République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université 20 Août 1955 - Skikda

Faculté des Sciences / Département d'Informatique

Mémoire

Pour l'obtention du diplôme de master académique en informatique

Option : Réseaux et systèmes distribués

Intitulé :

Dissémination des messages d'urgences dans les réseaux VANET par le biais du Protocole SCTP

Réalisé par:

Belbali Ahlem

Mechehoud Fatima Zahra

Encadré par:

Mme: Remichi A

2022/2023

Remerciement

Nous tenons tout d'abord à remercier le bon Dieu tout puissant de nous avoir aidés à réaliser ce modeste travail.

Nous sommes extrêmement reconnaissants envers le professeur Remichi Amina. On le remercie pour le sujet très passionnant qu'il nous a proposé, nous permettant ainsi d'enrichir nos connaissances, on le remercie également pour la disponibilité sans réserve dont il a fait preuve à notre égard, les nombreuses discussions qu'on a eues avec lui ainsi que les précieuses et judicieuses aides.

Au terme de réalisation de ce mémoire, nous tenons à adresser nos remerciements :

Aux membres de jury d'avoir accepté l'évaluation de ce travail.

A toute personne ayant participé, de loin ou de près à la réalisation de ce mémoire.

Merci

Dédicaces

Grace à Dieu le tout qui puissant qui nous a aidés vers le droit chemin et nous a donné le courage et la patience pour finir ce travail.

Je dédie ce modeste travail à :

Mes très chers parents : « Salah, Zineb ».

Mes Sœurs : « Loubna, Fayza, Roukaya ».

Mon Frères : « Mehdi ».

Mes chers petits nièces « Amin, Neriman, Sami, Iyad, Miral, Zin et Israa ».

Mon amie de ce travail : « Ahlem ».

Fatima Zahra

Dédicaces

Grâce à dieu tout puissant, nous dédions ce modeste travail à toutes les personnes qui nous ont aidés de près ou de loin à la réalisation de ce mémoire plus particulièrement.

À l'âme de mon défunt père. Si Dieu le veut, je t'ai accordé ton souhait et je ne t'ai pas déçu pendant que tu étais dans ta tombe. Que Dieu ait pitié de vous et accorde votre repos au ciel et à ma bien-aimée, ma chère mère. Autant de phrase expressives soient elles ne sauraient montrer le degré d'amour et d'affection que je prouve pour vous. Vous n'aviez pas cassé de me soutenir et de m'encourager durent toutes les années de mes études, vous m'avez inculqué le sens de la responsabilité, de l'optimisme et de la confiance en soi et vos Conseil qui m'ont toujours guidé dans ma vie, je vous dois ce que je suis aujourd'hui et ce que je serai demain, je ferai tout pour que je serai votre fierté.

À mon encadrante Amina Remichi pour ces consécrations et ces conseils et sa touche encourageante et optimiste toute au long de la réalisation de ce projet.

À mon binôme F. Zahra pour son aide et sa compréhension et toute la famille Mechehoud.

A mes sœurs Rahma et Radia, leurs maris Djamél et Houssém et leurs enfants Karim, Dania,Mélék,Khalil .Pour leurs encouragements, leurs conseils et la confiance en ma personne, Je leur souhaite aussi que de succès dans leur vie.

À mon fiancé Aissa qui m'a pris la main vers ce que je veux et m'a redonné confiance en ma capacité à progresser.

À mes amies, Inès, Nour, Achwak, Manar, Inès sdj, pour tous les moments partagés avec mois et leur Soutien je leur souhaite que de bonheur et de succès dans leur vie.

Ahlem

Résumé

Les réseaux ad hoc véhiculaires (VANETs) permettent le partage de différents types de données entre les véhicules, de manière collaborative. Dans ce mémoire nous nous intéressons aux applications de sûreté et de sécurité routière, dédiées à l'échange des informations sur l'état de l'environnement routier.

L'objectif principal de ce travail vise à disséminer des messages d'urgences en temps réel par le biais du mécanismes de QOS**services différenciés** (DiffServ).

nous intégrons un mécanisme de gestion des files d'attentes en mode Priority Queuing au niveau du protocole SCTP afin de gérer ses différent Flux.

Les résultats prometteurs de nos simulations réalisées avec le simulateur OMNET++ ont démontré une amélioration concernant la délivrance des messages d'alerte en fonction de la densité du réseaux (le nombre de nœud mobile) , et de la vitesse des nœuds (véhicules)

Mots-clés : réseaux ad hoc, réseaux ad hoc véhiculaire (VANET), qualité de service (QOS), protocole SCTP, ordonnancement, gestion file d'attente, file prioritaire.

Summary

Vehicular ad hoc networks (VANETs) allow the sharing of different types of data between vehicles, in a collaborative way. In this dissertation, we are interested in road safety and security applications, dedicated to the exchange of information on the state of the road environment.

The main objective of this work is to disseminate emergency messages in real time through the QOS differentiated service (DiffServ) mechanism.

we integrate a mechanism for managing queues in Priority Queuing mode at the level of the SCTP protocol in order to manage its different flows.

The promising results of our simulations carried out with the OMNET++ simulator have

demonstrated an improvement in the delivery of alert messages depending on the density of the networks (the number of mobile nodes), and the speed of the nodes (vehicles)

Keywords: ad hoc networks, vehicular ad hoc networks (VANET), quality of service (QOS), SCTP protocol, scheduling, queue management, priority queue.

Des Illustrations

1. Liste des figures

2. Liste des Tableaux

Liste des abréviations

Aloha	**Areal Locations of HazardousAtmospheres**
C++	**language source code file**
DARPA	**Advanced ResearchProjects Agency**
GLoMo	**Global Mobile Information Systems**
IBSS	**Independent Basic Service Set**
IETF	**Internet Engineering Task Force**
IP	**Internet Protocol**
IPV4	**Internet Protocol, version 4**
IPV6	**Internet Protocol, version 6**
LPR	**LowCostPacket Radio**
MANet	**Mobile Ad hoc Net Work**
NTDR	**Near Term Digital Radio**
OBU	**On-Board Unit**
OMNet++	**Online Summit**
P2P	**Peer to Peer**
PMTU	**Path Maximum Transmission Unit**
PRNet	**Packet Radio Network**
QoS	**Quality of Service**
RSU	**Road Side Unit**
SCTP	**Stream Control Transmission Protocol**
SIG	**SpecialInterest Group**
SSN	**Stream SequenceNumber**
STI	**Systèmes de Transport Intelligents**

SURAN	**Survivable Radio Networks**
TA	**Trusted Authority**
TSN	**Transmission SequenceNumber**
TTL	**Trusted Authority**
V2I	**Vehicle to Infrastructure**
V2V	**Vehicle to vehicle**
V2P	**Vehicle to person**
VANet	**Vehicular Ad-Hoc Network**
WSN	**Wireless Sensor Network**

Table des matières

Introduction générale

L'avènement des technologies sans fil d'aujourd'hui ouvre des perspectives passionnantes dans les télécommunications. Les progrès récents des communications sans fil ont rendu possible la manipulation d'informations via un dispositif mobile accédant à un réseau via une interface de communication sans fil.

L'environnement mobile offre une plus grande flexibilité de travail. En particulier, il peut se connecter à des sites de réseau où le câblage est difficile ou impossible (par exemple, en utilisant des composants mobiles). Contrairement à l'environnement statique, l'environnement mobile permet aux appareils de se déplacer librement entre les segments du réseau et n'est pas limité par l'emplacement de l'utilisateur, La mobilité croissante des personnes et des biens a un coût sociétal très élevé en termes de congestion routière, de décès et de blessés chaque année.

Dans ce contexte, les systèmes de transport intelligents (STI) sont considérés comme une technologie clé pour accroître la sécurité, améliorer les infrastructures de transport et fournir des informations essentielles sur la sécurité aux usagers de la route

les réseaux ad hoc véhiculaires VANET (Vehicular Ad hoc NETwork) ont émergé en tant que nouvelle technologie prometteuse qui utilise les véhicules comme des nœuds pour créer un réseau mobile. Ces véhicules sont équipés d'interfaces sans fil leur permettant de communiquer entre eux. En effet, les VANETs peuvent être utilisés pour étendre la portée des informations de sécurité (messages d'alerte, informations sur les anomalies, etc.)

Des actions sont proposées pour mettre en œuvre la qualité de service ainsi que les besoins des utilisateurs.

Dans ce projet nous nous intéressons au problème de la dissémination des messages d'urgences (messages d'alertes) dans les réseaux VANETs, Afin de palier au problème de perte des messages suite à la congestion du réseau et d'améliorer la délivrance de ces derniers.

nous avons étudier les mécanismes de QOS, et nous avons procéder à l'intégration du **mécanisme services différenciés** (DiffServ) au niveau du protocole SCTP en utilisant la

gestion de files d'attentes en mode Priority Queuing, afin de gérer les différents flux, ce mécanisme est détaillé dans le chapitre 3.

Ce document contient 4 chapitres:

Dans le premier chapitre nous présentons les environnements mobiles et les principaux concepts liés à ces environnements. Nous commençons par la définition de l'environnement sans fil et la présentation des deux classes qui le constituent. Le concept de réseau ad hoc et le problème de mobilité, en particulier les caractéristiques et les applications de l'environnement ad hoc et ses types en situons les domaines d'utilisation des réseaux ad hoc.

Dans le 2ème chapitre, nous présentons le réseau ad hoc véhiculaire, en commençons par ses caractéristiques, les modes de communications des Vanets, ainsi que leur objectif.

Nous citons le protocole du transport SCTP.

La notion de la qualité du service (QOS), Nous commençons par la définition de la notion et l'amélioration de QOS, nous citons les critères déterminants du QoS. Nous terminons le chapitre par les principaux modèles de QoS utilisés pour le réseau ad hoc.

La présentation de la proposition pour disséminer les messages d'urgences par le biais des mécanismes de QOS au niveau du protocole SCTP est détaillée dans le chapitre 3.

Dans le chapitre 4, nous présentons le simulateur OMNET++ et les résultats de simulation de la solution proposée.

Chapitre 01: Les Réseaux Sans Fils

Introduction

L'essor actuel de la technologie sans fil offre une nouvelle perspective domaine des télécommunications. Évolution récente des moyens Communication sans fil (communication entre machines sans connexion filaire) Permet la manipulation d'informations via des unités de calcul dynamiques présentent des caractéristiques spécifiques (faible capacité de stockage, énergie autonome...) et accès au réseau via une interface de communication sans fil, donnant ainsi naissance à un nouvel environnement de communication appelé environnement mobile sans fil. Dans ce chapitre, nous verrons l'importance des réseaux sans fil et spécifiquement les réseaux ad hoc, et Différentes technologies telles que les réseaux personnels sans fil, le Web MAN, LAN et WAN, qui seront classés selon leur couverture, et après avoir énuméré certains des avantages du sans fil.

I. Réseaux Sans Fils

Un réseau sans fils est système de communication qui repose sur la transmission du signal sur un support sans la présence de fils ou de câbles entre l'expéditeur et le récepteur. Les supports de communication possibles pour la communication sans fil comprennent l'air, l'eau ou le vide. Les communications sans fil peuvent prendre en charge un degré élevé de mobilité et de flexibilité de déploiement, ils sont donc le principal moyen de communication de choix pour les réseaux ad hoc et de capteurs. L'attractivité des communications sans fil, l'absence de fils, présente également des inconvénients, à savoir la vulnérabilité aux interférences et au bruit de fond lors de la traversée du support sans fil. En conséquence, la qualité de signal attendue d'une liaison de communication sans fil est relativement inférieure, moins stable et moins prévisible qu'une liaison filaire. Les communications sans fil sont également intrinsèquement moins sécurisées que les communications filaires.(1)

1. Classification des réseaux sans fils

Les réseaux sans fil peuvent être classés selon la zone de couverture (le périmètre géographique offrant une connectivité) ou selon l'existence d'une infrastructure de communication.(2)

1.1. Suivant la portée (zone de couverture)

En fonction de cette caractéristique les réseaux sans fils peuvent être classés en 4 catégories :

- **Les réseaux personnels sans fil (**WPAN pour Wireless Personal Area Network**)** : technologie Bluetooth, infrarouge.
- **Les réseaux locaux sans fil (**WLAN pour Wireless Local Area Network**)** : technologie Wifi.
- **Les réseaux métropolitains sans fil (**WMAN pour Wireless Metropolitain Area Networks**)** : technologie BLR (Boucle Local Radio).
- **Les réseaux étendus sans fil** (WMAN pour Wireless Wide Area Networks) technologie GPRS.(3)

1.2. Suivant l'infrastructure

Selon l'infrastructure d'un réseau, on peut distinguer deux types de réseaux sans fils :

- **Réseaux avec infrastructure (cellulaire):** En mode infrastructure, les unités mobiles communiquent avec les unités fixes appelés points d'accès AP uniquement via des liaisons sans fil, L'ensemble composé du point d'accès et de l'unité mobile est appelé l'ensemble de station de base. A chaque station de base correspond une cellule à partir de laquelle des unités mobiles transfèrent des messages. La Figure 1 illustre un réseau avec infrastructure.

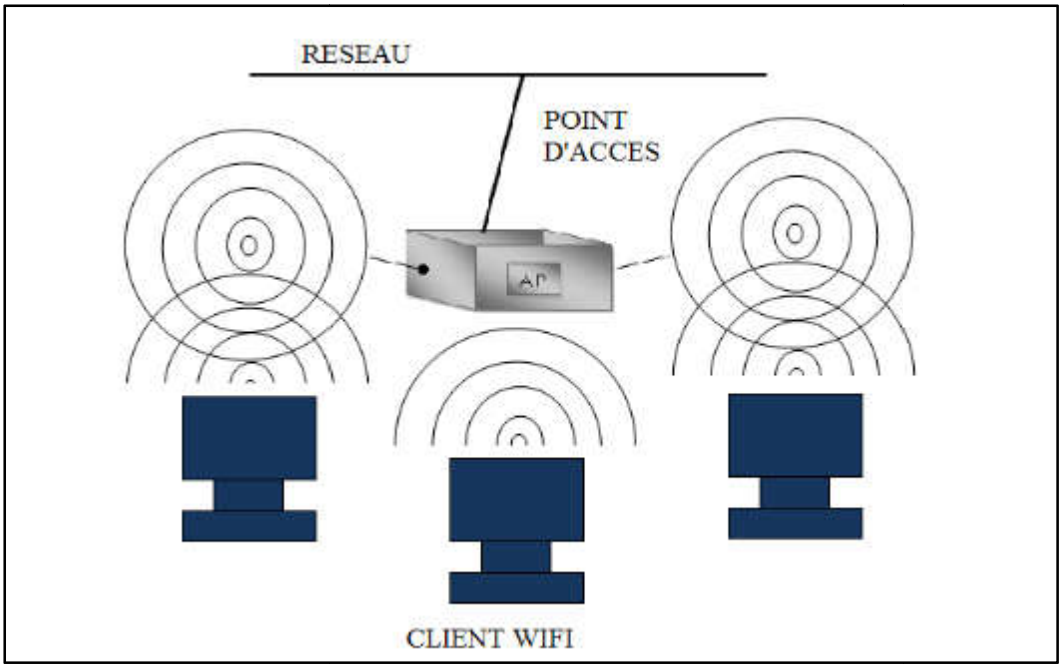

Figure 1 le mode sans infrastructure (IBSS)

➢ **Réseaux sans infrastructure (ad hoc):** Les réseaux mobiles sans infrastructure, appelés réseaux Ad hoc ou IBSS (Independent Basic Service Set), dans un modèle sans infrastructure, il n'y a pas de concept de sites fixes ou de points d'accès. Dans ces tous les sites du réseau sont connectés les uns aux autres pour construire un réseau peer-to-peer (P2P pour Peer to Peer). Par conséquent, chaque machine agit à la fois comme client et point d'accès, ce type de réseau est appelé un réseau Ad-Hoc. La figure 2 présente le schéma d'un réseau sans infrastructure.

