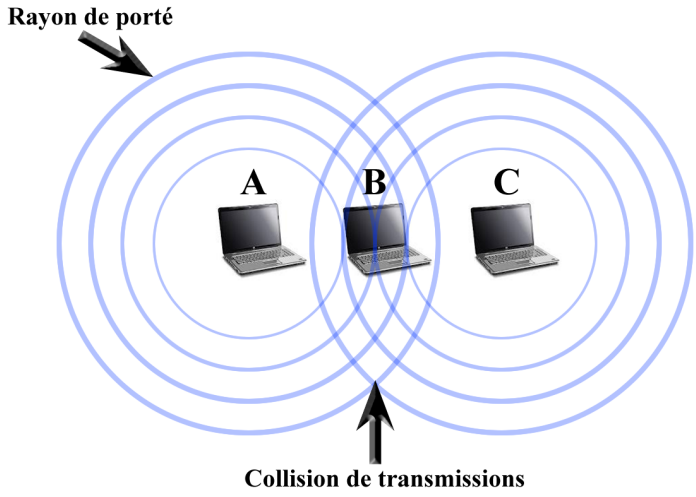

FIGURE 2.1 – Problème de la station cachée

Une situation particulière d'une station cachée est représentée dans la figure 2.1. Les stations A et C ont des trames à transmettre à la station B. La station A ne peut pas détecter les transmissions de la station C parce qu'elle est hors rayon de porté de la station C. La station A (respectivement C) est donc cachée pour la station A (respectivement C), puisque les rayons de porté de la station A et C sont disjoints, les paquets transmis à la station B peuvent provoquer des collisions. Ces collisions compliquent les transmissions des stations A et C vers la station B. pour alléger le problème des stations cachés, la détection virtuel de la porteuse a été introduite. Elle est basée sur un protocole de transfert bidirectionnel qui précède la transmission des données. Plus précisément, la station source transmit une petite trame de contrôle, appelée RTS (Request To Send), à la station destination. Après la réception de la trame RTS, la station destination répond par une trame CTS (Clear To Send), qui indique qu'elle est prête à recevoir les trames de données. Les deux trames RTS et CTS contiennent la durée totale de la transmission des données. Toutes les stations recevant soit la trame RTS ou CTS garde le silence pendant la durée de transmission des données (ex la station C dans la figure 2.2).

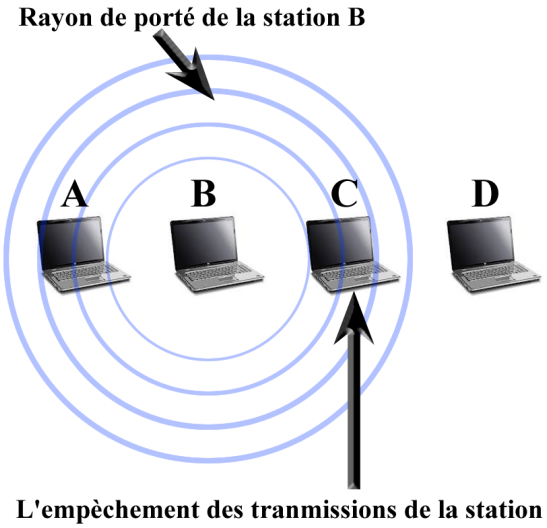

FIGURE 2.2 – Problème de la station exposée

Cependant, le problème de la station cachée peut persister dans le réseau ad hoc IEEE 802.11, même avec l'utilisation des trames de contrôle RTS-CTS. Cela est dû au fait que la puissance nécessaire pour interrompre une réception de paquets est beaucoup plus faible que la livraison d'un paquet avec succès. En d'autre terme la portée de transmission d'un nœud est plus petite que la portée de détection du nœud.
Le problème de la station exposée résulte à partir d'une situation où une transmission doit être retardée à cause des transmissions entre deux autres stations situées dans la portée de la station émettrice. La figure 2.2, présente un scénario typique où le problème de la station exposée survient. On suppose que les stations A et C sont dans le rayon de porté de la station B, et la station A est hors du rayon de porté de la station C. On suppose aussi que la station B elle est entrain de transmettre à la station A, et la station C a une trame à transmettre vers la station D. Selon le mécanisme de détection de porteuse, le lien de la station C devient un canal occupé en raison des transmissions de la station B. Par conséquent, la station C ne peut pas transmettre à D, même si cette transmission ne provoque pas d'interférence dans la station A. Le problème de la station exposé peut donc conduire à une réduction d'utilisation du lien.
Notant que les problèmes des stations cachées et les stations exposées sont liés à la distance de transmission. En augmentant le rayon de porté, le problème de la station cachée se produit fréquemment moins. D'autre part, le problème de la station exposée devient plus important puisque la portée de transmission identifie la zone touchée par une seule transmission.

Les chemins asymétriques

Les chemins asymétriques dans les réseaux ad hoc peuvent apparaître sous plusieurs formes comme la bande passante asymétrique, le taux d'erreurs asymétriques et le chemin asymétrique.

Bande passante asymétrique : Les réseaux satellite soufrent d'une large bande passante asymétrique, issue de divers compromis d'ingénierie (tel que l'énergie, la masse et le volume), la plus part des données proviennent du satellite. En générale, Le lien de retour n'est pas utilisé pour transférer les données. Par exemple, dans les réseaux de diffusion par satellite, le ratio de la bande passante du lien satellite-terre sur la bande passante du lien terre-satellite est d'environ 1000 kbps. D'autre part, dans les réseaux ad hoc, le degré de la bande passante asymétrique n'est pas très élevé. Par exemple, le ratio de la bande passante est compris entre 2 et 54 kbps dans les réseaux ad hoc qui implémentent le protocole IEEE 802.11g. L'asymétrie résulte de l'utilisation des taux de transmission différents. En raison de ces différentes vitesses de transmission, même les chemins source-destination symétriques peuvent souffrir de la bande passante asymétrique.

Taux d'erreur asymétrique : Ce type d'asymétrie apparait lorsque le chemin vers l'arrière (à la source) a un taux de perte plus élevé que le chemin vers l'avant (à la destination). Dans les réseaux ad hoc, cette asymétrie est due au fait que les pertes de paquets dépendent des contraintes locales qui peuvent varier d'un endroit à un autre. Notant que le taux de perte asymétrique peut produire une bande passante asymétrique. Par exemple, dans le protocole IEEE 802.11 version multi-cadence, les expéditeurs peuvent utiliser l'algorithme ARF (Auto Rate Fallback) pour la sélection du taux de transmission. Avec l'algorithme ARF, les expéditeurs tentent d'utiliser les taux de transmission plus élevés après les succès de transmission consécutifs, et de revenir à des taux plus bas, après des échecs. Donc, si le taux de perte augmente, l'expéditeur continue à

utiliser les taux de transmission faibles.

Chemin asymétrique : Contrairement aux deux formes d'asymétrie précédentes, où le chemin vers l'avant et le chemin vers l'arrière peut être le même. Le chemin asymétrique implique l'utilisation des chemins distincts pour les données TCP et leurs acquittements TCP. Cette asymétrie peut être un artefact du protocole de routage utilisé. Les chemins asymétriques augmentent la surcharge et la perte des paquets dans le cas de haut degré de mobilité, pendant le déplacement des nœuds. L'utilisation des chemins distincts (vers l'avant et vers l'arrière) augmentent la probabilité des chemins défaillants existants dans les connexions TCP. Cependant, ce n'est pas le cas avec les réseaux fixes ou des réseaux qui ont un faible degré de mobilité, comme le cas d'un réseau avec des chemins ayant une durée de vie élevée par rapport au temps de transfert de la session.

Le partitionnement du réseau

Un réseau ad hoc peut être représenté par un graphe simple G. les stations mobiles sont les « sommets ». Une transmission réussie entre deux stations est un « arc »non orienté. Le partitionnement du réseau se produit lorsque le réseau G est déconnecté. La raison principale de ces déconnexions dans les MANET est la mobilité des stations (nœuds). Un autre facteur peut conduire au partitionnement du réseau est la contrainte de l'énergie des nœuds. Un exemple de partitionnement du réseau est présenté dans la figure 2.3. Dans cette figure les lignes en pointillées sont les liens entre les nœuds. Quand le nœud « D »se déplace loin du nœud « C », le réseau sera partitionné en deux sous réseaux séparés. Evidement, l'agent étranger du nœud « A »ne peut pas recevoir des acquittements TCP transmises par le nœud « F ». Si cette déconnexion persiste pendant une durée supérieure à RTO (Retransmission TimeOut) du nœud « A », l'agent TCP exécute l'algorithme du backoff exponentiel, qui consiste à doubler le RTO quand le temps expire. A l'origine, le protocole TCP n'a pas d'indication sur le temps exacte de la nouvelle connexion du réseau. L'absence de cette indication peut mener à une longue période d'inactivité pendant la tentative d'une nouvelle connexion du réseau à cause du TCP qui est à l'état de backoff.

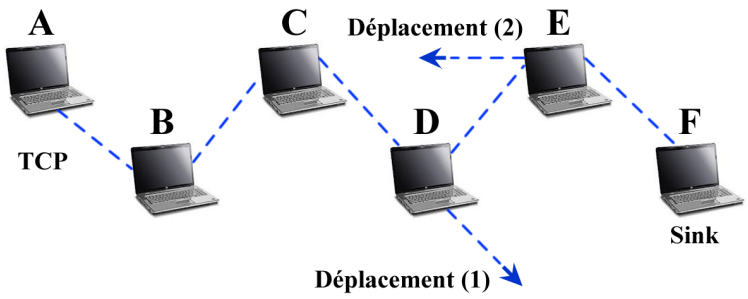

FIGURE 2.3 – Scénario de partitionnement du réseau

Les échecs du chemin

Dans les réseaux câblés, les échecs du chemin se produisent très rarement. Dans les MANET, ils sont des événements fréquents. La mobilité est la cause principale des échecs de chemin. Un autre facteur qui peut mener à ces échecs est les confusions dans le canal sans fil, ce qui dégrade les performances du protocole TCP dans les MANET. La durée de rétablissement du chemin après l'échec du chemin dans les réseaux ad hoc dépend du protocole de routage sous-jacent, du modèle de mobilité des nœuds mobiles, et des caractéristiques du trafic. Si un expéditeur TCP n'a pas d'indication sur l'événement de rétablissement du chemin, le débit et le délai de session se dégrade en raison du grands temps d'inactivité. Si le nouveau chemin établi est plus ou moins court en termes de saut que l'ancien chemin, le protocole TCP sera face à une fluctuation brutale dans le RTT (Round Trip Time).
En outre, les protocoles de routage des réseaux ad hoc qui s'appuient sur la diffusion du message « Hello »pour détecter l'accessibilité des voisins, peuvent souffrir du problème « des zones de communication grise ». Dans ces zones, les messages de données ne peuvent pas s'échanger, bien que, les messages « Hello »diffusés et les trames de contrôle indiquent que les voisins sont accessibles. Donc lors de l'envoie des messages de données, le protocole de routage indiquera des échecs de routage.

Les contraintes de l'énergie

Puisque les batteries associées à chaque nœud mobile ont une alimentation limitée, la puissance de traitement est limitée. Ceci est un problème majeur dans les réseaux ad hoc, parce que chaque nœud agit comme un système d'extrémité et un routeur au même temps, ceci implique que l'énergie supplémentaire est nécessaire pour acheminer et traiter les paquets. TCP doit pouvoir gérer cette ressource d'une manière efficace. L'efficacité signifie réduire le nombre de retransmissions inutiles dans la couche transport ainsi que dans la couche liaison. En générale, dans les réseaux ad hoc, il y a deux problèmes d'énergie en corrélation : le premier et « l'économie de l'énergie »qui vise à réduire la consommation de l'énergie, et le second c'est « le contrôle de l'énergie »qui vise à ajuster la puissance de transmission des nœuds mobiles. Les stratégies d'économie d'énergie ont été étudiées à plusieurs niveaux des dispositifs mobiles, incluant la couche physique, le système d'exploitation et les applications. Les stratégies de contrôle de l'énergie peuvent être utilisées conjointement dans les agents de transport et de routage pour améliorer les performances des réseaux ad hoc. Les communications de la contrainte de l'énergie révèlent également le problème de la coopération entre les nœuds, puisque les nœuds ne participent pas dans les procédures de routage et d'acheminement pour économiser la batterie.

2.3.2 Les objectifs de la conception

Lors la conception du protocole de la couche transport pour les réseaux ad hoc, les objectifs suivants doivent être remplis :

- Le débit d'une connexion doit être maximisé.
- L'équité du débit doit être fournit.

– Le temps d'installation de connexion doit être minimisé.
– La maintenance des saturations de connexion doit être minimisée.
– Le protocole doit incorporer un mécanisme de contrôle de flux et de congestion.
– Le protocole doit fournir un transport fiable ou non fiable.
– La bande passante doit être disponible d'une manière efficace.
– Le protocole doit prendre en considération les contraintes, et les ressources tel que l'énergie et la taille des tampons.
– Pour améliorer les performances, les informations de la couche inférieure doivent être utilisées d'une manière efficace.
– L'interaction entre les couches doit être efficace, extensible et indépendante.

2.4 Les performances du protocole TCP dans les réseaux mobiles ad hoc (MANET)

2.4.1 Les performances du protocole TCP

L'implémentation du protocole TCP dans un nœud mobile doit prendre en considération les facteurs suivants :

Délai de non congestion

L'un des problèmes principaux du TCP sous MANET est qu'il assigne toutes les pertes de paquet à la congestion, et que ce phénomène de perte est géré par le schéma de contrôle de cogestion. Quand une perte de paquet est détectée, la fenêtre de congestion est réduite, et l'horloge de retransmission est réinitialisée à un intervalle de backoff. Les algorithmes de contrôle de congestion doivent être utilisés seulement en cas d'une véritable congestion du réseau. Notant que VAN JACOBSON[26] suppose que la perte de paquets endommagés pendant le transport est rare ; d'où probablement les paquets sont perdus en raison de la congestion du réseau et non en raison de l'endommagement des paquets. Dans l'algorithme de VAN JACOBSON, on trouve que le schéma de contrôle de congestion est non sensible à la perte des paquets endommagés. Le taux de perte le plus élevée due à un paquet endommagé par fenêtre dégrade le débit du protocole TCP jusqu'à 60%.

Le délai périodique

Les déconnexions fréquentes causent une condition appelée « Le délai périodique »à l'émetteur TCP. Ceci se produit au doublement de l'horloge de retransmission pour chaque tentative de retransmission infructueuse, pour réduire le taux de transmission. Puis, à la reconnexion du nœud mobile, TCP prendra beaucoup de temps pour récupérer une telle réduction et les données ne seront pas transmises pendant cette période de temps.

La variation de la taille du paquet

Les liens sans fil supportent une taille des paquets généralement beaucoup plus petite par rapport à la taille des paquets dans des liens filaire. Donc, chaque paquet dans un réseau filaire est fragmenté lorsqu'il est transmis sous un lien sans fil. En conséquence, la récupération de la taille du paquet optimal dans des liens sans fil est une question clé pour les performances du réseau[27].

Le problème de collision des paquets de données et d'acquittements

Le mécanisme d'évitement de collision IEEE 802.11b élimine toutes les collisions. Puisque le trafic du protocole TCP est bidirectionnel (avec les paquets de données dans un sens et les paquets de d'acquittement dans l'autre sens), il peut y avoir une collision de paquets de données et les paquets d'acquittement. Ces collisions causent une retransmission au niveau de la couche MAC ou au niveau de la couche TCP lorsque la récupération d'erreur n'est pas utilisée dans la couche liaison. Ici, JACOBSON a testé le taux de retransmission des paquets UDP et TCP dans un environnement peut susceptible aux interférences. Dans UDP, les retransmissions sont relativement lent (presque 1%), mais quand il a utilisé TCP, les retransmissions ont augmenté à 5%. Il a qualifié cette augmentation aux collisions des paquets de données et d'acquittements. La réduction des performances n'est pas importante, mais les performances sont encore plus faibles si la récupération d'erreur n'est pas utilisée par la couche liaison. Le débit du TCP sans retransmission dans la couche MAC est inférieur à 23% que celui avec la retransmission dans la couche MAC. Les performances de l'UDP, même sans retransmission dans la couche MAC, sont un peu plus élevées que celles du TCP avec retransmission dans la couche MAC.

2.4.2 Autres problèmes

Les autres problèmes dans les couches inférieures sont décrits ci-dessous.

Propagation des chemins périmés

Le déplacement des nœuds mobiles cause un changement dans les chemins ce qui nécessite une mise à jour pour chaque changement. Même dans les changements de topologies lentes, l'émetteur TCP est lent à purger les chemins corrompus de son cache, ce qui entraîne des échecs répétés dans les chemins. Après ces échecs, les nœuds intermédiaires renvoient des chemins qui existent dans leurs caches répondant aux demandes de découvertes de nouveaux chemins, ce qui complique ce problème quand ils répondent parfois avec des chemins corrompus. Lorsque d'autres nœuds prennent en considération ces chemins corrompus inclus dans les réponses, la complexité du problème augmente ; en conséquence, les chemins corrompus se répartissent sur le réseau, en produisant plus d'échecs de chemins. Ce sont quelques faits qui ont un impact néfaste sur les performances du TCP. Ce problème peut être résolu en ajustant le délai du cache des chemins en fonction du taux d'échecs de chemins observé.

Le problème d'adaptation du taux dans la couche MAC

Ce problème concerne l'algorithme d'adaptation du taux dans la couche MAC. L'adaptation du taux dans la couche MAC est supposée comme une augmentation du débit quand le taux d'erreurs du canal est élevé. Un algorithme d'adaptation du taux faible pourrait diminuer le débit. L'algorithme d'adaptation du taux MIMD (Multiplicative Increase-Multiplicative Decrease) entraîne des retransmissions périodiques de paquets TCP. Ce mécanisme de « bande passante sondé »cause une défaite du réseau dans le WLAN. Le canal sans fil est gaspillé lorsque plusieurs retransmissions se produisent dans la couche MAC, et plusieurs retransmissions efficace ou inefficace se produisent dans la couche transport. Donc, un algorithme de meilleure adaptation du taux et un nouvel algorithme de « bande passante sondé »seront nécessaires.
La cause de dégradation des performances du TCP dans les MANET est due à quatre problèmes majeurs :

- TCP ne distingue pas entre les paquets perdus due aux échecs du chemin et la congestion du réseau.
- TCP soufre des échecs des chemins fréquents.
- Le conflit dans le canal sans fil.
- L'injustice du TCP.

2.5 Protocoles de transport Ad hoc

Cette section s'adresse aux différentes solutions proposées pour améliorer les performances du protocole TCP dans les réseaux sans fil. Ces solutions sont classées en deux approches :

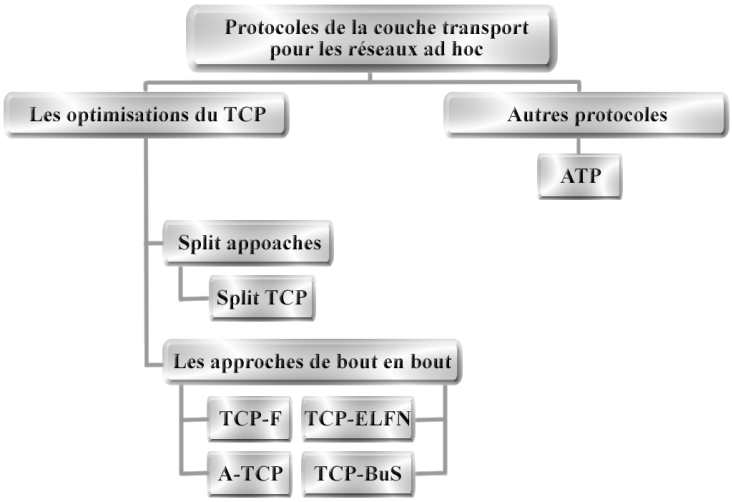

FIGURE 2.4 – Classification des protocoles de la couche transport

Split Approaches : destinées pour les réseaux ad hoc (sans infrastructure), elles reposent sur des acquittements entre les nœuds intermédiaires jusqu'à l'arrivée de la donnée au nœud source. Donc cette solution ne garde pas la sémantique de bout en bout.

Les approches de bout en bout : destinées pour les réseaux ad hoc tout en gardant la sémantique de bout en bout du protocole TCP.

2.5.1 Split approches

Cette section aborde le problème des échecs de chemins fréquents dans les MANET.

2.5.2 Split TCP

Les connexions TCP qui on un grand nombre de sauts soufrent des échecs de chemin fréquent due a la mobilité. Pour améliorer le débit de ces connexions et pour résoudre ce problème d'injustice, le schéma Split TCP[28] a été introduit pour diviser les longues connexions TCP à des petites segments (voire figure 2.5). Les nœuds interfaces situés entre deux segments sont appelés « Proxy ».

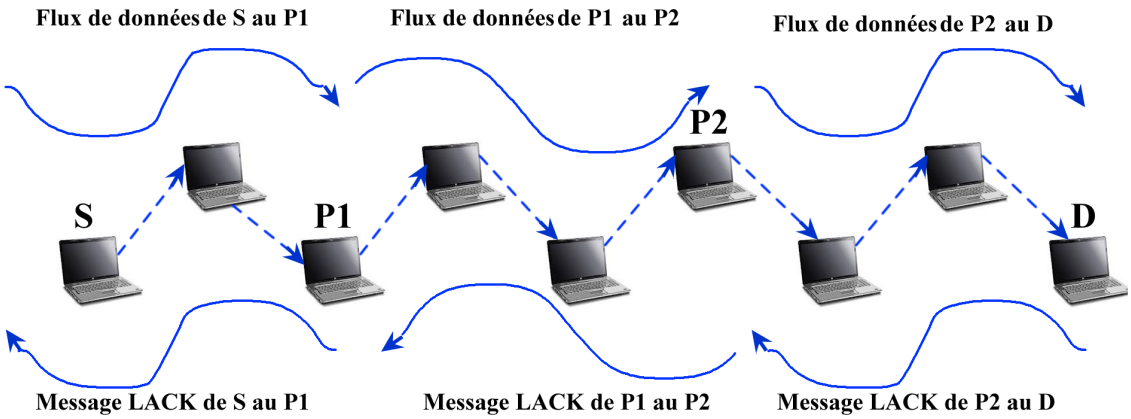

FIGURE 2.5 – Split TCP

L'agent de routage décide si son nœud a le rôle de proxy selon le paramètre de distance inter-proxy. Le proxy intercepte les paquets, et garde des copies de ces paquets et acquitte leur réception à la source (ou proxy précédent) en envoyant un acquittement local (LACK). En outre, un proxy est responsable de la livraison des paquets à un taux approprié pour le prochain segment local. A la réception d'un LACK (à partir du proxy suivant ou de la destination finale), un proxy purge le paquet de son tampon. Pour assurer une fiabilité de la source à la destination, un ACK similaire à la norme TCP est envoyé par la destination vers la source. En fait, ce schéma aussi découpe les fonctions de la couche transport à une fiabilité de bout en bout et un contrôle de congestion. Ceci est réalisé en utilisant deux fenêtres de transmission à la source qui sont la fenêtre de congestion et la fenêtre de bout en bout. La fenêtre de congestion est une sous fenêtre de la fenêtre de bout en bout. Le changement de la fenêtre de congestion est conforme au taux d'arrivée des LACK à partir du prochain proxy, le changement de la fenêtre de congestion de bout en bout est conforme au taux d'arrivée des ACK à partir de la destination. A chaque proxy, il y aurait une fenêtre de congestion qui régie le rythme d'envoi entre les proxys.

2.5.3 Les approches de bout en bout

Cette section aborde le problème de l'incapacité du protocole TCP de distinguer entre les pertes de paquets dues aux échecs des liens et celles dues à la congestion du réseau dans les MANET.

TCP Feedback (TCP-F)

Dans les réseaux mobiles ad hoc, la topologie subit des changements rapides en raison du mouvement des nœuds mobiles. Les changements fréquents de topologies entraînent des pertes de paquets et des expirations de délai. TCP interprète mal les pertes, et il considère ces pertes comme congestion du réseau et invoque un contrôle de congestion qui conduit à une retransmission inutile et une perte du débit. Pour surmonter ce problème, TCP-F (TCP Feedback)[29] à été proposé afin que l'expéditeur puisse distinguer entre l'échec du chemin et la congestion du réseau. Dans ce schéma l'expéditeur est obligé d'arrêter la transmission sans réduire la taille de la fenêtre en cas d'échec de chemin. Dés que la connexion est rétablie, la retransmission rapide est activée.

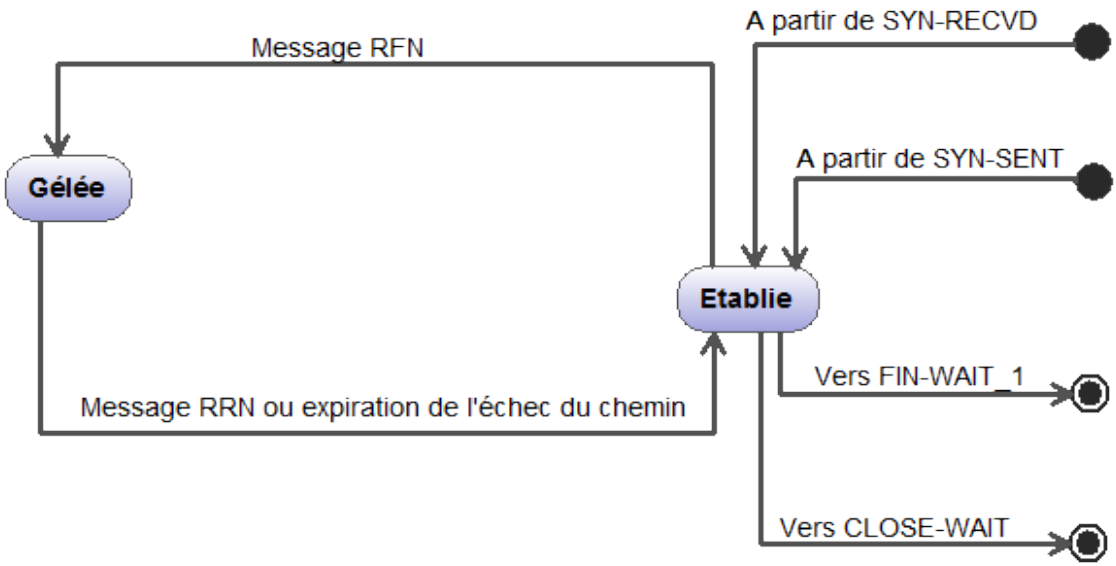

FIGURE 2.6 – Diagramme d'état du TCP-F

TCP-F repose sur la couche réseau d'un nœud intermédiaire pour détecter l'échec du chemin due à la mobilité de son nœud voisin en aval à travers le chemin. Un expéditeur peut être dans un « état actif »ou dans un « état répétition ». Dans l'état actif, la couche transport est contrôlée par le protocole TCP normale. Dès qu'un nœud intermédiaire détecte un chemin cassé, il envoie explicitement un paquet de notification d'échec du chemin (RFN) à l'expéditeur et enregistre cet événement. A la réception d'un RFN, l'expéditeur revient à l'état répétition, auquel l'expéditeur arrête complètement l'envoi de paquets supplémentaires et gèle toutes ses compteurs et les variables d'état tels que le RYO, et la fenêtre de congestion. En attendant, tous les nœuds intermédiaires en amant qui reçoivent le RFN invalident le chemin particulier pour éviter la perte de paquets supplémentaires. L'expéditeur reste à l'état répétition jusqu'à ce qu'il soit avisé de la restauration du

chemin par un paquet RRN (Route Reestablishment Notification) d'un nœud intermédiaire. Puis il reprend la transmission de l'état gelé. La figure 2.6 montre un diagramme d'état du TCP-F.

