

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique
Université Saad Dahlab Blida 1



N° D'ordre :

Faculté des Sciences
Département d'informatique
Filière : Informatique
Spécialité : Informatique
Option : Systèmes informatiques et réseaux
Mémoire de Fin d'Études
En vue de l'obtention du diplôme Master

Thème

Amélioration des performances du protocole TCP dans les réseaux sans fil Mono-Saut mobiles

Présenté par :

Boussada Hadj Hammou

Mebarki Ahmed Imad Eddine

Soutenu le : 16 Juillet 2019

Devant le jury composé de :

Mme OUKID Lamia	Grade	Président
Mr BENAÏSSI Sellami	Grade	Examineur
Mr DOUGA Yassine	Grade	Promoteur

Année universitaire : 2018 / 2019

Dédicaces

Je dédie ce modeste travail :

A Celle qui m'a donné la vie, au vrai sens de la tendresse, au soleil

de ma vie et ma raison d'exister, à ma très chère

"MAMAN". Que dieu vous garde pour moi

A Celui qui ne cesse de se sacrifier pour moi, à ma fierté et mon bonheur à

mon espoir et ma sécurité, à mon très cher "PAPA" Que Dieux garde son

âme

A Mon Oncle l'artiste Mohamed que dieu repose son âme en paix

(Décès 12-06-2019)

Je dédie ce travail aussi

A Mes très chères sœurs : Yamina, Fatima, Latifa.

Mes très chers frères : Ilyes, Hamza, Idriss, Rabie mon neveu.

A toute ma famille.

A Mon binôme Imad Eddine

A tous mes amis et collègues. Ceux qui sont dans mon cœur et qui ont

une relation de proche ou de loin avec la réalisation et a tous mes

enseignants qui m'ont donné le savoir durant tout mon cursus. Merci.

Hadj hammou

Dédicaces

Je dédie ce modeste travail :

A mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi mon père.

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur, maman que j'adore.

Aux personnes dont j'ai bien aimé la présence dans ce jour, à tous mes frères et mes sœurs (Mohamed, Sara, Younes, Choueib).

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagné durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude et frères de cœur, toi Abdeslam, Farid, Mohamed et Mon binôme Hadj hammou.

Ahmed Imad Eddine

Remerciement

Tout d'abord, nous tenons à remercier « Allah », le clément et le miséricordieux de nous avoir donné la force et le courage de mener à bien ce modeste travail.

Nous aimerons exprimer nos vifs remerciements à mon promoteur **Mr. YASSINE DOUGA** pour tout le temps et l'intérêt qu'il nous a consacré et pour ses orientations et les conseils qu'il nous a prodigué tout au long de ce travail. Qu'il trouve ici nos sentiments de gratitude et de profonde reconnaissance.

Nous aimerons aussi remercier tous les professeurs qui ont contribué à nos formations. Que tous les membres du jury trouvent ici l'expression de nos profonds respects pour avoir pris la peine d'examiner notre mémoire. Enfin, nos remerciements vont également à toutes les personnes qui ont, de près ou de loin, apporté aide et encouragement.

ملخص

في الوقت الحاضر، يستخدم العديد من مستخدمي الأجهزة المحمولة بروتوكول النقل (TCP) في الشبكات اللاسلكية للوصول إلى الخدمات عبر الإنترنت. أصلاً تم تصميم TCP للشبكات السلكية. ومع ذلك، فإنه يعاني من بعض العيوب على الشبكات اللاسلكية، مثل تدهور قيمة التدفق المدين بسبب الخسائر العشوائية والاتصال المتقطع. يحدث هذا بسبب عدم تمييز خسائر الحزمة التي تحدث بسبب ازدحام من تلك المشكلات المتعلقة بالبيئة اللاسلكية. في هذه المذكرة، نقتراح تقنية لتحسين أداء بروتوكول TCP في شبكات القفز الأحادي.

يستخدم النهج المقترح نقاط قوة إشارات الاتصالات السابقة لتقدير قوة الإشارة الحالية، وسيتم استخدام هذه القيمة المقدرة للتمييز بين فقدان الازدحام وخسارة الحركة.

تم اختبار الحل في بيئة ملائمة، وتم النظر في سيناريوهات مختلفة، وتمت مقارنة النتائج مع النهج المقترح في [1] وإصدار TCP الخاص بنظام تشغيل اوبانتو (TCP Reno) في ظل نفس الظروف. بعد المقارنة، يوفر النهج المقترح أداء إنتاجية أفضل في بيئة اختبار لاسلكية متنقلة.

كلمات البحث:

TCP، الازدحام، قوة الإشارة، النقل، المحاكاة، العتبة، GPS، فقدان الحزمة.

Résumé

De nos jours, nombreux d'utilisateurs mobiles se servent du protocole de transport (TCP) dans les réseaux sans fil afin d'accéder à des services sur internet. A l'origine le protocole TCP a été conçu pour les réseaux filaires. Cependant, il souffre de quelques défauts sur les réseaux sans fil, tels que la dégradation de débit en raison de pertes aléatoires et de la connectivité intermittente. Cela est causé par l'indistinction des pertes de paquets qui sont dus à la congestion de ceux dû aux problèmes de l'environnement sans fil. Dans ce mémoire, une technique est proposée pour améliorer les performances du protocole TCP dans les réseaux sans fil Mono-Saut.

L'approche proposée utilise les puissances des signaux des communications précédentes afin d'estimer la puissance de signal actuelle, cette puissance estimée va être utilisée pour faire la différence entre une perte due à la congestion et celle due à la mobilité.

La solution a été testée dans un environnement d'émulation, différents scénarios ont été considérés, les résultats ont été comparés avec l'approche proposée dans [1] et la version TCP de Ubuntu (TCP Reno) sous les mêmes conditions. Après une comparaison, l'approche proposée donne de meilleures performances en matière de débit accompli dans un environnement de test sans fil mobile mono saut.

Mots clés :

TCP, Congestion, Puissance de signal, Transport, émulation, Seuil, GPS, Perte de paquet.

Abstract

Nowadays, many mobile users use the transport protocol (TCP) in wireless networks to access services over the internet. Originally TCP was designed for wired networks. However, it suffers from some defects on wireless networks, such as debit degradation due to random losses and intermittent connectivity. This is caused by the incapability of distinction between the packet losses that are due to congestion and those due to the problems in the wireless environment. In this thesis, a technique is proposed to improve the performance of the TCP protocol in single-hop networks.

The proposed approach uses the signal strengths of previous communications to estimate the current signal strength, this estimated power will be used to differentiate between congestion and mobility losses.

The solution was tested in an emulation environment, different scenarios were considered, the results were compared with the approach proposed in [1] and the TCP version of Ubuntu (TCP Reno) under the same conditions. After a comparison, the proposed approach provides better throughput performance in a single-hop mobile wireless test environment.

Keywords:

TCP, Congestion, Signal Strength, Transport, Emulation, Threshold, GPS (Global Positioning System), Packet Loss.

Table des matières

ملخص	v
Résumé.....	vi
Abstract.....	vii
Table des matières	viii
Liste des figures.....	xi
Liste des tableaux.....	xii
Liste des acronymes	xiii
Introduction Générale	1
Chapitre.1 Notions de base	3
1.1 Introduction	4
1.2 Les réseaux sans fil	4
1.2.1 La norme IEEE 802.11	4
1.2.2 Caractéristiques de l'environnement sans fil.....	5
1.2.2.1 Effet du doppler	5
1.2.2.2 Le bruit	5
1.2.2.3 Indicateur d'intensité du signal reçu (RSSI)	6
1.3 Le protocole TCP	6
1.3.1 Fonctions de contrôle de congestion.....	7
1.3.2 Slow Start (démarrage lent)	7
1.3.3 Congestion avoidance	8
1.3.4 L'algorithme Additive Increase and Multiplicative Decrease (AIMD).....	8
1.3.5 Différentes versions de TCP	9
1.3.5.1 TCP Tahoe	9
1.3.5.2 TCP Reno	10
1.3.5.3 TCP Vegas	11
1.3.6 Défis et limitations du TCP dans les réseaux sans-fil.....	11
1.3.6.1 Erreur de Bit élevé (pertes aléatoires)	12
1.3.6.2 La mobilité.....	12
1.3.6.3 Le conflit de Canal	12
1.3.6.4 Paquet non séquencé.....	13
1.3.6.5 Contrôle d'accès au support (MAC)	13
1.3.6.6 Impossible de différencier les pertes de transmission et de congestion ` ...	15
1.4 Système de positionnement global.....	15
1.4.1 Coordonnées GPS	16
1.4.2 Comment fonctionne le GPS	16
1.5 Composants et matériels.....	17

1.5.1	Arduino Uno	17
1.5.2	NEO-6M GPS	18
1.5.3	Schémas	19
1.6	Conclusion.....	19
Chapitre.2	État de l'art	20
2.1	Introduction	21
2.2	Des mécanismes pour l'optimisation de protocole TCP dans les réseaux sans fil....	21
2.2.1	TCP MUZHA.....	21
2.2.2	TCP NRT	25
2.2.2.1	La phase de détection	25
2.2.2.2	La différenciation de non-congestion RTOs	26
2.2.2.3	La phase de réaction	26
2.2.3	Solution En Mono-Saut.....	29
2.2.4	TCP-WELCOME.....	32
2.2.4.1	Algorithme de différenciation LDA-Loss	32
2.2.4.2	Algorithme de récupération LRA-Loss	32
2.2.5	Solution en Multi-Saut.....	33
2.3	Critique et comparaison	34
2.4	Conclusion.....	36
Chapitre.3	Conception	36
3.1	Introduction	38
3.2	Description globale de l'approche.....	38
3.2.1	Un aperçu sur la solution	38
3.2.2	Récupération de la valeur de la vitesse	38
3.2.3	Calcule d'intervalle de temps	38
3.2.4	Récupération de la puissance de signal.....	39
3.2.5	Formule d'estimation de la prochaine puissance du signal	39
3.2.6	La précision d'estimation de la puissance du signal	41
3.3	Algorithme descriptif	41
3.3.1	Algorithme de récupération de l'intervalle de temps	41
3.3.2	Algorithme générale de la solution proposé :	42
3.4	Organigrammes de la solution :	44
3.5	Conclusion.....	47
Chapitre.4	Implémentation et Evaluation.....	48
4.1	Introduction	49
4.2	Environnement d'implémentation.....	49
4.2.1	Équipement logiciel (Système d'exploitation).....	49

4.2.2	Équipement matériel (GPS)	49
4.2.3	Description de la topologie d'émulation.....	50
4.3	Étude de l'évolution de la puissance du signal dans le temps	51
4.3.1	Les résultats obtenus de l'expérience	51
4.3.2	Interprétation des graphes	54
4.3.3	La portée minimale de la puissance du signal	54
4.4	Evaluation des performances	54
4.4.1	Les paramètres de l'émulation	55
4.4.2	Un aperçu des scénarios	55
4.4.3	Les résultats des tests	56
4.4.4	Discussion des résultats	61
4.5	Conclusion.....	61
	Conclusion Générale et Perspectives.....	63
	Références.....	64
	Annexes	68
	1.Les environnements mobiles	68
	1.1. Les réseaux avec infrastructure	68
	1.2. Les réseaux sans infrastructure (Ad Hoc).....	69

Liste des figures

Figure 1 :Le démarrage lent de TCP.....	08
Figure 2 :TCP Tahoe.....	10
Figure 3 :TCP Reno	11
Figure 4 :Le phénomène des nœuds exposés.....	14
Figure 5 :Le phénomène des nœuds cachés.....	14
Figure 6 :La carte Arduino Uno.....	17
Figure 7 :Module GPS NEO-6M.....	18
Figure 8 :Montage du Module GPS avec la carte Arduino.....	19
Figure 9 : Diagramme de transition de phase de TCP Muzha.....	23
Figure 10 :Organigramme de la procédure pour TCP Muzha lors de la réception d'un ACK	24
Figure 11 :Organigramme de TCP NRT.....	28
Figure 12 : Organigramme de Solution En Mono-Saut.....	31
Figure 13 :Organigramme du calcul de l'intervalle de temps	45
Figure 14 :Organigramme de l'approche proposée en mono saut.....	46
Figure 15 :Topologie de l'émulation	50
Figure 16 : Les résultats de RSSI dans un rayon de 12m	52
Figure 17 : Les résultats de RSSI dans un rayon de 14m	52
Figure 18 : Les résultats de RSSI dans un rayon de 16m	53
Figure 19 : Les résultats de RSSI dans un rayon de 18m	53
Figure 20 : Les résultats du scénario numéro un	56
Figure 21 : Les résultats du scénario numéro deux	57
Figure 22 : Les résultats du scénario numéro trois	58
Figure 23 : Les résultats du scénario numéro quatre	59
Figure 24 : Les résultats du scénario numéro cinq	60
Figure 25 : Le modèle de réseaux mobile avec infrastructure.....	67
Figure 26 : Le modèle de réseaux mobile sans infrastructure	68

Liste des tableaux

Tableau 1: Description fonctionnelle de chaque broche du module GPS NEO-6.....	18
Tableau 2: Formule de DRAI.....	22
Tableau 3: Comparaison entre les différentes approches de TCP	35
Tableau 4: La structure de données utilisée.....	39
Tableau 5: Les vitesses avec l'intervalle de temps correspondant.....	55

Liste des acronymes

ACK: Acknowledgement
AIMD: Additive Increase/Multiplicative Decrease
BER: Bit Error Rate
CSMA / CD: Carrier-sense multiple access with collision detection
Cwnd: Congestion Window
DRAI: Data Rate Adjustment Index
DUAL: Diffusing Update Algorithm
ECN: Explicit Congestion Notification
GHz: GigaHertz
GPS: The Global Positioning System
IEEE: Institute of Electrical and Electronics Engineers
IP: Internet Protocol
LAN: Local area network
MAC: Medium access control
MAN: Metropolitan area network
Mbps: Megabit per seconds
MRAI: Minimum Data Rate Adjustment Index
MSS: Maximum Segment Size
NAVSTAR: Navigation Satellite Timing and Ranging
PDAs: Personal Digital Assistant
PWM: Pulse Width Modulation
QoS: Quality of service
QoE: Quality of Experience
Rcvwnd: Receiver Window
RFC: Request for Comments
RSSI: Received signal strength indication
RTO: Retransmission Timeout
RTT: Round-Trip Time
ssthresh: startthreshold
TCP: Transmission Control Protocol
TWND: Transmission Window
USB: Universal Serial Bus

WiFi: Wireless Fidelity

WLAN: Wireless LAN

Introduction Générale

Avec l'augmentation de l'utilisation d'équipements mobiles tels que les ordinateurs portables, les PDAs(Personal Digital Assistant), et les téléphones cellulaires, il y a de plus en plus d'accès sans fil à l'Internet [2]. La technologie sans fil a envahi le marché des réseaux de télécommunication. Plusieurs standards des IEEE(Institute of Electrical and Electronics Engineers) ont vu jusqu'à nos jours, on peut citer à titre d'exemple : WiFi (Wireless Fidelity IEEE 802.11), Bluetooth (IEEE 802.15.1), Zigbee (IEEE 802.15.4). Ces standards équipent une large gamme d'équipements mobiles [3]. Ce progrès technologique fait que les réseaux de télécommunication sans fil sont actuellement un des domaines de recherche de l'informatique les plus actifs.

Dans un réseau à un saut, lorsqu'un paquet quitte la source, il ne prend qu'un seul saut (Un saut signifie le nombre de réseaux différents qu'un paquet doit traverser pour atteindre son adresse de destination finale) avant d'atteindre son adresse de destination. Le protocole de contrôle de transmission (TCP) est actuellement le protocole de couche de transport le plus répandu sur Internet. Il transporte plus de 90% du trafic sur Internet à cause de sa fiabilité [4].Néanmoins, plusieurs études ont démontré que le protocole TCP n'est pas vraiment adapté aux environnements sans fil en générale et les réseaux sans fil mobile en particulier, cela causé l'incompatibilité de TCP de différencier les pertes dues à une congestion et celles dues à l'environnement. Cette erreur entraîne des baisses de performances lorsque des réseaux sans fil sont traversés [5].

Par conséquent, plusieurs améliorations et optimisations ont été proposées au cours des dernières années pour améliorer les performances du TCP sur les réseaux sans fil à saut unique. L'approche [1] a été étudiée et émulée dans un réseau sans fil à saut unique, cette variante a prouvé et démontré sa propre efficacité en matière de QoS (Quality of service) et de QoE(Quality of Experience). La problématique de ce mémoire est de prédire la puissance de signal minimal du prochain saut dans un environnement sans fil saut-unique ou les nœuds se déplacent à des vitesses variables afin d'améliorer les performances de protocole TCP dans l'environnement sans fil.

Dans ce mémoire, nous proposons une nouvelle adaptation du protocole TCP. Cette adaptation a pour but d'améliorer les performances (débit de transfert) du protocole TCP dans les réseaux sans fil saut-unique mobile.

Le reste de mémoire est structuré comme suite :

- Le premier chapitre, fourni des notions de base sur tout ce qui est en relation avec les réseaux sans fil et le protocole de transport TCP, afin de servir de point de repère pour les lecteurs dans ce qui suit dans les chapitres suivants.
- Le deuxième chapitre, l'état de l'art, décrit, compare et critique les travaux existants. Le but de ce chapitre est de pointer sur les limites des travaux précédents et leurs inconvénients.
- Dans le troisième chapitre, nous décrivons avec détails notre solution proposée, les formules et les méthodes utilisées, en plus des algorithmes explicatifs.
- Dans le quatrième chapitre, nous parlons de l'environnement et des conditions d'émulation, les résultats obtenus et les critiques qui se rapportent avec.

Enfin nous terminons par une conclusion et perspectives.

Chapitre

1

Notions

de Base

1.1 Introduction

Les réseaux sans fil sont en plein développement. Conduit par l'émergence des appareils de communication tels que les téléphones cellulaires, les ordinateurs portables et par la flexibilité de leur interface, ce qui permet à un utilisateur de changer de place tout en restant connecté. Ainsi l'utilisateur transite d'un environnement restreint vers un environnement ubiquitaire où les traitements se font à travers plusieurs infrastructures.

Dans le reste de ce chapitre, on présentera les éléments fondamentaux des réseaux sans fil, puis on décrira une généralité sur le protocole de transport TCP et les différentes versions de TCP, et à la fin une généralité sur le système de positionnement global GPS.

1.2 Les réseaux sans fil

Un réseau sans fil (en anglais Wireless network) [6] [7] [8] est comme son nom l'indique un réseau dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire. Grâce aux réseaux sans fil, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu, c'est la raison pour laquelle on entend parfois parler de "mobilité" [9].

1.2.1 La norme IEEE 802.11

L'IEEE 802.11 est supervisé par l'IEEE, en particulier le comité des normes IEEE LAN (Local area network) / IEEE 802 MAN (Metropolitan area network). Elle fait référence à l'ensemble de normes qui définissent la communication pour les LAN sans fil (réseaux locaux sans fil ou WLAN (Wireless LAN)). La technologie à la base de la norme 802.11 est présentée aux consommateurs sous le nom de Wi-Fi.

Il existe plusieurs spécifications dans la famille 802.11 [10].

- 802.11a une extension à 802.11 qui s'applique aux réseaux locaux sans fil et fournit jusqu'à 54 Mbps (Mégabit per seconds) dans la bande de 5 GHz (GigaHertz).
- 802.11b (également appelé débit haut débit 802.11 ou Wi-Fi) s'applique aux LANs sans fil et assure une transmission à 11 Mbps (avec un repli à 5,5, 2 et 1 Mbps) dans la bande des 2,4 GHz.
- 802.11e un projet de norme sans fil qui définit la prise en charge de la qualité de service (QoS) pour les réseaux locaux et constitue une amélioration des spécifications des réseaux locaux sans fil (WLAN) 802.11a et 802.11b.

- 802.11g utilise des techniques OFDM similaires à la 802.11a, mais en restant dans la bande passante ISM à 2.4 GHz. Les débits possibles atteignent également les 54 Mbit/s tout en gardant la compatibilité avec les équipements 802.11b.
- 802.11n augmente la vitesse des réseaux sans fil local (WLAN), elle améliore la fiabilité et elle étend la zone de couverture sans fil.

1.2.2 Caractéristiques de l'environnement sans fil

1.2.2.1 Effet du doppler

Lorsqu'un émetteur et un récepteur se déplacent l'un par rapport à l'autre, la fréquence du signal reçu ne sera pas la même que celle du signal émis. Par exemple, lorsqu'ils se déplacent l'un vers l'autre, la fréquence du signal reçu est supérieure à celle du signal émis. C'est ce qu'on appelle l'effet Doppler. La variation de fréquence due à l'effet Doppler dépend du mouvement relatif entre la source et le récepteur et de la vitesse de propagation de l'onde [11].

1.2.2.2 Le bruit

Le bruit radioélectrique regroupe l'ensemble des signaux ne transportant pas d'information utile et venant perturber le signal désiré, il est donc indépendant du signal émis. Il s'agit d'une perturbation aléatoire dont les origines sont le milieu de transmission (bruit externe) et les dispositifs utilisés dans le récepteur (bruit interne) [11] [12].

Les sources de bruit externe peuvent être d'origine extra-terrestre ou terrestre. Elles regroupent les bruits et des parasites atmosphériques, les rayonnements divers captés par l'antenne, les interférences éventuelles entre les utilisateurs du milieu de transmission ou encore les bruits d'origine industrielle. Le bruit interne a pour origine le mouvement brownien des électrons étant présents dans les composants électroniques du récepteur. Ces électrons étant présents dans la matière en très grand nombre et évoluant indépendamment les uns des autres tout en suivant une même loi, le bruit interne peut alors être modélisé, d'après le théorème de la limite centrale [13], par un processus gaussien.

Toutes les contributions du bruit interne et externe seront prises en compte dans une source unique de bruit $n(t)$ située en amont de récepteur. Néanmoins, le bruit interne est en général celui qui est prépondérant dans le système de transmission. Par conséquent, lorsque des systèmes à antennes multiples sont étudiés, on peut judicieusement supposer que les bruits propres à chacune des antennes sont décorrélés d'une antenne à l'autre et au cours du temps.

1.2.2.3 Indicateur d'intensité du signal reçu (RSSI)

RSSI (soit Received Signal Strength Indicator, en anglais) est le nom de la force de signal de l'environnement d'un réseau sans fil. Il n'est pas remarquable par l'utilisateur d'un dispositif receveur. Cependant, les dispositifs IEEE 802.11 fournissent à leurs utilisateurs des données de mesure du signal, car la force du signal peut varier de manière considérable, affectant ainsi la fonctionnalité au sein du réseau sans fil.