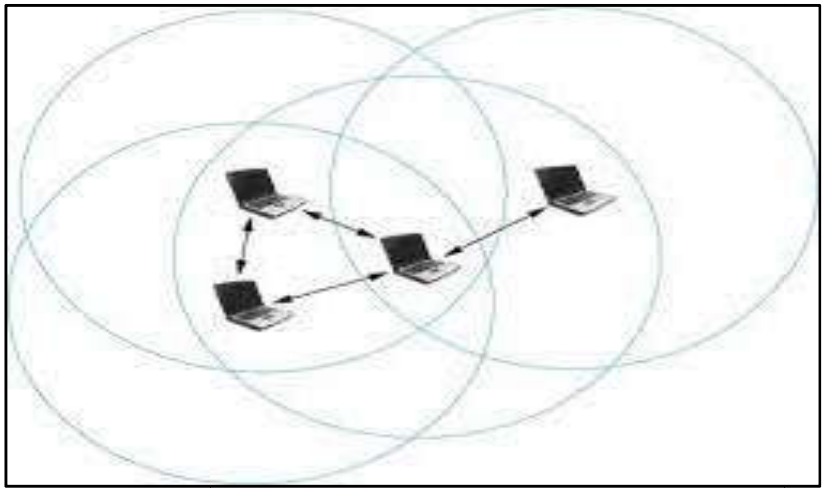

Figure 2 le mode sans infrastructure (IBSS)

Réseau cellulaire	Réseau sans fils Ad hoc
Réseau d'infrastructure	Réseau sans infrastructure
Sites cellulaires et station de base fixes et pré-localisés	Aucune station de base et déploiement rapide
Topologie de réseau dorsale statique	Topologies de réseau hautement dynamiques avec multi-sauts
Environnement relativement attentionné et connectivité stable	Environnement hostile (bruit, pertes) et connectivité irrégulière
Planification détaillée avant l'installation de la station de base	Le réseau ad hoc forme et s'adapte automatiquement aux changements
Coûts d'installation élevés	Couts rentable
Plus de temps de configuration	moins de temps d'installation

Tableau 1 Tableau comparatif entre réseau cellulaire et Ad hoc(4)

II. Le réseau ad hoc

1. Définition du réseau ad hoc

Un réseau ad hoc est un réseau sans fil composé de deux nœuds ou plus, capables de communiquer entre eux sans aucune administration centralisée contrôlée par des points d'accès. Chaque nœud dans le réseau fonctionne, à la fois, comme routeur et hôte . Ainsi, une absence d'infrastructure fixe laisse la place à une auto-organisation arbitraire des nœuds. En ajoutant le facteur de mobilité, on nomme le réseau MANET.(5)

2. Historique du réseau ad hoc

Le réseau traditionnel, avec son infrastructure fixe, présente différents défis dans son déploiement. On peut citer, par exemple, les pannes imprévisibles des stations de base pouvant engendrer une interruption de service dans le réseau. Dans le milieu militaire, un tel cas rendrait le réseau vulnérable et non robuste.

En 1972, après un premier projet par Defense Advanced ResearchProjects Agency (DARPA), la première génération du réseau ad hoc est apparue et a été nommée Packet Radio Network (PRNET). PRNET a été associé avec un autre projet Areal Locations of HazardousAtmospheres (ALOHA). Ces deux projets ont abouti à des approches d'accès au médium et de routage de type vecteur de distance, testées dans un environnement militaire.

La deuxième génération du réseau ad hoc est apparue dans les années 1980. En 1983, le projet Survivable Radio Networks (SURAN) a été mis en place. Il visait à fournir une connectivité ad hoc en se servant des équipements avec un coût réduit et une faible énergie consommée. Également, le nombre de nœuds dans le réseau a été augmenté afin de vérifier et d'améliorer le facteur de mise à l'échelle. Ainsi, la technologie nommée LowCostPacket Radio (LPR) a été mise en place, en 1987 .

Dans les années 1990, la croissance du nombre d'ordinateurs et la disponibilité des équipements sans fil ont poussé les communautés scientifiques et industrielles à créer le concept de la commercialisation de la technologie ad hoc. L'accès du public à cette technologie est ainsi devenu une réalité. Les grandes sociétés informatiques et de télécommunication (Ericsson, IBM, Intel,...) ont formé le groupe **Special Interest** Group (SIG), ayant pour but de fournir des solutions pour le déploiement de la connectivité ad hoc entre des équipements avec des caractéristiques hétérogènes. S'ajoute à cela la poursuite de l'effort de DARPA qui a mis en place : Global Mobile Information Systems (GloMo) et Near Term Digital Radio (NTDR). Ces deux projets visaient à offrir un milieu de connexions multimédia entre des équipements portables.

Le comité d'IEEE 802.11 a adopté le terme ad hoc et les chercheurs ont commencé à voir la possibilité de déploiement de réseaux ad hoc avec des nouvelles applications. En plus en 1998, le groupe de travail Mobile ad hoc Network (MANET) a été créé au sein d'Internet Engineering Task Force (IETF) afin de fournir des standards valides de protocoles de routage basés sur la technologie IP dans le réseau ad hoc.

3. Avantages des réseaux Ad hoc

Les avantages de cette technologie sont nombreux du fait qu'il n'y a pas besoin d'infrastructure préexistante (6):

- Les réseaux ad hoc peuvent être déployés dans un environnement quelconque.
- Le coût d'exploitation du réseau est faible : aucune infrastructure n'est à mettre en place initialement et surtout aucun entretien n'est à prévoir.

• Le déploiement d'un réseau ad hoc est simple : ne nécessite aucun pré requis puisqu' il suffit de disposer d'un certain nombre de terminaux dans un espace pour créer un réseau ad hoc, et rapide puisqu' il est immédiatement fonctionnel dès lors que les terminaux sont présents.

• La souplesse d'utilisation : est un paramètre très important puisque les seuls éléments pouvant tombés en panne sont les terminaux eux-mêmes. Autrement dit, il n'y a pas de panne "pénalisante" de manière globale (une station qui sert au routage peut être remplacée par une autre si elle tombe en panne).

4. Inconvénients des réseaux Ad hoc

Même si les perspectives pour les réseaux ad hoc sont prometteuses, plusieurs contraintes restent encore à traiter(6) :

• La connectivité limite les possibilités de communication. Ainsi, deux stations ne sont joignables que s'il existe un ensemble de stations pouvant assumer la fonction de routeur afin de faire suivre les paquets de données échangées entre les deux stations.

• Les liens entre les stations ne sont pas isolés les uns des autres et polluent le voisinage, par diffusion, lors de chaque émission/réception de données. Par conséquent, tout paquet de diffusion émis vers une station en cours de communication (que le paquet lui soit destiné ou pas) va altérer la communication de cette station. La diffusion est un facteur qui alourdit aussi d'autres paramètres tels que la bande passante et la consommation de batterie.

• La sécurité dans les réseaux ad hoc est difficile à contrôler, notamment parce que dans l'interface air l'écoute clandestine est très simple à réaliser.

• Enfin, la faible autonomie des batteries constitue un frein à une utilisation longue du terminal et à la mise en place de nouveaux services. C'est une contrainte qui existe certes dans les réseaux de type GSM ou UMTS, mais qui est plus forte dans les réseaux ad hoc, puisque les ressources énergétiques sont mises en commun même pour les besoins du routage. Nous nous intéressons dans cette thèse, plus spécialement, à ce dernier point. Nous proposons, dans le dernier chapitre, des solutions permettant de mieux gérer cette consommation des batteries.

5. Le concept d'auto-organisation

La grande évolution dans les réseaux sans fil, des réseaux traditionnels vers les réseaux ad hoc, est basée principalement sur le mécanisme distribué. Le caractère arbitraire oblige chaque nœud dans le réseau ad hoc d'avoir recours à un mécanisme fiable d'auto-

organisation. Qu'on le considère comme un but ou comme un moyen, le concept d'auto-organisation est primordial afin d'aboutir à une persistance dans le réseau ad hoc. Ainsi, une compréhension profonde de concept est nécessaire.

De manière abstraite, l'auto-organisation est présenté comme une émergence de la configuration globale du système à partir des seules collaborations et interactions de divers 5 dispositifs élémentaires du système.

Cette définition implique plusieurs autres concepts. En fait, on ne peut pas aboutir à une auto organisation fiable sans avoir recours à l'auto configuration, qui est présentée par les méthodes de reproduction des configurations adéquates aux circonstances environnementales. Dans le cas du réseau ad hoc, ces circonstances peuvent être l'état et la qualité de la connexion. En cohérence avec l'auto-configuration, le concept de l'auto-gestion impose aux différents dispositifs d'être toujours en lien avec les paramètres propres du système. Ensuite, l'auto-optimisation prend place afin de définir les choix optimaux des méthodes en se basant sur le comportement du système. Le dernier concept est l'adaptation aux conditions du milieu. Dans le réseau ad hoc, on peut citer comme exemple le nombre des nœuds voisins pour chaque nœud.

6. Les classifications du réseau ad hoc

D'un réseau ad hoc à un autre, plusieurs facteurs diffèrent, dépendant du but et des caractéristiques voulues du déploiement. On peut avoir une diversité dans les procédures de communication (un saut ou plusieurs sauts), une variété des architectures, des configurations homogènes ou hétérogènes des nœuds et des zones de couvertures distinctes. Ainsi, une classification du réseau ad hoc, en se basant sur ces facteurs, est possible .

La première classification est basée sur la procédure de communication. En effet, le réseau où tous les nœuds sont dans leurs portées mutuelles est appelé réseau ad hoc à un saut. Chaque nœud peut communiquer directement avec n'importe quel autre nœud dans le réseau sans aucun intermédiaire. Ce type de procédure de communication nécessite généralement une énergie de transmission élevée. Dans le cas où le trafic doit passer par un nœud intermédiaire, afin d'arriver à la destination désignée, le réseau est appelé réseau ad hoc multi sauts. C'est le cas le plus répandu. Ajoutant le critère de mobilité dans le réseau, la sélection du trajet pour l'envoi de trafic de données serait de plus en plus complexe, d'où la nécessité

de déploiement des protocoles de routages capables de s'adapter, essentiellement, aux changements rapides de la topologie

La Figure 3 illustre la différence entre les deux procédures de communication. La communication multi-sauts réutilise le domaine spatiale et temporel, tandis que la communication à un saut se concentre uniquement sur la disponibilité temporelle de médium

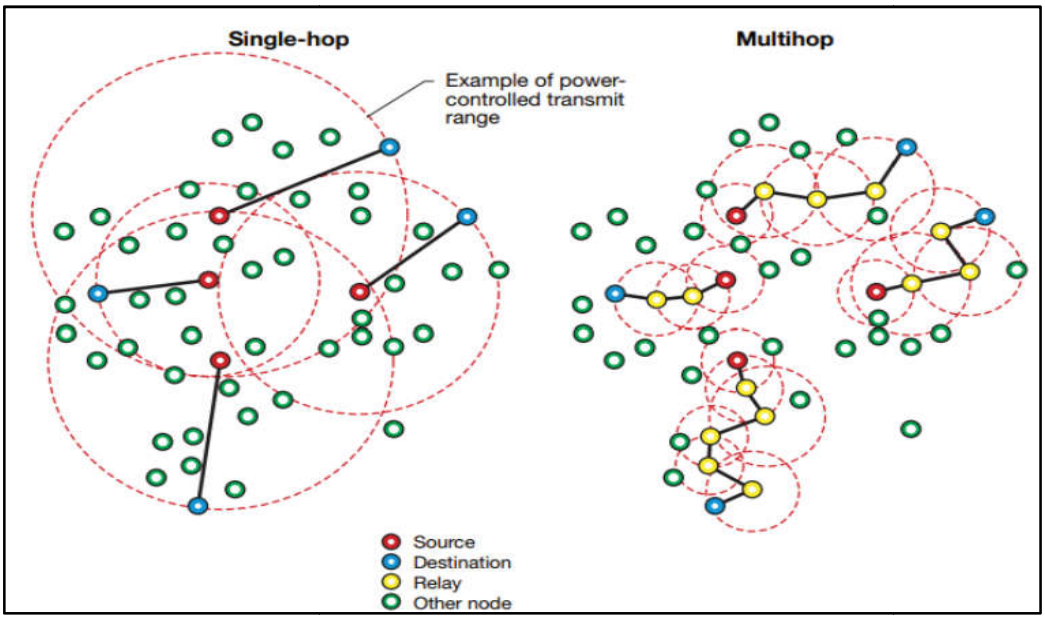

Figure 3 La communication multi-sauts et la communication àun saut

Aussi, les réseaux ad hoc peuvent avoir plusieurs architectures en se basant sur la topologie. En fait, cette classification se base sur l'existence ou non d'une hiérarchie entre les nœuds. On aboutit, en premier lieu, à une vue à plat du réseau où tous les nœuds fonctionnent de manière identique. La conception de ce type de réseau est relativement simple, mais une

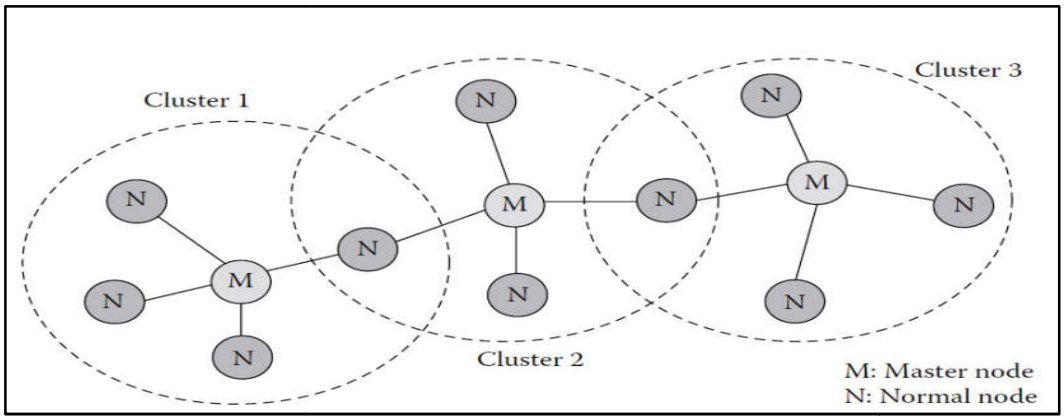

Figure 4 Une vue à hiérarchie du réseau ad hoc.

croissance 7 intensive du nombre de nœuds peut affecter le facteur de mise à l'échelle. Une deuxième vue du réseau ad hoc choisit de créer une hiérarchie . Tout d'abord, la procédure commence par la division du réseau en clusters liés entre eux. Chaque cluster confie à un noeud le rôle du chef (clusterhead) responsable de contrôle et de gestion des connexions entre les différents nœuds. Le routage serait simple puisqu'il est établit de passerelle à passerelle jusqu'à celle directement liée à la destination. L'apport de l'approche hiérarchique est essentiellement la facilité de maintien des informations topologiques vu que la grande part de gestion de connexions est supportée par le nœud chef. Ensuite, le déploiement des passerelles améliore le facteur de mise à l'échelle.

Enfin, une architecture agrégée du réseau ad hoc présente le troisième type de la classification basée sur la topologie . En fait, le réseau est décomposé en zones. Chaque nœud dans chaque zone possède deux niveaux d'identification. Le premier niveau est son identification propre. Le deuxième niveau présente l'identification de la zone.