Une technique basée sur la notification explicite de l'échec du lien (ELFN)

La technique basée sur l'ELFN[15] est similaire à TCP-F. Contrairement à TCP-F, l'évaluation de la proposition est fondée sur une interaction réelle entre TCP et le protocole de routage. Cette interaction a pour but d'informer l'agent TCP sur les échecs des chemins quand ils se produisent. Les auteurs utilisent un message ELFN, qui est inclus dans le message d'échecs du chemin envoyé par le protocole de routage à l'expéditeur. Le message ELFN est comme un message ICMP (Internet Control Message Protocol) d'un « hot inaccessible », qui contient les adresses et les ports de l'expéditeur et du destinataire, ainsi que le numéro de séquence des paquets TCP. A la réception du message ELFN, la source répond en désactivant ses horloges de retransmission et entre en mode « veille ». Pendant la période d'attente, l'émetteur TCP teste le réseau pour vérifier si la route est rétablie. Si un acquittement du paquet de test est reçu, l'expéditeur TCP quitte le mode veille, reprend ses temporisateurs de retransmission et continue les opérations normales.
Un intervalle de deux secondes effectue un bon test et rend cet intervalle en fonction du RTT au lieu de lui donner une valeur fixe. Pour les valeurs du RTO et de la fenêtre de congestion (CW) pendant la restauration du chemin, en utilisant les valeurs préalables avant l'échec du chemin est plus performant que l'initialisation du CW pour un paquet ou RTO à six secondes.

Ad hoc TCP (ATCP)

ATCP[30], utilise aussi la réaction de la couche réseau. Pour le traitement des échecs dans les chemins, ATCP tente de résoudre le problème du BER (Bit Error Rate) élevé. L'émetteur TCP peut être placé à l'état persistance, à l'état de contrôle de congestion ou à l'état de retransmission. Le diagramme d'états-transitions de l'ATCP expéditeur est présenté dans la figure 2.7. La couche ATCP est insérée entre le TCP et la couche IP des nœuds source TCP. ATCP écoute les informations de l'état du réseau fournie par le message explicite de notification de congestion et par le message ICMP « des destinations inaccessibles », puis l'ATCP place un agent TCP dans l'état approprié. A la réception d'un message « destination inaccessible »l'agent TCP entre dans l'état persistance. L'agent TCP dans cet état est gelé, et aucun paquet n'est envoyé jusqu'à la découverte d'un nouveau chemin en testant le réseau. L'ECN est utilisé comme un mécanisme pour informer explicitement l'expéditeur à propos d'une congestion sur le chemin. A la réception de l'ECN, le contrôle de congestion du TCP est appelé normalement sans attendre un événement de délai expiré. Pour détecter des pertes de paquets dues aux erreurs du canal, ATCP surveille l'ACK reçu. Quand ATCP observe que trois ACK dupliqués ont été reçu, il n'achemine pas la troisième ACK dupliqué mais, il met le TCP dans l'état persistance et retransmet rapidement le paquet perdu à partir du tampon TCP. Après la réception du prochain ACK, ATCP reprend le TCP à l'état normal. Notons qu'ATCP permet l'interopérabilité avec les sources et les destinations qui n'implémentent pas l'ATCP. Dans les cas tels que la congestion, les liens perdus, le partitionnement et la réorganisation des paquets, le temps de transfert d'un fichier donné qui utilise ATCP a rapporté des meilleures performances que le TCP. En plus de l'échec du chemin, ATCP essaye de résoudre les problèmes du BER élevé, de la congestion du réseau et la réorganisation des paquets.

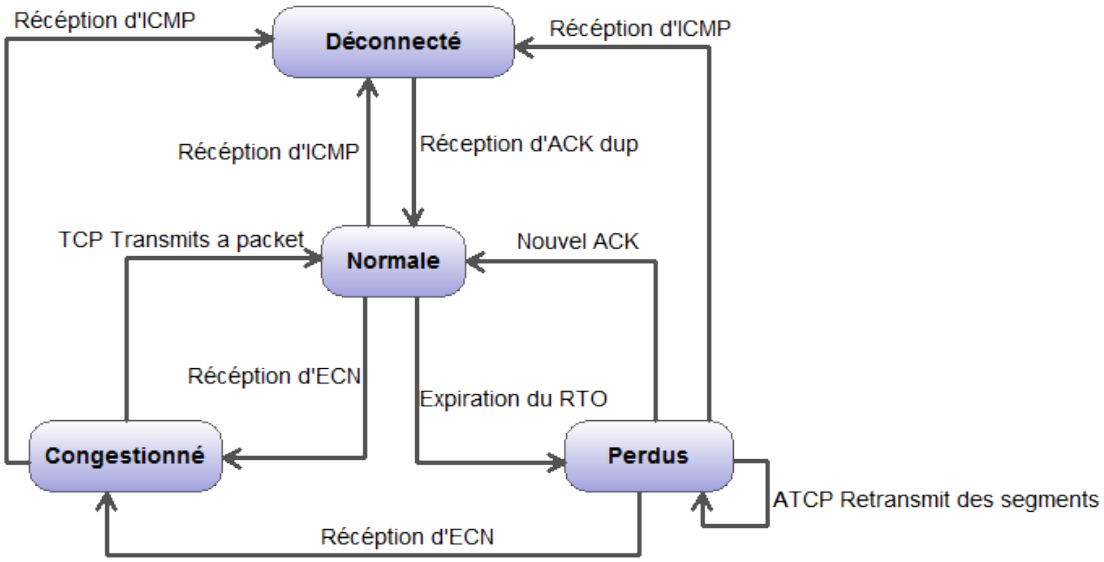

FIGURE 2.7 – Diagramme d'état du ATCP au niveau de l'expéditeur

TCP Buffering Capability and Sequencing Information (TCP-BuS)

TCP-BuS[31] utilise la réaction de la couche réseau pour détecter les événements d'échec du chemin et pour prendre des réactions convenables à cet événement. Il utilise le protocole de routage l'initiation de la source, à la demande, basé sur l'associativité (ABR). Les améliorations suivantes sont proposées :

La notification explicite : deux messages de contrôle sont utilisés pour informer la source à propos d'un échec du chemin et d'un rétablissement du chemin. Ces messages sont appelés ERDN (Explicit Rout Disconnetion Notification) et ERSN (Explicit Rout Successful Notification). A la réception d'un ERDN du nœud qui a détecté l'échec du chemin, appelé PN (Pivoting Node), le nœud source stop l'émission. Et parallèlement après le rétablissement du chemin par le PN en utilisant un LQ (Localized Query). A la réception de l'ERSN émis par le PN, la source reprend la transmission des données.

Les valeurs du délai étendu : Pendant la phase du RRC (Route ReConstruction), les paquets à travers le chemin de la source au PN sont copiés dans un tampon. Pour éviter les événements du délai expiré pendant la phase RRC, la valeur du temps de retransmission des paquets copiés est doublée.

Demande de retransmission sélective : comme pour la valeur du temps de retransmission est doublée, les paquets perdus à travers le chemin de la source au PN ne seront pas retransmis jusqu'à l'expiration de l'horloge de retransmission. Pour assumer ceci, une indication est faite à la source de sorte qu'elle puisse sélectionner la retransmission des paquets perdus.

Prévention des demandes inutiles pour une retransmission rapide : Quand le chemin est reconstitué, la destination informe la source à propos des paquets perdus à travers le chemin de la PN à la destination. A la réception de cette notification, la source retransmet simplement les paquets perdus. Mais les paquets copiés à travers le chemin de la source au PN peuvent arriver à la destination plutôt que les paquets retransmis. Pour éviter les demandes inutiles pour une retransmission rapide, la destination répond par un ACK dupliqué.

Comme les chemins ad hoc peuvent être invalidés par les mouvements des nœuds, on discutera les actions prises par TCP-Bus à la source, à la destination et aux nœuds intermédiaires pour faire face à la mobilité de l'hôte.

Les fonctions du TCP-BuS dans le nœud source : À la source, TCP-BuS transmet ses segments de la même manière que le TCP générale quand il n y'a aucun message de réaction (tels que des messages ERDN et ERSN). Les mécanismes de démarrage lent et d'évitement de congestion fonctionnent comme d'habitude. Cependant, quand la source reçoit un message de réaction ERDN du réseau elle cesse d'envoyer les paquets de données. En outre, elle gèle toutes les valeurs d'horloges et les tailles de fenêtre comme dans le TCP-F. Puis, elle extrait la valeur $ERDN\_GEN\_SEQ$ du message ERDN et elle calcule $ERDN\_RCV\_SEQ$. Comme exemple (voire la figure 2.8), Le nœud PN détecte un échec du chemin quand il a un segment (segment N° 10) à transmettre. Le PN génère un message ERDN qui contient le numéro de séquence (10) du segment dans l'entête de sa fille de transmission. Par conséquent, quand la source reçoit le message ERDN, le paramètre $ERDN\_GEN\_SEQ$ est ajusté à 10. Durant cette attente, la source avait envoyé des segments jusqu'au segment (14).

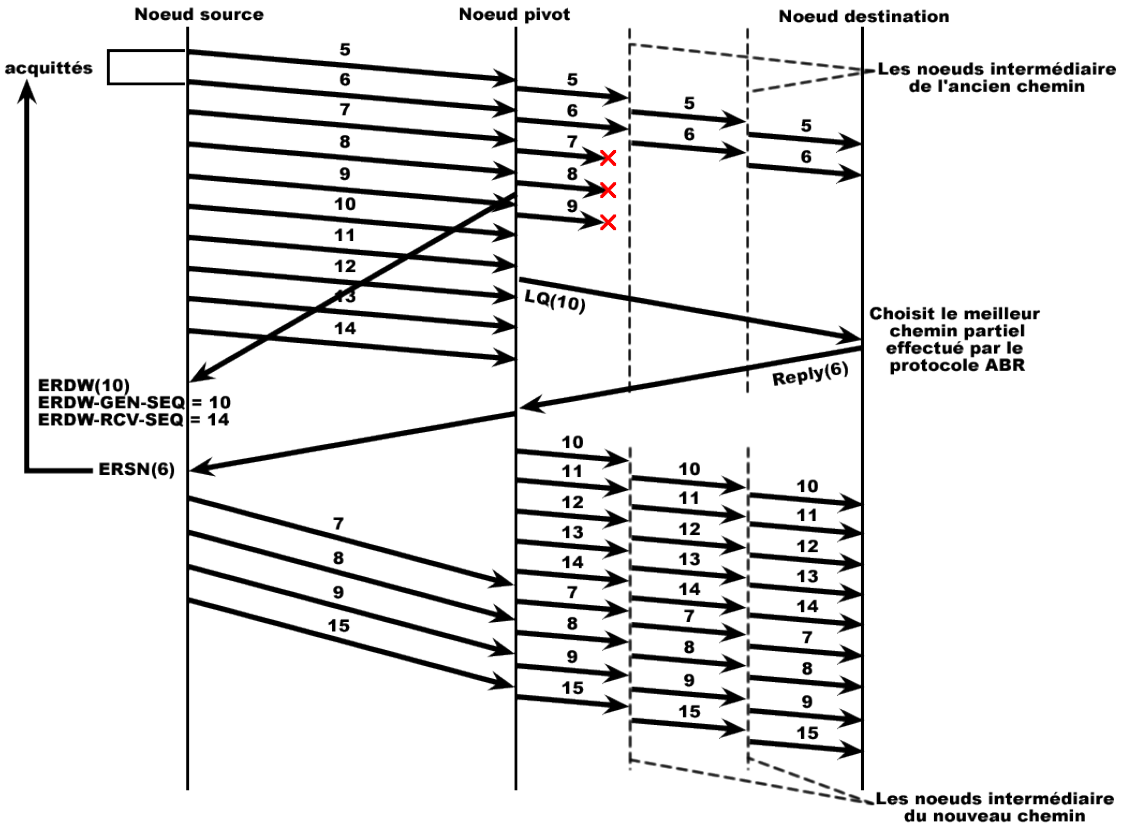

FIGURE 2.8 – Un exemple qui illustre le mécanisme de $ERDN\_GEN\_SEQ$ et $ERDN\_RCV\_SEQ$

A partir d'ici, on peut calculer $ERDN\_RCV\_SEQ$, qui et ajusté a quatorze (14). En plus, le prochain nœud descendant du PN enverra le message RN vers la destination, qui infirme l'ancienne partie du chemin et libère les paquets copiés dans les mémoires tampon des nœuds intermédiaire du chemin. Parce qu'un message ERDN indique qu'il y a un échec du chemin dans le réseau, la source attend juste un message ERSN.

A la réception du message ERSN, la source interprète ce message comme une réussite de rétablissement du chemin. La source peut alors reprendre la transmission des données selon le mécanisme de contrôle de congestion. A la réception d'un message ERSN après la reconstruction du chemin (c.-à-d., après que le nœud PN reçoit le message REPLY utilisée dans ABR, il génère le message ERSN avec le dernier paramètre ACK qui est extrait à partir du message REPLY), la source peut augmenter la fenêtre de congestion par la quantité de paquets acquittés. En même temps, la source peut supposer que les segments ayants les numéros de séquences entre ($Last\_ACK + 1$) et ($ERDN\_GEN\_SEQ - 1$) ont disparus dans le chemin qui commence du nœud PN jusqu'à la destination. Donc, la source devrait retransmettre ces paquets disparus. Dans la figure 2.8, parce que le message ERSN inclut le numéro de séquence du segment (6) reçu avec succès par la destination, la source se rend compte qu'elle devrait retransmettre les segments du numéro de séquence (7) au numéro de séquence (9), qui ont été disparus dans l'ancien chemin partiel. Cependant, la destination dépend de la situation de congestion dans le chemin vers le nœud PN. Le message ERSN peut inclure les informations de congestion informant l'état des files d'attente du routeur aux nœuds intermédiaires. On peut se servir des informations du message ICMP tel que l'adresse source à atteindre pour indiquer la présence d'une congestion, et la source devra arrêter les transmissions pendant une période de temps court d'où le nœud intermédiaire peut récupérer les paquets manquant. Notant qu'il faut ajuster le délai des segments non acquittés et non retransmis à partir d'ERDN_GEN_SEQ jusqu'à ERDN_RCV_SEQ en raison de l'augmentation prévue du temps d'arrivé du paquet à la destination due a la présence de rétablissement du chemin. Cependant, si la perte de paquets se répète plusieurs fois dans le chemin partiel de la source au PN due à la congestion, la source réagit à la congestion et sélectionne les paquets perdus à retransmettre à la réception du paquet de demande de retransmission sélective publié par le récepteur. Par conséquent, elle exécute une procédure de contrôle de congestion et réduit la taille de la fenêtre de congestion.
Les paramètres et les messages de contrôle sont les suivants :

$ERDN\_GEN\_SEQ$ **:** Quand un nœud intermédiaire détecte un échec du chemin et il ne peut pas acheminer les paquets de données du tampon, $ERDN\_GEN\_SEQ$ est défini comme un numéro de séquence du segment TCP en suspens dans la tête (HOL) de la file de transmission. L'information $ERDN\_GEN\_SEQ$ est propagée du nœud PN à la source via le message ERDN.

$ERDN\_RCV\_SEQ$ **:** Quand la source est entrain de transmettre des segments TCP, si la source reçoit un message ERDN du réseau, la source cesse d'envoyée les segments TCP. $ERDN\_RCV\_SEQ$ est défini comme le numéro de séquence du dernier segment TCP envoyé jusqu'à ce que la source TCP à reçu le paquet de contrôle ERDN.

$Last\_ACK$ **:** Pendant la reconstruction du chemin, la destination répond au message LQ avec un message REPLY. Donc $Last\_ACK$ est défini comme le numéro de séquence du dernier segment que la destination a reçu correctement.

En utilisant $ERDN\_GEN\_SEQ$ et $ERDN\_RCV\_SEQ$, mentionnés ci-dessus, on peut affirmer ce qui suit :

- Les segments non acquittés (copier à la source) jusqu'au segment dont le numéro de séquence est ($ERDN\_GEN\_SEQ - 1$) peuvent être acheminés à travers le chemin du nœud suivant au nœud PN vers la destination.
- Les segments non acquittés (copier à la source) dont les numéros de séquence à partir d'$ERDN\_GEN\_SEQ$ jusqu'à $ERDN\_RCV\_SEQ$ peuvent être copiés dans la mémoire tampon des nœuds intermédiaires.

Les fonctions du TCP-BuS dans le nœud intermédiaire : Après qu'un nœud PN détecte un échec du chemin, il envoie un message ERDN pour informer la source d'un échec du chemin et initie une découverte d'un chemin partiel en utilisant le processus LQ-REPLY. Tandis que le message ERDN est propagé vers la source, chaque nœud intermédiaire arrête encore la transmission des paquets de données et copie tous les paquets suspendus pour reporter leurs transmissions. Après la réception du message REPLY, le nœud PN informe la source d'une réussite de rétablissement du chemin via le message ERSN, qui inclut aussi l'information $Last\_ACK$. Chaque nœud intermédiaire qui reçoit ce message reprend la transmission des paquets copiés.

Les fonctions du TCP-BuS dans le nœud destination : Le récepteur exécute la procédure TCP de bout en bout normale dans le chemin construit au cas où il n'y aurait aucune déconnexion du chemin. Aussi, un mécanisme de retransmission sélective comme dans TCP-Selective ACK (SACK) peut être appliqué pour un contrôle de flux efficace. Une proposition additionnelle de schéma de retransmission sélective pour faire face aux paquets perdus dû à la congestion sur le chemin partiel de la source au récepteur. Une requête de la retransmission sélective des paquets perdus est générée au récepteur en détectant l'absence d'une séquence consécutive du segment. Ceci exige de la source pour réagir à la congestion. Etant donnée l'approche mentionnée ci-dessus, il est encore possible qu'il y'a beaucoup de requêtes de retransmission rapide dans le sens inverse. Considérant le cas où les segments ayant un numéro de séquence entre ($Last\_ACK + 1$) et ($ERDN\_GEN\_SEQ - 1$) arriveront plus tard à la destination que les paquets ayant un numéro de séquence entre $ERDN\_GEN\_SEQ$ et $ERDN\_RCV\_SEQ$. En conséquence, la destination continue à demander la transmission rapide en envoyant des paquets ACK dupliqués pour chaque paquet reçu avec des anomalies dans l'ordre de séquence. Pour éviter ce problème, une autre procédure additionnelle à la destination (voire la figure) est nécessaire après la réception du message LQ. Pour éviter les requêtes non nécessaires de la retransmission rapide, la destination envoi des ACK dupliqués pour sélectionner les paquets absents conformément à la règle suivante. Ici, on dénote le numéro de séquence du segment entrant comme « incoming_SEQ ». « $Pivot\_value$ »est le numéro de séquence dont la suite des segments est perdue due à la congestion. Donc, le récepteur informe la source du segment perdu d'une manière sélective. A la réception d'un message LQ pour une extension du chemin, $Pivot\_value = ERDN\_RCV\_SEQ$ si $incoming\_Seq = ERDN\_GEN\_SEQ$, alors la transmission des ACK dupliqués pour une retransmission rapide est privée. Si $incoming\_Seq > Pivot\_value$, ceci informe la source d'un segment absent. Sinon, la transmission des ACK dupliqués est autorisée.
Le TCP-BuS dépasse le TCP standard et le TCP-F sous différentes conditions. TCP-Bus n'a pas pris en compte que le nœud PN peut échouer dans l'établissement du chemin partiel à la destination.

2.5.4 Ad hoc transport protocol (ATP)

Le nouveau protocole de transport appelé ATP (Ad hoc Transport Protocol)[32] est destiné pour les réseaux ad hoc. Contrairement au protocole TCP, le protocole ATP est caractérisé par :

- des transmissions basées sur un taux,
- démarrage rapide pendant l'initiation de connexion et l'établissement de chemin,
- le réseau supporte la détection et le contrôle de congestion,

– pas de délai de retransmission,
– contrôle de congestion découplé et fiable,
– et la réaction du récepteur a grosse-gain.

Cette section trace les grandes lignes des éléments clé de conception du protocole ATP.

Coordination des couches

La coordination entre les différentes couches de la pile protocolaire est une nouvelle tendance dans l'adaptation des protocoles de transport pour les réseaux sans fil en générale et les réseaux ad hoc en particulier. Par exemple, la plus part des protocoles de routages conçu pour les réseaux ad hoc s'appuient sur des informations de la couche MAC pour détecter les échecs des liens (et aussi les chemins). Un degré supplémentaire de coordination possible dans les réseaux ad hoc et la coordination explicite entre les différents nœuds dans le réseau pour améliorer les performances de bout en bout. Comme dans le cas du TCP-ELFN qui utilise la notification d'un échec du lien dans un nœud intermédiaire pour geler la connexion TCP dans l'émetteur. ATP utilise les informations des couches inférieures et la réaction explicite des autres nœuds du réseau pour assister les mécanismes de la couche transport. Plus précisément, ATP utilise une réaction à partir des nœuds du réseau pour trois raisons différentes :

1. Taux initial de réaction pour l'estimation du taux de démarrage.
2. Taux progressif de réaction pour détecter, éviter, et contrôler la congestion.
3. Notification de l'échec du chemin. Bien que, n'importe qu'elle coordination de nœuds peut éventuellement contraindre l'évolution d'un protocole, ATP ne nécessite aucun état d'entretien par flux au niveau des nœuds intermédiaires, donc il est très évolutif.

Transmissions basées sur le taux

ATP utilise les transmissions basées sur le taux au lieu des transmissions basées sur la fenêtre exécutée par le protocole TCP. Les transmissions basées sur le taux aident dans l'amélioration des performances de deux façons :

– Elles évitent les inconvénients dus à la « sporadicité ».
– Puisque les transmissions sont ordonnancées par une horloge au niveau de l'émetteur, la synchronisation par l'arrivée des ACK est éliminée.

ATP utilise l'ordonnancement par une horloge pour séparer le mécanisme de contrôle de congestion du mécanisme de fiabilité et aussi pour affaiblir l'impact des caractéristiques du chemin inverse sur les performances éprouvées par les trains de bits véhiculés dans le chemin. Bien qu'une limitation évidente des schémas basés sur le taux est l'expiration de l'horloge au niveau de l'expéditeur. La granularité d'horloge nécessaire pour la bande passante dans le réseau ad hoc est assez grande pour être réalisé sans dépassement significatif. Par exemple, avec une charge raisonnable dans le réseau, 10 flux (ou 25 flux) ; une taille de paquet de 512 bits, et une capacité du canal supposée 2Mbps, la granularité d'horloge nécessaire est 40 millisecondes (ou 125 ms).

Découplage du contrôle de congestion et fiabilité

Différemment du TCP où le mécanisme de contrôle de congestion et le mécanisme de fiabilité sont étroitement accouplés par la dépendance d'arrivé du ACK, dans l'ATP les deux mécanismes sont découplés. Le premier mécanisme de contrôle de congestion est exécuté en utilisant la réaction du réseau, et le deuxième mécanisme de fiabilité est assuré par la réaction du récepteur et les ACK sélectifs.

- Pour faciliter le contrôle de congestion, les nœuds intermédiaires dans le réseau fournissent des informations de congestions en termes de taux disponible. La réaction est incluse dans les paquets de données du chemin vers la destination, et le récepteur ATP consolide une telle information et renvoie l'information de réaction assemblée.
- Pour une fiabilité, le récepteur utilise aussi des ACK sélectifs pour rapporter de nouveau à l'expéditeur tous les nouveaux trous observés dans le bloc de données. Différemment au TCP où l'information SACK est complémentaire au schéma cumulatif ACK, ATP se fonde seulement de l'information SACK.

Contrôle de congestion assisté

Le contrôle de congestion du protocole ATP est fondé sur la réaction au niveau des nœuds intermédiaires qui participent dans la connexion pour adapter le taux d'émission. Brièvement chaque nœud dans le réseau maintient deux paramètres :

- Q_t (une moyenne exponentielle du délai de la file éprouvée par les paquets qui ont traversés le nœud)
- T_t (une moyenne exponentielle du délai de transmission éprouvée par le paquet HOL au nœud)

Le T_t est influencé par la contention éprouvée entre les paquets dans les nœuds de la même proximité de contention, et Qt est influencé par la controverse entre les paquets appartenant à différents flux au même nœud. Pour chaque paquet qui traverse un nœud, le nœud estampe la somme « $Q_t + T_t$ »si la somme estampée déjà dans le paquet est inférieure à sa valeur courante. Le récepteur d'une connexion ATP exécute une moyenne exponentielle des valeurs estampées sur les paquets entrants. Pour chaque période de temps, le récepteur envoie le taux de réaction à l'expéditeur en utilisant la moyenne exponentielle. L'expéditeur se base sur son taux courant et le taux spécifié dans la réaction pour déterminer s'il incrémente, décrémente, ou maintien son taux. La phase de maintien est une différence critique selon les états d'une connexion ATP. En outre, les opérations d'incrémentation et de décrémentation exécutées par ATP sont plus précises en raison de la réaction du réseau reçu.

Convivialité du TCP et équité

La convivialité du TCP n'est pas une contrainte sous laquelle ATP est conçu, parce qu'il vise les environnements du réseau ad hoc où les nœuds du réseau posséderont une pile protocolaire dédiée pour ces réseaux. Cependant, l'équité des flux ATP est encore une clé qui concerne TCP, parce que l'ATP réagit avec la congestion du réseau en se basant sur les nœuds intermédiaire du réseau pour la réaction sur la congestion.

2.6 Conclusion

Dans ce chapitre, nous avons exposé les principaux défis auxquels un protocole de la couche transport est confronté dans les réseaux ad hoc. Les objectifs de conception majeurs ont été répertoriés et une classification des solutions existant de la couche transport a été fournie. TCP est le protocole de la couche transport le plus largement utilisé et il est considéré comme un point d'appui de l'internet aujourd'hui. Il fournit une fiabilité en ordre de livraison de paquets de bout en bout entre les nœuds source et destination. Comme le protocole TCP est conçu pour traiter les problèmes présents dans les réseaux filaires traditionnels, plusieurs questions ne sont pas prisent en compte dans les réseaux à topologie dynamique tel que les réseaux sans fil ad hoc. Ceci cause une réduction du débit quand on utilise TCP dans les réseaux sans fil ad hoc. Il est très important d'utiliser le protocole TCP dans les réseaux ad hoc puisqu'il est nécessaire de communiquer avec le réseau internet d'une manière transparente partout où il est disponible.
Un bilan non exhaustif est fait sur le nombre de solutions proposées récemment pour améliorer les performances du protocole TCP dans le monde sans fil et mobile. Le chapitre suivant sera consacré à la description d'une nouvelle solution pour améliorer les performances du protocole TCP dans les réseaux sans fil.

Chapitre 3

La solution TCP pour MANet

3.1 Introduction

Les améliorations du protocole TCP présentées dans le chapitre précédent ne répondent pas parfaitement aux contraintes imposées par l'environnement sans fil, et vu la nécessité d'utilisation du protocole TCP dans cette catégorie de réseau (i.e. réseaux sans fil ad hoc), les recherches portant sur l'amélioration du protocole TCP dans cet environnement ne cesse de croitre. La majorité des solutions (i.e. TCP-F, TCP-ELFN, ATCP et TCP-BuS) présentées précédemment pour les réseaux sans fil ad hoc sont basées sur les réactions avec les couches inferieurs de la pile protocolaire pour différencier entre la perte des paquets due à la congestion du réseau et la perte des paquets du à un lien erroné. Malgré la réponse des ces solutions au problème du contrôle de congestion dans les réseaux sans fil ad hoc, certaines d'entre elles ne répondent pas suffisamment au problème de contrôle de congestion du réseau et d'autres créent des conflits entre les communications du protocole TCP standard et les communications du protocole amélioré.
Dans ce chapitre, on présente une nouvelle proposition (solution) pour mieux adapter le protocole TCP dans les réseaux sans fil ad hoc. Cette solution se base aussi sur l'interaction des couches inférieures (Réseau et MAC) pour différencier entre les paquets perdus due à la congestion et les paquets perdus due à un lien erroné. Pour éviter les problèmes de conflit avec le protocole TCP standard, notre solution apporte des changements du coté expéditeur du protocole TCP et des changements majeurs dans l'agent de routage pour satisfaire le contrôle de congestion du réseau. En plus, cette solution utilise la puissance de transmission des paquets pour prédire la défaillance d'un lien et pour minimiser la perte des paquets due à cette coupure.
Les sections qui suivent présentent une description détaillée à l'aide de schémas et de digrammes UML de la solution proposée pour améliorer le protocole TCP dans les réseaux sans fil ad hoc.