Les mesures RSSI représentent la qualité relative d'un signal reçu sur un dispositif. RSSI indique le niveau de puissance reçu après une possible perte au niveau de l'antenne et du câble. Plus la valeur RSSI est élevée, plus le signal n'est fort. Lorsque mesuré avec des chiffres négatifs, si le chiffre est proche de zéro ceci signifie qu'il existe un meilleur signal. Par exemple : -50 est un assez bon signal, -75 est un signal plutôt raisonnable et -100 n'est pas un signal du tout.

Même si RSSI et dBm sont des unités de mesure différentes, elles indiquent la force du signal. Le dBm est un taux de puissance de la puissance mesurée en tant que références à un mW (milliwatt). Alors que dBm est un indice absolu, le RSSI est un indice relatif.

Pour mesurer de manière significative la bonne qualité d'un signal, il est nécessaire de supprimer le bruit sur la ligne de la puissance du signal. Une différence plus importante du signal par rapport au bruit détermine une meilleure force du signal [12].

1.3 Le protocole TCP

Le protocole de contrôle de transmission TCP a été défini pour fournir un service de transfert de données fiable entre deux applications sur des stations distantes raccordées par un réseau à commutation de paquets utilisant le protocole IP (Internet Protocol). La fiabilité du transfert est obtenue par différents mécanismes tels que l'établissement de connexion, la gestion de timers de retransmissions ou encore le contrôle de la fenêtre de retransmissions. La spécification initiale, définie dans le « RFC (Request for Comments) 793 » de 1981 a fait l'objet de nombreux travaux qui ont conduit à des améliorations de la spécification initiale.

Dans la pratique, la plupart des déploiements du protocole TCP ont été soigneusement conçus dans le contexte des réseaux câblés. Dans un environnement ad hoc l'implémentation de TCP peut conduire à des performances médiocres dues aux propriétés intrinsèques des réseaux sans fil [14].

1.3.1 Fonctions de contrôle de congestion

Le contrôle de congestion est un mécanisme qui permet de régir le débit d'un ou de plusieurs flux. Ce mécanisme vise généralement à répondre aux trois exigences suivantes [18] :

- Maximiser l'utilisation de la bande passante.
- Limiter le nombre de paquets perdus, c'est-à-dire ne pas dépasser les capacités des ressources du réseau.
- Partager équitablement la bande passante entre les différents flux parcourant le ou les liens les plus lents.

TCP régule ses envois à travers un système de fenêtre glissante (sliding window) qui définit la quantité de paquets pouvant être envoyés sans être acquittés, en termes de numéro de séquence. Cette fenêtre, dite fenêtre de transmission, que l'on abrègera en TWND (Transmission Window), est calculée à partir de deux autres fenêtres.

- a) La fenêtre permise, notée Rcvwnd (Receiver Window). Permet au récepteur d'annoncer le nombre de segments qu'il est capable actuellement de recevoir. L'envoi de cette valeur dans chaque paquet constitue le contrôle de flux.
- b) La fenêtre de congestion, notée Cwnd (Congestion Window). Celle-ci est maintenue par le contrôle de la congestion qui s'effectue au niveau de l'émetteur.

La fenêtre de transmission est alors régie par la formule : $\text{Min}(\text{Cwnd}, \text{Rcvwnd})$.

1.3.2 Slow Start (démarrage lent)

La phase de démarrage lent, ou slow-start est utilisée quand le TCP ignore quelle est la bande passante disponible au démarrage de la session, ou lorsque les conditions du réseau semblent avoir grandement changé forçant à redécouvrir la nouvelle bande passante disponible. TCP utilise ce mécanisme pour sonder le réseau : il commence par émettre une petite quantité de données puis augmente rapidement sa capacité d'émission jusqu'à la détection d'un événement de congestion ou au dépassement d'une variable appelée slow-startthreshold (ssthresh), qui détermine si la connexion doit continuer en slow-start ou passer à l'état stable. Durant le slow-start, l'émetteur commencera par émettre le contenu de la fenêtre de congestion cwnd, dont la taille est communément fixée à 3 segments par la RFC 3390 [19] et attend l'acquittement. Une fois l'acquittement reçu, l'émetteur peut alors doubler le nombre de segments et attendre les acquittements. Ce comportement se traduit par une croissance exponentielle de la fenêtre de congestion qui double pour chaque délai aller-retour, ou RTT (Round-Trip Time) jusqu'à trouver la limite de congestion du réseau. Une fois cette limite

atteinte, des segments commencent à se perdre et l'émission reprend avec une fenêtre de congestion à 1. TCP quitte alors la phase de slow-start pour passer en mode d'évitement de congestion et divise par deux la fenêtre de congestion. La **Figure 1** suivante illustre la phase du slow-start :

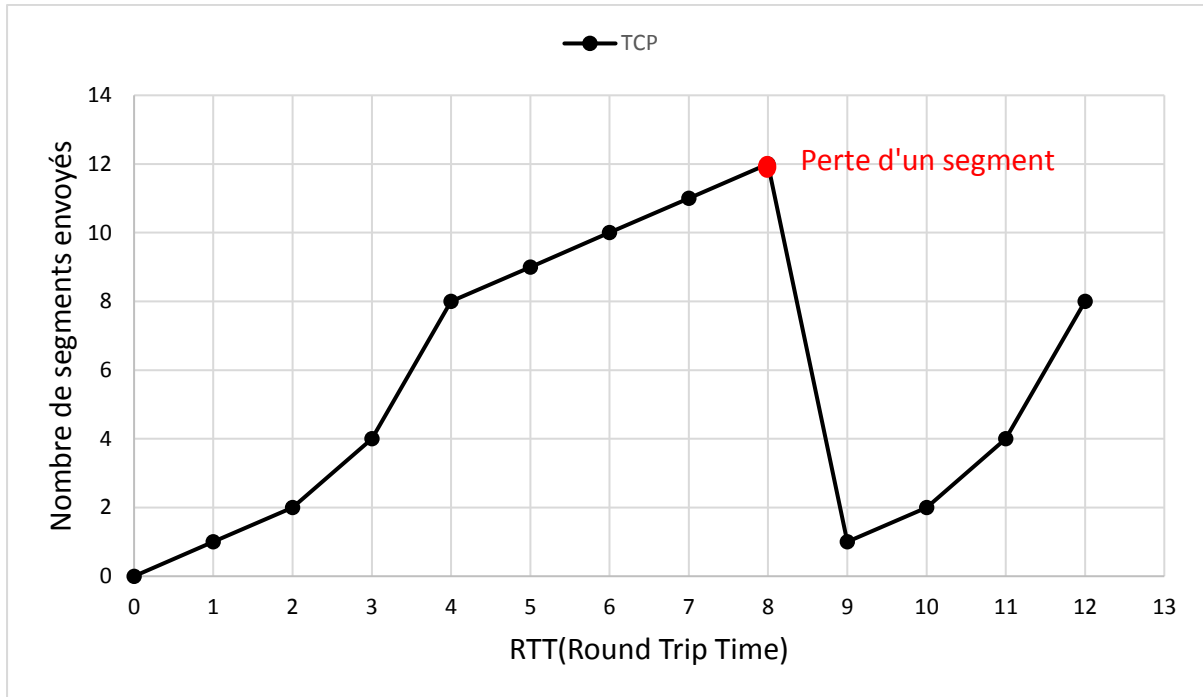


Figure 1 : le démarrage lent de TCP [23]

1.3.3 Congestion avoidance

Au-delà d'une certaine limite de valeur de cwnd (slow start threshold, ssthresh), TCP passe en mode d'évitement de congestion. À partir de là, la valeur de cwnd augmente de façon linéaire et donc bien plus lentement qu'en slow start : cwnd s'incrémente d'un MSS (Maximum Segment Size) égal à un paquet à chaque RTT. Dans ce mode de fonctionnement, l'algorithme détecte aussi rapidement que possible la perte d'un paquet : si nous recevons trois fois l'ACK(Acknowledgement) même paquet, on n'attend pas la fin d'un timeout pour réagir. En réaction à cette perte, on fait descendre la valeur de ssthresh ainsi que cwnd (on repasse éventuellement en mode de Slow Start). On utilise la technique de Fast Retransmit pour renvoyer rapidement les paquets perdus [19].

1.3.4 L'algorithme Additive Increase and Multiplicative Decrease (AIMD)

TCP utilise l'algorithme distribué AIMD (Additive Increase and Multiplicative Decrease). AIMD permettant de faire converger les différents flux partageant les mêmes conditions réseaux vers le même débit équitable. Cet algorithme est basé sur une augmentation

additive assuré par la phase d'évitement de congestion et une réduction multiplicative de la fenêtre cwnd réalisée à chaque congestion où cwnd est réduite de moitié.

Lorsque cwnd atteint la taille correspondant à un partage équitable de la bande passante, le même débit est obtenu par chaque flux TCP de même RTT traversant le lien congestionné. Dans ce cas, chaque flux subit les cycles d'une perte suivie d'une phase d'évitement de congestion [15].

Pour sortir d'une congestion d'autres solutions ont été proposées où la fenêtre de congestion est réduite différemment. Les objectifs à garder en vue pour ces algorithmes sont d'être efficaces, équitables et qu'ils présentent une convergence vers une solution adéquate à partir d'un état quelconque et avec une vitesse assez rapide. Ainsi Z. Wang et J. Crow Croft [16]proposent l'algorithme DUAL (Diffusion Update Algorithm) dans lequel la fenêtre de congestion CW est réduite d'un facteur 8 en fonction des RTT mesures tous les round-trip. De même que R. Jain [17] propose une solution TRI-S dans laquelle la fenêtre est également réduite d'un facteur 8.

1.3.5 Différentes versions de TCP

1.3.5.1 TCP Tahoe

La première version de TCP basée sur un système de fenêtrage variable est apparue pour répondre aux situations de congestion dans les réseaux longue distance. Cette version est baptisée TCP Tahoe. Elle utilise un système de slow-start, avec une valeur initiale de cwnd à 1 et une valeur de cwnd maximum de ssthresh (avec une valeur par défaut de 65535). Le graphique ci-dessous (voir **Figure 2**) représente les différentes phases de TCP Tahoe telque au début il effectue un slow start, une fois qu'une perte de paquet se produise, le nouveau ssthresh prend la valeur de cwnd courante et celle-ci est remise à la valeur 1. Une fois le seuil ssthresh atteint, TCP exécute une Congestion Avoidance. À partir de là la valeur de cwnd augmente de façon linéaire et donc plus lentement qu'en slow-start. A la réception de trois fois le même ACK, une congestion est déclarée et le paquet perdu est renvoyé sans attendre l'expiration du timeout (fast retransmit) [18].

Le problème majeur de Tahoe est le temps dépensé dans la détection de pertes. En effet, les temporisateurs sont généralement et même plus souvent d'une résolution de 500 ms, pendant que les délais d'aller-retour tendent de plus en plus à se raccourcir (câbles torsadé, fibre optique, ...). La détection d'une perte peut alors prendre au moins 500 ms ce qui constitue plusieurs délais d'aller-retour.

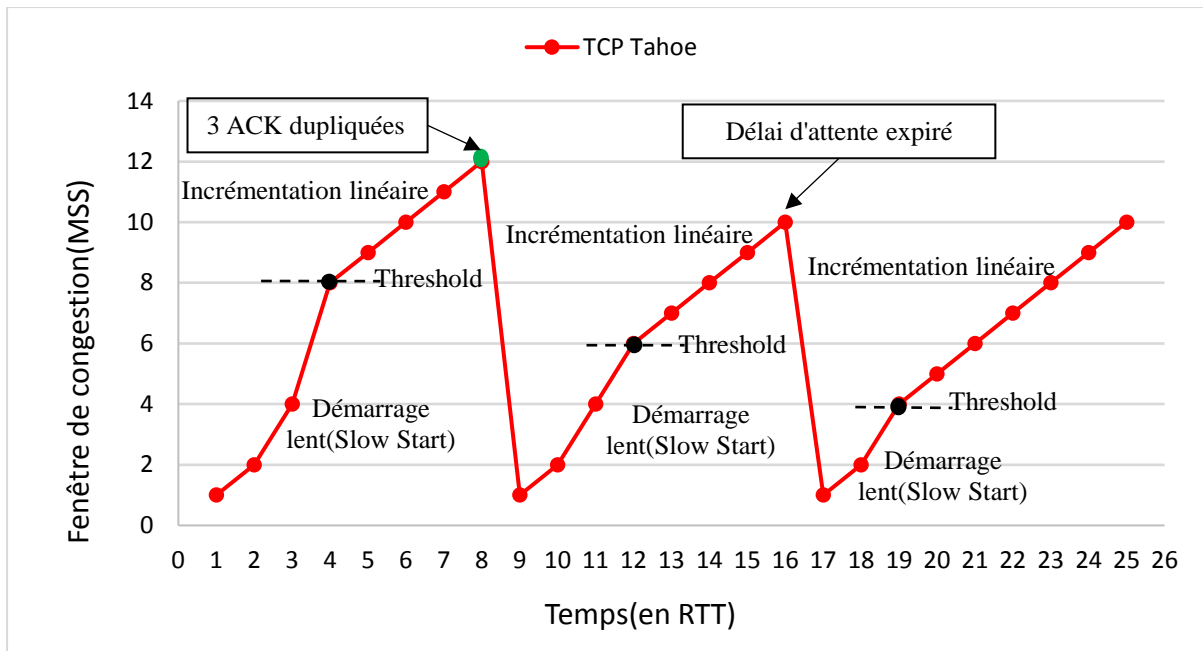


Figure 2 : TCP Tahoe [19]

1.3.5.2 TCP Reno

L'algorithme New Reno n'est qu'une modification légère mais significative apportée à l'algorithme Reno. En effet, cette modification, permet à TCP New Reno d'éviter de devoir attendre pour qu'un déclenchement de retransmission se produise lorsque plusieurs paquets sont perdus au sein d'une même fenêtre de congestion [20].

Le changement s'opère au niveau de la phase de recouvrement rapide (Fast recovery) : Tant que l'émetteur n'a pas reçu les ACK de tous les paquets perdus (réception d'un acquittement partiel pour seulement une partie des segments perdus), TCP New Reno reste dans cette phase. Un accusé partiel permet donc de se rendre compte que parmi les paquets retransmis une première fois, certains ont à nouveau été perdus. Dès lors, cette version n'attend pas un Time Out pour retransmettre ces paquets perdus. TCP New Reno ne quitte pas l'algorithme de Fast Recovery contrairement à Reno qui sort de ce mode dès la réception d'un ACK non dupliqué. La version TCP New Reno apporte des réponses aux pertes successives, cependant elle souffre du fait qu'elle utilise une RTT pour détecter chaque perte de paquet. Néanmoins elle est parmi les versions les plus utilisées de TCP actuellement. La **Figure 3** suivante illustre les différentes phases du TCP New Reno :

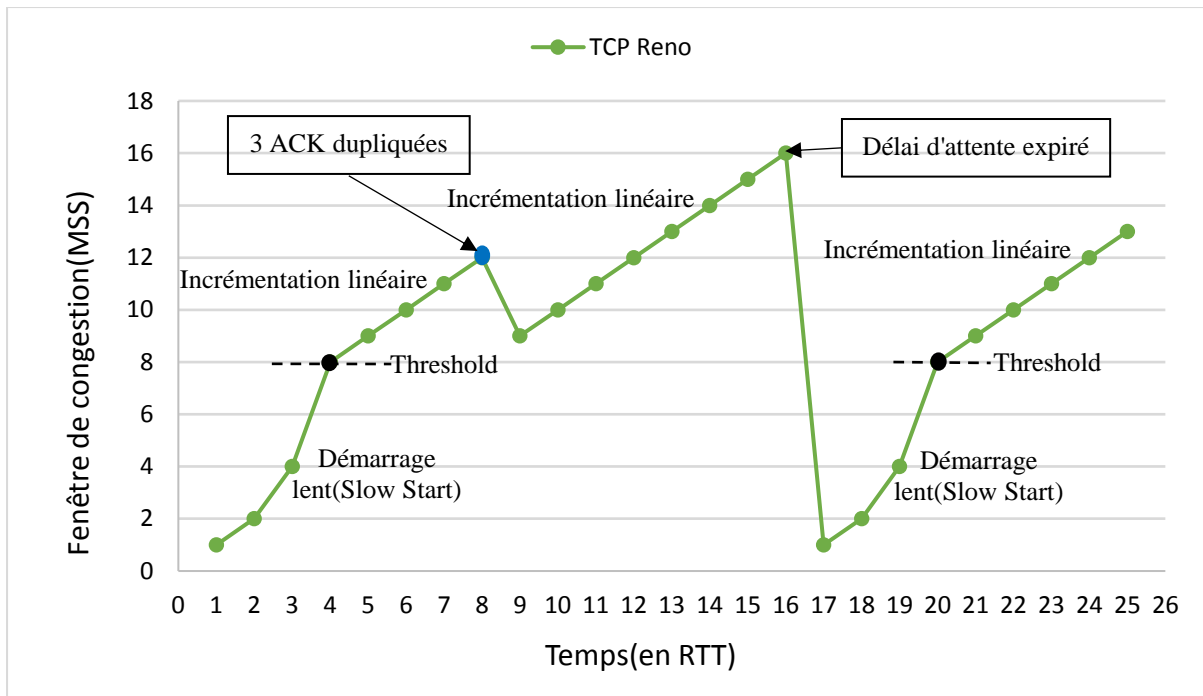


Figure 3 : TCP Reno [19]

1.3.5.3 TCP Vegas

Plutôt que d'attendre une perte de paquet, TCP Vegas prend en compte le temps de réponse du destinataire (le RTT) afin d'en déduire le ratio auquel envoyer les paquets. En fonction du temps de réponse, on est capable de supposer l'état des buffers des routeurs intermédiaires. Vegas modifie pour cela plusieurs algorithmes vus jusqu'ici (slow-start, Congestion Avoidance, Fast Retransmit). Grâce à cette technique, Vegas a de meilleurs débits et moins de pertes de paquets que Reno. Cet algorithme permet d'atteindre un partage équitable des ressources. De même, il y a assez peu de pertes avec Vegas puisqu'à l'état stable, le débit correspond de près à la capacité du lien [21].

TCP Vegas permet de s'adapter plus rapidement aux changements de disponibilité des liens. Cependant, les performances se dégradent lorsqu'il est utilisé avec d'autres protocoles qui ne prennent pas en compte l'état des liens avant une perte de paquet [21]. Son inconvénient est qu'il nécessite une modification de la pile TCP au niveau de l'émetteur et du récepteur.

1.3.6 Défis et limitations du TCP dans les réseaux sans-fil

L'environnement est sensiblement différent de celui des réseaux filaires, alors que TCP traite un réseau sans fil comme un réseau câblé. En conséquence, le protocole TCP doit faire face à des défis dans les environnements sans fil tels que :

1.3.6.1 Erreur de Bit élevé (pertes aléatoires)

Le taux d'erreur sur les bits BER (Bit Error Rate) varie de 10^{-6} à 10^{-8} pour les réseaux câblés, cela varie de 10^{-3} à 10^{-1} pour les réseaux sans fil. En raison du BER élevé, les pertes de paquets sont très fréquentes et TCP réagit à ces pertes en réduisant la fenêtre de congestion. Cela conduit donc à des performances non optimales [22]. En cas de pertes en rafales, TCP présente des performances plus médiocres, car TCP suppose que le premier paquet a été perdu et qu'il a formé les accusés de réception en double. Après avoir retransmis le premier paquet perdu, l'expéditeur doit attendre de recevoir un nouvel accusé de réception en double, avant de déterminer le prochain paquet perdu. Le TCP conventionnel récupère la perte d'un seul paquet par temps d'aller-retour (RTT). Ainsi, en cas de pertes plus consécutives, TCP doit les récupérer dans plusieurs RTT. Cela montre la limitation de TCP que le processus de récupération des erreurs est très lent. En termes simples, les performances de l'application ne sont pas uniquement affectées par la vitesse à laquelle TCP restaure sa transmission, mais également par sa propre capacité à remédier aux erreurs de transmission. Cela montre que le mécanisme d'encombrement de TCP n'est pas adapté à l'environnement sans fil [22].

1.3.6.2 La mobilité

La mobilité peut provoquer la rupture de lien et l'échec de l'itinéraire entre deux nœuds voisins, ce qui peut entraîner alternativement des pertes de paquets. Comme TCP ne distingue pas entre les pertes de paquets causées par les défaillances de routes et celles en raison d'une congestion, il déclenche les mécanismes de contrôle de congestion qui réagissent négativement. En effet, si la découverte d'un nouvel itinéraire prend plus de temps qu'un RTO (Retransmission Timeout), l'émetteur TCP invoque un contrôle de congestion et le débit déjà réduit en raison des pertes va encore diminuer. Il pourrait être encore pire lorsque la source et la destination d'une connexion TCP se situent dans différentes partitions du réseau. Dans un tel cas, plusieurs expirations RTO consécutives conduisent à une inactivité durant une ou deux minutes, même si l'expéditeur et le récepteur se reconnectent finalement [23].

1.3.6.3 Le conflit de Canal

Les performances TCP diminuent également à mesure que le nombre de conflits augmente. [24] Des conflits de canaux se produisent entre différents flux traversant le même voisinage ou entre différents paquets dans le même flux. Dans IEEE802.11, lorsque le nombre de tentatives d'accès au canal dépasse la limite prédéfinie, l'expéditeur assume que le récepteur est hors de portée et qu'il abandonne ses tentatives de transmission. Ensuite, le protocole MAC (Medium

Access Control) indique (à tort) à la couche supérieure que le chemin est indisponible. En réponse, la couche supérieure lance la procédure de récupération de la route [24] et TCP arrête sa transmission et le débit chute à zéro pendant le processus de récupération de route.

1.3.6.4 Paquet non séquentiel

L'un des inconvénients de TCP est le traitement inapproprié des paquets non séquentiels dans un réseau sans fil. Lorsqu'un destinataire reçoit un paquet non séquentiel, TCP suppose qu'une perte de paquet est survenue et le destinataire transmet l'accusé de réception en double, ce qui provoque la retransmission des paquets et l'activation du contrôle d'encombrement. En cas de réseau sans fil, des changements d'itinéraire fréquents se produisent et peuvent entraîner une remise de paquets en désordre. Il existe également un protocole de routage à trajets multiples qui mène également à une distribution de paquets non séquentielle. Cette situation entraîne l'activation inutile du contrôle de congestion et de la retransmission des paquets [25].