L'apport de cette approche est la facilité de la maintenance de liens aisément localisés grâce à l'indicateur de zone. 8 La troisième classification se base sur la configuration matérielle de chaque nœud dans le réseau . On a deux types de configurations. La configuration homogène impose à tous les nœuds d'avoir les mêmes caractéristiques matérielles (le processeur, la mémoire...). La deuxième configuration est la configuration hétérogène. La différence entre les caractéristiques matérielles de chaque nœud entraine un fonctionnement différent. Ainsi, chaque nœud a ses ressources et ses politiques. Cette hétérogénéité peut concerner des groupes de nœuds au lieu d'un seul nœud. Ces groupes appartiennent à des réseaux déployant des technologies différentes (Bluetooth, WIFI, etc). La figure 5 illustre clairement cette hétérogénéité de configuration.

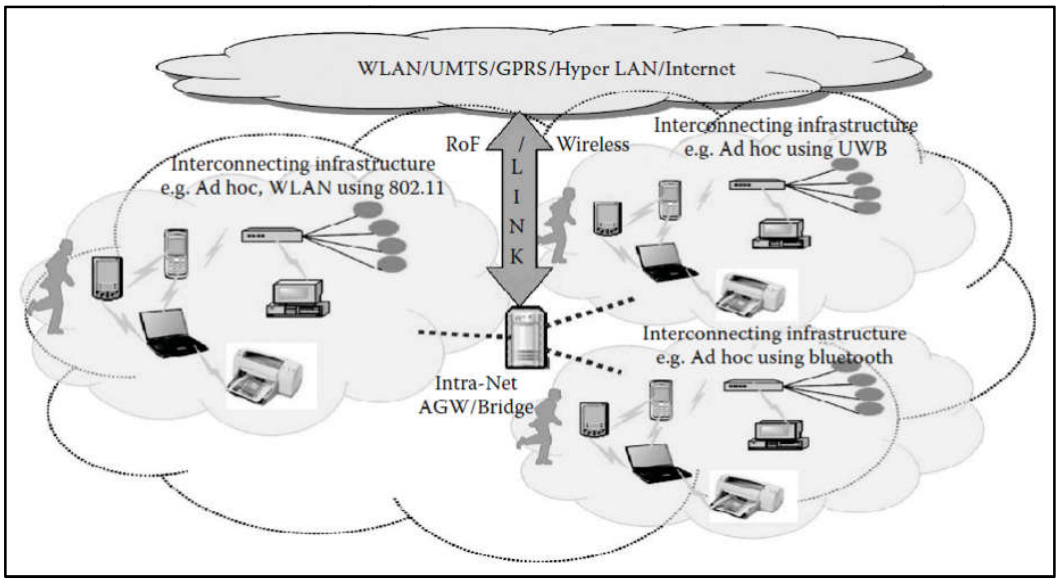

Figure 5 La configuration hétérogène du réseau ad hoc.

Cette diversité de classifications rend la technologie ad hoc plus flexible et applicable dans différentes applications. La section suivante souligne les applications les plus importantes.

7. Domaines d'utilisation des réseaux Ad hoc

7.1. Applications militaires

Comme le domaine militaire a déclenché l'étincelle de recherche pour le réseau ad hoc, le déploiement de la technologie ad hoc persiste grâce à son autonomie et sa robustesse. Afin de maintenir la communication dans un milieu de conflit, lors des déplacements rapides des 9 objets militaires, la connexion doit être fiable, rapide et sécurisée (Kumar Sarkar et al., 2013). La Figure 6 illustre la coordination entre les différents soldats dans un scénario de combat l'aide de la technologie ad hoc.

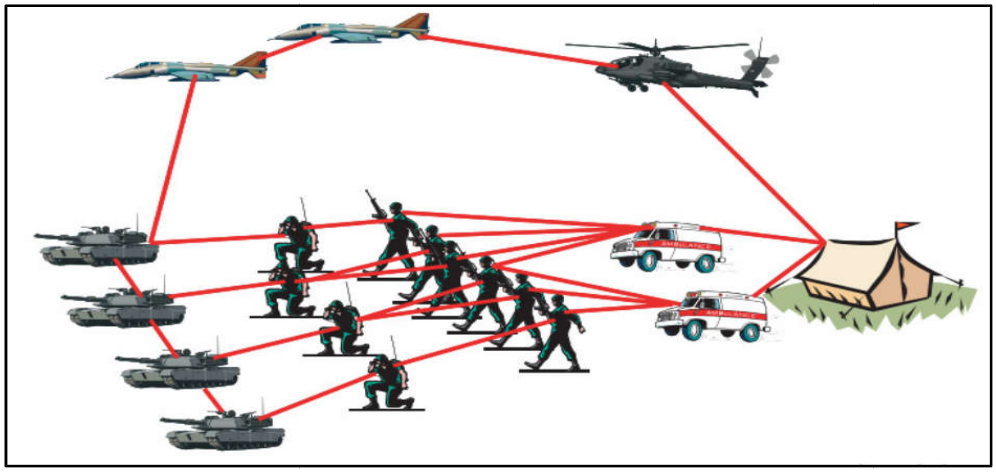

Figure 6 L'application militaire de la technologie ad hoc

7.2. Réseau Ad Hoc pour Plans d'urgences

La technologie ad hoc prend place dans les situations d'urgence, vu sa capacité de s'adapter à la mobilité aléatoire des nœuds et de s'auto organiser . Dans le cas des désastres naturels qui peuvent causer la destruction d'infrastructure existante, la technologie ad hoc aide à organiser les activités de sauvetage à travers l'acheminement des données par l'intermédiaire de connexions multi sauts. Aussi, vu que les communications vocales dominent dans ces cas, l'ad hoc doit supporter les applications temps réel. Autre application de la technologie est Wireless sensor network (WSN) qui est répandue dans les environnements ruraux et urbains. Plusieurs domaines ont profité de cette technologie, essentiellement où l'intervention humaine directe est impossible ou dangereuse. On peut citer le domaine industriel avec des conditions critiques comme la haute température, les réactions nucléaires et la haute pression. Afin de prévenir des désastres naturels, un déploiement de WSN peut fournir un contrôle en temps réel afin de prévenir des catastrophes.

La Figure 7 montre l'intégration de WSN dans un système pipeline souterrain responsable de la distribution du gaz naturel.

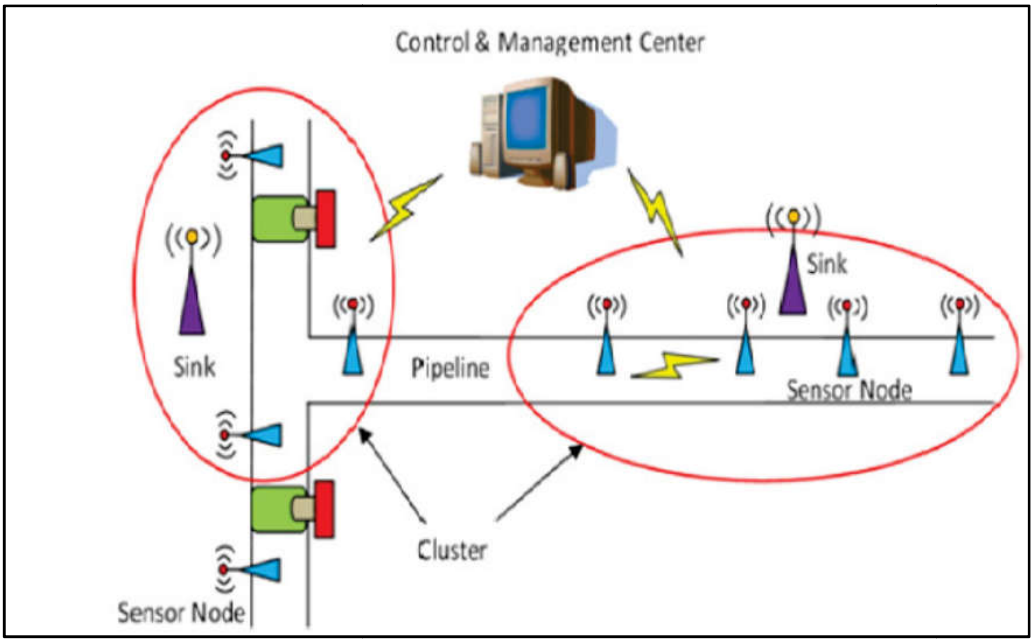

Figure 7 Un exemple de déploiement de WSN

7.3. Réseaux mobiles ad hoc (MANETs)

Un réseau mobile ad hoc (MANET) est un réseau continuellement auto-configurant, auto-organisant et sans infrastructure de dispositifs mobiles connectés sans fils. Ils sont parfois connus sous le nom de réseaux "à la volée" ou "réseaux spontanés**.**

7.4. Réseaux véhiculaires ad hoc (VANETs)

Les**VANETs** sont utilisés pour la communication entre véhicules et équipements routiers. Les réseaux véhiculaires ad hoc intelligents (InVANETs) sont un type d'intelligence artificielle qui aide les véhicules à se comporter de manière intelligente lors de collisions vehicle-to-vehicle ou accidents. Les véhicules utilisent des ondes radios pour communiquer entre eux, créant instantanément des réseaux de communication à la volée alors qu'ils se déplacent sur les routes.

8. Les contraintes du réseau ad hoc

Malgré ses avantages, le déploiement de la technologie ad hoc présente certains problèmes liés essentiellement au caractère imprévisible des nœuds, au médium et au routage. Parmi ces problèmes, on peut citer :

- **L'interférence** : Si des transmissions se font sur une même fréquence, des interférences peuvent prendre place, ce qui mène à une perturbation des connexions et à une réduction dans la qualité de liens.
- **Le terminal caché** : Dans le réseau sans fil, la détection de collision ne peut pas être exploitée à cause de la nature du médium. Le problème de terminal caché peut avoir lieu. Par exemple, on prend le cas de deux nœuds S1 et S2 qui sont hors de portée l'un de l'autre et un nœud R1 joignable par les deux. S2 ne peut pas détecter une transmission de S1 vers R1.

Une collision peut avoir lieu si S2 émet vers R1, simultanément avec S1.

- **Le terminal exposé** : Ce problème est référé par l'impossibilité d'un premier nœud de transmettre, à cause de son estimation de l'occupation du canal par les transmissions d'un deuxième nœud placé dans la portée du premier.

La Figure 8 illustre le problème du terminal expo**sé** et du terminal caché

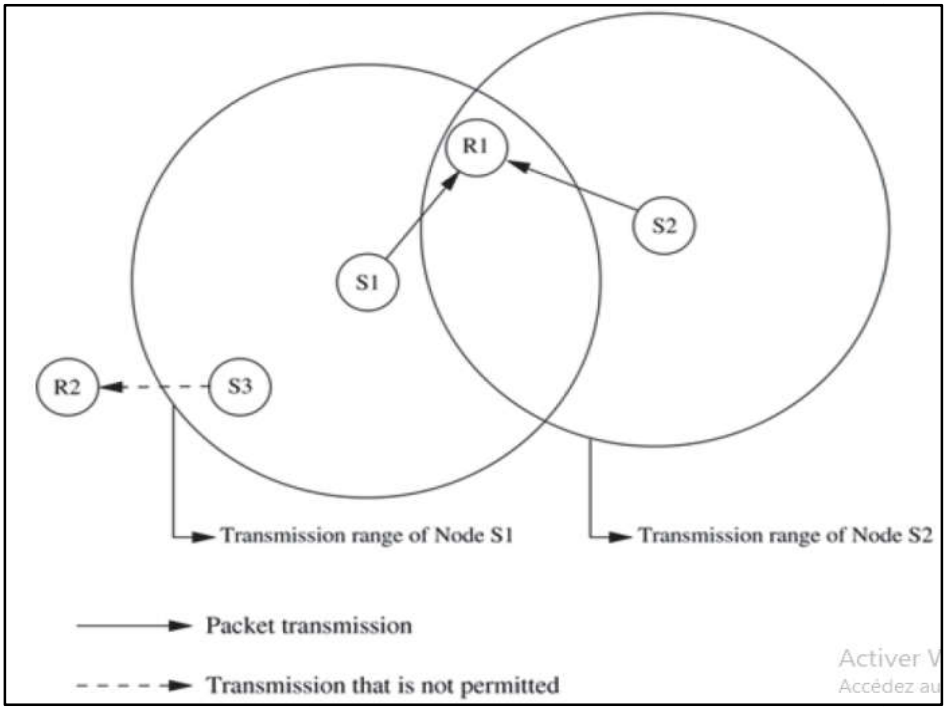

Figure 8 Le problème du terminal caché et du terminal exposé.

- **Débit** : Il est influencé par plusieurs facteurs dans le réseau comme l'occurrence des collisions et la disponibilité du canal. Ce défi s'aggrave avec le déploiement de certains services comme la communication vidéo et l'IPTV.

- Énergie : Les nœuds dans le réseau ad hoc sont caractérisés par des ressources d'énergie limitées. La consommation d'énergie accroit avec la mobilité des nœuds et l'acheminement multi sauts des paquets. Un contrôle et gestion de la consommation énergétique est important afin de garantir une continuité des services.

- La bande passante : Dans les réseaux sans fil, la bande passante est partagée par tous les nœuds. Ainsi sa disponibilité est affectée par le nombre des nœuds et les trafics à envoyer.

- **Routage** : La mobilité dans les réseaux sans fil ad hoc cause des variations aléatoires dans le voisinage immédiat de chaque nœud et dans la topologie du réseau entier. Ainsi, un défi est dans la conception des protocoles de routage capables de s'adapter rapidement à cette dynamique et à reconfigurer des trajets optimaux avec les minimums délais d'accès.

Conclusion

Dans ce chapitre nous nous sommes concentrés sur l'étude des environnements mobiles et les domaines d'application de la technologie de communication AdHoc.

Le réseau AdHoc offre beaucoup de simplicité et assez d'avantages par rapport aux autres réseaux (filaires) par sa facilité de déploiement et son coût réduit.

Le chapitre suivant sera consacré à la description de quelques approches et protocoles adaptatif dédiées aux réseaux VANETs.

Chapitre 02: Vue d'ensemble des réseaux adhoc véhiculaires

Introduction

L'avènement des nouvelles technologies de l'information et de la communicationà contribuer à l'amélioration des systèmes de transport. Le résultat de cette amélioration est ce que l'on appelle aujourd'hui les Systèmes de Transport Intelligents (STI).

Le développement de ces systèmes remonte à une quarantaine d'années. Les premières briques ont porté sur l'informations aux usagers. Plus récemment, avec l'implication de plus en plus forte de l'industrie automobile et la démocratisation des technologies de la communication les réseaux véhiculaires sont connu le jour.

Les réseaux véhiculaires sont une projection des systèmes de transports intelligents (Intelligent Transportation Systems - ITS). Les véhicules communiquent les uns avec les autres par l'intermédiaire de la communication inter-véhicule (V2V) aussi bien qu'avec les équipements de la route par l'intermédiaire de la communication d'équipement-à-Véhicule (V2I). Le but optimal est que les réseaux véhiculaires contribueront à des routes plus sûres et plus efficaces à l'avenir en fournissant des informations opportunes aux conducteurs et aux autorités intéressées.