3.2 Motivations

Les faiblesses (les insuffisances) du mécanisme de contrôle de congestion des solutions présentées précédemment nous a motivé à proposer une nouvelle solution pour améliorer les performances du protocole TCP dans l'environnement sans fil ad hoc. Dans la section suivante, on essaye de présenter les avantages et les inconvénients de chaque solution.

3.2.1 Split TCP (S-TCP)

Comme avantages à cette solution on trouve :
- L'amélioration du débit.
- L'amélioration de l'équité du débit.
- L'atténuation de l'impact de la mobilité.

L'amélioration du débit est due à la réduction dans l'effectif des chemins de transmission (nombre de saut dans une zone ou segment du chemin). Le débit se dégrade avec l'augmentation de la taille du chemin. S-TCP possède des petits segments de chemin concaténés, chacun opèrent sur son taux de transmission, d'où le débit est augmenté. Ceci mène aussi à une amélioration de l'équité du débit dans le système.
Les inconvénients de la solution S-TCP peuvent être listés comme suit :
- La nécessité de modifications majeures dans le protocole TCP.
- La sémantique de bout en bout de la connexion TCP traditionnel est violée.
- Les échecs dans les nœuds proxy peuvent mener à une dégradation du débit.

Le protocole TCP traditionnel possède la sémantique de bout en bout où les nœuds intermédiaires ne traitent pas les paquets TCP, tandis que dans la solution S-TCP, les nœuds intermédiaires doivent traiter les paquets TCP d'où certains mécanismes de sécurité qui nécessite le cryptage IP ne peuvent pas être utilisés. Durant les fréquentes coupures de chemins ou des échecs de liens, les performances de la solution S-TCP peuvent être affectées.

3.2.2 Le TCP basé sur la réaction (TCP-F)

TCP-F fournit une solution à base de réactions simples avec les couches inferieurs pour minimiser les problèmes produites par les ruptures fréquentes de chemin dans les réseaux sans fil ad hoc. Au même temps, il permet aussi au mécanisme de contrôle de congestion à répondre aux congestions du réseau. Cette solution dépond des capacités du nœud intermédiaire et du protocole de routage sous jacent pour rétablir rapidement un lien ou un chemin cassé. Ainsi, le point erroné doit être capable d'obtenir un chemin correct (le chemin traversé par le paquet) vers l'expéditeur TCP pour envoyer le paquet RFN. Ceci est simple avec un protocole de routage qui utilise le routage à la source (tel que le protocole DSR[33]). Si le chemin n'est pas disponible au niveau du point erroné, des paquets de contrôle additionnels peuvent être générés pour acheminer le paquet RFN. La solution TCP-F ajoute un état supplémentaire à l'automate d'états finis du protocole TCP standard, d'où son implémentation nécessite des modifications dans la bibliothèque TCP standard. Un autre inconvénient est que la fenêtre de congestion utilisée dans le nouveau chemin obtenu peut ne pas refléter le taux de transmission réalisable accepté dans le réseau et dans le récepteur TCP.

3.2.3 Le TCP avec une notification d'un lien erroné explicite (TCP-ELFN)

La solution TCP-ELFN améliore les performances du protocole TCP en découplant les informations du chemin cassé à partir des informations de congestion. Elle est peut dépendante du protocole de routage et nécessite uniquement une notification d'un lien erroné. Les inconvénients de cette solution sont :

– Au moment du partitionnement temporaire du réseau, la rupture du chemin peut durer beaucoup de temps et ceci peut mener à une génération périodique des paquets de test du chemin qui consomment la bande passante et l'énergie.
– La fenêtre de congestion utilisée après l'obtention d'un nouveau chemin peut ne pas refléter le taux de transmission réalisable accepté dans le réseau et dans le récepteur TCP.

3.2.4 Le TCP ad hoc (A-TCP)

Comme avantages de la solution A-TCP[2] on trouve :
– Le maintient de la sémantique de bout en bout du TCP.
– la compatibilité avec le protocole TCP traditionnel.

Ces avantages permettent à la solution A-TCP de fonctionner sur Internet. En plus, elle fournit une solution possible et efficace pour améliorer le débit du TCP dans les réseaux sans fil ad hoc.
Parmi les inconvénients de cette solution on cite :
– La dépendance du protocole de la couche réseau pour détecter les changements et les partitions ne peut pas être implémentée par tous les protocoles de routage.
– L'ajout de la couche ATCP dans la pile protocolaire du modèle TCP/IP nécessite des changements dans les interfaces actuellement utilisées.

3.2.5 Le TCP avec tampon et informations de séquence (TCP-BuS)

Les avantages de la solution TCP-BuS inclus une amélioration des performances et évite la retransmission rapide due à l'utilisation des tampons, les numéros de séquences et la retransmission sélective. TCP-BuS tire aussi l'avantage des protocoles de routage, spécialement les protocoles de routage à la demande tel que l'ABR. Les inconvénients du TCP-BuS inclus l'augmentation de la dépendance du protocole de routage et les tampons utilisés dans les nœuds intermédiaires (figure 3.1).

L'échec d'un nœud intermédiaire qui a copié des paquets dans son tampon peut mener à une perte des paquets et à une dégradation des performances. La dépendance du TCP-BuS sur le protocole de routage peut dégrader ses performances avec d'autres protocoles de routages qui ne possèdent pas des messages de contrôle similaire au protocole ABR[34].
Le tableau 3.1 présente une comparaison des solutions TCP :

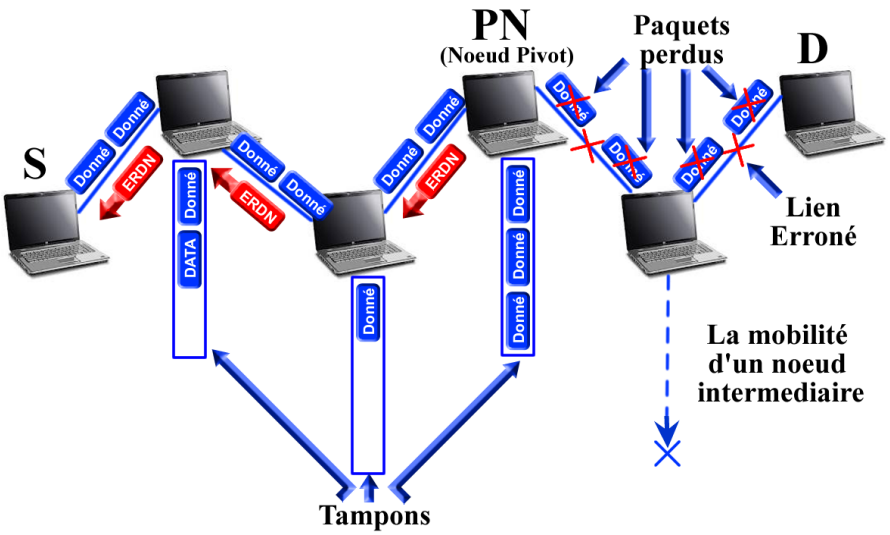

FIGURE 3.1 – Echec du chemin dans TCP-BuS

Défis	S-TCP	TCP-F	TCP-ELFN	A-TCP	TCP-BuS
Paquets perdus dues au BER ou à la collision	Idem au TCP traditionnel	Idem au TCP traditionnel	Idem au TCP traditionnel	Retransmit les paquets perdus sans exécuter un contrôle de congestion	Idem au TCP traditionnel
Coupures des chemins	Idem au TCP traditionnel	Le message RFN est envoyé à l'expéditeur TCP et l'état passe à geler	Le message ELFN est envoyé à l'expéditeur TCP et l'état passe en veille	Idem au TCP traditionnel	Le message ERDN est envoyé à l'expéditeur TCP et l'état passe à geler, le message ICMP DUR est envoyé à l'expéditeur TCP et TCP-BuS oblige le TCP à passer à l'état persistant
Les paquets en désordre	Idem au TCP traditionnel	Idem au TCP traditionnel	Idem au TCP traditionnel	A-TCP ré-ordre les paquets d'où TCP évite l'envoi dupliqué	Le désordre des paquets est détecté après l'exécution de la découverte du chemin
La congestion	Puisque la connexion est divisée, le contrôle de congestion est traité dans une zone par les nœuds proxy	Idem au TCP traditionnel	Idem au TCP traditionnel	Le message RFN est utilisé pour notifier l'expéditeur TCP. Le contrôle de congestion est idem au TCP traditionnel	Des messages explicites sont utilisés tel que ICMP

TABLE 3.1 – Une comparaison des solutions TCP pour les réseaux ad hoc sans fil

Défis	S-TCP	TCP-F	TCP-ELFN	A-TCP	TCP-BuS
La fenêtre de congestion après le rétablissement du chemin	Les nœuds proxy maintiennent la fenêtre de congestion et traitent la congestion	Idem au TCP traditionnel avant la coupure du chemin	Idem au TCP traditionnel avant la coupure du chemin	Recalculer pour le nouveau chemin	Recalculer pour le nouveau chemin
La notification explicite de l'échec du chemin	Non	Oui	Oui	Oui	Oui
La notification explicite de rétablissement du chemin	Non	Oui	Non	Non	Oui
La dépendance du protocole de routage	Non	Oui	Oui	Oui	Oui
Sémantique de bout en bout	Non	Oui	Oui	Oui	Oui
L'utilisation des tampons dans les nœuds intermédiaires	Oui	Non	Non	Non	Oui

TABLE 3.2 – Une comparaison des solutions TCP pour les réseaux ad hoc sans fil (Suite)

3.2.6 Motivations de la solution TCP pour MANet

La solution TCP-MANet a été proposée pour remédier aux insuffisances des solutions précédentes dans le but d'améliorer les performances du protocole TCP dans les réseaux sans fil ad hoc. Cette solution utilise le même mécanisme de la solution TCP-F pour différencier entre la perte des paquets due à la congestion du réseau et la perte des paquets due aux échecs des liens. Elle est basée sur les réactions avec les couches inferieures de la pile protocolaire (couche réseau et couche MAC). En conséquence, quelques modifications sont apportées sur le TCP traditionnel du coté expéditeur. Pour minimiser la perte des paquets, la solution TCP-MANet, s'inspire de la solution TCP-BuS en ajoutant un tampon uniquement dans le nœud intermédiaire ayant détecté un lien erroné (figure 3.2), ceci minimise l'utilisation de l'énergie et la perte de paquets.

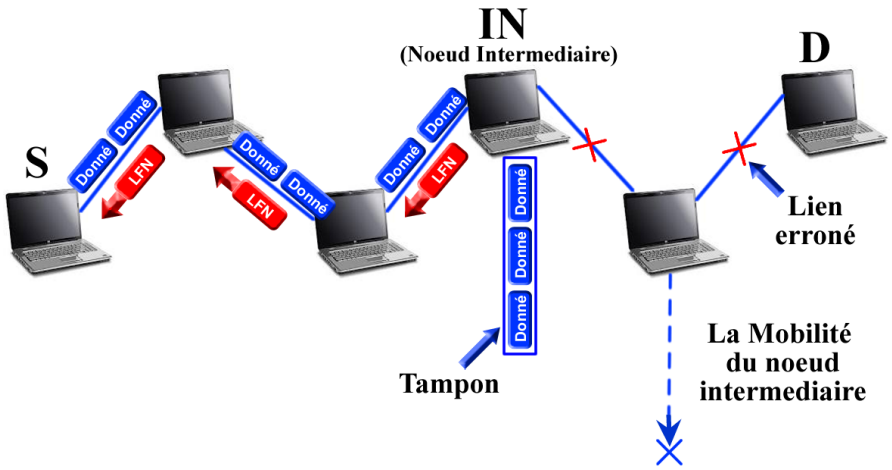

FIGURE 3.2 – Lien erroné dans TCP-MANet

Cette solution utilise aussi les informations de la puissance du signal inclus dans les paquets émis pour prédire l'échec des liens et pour minimiser la perte des paquets. Pour éviter la consommation inutile de la bande passante provoqué dans la solution TCP-ELFN, TCP-MANet garde le même mécanisme de la solution TCP-F dans le cas d'une découverte d'un nouveau chemin mais ajoute des améliorations pour éviter les messages de contrôle supplémentaires pour acheminer la notification de rétablissement du chemin présentés dans la solution TCP-F.
Les sections ci-dessous présentent une description détaillée des différents mécanismes de la solution TCP-MANet au niveau de l'expéditeur, au niveau du récepteur et au niveau des nœuds intermédiaires.

3.3 Description de la solution

La solution TCP-MANet est une extension du TCP traditionnel, elle regroupe un ensemble de mécanismes qui permettent à l'expéditeur TCP de différencier entre les paquets perdus suite à la congestion du réseau et les paquets perdus due à des liens erronés. Les mécanismes de la solution TCP-MANet reposent sur les capacités de la couche MAC et la couche réseau[35] pour savoir la cause de la perte de paquets au niveau des

nœuds intermédiaires. Elle utilise le protocole de routage AODV (Ad hoc On demande Distance Vector)[36]. Notre solution propose les améliorations suivantes :

La notification explicite : TCP-MANet utilise trois messages de contrôle pour notifier à l'expéditeur l'échec ou le rétablissement d'un chemin. Les messages utilisés sont : Link Failure Notification (LFN), Route Reestablished Notification (RRN) et Reestiblishment Expired Notification (REN). A la réception d'un message LFN transmit par le nœud intermédiaire ayant détecté un lien erroné, l'expéditeur gèle les horloges et stoppe la transmission. Et de façon similaire, après le rétablissement du chemin par le nœud intermédiaire ou le nœud source, ce dernier transmit un message RRN à l'expéditeur TCP pour l'initier à reprendre la transmission. Si le nœud source n'a pas réussi à rétablir le chemin il transmet le message REN à l'expéditeur TCP pour qu'il agit comme le TCP traditionnel dans le cas d'une déconnexion.

L'extension des valeurs d'expiration du temps de transmission : Pendant la reconstruction du chemin, les paquets qui ont traversés le chemin de la source jusqu'au nœud intermédiaire ayant détecté le lien erroné seront sauvegardés dans un tampon.

L'optimisation de la perte des paquets : Pendant la transmission normale des données, chaque nœud intermédiaire qui détecte une faible dégradation du signal de transmission, met les paquets caractérisés par une faible puissance de transmission dans un tampon. Les paquets sauvegardés ne seront pas éliminés du tampon jusqu'à ce que le nœud intermédiaire détecte une bonne puissance de transmission ou jusqu'à la réception d'un acquittement ayant un N° de séquence supérieur aux paquets copiés dans le tampon.

L'évitement de la retransmission rapide inutile : Après le rétablissement du chemin par le nœud intermédiaire vers la destination, l'expéditeur transmet les paquets à partir du dernier N° de séquence du paquet mis dans le tampon du nœud intermédiaire. Les paquets mis dans le tampon du nœud intermédiaire peuvent arriver à la destination plus tôt que les paquets transmis par l'expéditeur.

3.3.1 Les opérations de la solution TCP-MANet

Comme l'invalidation des chemins dépend des mouvements des nœuds, donc on doit discuter les actions prisent par notre solution au niveau du nœud source, au niveau du nœud intermédiaire et au niveau du nœud destination.

Les fonctions TCP-MANet au niveau du nœud source

Au niveau du nœud source, TCP-MANet transmet ses segments de la même manière que le TCP traditionnel lorsqu'il n y'a pas de messages de réaction (i.e. les messages LFN, RRN et REN). Le démarrage lent et les mécanismes de contrôle de congestion fonctionnent sans changement. Cependant, quand la source reçoit le message de réaction LFN, elle stoppe la transmission des paquets et gèle la connexion TCP comme le TCP-F (voir figure 3.3).

A titre d'exemple (voir figure 3.4), Lorsque le nœud intermédiaire détecte une faible puissance de transmission dans les segments (7, 8 et 9), il les met dans son tampon et vérifie l'état du lien. Le nœud intermédiaire

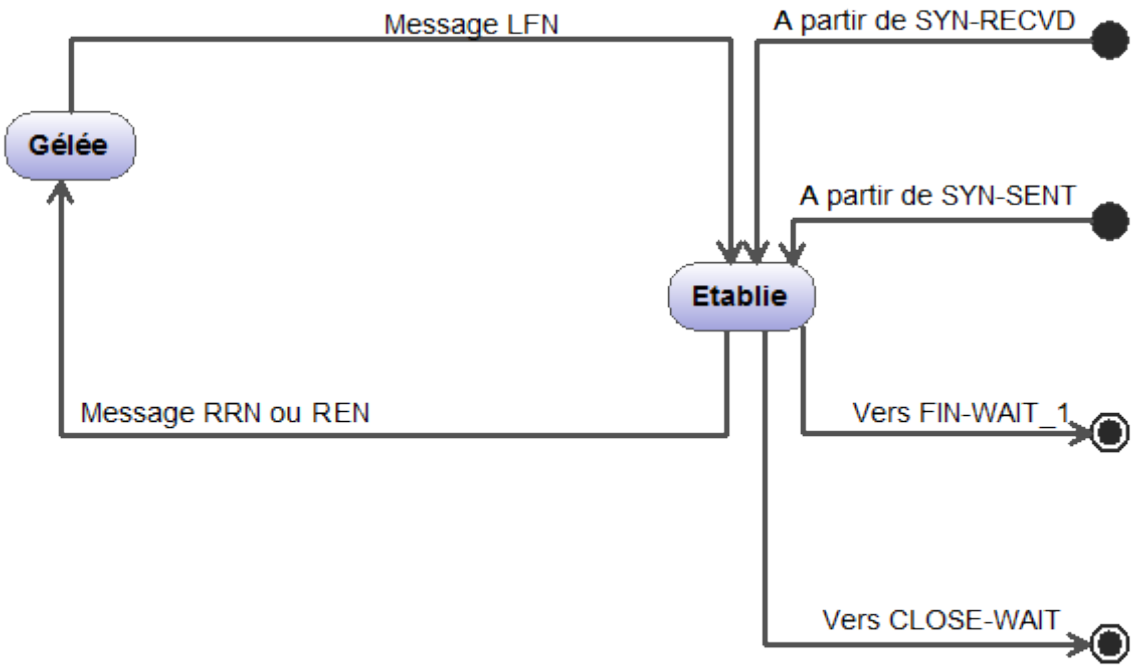

FIGURE 3.3 – Automate d'états finis TCP-MANet

utilise les capacités de la couche MAC pour vérifier la puissance de transmission du signal. Lorsque le nœud intermédiaire va transmettre le segment (10), il détecte que le lien est erroné, donc il génère un message LFN destiné à l'expéditeur TCP.
Quand le nœud source reçoit le message LFN, il stoppe la transmission des données et il gèle la connexion TCP. Comme le message LFN indique qu'un lien est erroné, le nœud source attend la réception d'un message RRN ou REN. A la réception du message RRN, le nœud source retire la valeur $LAST\_ACK$ du message RRN qui contient la valeur (15) et reprend la transmission des données à partir du segment (15) sans exécuter un démarrage lent. Les segments de (7) jusqu'à (14) sont transmis par le nœud intermédiaire après le rétablissement du chemin. Si la perte des paquets persiste après la reprise de transmission, TCP-MANet effectue un contrôle de congestion de la même manière que le TCP traditionnel.

L'expéditeur TCP gèle la connexion TCP et stoppe la transmission des données s'il reçoit un message LFN, le message LFN indique à l'expéditeur TCP qu'une perte de paquets est causée par un lien erroné. Ce message permet à la solution TCP-MANet de différencier entre la perte des paquets due à la congestion et la perte des paquets due à un lien erroné, donc l'expéditeur TCP continue dans l'état gelée jusqu'à la réception d'un message RRN ou REN, ceci permet d'éviter le test périodique de la découverte du chemin qui consomme assez de bande passante. Le message RRN indique à l'expéditeur TCP que la rupture du chemin est rétablie (comme illustré dans l'exemple précédant). Il est possible que le nœud intermédiaire ne réussi pas à établir un chemin vers la destination, dans ce cas ci le protocole de routage du nœud intermédiaire génère un message indiquant à la source une rupture du chemin vers sa destination. La solution TCP-BuS repose sur les capacités du protocole de routage pour rétablir le chemin erroné a partir du nœud PV jusqu'à la destination, si le protocole ne réussi pas à établir ce chemin, l'expéditeur TCP reste dans un état gelé jusqu'à ce que le protocole de routage rétabli le chemin. Ceci est un problème majeur de la solution TCP-BuS. Pour remédier à ceci, la solution TCP-MANet utilise le message REN qui indique à l'expéditeur TCP que le chemin vers la

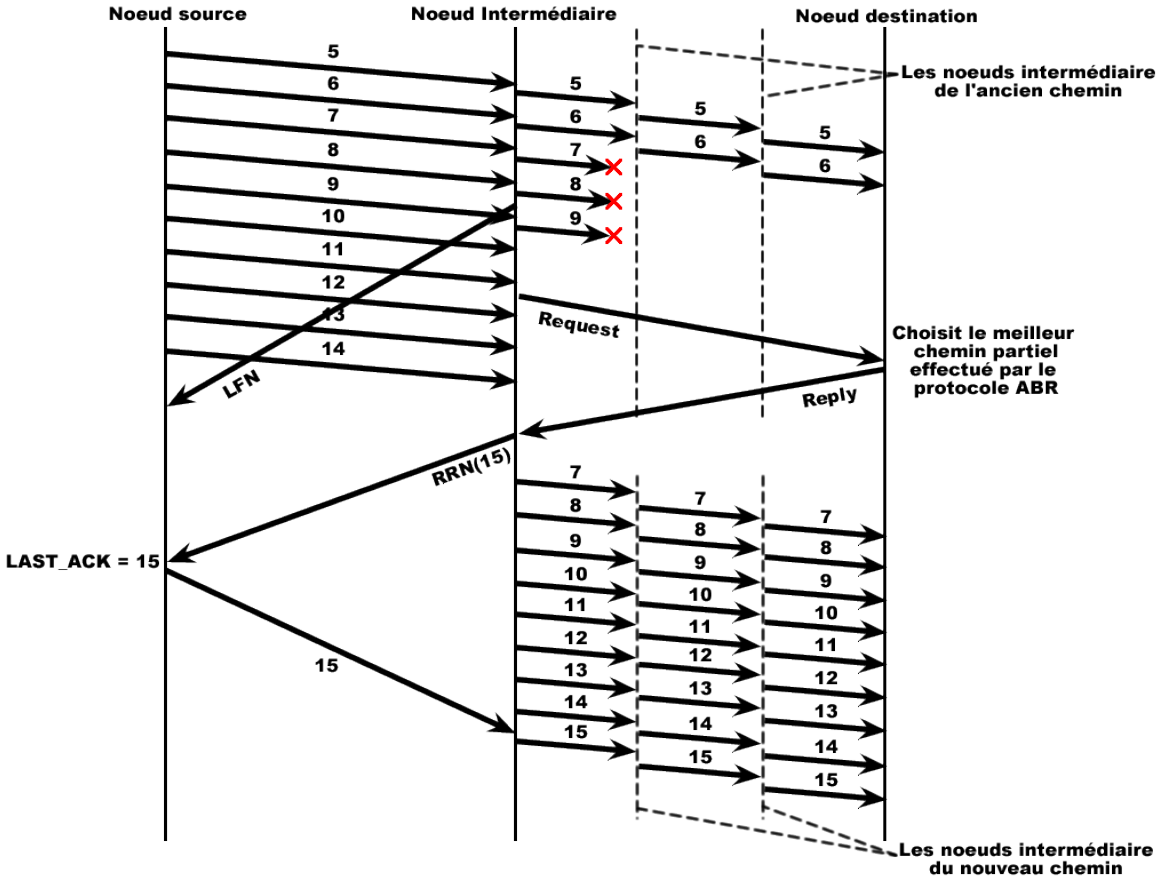

FIGURE 3.4 – Un exemple qui illustre le mécanismes de TCP MANet

source ne pourra pas être rétablie.
Comme il a été cité dans le paragraphe précédant, durant la période de l'état gelé, l'expéditeur TCP se met en attente de messages. S'il reçoit le message REN, l'expéditeur TCP quitte l'état gelé et reprend à l'état normal du TCP traditionnel ce qui permet à l'expéditeur TCP d'agir comme le TCP traditionnel lorsqu'il interrompe la connexion à cause d'une déconnexion du nœud destination.

Les fonctions TCP-MANet au niveau du nœud intermédiaire

TCP-MANet utilise les capacités de la couche MAC pour détecter un lien erroné, contrairement aux autres solutions discutées précédemment. TCP-MANet utilise la puissance du signal pour prédire l'échec du lien, ceci permet au nœud intermédiaire de prendre les opérations nécessaires pour réduire la perte des paquets et signaler rapidement le nœud source qu'un lien est erroné.
La couche MAC utilise les messages RTS et CTS pour maintenir les liens des voisins, donc chaque nœud transmet à son voisin un message RTS et se met en attente d'un message CTS de ce voisin. Si le nœud ne reçoit pas le message CTS pendant une durée déterminée, il conclut que le lien vers le voisin est erroné. Dans la solution TCP-MANet, chaque nœud intermédiaire teste la puissance du signal lors d'une réception du message CTS. Si la puissance du signal est inferieure au seuil des signaux faibles, le nœud intermédiaire ajoute le voisin ayant transmit le message CTS dans sa liste des mauvais voisins et continue son exécution normale. La figure 3.5 présente un organigramme propre à ce mécanisme.

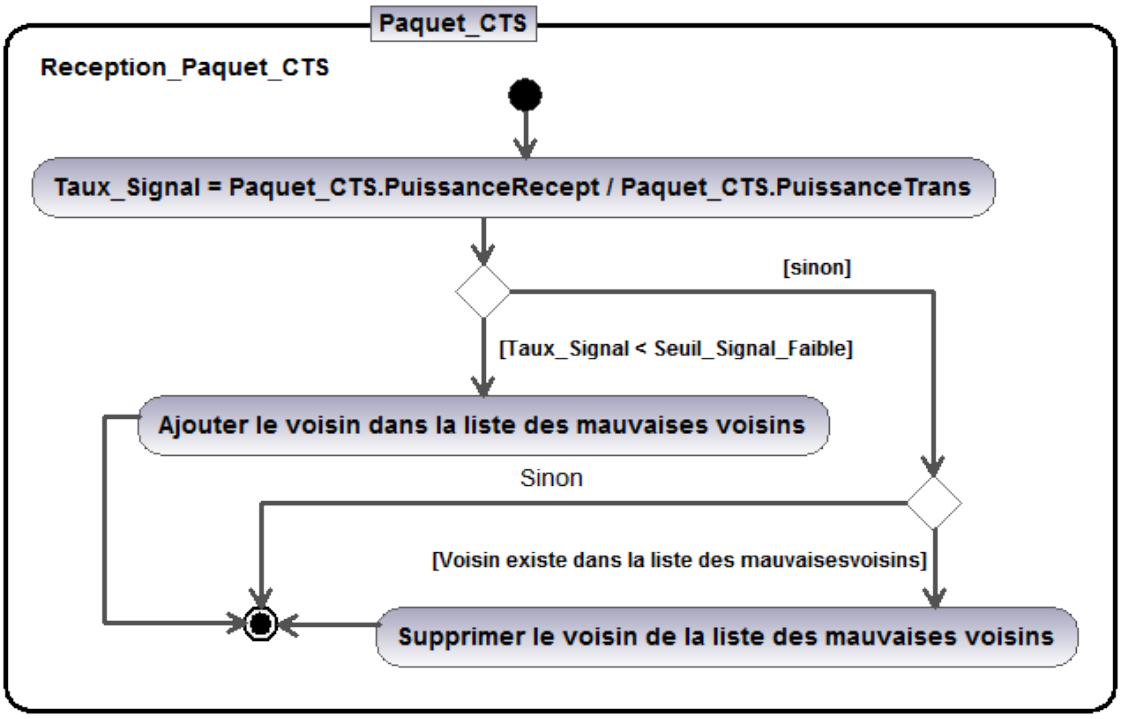

FIGURE 3.5 – Organigramme de réception du message CTS

Pour réduire la perte des paquets et éviter la retransmission sélective (cas du TCP-BuS), à chaque acheminement d'un paquet TCP par un nœud intermédiaire, ce dernier vérifie si le prochain saut du paquet TCP (prochain voisin) existe dans la liste des mauvais voisins. Si c'est le cas, le nœud intermédiaire ajoute la destination du paquet dans l'entrée du mauvais voisin et copie le paquet dans le tampon de la destination dans le but de le retransmettre en cas ou le lien est erroné. La figure 3.6 présente l'organigramme propre à ce mécanisme. Pour optimiser l'utilisation des tampons, et pour ne pas garder les paquets TCP acquittés dans les tampons. A la réception d'un paquet ACK, le nœud intermédiaire libère tous les paquets TCP mis dans le tampon ayant un N° de séquence inferieur à celui de l'acquittement.