1.3.6.5 Contrôle d'accès au support (MAC)

L'utilisation de supports de diffusion et de transmission partagés introduit une probabilité non négligeable de collisions de paquets et de conflits de supports. De plus, la détection de collision radio en semi-duplex n'est pas possible, ce qui réduit considérablement l'utilisation du canal ainsi que le débit, et pose de nouveaux défis aux protocoles conventionnels CSMA / CD (Carrier-sense multiple access with collision detection) et aux protocoles MAC en général. Parmi les principaux problèmes, il y a les problèmes de terminaux cachés et exposés [4].

1.3.6.5.1 Problème des nœuds exposés

Considérez le réseau sans fil ci-dessous dans **Figure 4** ayant quatre nœuds étiquetés A, B, C et D, où les deux récepteurs sont hors de portée l'un de l'autre, alors que les deux émetteurs (B, C) situés au milieu sont à portée l'un de l'autre. Dans ce cas, si une transmission entre A et B est en cours, le nœud C ne peut pas transmettre à D, car il conclut après le portage de la porteuse qu'il interfère avec la transmission par son nœud voisin B. Toutefois, il convient de noter que le nœud D pourrait toujours recevoir la transmission de C sans interférence car il est hors de portée de B. Par conséquent, la mise en œuvre d'une antenne directionnelle au niveau d'une couche physique de chaque nœud pourrait réduire la probabilité d'interférence du signal, car le signal se propageait dans une bande étroite [26].

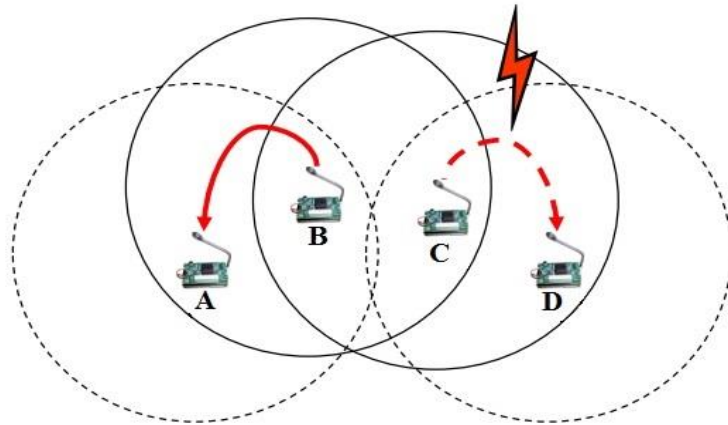


Figure 4 : *le phénomène des nœuds exposés [26]*

1.3.6.5.2 Problème des nœuds cachés

Maintenant considérons un réseau sans fil ci-dessous dans **Figure 5**, chaque nœud situé à l'extrémité de la plage du point d'accès, la plage de transmission du point d'accès A atteint en B, mais pas au point d'accès C, de même, la plage de transmission du point d'accès C atteint B, mais pas en A. Ces nœuds sont appelés terminaux cachés. Le problème se produit lorsque les nœuds A et C commencent à envoyer des paquets de données simultanément au point d'accès B. Les points d'accès A et C étant hors de portée les uns des autres, ils ne peuvent pas détecter une collision lors de la transmission. L'accès multiple par détection de porteuse avec détection de collision (CSMA / CD) ne fonctionne pas et des collisions se produisent, ce qui altère les données reçues par le point d'accès B en raison du problème du terminal caché [27] . De ce fait, le protocole TCP détecte les paquets corrompus et permet au contrôle de congestion de fonctionner, ce qui entraîne une baisse des performances du réseau.

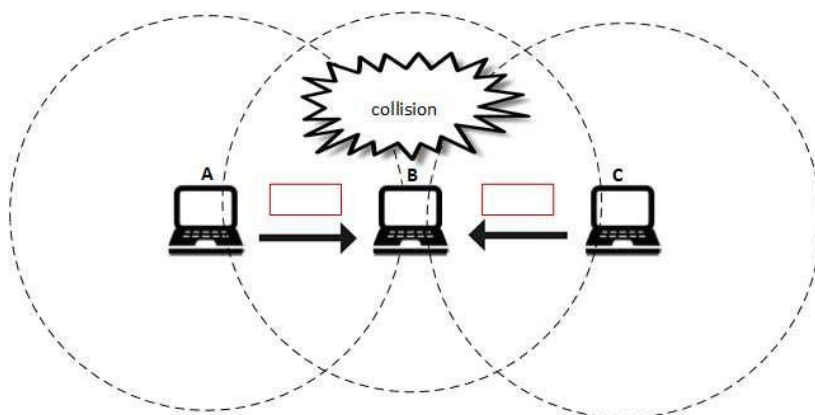


Figure 5 : *le phénomène des nœuds cachés [27]*

1.3.6.6 Impossible de différencier les pertes de transmission et de congestion `

Les réseaux sans fil ont un support partagé ouvert et lorsque le signal passe à travers le support, il peut faire face à différents obstacles et blocages lors de la construction et des gros objets en mouvement se rendant sur le chemin de destination et en raison de ces blocages les signaux de transmission sont reflétés, diffractés et diffusés et provoque ainsi la perte de paquets. TCP suppose que cette perte est survenue en raison de la congestion, il active son contrôle de congestion et réduit la fenêtre de congestion ce qui conduit à la dégradation des performances, un faible débit et faible utilisation de la bande passante disponible. Ainsi, TCP est incapable de différencier entre les pertes due à l'encombrement et les pertes due à la transmission pour éviter l'activation inutile du mécanisme de contrôle de congestion [25].

1.4 Système de positionnement global

La technologie de localisation la plus répandue est sans doute le GPS (Global positioning system). Il remplit très bien son rôle de système de localisation à l'échelle planétaire.

Le GPS appartient au département de la défense aux États-Unis. Son nom est NAVSTAR system (Navigation Satellite Timing and Ranging).

Il est composé de 3 parties ou segments :

- 1) Le segment spatial est composé d'au moins 24 satellites qui orbitent autour de la terre à une altitude d'approximativement 20 km. Chaque satellite fait 2 tours de la terre en 24h. Ils sont agencés de telle façon que chaque endroit (ou presque) de la terre ait à tout moment au moins 6 satellites en ligne de vue (en plein air). Ces trajectoires sont optimisées pour fournir la meilleure couverture dans la zone comprise entre la latitude 75° Nord et la latitude 75° Sud.
- 2) Le segment de contrôle est composé de plusieurs sites dispersés tout autour de la terre. Ils enregistrent les mouvements des satellites et envoient leurs données à une station centrale. Une ou deux fois par jour la station centrale transmet diverses informations aux satellites (mise à jour d'horloge, correction de trajectoire, ...).
- 3) Le segment des utilisateurs : les satellites émettent en continu des signaux radios. Ces signaux radios contiennent une description des trajectoires de chaque satellite. Les récepteurs GPS reçoivent ces informations et connaissent ainsi les positions des satellites. Ils calculent également les temps que prends l'onde radio pour arriver de chaque satellite

et en déduisent ainsi les distances jusqu'à chacun d'eux. Ils déduisent ensuite, en utilisant la multilatération et leurs positions. [28] [29] [30].

1.4.1 Coordonnées GPS

Les coordonnées GPS est un identifiant unique d'un lieu géographique spécifique, généralement exprimé en caractères alphanumériques. [29]

Les coordonnées, dans ce contexte, sont des points d'intersection dans un système de grille. Les coordonnées GPS (système de positionnement global) sont généralement exprimées sous la forme d'une combinaison de latitude et de longitude. La latitude est une mesure des degrés à partir de l'équateur, qui est 0 degrés. Les pôles nord et sud sont à 90 degrés dans une direction. Le méridien d'origine est la longitude 0 degré et les emplacements sont mesurés selon 90 degrés est ou ouest à partir de ce point.

Les coordonnées GPS sont exprimées dans deux formats différents.

Vous trouverez ci-dessous un exemple explicatif sur les deux formats :

Les coordonnées de l'Empire State Building à New York :

$N40^{\circ} 44.9064'$, $W073^{\circ} 59.0735'$

Cette même information peut être exprimée dans un format strictement numérique :

40.748440, -73.984559

Dans ce dernier cas, le signe moins indique que l'emplacement est situé au premier méridien ; un signe moins devant le premier chiffre indique les degrés au sud de l'équateur.

1.4.2 Comment fonctionne le GPS

Les satellites GPS [27] sont représentés à quatre endroits différents dans un ciel étoilé. Chacun émet un faisceau droit pointant vers le centre de l'affiche. Les quatre faisceaux convergent en un seul point sur le toit d'un tracteur vert. On voit un agriculteur conduire le tracteur à travers des rangées de cultures très serrées.

Les étapes principales du fonctionnement :

1. Les satellites GPS émettent des signaux radio indiquant leur position, leur statut et l'heure précise (T_1) à partir des horloges atomiques embarquées.
2. Les signaux radio GPS voyagent dans l'espace à la vitesse de la lumière (c), soit plus de $299\,792\text{ km / seconde}$.
3. Un appareil GPS reçoit les signaux radio, notant leur heure exacte d'arrivée (T_2), et les utilise pour calculer la distance qui le sépare de chaque satellite visible. Pour calculer

sa distance par rapport à un satellite, un appareil GPS applique cette formule au signal du satellite : $distance = fréquence * heure$, où fréquence est (c) et heure est la longueur du signal parcouru dans l'espace. Le temps de parcours du signal est la différence entre le temps diffusé par le satellite (T_1) et le moment où le signal est reçu (T_2).

4. Une fois qu'un GPS connaît sa distance par rapport à au moins quatre satellites, il peut utiliser la géométrie pour déterminer son emplacement sur la Terre en trois dimensions.

1.5 Composants et matériels

1.5.1 Arduino Uno

Arduino Uno est une carte à microcontrôleur mise à la terre sur l'ATmega328 (fiche technique). Il comprend 14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées en tant que sorties PWM (Pulse Width Modulation)), 6 entrées analogiques, un résonateur en céramique de 16 MHz, une facilitation pour la connectivité USB (Universal Serial Bus), une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation. Sa conception comprend des assistances qui supportent le microcontrôleur de toutes les manières possibles. Pour pouvoir l'utiliser, il suffit de le connecter à un ordinateur avec un câble USB ou de l'alimenter avec un adaptateur CA / CC ou une batterie.



Figure 6: La carte Arduino Uno [49]

Il est accessible en deux versions, à savoir Arduino Uno et Genuino Uno, qui peuvent être visualisées à la **figure 6**. Les variations sont observées par rapport à la région. L'Uno est unique par rapport à toutes ses anciennes cartes, car il n'utilise pas la puce de pilote FTDI USB-à-série.

Le mot "Uno" se réfère à un en italien et il est nommé ainsi pour marquer la prochaine version de Arduino 1.0. Uno et la version 1.0 sont les versions de référence d'Arduino dans le futur voyage. Le Uno est la dernière d'une série de cartes USB Arduino et le modèle de référence pour la plate-forme Arduino [31].

1.5.2 NEO-6M GPS

Le système utilise un récepteur de navigation GY-NEO6MV2, dont le module de traitement en bande de base utilise le module NEO-6 produit par la société u-blox. Ce module est principalement utilisé pour la navigation de véhicule en mouvement, il permet de visualiser en temps réel la position et la vitesse des véhicules. Généralement, la précision de positionnement est de $\pm 25m$. Comme le montre la **figure 7**, de droite à gauche, les broches du module de PIN1 à PIN4, une description détaillée de chaque broche est présentée dans le **Tableau** [32].



Figure 7: Module GPS NEO-6M [50]

Numéro de série	Nom	Description
1	RXD	Broche de réception en série
2	TXD	Broche de transmission série
3	GND	GND
4	VCC	VCC (3.3v-5.0v)

Tableau 1 : Description fonctionnelle de chaque broche du module GPS NEO-6 [33]

1.5.3 Schémas

Câblage du module GPS NEO-6M à Arduino en suivant le schéma ci-dessous :

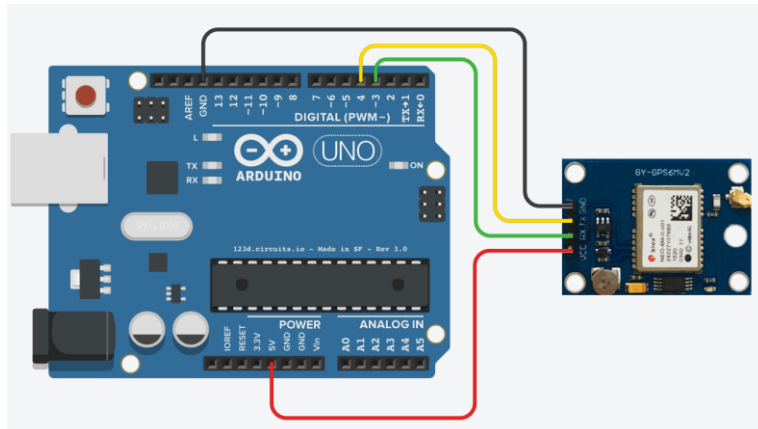


Figure 8 : Montage du Module GPS avec la carte Arduino [33]

- La broche GND du module est connectée à la broche GND Arduino.
- La broche RX du module est connectée à la broche 3 Arduino.
- La broche TX du module est connectée à la broche 4 Arduino.
- La broche VCC du module est connectée à la broche Arduino 5V.

1.6 Conclusion

Ce chapitre s'est focalisé sur les concepts des environnements mobiles. L'étude effectuée sur les réseaux sans fil et ces différents standards. En ajoutant, Une définition générale du protocole TCP, et ces mécanismes de contrôle de fiabilité (contrôle de flux, contrôle de congestion) et les algorithmes de Slow Start, Congestion Avoidance, AIMD.

Ce chapitre a présenté aussi quelques versions déjà proposées pour améliorer la performance de contrôle de congestion (TCP Tahoe, TCP Reno, TCP New Reno.....) et une définition générale sur le GPS et comment fonctionne.

Étant donné que le sujet principal de ce mémoire est le contrôle de congestion dans le contexte de la mobilité, le chapitre suivant présentera, de façon un peu détaillée, quelques solutions proposées dans la littérature.

Chapitre

2

État de l'art

2.1 Introduction

Les réseaux sans fil à mono-saut ont une infrastructure de réseau principal fixe, des points d'accès fixes et des stations de base. L'objectif le plus important dans ce type de réseau est de garantir la fiabilité de la transmission des données en raison des limites actuelles des communications sans fil. Le protocole de contrôle de transmission (TCP) est très important pour une transmission fiable des données.

Le protocole TCP a été initialement introduit pour les réseaux câblés où la congestion du réseau est la principale cause de perte de paquets. TCP fonctionne sur les hypothèses qui ne conviennent qu'aux réseaux câblés. Les réseaux câblés ont moins de chances de perdre des paquets en raison d'autres problèmes tels qu'erreurs de transmission et échecs de route par rapport à la congestion du réseau. TCP se concentre principalement sur le contrôle d'encombrement et considère toute perte de paquet comme une cause d'encombrement du réseau seulement. Par la suite, il réduit le débit de transmission pour éviter une congestion supplémentaire. L'interprétation par défaut de TCP de toute perte de paquet en tant que cause de congestion du réseau convient bien aux réseaux câblés mais pas aux réseaux sans fil [34].

TCP souffre d'une dégradation des performances dans les environnements sans fil parce qu'il doit faire face à des défis en raison de pertes de paquets ou paquets corrompus dues à des problèmes de transmission sans fil comme l'atténuation, bruit, contention et la mobilité. Les nœuds peuvent perdre des paquets en raison d'échecs de route ou à cause de la corruption, dans ce cas malheureusement TCP considère ces pertes comme une indication d'encombrement et par conséquent invoque le contrôle de congestion qui conduisant lui-même à une performance dégradée. L'impossibilité de différencier diverses pertes pour agir en conséquence est l'un des problèmes les plus critiques dans les réseaux sans fil.

Dans ce chapitre on se concentre sur l'exploration des différentes variantes de TCP qui ont été proposées.

2.2 Des mécanismes pour l'optimisation de protocole TCP dans les réseaux sans fil

Dans cette partie, nous examinerons les mécanismes d'optimisation correspondants de plusieurs approches améliorées.

2.2.1 TCP MUZHA

Le protocole TCP MUZHA [35] utilise des routeurs intermédiaires pour le contrôle d'encombrement entre la source et la destination, avec les informations des routeurs, cette

approche ajuste de manière dynamique la taille de la fenêtre d'encombrement et gère la perte de paquets aléatoire. L'approche utilise la bande passante en tant que paramètre, chaque routeur calcule lui-même la bande passante disponible et recommandent aux flux de trafic qui passent d'augmenter ou de diminuer leurs débits de données. Chaque routeur représente sa propre bande passante par un index appelé, DRAI (Data Rate Adjustment Index) et publie cette information avec les autres TCP session courante le DRAI peut prendre 5 valeur entière et chaque valeur représente un état de transmission comme indiqué dans le tableau suivant :

DRI	ETAT	Changement de fenêtre de congestion
5	Accélération agressive	$CWND = CWND * 2$
4	Accélération modérée	$CWND = CWND + 1$
3	Stabilisation	$CWND = CWND$
2	Décélération modérée	$CWND = CWND - 1$
1	Décélération agressive	$CWND = CWND * 1/2$

Tableau 2 : Formule de DRAI [35]

Il y'a une autre valeur appelée **MRAI** (Minimum indice d'ajustement du débit de données) en utilisant cette variable MRAI, l'expéditeur met à l'échelle la taille de la fenêtre d'envoi en l'augmentant ou en la diminuant donc l'expéditeur des données et les routeurs à la décision d'adaptation du débit de données.

Principalement les auteurs de cette approche ont ajoutés la valeur de **MRAI** dans l'en-tête du paquet IP. Ensuite l'expéditeur des données met à jour la valeur **MRAI** à une valeur maximale pour chaque paquet qu'il envoie plus tard, chaque nœud intercepte un paquet et compare son propre **DRAI** à cette valeur **MRAI** et le remplace si sa valeur est plus petite, maintenant lorsque le destinataire recevra les paquets, il notera la valeur minimale de **DRAI** et informera l'expéditeur de cette information en lui envoyant un accusé de réception. Par conséquent, l'expéditeur peut utiliser cette information pour régler son débit de données et éditer la taille de sa fenêtre de congestion dans le cas où il 'Ya une perte des paquets expéditeur

reçoit ACK avec un indice de décélération du débit, dans ce cas, l'expéditeur sait que la perte est due à un encombrement, sinon cette perte de paquets sera considérée comme une perte aléatoire et l'expéditeur transmettra les paquets perdus sans réduction de la fenêtre de congestion.

le protocole utilise deux phase pour le control de congestion ,la première est l'évitement de congestion et la deuxième Récupération et retransmission rapide, lorsque le protocole initial une connexion , il entre directement dans la phase de l'évitement de congestion, quand l'expéditeur reçoit la première ACK il met à jour la taille du CWND selon le **MRAI** ,dans le cas où il y'a une congestion et il reçoit trois ACK dupliquées , le protocole rentre dans la phase Récupération et retransmission rapide et réduit CWND à un demi ,Si le délai de transmission expire ,il met la valeur de CWND a un. Le diagramme ci-dessous représente les différentes phases de TCP MUZHA.

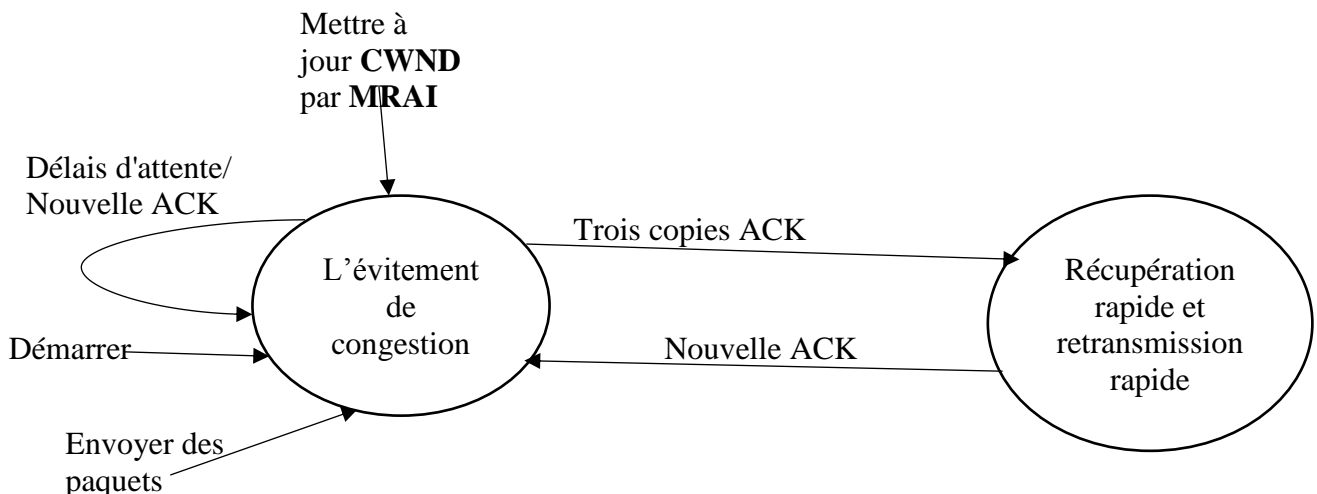


Figure 9 : *Diagramme de transition de phase de TCP Muzha [35]*

- **Les avantages de cette solution sont :**

- TCP MUZHA offre de bonnes performances en raison de son mécanisme d'ajustement dynamique du débit de données.
- Réduire l'occurrence de congestion.

- **Les inconvénients de cette solution sont :**

- Lorsqu'un paquet traverse le réseau d'un routeur à tous les autres routeurs, le débit du paquet peut donc être réduit.
- TCP MUZHA souffre énormément dans un environnement multi-protocoles car il ne peut pas partager la bande passante de manière équitable, ce qui entraîne des difficultés pour la mise en œuvre réelle.