I. Réseaux Ad Hoc Véhiculaires (VANET)

1. Définition d'un réseau VANET

La technologie sans fil a connu une progression rapide ces dernières années, ce qui a donné naissance à de nouveaux systèmes de communication, l'idée première a été de rendre les véhicules et les routes plus intelligents, L'architecture sur laquelle se base les systèmes de transport Intelligent est connue sous le nom de VANET (Vehicular Ad-Hoc Network) ou réseaux Ad-Hoc véhiculaire. Les réseaux véhiculaires ad hoc (VANETs) sont un type particulier de réseaux mobiles ad hoc (MANET), où les véhicules sont simulés comme des nœuds mobiles, ils ont leur propre modèle de mobilité contrôlée, asservi à la réglementation de la circulation automobile. Les véhicules se déplacent généralement plus rapidement que les autres nœuds de type MANET Étant donné que la portée du réseau sans fil de chaque véhicule peut être limitée à quelques centaines de mètres, les messages doivent traverser plusieurs nœuds pour fournir une communication de bout en bout sur de longues distances, La Figure 9 montre un exemple d'un réseau VANET.(7)

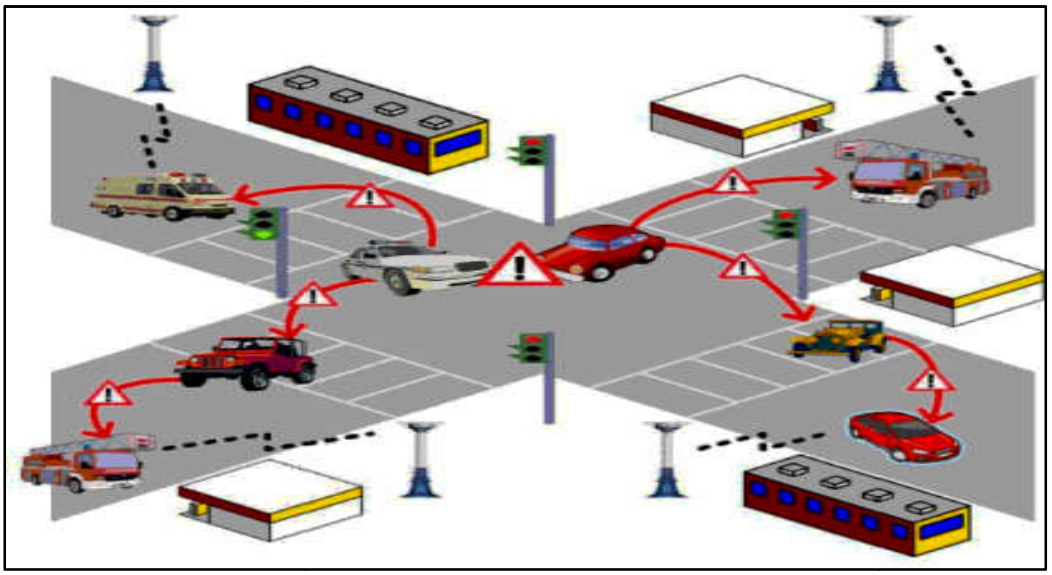

Figure 9 Un exemple d'un réseau VANET

2. Caractéristiques des VANETs

De nombreux objectifs de conception différente et parfois contradictoire doivent être pris en compte pour que les VANET assurent : (8)

- **Topologie hautement dynamique** : vitesse élevée des véhicules et possibilité de choisir entre plusieurs chemins définissent la topologie dynamique des réseaux VANET.
- **Mobilité :** La vitesse élevée des véhicules définit d'une part la topologie dynamique et d'autre part les besoins fréquents de l'unité de bord de route, alors, la connectivité est de courte durée.
- **Connectivité** : Une fois que nous avons un modèle de mobilité, nous n'avons pas encore terminé. Le modèle de mobilité peut avoir des caractéristiques différentes selon 8 l'architecture de la route, les autoroutes ou les environnements urbains. La communication dans ces situations doit être prise en compte.
- **Contraintes de délai :** En cas d'urgence, la livraison des messages à temps est un problème critique. Par conséquent, il est préférable de gérer de telles situations plutôt que de parler uniquement de débits de données élevés.
- **Capacité d'énergie et stockage** : Les nœuds des réseaux VANET ne sont pas limités en termes de puissance et de stockage comme c'est le cas dans les réseaux de capteurs, Cela signifie que ce type de réseau ne souffre pas de problèmes énergétiques même en cas d'arrêt du système par conséquent, l'optimisation du cycle de fonctionnement n'est pas aussi pertinente que dans les réseaux de capteurs.

3. Défis

Bien que les réseaux ad hoc véhiculaires partagent des caractéristiques communes avec les réseaux ad hoc et de capteurs conventionnels, comme l'auto-organisation et l'absence de contrôle central, les VANET présentent des défis uniques qui ont un impact sur la conception du système de communication et la sécurité de son protocole. Ces défis sont les suivants(9) :

3.1. Le nombre potentiellement élevé de nœuds

Si l'on considère les VANET comme la base technique des systèmes de transport intelligents, nous nous attendons à ce qu'une grande partie des futurs véhicules soient équipés de capacités de communication pour la communication véhiculaire. En tenant compte des unités routières potentielles supplémentaires, les réseaux VANET doivent être évolutifs avec un nombre très élevé de nœuds.

3.2. Forte mobilité et changements fréquents de topologie

Ces nœuds peuvent se déplacer à grande vitesse le long des routes et des autoroutes. Par conséquent, dans certains scénarios, comme lorsque des véhicules se croisent, le temps d'échange de paquets de données est relativement court, généralement de l'ordre de quelques secondes. De plus, les nœuds intermédiaires dans la chaîne sans fil multi-sauts du nœud expéditeur peuvent se déplacer rapidement cela représente le principal défi à relever lors de la conception des mécanismes de sécurité mis en œuvre dans ces réseaux. La route établie entre les deux nœuds est souvent interrompue, le réseau peut être divisé, l'impact de la mobilité La connectivité Internet reste l'une des principales difficultés des véhicules.

3.3. Exigences élevées des applications en matière de livraison de données

Les applications importantes de VANET sont exploitées pour la sécurité du trafic afin d'éviter les accidents de la route, incluant potentiellement la sécurité de la vie. Ces applications ont des exigences élevées en termes de temps réel et de fiabilité. Un retard de bout en bout de quelques secondes peut rendre une information de sécurité insignifiante. La perte de messages, par exemple en raison d'attaques de sécurité, peut affecter les décisions de mort ou de vie. En outre, ces applications sont généralement basées sur une distribution de données par diffusion inondation à portée géographique où les nœuds de destination sont ceux situés dans cette zone.

3.4. Sécurité

Pour les applications de sécurité, les informations contenues dans un message intéressent tous les usagers de la route et ne sont donc pas confidentielle, les capacités de communication des véhicules peuvent révéler des informations sur le conducteur ou l'utilisateur, telles que l'identifiant, la vitesse, la position et les habitudes de mobilité. Malgré la nécessité d'authentifier et de non-répudier les messages de sécurité, la vie privée des utilisateurs et des conducteurs doit être respectée, en particulier la confidentialité et l'anonymat de la localisation.

4. Modes de communication des VANETs

La sécurité est l'un des domaines d'intérêt de VANET qui connaît la croissance la plus rapide. L'Amélioration, la réactivité et la sécurité du conducteur en cas d'incidents routiers.

De nos jours, l'importance de la communication inter-véhicules est croissante en raison de son rôle dans l'organisation de la route, Pour concevoir une architecture de communication de réseaux de type VANETs, nous abordons trois modes essentiels (10):

4.1. Communication inter-véhicule (V2V)

Dans la communication V2V, véhicule à véhicule (V2V) les messages doivent être acheminés de la source vers une ou plusieurs destinations, Nous intéressons à la communication entre véhicules Une façon de propager l'information entre les véhicules très rapidement, chaque nœud qui reçoit cette information va simplement la rediffuser. Pour éviter la duplication infinie des paquets, chaque nœud ne diffusera un paquet donné qu'une fois au maximum. (11)

De plus un compteur de temps de vie (TTL) peut être utilisé pour limiter la zone où le paquet est distribué, La Figure 10montre un exemple d'un mode de communication inter-véhicule (v2v)

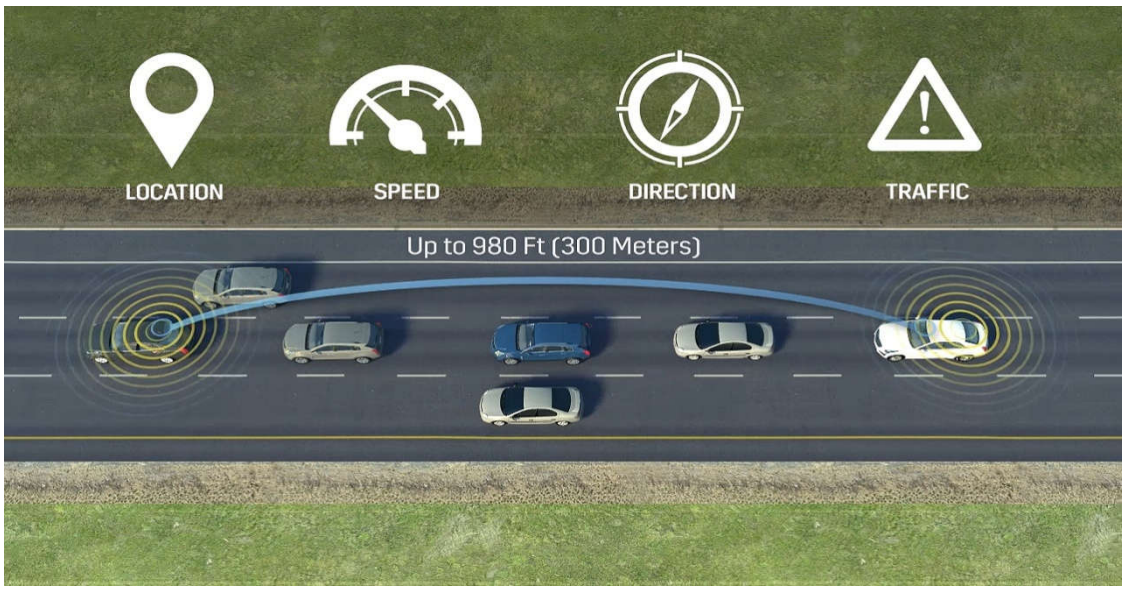

Figure 10 Un exemple d'un la communication inter-véhicule (V2V)

4.2. Communication véhicule-infrastructure (V2I)

La communication véhicule à infrastructure (V2I) est l'échange sans fil de données entre les véhicules et l'infrastructure routière, en utilisant des fréquences de communication dédiées à courte portée pour transférer des données.(12)

Ce mode de communication permet aux véhicules de partager des informations avec les composants qui prennent en charge le réseau routier peut fournir des avantages tel que les la réduction des embouteillages, des accidents dans les routes et les zones de construction. La Figure 11montre un exemple d'un mode de communication inter-véhicule(V2I).

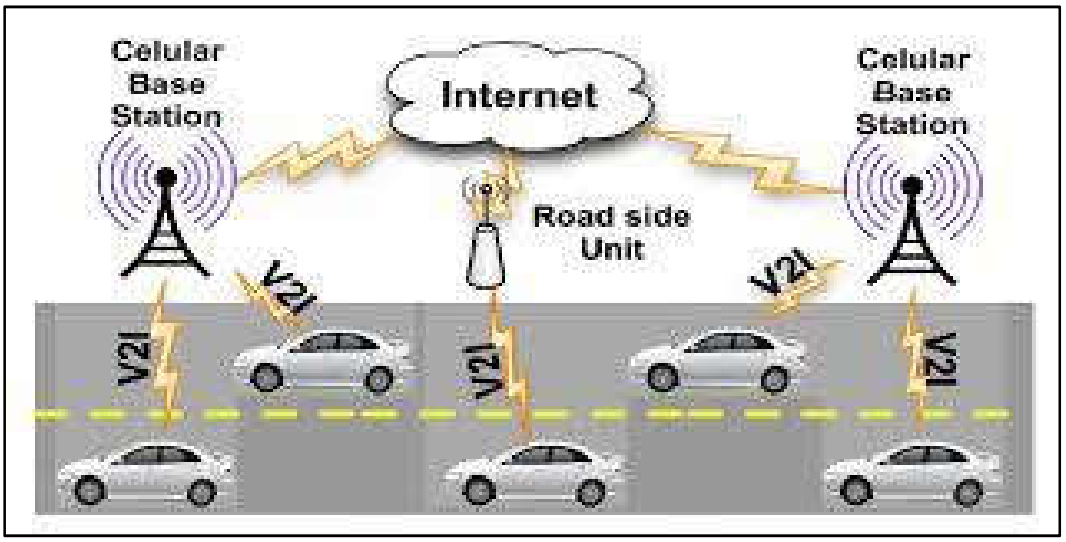

Figure 11 Un exemple d'un mode de communication inter-véhicule (V2I)

4.3. Véhicule-à-Réseau (V2N, Véhicule-to-Network)

Les technologies V2N (Véhicule to Network) vise à transmettre des informations entre les véhicules et les appareille qui appartiennent au réseau de communication.

De plus ces communications V2N garantissent l'accès à Internet et l'échange des donnés. Par conséquent, les véhicules peuvent les utiliser pour se connecter à des services distants. Ensuite, nous pouvons parler de communication Véhicule-à-Cloud (V2C, Véhicule to Cloud). Ces services Cloud peuvent notamment correspondre à des applications de gestion globale de sécurité routière ou de gestion du trafic routier. La Figure12 montre un exemple d'un mode de communication Véhicule-à-Cloud (V2C).(13)

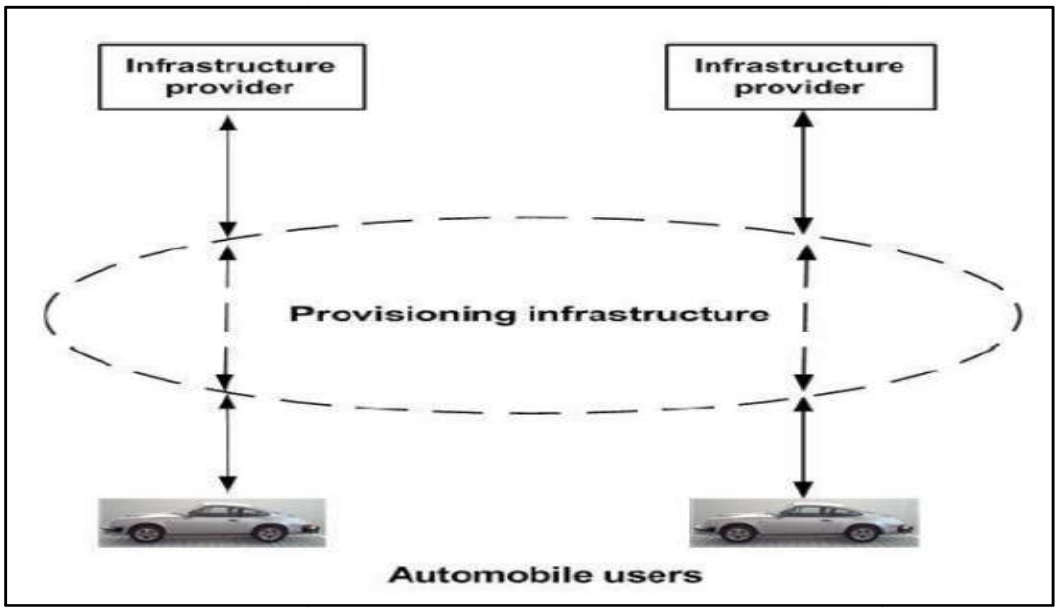

Figure 12 un exemple d'un mode de communication Véhicule-à-Cloud(V2C)

4.4. Communication hybride

C'est la combinaison de la Communication véhicule-infrastructure (V2I) avec la communication de véhicules à véhicule (V2V) pour faire une diffusion des données efficaces. Le principe est de rediffuser les mêmes paquets de données depuis l'infrastructure ou réduire le temps d'attente des véhicules demandeurs. Néanmoins, plusieurs véhicules demandeurs doivent concourir une bande passante limitée dans la zone de couverture d'un bord de route un unité (RSU). La Figure 13montre un exemple d'un mode de communication hybride.(14)