Quand le nœud intermédiaire détecte un lien erroné, il génère un message LFN à la source et délègue le protocole de routage de la couche réseau pour rétablir le chemin. Pendant le rétablissement du chemin, le nœud intermédiaire met tous les paquets reçus dans son tampon. Si le chemin est rétabli, le nœud intermédiaire achemine les paquets sauvegardés dans son tampon vers la destination, libère le tampon et génère un message RRN à la source. Si le chemin n'est pas rétabli, il délègue au protocole de routage la notification de la rupture du chemin à la source et libère le tampon.

Les fonctions TCP-MANet au niveau du nœud destination

Le récepteur effectue la procédure normale de bout en bout du TCP traditionnel dans le chemin acquis dans le cas ou il n y'a pas de déconnexion de chemin. Donc aucune modification n'est effectuée du coté récepteur. Ceci pour garantir la compatibilité de notre solution avec le TCP traditionnel de bout en bout.

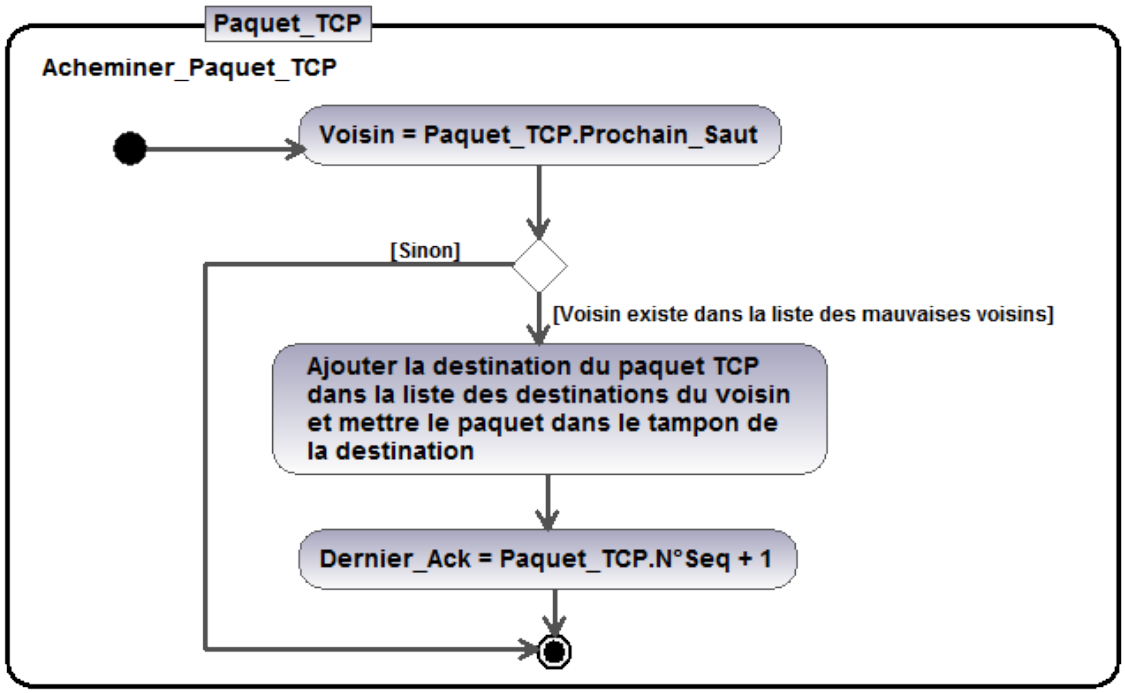

FIGURE 3.6 – Organigramme d'achminnement du paquet TCP

Le module de réaction inter-couche

Les solutions discutées précédemment soufrent du problème de la dépendance des capacités des couches inferieurs pour contrôler la congestion (tel que TCP-BuS). Si les couches inferieures ne fournissent pas ces capacités, la solution devienne inutile. Dans notre solution, ce problème est traité et on a utilisé la notion de modularité pour remédier à ce phénomène.
Dans TCP-MANet les réactions entre les couches sont regroupées dans une classe indépendante du protocole de routage et de la couche MAC. Cette classe implémente une interface de fonctions permettant aux protocoles des couches inferieurs à réaliser les réactions nécessaires pour signaler à l'expéditeur TCP la perte de paquets due à un lien erroné. La figure 3.7 présente le diagramme de classes des réactions inter-couches.

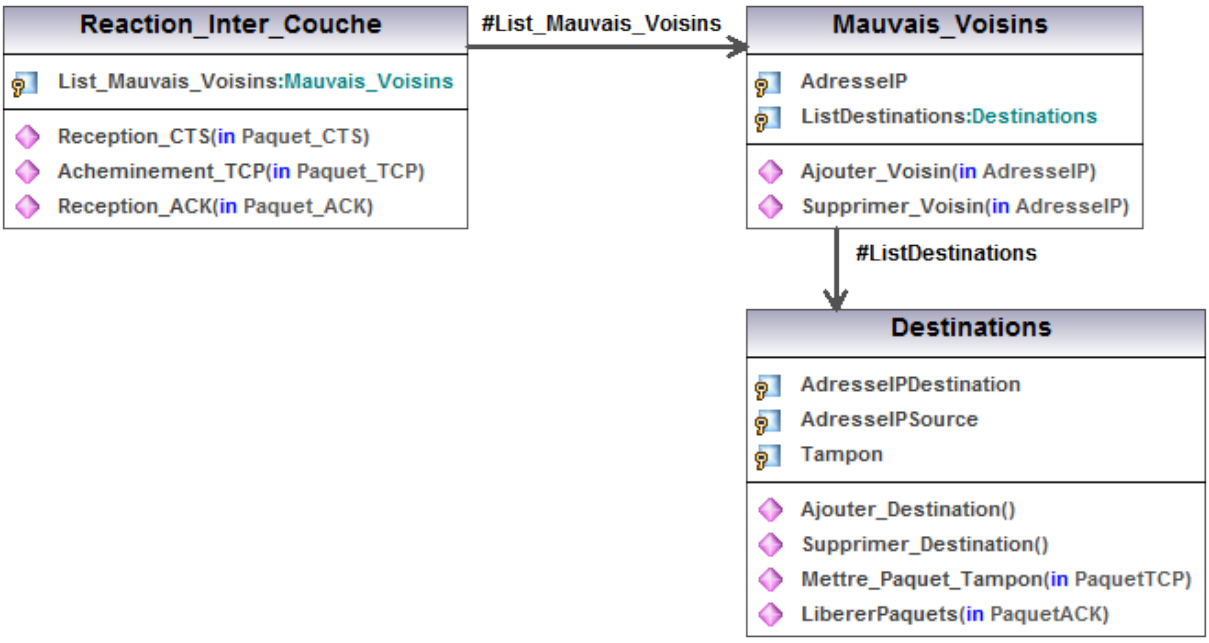

FIGURE 3.7 – Diagramme de classe des réactions inter-couche

3.4 Conclusion

Dans ce chapitre on a présenté les avantages et les inconvénients des solutions proposées dans la littérature pour améliorer les performances du protocole TCP dans le monde sans fil, motiver notre solution qu'on a baptisée TCP-MANet. Cette solution tente de remédier aux insuffisances des solutions existantes dans le but d'adapter le protocole TCP dans les réseaux sans fil ad hoc. Contrairement aux autres solutions, notre solution utilise les capacités des deux couches inferieurs (Réseau et MAC) pour contrôler réellement la congestion du réseau et utilise la puissance du signal pour réduire la perte des paquets.
Une description détaillée de la solution TCP-MANet a fait l'objet du reste du chapitre. Les fonctions effectuées par TCP-MANet au niveau du nœud source, du nœud destination et des nœuds intermédiaires sont expliquées et les diagrammes de classes utilisées pour concevoir notre solution. Il faut noter que notre solution peut être intégrée avec n'importe quel protocole de routage.
Pour mieux mettre en évidence les apports de notre solution, une analyse à base de simulation sous le simulateur NS-2 sera faite dans le chapitre suivant.

Chapitre 4

Les simulateurs réseaux

4.1 Introduction

La croissance d'utilisation des technologies réseaux hétérogènes à grande échelle et l'évolution du trafic des applications utilisateurs ont augmentés la complexité des réseaux. On remarque aujourd'hui que la complexité accrue a déjà influé sur internet et sur l'industrie des réseaux sans fil. Face à cette croissance de complexité, les concepteurs de réseaux et les chercheurs utilisent généralement la simulation afin de prédire les performances attendues des réseaux complexes et de comprendre le comportement des protocoles réseau existants non conçus à l'origine pour fonctionner dans les réseaux actuels. La simulation et de plus en plus utilisée pour prédire l'exactitude et les performances des modèles des nouveaux protocoles. En outre, l'utilisation de simulation apparaît maintenant comme exigence stricte dans les processus conduisant à des normes internationales, tel que la norme IMT-2000 de la troisième génération, sans fil, la téléphonie cellulaire.

4.2 Vue générale

Les simulateurs réseaux peuvent être divisés en plusieurs types : par protocoles, par technologie ou par méthodes de traitement, mais la catégorisation la plus utilisée est les méthodes de simulation. Il existe typiquement deux méthodes de simulation : méthode à événements discrets et méthode de simulation analytique[37]. La première méthode produit des prédictions dans le bas niveau du réseau (paquet par paquet), pour qu'ils deviennent exacts mais la génération des résultats est lente. La deuxième méthode utilise des modèles mathématiques pour produire les résultats à une vitesse beaucoup plus rapide, mais elle peut sacrifier l'exactitude. L'approche usuelle est de combiner les deux méthodes dans le but de fournir une exécution raisonnable en termes de temps et de maintenir l'exactitude dans les secteurs critiques.
De nos jours, plusieurs simulateurs de réseau existent pour répondre aux besoins d'évaluation des réseaux avant leurs déploiements. Actuellement les simulateurs ns-2, OPNET, OMNet++, QualNet sont couramment utilisés. Dans le haut niveau, tous ces simulateurs sont similaires : ils se focalisent sur une simulation à événements discrets au niveau des paquets et ils modélisent une large gamme de protocoles. Chaque simulateur a sa propre particularité.

4.3 Outils de simulation

Les outils de simulation des réseaux (simulateurs) fournissent un environnement pour créer virtuellement la configuration du réseau souhaité et effectuer par la suite un trafic sur ce réseau. Après l'exécution de la simulation, les résultats de simulation sont analysés. Ceci peut être réalisé en utilisant des outils qui visualisent l'exécution de la simulation, recapitalisent les résultats de simulation ou en analysant manuellement le fichier trace. Il existe une variété de simulateurs[38], [39]. Plusieurs dimensions sur la catégorisation de ces outils sont listées dans[40] et inclut l'utilisation, le niveau d'extensibilité, les mécanismes de personnalisations, la vitesse d'exécution (exécution parallèle), l'évolutivité avec la taille, le support d'émulation, la diversité des modèles et le niveau de support. En ce qui concerne l'utilisation quelqu'un peut imaginer une interface utilisateur graphique (GUI) qui peut aider le débutant à obtenir les premiers pas. Concernant l'extensibilité et la personnalisation, un simulateur peut permettre à l'utilisateur de créer ces propres modèles ou fournir une caractéristique d'utilisation avec des scriptes. Le niveau de support peut être aussi un facteur important. Un logiciel libre (open source) avec une mauvaise documentation, ex : une limite de commentaires dans le code source, ou des réponses dans les forums internet. « Malgré le simulateur ns-2[41] se développe pour inclure des nouveaux protocoles, mais malheureusement la documentation ne se développe pas ». Un produit commercial exige des fonds financiers, finalement quelques outils sont spécialisés sur certaines caractéristiques ou une région d'application telle que les environnements sans fil ou les simulations fluides à grande échelle. Selon[42] les facteurs les plus importants influant sur le choix du simulateur à utiliser sont : le coût, la simplicité d'utilisation, la disponibilité des modèles de simulation pour les applications désirées et l'exactitude (ou la justesse).
D'un autre coté, l'utilisation des outils de simulation présente des inconvénients. Comme il est indiqué dans[43], il faut ne pas faire aveuglement confiance aux résultats de simulation mais il faudra avoir une intuition et un bon jugement pour valider les résultats (même s'ils sont meilleurs) en utilisant des mesures, des expérimentations ou des analyses (mathématique). Là aussi c'est un danger quand plusieurs chercheurs utilisent le même outil de simulation, « toutes les simulations par les même bogues et les modèles supposés »[43]. Ce n'est pas évident de se familiariser facilement avec un outil de simulation, ceci nécessite beaucoup de temps. En conséquence les chercheurs utilisent souvent un seul outil de simulation pour toutes leurs études[44].
Dans ce qui suit, une description des simulateurs les plus utilisés et traités dans la littérature.

4.3.1 Le simulateur ns-2

Network Simulator Seconde version (ns-2) est un outil de simulation réseau à événement discrets. Ns-2 est un logiciel libre (open source) est peut être téléchargé à partir du site officiel ns [1]. Le développement de ns a commencé en 1989 et depuis il ne cesse d'être complété et amélioré. Depuis 1995 ns à été supporté par le centre de recherche et l'organisation de développement du département de la défense « DARPA »(Defense Advanced Research Projects Agency [2]). En 1996, la version 2 de ns a été introduite avec des changements

1. www.isi.edu.nsnam/ns
2. www.darpa.mil

majeurs architecturaux[41], et le simulateur a changé le nom vers ns-2. En 1997 (initialement destiné pour les réseaux câblés) le simulateur a été étendu par le projet Monarch[45] pour qu'il supporte les réseaux sans fil. Actuellement ns-2 est l'outil de simulation réseaux libre le plus largement utilisé, pas uniquement pour les simulations réseau, mais aussi pour « l'allocation des ressources, la communication au temps réel, les questions énergétique dans les réseaux ad hoc, les protocoles de transport dans les réseaux de capteurs et les stratégies de contrôle pour les robots sans fil »[46]. Une revue sur les simulations de réseau mobile ad hoc (MANet)[47] a publiée que ns-2 tient la première place avec 43,8% d'utilisation.
L'architecture ns-2 suit de près celle du model de référence OSI (Open Systems Interconnection). Cela signifie qu'un paquet simulé doit passer les couches réseau, liaison, MAC et physique. Actuellement, les couches supérieures de ns-2 supportent le réseau local (LAN), le réseau local sans fil (WLAN) et les réseaux par satellite. Le routage statique, dynamique et multicast est pris en charge ainsi que plusieurs techniques de file d'attente, tels que premier arrivé premier servi (FIFO) ou les algorithmes d'attentes équitables stochastiques. Dans la couche transport, ns-2 supporte les protocoles habituels tels que Transport Control Protocol (TCP), User Datagram Protocol (UDP) et Real-time Transport Protocol (RTP). Outre que les modèles fournis avec ns-2, il existe un grand dépôt de code contribué. Environ 120 contributions couvrant toutes les couches OSI, ainsi que des outils d'analyses ou de générations de trafic peuvent être téléchargés à partir de la contribution du site web[41].
En se rapprochant de l'implémentation du simulateur ns-2, un premier avis est le code source séparé en deux parties (c'est le changement majeur à partir de la première version). Le simulateur principal est implémenté dans le langage C++ pour fournir un temps d'exécution suffisant. La partie configuration est écrite dans l'extension orientée objet pour le langage de scripte TCL appelé OTCL. Pour commencer la simulation, l'utilisateur doit écrire un scripte TCL et il n'a pas besoin d'écrire ou de compiler les programmes C++. Donc la combinaison du langage C++ pour la simulation elle même et le langage de scripte TCL comme interface offre un compromis de performances et de convivialité. Uniquement les utilisateurs désirant ajouter des nouveaux protocoles ou modèles qui ont besoin d'écrire en C++.
Comme il est mentionné au-dessus, pour simuler sous ns-2, il faut écrire un scripte TCL. Le script comporte toutes les informations nécessaires pour exécuter la simulation, tels que les définitions de topologie et les protocoles à utiliser, la génération du trafic et les événements contrôlés par le temps ainsi que les commandes qui produisent les fichiers traces. Après simulation, il est nécessaire d'analyser les résultats de simulation qui sont inclus dans les fichiers traces, précisément pendant l'exécution de la simulation, ns-2 produit des fichiers traces pour différentes objectifs (avec différents formats), l'un des objectifs serait l'analyse du fichier trace à la main, ce fichier trace inclus les paquets transmis, reçus, acheminés ou perdus de tous les nœuds, chronologiquement ordonnés par temps d'événement. Plusieurs utilitaires existent pour visualiser et analyser les exécutions de simulation sous ns-2.

Les outils d'analyse ns-2

Trois outils sont utilisés pour analyser les résultats de simulation : Network Animator « nam [3] », l'analyseur de fichier trace « Trace Graphe [4] »et l'outil de visualisation et d'analyse pour les simulations sans fil

3. www.isi.edu/nsnam/nam
4. www.tracegraph.com

« iNSpect [5] ».

nam : Permet à l'utilisateur de voir l'exécution de la simulation. Cet outil peut visualiser la topologie du réseau (filaire et sans fil) et peut afficher le flux de paquet et les files d'attentes. En utilisant la barre et le curseur de temps, l'utilisateur peut y allez a n'importe quel moment dans le temps de simulation et peut aussi accélérer ou ralentir le temps d'exécution. « nam »est une partie du paquet ns-2.

Trace graphe : est un outil qui permet de présenter les fichiers traces graphiquement. Il possède plus que 200 différent graphs en 2d et 3d qui permet d'afficher le débit, le temps d'allez retour ou les informations dépendant a un nœud tel que le nombre des paquets acheminés. En plus des sorties graphiques, l'outil peut aussi associer les informations aux nœuds correspondants (ex : la somme de tous les bits TCP reçus à un certain nœud). Il faut noter que Trace graphe ne fait pas partie du paquet ns-2.

INSpect : est un autre outil de visualisation, initialement émergé du fait que « nam »est incapable de visualiser le trafic sans fil. INSpect peut visualiser les environnements sans fil, afficher les graphes de connectivité, afficher les plages de communication et afficher les coordonnées des nœuds statiques et mobiles. Il montre les flux de trafic et il utilise des couleurs pour indiquer les paquets transmis, reçus, acheminés ou perdus[48].

4.3.2 Le modélisateur OPNET

Le modélisateur OPNET est un outil de simulation réseau à événements discrets utilisé que ce soit par la communauté académique ou la communauté commerciale. Il n'est pas un logiciel libre comme ns-2, mais il possède une interface conviviale qui permet des créer et configurer des modèles de simulation, ainsi que l'analyse des résultats. OPNET est un projet de fin d'étude supérieur d'Alain Cohen (le co-fondateur et actuellement le directeur et le président) lorsqu'il été dans le MIT. OPNET se rapporte à un outil d'optimisation de l'ingénierie du réseau. Depuis sa création, Alain et son frère Marc (co-fondateur et actuellement le DG et le président du conseil) ont décidés de commercialiser le logiciel. Le premier projet été « OPNET Modeler »en aout 2000, un outil pour la modélisation et la simulation du réseau.
L'architecture OPNET divisé en plusieurs niveaux de modélisation, offre des solutions dans le domaine de la gestion des applications et des réseaux. Dans le domaine applicatif ces solutions permettent le monitoring et le suivi de l'expérience de l'utilisateur ; le monitoring en profondeur de l'environnement applicatif et la résolution des problèmes de performances. Dans le domaine réseau, les solutions OPNET permettent l'assurance de la configuration correcte du réseau, la planification et l'optimisation du réseau. OPNET offre aussi des logiciels de modélisation et de simulation soutenant la recherche et le développement dans les domaines des protocoles de communication et l'architecture des nouveaux systèmes de télécommunication ou les systèmes distribués.
En se rapprochant de l'architecture, on peut savoir que l'hiérarchie OPNET est constituée de trois niveaux principaux : la simulation réseau, les modèles des nœuds et les modèles des processus. Le haut niveau se

5. toilers.mines.edu/Public/NsInspect

réfère à la simulation réseau. Ce niveau contient la définition des couches du réseau, des nœuds et la configuration des attributs des nœuds. Les modèles des nœuds sont situés dans le deuxième niveau de l'hiérarchie, ils sont constitués d'un ensemble de modules organisés décrivant les différentes fonctions du nœud. Les modules dans les nœuds sont implémentés en utilisant les modèles de processus, c'est le bas niveau dans l'hiérarchie. Les modèles de processus sont constitués d'automates d'états finis, de définitions des fonctions des modèles et d'interfaces de processus qui définissent les paramètres pour communiquer avec d'autres modèles de processus et pour configurer les attributs. Les modèles d'automates à états finis sont implémentés en utilisant Proto C, la bibliothèque à événements discrets basés sur les fonctions C. La structure hiérarchique des modèles accouplés avec le support du langage de programmation C permet aux utilisateurs un développement facile des modèles de réseau ou de communication[49].
La zone de transite principale pour créer une simulation réseau est l'éditeur de projet. Il est utilisé pour créer un model réseau en utilisant les modèles de la bibliothèque standard, collecter des statistiques à propos du réseau, exécuter la simulation et afficher les résultats. L'éditeur de nœuds est utilisé pour créer des modèles de nœuds. Les modèles de nœuds sont utilisés par la suite pour créer des instances de nœuds dans les réseaux en utilisant l'éditeur de projet. Comme il est mentionné dans le paragraphe précédant, la structure interne des modèles de nœud dans OPNET est organisée sous forme de modules. La définition d'un nœud se fait en connectant plusieurs modules avec le flux de paquets et le câble statique. La connexion entre les modules permet aux paquets et aux informations d'état d'être échanger entre les modules. Chaque module placé dans un nœud sert à un objectif spécifique, tel que la génération des paquets, mettre les paquets dans la file d'attente, traiter les paquets ou transmet et réceptionne les paquets. Pour créer des modèles de processus qui contrôlent la fonctionnalité sous-jacente des modèles de nœud créé dans l'éditeur de nœud, on peut utiliser l'éditeur de processus. Les modèles de processus sont définit par un automate à état finit, les nœuds de l'automate représentent les états du processus et les arcs représentent les transitions entre les états. Les opérations effectuées dans chaque état ou pour chaque transition sont écrit dans un block C ou C++ incorporé[50]. L'interface utilisateur OPNET fournit des outils d'analyse des résultats qui permet de visualiser et analyser les exécutions de simulation.

L'outil d'analyse OPNET

Les outils d'analyse OPNET peuvent être employés durant le test de l'application pour observer en profondeur le comportement de toute l'architecture, y compris toutes l'infrastructure serveur, les bases de données et toutes les méthodes Java ou .NET permettant d'isoler les points de faiblesse de l'application. Le même système d'analyse en profondeur est employé en temps réel dans l'environnement de production grâce à la charge minime imposée par les agents de métrologie OPNET, permettant le diagnostic rapide de la cause des problèmes de performance.
Ces outils d'analyse réalisent plusieurs fonctions utiles supplémentaires comme par exemple, la création des graphes scalaires pour des études paramétrique, la définition des modèles pour les données statiques, la création et la sauvegarde des configurations d'analyses pour les voir ultérieurement, etc.

4.3.3 Le simulateur pour OMNet++

OMNet++ est un outil de simulation à événement discret multi objectif. Il est libre à un usage éducatif et non lucratif ; une version commercial qui s'appel « OMNEST [6] »est aussi disponible. OMNet++ est à l'origine d'un projet d'étudiant à l'université technique de Budapest en 1992. Depuis cette année, il a été constamment amélioré ; la version courante (en février 2012) est « 4.2.1 ».
L'architecture des composants de base et la structure orientée objet d'OMNet++ permettent la simulation de n'importe quel système où l'approche événement discret est adéquat[51]. Bien que son domaine d'application principale est la simulation des réseaux de télécommunication, « il a été utilisé avec succès dans d'autres domaines tel que la simulation des systèmes IT, les files d'attentes des réseaux, les architectures matériels et les processus de commerce »[51]. Comme OMNet est un simulateur a multi objectif, il fournit (uniquement) le noyau de simulation et les interfaces de programmation (API) pour écrire les simulations. Il ne fournit pas des composants pour un domaine d'application spécifique tel que les réseaux informatique. Cependant, la simulation des réseaux informatique est le domaine d'application principale, ils existent des plateformes qui fournissent des modules pour les simulations des réseaux filaires et sans fil tel que la INET Framework [7], ou Mobility Framework [8]. En se rapprochant de l'architecture, on peut savoir que dans OMNet++ toutes est « module ». Ce qui signifie que le model OMNet++ est constitué de plusieurs modules (hiérarchiquement emboîtés), en commençant par le module système (figure 4.1).

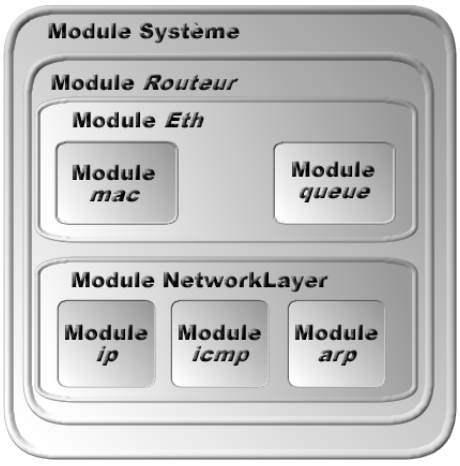

FIGURE 4.1 – Hiérarchie des modules OMNet++

Les modules des niveaux inferieures sont appelés « simple module ». Ils sont écrits en C++ et ils encapsulent le comportement actuel du système. Les modules des niveaux supérieurs appelés « compound module »sont constitués d'un ou plusieurs autres modules. Un exemple d'un module composé peut être le routeur. Le module routeur lui même est constitué de plusieurs autres modules ex : le module « eth »qui représente l'interface réseau ou le module « NetworkLayer ». Pour compléter l'hiérarchie, le module composé « eth »peut être constitué de modules simples « mac »et « queue », le module composé « NetworkLayer »peut être constitué de modules simples « ip », « icmp »et « arp ». Les modules OMNet++ sont décrits dans le

6. www.omnest.com
7. www.omnetpp.org/pmwiki/index.php ?n=Main.INETFramework
8. www.omnetpp.org/pmwiki/index.php ?n=Main.MobilityFramework

langage de description de réseau NED (notant qu'OMNet++ est un simulateur à multi objectif est le terme « réseau »ne signifie pas le réseau informatique mais un réseau OMNet++). La communication des modules se fait avec des messages à travers des portes. Une porte est l'interface d'entrée et sortie d'un module. Un message dans la simulation d'un réseau informatique est généralement une trame ou un paquet. Les messages typiquement se voyagent depuis un module simple vers un autre module simple.
Contrairement à ns-2 (qui utilise un langage de script pour décrire les modèles), les modèles OMNet++ sont écrits entièrement dans le langage de programmation C++. Ce qui signifie qu'un programme de simulation à besoin d'être compilée avant l'exécution de la simulation. Puisqu'on ne désire pas recompiler la simulation pour chaque changement d'un paramètre, certaines informations peuvent être chargées dynamiquement par la suite.
OMNet++ prend en charge une interface utilisateur graphique (GUI), ce qui signifie que l'exécution du programme de simulation affiche une interface graphique qui visualise la topologie du réseau et les flux de paquet et elle permet une analyse plus détaillée pour l'exécution de la simulation. L'utilisateur peut accélérer ou ralentir la simulation, " plonger " en profondeur dans l'hiérarchie du model à examiner ex : les valeurs de certaines paquets TCP.