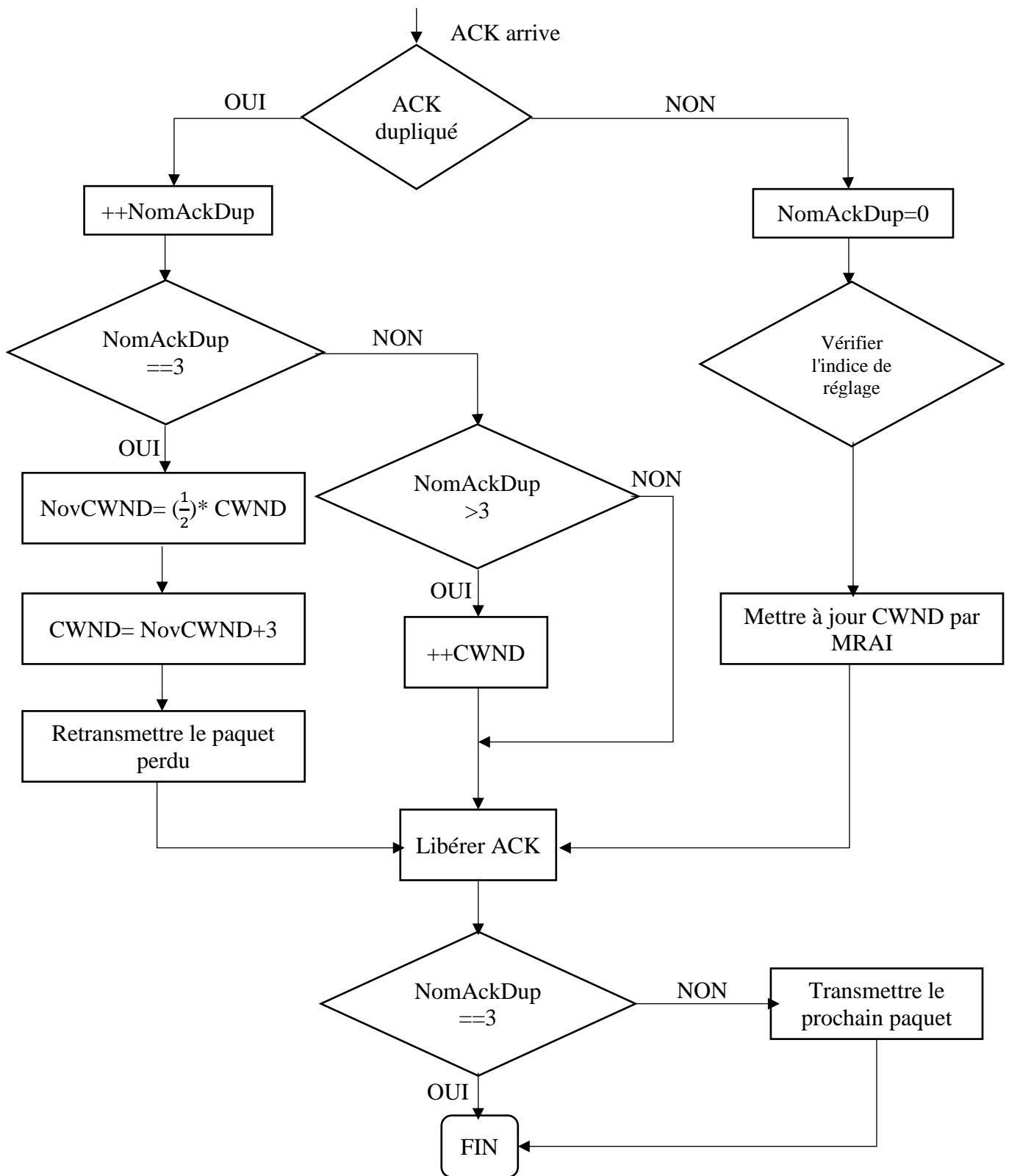


Figure 10: Organigramme de la procédure pour TCP Muzha [35] lors de la réception d'un ACK

2.2.2 TCP NRT

Cette solution [36] permet à TCP de reconnaître les événements de non-congestion qui entraînent la perte des paquets et le retard d'accusé de réception, TCP NRT identifie le délai de transmission expirés RTO (retransmission timeout) a deux catégories, la première est connu comme congestion RTO et la deuxième non-congestion RTO de même le non-congestion RTO peut être divisé à un RTO à cause des pertes des paquets et un RTO faux ou parasite à cause des retards soudains des paquets en cours de la communication. Cet algorithme sera activé par le côté émetteur lorsque le RTO apparaît, sinon l'expéditeur réagira de la même manière que le TCP classique. Le TCP NRT fonctionne selon trois phases : la phase de détection, la phase de différenciation de non-congestion RTO et la phase de réaction.

2.2.2.1 La phase de détection

Le but de cette étape est de faire la distinction entre les RTOs faux à cause des retards soudains des paquets et les RTOs à cause de la perte des paquets afin de résoudre ce problème, premièrement ils devaient trouver une solution pour distinguer entre les RTOs de congestion et les RTOs de non-congestion, les auteurs ont modifié un mécanisme appelé ECN (Explicit Congestion Notification) [37]. Cette dernière est recommandée d'être largement utilisé comme mécanisme de routeur.

Dans le TCP classique, les routeurs fournissent des signaux de perte de paquets, il conduit le réseau à une congestion en laissant tomber les paquets puis en les récupérant de sorte que l'expéditeur a besoin de voir la perte pour ralentir la fenêtre d'envoi , cette méthode fonctionne plutôt bien mais pour construire un très bon réseau avec un faible délai et une faible perte, il vaut mieux éviter la congestion et ceci est l'une des raisons pour lesquelles les développeurs ont utilisé ECN (Explicit Congestion Notification) dans ce travail cette méthode permet de signaler l'encombrement du réseau avant que la perte de paquets ne se produise par marquer les paquets pour alerter l'expéditeur de la congestion naissante , pour mieux comprendre en premier lieu nous expliquons ce processus puis comment ils l'ont modifié.

Un routeur compatible ECN est configuré avec trois paramètres principaux. Ils sont le seuil minimum min^{th} , seuil maximum max^{th} et le maximum probabilité de marquage P_{max} . Quand un paquet arrive au routeur, ECN calcule la taille moyenne de la file d'attente AQL puis faire des tests de comparaisons, si AQL est inférieure à min^{th} le routeur ne marque pas le paquet. Si AQL dépasse min^{th} et inférieure à max^{th} le routeur marque le paquet avec probabilité.

$$P = \left(\frac{AQL - min^{th}}{max^{th} - min^{th}} \right) P_{max}$$

Comme résultat, l'expéditeur réduit la taille de la fenêtre de congestion pour contrôler le débit d'envoi à l'avance avant le réseau devient lourdement encombré, si **AQL** est supérieur à **maxth**

Le routeur marque le paquet avec une probabilité égale à un. La congestion dans les réseaux sans fil est rare donc il n'est pas nécessaire de marquer et réduire la taille de la fenêtre de congestion quand **AQL** est entre **minth** et **maxth** en conséquence, cela ralentit le débit d'envoi et le réseau ne peut utiliser pleinement la bande passante. Pour éviter le ralentissement du débit d'envoi le TCP NRT marque les paquets uniquement lorsque **AQL** est supérieur ou égale à **maxth** à cause de cette modification sur ECN, l'expéditeur peut envoyer plus de paquets que l'ECN d'origine sans réduire la taille de la fenêtre de congestion inutilement avant que l'expéditeur détecte une perte de paquet par RTOs où Acks dupliqué. Quand le RTO a expiré pendant la période de transmission des paquets, l'expéditeur utilise l'ECN modifié pour vérifier si le dernier Ack a été marquée ou non donc si l'accusé de réception (Ack) est marqué cela signifie qu'il y a une congestion dans le réseau et le paquet correspondant est perdu autrement le RTO a expiré en raison d'un événement de non-congestion comme des pertes de paquets aléatoires ou des RTO parasite dus à des retards soudains.

2.2.2.2 La différenciation de non-congestion RTOs

TCP NRT différencie non-congestion RTOs en RTOs parasite et RTOs en raison de la perte de paquets, quand RTO expire du côté de l'expéditeur, il vérifie si le dernier accusé de réception (ACK) reçu est marqué ou non, si ACK n'est pas marqué cela signifie que RTO expiré en raison du perte des paquets ou il y'a un RTO faux dans ce cas l'expéditeur stock le numéro de séquence du prochain ACK attendu dans une variable Seq_Exp_Ack . Dans le cadre du processus de différenciation, un nouveau paquet est envoyé et lorsque l'expéditeur reçoit un accusé de réception (ACK) son numéro de séquence Seq_New_Ack, il vérifie si (Seq_Exp_Ack < Seq_New_Ack) alors le RTO était fausse sinon le RTO était en raison de perte de paquet aléatoire donc l'expéditeur retransmet le paquet perdu immédiatement sans réduire la taille de cwnd.

2.2.2.3 La phase de réaction

Le protocole réagit à la perte de paquet après l'identifier comme ci-dessous :

- Sur RTO de congestion, le mécanisme de contrôle de congestion de TCP New Reno est activé.
- Sur RTO parasite, les nouveaux paquets sont envoyés en fonction de la valeur actuelle de Cwnd.
- Sur RTO en raison d'une perte de paquet aléatoire, le paquet perdu est retransmis et la transmission se poursuit sans changement de Cwnd.
- **Les avantages de cette solution sont :**
 - La détection précoce de l'encombrement et la réduction des pertes et des retards de paquets.
 - Les routeurs envoient des signaux clairs aux hôtes pour gérer la congestion.
 - Une méthode systématique de gérer les pertes. L'ECN modifié est plus efficace par rapport à ECN
- **Les inconvénients de cette solution sont :**
 - Les routeurs et les hôtes doivent être mis à niveau.
 - L'activation de la gestion des pertes est tard (délai d'attente RTO uniquement)
 - La perte potentielle de messages ECN sur le réseau. Dans ce cas, le message de notification d'encombrement peut ne pas atteindre le nœud final.

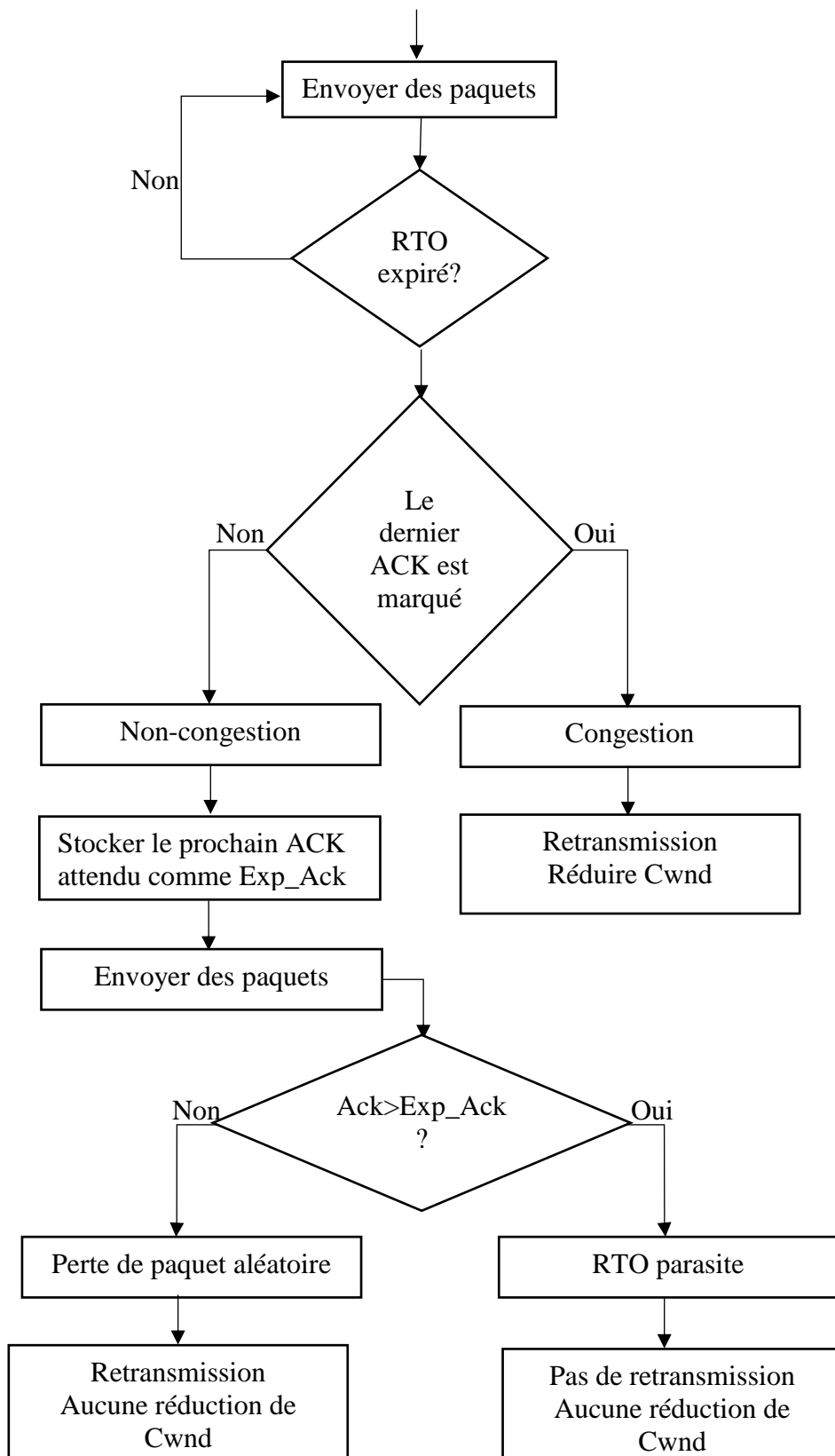


Figure 11 : Organigramme de TCP NRT [38]

2.2.3 Solution En Mono-Saut

Dans cette section, nous nous concentrerons sur l'approche proposée [1] afin de trouver une solution permettant d'améliorer les performances TCP dans le réseau sans fil en mono-saut, elle doit être réalisée dans un environnement réel sur des machines véritable. Le but de ce mécanisme est de résoudre le problème de la perte de paquets en raison de la mobilité du périphérique en utilisant la puissance du signal. Le récepteur intercepte la valeur de la puissance du signal (RSSI), cette dernière peut être récupérée à partir de la couche physique du modèle OSI. Les auteurs ont proposé une formule probabiliste permettant d'estimer la prochaine valeur du signal en utilisant les 5 valeurs de la puissance du signal précédentes. La récupération se fait par intervalle de 3 secondes entre chaque valeur, l'intervalle choisit est suffisant pour impacter la puissance du signal. L'interception était comme ça : récupérer la première valeur de signal $V1$ puis attendre trois secondes pour récupérer la deuxième valeur de signal $V2$ puis attendre trois secondes pour récupérer la prochaine valeur de signal $V3$ ainsi de suite jusqu'à obtenir 5 valeurs de signal. Après l'évaluation des valeurs du signal et la moyenne de variation de la puissance du signal, l'expéditeur peut prédire la prochaine valeur du signal (RSSI). TCP utilise cette valeur d'estimation pour faire comparer au seuil de la puissance du signal (chaque carte réseau est caractérisée par une valeur du seuil et peut être obtenue auprès du fabricant) pour différencier entre la perte due à l'erreur de lien et la perte due à la congestion. Ils ont étudié le changement de la puissance de signal puis calculer la moyenne de variation de la puissance du signal et faire l'estimation de la prochaine puissance du signal à l'aide des formules suivantes :

-Calculer les valeurs des changements du signal est comme suit :

$$V1 = S2 - S1$$

$$V2 = S3 - S2$$

$$V3 = S4 - S3$$

-Calculer la moyenne de variation de la puissance du signal est comme suit :

$$MVPS = (V1 + V2 + V3) / 3$$

-L'estimation de la prochaine puissance du signal est comme suit :

$$PSE = S5 + MVPS$$

Après avoir obtenu la prochaine valeur du signal, ils ont fait des comparaisons par rapport à la valeur de seuil de l'interface réseau. Si la valeur **PSE** (puissance du signal estimé) supérieure à la valeur de seuil alors la perte est due à la mobilité de nœud et par conséquent désactiver temporairement la mise à l'échelle de la fenêtre TCP (*window scaling* = 0) et la

bande passante maximale devient une question de distance. Sinon la perte des paquets est causée par une congestion du réseau. Dans ce cas Le TCP mettre à jour la mise à l'échelle par défaut de la fenêtre d'envoi (*window scaling* = 1).

- **Les avantages de cette solution sont :**

- Ce mécanisme diminue le nombre de paquets rejetés et augmente le nombre de paquets transmis avec succès à leur destination finale
- une approche simple et facile à implémenté

- **Les inconvénients de cette solution sont :**

- la vitesse de déplacement du nœud égale à la vitesse d'un être Human, elle est constante et ne change pas.
- plus les nœuds sont loin les uns des autres, plus la puissance du signal reçu devient faible

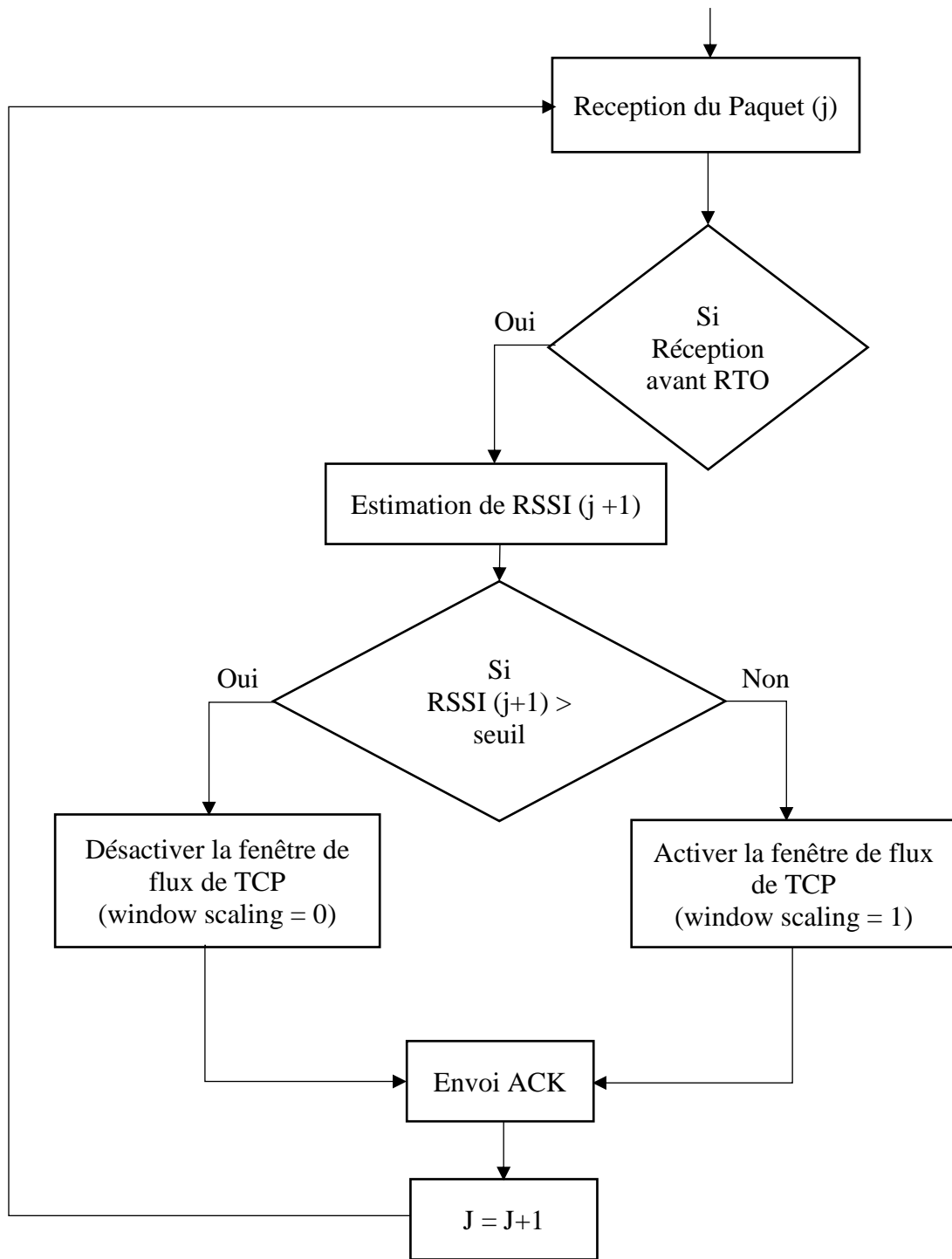


Figure 12: Organigramme de Solution En Mono-Saut [1]

2.2.4 TCP-WELCOME

Cette approche [38] est la solution de bout en bout qui ne nécessite pas la coopération de nœuds intermédiaires, elle est basée sur l'estimation la quantité de changements dans les valeurs RTT (Round Trip Time) au niveau de côté émetteur afin de faire la différence entre les pertes des paquets à cause de congestion, les pertes à cause d'échec de route et les pertes à cause de perte du canal. TCP WELCOME comprend deux parties : l'algorithme de différenciation des pertes et l'algorithme de récupération des pertes.

2.2.4.1 Algorithme de différenciation LDA-Loss

L'expéditeur active l'algorithme LDA-LOSS lorsqu'il reçoit 3 ACK dupliquées ou les évènements de temporisation du RTO. TCP-WELCOME analyse les valeurs de RTO et l'algorithme LDA-LOSS utilise cette information pour différencier les raisons de la perte. Si les valeurs RTT augmentent continuellement avec le temps, une perte est considérée comme une cause d'encombrement du réseau, quel que soit le type d'événement (3 ACK dupliquées ou temporisation du RTO). Si les valeurs RTT ne changent pas et restent presque stables autour de la moyenne, une perte est considérée comme une cause d'un événement de non-congestion. TCP WELCOME différencie en outre la perte de non-congestion en deux catégories : perte d'échec de route ou perte de canal. Si l'expéditeur TCP reçoit 3 ACK dupliquées alors la cause est supposée être une perte de canal. Si l'expéditeur TCP est confronté à un délai d'attente RTO, la cause est supposée être une perte d'échec de route.

2.2.4.2 Algorithme de récupération LRA-Loss

TCP WELCOME retransmet immédiatement le paquet perdu dans tous les cas. Ensuite, il prend les mesures appropriées pour la valeur du temporisateur RTO et le débit de transmission. La valeur du temporisateur RTO et le débit de transmission sont modifiés conformément au schéma de contrôle d'encombrement en cas de perte d'encombrement. La valeur du temporisateur RTO et le débit de transmission ne sont pas modifiés en cas de perte de canal. La valeur du temporisateur RTO et le débit de transmission sont modifiés conformément aux statistiques du nouvelle route en cas de perte de route. RTT_{New} correspond à la nouvelle route et RTT_{Old} à la route précédente.

$$RTO_{New} = (RTT_{New}/RTT_{Old}) * RTO_{Old}$$
$$CWND_{New} = (RTO_{Old}/RTO_{New}) * CWND_{Old}$$

- **Les avantages de cette solution sont :**

- Les valeurs des RTT sont utilisées pour différencier entre les événements de 3 ACK dupliquées et les événements de temporisation du RTO.
- Facile de mettre en œuvre avec l'un des variantes du TCP basées sur AIMD.