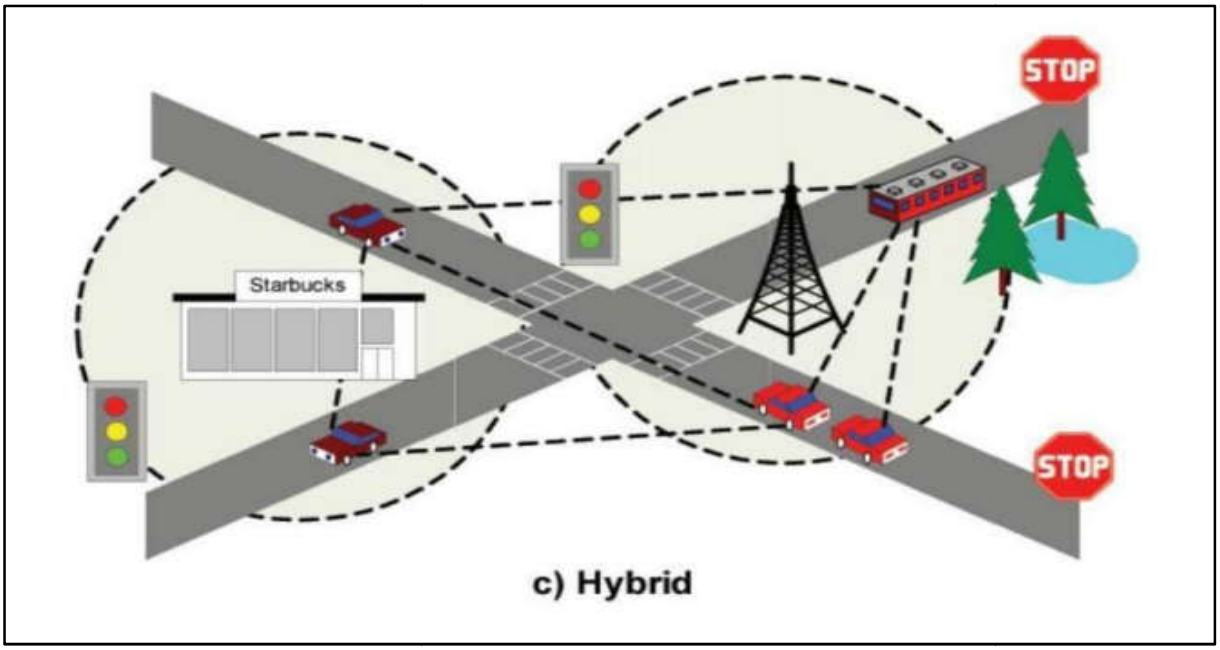

Figure 13 Un exemple d'un mode de communication hybride

4.5. Véhicule-à-Personne (V2P, Véhicule-to-Person)

Il s'agit des communications entre véhicules et usagers non motorisés de la route : piétons, cyclistes, etc. Ces communications doivent réduire les risques de collisions entre les véhicules et les usagers de la route. Pour cela, la communication V2P peut s'appuyer sur différentes technologies 14 (Bluetooth Löw Energy, Ultra Wideband, Cellular). Ces communications peuvent également prendre différentes formes. Tout d'abord, unilatérale, dans ce cas, une seule des entités est avertie de l'arrivée de l'autre entité (usager de la route ou véhicule). Ensuite, bilatérale, dans ce cas, les deux entités (usager de la route et véhicule) sont

informées du risque lié à l'arrivée de l'autre entité. La Figure14 montre un exemple d'un mode de communication Véhicule-personne V2P.(14)

Figure 14 Un exemple d'un mode de communication Véhicule-à-Personne V2P

5. Composants utilisés par les modes de communications

Station de bord de la route (RSU)

Un RSU est un dispositif ondulé qui est généralement fixé le long d'une route ou à un endroit spécifique. Le RSU est équipé d'un périphérique réseau pour une Communication à courte portée, qui peut également communiquer en diffusant des donnés lies à l'état du trafic et la route.

OBU (On-Board Unit)

OBU est un appareil à ondes, généralement installé sur le véhicule Utilisé pour échanger des informations avec RSU ou d'autres OBU et Comprend un processeur de contrôle des ressources (RCP) et des ressources.

Autorité centrale TA (trustedAuthority)

Une autorité centrale ou autorité de confiance est un tiers de confiance dont le rôle est de signer et d'émettre des certificats numériques. Dans certains cas, l'Autorité centrale (AC) peut également révéler l'identité de l'expéditeur du message

6. Objectifs de VANET

En général, les objectifs du VANET sont d'améliorer la sécurité et l'efficacité des systèmes de transport.

Les VANET fourniront différents services, dont certains sauveront des vies en prévenant les accidents, d'autres apporteront le luxe dans les voyages, et d'autres encore soutiendront les transactions commerciales. Certains services importants des VANET sont énumérés ci-dessous :

Mises à jour du trafic : VANET fournira des mises à jour localisées du trafic. Les conducteurs peuvent obtenir les dernières informations sur le trafic de loin. Ces informations peuvent aider les conducteurs à prendre des décisions pour éviter les embouteillages avant d'atteindre leur destination.

Suggestion d'itinéraire : Le réseau VANET fournira une analyse comparative des itinéraires possibles vers la destination. Ces informations peuvent aider les conducteurs à choisir le bon itinéraire.

Signal d'avertissement d'urgence : le réseau VANET aidera à diffuser des messages d'avertissement lorsque la voiture qui précède freine soudainement.

Objectif commercial : Les organisations privées peuvent utiliser le réseau VANET pour faire la publicité de leurs services et biens. Par exemple, les hôtels peuvent utiliser des RSU (unités routières) devant eux pour annoncer leurs services. Les clients peuvent réserver n'importe quel service à distance en utilisant le réseau VANET. Par exemple, un client peut réserver une chambre d'hôtel avant d'arriver à l'hôtel, en utilisant les VANET.

Signal d'alerte environnemental : Les réseaux VANET peuvent fournir des avertissements de danger environnemental. Tels que le verglas, les dommages aux routes, la construction du site, etc.

II. Protocoles de mobilité Dans les réseaux ad hoc

1. Protocoles de mobilité au Niveau IP

Le développement des extensions du protocole IP pour la mobilité des hôtes a débuté au sein de l'IETF (*Internet Engineering Task Force*) au début des années 90. Les extensions du protocole IP sont regroupéesdans le protocole appelé *Mobile IP*, le même nom que le groupe de travail qui les a introduits. L'Internet acontinué d'évoluer et des nouvelles problématiques et contraintes sont apparues.

Mobile IP est une solution qui intervient au niveau IP et qui fournit la transparence vis-à-vis des couches supérieures, y compris le protocole TCP. L'autre point important pris en compte dès le début dans la conception de Mobile IP, au moins pour sa version v4, a été la compatibilité avec les hôtes correspondants(15).

1.1. Mobile IPv4

(Julien Danjou — Renaud Galante mobile IP)

Le but de Mobile IP est de rendre joignable à tout instant un périphérique mobile.

Pour que cela soit possible, il faut que ce périphérique garde toujours un identifiant unique quel que soit le réseau dans lequel il se trouve. En d'autres termes, l'équipement doit toujours avoir la même adresse IP(15).

Pour que cela soit possible, quatre entités doivent être présentes :

– le périphérique mobile lui-même ;

– les correspondants de ce dernier ;
– un routeur situé dans le réseau administratif du mobile appelé agent mère ;
– un routeur situé dans le réseau visité par le mobile appelé, agent relais. Ce dernier est utilisé uniquement dans le cas de la mobilité IPv4.
Lorsque le mobile envoie une requête vers un correspondant, l'adresse source contenue dans le paquet IP envoyé ne contient pas l'adresse IP courante du mobile, mais l'adresse de son agent mère. Le site distant enverra donc sa réponse vers l'agent mère. Celui-ci a deux possibilités pour transmettre le paquet.
1. Envoyer le paquet vers un agent relais. Ce nœud se trouve sur le réseau visité par le mobile. C'est lui qui réceptionne les paquets envoyés par l'agent mère et les délivre au mobile. Cette solution a l'avantage d'éviter d'allouer une nouvelle adresse IP au mobile.
En effet, celui-ci utilise toujours l'agent relais situé dans le réseau visité comme nœud intermédiaire.
2. Utiliser un mécanisme d'auto configuration d'adresses. Cette solution permet à un mobile d'acquérir une adresse temporaire dans le nouveau réseau d'attachement. Dans ce cas, l'agent mère ne relaie plus les paquets vers un agent relais, mais envoie directement les paquets vers l'adresse temporaire du mobile. Cette alternative requière toutefois la réservation d'un pool d'adresses pour la gestion des mobiles dans un réseau.
Quelle que soit la solution utilisée, l'agent mère doit encapsuler les paquets interceptés pour les rediriger vers le nouveau réseau visité. Il y a donc création d'un tunnel entre l'agent mère et le relais/mobile.

1.2. Mobile IPv6

Mobile IPv6(16) est basé sur les mêmes principes de base que IPv4, et il comporte un nombre d'améliorations supplémentaires.
Également, puisqu'il n'y a que peu de systèmes qui utilisent le protocole IPv6, le protocole Mobile IPv6 bénéficie d'un avantage important, car il ne vise pas la compatibilité avec les machines existantes. Cet avantage majeur est utile surtout pour optimiser le routage ; rappelons que la difficulté rencontrée dans l'extension Mobile IPv4 était justement l'incompatibilité avec les hôtes correspondants n'implémentant pas les mécanismes en cause. Un autre élément important est que la protection de mises à jour envoyées aux hôtes correspondantes par l'hôte mobile ne demande ni l'établissement d'une association de sécurité auparavant, ni l'existence d'une infrastructure d'authentification. À la place, et pour s'assurer

que la vraie hôte mobile envoie le message de mise à jour une méthode appelée *return routability* est utilisée (17)

Avec le mécanisme standard d'auto-configuration d'IPv6, L'hôte mobile peut acquérir son adresse temporaire dans le domaine visité (18)

Une différence très importante qui caractérise IPv6 par rapport à la version IPv4, les agents visités sont supprimés de l'architecture d'IPv6 car l'espace d'adressage est très large et il n'est plus besoin qu'un agent visité représente plusieurs hôtes mobiles par une seule adresse.

En revanche, la possibilité que les agents visités coopèrent pour minimiser la perte des paquets lors d'un handoff est éliminée. À la place, dans Mobile IPv6 la machine peut avoir plusieurs adresses IPv6 par interface. Les terminaux peuvent ainsi garder l'ancienne connexion ouverte et continuer à recevoir des paquets à cette adresse même après qu'il est configuré avec une nouvelle adresse.

En utilisant l'adresse temporaire de l'hôte mobile sur le réseau visité les correspondants d'un hôte mobile peuvent envoyer les datagrammes. L'adresse fixe est incluse dans le nouvel en-tête de routage IPv6. A l'opposé, l'option de destination de type *home address*, est utilisée par l'hôte mobile pour s'identifier.

L'utilisation de ces mécanismes à la place de l'encapsulation réduit la surcharge observée dans Mobile IPv4, tout en permettant aux niveaux supérieurs 0. de ne voir que l'adresse fixe du mobile et donc continuer à fonctionner de manière transparente.

III. Protocole de mobilité au Niveau transport

Nous présentons seulement le protocole SCTP au quel en s'intéresse

1. Le protocole SCTP

1.1. Généralités

SCTP (Stream Control Transmission Protocol) est un protocole de transport proposé par l'IETF. Il a été conçu pour palier certaines limitations inhérentes à TCP lors du transport de signalisation téléphonique sur IP. SCTP apporte un service de transport fiable de messages issus des applications utilisateurs (e.g. protocoles de signalisation). Il est orienté connexion. Une connexion SCTP est appelée association. Il est possible de multiplexer plusieurs flux de messages au sein d'une même association (service de multi streaming). Ceci se traduit au niveau paquet par la possibilité de transporter plusieurs messages de signalisation dans un

même paquet SCTP. Les messages sont encapsulés dans des structures de données appelés chunks. Les chunks sont eux-mêmes encapsulés dans des paquets SCTP(19).

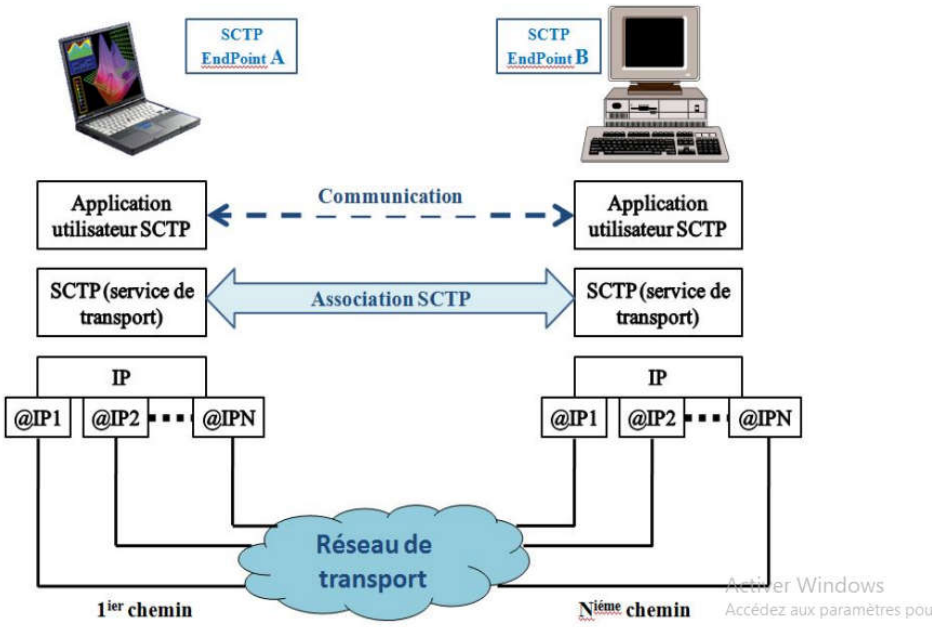

Figure 15 Schéma d'une association SCTP

SCTP permet également de mettre en œuvre le multi homing en introduisant la possibilité d'associer plusieurs adresses IP à un même port SCTP (Figure 19). Plusieurs chemins sont alors (20)disponibles pour mettre en relation deux nœuds SCTP distants. A un instant donné, seul un chemin est actif (i.e. est utilisé pour transporter les données) les autres chemins sont utilisés comme sauvegarde au cas où le chemin actif deviendrait indisponible.