Les outils d'analyse pour OMNet++

Comme ns-2, différents fichiers trace peuvent être créé durant l'exécution de la simulation. Avec la possibilité de créer des fichiers trace ayant un seul format, typiquement des fichiers de sortie vecteur et/ou des fichiers de sortie scalaire peuvent être créés. Les fichiers vecteurs sont des paires basées sur clé-valeur, les clés sont les estampilles de temps et les valeurs sont certaines valeurs dans ces estampilles (ex : la taille de la fille d'attente dans le temps). Les fichiers vecteurs peuvent être tracés avec l'outil « Plove »qui est une partie du paquet OMNet++. Les fichiers scalaires sont totalement utiles pour les statistiques, tels que le nombre de paquets perdus, ou le sommet du débit atteint. L'outil « **Scalars** », aussi est une partie du paquet OMNet++, pouvant être utilisé pour tracer les fichiers scalaires, et il permet aussi d'exporter (comme un format texte simple) les fichiers vers Matlab, ou des tabulaires tel que OppenOffice Calc ou Microsoft Exel.

4.3.4 Le simulateur QualNet

QualNet (GlomoSim) est un programme commercial pour les simulations de réseaux filaire et sans fil par « Scalable Network Technologies [9] ». Son architecture est constituée de trois couches : le noyau de simulation (un ordonnanceur à événements discret) comme couche de base, et la bibliothèque des modèles comme deuxième couche. La couche inferieur est définie par GUI Developer QualNet. Elle consolide six outils différents tels que le concepteur de scénario, l'animateur ou l'analyseur. Désormais toute l'architecture est simplement appelée le simulateur QualNet. Contrairement aux simulateurs disponible gratuitement ns-2 et OMNet++, l'environnement graphique QualNet permet de créer visuellement les scénarios de réseaux, et il permet aussi d'analyser les résultats de simulation dans une seul interface graphique[52].
Comme il est indiqué au-dessus, la création de scénario réseau est faite dans une interface graphique. Les

9. www.scalable-networks.com

informations de base (ex : l'emplacement des nœuds et les paramètres globaux et généraux) sont stockées dans un fichier de description de scénario (au format XML). Comme l'interface graphique est plus sophistiquée, il n'est pas nécessaire de changer ce fichier à chaque moment. Ceci s'applique essentiellement à tous les fichiers de configuration QualNet ; bien qu'il soit possible de manipuler ces fichiers dans un éditeur de texte (comme ils ne sont pas coder en binaire). Après la spécification des emplacements des nœuds et la définition de la topologie et les paramètres généraux, il est possible de configurer l'environnement sans fil, la pile protocolaire, la collection statique ou le traçage des paquets. QualNet supporte plusieurs protocoles de la couche application tel que Constante Bit Rate (CBR), File Transfert Protocol (FTP) ou VoIP.
L'exécution de la simulation est visualisée dans l'outil « Animator ». Durant l'exécution de la simulation, plusieurs sorties peuvent être activées ou désactivées dans cet outil tel que les messages de diffusions, les paquets reçus par succès (flux de paquet), les paquets perdus ou les files d'attentes. Comme dans « nam »et « OMNet++ », la vitesse de simulation pourrait être accélérée ou ralentie. Contrairement à « OMNet++ », l'analyse durant la simulation n'est pas possible. En sortie, les exécutions de simulations produisent différents fichiers. Le premier fichier de sortie (**.stat**) contient des informations statistiques, dépendant de la simulation et de la configuration, d'autres fichiers peuvent être crées, ex : à l'activation de traçage de paquets, le fichier (**.trace**) contient tous les paquets enregistrés.

Analyse de simulation avec QualNet

Les fichiers de sorties après exécution de la simulation peuvent être analysés dans le GUI Developer. A l'ouverture d'un fichier (**.stat**), l'outil d'analyse GUI Developer s'exécute. L'analyseur fonctionne dans un mode hiérarchique basé sur le modèle OSI. Ce qui signifie qu'il faut choisir d'abord la couche (ex : physique, MAC, réseau, transport, application), puis le protocole (ex : 802.11, 802.11DCF, IP, TCP ou FTP), et à la fin un métrique spécifique disponible (ex : les signaux reçu, les paquets CTS transmis, les fragments perdus, les paquets de données retransmit ou le débit), puis l'analyseur trace un graphe à barres montrant la métrique choisie pour les différents nœuds. Le fichier trace est affiché sous forme de tableur, chaque ligne montre un seul paquet. Les colonnes inclus le temps de simulation, le protocole, le nœud, le numéro de séquence ou l'action (émission, réception, mise en attente, retirer de la file d'attente,...). Il est possible de chercher dans le tableau ou filtrer des lignes pour faciliter l'analyse.

4.4 Choix du simulateur ns-2

Dans notre cas d'étude on a choisi le simulateur ns-2 pour mesurer les performances de notre solution. Comme il est présenté dans la section 3, le choix de l'outil de simulation dépend de quatre facteurs :

Le coût : Le simulateur ns-2 est un logiciel libre est peut être téléchargé à partir du site officiel ns.

La simplicité d'utilisation : Le simulateur ns-2 est implémenté dans le langage C++, et comme le simulateur est libre (le code source du simulateur est disponible) les utilisateurs ayant des connaissances sur le langage C++ peuvent ajouter facilement des nouveau modèles de protocole au simulateur. Les modèles de

simulation sont écrits dans le langage TCL sous forme de script, ce qui offre un compromis de performance et de convivialité. En plus, Le simulateur ns-2 offre trois outils pour analyser les résultats de simulation : l'outil d'animation du modèle de simulation « nam », l'outil d'analyse de fichier trace « Trace Graphe »et l'outil de visualisation et d'analyse pour les simulations sans fil « iNSpect ».

La disponibilité des modèles de simulation : comme il est indiqué précédemment (section 3.1) un taux de 43.8% d'utilisation du simulateur ns-2. Grâce à sa popularité, plusieurs modèles de simulation sont disponibles pour différents type de réseaux (réseaux filaires, réseaux sans fil, réseaux cellulaires, ...) et dans différents situations (différents modèle de mobilité de nœud, différents densité de nœud, différents topologie de réseau, ...).

L'exactitude : Selon[53] et[54], après l'utilisation des simulateurs ns-2, OPNET, OMNet++ et QualNet. La comparaison des résultats de simulations avec les résultats réels ont démontrés que le simulateur ns-2 produit des résultats plus proches à la réalité.

4.5 Le simulateur ns-2

Ns-2 est un simulateur à événements discrets, orienté objet écrit en C++ et en OTCL (C++ implémente le code qui s'exécuté fréquemment et OTcl configure le système de communication), il est fourni avec des outils d'analyses complémentaires eux-mêmes écrits en C/C++ ou TCL/Tk. Il contient des fonctionnalités fournissant un environnement permettant de réaliser des simulations entre autre, simuler les protocoles IP, TCP, le routage et les protocoles dans les réseaux filaires ainsi que mobiles. L'architecture réseau de ns-2 est basée sur le modèle en couches OSI[54].
Ns-2 permet d'exécuter tous types de scénarios sur des topologies définies par l'utilisateur. Le réseau est modélisé par ses sources de trafic (applications), ses protocoles (UDP, TCP), ses routeurs (avec leurs files d'attente) et les liens qui les relient. Le réseau est ensuite simulé, ce qui produit des traces et des statistiques. Des outils périphériques permettent l'animation du réseau (NAM : Network Animator) ou la conversion vers d'autres outils (par exemple xgraph pour dessiner des courbes) après une opération de filtrage.

4.5.1 Les nœuds mobiles

Avant de montrer comment utiliser le simulateur ns-2, comment écrire un script TCL pour simuler un réseau Ad hoc, il faut d'abord décrire la structure d'un nœud mobile utilisé dans ns-2 pour les réseaux ad hoc et qu'un script est écrit à partir de ses composants partant de la couche physique à la couche transport (du modèle en couches OSI sur le quel est basé le simulateur ns-2).

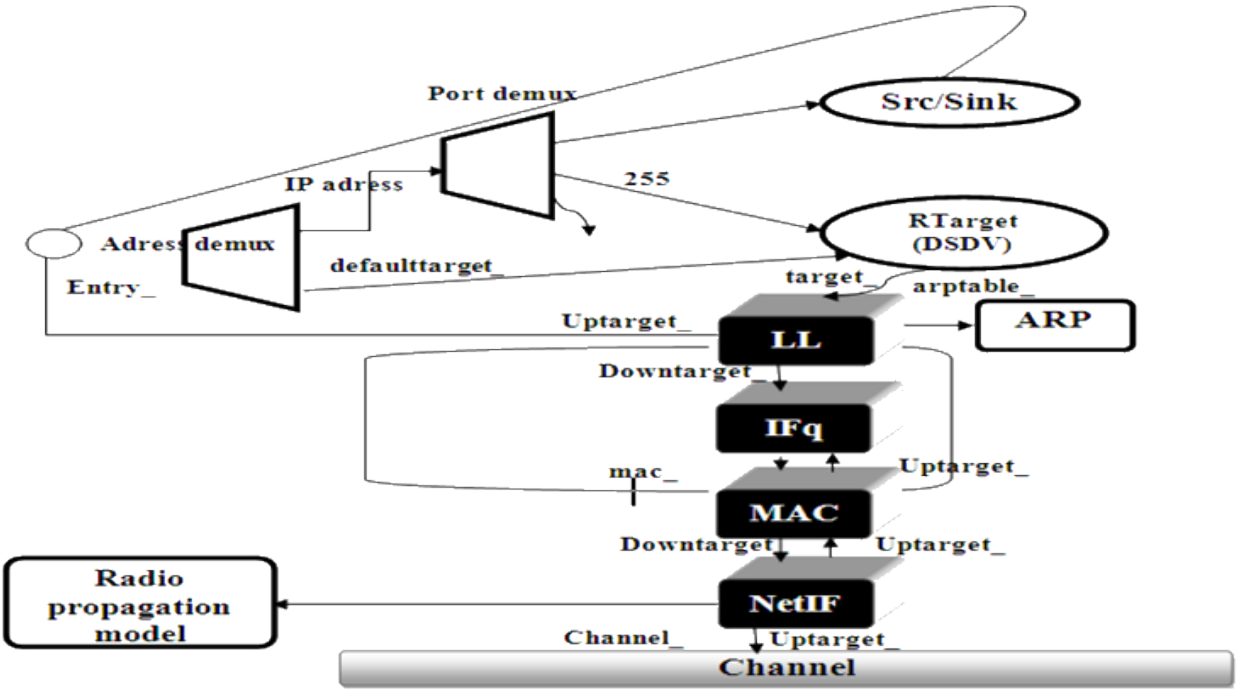

FIGURE 4.2 – Structure d'un nœud mobile sous NS pour le protocole DSDV et AODV

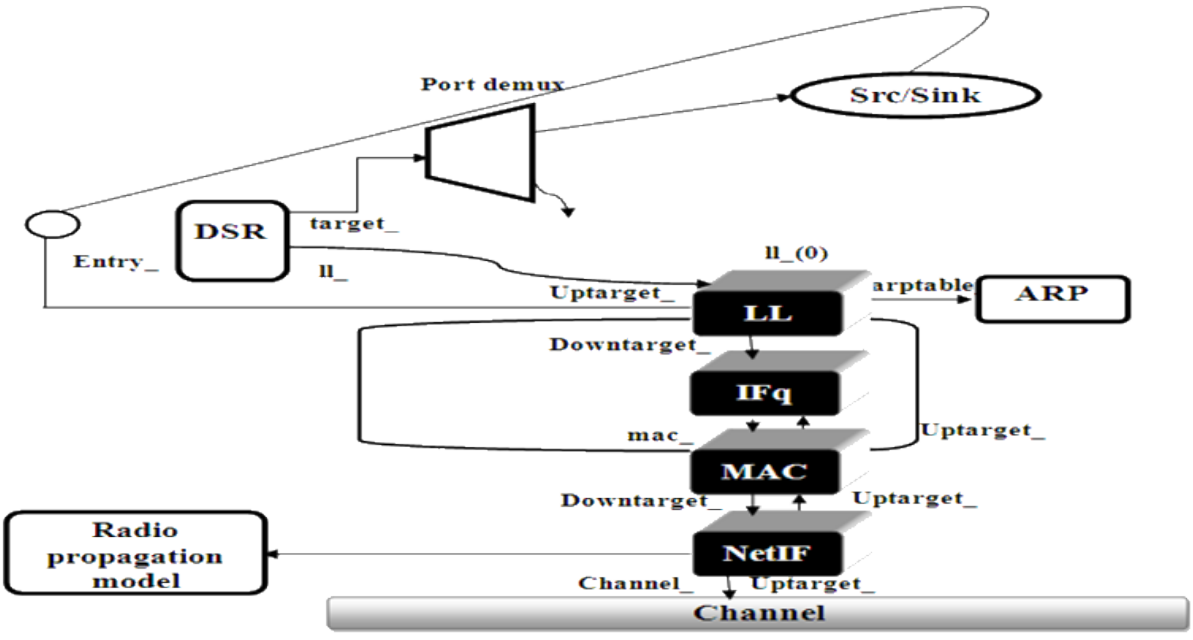

FIGURE 4.3 – Structure d'un nœud mobile sous NS pour le protocole DSR

La couche liaison (LL)

Les paquets à transmettre sont remis à LL par l'agent de routage, qui à son tour les remet à l'interface de la file d'attente. Pour les paquets reçus, la couche MAC les remet à LL puis au point « node entry »comme présenté sur la figure 4.2. Lorsque le protocole DSR est utilisé, les fonctionnalités du nœud mobile sont différentes (voir figure 4.3) ; Tous les paquets reçus par le nœud mobile sont dirigés vers l'agent DSR. C'est l'objet SRNode, dérivé du nœud mobile, qui réalise cette redirection.

L'ARP

Est un Protocole de résolution d'adresses qui reçoit les requêtes LL. Si l'ARP possède l'adresse matérielle de la destination, il la recopie dans l'en-tête du paquet MAC. Autrement dit, il annonce une requête ARP. Une fois l'adresse matérielle du prochain saut est connue, le paquet sera ainsi transmis.

L'interface de la file d'attente

La classe « PriQueue »de ns-2 est implémentée avec une priorité de file d'attente qui accorde la priorité aux paquets du protocole de routage en les insérant dans la tête de la file d'attente. Elle supporte un fonctionnement de filtrage qui filtre tout les paquets dans la file d'attente et supprime ceux qui ont une adresse de destination spécifiée.

La couche MAC

Utilise une structure RTS/CTS/DATA/ACK pour tous les paquets unicast et un envoi simple pour tous les paquets en broadcast.

L'interface réseau

Utilisée par le nœud mobile pour accéder au canal de communication. Le modèle de propagation radio reçoit des paquets transmis par l'interface de l'autre nœud.

Le modèle de propagation radio

Utilise une atténuation « Free-space »pour des distances proches, « Two Ray Ground »pour des distances lointaines, et le shadowing modèle.

4.5.2 L'utilisation de ns-2

L'utilisation des fonctionnalités du simulateur ne demande aucune connaissance de ce qui existe au niveau du C++. Il faut seulement connaître la programmation en langage TCL ainsi que les classes supportées par le simulateur au niveau OTCL et les méthodes et les attributs de chaque objet OTCL.

1. Du point de vue utilisateur, la mise en œuvre d'un modèle sous NS-2 se fait via une étape de programmation en langage TCL qui décrit la topologie du réseau et le comportement de ses composants ...etc., créant ainsi un fichier qui contient le script TCL dans n'importe quel éditeur de texte via l'instruction « **gedit <nom du fichier>.tcl** »

2. Vient ensuite l'étape de simulation (interprétation du script) par la commande « **./ns <nom du fichier>.tcl** ».
   L'interprétation d'un fichier d'extension tcl (interpréter ou lieu de compiler) permet de créer deux fichiers distincts pouvant avoir le même nom. Le nom de ces deux fichiers se trouve à l'intérieure du script tcl (le programme de simulation) :
   – Un fichier trace : <nom du fichier>.tr
   – Un fichier pour l'animation <nom du fichier>.nam
3. Visualiser la simulation avec la fenêtre NAM via la commande « ./nam <nom du fichier nam crée >.nam ».
4. L'interprétation des résultats se fait en explorant le fichier trace dans un éditeur de texte ou bien l'utilisation d'un outil graphique (Xgraph ou gnuplot) pour tracer des graphes via la commande : « ./ xgraph <nom du fichier>.xgr ou <nom du fichier>.tr ».

4.5.3 Un modèle de réseau Ad-hoc sous NS-2

Un modèle de réseau est constitué d'un ensemble d'éléments de base représentant les couches suivantes :

La couche physique

Est définie par la spécification du type du canal utilisé, la spécification du type de l'interface réseau, ainsi que le modèle de propagation. La spécification du nombre des nœuds mobiles (composant de base de tout script Tcl) est nécessaire.

La couche liaison

Définie par :
– Le type de l'interface de la file d'attente
– Le nombre maximal des paquets dans la file d'attente
– Le type du protocole MAC (ex 802.11)
– Le type de la couche liaison.

La couche réseau

Définie le type du protocole utilisé.

La couche transport

Définie par la spécification de :

– L'agent de communication, représentant les protocoles de la couche transport (TCP, UDP, TCPSink, Null,...etc.) ; ces agents sont attachés aux nœuds et connectés l'un à l'autre, ce qui représente un échange de données (connexion TCP, flux UDP).
– Chaque agent de transport est attaché à une application (un générateur de trafic de données) (CBR, ftp), chargé de la transmission des paquets depuis un agent de transport source à un agent de transport puits (destinataire).

4.5.4 Le protocole TCP

Cette section décrit les opérations d'un agent TCP sous ns-2. Il existe deux types majeurs d'agents TCP : les agents à un sens et les agents à deux sens. Les agents à un sens sont divisés en un ensemble d'agents TCP expéditeurs (qui obéissent à la congestion et les techniques de contrôles d'erreurs) et les récepteurs (« sink »). L'agent à deux sens est symétrique dans le sens pour représenter l'expéditeur et le récepteur, il est en cours de développement.
Les agents d'expédition TCP à un sens supportés sous ns-2 sont :

– **Agent/TCP** - l'expéditeur TCP « tahoe ».
– **Agent/TCP/Reno** - l'expéditeur TCP « Reno ».
– **Agent/TCP/Newreno** - l'expéditeur TCP « Reno »avec des modifications.
– **Agent/TCP/Sack1** - l'expéditeur TCP avec une répétition de sélection (suivant RFC2018).
– **Agent/TCP/Vegas** - l'expéditeur TCP « Vegas ».
– **Agent/TCP/Fack** - l'expéditeur TCP « Reno avec acheminement d'acquittement ».
– **Agent/TCP/Linux** - l'expéditeur TCP avec la prise en charge d'acquittements sélective qu'il exécute un module de contrôle de congestion à partir d'Unix.

Les agents de réception TCP à un sens supportés sous ns-2 sont :

– **Agent/TCPSink** - le récepteur TCP qui acquitte chaque paquet reçu.
– **Agent/TCPSink/DelAck** - le récepteur TCP avec un délai configurable par acquittement.
– **Agent/TCPSink/Sack1** - le récepteur TCP avec des acquittements sélectives.
– **Agent/TCPSink/Sack1/DelAck** - le récepteur TCP « Sack1 »avec « DelAck ».

L'agent TCP à deux sens supporté sous ns-2 est « **Agent/TCP/FullTcp** ».

Les expéditeurs TCP à un sens

Le simulateur ns-2 supporte plusieurs versions d'un expéditeur TCP abstrait. Ces objets tentent de capturer l'essentiel du comportement de congestion et le contrôle d'erreur TCP, mais ils ne sont pas destiner à être identiques à la réalité. Ces versions ne possèdent pas de fenêtre de congestion dynamique, ils calculent le nombre de segments et d'acquittements dans les unités de paquets. Il n y'a pas d'établissement ou de fermeture de connexion SYN/FIN.

L'expéditeur TCP de base (Tahoe TCP) L'agent TCP « Tahoe »effectue le contrôle de congestion et l'estimation du temps aller-retour d'une façon similaire à la version TCP publiée avec le système Unix « Ta-

hoe »4.3BDS de l'université de Berkeley. La fenêtre de congestion est incrémentée par un paquet à chaque réception d'un nouvel acquittement pendant le démarrage lent (quand $cwnd\_ < ssthresh\_$) et incrémentée par $\frac{1}{cwnd\_}$ à chaque réception d'un nouvel acquittement pendant l'évitement de la congestion (quand $cwnd\_ \geq ssthresh\_$)).

Les réponses à la congestion : TCP « Tahoe »suppose qu'un paquet est perdu à cause d'une congestion quand il observe $NUMDUPACK$ (défini dans le fichier tcp.h, actuellement dans la version ns-2.34 il est définit à trois) duplications d'acquittements, ou à l'expiration du temps de transmission. Dans les deux cas, TCP « Tahoe »réagit en ajustant ssthresh_ à la moitié de la taille de la fenêtre actuelle (le minimum de $cwnd\_$ et $window\_$) ou à 2 si elle est la plus grande valeur. Puis il recule la fenêtre de congestion $cwnd\_$ à la valeur initiale $windowInit\_$, d'où un démarrage lent.

L'estimation du temps aller-retour et l'expiration de la sélection RTO : Cinq variables sont utilisées pour estimer le temps d'aller-retour et ajuster le temps de retransmission : $rtt\_$, $srtt\_$, $rttvar\_$, $tcpTick\_$ et $backoff\_$. TCP initialise $rttvar\_$ à $\frac{3}{tcpTick\_}$ et $backoff\_$ à 1. A chaque ajustement du temps de retransmission, le temps d'expiration est ajusté à $Current\_Time + \max(backoff\_ * tcpTick\_(srtt\_ + 4 * rttvar\_ + 1), 64)$ secondes.
Un échantillon RTT (Round-Trip Time) est calculé en déduisant le temps actuel du champ « time echo »inclus dans le paquet d'acquittement. Lorsque le premier échantillon est prélevé, sa valeur est utilisée comme valeur initiale de la variable $rttvar\_$. Pour les prochains échantillons, les valeurs sont mises à jours comme suite :

$$srtt\_ = \frac{7}{8} * srtt\_ + \frac{1}{8} * sample \tag{4.1}$$

$$rttvar\_ = \frac{3}{4} * rttvar\_ * \frac{1}{4} * |sample - srtt\_| \tag{4.2}$$

La configuration

L'exécution d'une simulation TCP nécessite la création et la configuration de l'agent TCP en attachant l'agent à une source de donnée dans le niveau application (un générateur de trafic), puis on démarre l'agent et le générateur de trafic.

La configuration simple

```
set ns [new Simulator];# preamble initialization
set node1 [$ns node];# l'agent résider dans ce nœud
set node2 [$ns node];# l'agent résider dans ce nœud
set tcp1 [$ns create-connection TCP $node1 TCPSink $node2 42]
$tcp set window_ 50;# configure l'agent TCP
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 0.0 « $ftp start »
```

Cet exemple illustre l'utilisation du simulateur construit dans la fonction create-connection. Les paramètres

de cette fonction sont : l'agent source à créer, le nœud source, l'agent cible à créer, le nœud cible et l'identifiant du flux utilisé dans la connexion. Cette fonction crée deux agents, ajuste le champ « ID »dans les agents, attache l'agent source et l'agent cible à leurs nœuds respectifs et à la fin connecte les deux agents (ex : ajuste les ports et les adresses appropriés à la source et la destination). La valeur de retour de cette fonction et le nom de l'agent source créé.

La source de données TCP : l'agent TCP ne génère aucune donnée d'application ; donc, l'utilisateur de simulation doit connecter n'importe quel module de génération de trafic a l'agent TCP pour générer des données. Deux applications communes sont utilisées pour TCP : FTP et Telnet. FTP représente un transfert d'une masse de données de grande taille, et Telnet choisit sa taille de transfert aléatoirement à partir d'une bibliothèque TCP.

Autres paramètres de configuration

En plus du paramètre de la fenêtre $window\_$ listé au-dessous, l'agent TCP supporte des variables de configuration additionnelles. Chacune de ces variables décrites dans cette section sont des classes et instances de classe. Le changement de la classe change les valeurs par défaut pour tous les agents qui sont créé ultérieurement. Le changement de l'instance d'un agent particulier affecte uniquement la valeur utilisée par cet agent. Par exemple :

```
Agent/TCP set window_ 100;# Change la valeur de la classe
$tcp set window_ 2.0;# Change la valeur de window_ uniquement pour l'objet
$tcp
```

Dans la simulation, il n'est pas nécessaire d'ajuster tous les paramètres. Les paramètres les plus souvent modifier sont $window\_$ et $packetSize\_$. Le premier paramètre limite la fenêtre de congestion, et il est considéré pour jouer le rôle de la fenêtre annoncée du récepteur dans le TCP réel (bien qu'il reste constant). La taille du paquet essentiellement fonctionne comme la taille MSS dans le TCP réel. Les modifications apportées sur ces paramètres peuvent avoir un effet sur le comportement du protocole TCP.

Les autres expéditeurs TCP à un sens

TCP Reno L'agent TCP Reno est plus similaire à l'agent TCP Tahoe, à l'exception qu'il inclut aussi le mécanisme de récupération rapide « fast recovery », d'où la fenêtre de congestion est « gonflée »par le nombre d'acquittements dupliqués reçus par l'expéditeur avant la réception d'un nouvel acquittement. En plus, l'agent TCP Reno ne retourne pas au démarrage lent durant la retransmission rapide.

TCP new Reno Cet agent est basé sur le TCP Reno, mais il change les actions pris à la réception d'un nouvel acquittement. Pour sortir de la récupération rapide, l'expéditeur doit recevoir un acquittement pour le plus grand numéro de séquence envoyé. Ainsi, de nouveaux acquittements ne gonflent pas la fenêtre.

TCP Vegas Cet agent implémente TCP Vegas[55]. Il a été contribué par Ted Kuo. TCP Vegas est une nouvelle implémentation du protocole TCP qui permet d'obtenir une amélioration entre 40 et 70% de débit. Cet agent ne comporte pas des changements, c'est une implémentation simple qui agit avec n'importe quelle

autre implémentation TCP. En fait, tous les changements sont limités au coté expéditeur. En outre, cette amélioration du débit n'est plus atteinte par une stratégie agressive de retransmission qui gaspille effectivement beaucoup de bande passante des connexions TCP. Plutôt, elle est atteinte par une utilisation plus efficace de la bande passante disponible en ajoutant un nouvel mécanisme de retransmission, en améliorant le mécanisme de contrôle de congestion et en modifiant le mécanisme de démarrage lent.

TCP Sack Cet agent implémente la retransmission sélective, basée sur des acquittements fournit par le récepteur. Il suit la solution d'acquittements décrit dans[56], et il a été développé par Matt Mathis et Jamshid Mahdavi.

TCP Fack Cet agent implémente TCP « acheminement d'acquittement », une modification du TCP Sack décrit dans[57].

TCP Linux Cet agent exécute des modules de contrôle de congestion TCP importés du noyau Linux. L'agent génère des résultats de simulation qui sont conformément avec le comportement des hôtes Linux.

Les récepteurs TCP à un sens

Les récepteurs TCP décrits en-dessous représentent des récepteurs de données à un sens. Ils doivent échanger le trafique avec un objet « TCP sink ».

TCP Sink de base L'objet de base TCP Sink (Agent/TCPSink) est responsable pour retourner des acquittements pour l'objet TCP source. Il génère un acquittement par paquet reçu. La taille du paquet peut être configurée. La création et la configuration de l'objet TCP Sink est effectuée automatiquement par un appel de la bibliothèque.