- **Les inconvénients de cette solution sont :**

- Algorithme de récupération de perte est moins efficace car il utilise un seul nouveau valeur RTT pour définir RTO et taux de transmission et la prise de décision basée sur RTT est inexacte.

2.2.5 Solution en Multi-Saut

Cette approche [39] est l'implémentation du Solution En Mono-Saut sur un réseau sans fil en multi-saut dans le cadre d'estimation la valeur de la puissance du signal minimale sur les réseaux sans fils multi-saut et bien sur l'objectif principal était et sera toujours l'amélioration des performances de TCP sur les réseaux mobiles et ad hoc. Les auteurs de cette solution ont développé un mécanisme qui fait l'estimation et la prédiction de la puissance du signal minimale d'un paquet en se basant sur les valeurs précédentes dans un environnement sans fil multi-sauts. Lors se qu'une perte de paquet se produire, la valeur estimée va leurs permettre de déterminer la cause de cette perte de paquet (congestion ou mobilité). Ils l'ont émulé dans un environnement réel avec un routage statique. Ce mécanisme fonctionne selon deux étapes principales, la première consiste à récupérer la puissance minimale de toutes les puissances estimées dans les nœuds, deuxièmement l'envoyer à un nœud de traitement aléatoire qui prendra la décision de contrôler la fenêtre de flux de TCP selon la valeur de la puissance minimal.

Chaque un des nœuds intercepte la puissance du signal (RSSI) à partir de son propre interface réseaux et commence par faire une estimation de la prochaine valeur. Les auteurs ont utilisé la formule probabiliste du l'approche « Solution Mono-Saut » [1] pour estimer la prochaine valeur du signal en utilisant les 5 valeurs de la puissance du signal précédentes. Ils ont supposé que la vitesse de déplacement soit moyennement celle d'un être humain. La récupération se fait par intervalle de 3 secondes entre chaque valeur, l'intervalle choisit est suffisante pour impacter les puissances du signal reçu (S1, S2, S3, S4, S5). Ensuite, ils ont étudié les changements du signal selon les relations suivantes :

- Changement du signal: $V1 = S4 - S3$; $V2 = S3 - S2$; $V3 = S2 - S1$
- Moyenne de variation de la puissance du signal : $MVPS = ((V1 + V2 + V3)) / 3$

- Prédiction de la prochaine puissance du signal : $RSSI = S5 + MVPS$

Tous les nœuds envoient leur propre puissance de signal estimée à un nœud maître qui calculera la puissance de signal minimale $minRSSI$ après que ce nœud maître renverra cette valeur minimale à tous les nœuds du réseau afin qu'ils puissent l'utiliser pour contrôler leur propre fenêtre de flux. Après avoir obtenu la valeur du signal, chaque un des nœuds fait des comparaisons par rapport à la valeur de seuil de l'interface réseau (chaque carte réseau est caractérisée par une valeur du seuil et peut être obtenue auprès du fabricant). Si la valeur $minRSSI$ (minimale puissance du signal estimé) supérieure à la valeur de seuil alors la perte est due à la mobilité de nœud et par conséquent désactiver temporairement la mise à l'échelle de la fenêtre TCP ($window\ scaling = 0$) et la bande passante maximale devient une question de distance. Sinon la perte des paquets est causée par une congestion du réseau. Dans ce cas Le TCP mettre à jour la mise à l'échelle par défaut de la fenêtre d'envoi ($window\ scaling = 1$).

- **Les avantages de cette solution :**

- La bande passante n'est pas encombrée avec des messages d'information et de routage et par conséquent un bon débit.
- Cette approche est facile à mettre en œuvre
- Aucun cycle de processeur n'est utilisé pour calculer et communiquer des routes

- **Les désavantages de cette solution :**

- Dans le cas où la machine de traitement tombe en panne ou quitter le réseau, le mécanisme cesse de fonctionner, un mécanisme de choix dynamique pour choisir la machine de traitement aléatoire doit être implémenté.
- La vitesse du déplacement des nœuds est celle d'un être humain.
- Plus les nœuds sont loin les uns des autres, plus la puissance du signal reçu devient faible.

2.3 Critique et comparaison

Cinq approches ont été étudiées et examinées dans ce chapitre. Chaque solution a des avantages et des inconvénients. La plupart de ces solutions sont des approches inter couches récentes et chacune d'entre elles dispose d'une manière spécifique de traiter le problème et d'agir différemment en cas des pertes de paquets dues aux événements de congestion ou de non-congestion afin d'améliorer les performances de TCP dans les réseaux sans fil.

Le tableau ci-dessous représente une comparaison entre les différentes approches étudiées.

Approches Critères	TCP- MUZHA	TCP-NRT	Solution en mono-saut	TCP- WELCOME	Solution en multi-saut
Type de contrôle de congestion	Contrôle de congestion assisté par réseau	Contrôle de congestion assisté par réseau	Contrôle de la congestion bout à bout	Contrôle de la congestion bout à bout	Contrôle de la congestion bout à bout
Paramètre d'estimation	Le débit	RTOs	RSSI	RTT	RSSI
Différenciation des pertes	Considérer toutes les pertes en raison d'une congestion	Détecte les événements du non-congestion	Entre la mobilité et la congestion	Détecte les événements du non- congestion	Entre la mobilité et la congestion
Evaluation	Simulation	Simulation	Emulation	Simulation	Emulation
Degré de complexité	Haut	Haut	Moyen	Moyen	Moyen
Congestion	Contrôle de congestion modifié	ECN est utilisé pour notifier l'expéditeur TCP. Le contrôle de congestion est identique à TCP	Le contrôle de congestion est identique à TCP	Le contrôle de congestion est identique à TCP	Le contrôle de congestion est identique à TCP
Paquets mis en mémoire tampon aux nœuds intermédiaires	Oui	Oui	Non	Non	Oui
Couche impliquée	Couche réseau et transport	Couche réseau et transport	Couche application	Couche réseau	Couche application
Type de communication	Multi-sauts	Multi-sauts	Mono-sauts	Multi-sauts	Multi-sauts

Tableau 3 : Comparaison entre les différentes approches de TCP [23] [4]

2.4 Conclusion

Les réseaux sans fil posent de gros problèmes au protocole TCP car il n'a pas été conçu pour fonctionner dans un tel environnement où les nœuds sont mobiles et le taux d'erreur sur les bits (BER) est généralement assez élevé. L'amélioration de TCP dans les réseaux sans fil est devenue une tâche importante pour garantir la fiabilité de la transmission des données et pour surmonter ces obstacles.

Dans ce chapitre, nous avons décrit quelques approches proposées pour améliorer les performances du protocole TCP dans les réseaux sans fil. L'une de ces approches était d'améliorer le débit de transfert en se basant sur la prochaine valeur de la puissance du signal en mono-saut [1]. Cette approche a permis à TCP de différencier entre les pertes dues à l'environnement et les pertes dues à la congestion. Cependant, l'un des majeurs inconvénients de cette solution c'est que les auteurs ont basé leurs calculs sur une vitesse de déplacement des nœuds mobile qui est fixe (vitesse moyenne de déplacement d'un être humain qui marche). **Le cadre de ce projet est d'améliorer cette approche en ajoutant des mécanismes d'adaptation tel que la considération des vitesses variables de déplacement des nœuds.**

Chapitre

3

Conception

3.1 Introduction

Le but de ce chapitre est de trouver une solution pour améliorer les performances de TCP dans les réseaux sans fil. Nous avons étudié des différentes approches proposées dans le chapitre précédent et l'une de ces approches était d'améliorer les performances de protocole TCP en se basant sur la puissance de signal en mono-saut [1]. Cette solution a donné de bons résultats mais elle pose un inconvénient est que cela fonctionne selon la vitesse d'un être humain. Notre objectif consiste à trouver une méthode pour rendre l'approche fonctionnelle selon des différentes vitesses puis l'émuler dans un environnement réel en mono-saut pour l'évaluation.

3.2 Description globale de l'approche

3.2.1 Un aperçu sur la solution

La solution que nous proposons vise à prédire la prochaine valeur de puissance du signal en se basant sur les valeurs de puissances du signal précédentes afin de maîtriser le mécanisme de perte de paquet et d'ajuster le mécanisme d'encombrement de TCP dans le réseau sans fil. Les valeurs des signaux sont capturées avec un intervalle de temps approprié qu'est déterminé par la nature de la vitesse du nœud mobile. Le nœud de traitement utilise la vitesse du nœud mobile pour définir le temps d'attente entre deux récupérations de la puissance de signal et estime le signal plus tard. Les sections suivantes expliqueront brièvement la solution.

3.2.2 Récupération de la valeur de la vitesse

La valeur de la vitesse reçue est récupérée par un récepteur distinctif au niveau du nœud mobile et elle est mesurée en kilomètres par heure (km/h). Cette valeur sera envoyée au nœud traitante qui a son tour déterminera l'intervalle du temps idéal et l'estimation de la prochaine valeur du signal.

3.2.3 Calcule d'intervalle de temps

Dans cette partie on veut récupérer l'intervalle du temps suffisant pour impacter le changement de signal. L'intervalle du temps est une variable qui change fréquemment en fonction de la valeur de la vitesse du nœud, l'un dépend de l'autre, lorsque l'une des deux variables décroît l'autre augmente. La valeur de l'intervalle du temps sera utilisée entre deux récupérations consécutives de la puissance du signal. Pour faciliter le travail, on propose une structure des données qui contient des différentes vitesses avec l'intervalle de temps correspondant comme suite :

<i>Vitesse(km/h)</i>	V_1	V_2	V_3	$V_4 \dots \dots V_n$
<i>Temps(s)</i>	T_1	T_2	T_3	$T_4 \dots \dots T_n$

Tableau 4 : *La structure de données utilisée*

Après avoir reçu la vitesse du nœud mobile, on la passe en tant que paramètre d'une fonction de recherche qui recherche l'intervalle de temps correspondant dans le tableau, après avoir obtenu l'intervalle du temps, ensuite, nous pouvons l'utiliser pour estimer la puissance du signal.

3.2.4 Récupération de la puissance de signal

La valeur de la puissance du signal peut être récupérée à partir de la couche physique du modèle OSI à l'aide d'un récepteur. Elle est mesurée en dbm qui est le degré de puissance mesurée en décibels (**dB**) par rapport à un milliwatt (**mW**).

3.2.5 Formule d'estimation de la prochaine puissance du signal

Tout d'abord, en supposant que la vitesse moyenne du nœud mobile est variable. La première tâche à réaliser est de prédire la prochaine valeur de la puissance du signal à l'aide des **N** valeurs de puissance du signal précédentes. Premièrement pour un nombre donné de puissances de signal capturées **N**, nous commençons à récupérer la première valeur de la puissance du signal (S_1) puis nous attendons un instant (on obtient l'intervalle de temps à l'aide de la méthode que nous avons décrite dans la section précédente), puis nous continuons à récupérer la deuxième valeur de la puissance du signal (S_2) et faire la même chose que nous avons fait avant, ainsi de suite on répète le même processus jusqu'à ce que nous récupérons **N** valeurs de puissance du signal ($S_1, S_2, S_3 \dots \dots S_N$). Les intervalles de temps que nous avons utilisés entre deux valeurs sont suffisants pour impacter le changement du signal.

Remarque : le nombre donné de puissance de signal capturé ne peut pas être inférieur à quatre ($N \geq 4$ et $N \in \mathbb{N}$), car dans notre solution, nous avons besoin d'au moins un nombre minimal de valeurs de la variation égal à deux pour pouvoir estimer la puissance de signal suivante.

Après avoir obtenu le nombre requis de puissance du signal capturé **N**, le moment est venu de faire une étude sur le changement de la puissance du signal, nous étudions les changements de la puissance du signal selon les relations ci-dessous :

$$Var_1 = S_2 - S_1 \dots \dots \dots (1)$$

$$Var_2 = S_3 - S_2 \dots \dots \dots (2)$$

$$Var_{N-2} = S_{N-1} - S_{N-2} \dots \dots \dots (N - 2)$$

Ensuite, nous déterminons la moyenne de la variation de la puissance du signal à l'aide de la formule ci-dessous afin de prédire la prochaine puissance du signal :

$$MVPS = \sum_{i=1}^{k=N-2} \frac{Var_i}{k}$$

Enfin, nous combinons la moyenne de la variation de la puissance du signal et la ($N^{ème}$) puissance du signal (S_N) pour prédire approximativement la valeur de la prochaine puissance du signal (S_{N+1}) comme suit :

$$VPPS = S_N + MVPS$$

Nous définissons comme suit :

- **N** : le nombre de puissances de signaux à capturer.
- **S** : la valeur de la puissance du signal.
- **Var** : La valeur de variation entre deux puissances du signal consécutives.
- **MVPS** : La moyenne de la variation de la puissance du signal.
- **VPPS** : la valeur de la prochaine puissance du signal.

Suite à la conclusion de la prochaine valeur de **RSSI**. Dans le cas où il y'a une perte des paquets, Le protocole TCP utilise cette valeur en la comparant au seuil de puissance du signal pour déterminer la cause de perte des paquets (évènement d'encombrement ou mobilité) :

- Si **VPPS** (la valeur de puissance du signal estimée) est supérieur au seuil de l'interface réseau. TCP déduira que la perte est due à la mobilité, dans ce cas le mécanisme désactive la fenêtre de flux de TCP temporairement afin de garder un débit élevé.
- De l'autre côté dans le cas contraire, si la valeur de puissance du signal estimé est inférieure ou égale à la valeur du seuil de puissance du signal, le TCP conclut que la perte de paquets est survenue à cause d'un encombrement du réseau et par conséquent le mécanisme active la fenêtre de flux de TCP temporairement.

Remarque : La valeur du seuil de puissance de signal de la carte réseau est donnée par son constructeur.

3.2.6 La précision d'estimation de la puissance du signal

D'un point de vue mathématique, les modèles de prédiction donnent des résultats approximatifs, la formule probabiliste que nous avons proposée dans la section précédente nous permet de prédire la valeur suivante de la puissance du signal en se basant sur N valeurs de puissance du signal précédentes, mais parfois la valeur de prédiction n'est pas précise et ce genre de problème dans la prédiction appelée erreur de prédiction. Dans cette section, nous souhaitons aider notre modèle de prévision en corrigeant l'erreur de prédiction afin qu'il puisse donner des résultats plus précis.

D'abord nous proposons une méthode adaptative pour aider notre modèle d'estimation, lorsque nous obtenons la valeur estimée de la prochaine puissance du signal $E_{S_{N+1}}$, nous la comparons à la valeur réelle S_{N+1} si les deux valeurs sont trop éloignées l'une de l'autre on corrige cette erreur en ajoutant plus de valeurs capturées au nombre initial de valeurs capturées N , sinon nous vérifions si nous avons corrigé l'erreur avant et si oui, nous diminuons le nombre de valeurs capturées afin d'atteindre l'état de départ avec N de puissance du signal à capturer.

3.3 Algorithme descriptif

Dans cette section, nous décrivons nos algorithmes.

3.3.1 Algorithme de récupération de l'intervalle de temps

Algorithme 1 : Fonction d'obtenir la valeur de l'intervalle de temps

Fonction Obtenir_intervalle_de_temps () : Entier ;

Variables :

$i, n, \text{Timer}, \text{Vitesse}$: Entier ;

$\text{Vit_Temps} [][]$: Matrice d'entier ;

Trouve : Booléen ;

Début :

Exécuter (Envoyer la valeur de vitesse (vitesse) au nœud de traitement) ;

Exécuter (Réception de la valeur de vitesse (vitesse) de nœud mobile) ;

$\text{Vitesse} \leftarrow \text{getVitesse} ()$; //Récupérer la valeur la vitesse de nœud mobile

$\text{Vit_Temps} \leftarrow \text{getMatrice} (\text{Vit_Temps})$; //Cette fonction permet de mettre à jour la matrice vitesse_temp

*/*La matrice Vit_Temps est constituée des valeurs de vitesse dans la première ligne et des valeurs des temps correspondant aux vitesses dans la deuxième ligne */*

$\text{Trouve} \leftarrow \text{Faux}$;

```

i ← 1 ;
Tant Que (non Trouve ET i ≤ n)
    Si (Vit_Temps[1][i] = Vitesse) alors /*Si la vitesse de nœud mobile existe dans
    la matrice*/
        Timer ← vit_temps[2][i] ; //Récupérer l'intervalle de temps idéale
        Trouve ← Vraie ;
    Sinon
        i ← i+1 ; //Incrémenter la variable
    Fin Si ;
Fin Tant Que
Obtenir_intervalle_de_temps ← Timer ; //Retourne la valeur trouvée d'intervalle de
temps

```

Fin

3.3.2 Algorithme générale de la solution proposé :

Algorithme 2 : La solution proposée

Variables :

Tab_Signal [] : Tableau d'entier ;

N, Seuil, Etat, Ecart, Time, VPPS, Coeff_Erreur, Coeff_Correction : Entier ;

Début :

```

Lire(N) ; //Lire le nombre de valeurs de puissance du signal à récupérer
Lire (Coeff_Erreur) ; //Lire le nombre qui permet de détecter l'erreur de la prédiction
Lire (Coeff_Correction) ; //lire le nombre qui permet de corriger l'erreur de la
prédiction
Etat ← N ; //Sauvegarder le nombre des valeurs de puissance du signal à récupérer
afin que nous puissions revenir à l'état initial
Pour i allant de 1 à N faire
    Tab_Signal [i] ← getPuissanceSignal () ; //Récupérer une valeur de la puissance
    du signal.
    Time ← Obtenir_intervalle_de_temps () ; //Récupérer une valeur d'intervalle de
    temps
    Sleep (Time) ; //Lancer une minuterie pour attendre pendant une durée du temps
Fait ;
Tant Que (vraie)

```


Pour i allant de 1 à N-2 **faire**

MVPS ← $((\text{Tab}[i+1] - \text{Tab}[i]) / (N-2)) + \text{MVPS}$; //Étudier le changement de la puissance du signal et calculer la moyenne de changement du signal en même temps.

Fait ;

VPPS ← $\text{Tab_Signal}[N] + \text{MVPS}$; //Estimation de la valeur de la prochaine puissance du signal

Seuil ← $\text{getSeuil}()$; //Récupérer la valeur du seuil de la puissance minimale du signal de l'interface réseau

Si (VPPS > Seuil) **alors**

Exécute (désactiver le mécanisme de réduction de fenêtre TCP) ;

Sinon

Exécute (activer le mécanisme de réduction de fenêtre TCP) ;

Fin Si ;

Pour i allant de 1 à N-1 **faire**

Tab_Signal [i] ← $\text{Tab_Signal}[i+1]$; //Mettre a jour le tableau des puissances du signal

Fait ;

Time ← $\text{Obtenir_intervalle_de_temps}()$; //Récupérer une valeur de temps

Sleep (Time) ; //Lancer une minuterie pour attendre pendant une durée du temps

Tab_Signal [N] ← $\text{getSignal}()$; //Récupérer et sauvegarder une nouvelle valeur de la puissance de signal

Ecart ← $\text{Tab_Signal}[N] - \text{VPPS}$; //Étudier la distance entre la valeur réelle de puissance du signal et la valeur prédite de puissance du signal

 Ecart ← $\text{Abs}(\text{Ecart})$; //La fonction ABS renvoie la valeur absolue d'un nombre

Si (Ecart >= Coeff_Erreur) **alors**

Pour i allant de N à N+ Coeff_Correction **faire**

 Tab_Signal [i] ← $\text{getPuissanceSignal}()$; //Récupérer une valeur de la puissance du signal.