Une association SCTP est établie à la demande d'une application. L'établissement de l'association se fait en deux phases impliquant l'échange de quatre messages (4-way Handshake ,cf.). Durant la première phase d'établissement de l'association, un mécanisme de sécurité est mis en place afin d'éviter les attaques de Deny of Service (ce mécanisme qui utilise des Cookie, La deuxième phase est celle qui établit effectivement l'association en réservant les ressources associées aux sockets1 de l'association sur chaque point terminal. Il est également possible au cours de cette deuxième phase d'envoyer des données utilisateurs dans les paquets d'établissement (chunk Cookie Echo et Cookie Ack).(19)Une des caractéristiques du protocole SCTP est qu'en cas de nécessité, il permet de fragmenter les

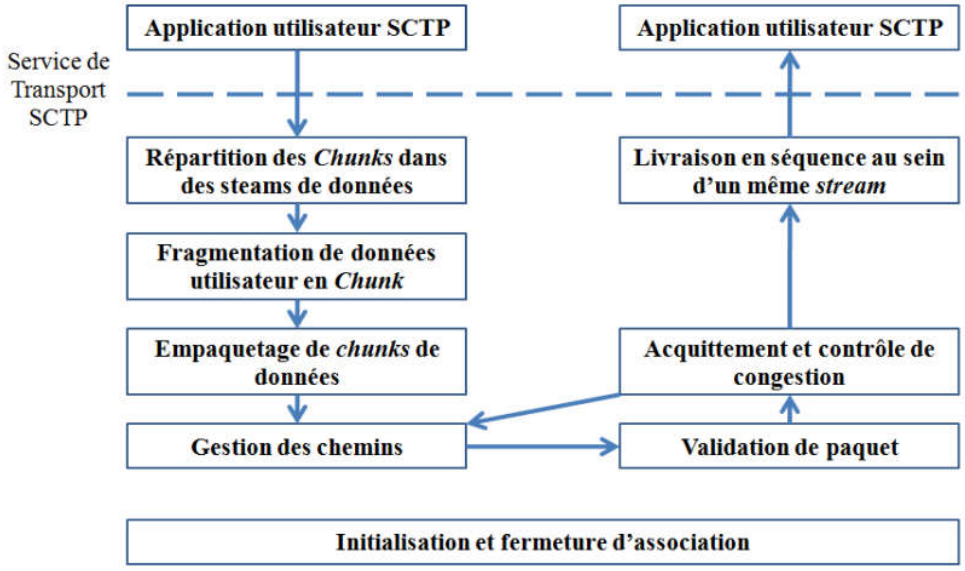

Figure 16 Fonctions du service de transport SCTP.

messages utilisateurs afin de s'assurer que les paquets SCTP soient passés aux couches inférieures conformément au PMTU (Path Maximum Transmission Unit) spécifié. En réception les fragments sont rassemblés en messages complets avant qu'ils ne soient dirigés vers le nœud SCTP de destination. Les différents fragments d'un même message portent le même SSN (Stream SequenceNumber). SCTP attribue un TSN (Transmission SequenceNumber) à chaque fragment de données utilisateur. Le protocole SCTP utilise une procédure d'acquittement sélectif. La retransmission de paquet est conditionnée par les procédures de contrôle de congestion décrites ultérieurement dans ce chapitre.(19)

1.2. Format générale d'un paquet SCTP

Un paquet SCTP (Fig 17), constituant la charge utile d'un paquet IP, est constitué d'un entête commun suivi d'un ou plusieurs chunks. Les chunks étant des blocs de données de taille variable qui peuvent appartenir à des streams différents. Chaque chunk (Figure) peut contenir aussi bien des informations de contrôle (nous parlons alors de chunk de contrôle) que des données utilisateur (nous parlons alors de chunk de données (Data) (Figure). Les chunks de contrôle précèdent toujours des chunks de données dans un paquet SCTP(21).

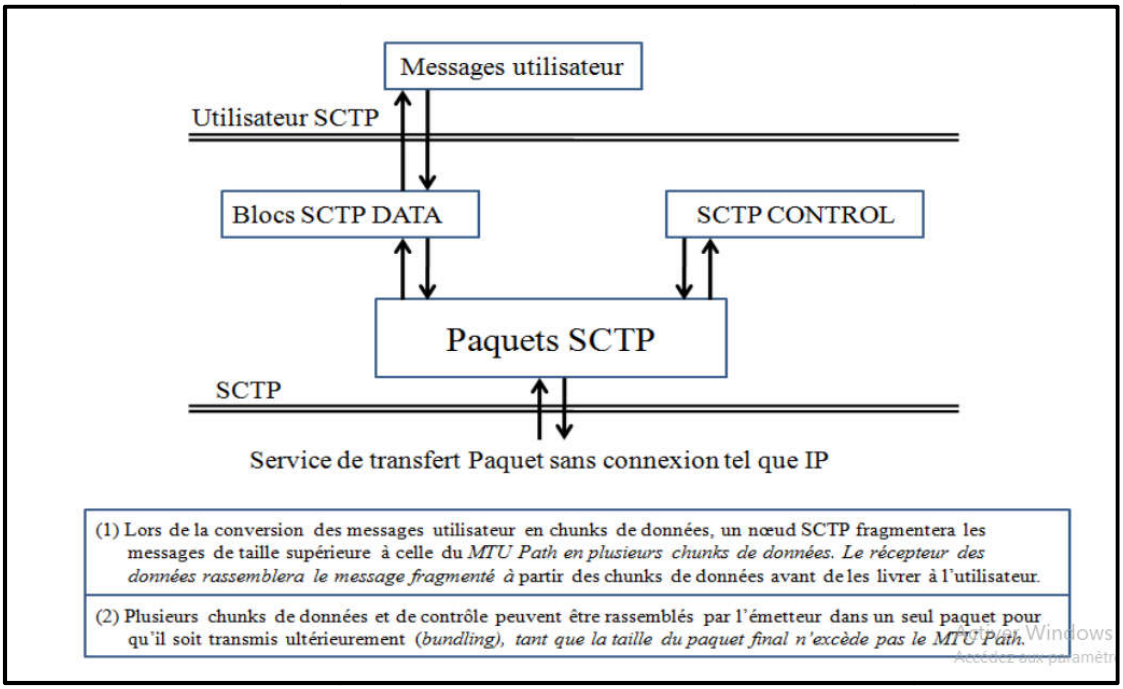

Figure 17 Transfert de données utilisateur.

Source Port Number	Destination Port Number
Vérification Tag	
Cheksum	
Chunk - 1	
. . . .	
Chunk - N	

Tableau 2 Format du paquet SCTP.

Chunk type	Chunk flags	Chenklunth
TSN		
Stream ID		SSN
Protocol ID		
User Data		

Tableau 3 Format d'un Chunk

L'entête commun d'un paquet SCTP est de taille 12 octets. Pour la détection des erreurs de transmission chaque paquet SCTP est protégé par un Checksum de taille 32 bit (Suivant l'algorithme Adler-32). Le checksum sert pour la détection des erreurs. Il est plus robuste que les 16 bit de TCP et UDP. L'entête commun contient également un champ appelé « Vérification Tag » de taille 32 bit. Ce champ identifie l'association au niveau d'un nœud SCTP (il identifie l'émetteur). Le Vérification Tag est choisi aléatoirement par chaque nœud SCTP lors de l'ouverture de l'association. L'entête commun est suivi des différents chunks de contrôle et des données.

Chaque chunk est formé également d'un entête suivi de la charge utile de longueur variable. Il s'agit des données transmises de l'émetteur vers le récepteur. L'entête de chaque chunk contient les champs suivants :

- Chunk Type (8 bits) : permet de déterminer le type de la charge utile du chunk (données ou informations de contrôle...),
- Chunk Flag (8 bits) : utilisé différemment selon le type de chunk. À ce niveau le champ Flag est plus précis c'est qu'il contient des indicateurs du type de livraison en séquence ou dans le cas de séquencement de données non requis.

Le tableau 3résume les états de fragmentation possibles pour un message utilisateur en fonction des bits du champ Flag. De plus ce champ contient des bits réservés (Reserved 5bits) qui doivent être mis à "0" et ignorés par le récepteur.

U(1bit): Unordered	B(1bit): Begining	E (1bit) : Ending	Description
1	X	X	Ce bit indique au récepteur d'ignorer le numéro de séquence du Chunk
0	1	0	Premier Chunk du message fragmenté
	0	0	Chunkintermédiaire un message fragmenté
	0	1	Dernier Chunk du message fragmenté
	1	1	Message non fragmenté

Tableau 4 Description des bits du champ Chunk flag

- ChunkLength (16 bits) : indique la taille en octets des données transmises. Ce paramètre représente la taille du chunk de données en octets incluant les champs chunk type, chunk flags, chunklength et chunk value. Si le chunk ne contient pas des informations à transmettre (i.e le champ chunk value est de longueur nulle), le champ length sera affecté à 4. Le champ chunklength ne compte pas les bits de bourrage (padding). En effet, la longueur d'un chunk de données doit être un multiple de 4 octets. Si elle ne l'est pas, l'émetteur doit remplir le chunk avec des zéros (en octets) et ce bourrage n'est pas inclut dans le champ chunklength. L'émetteur ne peut jamais remplir plus que 3 octets. Le récepteur doit ignorer les octets de bourrage.

L'entête n'est pas suivi uniquement de charge utile à transmettre, mais aussi des champs de contrôle qui varient en fonction du type de chunk. Dans ce qui suit nous abordons seulement le type 0 relatif au chunk Data.

- Le numéro de séquence TSN (Transmit Sequencenumber) sur 32 bits : numéro de séquence des fragments. Le TSN permet de reconstruire un message fragmenté.
- L'identité du flot (stream ID) (sur 16 bits) : permet d'identifier un flot (stream). Ce qui offre la possibilité d'avoir plusieurs streams dans un même paquet SCTP.
- Le numéro de message SSN (Stream SequenceNumber) dans le flot, sur 16 bits : il s'agit du numéro de séquence du chunk dans le stream. Le SSN permet le ré ordonnancement des données par le récepteur.
- Le champ Protocol_ID, codé sur 32 bits, sert à indiquer le type d'information contenue dans les chunks de données. C'est une couche non utilisée par SCTP mais par la couche application.

Enfin la charge utile est contenue dans le champ User Data. C'est un champ de taille variable, il s'agit des données appartenant au stream échangées entre deux nœuds SCTP (émetteur/récepteur).

1.3. Établissement d'une association

La procédure d'ouverture d'une association SCTP est donnée par la figure18

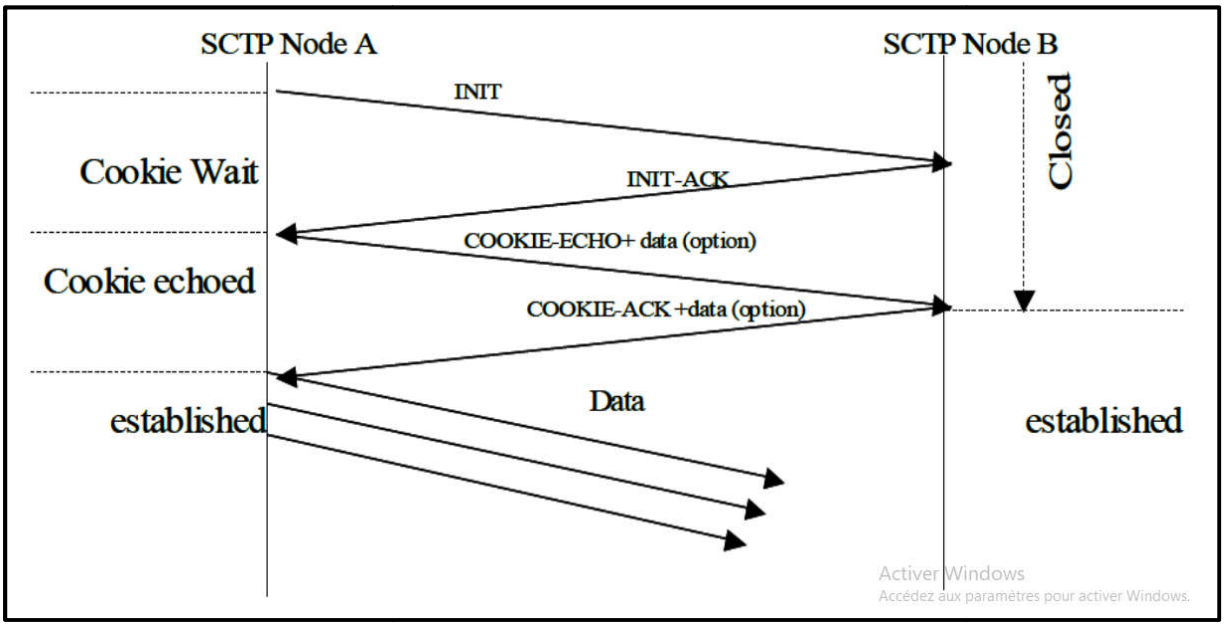

Figure 18 Phase d'initiation : échange quadruple

L'établissement d'une association s'effectue en quatre temps (alors que en 3 temps en TCP). Cela peut paraître comme inconvénient pour le temps mis pour le déclenchement, mais des données peuvent être transmises au 3ième et 4ième temps du «4-way startup handshake». De plus, ce mécanisme permet au serveur SCTP de se protéger de plusieurs types d'attaques. Dans ce qui suit nous décrivons les quatre étapes de l'établissement d'une association SCTP entre un nœud A (client) et un nœud B (serveur).(22)

Étape n°1

L'association étant à l'état CLOSED du côté du serveur (le nœud B) qui attend un message du nœud A avec un chunk INIT, dont le format est donné par la figure.

Ce chunk signale la requête qu'un client (nœud A) demande la création d'une association. Le client A envoie un chunk INIT à destination du serveur B et déclenche un temporisateur lui

permettant de réémettre ce paquet (contenant le chunk INIT) en l'absence de réponse. Ensuite le client A se met en attente du paquet de réponse contenant le cookie.(22)

Dans ce chunk, le client A copie son Vérification Tag (nous le notons par Tag-A dans la suite de ce chapitre) dans le champ Initiate Tag du chunk INIT. Le Tag-A est un nombre choisi aléatoirement par le client A entre 1 et (232-1). Après avoir émis le chunk INIT, le client A déclenché le compteur T1-INIT (Timer) et l'association, du côté client A, aura un état COOKIEWAIT.

Il est important que la valeur initiale du Tag (Initiate-Tag) soit attribuée aléatoirement afin de se protéger contre les attaques de type « man in the middle » «sequencenumber» . De même pour le champ initial TSN. Deux champs de 16 bits chacun (Number of InboundStreams et Number of OutboundStreams) servent à indiquer le nombre maximal de streams entrants et sortants que le client et le serveur peuvent supporter au niveau de l'association en cours d'établissement.

Étape n°2

Une fois le chunk INIT reçu, le serveur B génère un cookie regroupant les informations permettant d'établir la connexion avec le client A. Ensuite le serveur B crée une signature numérique du cookie appelée MAC (Message Authentication Code). Le serveur B inclut le MAC au cookie et envoie l'intégralité au client A dans un chunk INIT-ACK dont le format est donné par la table 5. L'adresse IP de destination, utilisée par le serveur B, doit être celle de l'adresse IP source du chunk INIT. L'association garde l'état CLOSED du côté serveur B.(23)

<table>
<tr><td>Type = 2</td><td>Chunk Flags</td><td>Chenklength</td></tr>
<tr><td colspan="3">Initiate Tag</td></tr>
<tr><td colspan="3">Advertised Received Window Credit (a_rwnd)</td></tr>
<tr><td colspan="2">Number Of Outbound Streams</td><td>Number OF Inbound Streams</td></tr>
<tr><td colspan="3">Initial TSN</td></tr>
<tr><td colspan="3">Optional/Variable – Length Parameters</td></tr>
</table>

Tableau 5 Format du Chunk INIT-ACK

Dans sa réponse, le serveur B doit recopier la valeur de Tag-A dans le champ Verification-Tag et fournir son propre Verification-Tag (Tag-B) au niveau du champ Initiate

Tag. Le chunk INIT-ACK contient (dans le champ State COOKIE Parameter) en plus du cookie et du MAC, une indication sur l'instant de création du cookie, sa durée de vie, ainsi que toutes les informations qui lui sont nécessaires afin d'établir l'association.

Pour générer un cookie certaines étapes doivent être considérées par le serveur B :

- Créer une association TCB (Transmission Control Block) en utilisant les informations du chunk INIT reçu et du chunk INIT-ACK à émettre, – Dans le TCB on conserve la date de création du cookie et sa durée de vie contenue dans le paramètre «valid cookie life»,
- A partir du TCB, on collecte un minimum d'informations utilisé pour recréer le TCB, et on génère un MAC qui est un hash du cookie chiffré avec la clé privée du serveur B,
- _Générer le cookie en combinant ces informations et le MAC résultant. Après l'émission du chunk INIT-ACK avec les paramètres du cookie, le serveur B doit supprimer le TCB et toute information liée à la demande d'établissement de la nouvelle association avec le client A. Toutes les données nécessaires à la connexion étant envoyées dans le cookie, aucun buffer de mémorisation n'est alloué à l'association tant que la connexion n'est pas totalement établie. Cet aspect protège le protocole SCTP de certaines attaques de type «SYNAttacks».