Paramètres de configuration `Agent/TCPSink set packetSize_ 40`

TCP Sink Delayed-ACK L'objet Delayed-ACK Sink (Agent/TCPSink/DelAck) est disponible pour simuler un récepteur TCP qui acquitte au moins une fois par paquet reçu. Cet objet contient une variable $interval\_$ qui donne le nombre de secondes d'attente entre deux acquittements. L'objet TCP sink avec des acquittements par délais implémente une politique agressive où seulement les paquets dans l'ordre qui sont acquittés dans le délai. Les paquets en désordre causent immédiatement une génération d'acquittement (acquittement dupliquer).

Paramètres de configuration `Agent/TCPSink/DelAck set interval_ 100ms`

TCP Sink Sack L'objet TCP selective-acknowledgment sink (Agent/TCPSink/Sack1) implémente la génération SACK modélisée après la description de l'ACK dans RFC 2018. Cet objet inclus une variable $maxSackBlocks\_$ qui donne le nombre maximum de bloques d'informations dans un acquittement disponible pour contenir les informations SACK. La valeur par défaut de cette variable est 3, en conformité de

l'utilisation prévue de SACK avec RTTM. Les acquittements sélectifs avec délai sont aussi implémentés par l'objet Agent/TCPSink/Sack1/DelAck.

Paramètres de configuration `Agent/TCPSink set maxSackBlocks_ 3`

Les agents TCP à deux sens (TCP complet)

L'objet Agent/TCP/FullTCP est une nouvelle adition à la suite des agents TCP supportés dans le simulateur ns-2 et il est en cours de développement. Il est différent (et incompatible avec) des autres agents, mais il utilise une partie de la même architecture. Il diffère des autres agents dans les manières suivantes :

- Les connexions peuvent être établies et libérées (les paquets SYN/FIN sont échangés).
- Le transfert de données bidirectionnelles est supporté.
- Les numéros de séquences sont inclus dans les octets et non pas dans les paquets.

La génération des paquets SYN (et leurs acquittement) peut être d'une importance cruciale pour modéliser le comportement réel lors de l'utilisation de nombreux transfert de données courts. Une connexion typique TCP se produit avec une ouverture active en envoyant un SYN, l'ouverture passive répond par un SYN+ACK, puis l'ouverture active répond par un ACK, a quelque moment plus tard l'ouverture active envoi le premier segment de donnée.

4.5.5 Traçage dynamique du TCP

Le comportement du protocole TCP est souvent observé par la construction d'un graphe qui représente le numéro de séquence par rapport au temps. Typiquement, le traçage est effectué en permettant le suivi d'un lien sur lequel les paquets TCP vont passer. Deux méthodes de traçage sont supportées : celle par défaut est utilisée pour tracer les agents TCP, et une extension est utilisée pour l'agent Full TCP.

4.5.6 Traçage dynamique du TCP à un sens

Les paquets TCP générés par les agents TCP à un sens et destinés à un agent TCP sink en passant par un lien de traçage générant des lignes dans le fichier trace dans la forme suivante :

```
+ 0.94176 2 3 tcp 1000 ---- 0 0.0 3.0 25 40
+ 0.94276 2 3 tcp 1000 ---- 0 0.0 3.0 26 41
d 0.94276 2 3 tcp 1000 ---- 0 0.0 3.0 26 41
+ 0.95072 2 0 ack 40 ---- 0 3.0 0.0 14 29
- 0.95072 2 0 ack 40 ---- 0 3.0 0.0 14 29
- 0.95176 2 3 tcp 1000 ---- 0 0.0 3.0 21 36
+ 0.95176 2 3 tcp 1000 ---- 0 0.0 3.0 27 42
```

Lors du traçage TCP, les paquets de types **TCP** et **ACK** sont pertinentes. Le premier champ d'une ligne inclus l'évènement produit par un paquet (transmission, réception ou perte de paquet), le deuxième champ c'est le temps d'arriver de l'évènement, le troisième et le quatrième champ inclus les adresses sources et

destination, le cinquième champ inclut le type de paquet, le sixième champ inclut la taille du paquet et le onzième champ inclut le numéro de séquence. Comme par exemple, la première ligne du fichier trace présente une transmission du 25^{e}(numéro de séquence) paquet TCP de taille 1000 bits dans 0.94276^{e}seconde à partir de la source 2 vers la destination 3.

4.5.7 Traçage dynamique du TCP a deux sens

Les paquets TCP générés par l'agent Full TCP et passés par un lien de traçage contenant des informations supplémentaires non affichées par l'objet de traçage par défaut. En activant l'état $show\_tcphdr\_$ dans l'objet de traçage, trois champs d'entêtes additionnels sont écrits dans le fichier trace : numéro d'acquittement, les états TCP spécifique et la taille de l'entête.

4.6 Conclusion

Dans ce chapitre quatre outils largement utilisés pour les simulations du réseau (ns-2, OPNET, OMNet++ et QualNet) sont décrits et comparés. Deux études de[53] et [54] on montrés que le simulateur ns-2 produit des résultats de simulation plus proches aux résultats réels. En se basant sur ces études, notre choix s'est penché vers le simulateur ns-2.
Une brève description des différentes versions du protocole TCP implémentés dans le simulateur ns-2 est faite. Pour se rapprocher de ces implémentations, on présenté aussi les configurations de base utilisées pour chaque version.
Le prochain chapitre sera consacré à la description de notre proposition qu'on a appelée TCP MANet. Cette solution est une extension de l'expéditeur TCP standard pour qu'elle supporte l'environnement ad hoc.

Chapitre 5

Simulation de la solution TCP-MANet

5.1 Introduction

Le recours croissant à la simulation soulève des enjeux pour déterminer l'exactitude et le fond de prédiction des modèles de simulation spécifiques. Actuellement il n'existe aucune pratique ou technique qui permet de valider les simulations réseau et d'évaluer la fiabilité de leurs résultats. Les premiers travaux de recherches sur les réseaux utilisent l'expérimentation et la modélisation mathématique pour prouver la faisabilité et pour établir des bornes sur les performances attendues. Aujourd'hui la simulation est utilisée pour :

- Prédire les performances des réseaux et des protocoles afin de faciliter l'évaluation des technologies et la planification des capacités,
- Prédire le comportement attendu des nouveaux protocoles réseau,
- Explorer rapidement une gamme potentielle de conception des protocoles à travers une évaluation rapide. Pour chacun de ces fins, les résultats de simulation doivent être justifiés. La validation des résultats obtenus est un processus d'assurance que le model simulé soit capable de fournir des réponses significatives aux questions étudiées[58].

Dans ce chapitre on va présenter en premier lieu les hypothèses qu'on doit considérer pour valider les résultats de simulation, et par la suite on teste notre solution TCP-MANet dans différentes situations.

5.2 Hypothèses de validation

Pour valider une simulation particulière, l'utilisateur doit d'abord voir ce qui existe dans la réalité. L'approche évidente est de comparer les résultats de simulation à partir d'un cas particulier de l'implémentation d'un réseau dans le monde réel. Ceci permet de faire une comparaison des résultats de simulation avec des expériences réelles. La comparaison réelle peut fonctionner pour des petits réseaux mais lorsque les topologies de réseau sont larges ou lorsque les protocoles sont sous-spécifiés, la validation à travers une comparaison directe peut s'avérer difficile[58].

5.2.1 Comparaison des spécifications par rapport aux implémentations

Traditionnellement, les protocoles réseau ont été définis uniquement au niveau nécessaire pour assurer une communication réussite entre les nœuds et pour obtenir des performances raisonnables. Ceci implique que plusieurs décisions de nombreuses optimisations d'ingénierie peuvent être laissées aux exécutants du protocole. Dans la plus part des cas, différentes décisions mènent à différentes performances, mais sans compromettre le comportement de base codé dans la spécification.
La comparaison avec les implémentations du protocole particulier peut ne pas être idéale dans tous les cas. Une simulation particulière peut être dépassée avec l'évolution des protocoles ou avec le changement du trafic. Dans ce cas les simulations peuvent nécessiter une autre validation avec les futures implémentations plutôt que les implémentations actuelles. Les utilisateurs de simulation doivent comprendre ce qui est fournit par le simulateur et ce qui est approprié pours leurs expériences.
TCP fournit un exemple où la spécification admet une gamme d'implémentations avec différents performances. Les détails de l'algorithme d'acquittement et les paramètres tel que la taille de la fenêtre peuvent altérer de deux jusqu'à dix fois l'état initiale ou l'état stable du débit. Dans un tel cas, les simulations peuvent être validées par rapport à une implémentation spécifique ou par rapport à une enveloppe de performances de la spécification.

5.2.2 Comparaison des simulations avec l'évolution des conceptions du protocole

Les conceptions du protocole évoluent avec le temps et les implémentations publiées créent des décalages avec les versions académiques. Par exemple, l'implémentation de TCP Reno a connu des problèmes de performances quand plusieurs paquets sont perdus dans un seul aller-retour. Ces problèmes de performances corrigés dans l'option TCP avec des acquittements sélectifs peuvent produire une large différence de débit entre TCP Reno et d'autres variantes TCP. La validité de telles comparaisons dépend de leur interprétation : ils sont valides à la comparaison des implémentations spécifique, mais ils déforment les performances TCP obtenues en utilisant des techniques connus non inclus dans le modèle[58].

5.2.3 Comparaison de simulations avec les changements du trafic réseau

Internet à connu des changements dramatiques dans les mélanges de trafic (par exemple, la croissance du web et la croissance possible des flux de données en temps réel). Les validations par rapport à l'ancien trafic peuvent ignorer le trafic actuel et les validations par rapport au trafic actuel peuvent déformer les tendances futures.

5.2.4 Choix des métriques appropriés pour la comparaison

Que ce soit dans le monde réel, dans la spécification ou dans une implémentation particulière, les méthodes de validation doivent définir des métriques pour comparer les résultats du modèle de simulation par

rapport au monde réel. La première étape consiste à comparer des phénomènes attendus dans le protocole. Par exemple, TCP est constitué de plusieurs algorithmes (tel que la fenêtre de transmission, le démarrage lent et la retransmission rapide). Tester ces algorithmes dans la simulation s'apparente à des tests comportementaux d'une implémentation dans le monde réel, et plusieurs de ses mêmes approches peuvent s'appliquer. En outre, les fractions temps/événement, les animations de paquets sont souvent des outils utiles dans ce processus pour trouver des approches générales et quantifier les différences entre les fractions temps/événements similaires et non identiques reste une question de recherche ouverte.
L'agrégation des mesures statistiques telles que les paquets envoyés, le débit et le temps d'achèvement peut donner une image utile des résultats simulés. Les mesures globales doivent être choisis avec soins et utilisées en conjonction avec d'autres approches sachant qu'une métrique mal choisie pourra déformer la comparaison. Par exemple, la comparaison de données moyennes envoyées sur une période de temps ne reflète pas les différences de sporadicité du protocole[58].

5.2.5 Evaluation de la sensibilité des simulations

Une fois la simulation est validée sous un ensemble de conditions, l'analyse de sensibilité permet de comprendre comment différentes configurations changent la précision de la simulation. Par exemple, les variations de la façon dont la retransmission est traitée ne peuvent pas apparaître si la simulation est évaluée uniquement sous des conditions de faible perte. A grande échelle, la simulation réseau présente un défi supplémentaire qui n'est pas abordé par une analyse de sensibilité afin de vérifier qu'un modèle de simulation présente des comportements spécifiés indépendamment des variations de la topologie du réseau, de la taille et des modèles de trafic. Ces comportements sont appelés parfois des variantes du modèle. Les outils qui aident le processus d'analyse de sensibilité sont un domaine de recherche[58].

5.2.6 Evaluation du bilan coût avantages

L'extension et le coût de la validation doivent être pris en considération par rapport aux avantages probables, dans un tel cas, la validation moins coûteuse peut être appropriée. Pourtant, dans des situations spécifiques, il pourrait s'avérer impossible d'atteindre le niveau souhaité de la validation quelque soit son coût. Dans d'autres cas, une validation extensive, tant qu'elle est réalisable, pourrait bien se révéler inutile. En générale, les protocoles les plus stables, pour lesquels les conceptions ne varient pas fréquemment ou de manière significative, permettent une validation plus spécifique.
En fin de compte, il faut tenir compte de la validation dans le contexte de la recherche, et les questions opérationnelles à l'étude. La validation d'une simulation est destinée à prouver à un client que le produit satisfait ses spécifications qu'ils peuvent être exigeantes et plus coûteuses que la validation d'une recherche de simulation en explorant une dizaine de variantes possible du protocole[58].

5.3 Les directives pour une validation réussite

Une fois, les décisions concernant la résolution des questions générales affectant la validation de simulation sont prisent en compte, il est nécessaire de sélectionner par la suite une approche particulière pour valider une simulation spécifique[59].

1. Diverses formes de modèles et d'implémentations peuvent mettre l'accent sur différents aspects d'un système réseau. Pour cette raison les modélisateurs doivent comparer les résultats autant que possible avec des représentations alternatives. Cela pourrait inclure des expériences de labo avec des exercices sur terrain, des modèles analytiques et d'autres simulations développées indépendamment. L'augmentation du nombre de représentations alternatives par rapport auquel le modèle est comparé augmente la probabilité de découvrir les erreurs, les incohérences et les hypothèses non valides.
2. Concevoir avec plusieurs moyens pour examiner l'état de la simulation et l'utilisation des représentations visuelles le maximum possible. Une analyse statistique prudente est certainement utilisée. Le plus souvent, les comportements invalides seront reconnus rapidement à partir de la visualisation de l'animation. Chercher des approches effectives pour examiner et visualiser les larges modèles, (10,000 nœuds ou plus), en particulier pour les petites différences, mais significatifs, demeure un défis pour la recherche. Ces modèles nécessite une instrumentation intégrée à multi étage de filtrage et de classification des données.
3. Lorsque le modèle implique des interactions au cours de temps entre les diverses entités indépendantes, il faut s'assurer d'introduire un asynchronisme en cas de besoin pour imiter le fonctionnement des systèmes réels. Par exemple, chaque station de base sans fil maintient une horloge indépendante. La modélisation de ce comportement est souvent utile de l'effort supplémentaire.
4. Les résultats de simulations doivent être reproductibles. Plusieurs facteurs jouent un rôle important pour approuver la reproductibilité, incluant des algorithmes déterministes pour générer les numéros de séquences pseudo-aléatoire et l'atténuation des erreurs d'arrondissement des représentations en virgule flottante. Les erreurs d'arrondissement peuvent affecter la concurrence des événements, spécialement où la synchronisation optimiste est utilisée quand la simulation est exécutée dans les systèmes d'exploitation parallèles. En générale, des soins doivent être pris en compte pour assurer et veiller à ce que le temps et la cause de l'événement sont modélisés avec précision quand les systèmes de traitements parallèles sont utilisés pour exécuter des simulations.
5. La validation est beaucoup plus facile lorsque le modèle est axé sur des comportements comparatifs plutôt qu'absolue. Ceci est naturel dans plusieurs cas, où une nouvelle proposition est comparée par rapport à un régime existant, déjà déployé.
6. Lorsque la taille de la simulation doit être réduite pour qu'elle s'exécute dans une mémoire et un cycle CPU limités, il faut être prudent pour éviter d'introduire des limites artificielles dans le modèle. Par exemple, les effets transitoires de démarrage ou une topologie physique artificielle peuvent introduire des erreurs.

Une étape importante pour améliorer la qualité de la validation dans la recherche est la reproductibilité des résultats de simulations. Un document qui utilise des études de simulation doit être accompagné d'un lien vers

un modèle accessible au public et bien argumenté (soit le fichier source ou le fichier binaire) afin de permettre une confirmation indépendante des résultats. La disponibilité publique du code source de la simulation et des bibliothèques du modèle de protocole est aussi importante pour permettre une examinassions pour les opérations correctes et pour autoriser la modification pour une utilisation dans des situations additionnelles. Bien que, ces recommandations sont plus importantes pour les développeurs des simulations réseau, ils sont applicables aux utilisateurs de simulation qui doivent évaluer la validité de leurs conclusions. Tout comme un développeur, l'utilisateur doit sélectionner une technique d'analyse (1er point), s'assurer que l'approche n'introduit pas des erreurs additionnelles (les points de 2 à 5), que les résultats sont interprétés de manière appropriée (5ème point), même à des échelles différents (6ème point).

5.4 Echelle et validation

Si la validation des petites simulations semble difficile, la validation des grandes simulations s'avère encore plus difficile. Tenant compte de la portée Internet Aujourd'hui, la compréhension du comportement du protocole avec un grand nombre de nœuds, avec des niveaux de trafic variés et avec plus ou moins de détails reste des questions importantes. Une autre dimension de l'échelle est le nombre de composants développés indépendamment dans un modèle. Dans ce qui suit, on va voir comment ces deux types d'échelles affectent la validation.

5.4.1 Mise à l'échelle pour un grand nombre de nœuds

Deux approches complémentaires de simulation à grande échelle en l'occurrence : l'exécution en parallèle et l'abstraction. Plusieurs simulateurs supportent le parallélisme[60], [61]. L'utilisation d'une machine avec plusieurs CPU ou clusters de stations de travail apporte plus de puissance et de mémoire pour un problème donné, permettant d'élargir de 10 fois jusqu'à 100 fois les simulations. Une approche complémentaire est l'utilisation de l'abstraction pour factoriser les détails non important à la simulation manuelle[62]. L'abstraction a été utilisée pour fournir une augmentation de 100 fois jusqu'à 1000 fois dans les tailles possibles de la simulation pour des questions d'une recherche particulière. Ceci veut dire que l'abstraction doit être appliquée soigneusement parce que dans l'absence d'une dérivation mathématique explicite, un modèle abstrait doit être encore validé par rapport à un modèle plus détaillé fonctionnant à une vitesse plus lente ou par rapport à des expériences de terrain à une échelle suffisamment grande. En outre, des nouveaux phénomènes peuvent s'émerger à partir des interactions avec l'augmentation de la taille du réseau.

5.4.2 Mise à l'échelle avec des éléments de modèle hétérogènes

Les simulations à grande échelle peuvent s'appuyer sur des sous-modèles à petite échelle validés. Une approche est la composition récursive : on commence avec des composants bien validés, puis on génère des larges modèles en utilisant la composition hiérarchique. Une autre approche est de comparer les simulations à petite échelle détaillées et abstraites avec des simulations à grande échelle ; puis on génère les larges scénarios

abstraits. L'abstraction et la construction supposent que les inexactitudes possibles dans les scénarios à petite échelle ne sont pas amplifiées à grande échelle. Cette hypothèse doit être encore validée cas par cas.

5.5 Validation de la solution TCP-MANet

Actuellement les réseaux sans fil ad hoc sont des réseaux en cours de développement et la majorité des protocoles proposés ne sont pas standardisés, donc la comparaison des résultats de simulation avec la réalité reste une question clé dans la recherche. En conséquence, on se limite à comparer les résultats de simulations de la solution proposée TCP-MANet avec les résultats de simulation du TCP standard. Pour répondre à la première question de la validation de notre solution, les simulations de la solution TCP-MANet ont été comparées par rapport à l'implémentation du protocole TCP standard dans le simulateur ns-2.
Toutes les simulations de la solution TCP-MANet sont validées par rapport au TCP standard, donc l'utilisation d'une autre implémentation du protocole TCP (tel que TCP New Reno) affecte les résultats du protocole TCP standard, parce que notre solution est une classe qui hérite les fonctions de la classe TCP standard (Deuxième question). Le trafic utilisé dans la simulation est un trafic aléatoire généré par l'expéditeur TCP, les paquets TCP sont ajustés à une taille fixe de 512 bits, l'évolution du trafic n'affecte pas les performances de la solution (troisième question). Pour quantifier les performances de notre solution TCP-MANet par rapport au TCP standard on a mesuré le débit (rapport bit reçus par seconde), la perte et la taille de la fenêtre de congestion (quatrième question).
Pour évaluer la sensibilité de la simulation (cinquième question), on a utilisé différents modèles de simulation, chaque modèle cible une caractéristique spécifique. Les caractéristiques qu'on a utilisées dans la simulation sont :

- La mobilité des nœuds du réseau sans fil ad hoc.
- Le nombre de nœuds mobile (la densité) dans le réseau sans fil ad hoc.
- Les dimensions (topologie) du réseau sans fil ad hoc.
- L'énergie consommée par les nœuds mobiles.

La validation de notre solution à été réalisée avec le simulateur ns-2 et elle a été basée sur une comparaison des résultats obtenues par simulation, donc on n'a pas réalisé des expériences dans le monde réel qui nécessite un équipement réseau coûteux.

5.5.1 Les modèles de simulations

Toutes les simulations ont été basées sur la configuration réseau suivante :

Paramètre	Valeur
Protocole de routage	AODV
Type d'accès au media (MAC)	802.11
Type d'interface de la file d'attente	File d'attente avec priorité
Temps d'arrêt de la simulation	1200 secondes
Nombre maximale de paquets dans la file d'attente	50 paquets
L'agent de transport	TCP standard et TCP-MANet
Limite maximale de la taille de la fenêtre	1000
Récepteur TCP (puits)	Acquittements avec délai
L'agent d'application	FTP
Taille du paquet	512 bits
Taux de transmission	81920 bits/seconde
L'intervalle entre les paquets	0.4 milliseconde
Nombre maximale de paquets à envoyés	10 paquets

TABLE 5.1 – Configuration des modèles de simulations

Les sections qui suivent décrivent chaque caractéristique et leurs modèles de simulation.

La mobilité dans le réseau

Ces modèles de simulation étudient l'impact de la mobilité sur les performances de notre solution. Pour réaliser cette étude on a utilisé deux modèles : la mobilité faible et la mobilité forte. Ces deux modèles sont constitués de 15 nœuds distribués aléatoirement sur une grille à deux dimensions. La topologie utilisée est 2200m * 1200m. La communication concerne le nœud (0) et le nœud (4) sélectionnés à partir des nœuds du réseau. Trois vitesses différentes respectivement : 2m/s, 3m/s et 5m/s avec 24 mouvements de nœuds dans le modèle à faible mobilité et une vitesse de 5m/s avec 86 mouvements de nœuds dans le modèle à forte mobilité.

Le nombre de nœuds du réseau

Ces modèles de simulation étudient l'impact de la densité (nombre de nœuds dans le réseau) sur les performances de notre solution. Pour réaliser cette étude, on a utilisé quatre modèles constitués de différents nombre de nœuds respectivement 10, 50, 100 et 150 nœuds. Les nœuds sont distribués aléatoirement sur une grille à deux dimensions et la topologie utilisée est de 2200m * 1200m. La communication est établie entre le nœud (0) et le nœud (4) sélectionnés à partir des nœuds du réseau. On a considéré des mouvements aléatoires pour les nœuds du réseau.

La topologie du réseau

Ces modèles de simulation étudient l'impact de la topologie du réseau sur les performances de notre solution. Pour réaliser cette étude on a utilisé quatre modèles composé de 35 nœuds distribués aléatoirement sur une grille à deux dimensions avec différentes topologies. Le premier modèle est 500m * 500m, le deuxième modèle est 1500m * 1500m, le troisième modèle 3000m * 3000m et le quatrième modèle est 6000m * 6000m. La communication a été réalisée entre le nœud (0) et le nœud (2) sélectionnés à partir des 35 nœuds du réseau. On a considéré des mouvements aléatoires pour les nœuds du réseau.

L'énergie des nœuds mobiles

Ces modèles de simulation évaluent la consommation de l'énergie des nœuds du réseau en utilisant notre solution. Pour réaliser cette étude on a utilisé trois modèles constitués de 5 nœuds distribués aléatoirement sur une grille à deux dimensions avec une topologie de 2200m * 950m. La communication a été réalisée entre le nœud (0) et le nœud (2) sélectionnés à partir des 5 nœuds du réseau. On a considéré des mouvements aléatoires pour les nœuds du réseau. Le premier modèle attribue pour chaque nœud mobile une énergie initiale de 50 Joules avec une puissance de transmission de 0,8W et une puissance de réception de 0,4W, le deuxième modèle attribue pour chaque nœud mobile une énergie initiale de 100 Joules avec une puissance de transmission de 0.5W et une puissance de réception de 0,5W et le troisième modèle attribue pour chaque nœud mobile une énergie initiale de 100 Joules avec une puissance de transmission de 0,8W et une puissance de réception de 0,4W.

5.5.2 Evaluation des performances

Deux scénarios de simulation ont été effectués pour chaque modèle, chacun porte sur l'analyse des performances des deux versions du protocole TCP à savoir TCP Standard et TCP-MANet (la solution proposée) dans différentes situations. Dans le paragraphe suivant, on va présenter les résultats de simulation sous forme de courbes à diverses contraintes (la mobilité des nœuds, le nombre de nœuds mis en jeu dans une simulation, la topologie du réseau, et la gestion de l'énergie sur le comportement des deux protocoles).

5.5.3 La mobilité dans le réseau

La figure 5.1 schématise deux courbes dans un environnement ad hoc avec faible mobilité. La courbe de gauche illustre le débit et la deuxième représente le comportement de la taille de la fenêtre de congestion du TCP traditionnel avec TCP-MANet. On constate que la solution proposée améliore le débit, d'où une dégradation du débit dans l'intervalle de temps de 500s jusqu'à 600s lorsqu'on utilise le TCP traditionnel et une stabilité du débit lorsqu'on utilise le TCP-MANet.