 Time ← $\text{Obtenir_intervalle_de_temps}()$; //Récupérer une valeur de temps

 Sleep (Time) ; //Lancer une minuterie pour attendre pendant une durée du temps

Fait ;

 N ← $N + \text{Coeff_Correction}$; // Corriger l'erreur d'estimation en incrémentant le nombre de valeurs de puissance du signal à récupérer

Sinon

```

Si (Etat <>N) alors
  Pour i allant de 1 à Coeff_Correction faire
    Pour i allant de j à N-1 faire
      Tab_Signal [j] ← Tab_Signal [j+1] ; //Mettre à jour le tableau des
      puissances du signal
    Fait ;
  Fait ;
  N ← N - Coeff_Correction ;//Réduire le nombre de valeurs de
  puissance du signal car l'estimation donne de bons résultats
Fin Si ;
Fin Si ;
Fin Tant Que
Fin

```

3.4 Organigrammes de la solution :

Les organigrammes de chaque algorithme sont présentés comme suite :

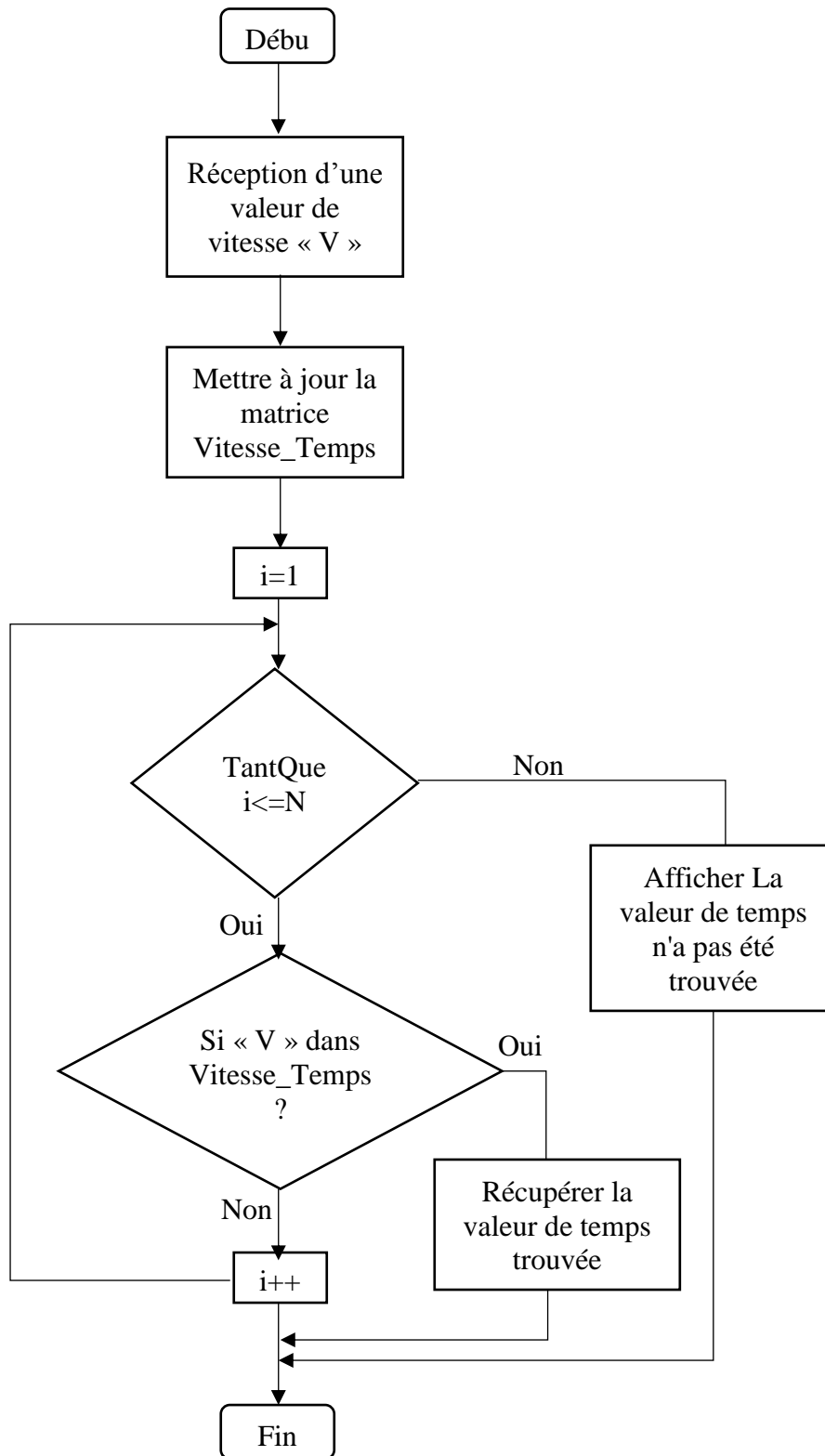


Figure 13: Organigramme du calcul de l'intervalle de temps

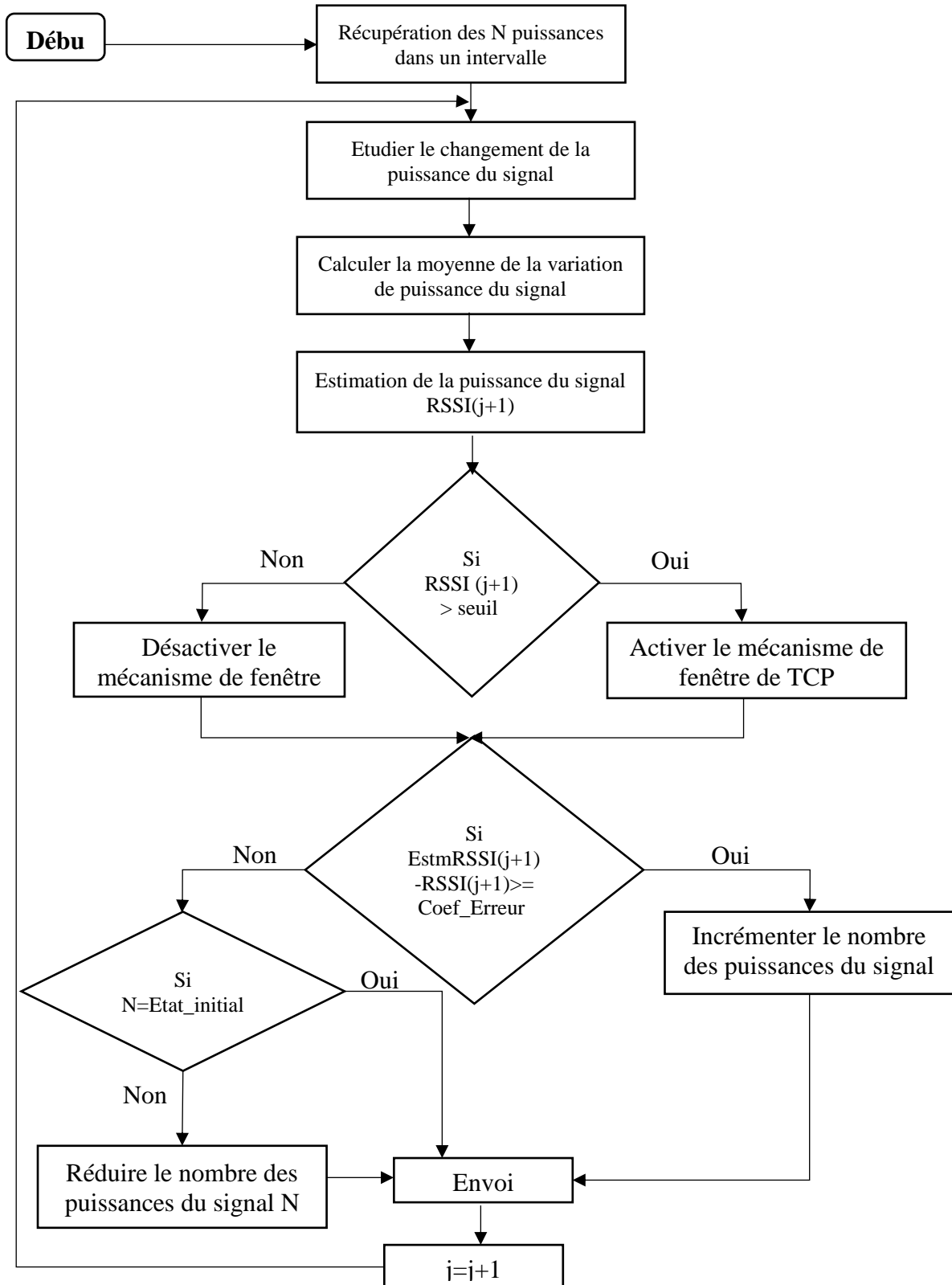


Figure 14: Organigramme de l'approche proposée en mono saut

3.5 Conclusion

Dans cette phase de notre mémoire de fin d'étude nous avons mis en évidence notre approche proposée dans le but d'améliorer les performances du protocole TCP dans les réseaux sans fil en mono-saut. Le mécanisme que nous avons présenté est basé sur l'estimation de la valeur du RSSI, il peut prédire la prochaine valeur de puissance du signal du prochain paquet à l'aide des valeurs de puissance du signal précédents avec des vitesses de nœud variables afin de différencier entre les pertes de paquets dues à la congestion et les pertes de paquets dues à la mobilité. De plus, nous avons proposé une méthode adaptative qui permet à notre solution de prédire les prochaines valeurs de puissance du signal avec une bonne précision. Le chapitre qui suit a pour objectif la mise en œuvre de notre approche.

Chapitre

4

Implémentation et Evaluation

4.1 Introduction

Cette phase a pour objet de mettre en œuvre notre approche proposée. En premier lieu, nous allons présenter l'environnement d'implémentation qui illustre les équipements matériels et logiciels pour la construction de notre système d'émulation, puis nous spécifions les outils, la configuration de la topologie et les paramètres utilisés, quant à la deuxième partie nous faisons une étude sur la variation de la puissance du signal et nous finissons par présenter les scénarios les plus généraux de notre solution suivi par les discussions des résultats de l'émulation.

4.2 Environnement d'implémentation

4.2.1 Équipement logiciel (Système d'exploitation)

L'environnement Linux (Ubuntu) a été choisi pour l'implémentation et la validation de notre approche. Ubuntu [40] est une des nombreuses distributions du système d'exploitation GNU/Linux, souvent abrégée en Linux. Elle est inspirée d'une autre distribution (l'une des plus célèbres, nommée Debian) et est sponsorisée par la société Canonical Ltd. Ubuntu a été créé avec les caractéristiques principales d'être :

- Convivial : Cette distribution est donc l'une des plus faciles à l'utilisation.
- Simple : un seul outil pour chaque tâche qui fait ce qu'on demande.
- Libre : Comme un logiciel libre est le plus souvent gratuit et le code ouvert, l'avantage est que chacun peut participer à son développement.
- Gratuit La gratuité fait partie intégrante de la philosophie d'Ubuntu : tout le monde doit pouvoir avoir accès à un système d'exploitation et à des applications performantes sans être obligé de payer [40].

De plus, Ubuntu est considéré comme une distribution stable et sécurisée pour le développement, c'est pourquoi nous avons décidé de l'utiliser dans notre travail.

4.2.2 Équipement matériel (GPS)

L'approche que nous avons proposée nécessite la vitesse pour pouvoir fonctionner pour cela on a décidé d'utiliser un récepteur GPS. Cet appareil capable de recevoir des informations des satellites GPS puis de calculer sa position géographique en plus il peut offrir de nombreuses fonctionnalités telles que le temps et la vitesse en temps réel mais en raison de la rareté de l'appareil sur le marché et du prix élevé, on était obligé de fabriquer notre propre tracker GPS à moindre coût pour obtenir la vitesse de nœud mobile. Pour concevoir ce petit GPS, on a utilisé les éléments suivants :

- Une carte Arduino Mini Pro à base ATMEGA328P.

- Module GPS NEO-6M.

4.2.3 Description de la topologie d'émulation

La topologie que nous avons mise en place est composée de deux nœuds :

- Le premier nœud est la station de base (Nœud Traitant), il est fixé à son propre emplacement et en même temps il est le responsable de la prise de décision et l'estimation de la valeur du RSSI.
- Le second nœud est mobile, sa tâche est d'alimenter la station de base en transmettant sa vitesse de manière continue c'est-à-dire informer le nœud traitant périodiquement. Il est connecté à un récepteur GPS afin de récupérer sa propre vitesse variable.

La figure ci-dessus représente notre topologie de réseau sans fil en mono-saut avec les extensions requises.

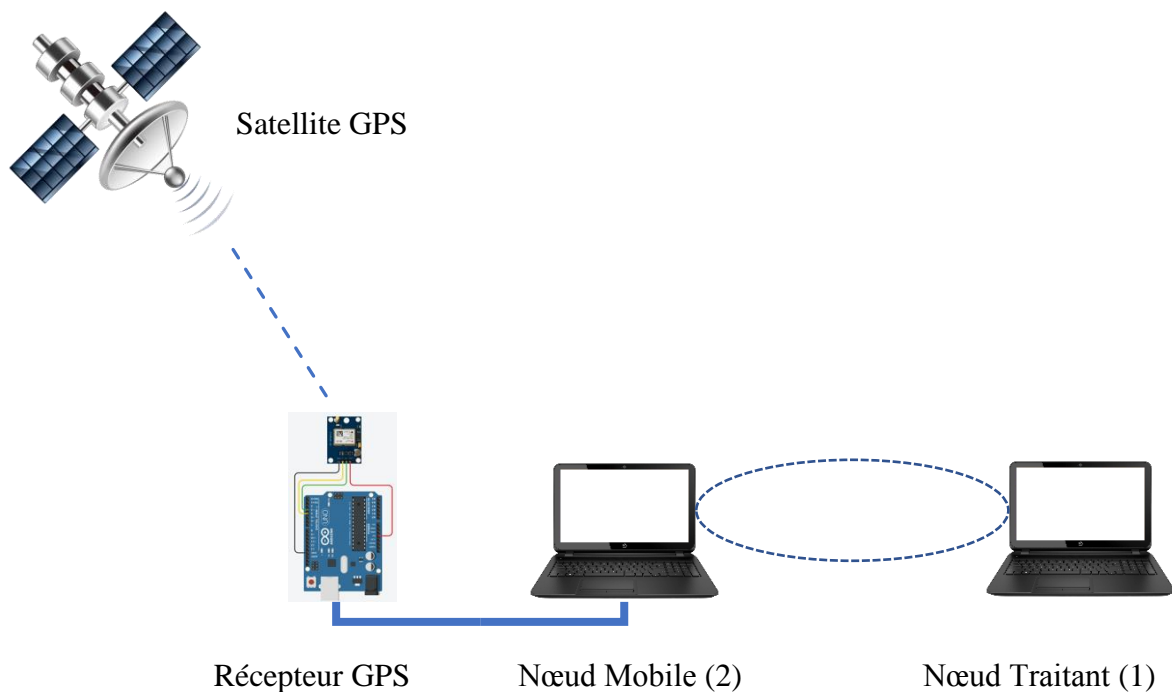


Figure 15: *Topologie de l'émulation*

Nous terminons cette section en illustrant les principes liés à notre topologie de travail. Ils sont décrits comme suit :

- Les nœuds ne sont pas sous l'effet des interférences.
- Le transfert des paquets se fait en mono-saut.
- La donnée utilisée est un fichier « WinRAR ».
- La valeur maximale de débit du transfère varie entre 50 Mbps et 55 Mbps.

N.B : Pendant la durée de l'émulation, nous avons pris en considération certains scénarios afin de comparer et évaluer notre solution proposée en mono-saut par rapport à l'approche [1] et la version de TCP d'Ubuntu (TCP Reno), les tests qui ont été effectués entre les trois variantes visaient à situer la différence de performance de chacun d'entre eux.

4.3 Étude de l'évolution de la puissance du signal dans le temps

L'indicateur d'intensité du signal reçu (RSSI) est une mesure estimée de la qualité de l'audition, de la détection et de la réception des signaux. L'avantage du RSSI est qu'il vous aide à déterminer et à savoir si un signal est suffisant pour établir une connexion sans fil et elle joue un rôle très important dans la bonne transmission des paquets dans les réseaux sans fil. RSSI ou cette valeur de signal est mesurée en décibels de 0 dbm (zéro) à -120 dbm (moins 120). Plus la valeur est proche de 0 dbm (zéro), plus le signal sera fort de même plus la valeur est proche de -120 dbm, plus le signal sera faible et insatisfaisant. Pour rendre la vision facile et simple, nous ajoutons 100 à chaque valeur récupérée de la puissance du signal et ainsi nous obtenons des valeurs positives, dans ce cas, plus la valeur de puissance du signal est élevée, plus la puissance est forte, le contraire est vrai.

Dans cette section nous étudions la variation de la puissance du signal dans des différentes positions fixes au cours du temps suivant les étapes suivantes :

- Etape-1 : on place les deux nœuds dans une distance appropriée environ dix mètres puis on lance un transfert d'un fichier compressé (.rar) de taille 9.58 go du nœud 1 vers le nœud 2.
- Etape-2 : on récupère la valeur de la puissance du signal reçu chaque dix secondes jusqu'à obtenir sept valeurs de la puissance du signal reçu dans la même position.
- Etape-3 : Ensuite, nous augmentons la distance entre les deux nœuds de deux mètres et nous répétons les mêmes étapes précédentes, jusqu'à atteindre un certain seuil de puissance du signal où la communication commence à interrompre.

4.3.1 Les résultats obtenus de l'expérience

Les graphiques suivants montrent l'évolution du débit en fonction du temps :

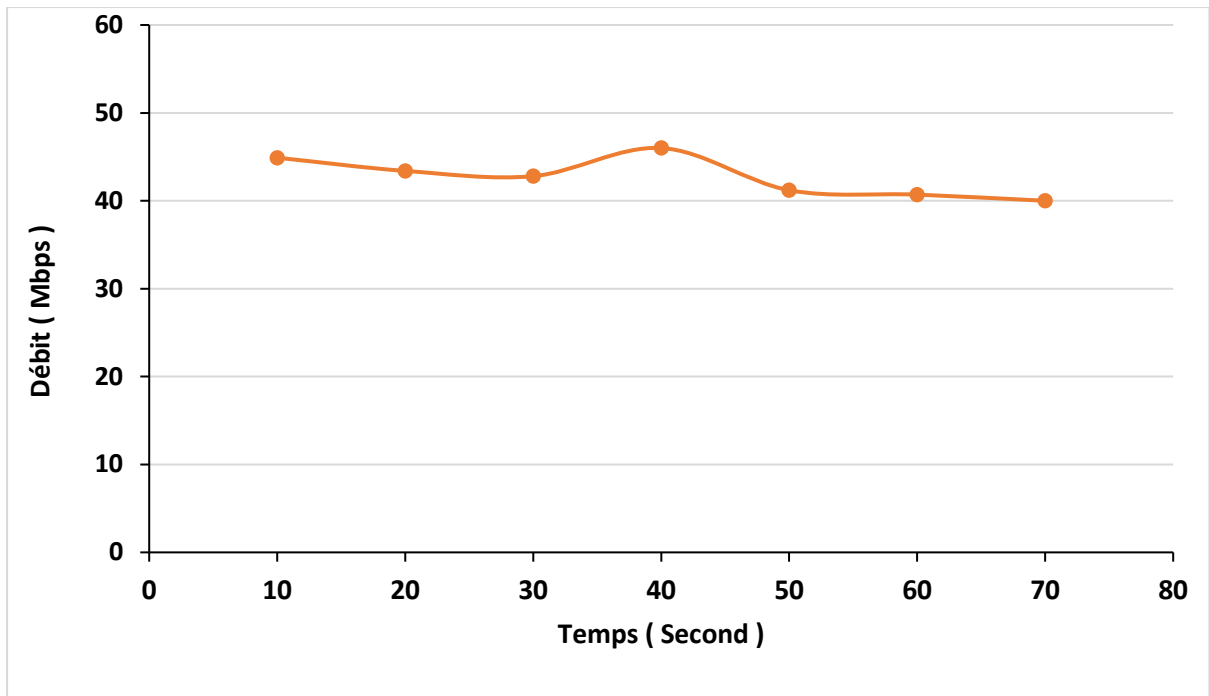


Figure 16 : *Les résultats de RSSI dans un rayon de 12m*

Le graphe de **Figure 16** représente les résultats obtenus dans la première expérience où le nœud mobile est loin de nœud fixe à une distance de douze mètres.

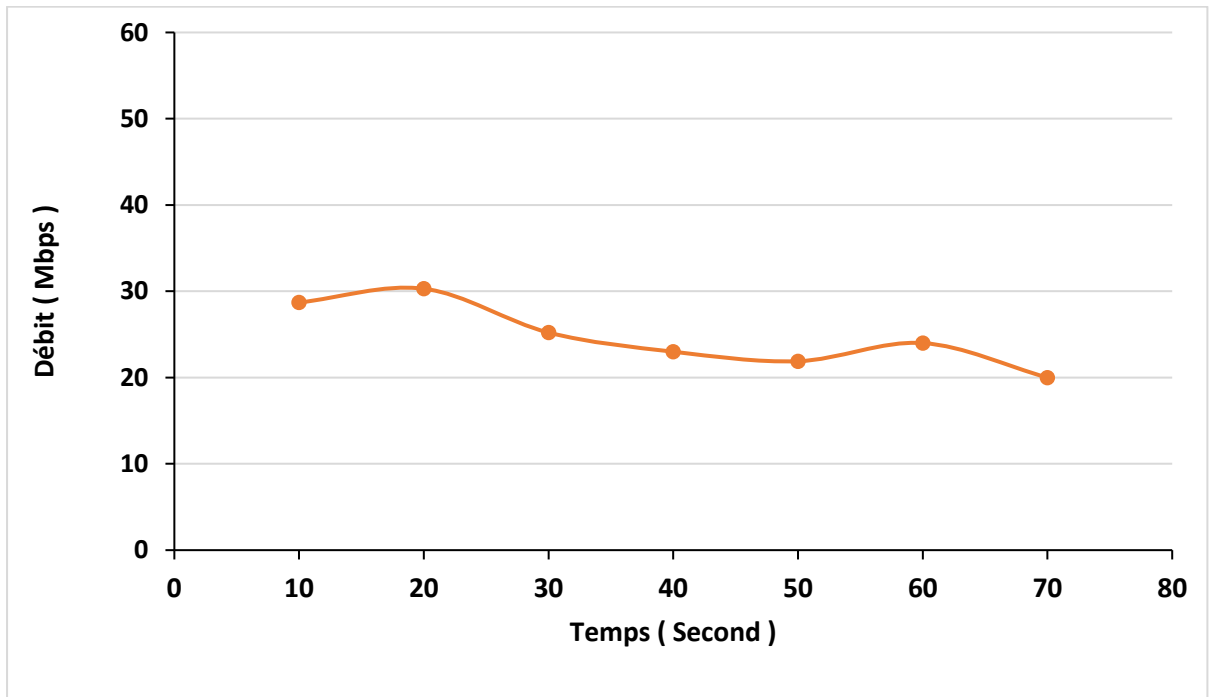


Figure 17 : *Les résultats de RSSI dans un rayon de 14m*

Le graphe de **Figure 17** représente les résultats obtenus dans la troisième expérience où le nœud mobile est loin de nœud fixe à une distance de quatorze mètres.

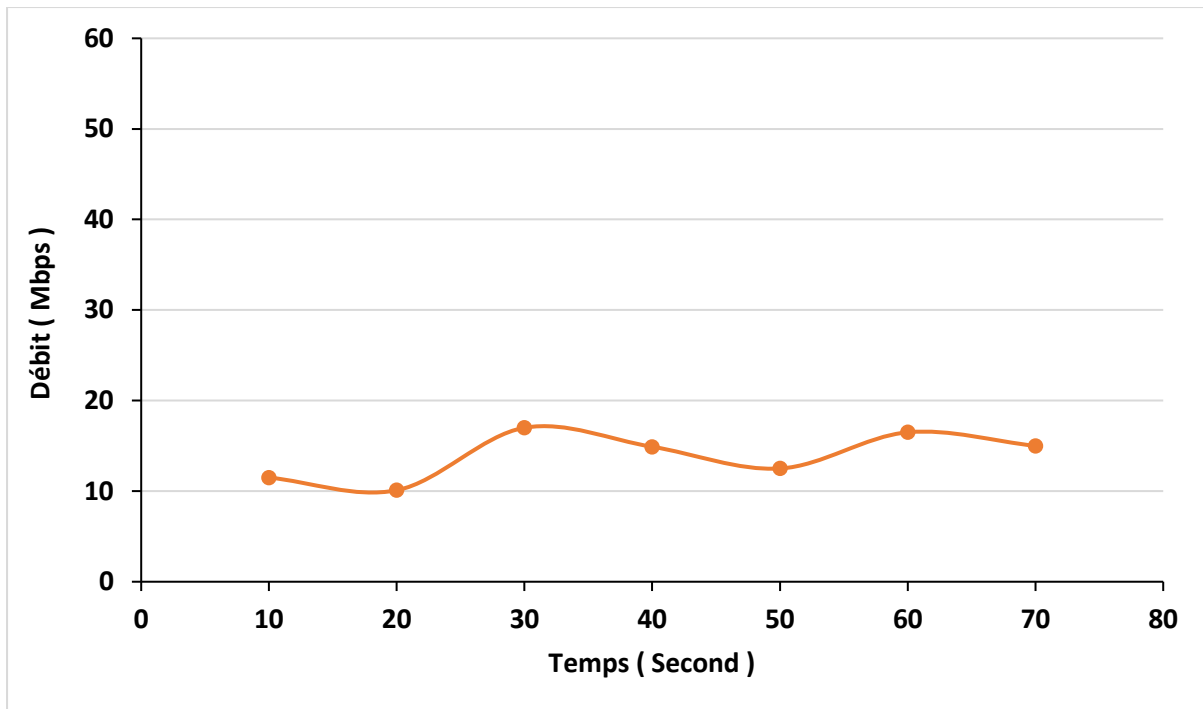


Figure 18 : Les résultats de RSSI dans un rayon de 16m

Le graphe de **Figure 18** représente les résultats obtenus dans la deuxième expérience où le nœud mobile est loin de nœud fixe à une distance de seize mètres.