Étape N°3

Après la réception d'INIT-ACK du serveur B, le client A doit arrêter son compteur T1-INIT Timer et quitter l'état COOKIE-WAIT. Le client A doit alors :

a. Émettre le cookie qu'il a reçu dans un chunk COOKIE-ECHO
b. Déclencher son compteur T1-COOKIE Timer et
c. Entrer dans l'état de COOKIE-ECHOED.

Type = 10	Chunk Flags	Chunklength
Cookie		

Figure 19 Format du Chunk COOKIE-ECHO

A ce stade de la procédure d'initialisation de l'association, le serveur B vérifie la signature électronique MAC du cookie contenue dans le chunk COOKIE-ECHO venant du client A, et qui a été retournée telle quelle. Cette vérification permet au serveur B de contrôler qu'il s'agit bien du cookie qu'il l'a déjà envoyé au client A et que les données de connexion n'ont pas été modifiées. Le client A, de son côté, peut ajouter au paquet envoyé, contenant le

chunk COOKIE-ECHO, des chunks de données mais il doit respecter l'ordre des chunks dans le paquet (les chunks de contrôle sont placés les premiers devant les chunks de données). Cependant, le client A ne doit émettre aucun autre paquet vers le serveur B jusqu'à la réception de la confirmation d'établissement de l'association de la part du serveur B. (22)

Étape N°4

Suite à la réception du chunk COOKIE-ECHO, le serveur B doit répondre avec un chunk COOKIE-ACK après la construction du TCB. Le serveur B, après vérification du cookie et du MAC, signale au client A que l'association est établie par l'envoie d'un paquet contenant un chunk COOKIE-ACK et l'association, du côté serveur, passe à l'état ESTABLISHED. Un chunk COOKIEACK peut être empaqueté avec des chunks de données ou de contrôle (SACK chunks par exemple).

Le client A à la réception d'un chunk COOKIE-ACK, conclut l'ouverture de l'association SCTP et informe son ULP (Upper Layer Protocol) du succès de l'établissement de l'association avec une primitive Communication UP Notification3. C'est à partir de cet instant que le client A commencé à émettre des paquets à destination du serveur B. L'association du côté client A passe donc à l'état ESTABLISHED. (24)

Type = 11	Chunk Flags	Chunklength = 4

Figure 20 Format du Chunk COOKIE-ACK

Remarque :

- Lorsque le TCB est créé, chaque point terminal doit affecter son Cumulative TSN Ack Point interne à la valeur de son Initial TSN transmis auquel on retranche 1.
- Les paquets SCTP contenant les chunks COOKIE-ECHO et COOKIE-ACK peuvent tous comporter des chunks DATA. Ces données ne seront traitées par le serveur que si l'association est établie.
- Si un nœud SCTP recevant soit un chunk INIT, INIT-ACK ou COOKIE-ECHO décide de ne pas établir la nouvelle association suite à un manque de paramètres obligatoires dans les chunks INIT ou INIT-ACK reçus, ou à des valeurs de paramètres invalides ou à un manque de ressources locales, il doit répondre par un chunk ABORT. Il doit spécifier aussi la

cause de ce message ABORT, en incluant les paramètres causant l'erreur dans le chunk ABORT (par exemple le type des paramètres obligatoires manquants).

Type = 6	Reserved	T	ChunkLength
Zero Or more Error Causes			

Figure 21 Format du Chunk ABORT

1.4. Terminaison d'une association

Pour accomplir sa fiabilité, SCTP doit assurer une procédure de fermeture de connexion. La terminaison normale d'une association SCTP est basée sur une procédure à trois temps Cette fermeture normale de l'association est similaire à TCP à la différence que SCTP ne supporte pas l'état de « semi-connecté ». Chaque nœud SCTP attend la confirmation de la réception des chunks de données en cours de transmission avant de libérer l'association. Il existe dans SCTP une autre manière de terminer une association entre le client et le serveur, c'est une méthode plus brutale. Il s'agit d'une procédure ABORT, dont un exemple est présenté ci-dessus, qui signale uniquement qu'un des nœuds terminaux de l'association s'est retiré. Cette fermeture brutale se produit lorsqu'un arrêt immédiat est requis par les applications (situation d'exception, occurrence d'erreurs irrécupérables).(25)

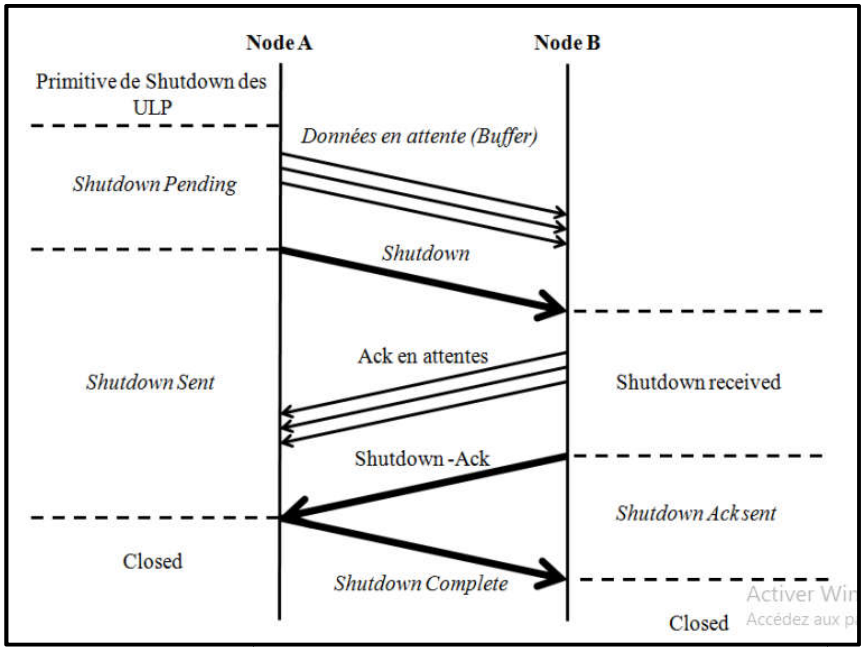

Figure 22 Terminaison d'une association.

1ère Phase :

Le client A désiré mettre fin à l'association suite à la réception d'une primitive SHUTDOWN de sa couche supérieure. À cet instant l'association du côté client A changé d'état en SHUTDOWN– PENDING. Le client A n'accepte plus de données à émettre de sa couche supérieure et assure le transfert de tous les messages qui sont stockés dans le buffer à destination du serveur B avant d'envoyer le chunkShutdown vers le serveur B. Le client A vérifié enfin que tous les acquittements des paquets SCTP précédemment envoyés ont été reçus. Cependant, le client A peut retransmettre des données qui ont été perdues et signalées par le serveur B et l'association du côté du client passe à l'état SHUTDOWN-SENT.

2ème Phase :

Ensuite, le client A doit émettre un chunk SHUTDOWN vers son nœud de destination (le serveur B). Le client A précisé dans ce chunk le dernier TSN reçu du serveur et également les paquets perdus. Lors de l'émission du SHUTDOWN le client A doit déclencher un compteur T2- shutdownTimer. Dans le cas où ce compteur expire, le message SHUTDOWN doit être réémis avec la mise à jour du dernier TSN reçu en séquence. À la réception d'un chunk SHUTDOWN l'association du côté du serveur B change d'état en SHUTDOWN-RECEIVED. À son tour le serveur B n'accepte plusde nouveaux paquets en provenance du client A et arrête de transmettre de nouvelles données de ses couches supérieures (ULP). Le serveur B fini par émettre les données stockées dans son buffer de transmission et vérifie que tous les paquets précédemment transmis ont été reçus au moyen de la valeur de TSN fournie

dans le chunk SHUTDOWN. Pour chaque paquet reçu, le client A acquitte les paquets reçus par un chunkSACK, envoie un chunk SHUTDOWN et réinitialise son temporisateur.

Type = 7	Chunk Flags	Chunklength = 8
Cumulative TSN Ack		

Chunk flags : 8 bits, mis à zéro lors de la transmission et ignoré en réception

Length : 16 bits (unsignedinteger), indique la longueur du paramètre mis à 8,

Cumulative TSN Ack : 32 bits (unsignedinteger), ce paramètre contient le TSN du dernier chunk reçu en séquence avant toute pertes (gap).

Figure 23 Format du Chunk shutdown

Une fois que le serveur B a atteint l'état SHUTDOWN-RECEIVED, il doit ignorer toute demande de shutdown de l'association provenant de son ULP. Enfin, une fois tous les paquets en attente envoyés et reçus par le client A, le serveur B émet un chunk SHUTDOWN-ACK (Fig 3.14) et déclenche son temporisateur T2-shutdown Timer. C'est à partir de là que l'association du côté serveur change d'état en SHUTDOWN-ACK-SENT. Si son temporisateur expire, le serveur B doit réémettre le chunk SHUTDOWN-ACK.

Type = 8	Chunk Flags	Chunklength = 4

Figure 24 Format du Chunk Shutdown-Ack

3ème Phase :

Suite à la réception d'un SHUTDOWN-ACK, le client A, s'assure qu'il n'a pas eu de perte de données de côtés des deux intervenants dans l'association, arrête son temporisateur T2-shutdown Timer et envoie un chunk SHUTDOWN-COMPLETE (Fig 3.15) au serveur B. Le client A peut alors quitter l'association et supprimer toute information liée à la connexion et l'état de l'association passe définitivement à CLOSED. De son côté le serveur B est garanti de la bonne réception des données au cours de l'association courante en recevant un chunk

SHUTDOWN-COMPLETE. À cet instant, le serveur B vérifiequ'il est dans l'état SHUTDOWN-ACK-SENT, si ce n'est pas le cas le chunk sera ignoré. S'il est bien dans l'état SHUTDOWN-ACK-SENT, le serveur B arrête son T2-shutdown Timer et supprime l'association (et ainsi l'association entre dans l'état CLOSED).

Reserved : 7 bits (mis à 0 lors de transmission et ignoré en réception)

T : 1bit, T=0 si l'émetteur a un TCB qui a été détruit sinon T=1.

Type = 14	Reserved	T	Length = 8

Figure 25 Format du Chunk SHUTDOWN-COMPLETE

1.5. Transfert des données utilisateurs (gestion des acquittements)

La transmission de données en SCTP doit se produire uniquement lors des états, décrits précédemment, suivants :

– ESTABLISHED

– SHUTDOWN-PENDING

– SHUTDOWN-RECEIVED

L'exception de ceci, est lorsque les chunks de données sont autorisés d'être transmis avec un chunk de contrôle de type COOKIE-ECHO quand l'association est dans l'état COOKIE-WAIT. Un récepteur SCTP doit être capable de recevoir au minimum un paquet SCTP de taille 1500 octets, c'est à dire qu'un nœud SCTP ne doit pas indiquer moins de 1500 octets dans son envoie initial (arwnd = 1500 byte) de INIT ou INIT-ACK. Pour assurer l'efficacité de transmission, SCTP définit des mécanismes pour l'empaquetage des messages utilisateurs de petites tailles et la fragmentation de ceux de tailles importantes.(22)

Le protocole SCTP, possède des mécanismes de contrôle de flux et de congestion. Il permet, grâce à sa caractéristique de Multistreaming, l'envoie de données correspondant à plusieurs flots (streams) dans un même paquet grâce à un mécanisme d'identification de stream dans chaque chunk. Lors de transfert de données, si un message utilisateur est de taille supérieure au PMTU (Path Maximum Transmission Unit) d'une association, il sera fragmenté en différents chunks. Les chunks formant un même message ont des TSNs successifs, et sont

identifiés par le même SSN. L'identification de l'ordre des différents morceaux d'un message utilisateur est assurée au moyen des flags B/E (premier chunk, les chunks intermédiaires, dernier chunk). Pour la transmission des données, SCTP assure un acquittement sélectif au moyen du SACK chunk. Ce dernier est utilisé par le récepteur SCTP qui l'envoie en retour afin d'acquitter les DATA chunks reçus. Le SACK chunk permet à l'émetteur d'identifier la fenêtre avertie de réception, les chunks qui ont été perdus, les chunks qui ont été reçus dupliqués, etc. Ainsi toutes les informations nécessaires à d'éventuelles retransmissions ou à la gestion de streams.(23)

Lorsque un chunk de données (DATA chunk) atteint un récepteur SCTP, ce dernier doit envoyer un SACK afin d'acquitter sa réception. Un champ Cumulative TSN Ack est utilisé pour acquitter la réception de tous les chunks reçus en séquence sans erreurs. Ce champ indique jusqu'à quelle valeur de TSN des données ont été correctement reçues. Un SACK contient aussi les Gap Ackchunks qui sont utilisés pour acquitter des plages de chunks de données reçues séparément. Le Gap Ackchunk Start et Gap Ackchunk End sont codés sur 16 bits. Ils indiquent la position relative (par rapport au Cumulative TSN Ack) des différentes plages de chunks isolés correctement reçus. Le SACK comporte également une indication sur les TSNs dupliqués, donnée par les champs Number of duplicate TSNs et Duplicate TSN. Le champ Number of duplicate TSNs, codé sur 16 bits, indique le nombre de chunks dupliqués. Le champ Duplicate TSN (sur 32 bits) donne le nombre de fois qu'un TSN a été reçu de façon dupliquée à partir du dernier SACK transmis. Un TSN dupliqué apparaît dans le SACK autant de fois que le récepteur l'a reçu de façon redondante.(26)

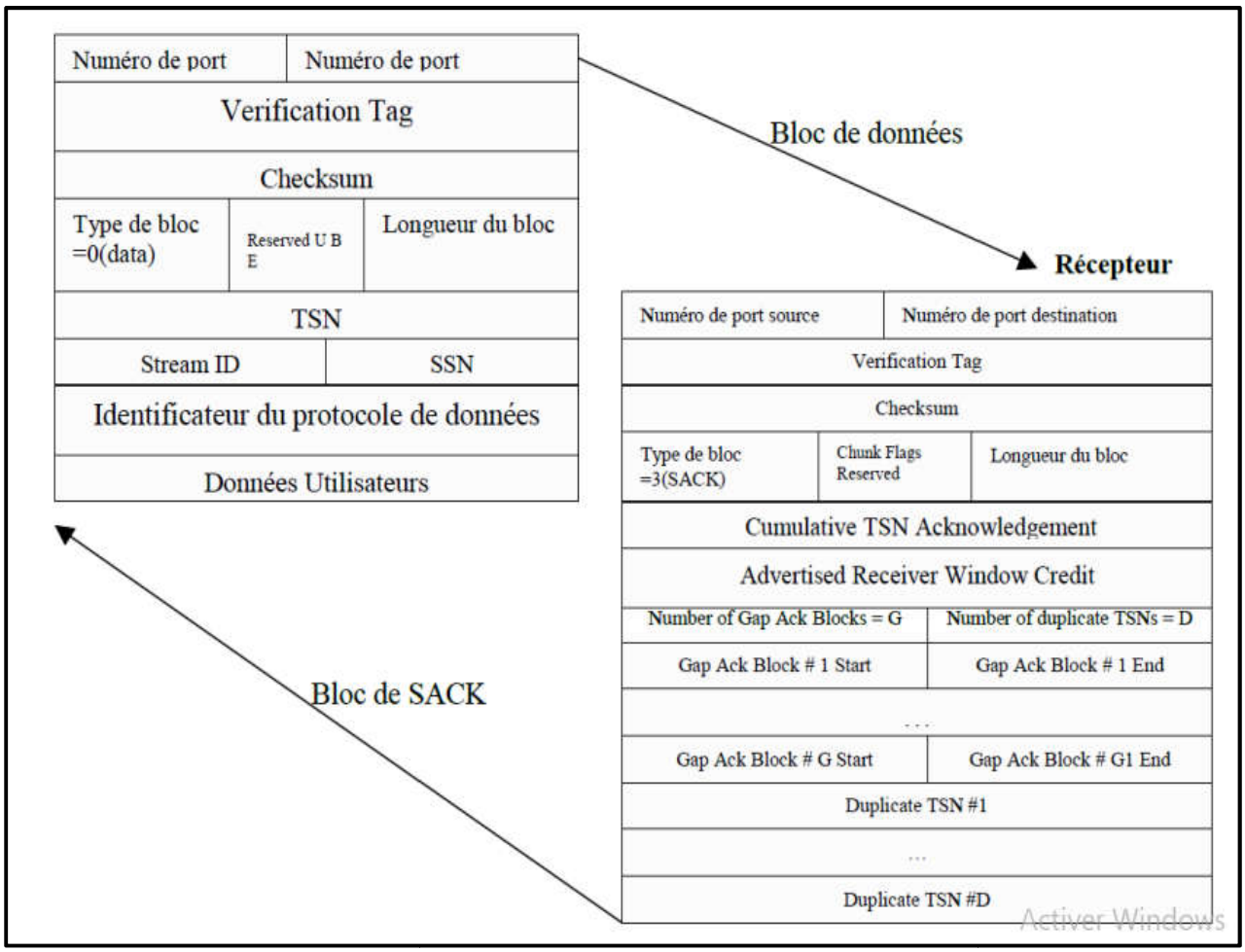

Figure 26 Transmission de données.