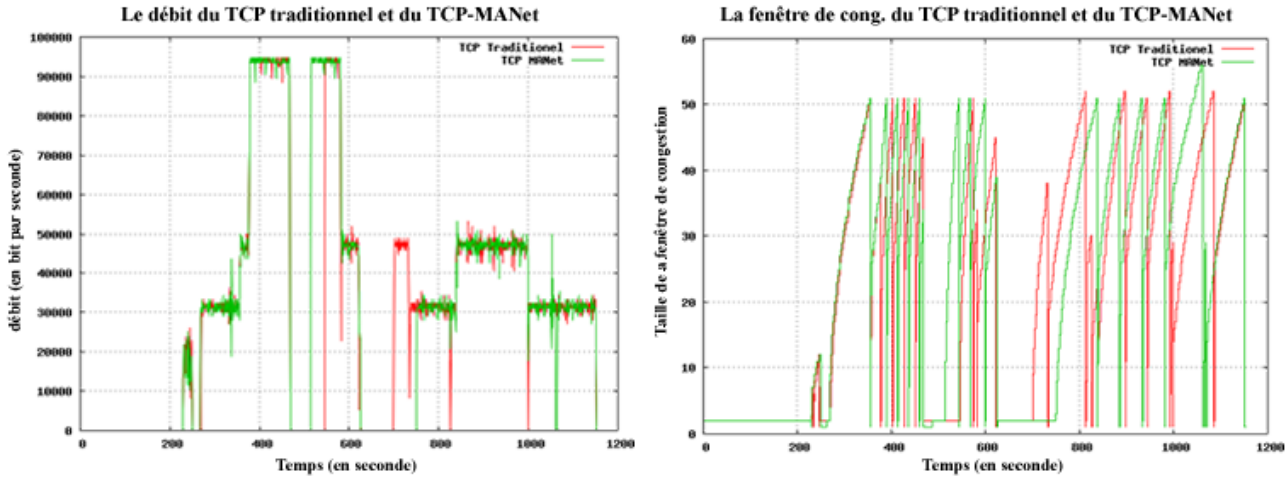

FIGURE 5.1 – La courbe du débit et de la fenêtre de congestion dans une faible mobilité

Pour la fenêtre de congestion, on constate que la solution proposée fait nettement la différence entre la perte des paquets due à la congestion du réseau et aux erreurs de transmission, l'utilisation du message LFN permet de notifier ce genre d'erreurs sans les considérer comme une congestion du réseau, ce mécanisme de contrôle de congestion permet une croissance exponentielle de la fenêtre de congestion dans TCP-MANet, ce qui n'est pas le cas pour TCP traditionnel où la notion d'erreur de transmission et la congestion n'est pas claire, cette confusion entre les deux cas de pertes entrainera un démarrage lent dans les intervalles de temps (550s-600s et 800s-850) et dégrade le débit dans TCP traditionnel.
La figure 5.2 présente deux courbes concernant les paquets reçus et perdus pour les deux protocoles en fonction du temps de simulation. On constate

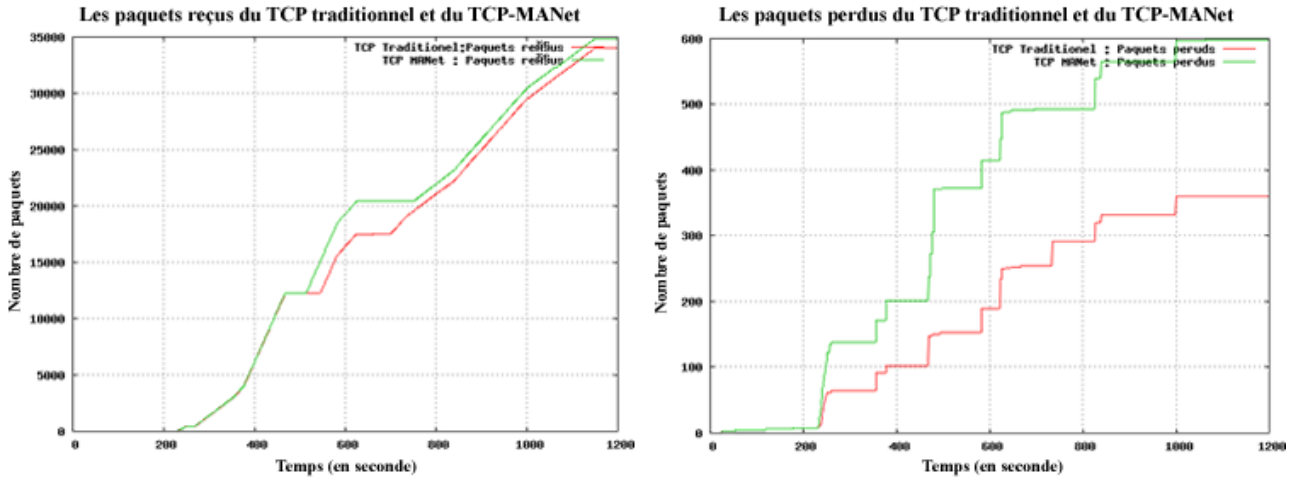

FIGURE 5.2 – La courbe des paquets reçs et les paquets perdus dans une faible mobilité

La première courbe illustre une comparaison des paquets reçus par l'expéditeur TCP dans les deux protocoles (TCP traditionnel et TCP-MANet). On constate une différence considérable de paquets reçus entre les deux versions simulés, et par conséquent une augmentation de la perte des paquets dans la solution proposée (TCP-MANet). Cette augmentation est due à l'utilisation de la fenêtre de congestion d'une manière efficace sans entrer dans un démarrage lent même dans des situations où la perte de paquets causée par des erreurs

de transmission dure long temps. Dans une telle situation, TCP-MANet ne fait pas diminuer la fenêtre de congestion et continue à transmettre les paquets suivant la dernière valeur de la fenêtre de congestion. Si le nouveau chemin rétabli est erroné, TCP-MANet produit une perte élevée par rapport au TCP traditionnel.

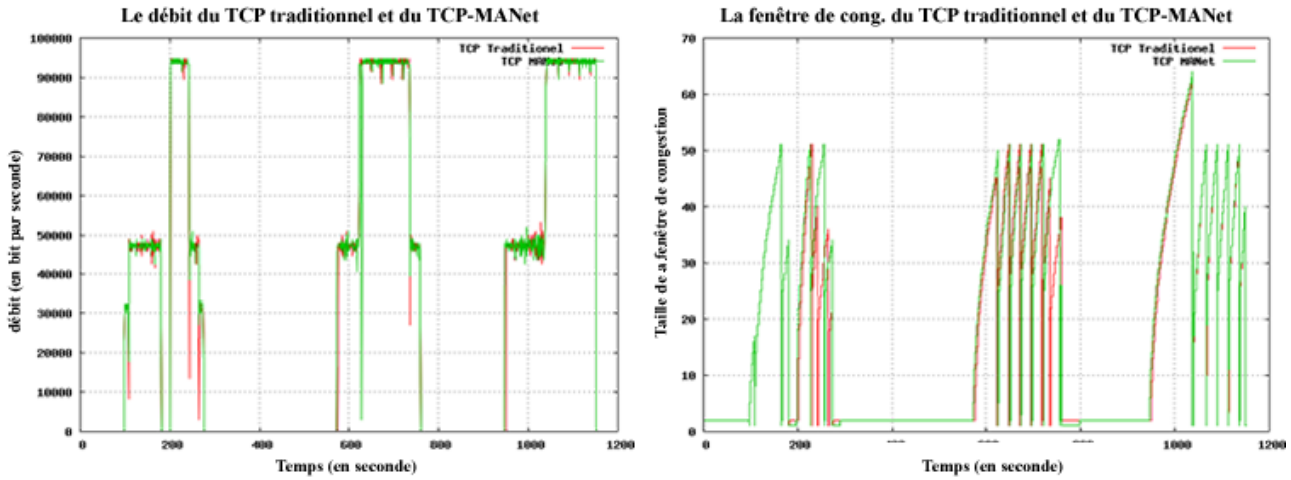

FIGURE 5.3 – La courbe du débit et de la fenêtre de congestion dans une forte mobilité

La figure 5.3 illustre aussi deux courbes pour le débit et la taille de la fenêtre de congestion (idem à figure 5.1), mais dans un environnement ad hoc avec forte mobilité. On constate un apport important du débit dans la solution proposée grâce au changement du mécanisme de contrôle de congestion ; pour une dégradation dans l'intervalle de temps allant de 200s jusqu'à 280s dans TCP traditionnel alors qu'une stabilité du débit dans TCP-MANet est observée.
Pour la deuxième courbe on constate que la fenêtre de congestion du TCP-MANet agit de la même manière que TCP traditionnel sauf dans quelques situations où la fenêtre de congestion du TCP traditionnel recommence un démarrage lent surtout dans les intervalles de temps (200s-280s et 750s-780) alors que dans le TCP-MANet le mécanisme de croissance exponentiel est maintenu. Dans un tel environnement, la seule remarque qu'on peut observer est que les performances du TCP-MANet sont presque identiques aux performances du TCP traditionnel.

La figure 5.4 illustre aussi deux courbes associées au nombre de paquets reçus et perdus (idem à figure 5.2), mais dans un environnement ad hoc avec forte mobilité. La aussi on constate une différence considérable des paquets reçus entre les deux versions de TCP (TCP traditionnel et TCP-MANet), en conséquence, une augmentation de la perte des paquets dans notre solution (TCP-MANet). Cette augmentation s'explique de la même manière que pour un cas de faible mobilité.
A base de ces résultats, on peut conclure, que la mobilité des nœuds a un impact sur les performances du TCP-MANet.

Le nombre de nœuds du réseau

Dans la figure 5.5, deux courbes sont présentées. Celle de gauche schématise le débit et celle de droite décrit le comportement de la fenêtre de congestion dans un réseau ad hoc composé de 10 nœuds. On constate

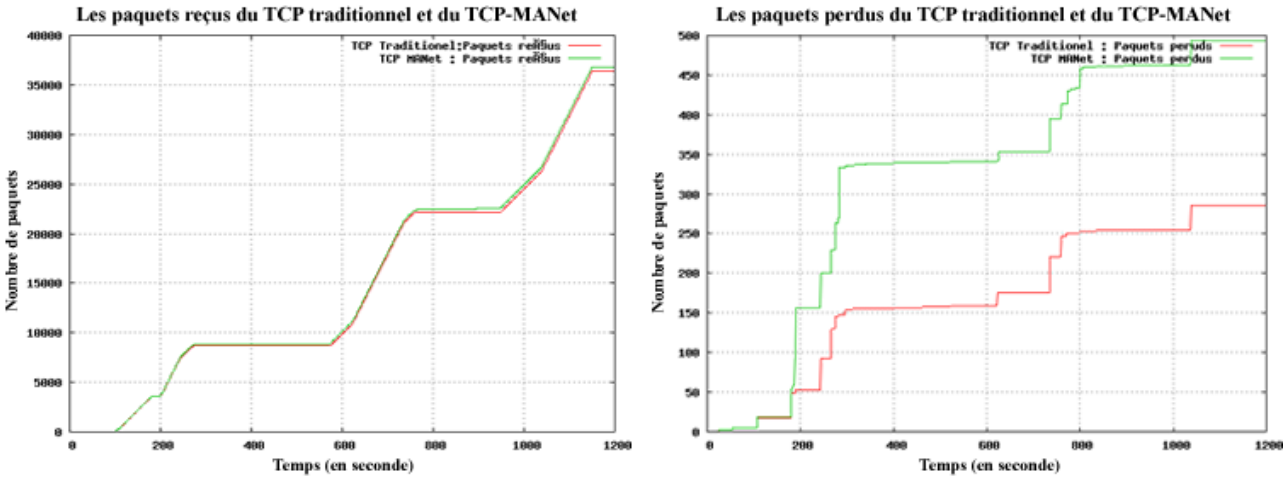

FIGURE 5.4 – La courbe des paquets reçus et les paquets perdus dans une forte mobilité

aussi un apport en terme de débit concernant la solution proposée et à l'opposé, une dégradation du débit surtout dans l'intervalle de temps de 300s jusqu'à 700s en utilisant le TCP traditionnel. Le TCP-MANet affiche une mauvaise position devant le TCP standard dans l'intervalle de temps (700s jusqu'à 900s), la cause est due à l'échec d'établissement d'un nouveau chemin après la détection d'un chemin erroné au niveau d'un nœud intermédiaire.

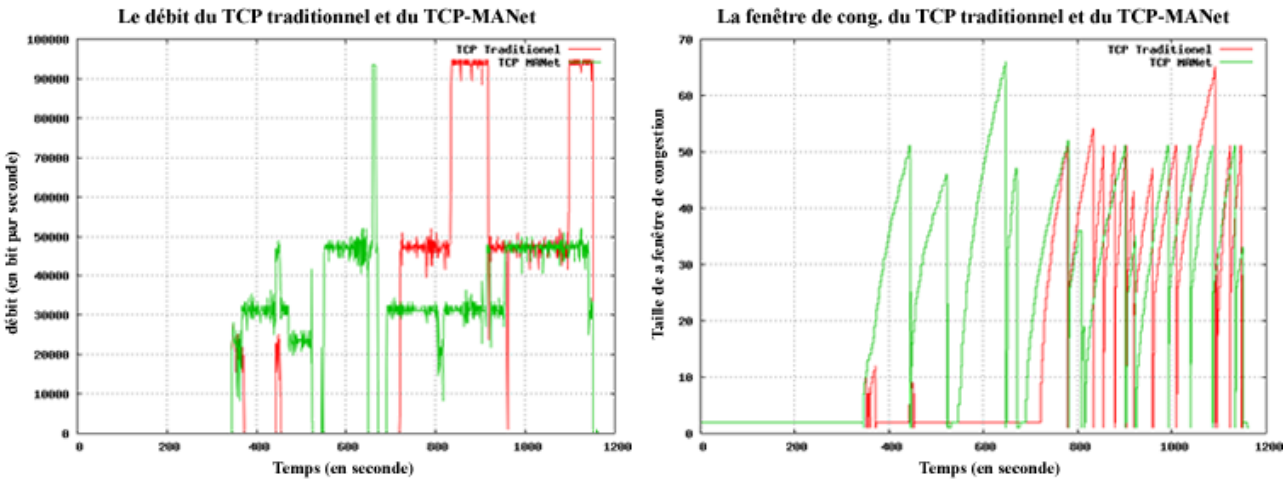

FIGURE 5.5 – La courbe du débit et de la fenêtre de congestion dans une réseau ad hoc de 10 nœuds

Le comportement de la fenêtre de congestion continue dans la croissance exponentielle pour TCP-MANet, alors que dans TCP traditionnel, elle subie des dégradations fréquentes à cause des mouvements des nœuds du réseau et le changement de la topologie (300s-700s). Dans ces intervalles de temps, le TCP traditionnel exécute un démarrage lent en considérant la perte des paquets due à une erreur de transmission comme une congestion.

Dans la figure 5.6, deux courbes sont présentées. Celle de gauche schématise le nombre des paquets reçus et celle de droite les paquets perdus en fonction du temps dans réseau ad hoc composé de 10 nœuds. On observe une différence considérable des paquets reçus entre les deux versions de TCP (TCP traditionnel et

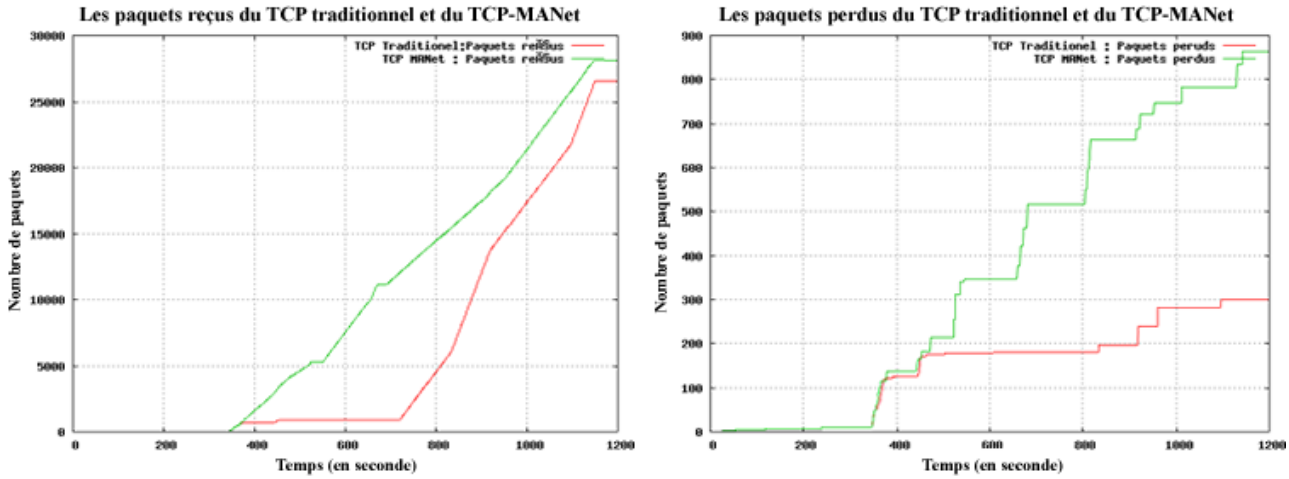

FIGURE 5.6 – La courbe des paquets reçs et les paquets perdus dans un réseau ad hoc de 10 nœuds

TCP-MANet) alors que la perte des paquets est importante dans notre solution (TCP-MANet). Cette augmentation est due à l'utilisation de la fenêtre de congestion d'une manière efficace sans entrer dans un démarrage lent dans des situations où la perte des paquets est causée par des erreurs de transmission. Dans une telle situation, TCP-MANet ne démunie pas la fenêtre de congestion et continu à transmettre les paquets suivant la dernière valeur de la fenêtre de congestion même en présence de chemin erroné, ce qui cause une perte élevée dans TCP-MANet par rapport au TCP traditionnel.

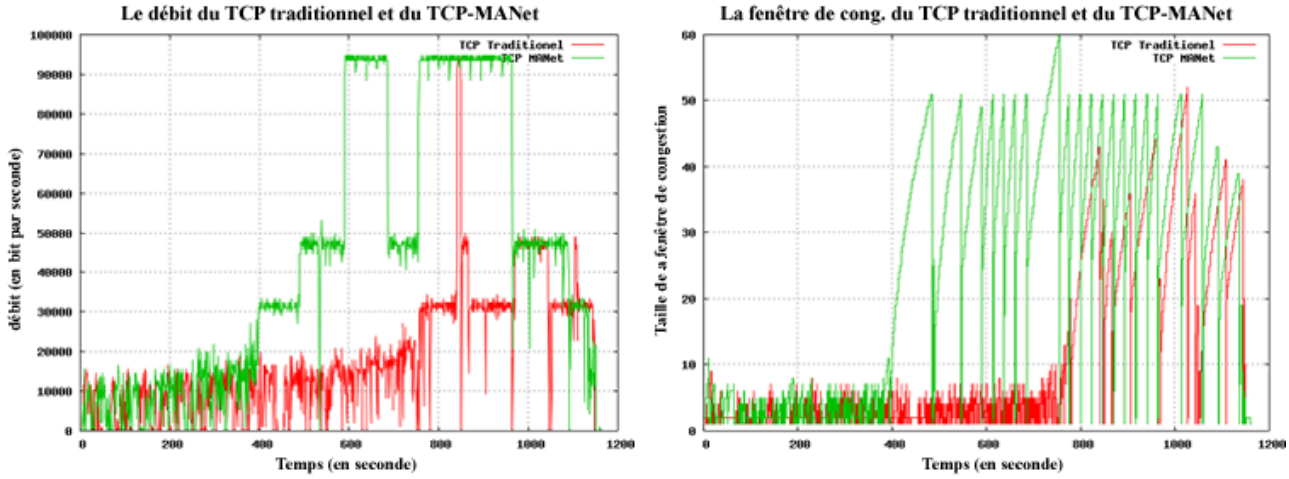

FIGURE 5.7 – La courbe du débit et de la fenêtre de congestion dans une réseau ad hoc de 100 nœuds

La figure 5.7 présente deux courbes dans réseau ad hoc composé de 100 nœuds. Cette figure montre un meilleur débit pour la solution proposée. On constate une dégradation du débit dans l'intervalle de temps allant de 250s jusqu'à 1100s en utilisant le TCP traditionnel. Pour la solution proposée (TCP-MANet), on constate un meilleur débit. La plupart des temps, TCP-MANet gère la bande passante du réseau d'une manière efficace. Dans la deuxième courbe, on observe que la fenêtre de congestion du TCP-MANet continue sa progression exponentielle, alors que dans le TCP traditionnel la fenêtre subie des dégradations fréquentes à cause des mouvements des nœuds du réseau et le changement de la topologie (400s-1100s). Dans ces intervalles de temps, le TCP traditionnel exécute un démarrage lent en considérant la perte des paquets due à

une erreur de transmission comme une congestion, mais dans TCP-MANet on observe que l'utilisation de la fenêtre de congestion est efficace.

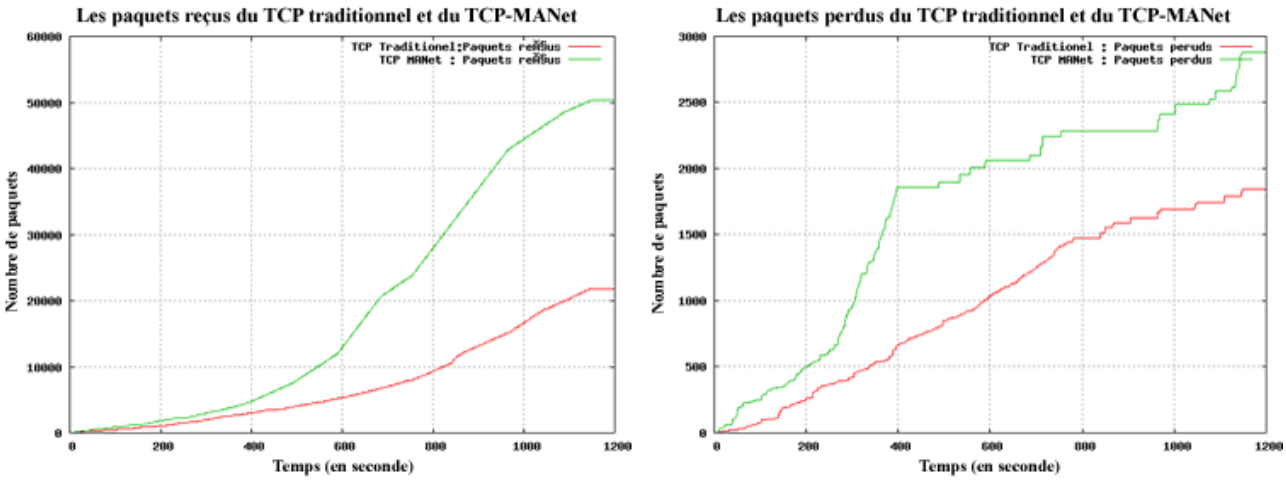

FIGURE 5.8 – La courbe des paquets reçus et les paquets perdus dans un réseau ad hoc de 100 nœuds

La figure 5.8 présente aussi deux courbes dans un réseau composé de 100 nœuds. On observe que dans TCP-MANet le récepteur TCP a reçu plus que le double des paquets reçus par l'expéditeur TCP en utilisant le TCP traditionnel, en conséquence, une augmentation de la perte des paquets dans notre solution (TCP-MANet). Cette augmentation s'explique de la même manière que celle de la figure 5.6.

La topologie du réseau

La figure 5.9 présente deux courbes dans réseau ad hoc de dimension 500m*500m. On constate toujours un apport du débit dans notre solution avec une dégradation du paramètre débit dans l'intervalle de temps allant de 600s jusqu'à 1100s dans TCP traditionnel, et une stabilité pour TCP-MANet.

Dans la deuxième courbe, on observe que la fenêtre de congestion du TCP-MANet croit toujours en fonction exponentielle, alors que dans le TCP traditionnel la fenêtre subie des dégradations fréquentes à cause des mouvements des nœuds du réseau et le changement de la topologie (600s-1100s). Dans ces intervalles de temps, le TCP traditionnel exécute un démarrage lent en considérant la perte des paquets due à une erreur de transmission comme une congestion.

La figure 5.10 présente aussi deux courbes dans un réseau de dimension 500m*500m. On observe une différence considérable des paquets reçus entre les deux versions de TCP (TCP traditionnel et TCP-MANet), en conséquence, une augmentation de la perte des paquets dans notre solution (TCP-MANet). Cette augmentation est due à l'utilisation de la fenêtre de congestion d'une manière efficace sans qu'un démarrage lent soit initié dans les situations où la perte des paquets est causée par des erreurs de transmission. Dans une tel situation, TCP-MANet ne démunie pas la fenêtre de congestion et continu à transmettre les paquets suivant la dernière valeur de la fenêtre de congestion même en présence d'un nouveau chemin erroné, ce qui cause

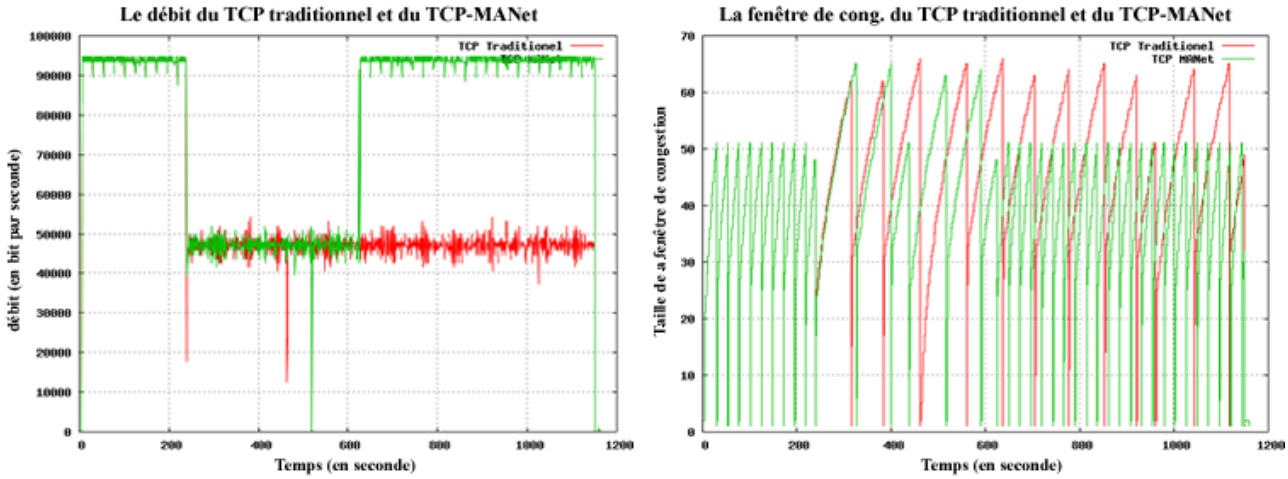

FIGURE 5.9 – La courbe du débit et de la fenêtre de congestion dans une réseau ad hoc de topologie 500m*500m

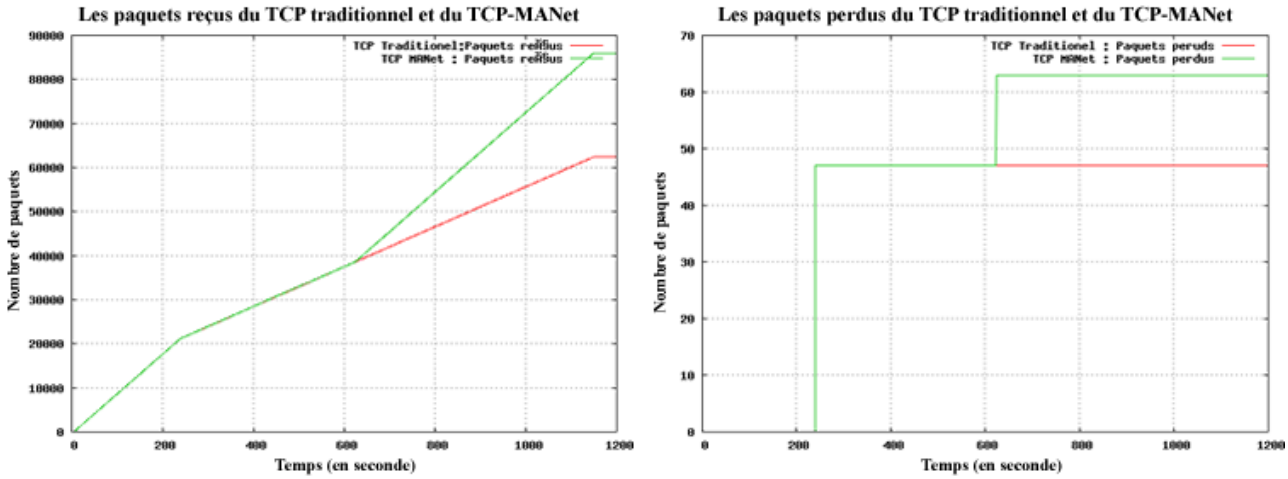

FIGURE 5.10 – La courbe des paquets reçus et les paquets perdus dans un réseau ad hoc de topologie 500m*500m

une perte élevée dans TCP-MANet par rapport au TCP traditionnel.

La figure 5.11 présente deux courbes dans un réseau ad hoc de dimension 3000m*3000m. Cette figure présente aussi un apport du débit dans notre solution. Donc on constate une dégradation du débit dans l'intervalle de temps de 750s jusqu'à 1100s en utilisant le TCP traditionnel alors qu'une stabilité du débit dans TCP-MANet. Dans la deuxième courbe on constate que la fenêtre de congestion du TCP-MANet continue dans la croissance exponentielle, alors que dans le TCP traditionnel la fenêtre subie des dégradations fréquentes à cause des mouvements des nœuds du réseau et le changement de la topologie (750s-1100s). Dans ces intervalles de temps, le TCP traditionnel exécute un démarrage lent en considérant la perte des paquets due à une erreur de transmission comme une congestion.