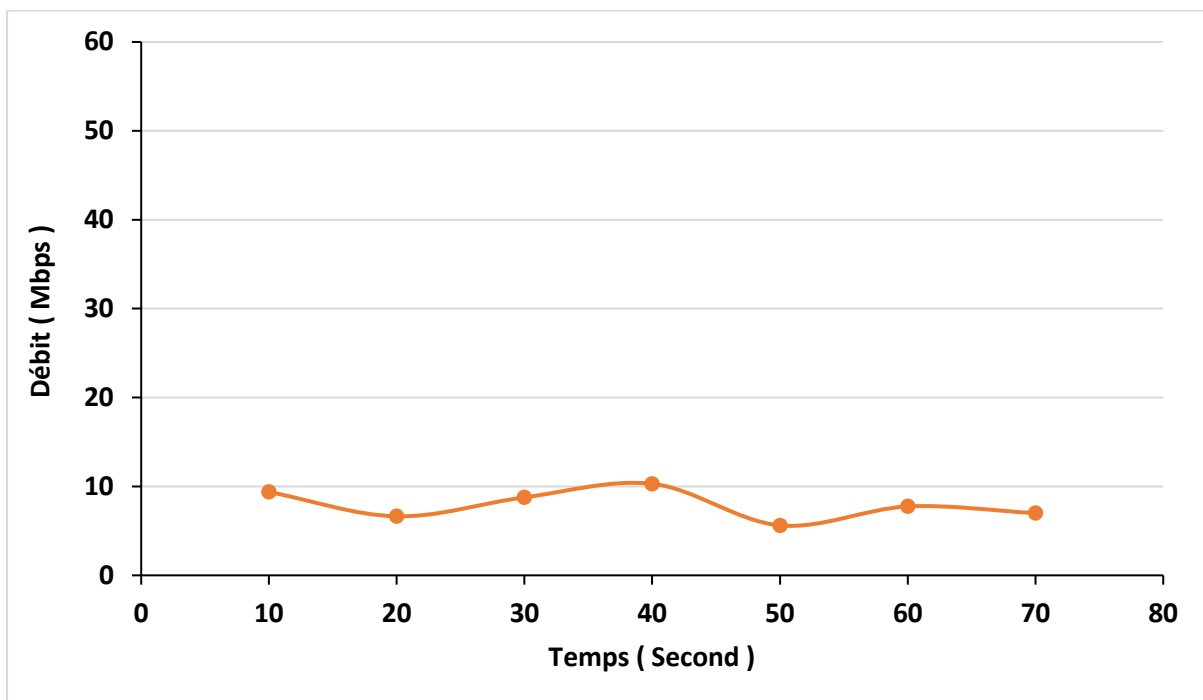


Figure 19 : Les résultats de RSSI dans un rayon de 18m

Le graphe de **Figure 19** représente les résultats obtenus dans la quatrième expérience où le nœud mobile est loin de nœud fixe à une distance de dix-huit mètres.

4.3.2 Interprétation des graphes

Après avoir analysé les graphes ci-dessus, on peut remarquer que lorsque on diffuse sur une plus grande distance, le signal s'affaiblit et la bande passante de la connexion de données sans fil devient plus lente. Il en résulte une production globale de données de puissance du signal plus faible parce que le nœud s'éloigne de la zone de couverture sans fil, Dans le cas contraire la valeur de la puissance du signal reçu ne cesse de s'améliorer à mesure que nous diminuons la distance entre les deux nœuds. Les valeurs de puissance du signal reçu au même endroit sont proches les unes des autres et varient autour d'une valeur de puissance du signal spécifique qui se répète la plupart du temps.

À la fin, nous concluons de ces expériences que la valeur de puissance du signal change en fonction de plusieurs facteurs. Dans un environnement libre d'obstacle, la distance entre les deux nœuds est un facteur important pour exprimer la qualité de la puissance du signal, l'intensité du signal fort indique la convergence des deux nœuds, tandis que la faible intensité du signal indique que la transmission des paquets peut être interrompue en raison de la longue distance qui sépare les deux nœuds.

4.3.3 La portée minimale de la puissance du signal

D'une manière régulière, chaque carte réseau a un seuil minimal d'intensité du signal donné par le fabricant, mais malheureusement, les interfaces réseau que nous avons ne fournissent pas ces informations car elles ne sont pas professionnelles et récentes. Les tests effectués par les auteurs de la solution [1] ont montré que le seuil minimal d'intensité du signal est d'environ (-30 dbm), sa valeur réelle étant d'environ (-70 dbm) dans l'intervalle de signal spécifié sur une distance de 14 mètres. Les tests que nous avons effectués précédemment sur la variation de la puissance du signal et comme indiqué dans le graphe du **Figure 17** confirment cette information tel que le transfert de paquets commence à se dégrader à partir d'une distance de 14 mètres du seuil de communication qui est équivalent à une puissance du signal qui est égale à (-70dbm) à cause des pertes provoquées par la distance entre les deux nœuds.

4.4 Evaluation des performances

Pour évaluer l'efficacité de l'approche proposée, nous étudions la variation du débit au cours du distance. Le processus a été réalisé selon les étapes suivantes :

- Etape-1 : Lancer un transfert d'un fichier compressé (.rar) de taille 9.58 go du nœud traitant vers le nœud mobile.

- Etape-2 : Nous faisons en sorte que le nœud mobile s'éloigne du nœud de traitement en fonction d'une certaine vitesse.
- Etape-3 : On récupère le débit tous les deux mètres jusqu'à la fin de la piste (La longueur de la piste est de 22 mètres).
- Etape-4 : Faire les étapes précédentes avec les trois variantes de TCP (notre approche proposée, TCP-Reno et solution [1]).
- Etape-5 : Répéter les mêmes étapes avec des vitesses différentes.

4.4.1 Les paramètres de l'émulation

Le nombre initial des valeurs de puissance du signal utilisées pour l'estimation de la prochaine valeur est égale à 5.

On utilise 3 comme un coefficient d'erreur pour vérifier la distance entre la valeur estimée et la valeur réel si les deux nombres sont proches l'un de l'autre on garde l'estimation avec 5 valeurs.

Dans le cas où les deux sont loin l'un de l'autre l'approche ajoute automatiquement 2 à 5 pour corriger cette erreur d'estimation, comme ça le nombre de valeurs devient 7 pour la prochaine estimation, le même processus sera répétée si la prédiction donne de bons résultats , l'approche réduit le nombre des valeurs de puissance du signal à 5 sinon elle ajoute des valeurs jusqu'à arriver à 9 valeurs pour la correction et s'arrête à ce nombre de valeurs de puissance du signal et n'augmente plus mais si la prédiction est bonne, les valeurs diminuent à 7 et ainsi de suite.

Pour envoyer la vitesse du nœud 2 au nœud 1, nous utilisons la communication bout à bout par socket en python, le « server.py » en mode passif à l'écoute des messages du « client.py » sur le port 12345

Ce tableau ci-dessous est utilisé comme un paramètre dans l'approche proposée pour trouver la valeur d'intervalle de temps appropriée.

Vitesse(km/h)	[03 – 05]	[07 – 09]	[10 – 11]	[20 - 27]	[30 -35]
Slot time(s)	5	3	2.2	0.74	0.31

Tableau 5 : *Les vitesses avec l'intervalle de temps correspondant*

4.4.2 Un aperçu des scénarios

Dans cette partie nous présentons les différents scénarios réalisés pour tester les performances, ils sont décrits comme suit :

- Scénario 1 : Nous faisons en sorte que le nœud 2 s'éloigne du nœud de base avec une vitesse de marche très lente d'un être humain jusqu'à la fin du chemin.
- Scénario 2 : en suite le nœud 2 s'éloigne du nœud de base avec une vitesse de marche normale d'un être humain jusqu'à atteindre la fin du trajet.
- Scénario 3 : Cette fois, le nœud 2 s'éloigne du nœud de base avec une vitesse de marche très rapide d'un être humain jusqu'à la fin du chemin.
- Scénario 4 : Le nœud s'éloigne de la station de base avec une vitesse moyenne de sprint d'un être humain.
- Scénario 5 : En utilisant une bicyclette, nous avons porté le nœud 2 et nous l'avons éloigné du nœud de traitement avec une vitesse de conduite maximale.

4.4.3 Les résultats des tests

Les résultats des différents scénarios sont indiqués ci-dessous :

1. Les résultats du premier scénario

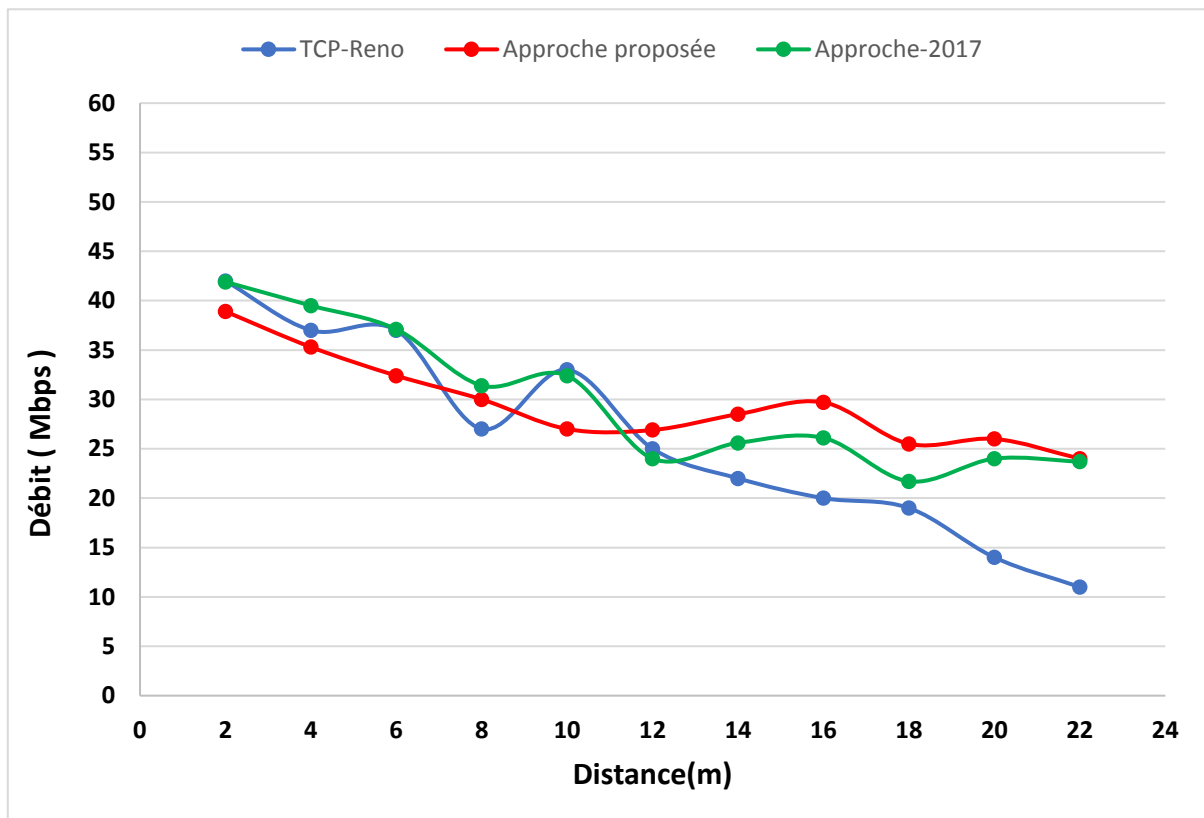


Figure 20 : Les résultats du scénario numéro un

Le graphe de **Figure 20** représente les résultats obtenus des trois variants après avoir effectué le premier scénario, le nœud mobile se déplaçait selon une vitesse variable comprise

entre 3 km/h et 5 km/h, et la valeur d'intervalle de temps utilisé pour impacter la puissance de signal était égal à 5 seconds.

2. Les résultats du deuxième scénario

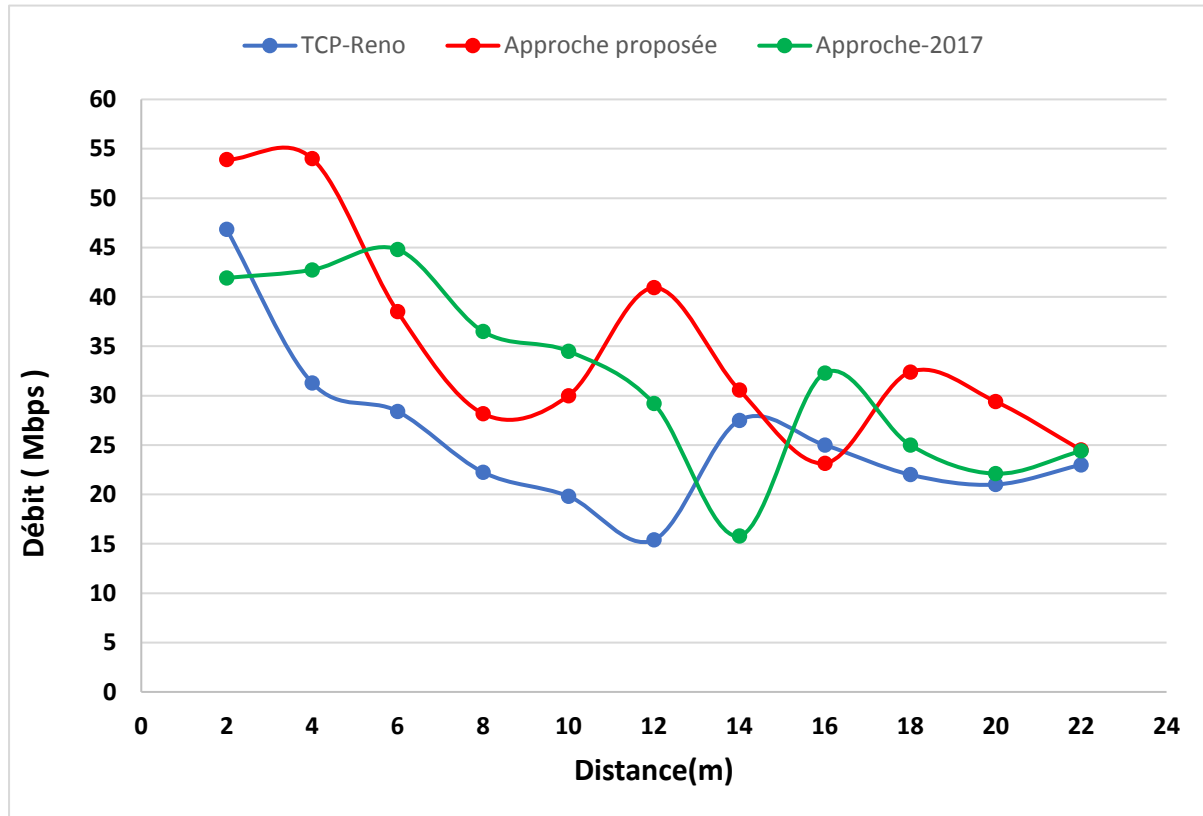


Figure 21 : Les résultats du scénario numéro deux

Le graphe de **Figure 21** représente les résultats obtenus des trois variants après avoir effectué le deuxième scénario, le nœud mobile se déplaçait selon une vitesse variable comprise entre 7 km/h et 9 km/h et la valeur d'intervalle de temps utilisé pour impacter la puissance de signal était égal à 3 seconds.

3. Les résultats du troisième scénario

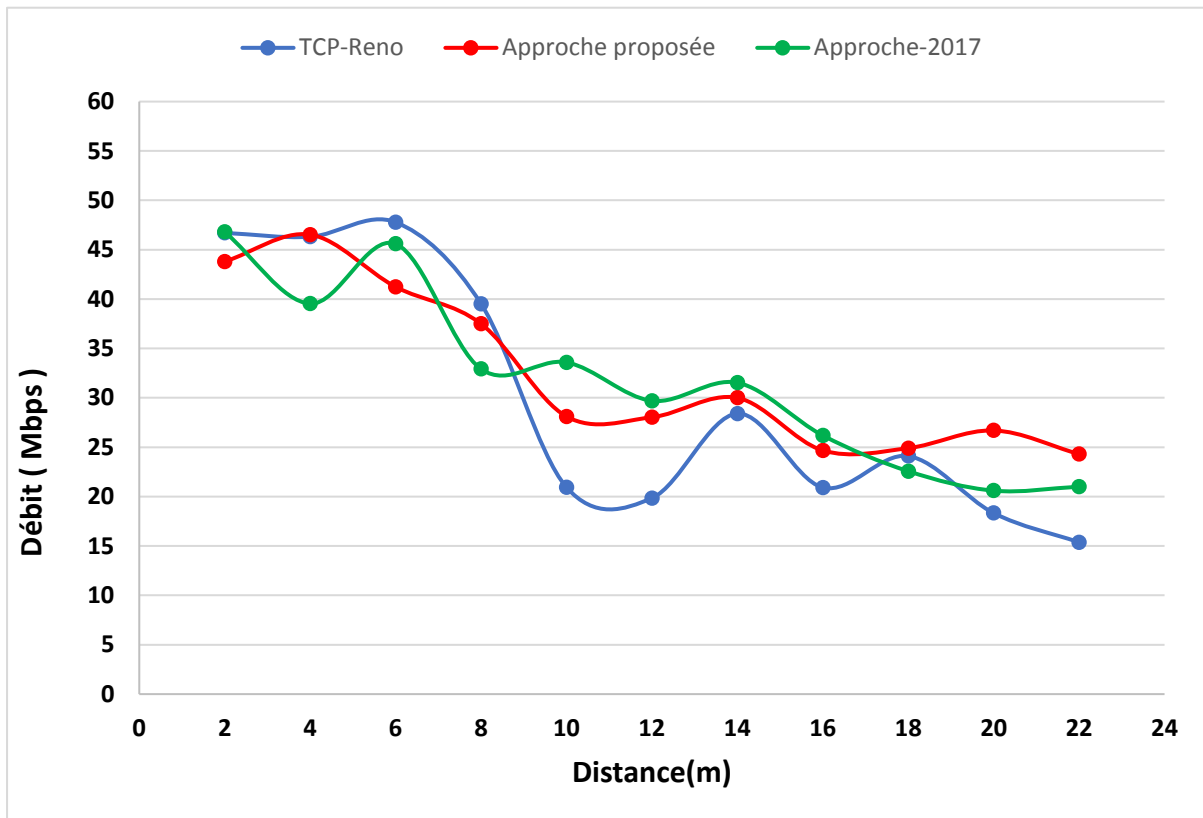


Figure 22 : Les résultats du scénario numéro trois

Le graphe de **Figure 22** représente les résultats obtenus des trois variants après avoir effectué le troisième scénario, le nœud mobile se déplaçait selon une vitesse variable comprise entre 10 km/h et 11 km/h et la valeur d'intervalle de temps utilisé pour impacter la puissance de signal était égal à 2.2 seconds.

4. Les résultats du quatrième scénario

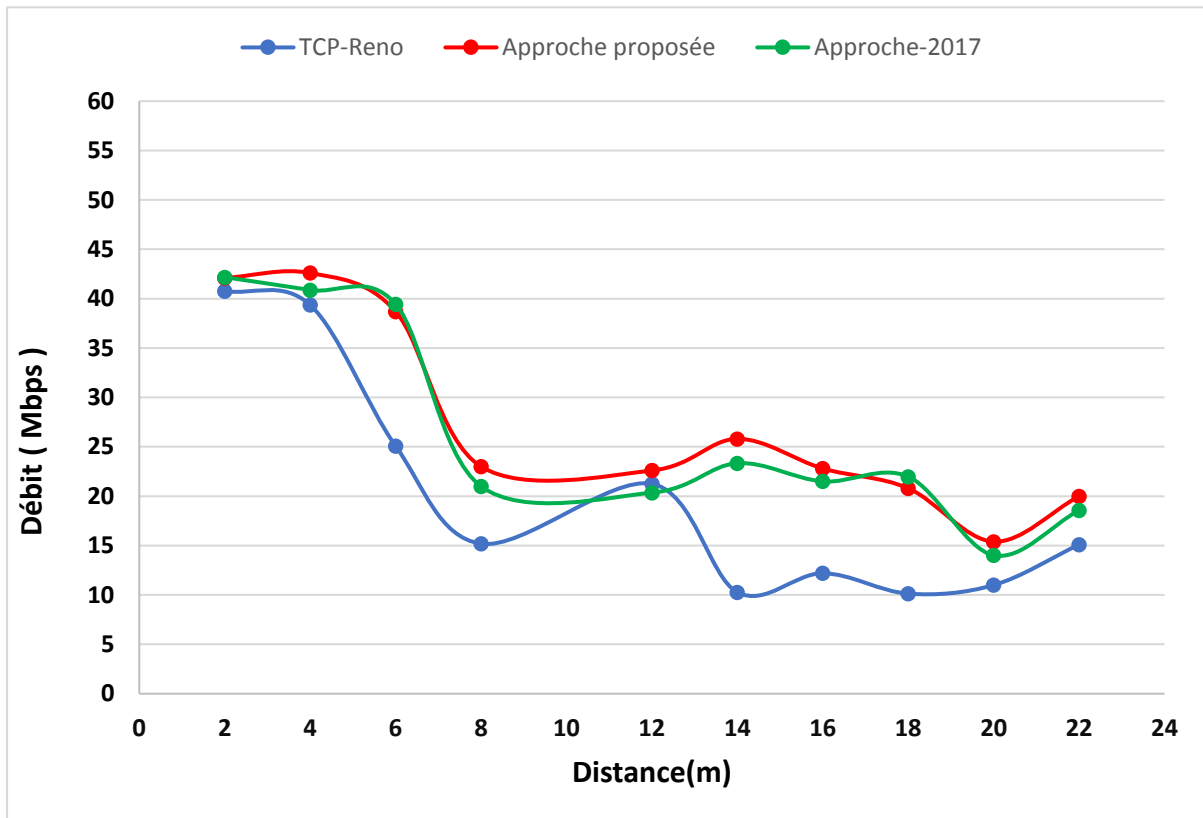


Figure 23 : *Les résultats du scénario numéro quatre*

Le graphe de **Figure 23** représente les résultats obtenus des trois variants après avoir effectué le quatrième scénario, le nœud mobile se déplaçait selon une vitesse variable comprise entre 20 km/h et 27 km/h et la valeur d'intervalle de temps utilisé pour impacter la puissance de signal était égal à 0.74 seconds.