La figure 5.12 présente aussi deux courbes dans un réseau ayant une dimension de 500m*500m. On constate que dans TCP-MANet le récepteur TCP a reçu plus que le double des paquets reçus par l'expéditeur TCP en utilisant le TCP traditionnel, mais cette fois si notre solution a affiché un gain dans la perte des

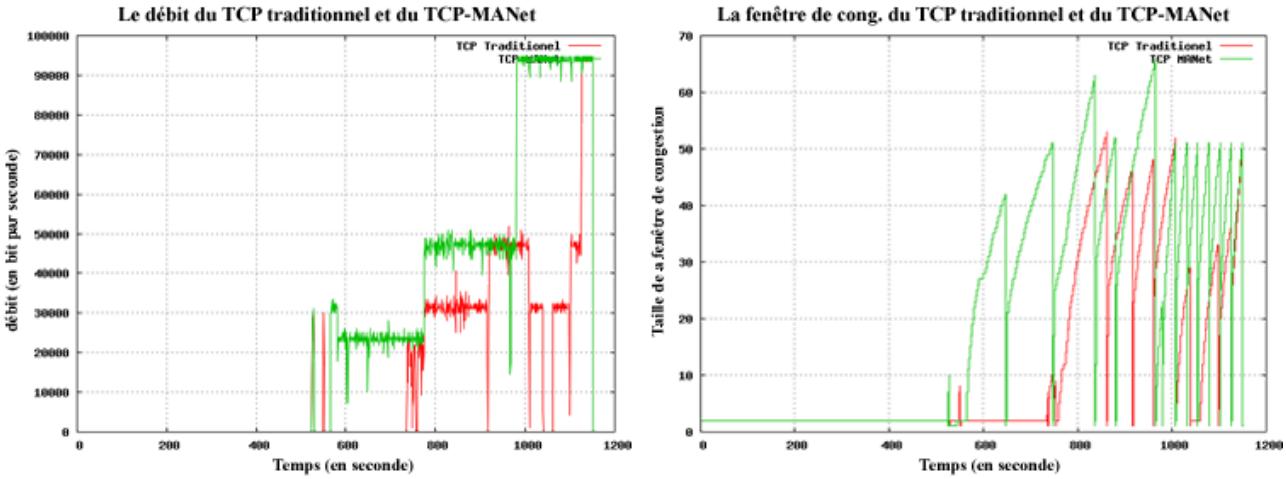

FIGURE 5.11 – La courbe du débit et de la fenêtre de congestion dans une réseau ad hoc de topologie 3000m*3000m

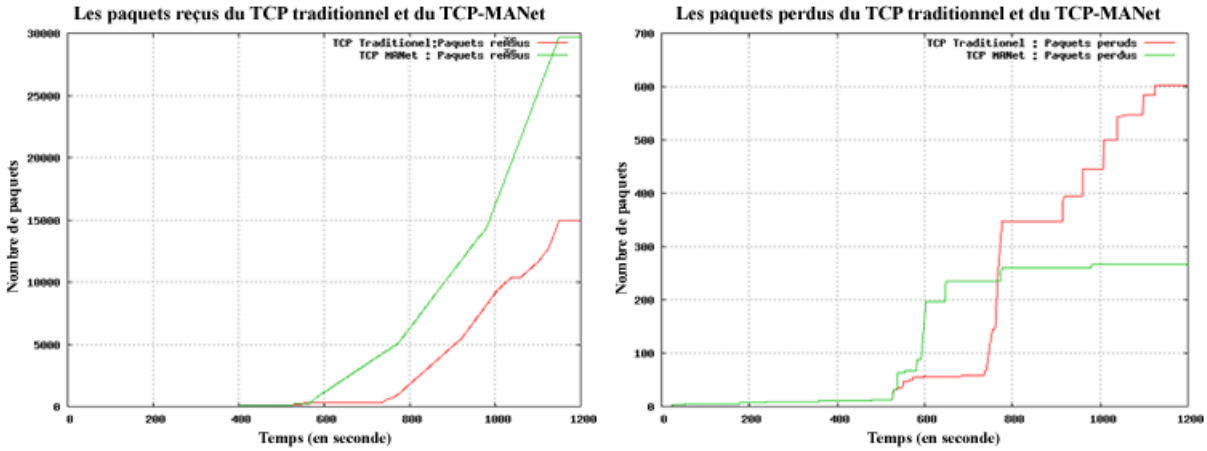

FIGURE 5.12 – La courbe des paquets reçus et les paquets perdus dans un réseau ad hoc de topologie 3000m*3000m

paquets par rapport au TCP traditionnel. La diminution de la perte des paquets est réalisée par le mécanisme de retransmission des paquets mis en tampon dans le nœud intermédiaire, ce qui permettra à l'expéditeur TCP de reprendre la transmission après l'établissement d'un nouveau chemin à partir du dernier paquet transmis.

L'énergie des nœuds mobile

La figure 5.13 illustre une courbe qui représente le taux de l'énergie des nœuds expéditeur, récepteur et intermédiaires au cours du temps. L'énergie initiale de chaque nœud est de 50 joules, la puissance de transmission est 0,8W et la puissance de réception est 0,4W. On constate que l'utilisation du TCP traditionnel consomme plus d'énergie que TCP-MANet. A titre d'exemple au niveau du nœud expéditeur, on remarque une dégradation de l'énergie à partir de la 580ème seconde dans les deux solutions (TCP traditionnel et TCP-MANet), mais dans TCP traditionnel l'énergie du nœud expéditeur chute dans la 730ème seconde, cependant dans TCP-MANet l'énergie du nœud expéditeur chute dans la 830ème seconde. La dégradation de l'énergie dans TCP traditionnel est causée par les paquets supplémentaires transmis par l'expéditeur pendant la répa-

ration local du chemin vers la destination effectué par le nœud intermédiaire. Dans TCP-MANet, lorsqu'un lien est erroné la connexion TCP est gelée.

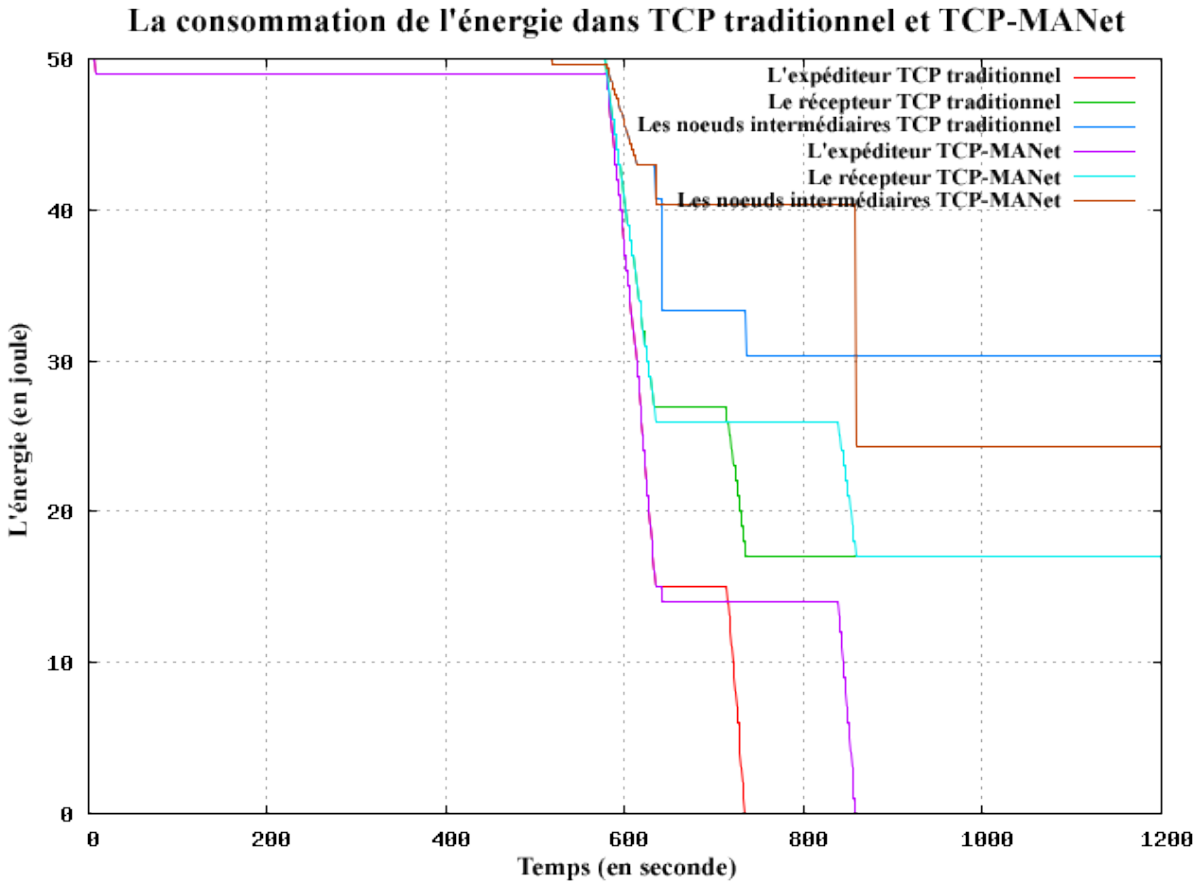

FIGURE 5.13 – La courbe des de la consommation de l'energie des nœuds mobiles avec un energie initial de 50 joules

Au niveau du nœud récepteur on constate que TCP-MANet consomme moins d'énergie par rapport au TCP traditionnel dans l'intervalle allant de 730s jusqu'à 830s, et au-delà de ce temps, la consommation de l'énergie est la même dans les deux solutions, malgré que le récepteur du TCP-MANet à reçus plus de paquets supplémentaires par rapport au récepteur du TCP traditionnel. Cette consommation est obtenue grâce au mécanisme de contrôle de congestion utilisé dans TCP-MANet qui gèle la connexion TCP lorsqu'un lien erroné est détecté et aucune retransmission de paquets n'est initiée jusqu'à ce qu'un nouveau chemin vers la destination soit établi.
Au niveau des nœuds intermédiaires on observe que TCP-MANet consomme moins d'énergie par rapport au TCP traditionnel dans l'intervalle allant de 630s jusqu'à 830s, puis la consommation de l'énergie se dégrade dans TCP-MANet après la 830ème seconde. Cette consommation supplémentaire au niveau des nœuds intermédiaires est causée par le mécanisme de retransmission des paquets mis en tampon des nœuds intermédiaires. Cette retransmission nécessite une puissance de transmission supplémentaire au niveau des nœuds intermédiaires par rapport au TCP traditionnel, ce qui cause une consommation considérable d'énergie au niveau des nœuds intermédiaires dans TCP-MANet.

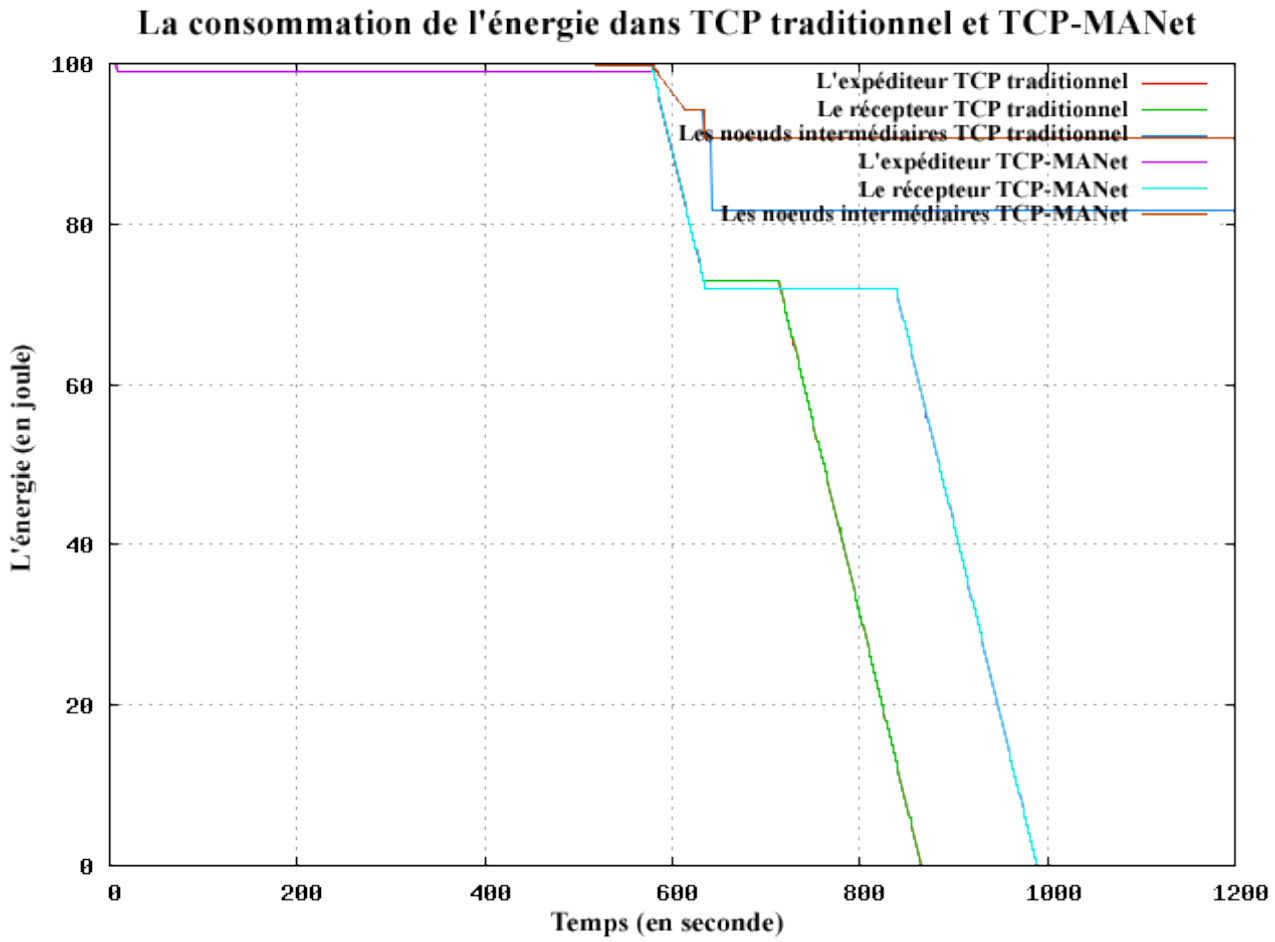

FIGURE 5.14 – La courbe des de la consommation de l'energie des nœuds mobiles avec un energie initial de 100 joules

La figure 5.14 illustre une autre courbe qui représente le taux de l'énergie des nœuds expéditeur, récepteur et intermédiaires au cours du temps, mais cette fois ci avec une énergie initiale de 100 joules pour chaque nœud avec une puissance de transmission et de réception de 0,5W. On observe que la consommation de l'énergie au niveau du nœud expéditeur et récepteur est la même que ce soit dans TCP traditionnel ou TCP-MANet, puisque la puissance d'émission est égale à la puissance de réception. Au niveau des nœuds intermédiaires on observe que le TCP traditionnel consomme plus d'énergie par rapport au TCP-MANet.
A base de ces résultats, on conclut que TCP-MANet gère l'énergie d'une manière efficace lorsqu'on utilise une puissance d'émission identique à la puissance de réception.

5.6 Conclusion

Dans ce chapitre on a présenté les résultats des différents scénarios de simulation pour valider la solution proposée (TCP-MANet). Pour avoir une validation réussie, on doit suivre certaines directives. On a commencé par exposer les questions clés à considérer pour réaliser cette validation, puis on a décrit six directives nous aidant à réussir une validation. La simulation de la solution TCP-MANet à été exécuté sur un ensemble

de modèles de simulation. Chaque modèle de simulation vise à faire apparaître l'effet d'une certaine caractéristique des réseaux ad hoc (tel que la mobilité, le nombre de nœuds, la topologie, et l'énergie). Dans la majorité des cas, les résultats obtenus montrent que la solution proposée offre de meilleures performances par rapport au TCP traditionnel grâce aux changements du mécanisme de contrôles de congestion. Dans nos simulations, on a essayé de créer quelques situations où la perte des paquets est due aux erreurs de transmission pour voir comment notre solution doit distinguer entre la perte et la congestion.

Conclusion générale

Dans ce mémoire qui s'articule sur deux volets, dont le premier consistait à faire un bilan sur les travaux concernant le protocole TCP dans les réseaux ad hoc afin d'identifier leurs carences, et le second, de proposer une nouvelle variante pour le protocole TCP ; Nous avons proposé une solution qui tend à améliorer les performances de TCP dont les mesures ont été effectuées par simulation sous Network Simulator.

Notre proposition est basée sur les interactions avec les couches inférieures pour remédier à la dégradation des performances du TCP dans les réseaux ad hoc. Le changement fréquent de la topologie dû à la mobilité des nœuds, et l'incertitude des liens sans fil ont incité les chercheurs et les développeurs des réseaux ad hoc à faire face à ces nouveaux défis pour mieux adapter le protocole TCP dans cette catégorie de réseau (i.e. ad hoc). L'incapacité du protocole TCP traditionnel à distinguer entre les erreurs de transmission causées par des liens erronés, et ceux causées par la congestion du réseau influent sensiblement sur les performances du protocole TCP traditionnel dans les réseaux ad hoc. Ce qui mène vers une mauvaise utilisation de la bande passante, et une exécution d'un contrôle de congestion dans des situations inappropriées.

Ces dernières années, une multitude de solutions ont été proposées, où chacune d'elles tente à améliorer les performances du protocole TCP dans les réseaux ad hoc. Toutes ces solutions ont un mécanisme commun pour distinguer entre la perte des paquets provoqués par la rupture des liens, et celle due la congestion du réseau. Elles utilisent la réaction avec la couche réseau pour notifier les ruptures des chemins, mais elles ne répondent pas parfaitement aux contraintes imposées par les réseaux sans fil. Pour remédier aux insuffisances de ces solutions, on a proposé une nouvelle variante qu'on a appelé TCP-MANet.

TCP-MANet est une amélioration de la solution TCP-BuS qui repose sur le même principe de distinction entre les causes de la perte de paquets. Elle utilise des tampons pour conserver les paquets qui doivent être acheminés une fois le chemin vers la destination est rétabli. Dans TCP-MANet, on se base sur la couche MAC pour notifier les liens erronés, ce qui permet de geler rapidement la connexion TCP avant que l'expéditeur transmet des paquets supplémentaires vers la destination. Cette solution utilise seulement un tampon unique au niveau du nœud intermédiaire ayant détecté un lien erroné pour sauvegarder les paquets à acheminer vers la destination. Cette technique apporte un gain considérable en termes de place mémoire, et de gestion d'énergie.

La validation de la solution dans un réseau ad hoc réel est une question clé de la recherche. Afin de contourner cette réalité, on a procédé par simulation. Actuellement, une grande variété de simulateurs existe pour simuler les réseaux informatiques. Pour valider notre solution on à utilisé le simulateur NS-2, puisqu'il produit des résultats plus proches à la réalité et il supporte l'extension de nouveaux modèles de protocole.

En se basant sur un ensemble de directives, la validation de la solution TCP-MANet a montré une nette amélioration des performances de ce dernier, en comparaison avec le protocole TCP traditionnel disponible sur le simulateur ns-2. La série de tests effectués sur plusieurs modèles de simulation a prouvé que TCP-

MANet répond parfaitement aux phénomènes de congestion du réseau et distingue sans aucune ambigüité entre les deux causes de pertes de paquets, et gère d'une manière efficace l'énergie.
Notre solution a permis d'ouvrir de nouvelles perspectives, et spécialement l'utilisation d'autres paramètres tels que la puissance du signal. Ces perspective nous orientent a pensé autrement, et à concevoir de nouvelles solutions pour mieux adapter le protocole TCP dans l'environnement mobile ad hoc afin de contrôler la congestion dans les situations appropriées et gérer l'énergie d'une manière efficace.
Comme autre perspective à notre travail, il est utile de s'intéresser aux calcul de paramètres tels que le temps d'aller-retour (RTT), le temps d'expiration de la découverte du chemin ...etc., l'utilisation de ces paramètres d'une manière efficace permettra surement de minimiser l'utilisation des messages de contrôle (tel que REN et RDN) et aussi d'utiliser la bande passante d'une manière efficace, ce qui contribuera forcement à améliorer les performances du protocole TCP dans les réseaux mobiles ad hoc.

Bibliographie

[1] S. G. STEFANO BASAGNI, MARCO CONTI and I. STOJMENOVIC, *Mobile ad hoc networking*. A JOHN WILEY & SONS, INC., PUBLICATION, 2004.

[2] G. B. C. P. Subir, Kumar Sarkar ; T, *Ad hoc Mobile Wireless Networks*. Auerbach Publications, 2007.

[3] E. H. P. Nabil and DAHBI, "Etude de la technologie wimax mobile," tech. rep., Telecom SudParis, Paris, 2010.

[4] S. Mian, "Wimax ou l'évolution des réseaux sans-fil ?," *Lex Electronica*, 2006.

[5] S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communications networks," *MOBILE NETWORKS AND APPLICATIONS*, vol. 1, pp. 89–104, 1996.

[6] P. J. Magnus Frodigh and P. Larsson, *Wireless ad hoc networking-The art of networking without a network*. Ericsson Review, 2000.

[7] R. Ramanathan, "Making ad hoc networks density adaptive," in *Proc. Milcom*, pp. 957–961, 2001.

[8] Z. J. al., "Special issue on wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, 1999.

[9] A. M. Chris Barrett, Martin Drozda and M. V. Marathe, "Characterizing the interaction between routing and mac protocols in adhoc networks," in *Proc. MobiHoc*, pp. 92–103, 2002.

[10] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks."

[11] J. Broch, D. A. Maltz, D. B. Johnson, Y. chun Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. Mobicom*, pp. 85–97, 1998.

[12] N. H. Vaidya, "Mobile ad hoc networks : Routing, mac and transport issues," Master's thesis, Texas A&M University, 2000.

[13] N. H. Vaidya, "Tcp for wireless and mobile hosts," 1999.

[14] D. Sun and H. Man, "Performance comparison of transport control protocols over mobile ad hoc networks," 2001.

[15] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks part i : Problem discussion and analysis of results," 1999.

[16] R. d. O. Braun and O., "Tcp in wireless mobile ad hoc networks," tech. rep., IAM-02-003, 2002.

[17] H. Balakrishnan, V. N. Padmanabhan, R. H. Katz, and Y. H. Katz, "The effects of asymmetry on tcp performance," 1997.

[18] M. Gerla, K. Tang, and R. Bagrodia, "Tcp performance in wireless multi-hop networks," in *in Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 41–50, 1999.

[19] L. Z. K. T. M. Gerla, R. Bagrodia and L. Wang, "Tcp over wireless multi-hop protocols : Simulation and experiments," Master's thesis.

[20] Z. Fu, P. Zerfos, K. Xu, H. Luo, S. Lu, L. Zhang, and M. Gerla, "On tcp performance in multihop wireless networks," 2003.

[21] D. I. Program and P. For, "Transmission control protocol," 1981.

[22] J. H. Schiller, *Mobile Communications*. Addison-Wesley, 2003.

[23] K. J. K. and Ross, *Computer Networking - A top-down approach featuring the Internet*. Addison-Wesley, 2003.

[24] R. Atkinson, "Security architecture for the internet protocol," in *RFC 1825*, 1995.

[25] K. M. M. V. Dawkins S., Montenegro G. and N. Vaidya, "End-to-end performance implications of links with errors." RFC 3155, 2001.

[26] V. Jacobson, "Compressing tcp/ip headers for low-speed serial links," 1990.

[27] B. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan, "Improving performance of tcp over wireless networks," in *in Proceedings of 17th International Conference on*, pp. 365–373, 1997.

[28] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S. K. Tripathi, "Split tcp for mobile ad hoc networks," in *in Proceedings of the IEEE Global Communications Conference (GLOBECOM 2002*, pp. 138–142, 2002.

[29] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback based scheme for improving tcp performance in ad-hoc wireless networks," in *International Conference on Distributed Computing Systems, Amsterdam*, pp. 472–479, 1998.

[30] J. L. Sun and S. Singh, "Atcp : Tcp for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1300–1315, 1999.

[31] D. Kim, C.-K. Toh, and Y. Choi, "Tcp-bus : Improving tcp performance in wireless ad hoc networks," in *Journal Of Communications And Networks*, pp. 1707–1713, 2001.

[32] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and R. Sivakumar, "Atp : A reliable transport protocol for ad-hoc networks," 2003.

[33] D. B. Johnson, D. A. Maltz, and J. Broch, "Dsr : The dynamic source routing protocol for multi-hop wireless ad hoc networks," in *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, pp. 139–172, Addison-Wesley, 2001.

[34] C.-K. T. University and C. keong Toh, "Associativity-based routing for ad-hoc mobile networks," *Wireless Personal Communications*, vol. 4, pp. 103–139.

[35] G. Holland and N. H. Vaidya, "Impact of routing and link layers on tcp performance in mobile ad hoc networks," 1999.

[36] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *IN PROCEEDINGS OF THE 2ND IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, pp. 90–100, 1997.

[37] R. Currier, "Test-drive your network designs," 1999.

[38] P. SEACORN, "Seacorn simulation tools," 2005.

[39] A. E. Rizzoli, "A collection of modelling and simulation." http ://www.idsia.ch/ andrea/simtools.html#network, 2007.

[40] K. S. P. Richard M. Fujimoto and G. F. Riley, "Network simulation," 2007.

[41] T. V. Project, "The ns manual." http ://www.isi.edu/nsnam/ns/ns-documentation.html, 2010.

[42] S. P. Punit Rathod and R. Rangarajan, "Bridging the gap between reality and simulations : An ethernet case study," 2006.

[43] S. Floyd and V. Paxson, "Difficulties in simulating the internet," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 392–403, 2001.

[44] D. Cavin, Y. Sasson, and A. Schiper, "On the accuracy of manet simulators," 2002.

[45] C. Monarch, "The cmu monarch project's wireless and mobility," 1998.

[46] S. Ivanov, A. Herms, and G. Lukas, "Experimental validation of the ns-2 wireless model using simulation, emulation, and real network," in *In 4th Workshop on Mobile Ad-Hoc Networks (WMAN'07*, pp. 433–444, 2007.

[47] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet simulation studies : The incredibles," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, pp. 50–61, 2005.

[48] S. Kurkowski, T. Camp, N. Mushell, and M. Colagrosso, "A visualization and analysis tool for ns-2 wireless simulations : inspect," 2005.

[49] V. Dham, "Link establishment in ad hoc networks using smart antennas," 2003.

[50] A. P. Tommy Svensson, "Development of laboratory exercises based on opnet modeler." http ://www.opnet.com/university_program/teaching_with_opnet/textbooks_and_materials/materials/Lab_Ex 2003.

[51] A. Varga, "Omnet++ user's manual." http ://www.omnetpp.org/doc/manual/usman.html, 2008.

[52] S. N. Technologies, "Qualnet 4.5 user's guide," 2008.

[53] G. F. Lucio, M. Paredes-farrera, E. Jammeh, M. Fleury, and M. J. Reed, "Opnet modeler and ns-2 : Comparing the accuracy of network simulators for packet-level analysis using a network testbed," in *In 3rd WEAS International Conference on Simulation, Modelling and Optimization (ICOSMO*, pp. 700–707, 2003.

[54] K. M. Reineck, "Evaluation and comparison of network simulation tools," Master's thesis, University of Applied Sciences (Bonn-Rhein-Sieg) - Department of Computer Science, 2008.

[55] L. S. Brakmo, S. W. O'malley, and L. L. Peterson, "Tcp vegas : New techniques for congestion detection and avoidance," in *In SIGCOMM*, 1994.

[56] M. Mathis, J. Mahdavi, S. Floyd, S. Floyd, and A. Romanow, "Tcp selective acknowledgment options," 1996.

[57] M. Mathis and J. Mahdavi, "Forward acknowledgment : Refining tcp congestion control," in *In Proceedings of the ACM SIGCOMM*, pp. 281–291, 1996.

[58] K. M. John Heidemann and S. Kumar, "Expanding confidence in network simulations,"

[59] B. D. Lubachevsky, “Recipes for validation,” 1999.

[60] R. Bagrodia, M. Takai, Y. an Chen, X. Zeng, and J. Martin, “Parsec : A parallel simulation environment for complex systems,” *IEEE Computer*, vol. 31, pp. 77–85, 1998.

[61] D. M. N. James H. Cowie and A. T. Ogielski, “Modeling the global internet,” 1999.

[62] P. Huang, D. Estrin, and J. Heidemann, “Enabling large-scale simulations : Selective . . .,” in *IN PROCEEDINGS OF THE INTERNATIONAL SYMPOSIUM ON MODELING, ANALYSIS AND SIMULATION OF COMPUTER AND TELECOMMUNICATION SYSTEMS*, pp. 241–248, IEEE, 1998.

Index