5. Les résultats du cinquième scénario

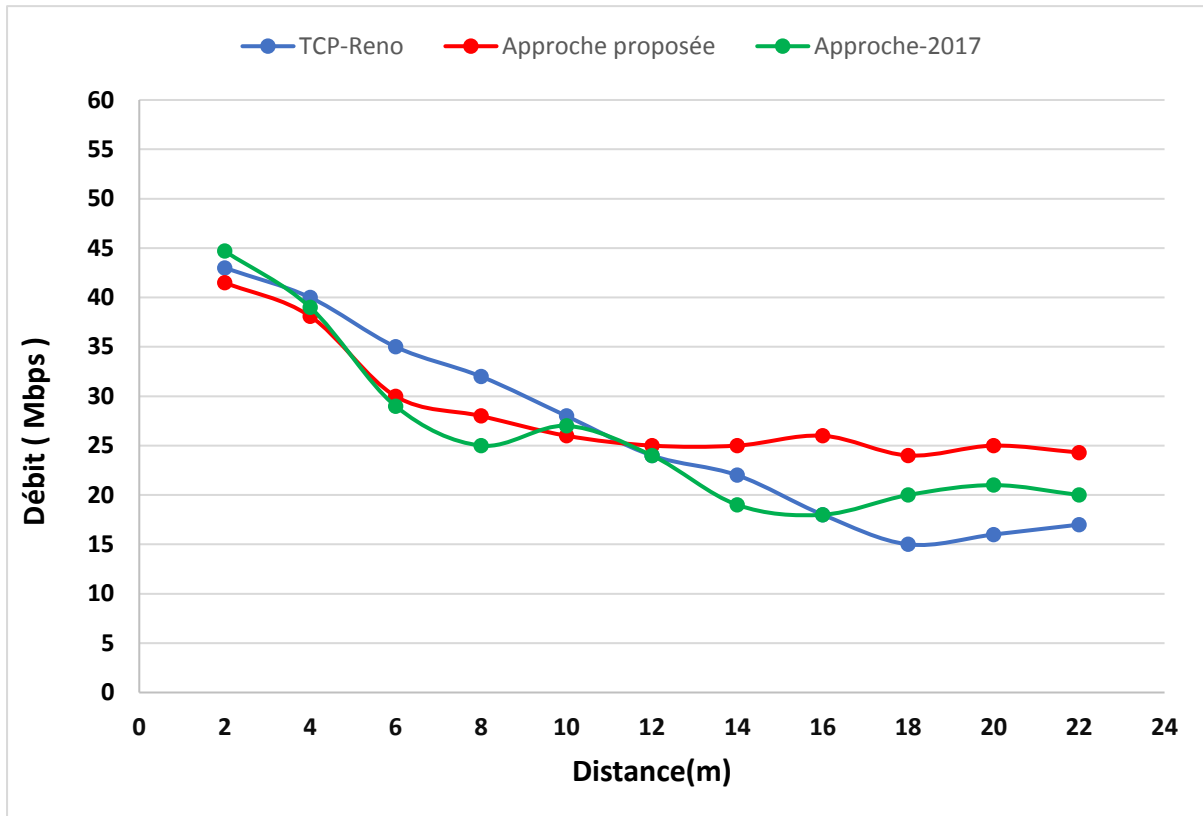


Figure 24 : *Les résultats du scénario numéro cinq*

Le graphe de **Figure 24** représente les résultats obtenus des trois variants après avoir effectué le cinquième scénario, le nœud mobile se déplaçait selon une vitesse variable comprise entre 30 km/h et 35 km/h et la valeur d'intervalle de temps utilisé pour impacter la puissance de signal était égal à 0.31 seconds.

4.4.4 Discussion des résultats

Dans cette section, nous démontrons les résultats des performances de débit à travers les différents scénarios. Après avoir lu attentivement les graphes (**Figure 20, Figure 21, Figure 22, Figure 23, Figure 24**) ci-dessus, on constate que lorsque la distance entre le nœud mobile et la station de base augmente d'environ 2 mètres à 12 mètres, le débit de toutes les variantes de TCP diminue approximativement environ 20 Mbps (La plupart du temps, le débit décroît de 45 Mbps à 22 Mbps) de plus les deux approches proposées (solution [1] et notre approche) gardent leurs fenêtres de flux de TCP (*TCP window scaling* = 1) actifs. Bien que lorsque la distance est proche où dépasse le seuil minimal environ 13 mètres à 14 mètres, la perte de paquets devient agressive et par conséquent notre approche proposée et la solution [1] commencent à s'améliorer son débit d'environ 41%. Cela s'explique par le fait que les deux TCP (solution [1] et notre approche) peuvent détecter et différencier les pertes de paquets aléatoires en raison de la mobilité via le modèle d'estimation de la valeur du RSSI en premier lieu puis réagir à cet événement de non-congestion en désactivant la fenêtre de flux de TCP (*TCP window scaling* = 0). De l'autre côté, le TCP Reno considère que cette perte est due à un événement de congestion et par conséquent il renvoie les paquets perdus tout en réduisant le débit de transmission tel que le débit diminue jusqu'à ce qu'il soit limité approximativement entre 10 Mbps à 15 Mbps pendant les 8 mètres.

La **Figure 23** et **Figure 24** montrent que notre approche proposée réagit rapidement aux pertes de paquets en raison de la mobilité mieux que la solution [1] où elle active le mécanisme de flux de TCP et elle améliore le débit en moyenne entre 20 Mbps à 30 Mbps avant d'atteindre le seuil minimal de la puissance du signal à une distance environ 12.5 mètres à 13.5 mètres. La fréquence d'estimation de la solution [1] est fixe avec une estimation toutes les trois secondes, ce qui est très tardif lorsqu'il s'agit d'estimer avec une vitesse supérieure à la vitesse d'un être humaine, notre solution peut modifier dynamiquement son propre taux d'estimation en fonction de la vitesse variable du nœud, bien qu'elle puisse ajuster parfaitement le temps d'intervalle pour une bonne récupération de la puissance du signal. La méthode adaptative a donné des prévisions précises qui permettent d'ajuster la fenêtre de flux de TCP régulièrement (activer ou désactiver).

4.5 Conclusion

Tout au long de ce chapitre, nous avons abordé notre environnement d'émulation. Par la suite, nous avons testé les performances de notre approche proposée par rapport aux autres

variantes de TCP. Les résultats recueillis au cours de l'émulation ont montré l'efficacité de notre solution proposée.

Conclusion Générale et Perspectives

L'objectif de ce mémoire était de proposer une adaptation du protocole de transport TCP opérant sous les réseaux sans fil mono saut, et ce en minimisant les erreurs de distinction entre les erreurs liées aux mobilités et les erreurs liées à la congestion afin d'améliorer les performances du protocole TCP dans les réseaux sans fil mono saut mobiles, La solution proposée consistait à estimer la prochaine valeur de la puissance du signal et l'utiliser afin d'améliorer les performances.

Afin de tester notre solution une émulation de la solution a été réalisée en considérant différents scénarios avec variation des principaux paramètres (pouvant influencer la communication) et ce afin d'avoir des résultats optimaux.

Les résultats ont été récupérées, traitées puis représentées dans des graphes afin de pouvoir mieux comprendre les comportements du protocole avant et après sa modification, ces graphes on était interprétés avec le plus grand soin en s'intéressant aux moindres changements. En se basant sur ces interprétations, une comparaison entre le protocole original et celui modifié a été faite, concluant que ce dernier a fait ses preuves et à montrer clairement son efficacité face au protocole original. L'adaptation a bien eu lieu, objectif atteint.

Comme perspectives de travail, on prévoit d'améliorer notre approche afin de l'adapter pour une utilisation dans un environnement sans fil multi saut. De cette façon, la solution deviendra adaptative avec les différents topologies du réseau sans fil infrastructurelle.

Références

- [1] R. Mansour, " Estimation de la valeur de la puissance minimale dans reseau sans fil à mono-saut" Mémoire de master en Informatique, Soutenu à l'université de Saad Dahleb Blida, le 5 octobre 2017.
- [2] P. Christo, "techjury," 12 March 2019. [Online]. Available: <https://techjury.net/stats-about/mobile-vs-desktop-usage/>.
- [3] M. T. Jan, "Comparison of the IEEE 802.11, 802.15.1,802.15.4 and 802.15.6 wireless standards", September 18, 2007.
- [4] K. Jatau Isaac, D. Yahaya Shagaiya and S. S. Bako, "TCP ISSUES IN MOBILE ADHOC NETWORKS: Challenges and Solutions," *International Journal of Modern Engineering Research (IJMER)*, vol. 3, no. 1, pp. 439-446, 2013.
- [5] X. George and S. Mika, "TCP performance issues over wireless links," *IEEE Communications Magazine*, vol. 39, no. 4, pp. 52-58, April 2001.
- [6] J. Geier, "Macmillan Network Architecture and Development Series," in *Wireless Lans: Implementing Interoperable Networks*, San Francisco, New Riders Pub, 1999, p. 418.
- [7] A. A. Khaldoun, P. Guy and V. Guillaume, Réseaux de mobiles et réseaux sans fil, Paris: Eyrolles, 2001.
- [8] A. Roberto, "Wireless Howto," Linux Documentation Project, July 31, 2002.
- [9] M. Nicolas, " La Mobilité dans les réseaux IP," Université Louis Pasteur de Strasbourg, Strasbourg, 2001.
- [10] IEEE SA Standards Board,802.11. IEEE, 1999.
- [11] M. S. Mezghanni, "Étude de la fiabilité des communications dans un réseau de capteurs sans-fils appliqué aux mines souterraines," Université du Québec en Abitibi-Témiscamingue, Québec, 2016.
- [12] W. Rong-Hou, L. Yang-Han, T. Hsien-Wei, J. Yih-Guang and C. Ming-Hsueh, "Study of characteristics of RSSI signal," in *2008 IEEE International Conference on Industrial Technology*, Chengdu, China, 26 August 2008 .
- [13] G. P. John, Digital Communications, Mcgraw-Hill College, March 1, 1994.
- [14] P. Jon, TRANSMISSION CONTROL PROTOCOL, University of Southern California, 1981.

- [15] M. Matthew, r. Je and M. Jamshid, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *ACM SIGCOMM computer communication*, vol. 27, no. 3, p. 67–82, 1997.
- [16] W. Zheng and C. Jon, "Eliminating Periodic Packet Losses in 4.3 Tahoe BSD TCP Congestion Control Algorithm," *ACM Computer Communication Review*, vol. 22, no. 2, pp. 9-16, avril 1992.
- [17] J. Raj, "A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks," *ACM SIGCOMM Computer Communication Review*, vol. 19, no. 5, pp. 56-71, 5 October 1989.
- [18] J. Van, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 314-329, August 1988.
- [19] N. -. O. propria, "Evoluzione di CongWin in TCP Tahoe e TCP Reno," Wikimedia Commons, 2017 June 07 15:00.
- [20] V. JACOBSON, "Berkeley TCP evolution from 4.3-tahoe to 4.3-reno," *Proceedings of 18th IETF*, vol. 365, Septembre 1990.
- [21] H. STEVEN, L. P. LARRY and W. LIMIN, "Understanding tcp vegas: A duality model," *ACM SIGMETRICS Performance Evaluation Review*, vol. 29, no. 1, pp. 226-235, June 2001.
- [22] K. Dzmitry and G. Fabrizio, "Cross-layer congestion control in ad hoc wireless networks," *Ad Hoc Networks*, vol. 4, no. 6, p. 687–708, 2006.
- [23] D. Yassine, "Amélioration de la performance de TCP dans les réseaux mobiles ad hoc", Thèse de doctorat en Informatique, Université d'Oran Ahmed Ben Bella, Oran, Novembre 2016.
- [24] C. Rung-Shiang and L. Hui-Tang, "A cross-layer design for TCP end-to-end performance improvement in multi-hop wireless networks," *Computer Communications*, no. 31, p. 3145–3152, 2008.
- [25] M. Noor, K. Shafiullah and L. KoK-Keong, "Limitation and Challenges of TCP in Mobile ad hoc Networks," in *2009 5th IEEE GCC Conference & Exhibition*, Kuwait City, Kuwait, 17-19 March 2009.
- [26] P. V. Er, "Hidden Terminal and Exposed Terminal problem and its solution," yuvayana, Feb 23, 2016.

- [27] W. Emilia, H. Masaki and W. K. Moo, "Optimizing RTS/CTS to Improve Throughput in Ad Hoc WLANs," in *Conference Paper* , September 2017.
- [28] H. D. Peter, Global Positioning System Overview, Department of Geography, University of Texas at Austin, 1999.
- [29] G. S. Frederic, "GPS: Theory, Practice and Applications," PDHonline Course, 2012.
- [30] T. A. Corporation, "GPS Primer: A student guide to the Global Positioning System," The Aerospace Corporation, National Science Digital Library (NDSL), 2003.
- [31] "Arduino UNO Product Manual".
- [32] L. Xin, F. Zhen, Y. Z. Yao and L. Si Qi, "GPS Positioning System Design Based on Micro Control Unit," *Advanced Materials Research*, pp. 1171-1174, April 2014.
- [33] [Online]. Available: <https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/>.
- [34] M. Noor and J. ., Thomas, "A survey of performance enhancement of transmission control protocol (TCP) in wireless ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, no. 96, p. 1–23, 2011.
- [35] L. Yao-Nan and H. Ho-Cheng, "A New TCP Congestion Control Mechanism over Wireless Ad Hoc Networks by Router-Assisted Approach," in *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, Toronto, Ont., Canada , 22-29 June 2007 .
- [36] S. Prasanthi and L. Meejeong, "TCP NRT: a new TCP algorithm for differentiating non-congestion retransmission timeouts over multihop wireless networks," *EURASIP Journal on Wireless Communications and Networking*, no. 172, pp. 1-20, 2013.
- [37] J. Jiang and R. Jain, "Explicit Congestion Notification," in *IEEE 802.1 Congestion Group Meeting*, Dallas, TX, Nov 14, 2006.
- [38] S.-G. Alaa, G.-D. Yacine and S. Sidi-Mohammed, "TCP WELCOME TCP variant for Wireless Environment, Link losses, and COngestion packet loss ModEls," in *2009 First International Communication Systems and Networks and Workshops*, Bangalore, India , 31 March 2009.
- [39] A. Dallal and M. Diao, "Estimation de la valeur de la puissance du signal minimale sur les réseaux sans fil multi-saut," *Mémoire de master en Informatique*, Soutenu à l'université de Saad Dahleb, Septembre 2018.

- [40] c. d. Vikidia, "Ubuntu," 7 Avril 2019. [Online]. Available: <https://fr.wikidia.org/w/index.php?title=Ubuntu&oldid=1243126>.
- [41] R. BEDOUHENE and M. BENMEDOUR, Protocole de Connexion des Réseaux Ad Hoc à Internet, Mémoire de fin d'études, université des sciences et de la technologie Houari Boumediene, 2014.
- [42] A. Philippe, Réseaux informatiques - Notions fondamentales (Normes, Architecture, Modèle OSI, TCP/IP, Ethernet, Wi-Fi, ...), Paris: Editions ENI, 2009.
- [43] L. Tayeb, Le Routage dans les Réseaux Mobiles Ad Hoc, Université des Sciences et de la Technologie Houari Boumediene, Institut d'Informatique, 2000.
- [44] N. BOUKHECHEM, ROUTAGE DANS LES RESEAUX MOBILES AD HOC PAR UNE APPROCHE A BASE D'AGENTS, Université de constantine , 2008.
- [45] S. BOUKLI HACENE and A. LEHIRECHE, Le multimédia et les réseaux ad hoc :Evaluation et optimisation de qualité de service, Dspace de universite Djillali Liabes de SBA, 2012.
- [46] F. K. James and W. R. Keith, Computer networking : a top down approach featuring the internet, Boston Mass: Addison-Wesley, 2003.
- [47] W. Shih-Lin and T. Yu-Chee, Wireless Ad Hoc Networking: Personal-Area, Local-Area, and the Sensory-Area Networks, Auerbach Publications , March 28, 2007 .
- [48] P. Chander, K. Surender and K. Ravinder, "Wireless Multi-hop Ad-hoc Networks: A Review," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 16, no. 2, pp. 54-62, Mar-Apr. 2014.
- [49] "Arduino UNO Product Manual," [Online]. Available: <https://store.arduino.cc/arduino-uno-rev3>.
- [50] "u-blox neo-6m," [Online]. Available: <https://www.u-blox.com/en/product/neo-6-series>.
- [51] Gwiza, "FARANAUX ELECTRONICS," GPS Reciever Ublox NEO-6M MOD35, 5 July 2017. [Online]. Available: <https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/>.

Annexes

1. Les environnements mobiles

Un environnement mobile est un système composé de sites mobiles qui permet à ses utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques. Les réseaux mobiles ou sans fil peuvent être classés en deux catégories.

1.1. Les réseaux avec infrastructure

Le modèle de système intégrant des sites mobiles et qui a tendance à se généraliser, est composé de deux ensembles d'entités distinctes :

- Les "sites fixes" d'un réseau de communication filaire classique (wired network).
- Les "sites mobiles" (Wireless network). Certains sites fixes, appelés stations de base (SB) sont munis d'une interface de communication sans fil pour la communication directe avec les sites mobiles ou unité mobile (UM) localisés dans une zone géographique limitée, appelée cellule comme le montre la **Figure 25** comme indiqué ci-dessous.

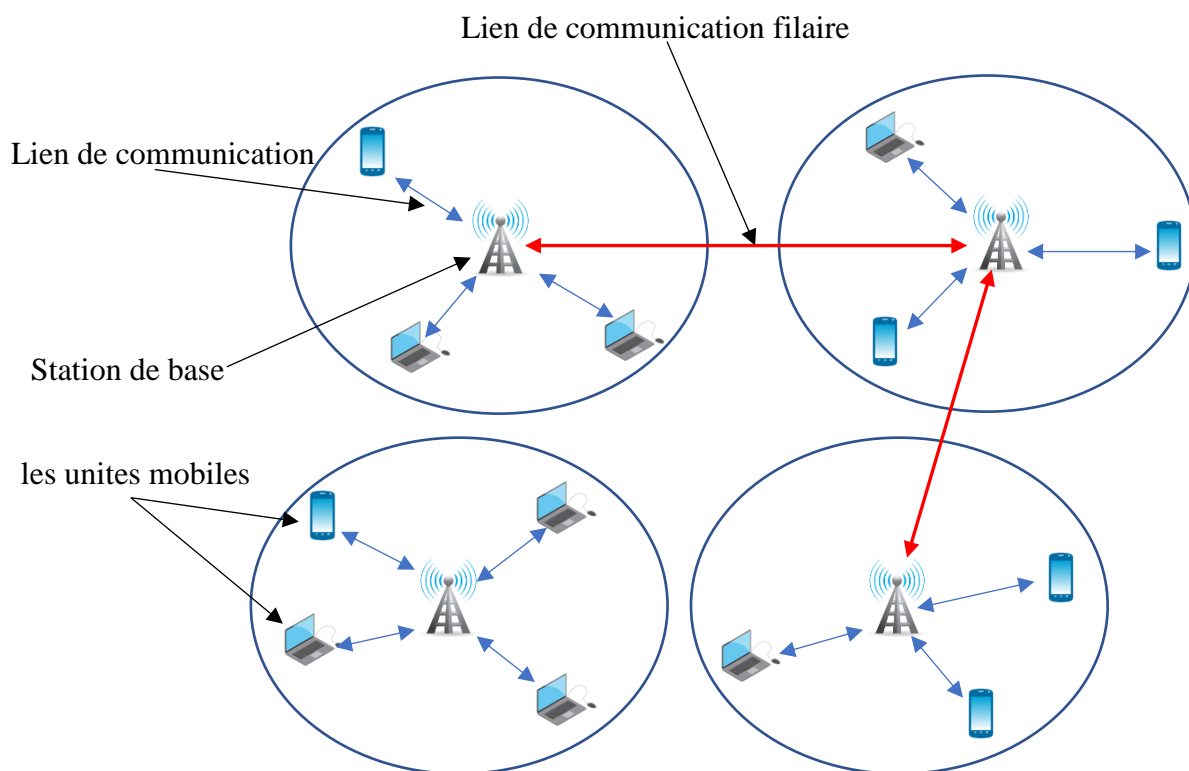


Figure 25 : Le modèle de réseaux mobile avec infrastructure

- A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Alors que les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire.
- Une unité mobile ne peut être à un instant donné directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée [9].

1.2. Les réseaux sans infrastructure (Ad Hoc)

Le modèle du réseau mobile sans infrastructure préexistante ne comporte pas l'entité « site fixe ». Tous les sites du réseau sont mobiles et communiquent d'une manière directe en utilisant leurs interfaces de communication sans fil (). L'absence de l'infrastructure ou d'un réseau filaire composé de station de base, oblige les unités mobiles (UM) à se comporter comme des routeurs qui participent à la découverte et à la maintenance chemins pour les autres hôtes du réseau. Ce type de réseau est appelé : MANETs (Mobile ad hoc networks) comme le montre la **Figure 26** comme indiqué ci-dessous.

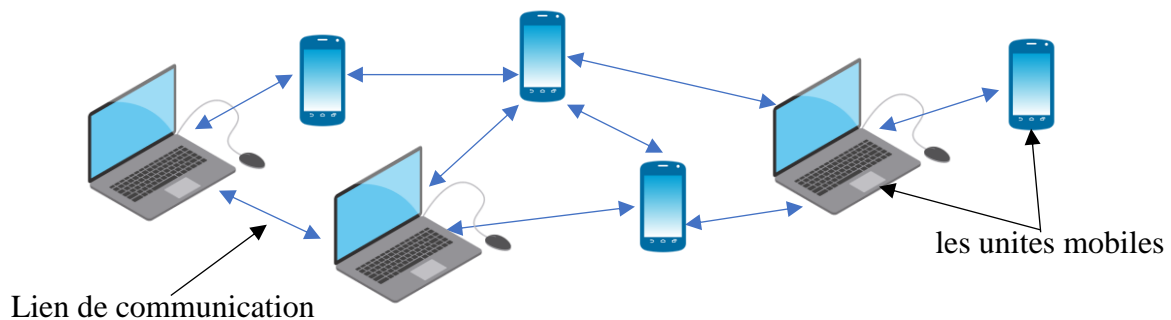


Figure 26 : *Le modèle de réseaux mobile sans infrastructure*