L'exploitation des données transmises par le chunk SACK, permet à l'émetteur SCTP de réémettre les DATA chunks qui n'ont pas été reçus et qui sont identifiés par leur numéro TSN. Cette retransmission s'effectue après l'expiration du temporisateur ou la réception de 4 chunks SACK indiquant la perte du même DATA chunk. Nous constatons que SCTP acquitte les paquets reçus non pas les octets comme c'est le cas de TCP. S'il y a eu un problème lors de la transmission des données, et qu'une retransmission de ces dernières a eu lieu, les DATA chunks hors séquence sont acquittés à l'aide du champ Gap Ack Blocks, qui permet l'acquittement d'une ou plusieurs séquences de TSN. Notons que comparativement à TCP, SCTP permet d'avoir de plus de GAP Blocks dans un chunk SACK. Il offre un espace réservé de taille 216 octets comme le spécifie le champ chunklength.

V. Notion de la qualité de service dans les réseaux ad hoc

1. Définition de la qualité de service :

L'association française de normalisation (afnor) **définie la qualité de service** comme étant «la capacité d'un service à répondre par ses caractéristiques aux différents besoins de ses utilisateurs ou consommateurs».

Et d'après Jean-Pierre BARUCHE:

« La **qualité de service** c'est la prestation, la mise à disposition, l'accompagnement, le conseil, le service après-vente associé au service principal, en quelque sorte la dimension humaine qui se trouve à chaque instant du service *».*

De ces deux définitions, nous avons retenu que la qualité de service est :

L'aptitude du service à satisfaire les besoins des utilisateurs ;

Associée aux éléments physique du service ;

Associée aussi aux résultats de la **transaction entre le client et le personnel en contact.**

2. Les avantages de la qualité de service :

Les avantages sont :

- Une meilleure qualité diminue les coûts :

Avant, la qualité était contrôlée par détection, c'est-à-dire que les défauts étaient recherchés après la fabrication des produits, ce qui impliquait des coûts additionnels dus principalement aux frais d'inspection, re-travail, perte en matière première….etc.

Aujourd'hui, l'accent est mis sur la prévention au cours de la conception et de la fabrication, afin de ne pas produire d'articles défectueux.

Ceci permet de réduire les coûts puisque les défauts et les dysfonctionnements sont évités.

- Une meilleure qualité entraine une augmentation de la productivité :

La productivité veut dire « produire mieux » et pas nécessairement « produire plus ».

En effet, en améliorant la qualité des produits et services offerts, les entreprises arrivent à baisser les coûts du fait de la diminution des rebuts et des déchets.

Et donc les produits sont bien fabriqués du premier coup, il n'y a plus besoin de les refaire, ce qui au final va augmenter la **productivité de l'entreprise.**

3. L'amélioration la qualité de service

- L'amélioration de la qualité ne concerne pas uniquement la production :

Plusieurs études menées sur le succès de différentes entreprises, ont indiqué que l'amélioration de la qualité doit impérativement s'étendre aux autres fonctions de l'entreprise, comme par exemple :

- ✓ La vente,
- ✓ Le marketing,
- ✓ Les finances et
- ✓ L'administration

Afin d'assurer une continuité dans le progrès et l'amélioration continue par produits et services.

- L'amélioration de la qualité ne nécessite pas de gros investissements :

Contrairement à ce que beaucoup d'entreprises pensent, l'amélioration de la qualité ne demande pas un investissement lourd, mais seulement un engagement profond de la part de la direction vers les envers de la qualité.

La **qualité des produits et services** peut être améliorée de façon conséquente en sensibilisant le personnel au respect des exigences du client, à la normalisation des procédés, à la formation

des agents d'exécution et à l'observation d'une discipline technique relative aux produits et services.

4. Les difficultés de la qualité de service :

- L'aspect immatériel des services :

Cet aspect immatériel rend difficile voire même impossible de les tester avant leur consommation, il rend tout aussi difficile de fixer les standards de production précis relatifs à un niveau de qualité homogène (un même service peut être offert différemment d'un client à un autre; en terme de délai par exemple).

- La relativité :

Deux clients obtenant exactement le même service pourraient avoir des perceptions totalement différentes de la qualité.

De plus, une même personne, obtenant un même service dans deux situations différentes peut ne pas éprouver la même satisfaction.

Par conséquent, le service sera chaque fois jugé de qualité ou non par le client après consommation. D'où la difficulté en matière de garantie et de **contrôle de la qualité de service**.

La capacité de maintenir un niveau de qualité constant et régulier auprès des différents clients est nécessaire au niveau de tous les réseaux.

C'est sur les variables que doivent être fournis les efforts de ceux qui conçoivent le service et aussi de ceux qui le mettent à la disposition du client jour après jour.

- La simultanéité entre production et consommation :

Le service a un caractère instantané car il est produit et consommé au même moment et au même endroit.

La **qualité d'un service** est d'autant plus difficile à gérer quand la participation d'un client est active, ce dernier juge autant le processus et la relation que le résultat de la prestation.

La qualité perçue du service reçu dépend en réalité de la qualité du **contrat entre le client et l'ensemble** de l'organisme prestataire (supports matériels, personnel en contact…..).

5. Les 10 critères déterminant la qualité de service :

Les utilisateurs apprécient la qualité des services sur les critères suivants qui sont évidemment variables en fonction du service proposé :

- ✓ **Tangibilité du service :**

Apparence physique des locaux, des équipements, du personnel et des documents.

- ✓ **Fiabilité :**

Capacité à réaliser le service promis de manière sûre et précise.

- ✓ **Rapidité (réactivité) :**

Volonté d'aider le client en lui fournissant un service rapide et adapté.

- ✓ **Compétence :**

L'organisation du service dispose des connaissances, des moyens, de savoir-faire et des capacités requises pour fournir le service.

Il s'agit ici du professionnalisme de l'organisation et du personnel en contact.

- ✓ **Courtoisie :**

Politesse, respect et personnel en contact amical.

- ✓ **Crédibilité et honnêteté de l'entreprise de service :**

Cette caractéristique concerne la notoriété de l'organisation, sa réputation, sa garantie de sérieux et son honnêteté.

- ✓ **Sécurité :**

Absence de danger, de doute, de risque, qu'il s'agisse d'un risque physique, financier ou moral.

- ✓ **Accessibilité :**

Le service doit être facilement accessible aux clients.

Il s'agit là d'une accessibilité physique et psychologique.

- ✓ **Communication :**

L'organisation veille à tenir les clients informés du contenu précis de l'offre de service et cela dans un langage compréhensible et adapté à chaque type de clients.

- ✓ **Compréhension du client :**

Les efforts déployés par l'entreprise pour connaître les besoins spécifiques des clients et pour s'y adapter le mieux possible.

6. Les principaux modèles de QOS utilisés pour les réseaux ad hoc

6.1. IntServ

Le premier modèle s'appelle " Services Intégrés " ou IntServ. Il utilise un concept basé sur la réservation ou chaque application transmet ses conditions à tous les nœuds de réseau traversés jusqu'au nœud de destination. Quand tous les nœuds ont acceptés les conditions, l'application commence l'acheminement de ses données. IntServ aborde la qualité de service en reprenant le concept du " meilleur effort " et en y rajoutant le support du trafic en temps réel. Le modèle IntServ est donc un modèle incluant le concept du " best effort ", un service en temps réel, et un partage de lien contrôlé. IntServ a plusieurs composants: la signalisation, le contrôle d'admission, la classification des flux, et l'ordonnancement des paquets.

- *La signalisation :* permettant la réservation de ressources qui est assurée par le protocole RSVP (*Resource ReSerVation Protocol*).

- *Le contrôle d'admission :* son rôle est de bloquer les flux dont les ressources demandées ne sont pas disponibles. Ce contrôle est opéré par chaque routeur sur le chemin. Chacun d'entre eux va accepter ou rejeter la demande de service suivant l'état actuel du réseau. Les routeurs indiquent à l'application, via RSVP, si le besoin de QoS peut être satisfait ou non.

- *La classification des flux :* elle est aussi effectuée par chaque routeur. Cette phase complexe permet de séparer les flux et d'insérer les paquets entrant dans les files d'attente appropriées.

- *L'Ordonnanceur :* il gère l'ordre de sortie des paquets des files d'attente afin d'assurer la QoS demandée.

6.2. DiffServ

Le deuxième modèle, " Services Différenciés " ou DiffServ, utilise une technique de marquage des paquets chaque paquet est identifié par un code dans son entête IP pour indiquer à quelle classe de trafic il appartient.

Les commutateurs traversés sur le chemin traitent donc les paquets différemment en fonction

de la classe de service à laquelle ils appartiennent.
Le comportement de chaque nœud du réseau est en effet choisi en se basant sur la classe de chaque paquet.
Diffserv emploie le champ ToS "Type Of Service" dans l'en-tête d'IP pour déterminer à quelle classe appartient chaque paquet. L'émetteur de flux spécialisé, spécifie une classe de service qu'il souhaite donner à ses paquets, au travers du champ ToS. Au cours de son parcours dans le réseau, ce paquet traverse des routeurs qui sont dotés d'algorithmes (Packet Classifier), qui lisent le champ TOS.
DiffServ est destiné à combler les défauts d'IntServ. Il cherche à supprimer le problème de passage à l'échelle en définissant des classes permettant d'agréger plusieurs flux. Ainsi, tous les flux appartenant à une même classe reçoivent le même service. Deux types de routeurs sont donc définis:

1. **Les routeurs de bord** (*Edge Routers*) : chargés de la classification, du marquage et du maintien de l'état des flux.
2. **Les routeurs de cœur** (*Core Routers*) : chargés uniquement de l'acheminement des paquets selon le marquage.

Conclusion

L'objectif principal de ce chapitre était de présenter les réseaux véhiculaires comme un nouveau paradigme de réseau et la notion de qualité de service et leurs différents modèles utilisés qui a été introduite afin de satisfaire les besoins des utilisateurs beaucoup de solutions sont aujourd'hui proposées pour fournir de la qualité de service aux réseaux mobiles ad hoc, mais beaucoup reste encore à faire. Ainsi que quelques protocoles de mobilité, parmi ces protocoles le protocole SCTP que nous voulons lui appliquer le mode de gestion de files d'attentes PriorityQueuing afin de pouvoir l'utiliser pour la dissémination des messages d'urgences.

Chapitre 03: Contribution

Introduction

La partie 1 a présenté l'état actuel des réseaux sans fil, en particulier les caractéristiques et les applications des réseaux ad hoc et spécifiquement les réseaux VANET.

La qualité de service ou quality of service (QoS) est un concept important dans les réseaux et les télécommunications. En effet, pour que des applications multimédia émergent sur un même réseau, des politiques de QoS doivent être établies pour gérer ces différents flux.

Après avoir défini le contexte de notre travail et quelques définitions de base. Dans cette partie du chapitre, nous présenterons notre contribution pour la dissémination des messages d'urgences par le biais des mécanismes de QOS.

Contexte du travail

Dans notre contribution pour la dissémination des messages d'urgences par le biais des mécanismes de QOS intégrés au niveau du protocole SCTP, nous avons (15):

1. Utiliser 4 files pour classifier les différents flux:

- Flux message d'urgence
- Flux donnés
- Flux audio
- Flux vidéo

2. Classifier les paquets en entrée pour les aiguiller dans leurs files selon le type de flux, Laclassification est basée sur les paramètres :

- Interface Source.
- Access list IP (standard and extended).
- Packet size (greater or smaller than specified).
- Fragments.
- Numéros des ports SCTP (source ou destination).

3. En tête de chaque file d'attente, l'ordonnanceurvérifie la présence des paquets à l'intérieur de chaque file d'attente et hiérarchise les paquets selon leurs priorités. Ainsi, la classe avec la priorité la plus élevée peut obtenir toute la bande passante disponible, tandis

que les autres classes ne peuvent obtenir la bande passante restante qu'après que les autres classes l'ont consommée.

I. Les besoins en services de transports nécessaires à une application

3 types de besoins au niveau des applications, en termes de :

- Perte de données
- Bande passante
- Délai

1. Besoin en termes de Perte de données

Certaines applications nécessitent une fiabilité à 100% (15):

- Courrier électronique (SMTP)
- Transfert de fichiers (FTP)
- Accès distant (Telnet)
- Transfert de documents Web (HTTP)
- Applications financières

D'autres peuvent tolérer des pertes (loss-tolerant applications) :

- Applications multimédia : audio/vidéo

2. Besoinen termes de Bande passante

Certaines applications (ex : multimédia) requièrent une bande passante minimale :

- Téléphonie sur Internet : si la voix est codée à 32 Kbps, les données doivent être transmises àce débit.
- Applications multimédia

D'autres utilisent la bande passante disponible (applications élastiques) :

- Courrier électronique, transfert de fichiers, accès distant, Web

3. Besoinen terme de Délai

Certaines applications nécessitent un délai de bout-en-bout faible (moins de quelques centaines de ms) :

- Applications temps réel interactives : Téléphonie sur Internet, Environnements virtuels, Téléconférence, Jeux en réseau ....

Pour les applications non temps réel, un délai court est préférable, mais pas de contrainte forte.

Le tableau ci-dessus présente les Besoin en service de transport(15) :

Application	**Pertes**	**Bande passante**	**Sensibilité temps**
Transfert de fichiers	Sans pertes	Elastique	Non
e-mail	Sans pertes	Elastique	Non
Web	Tolérant	Elastique	Non
Audio/vidéo Temps réel	Tolérant	Audio: 5Kb - 1Mb vidéo:10Kb - 5Mb	Oui, centaines ms
Audio/vidéo enregistré	Tolérant	Audio: 5Kb - 1Mb vidéo:10Kb - 5Mb	Oui, quelques secondes
Jeux interactifs	Tolérant	Quelques Kbps	Oui, centaines ms
Applications financières	Sans pertes	élastique	Oui et Non

Tableau 6 Les Besoin en service de transport

Algorithme de gestion des files d'attentes:

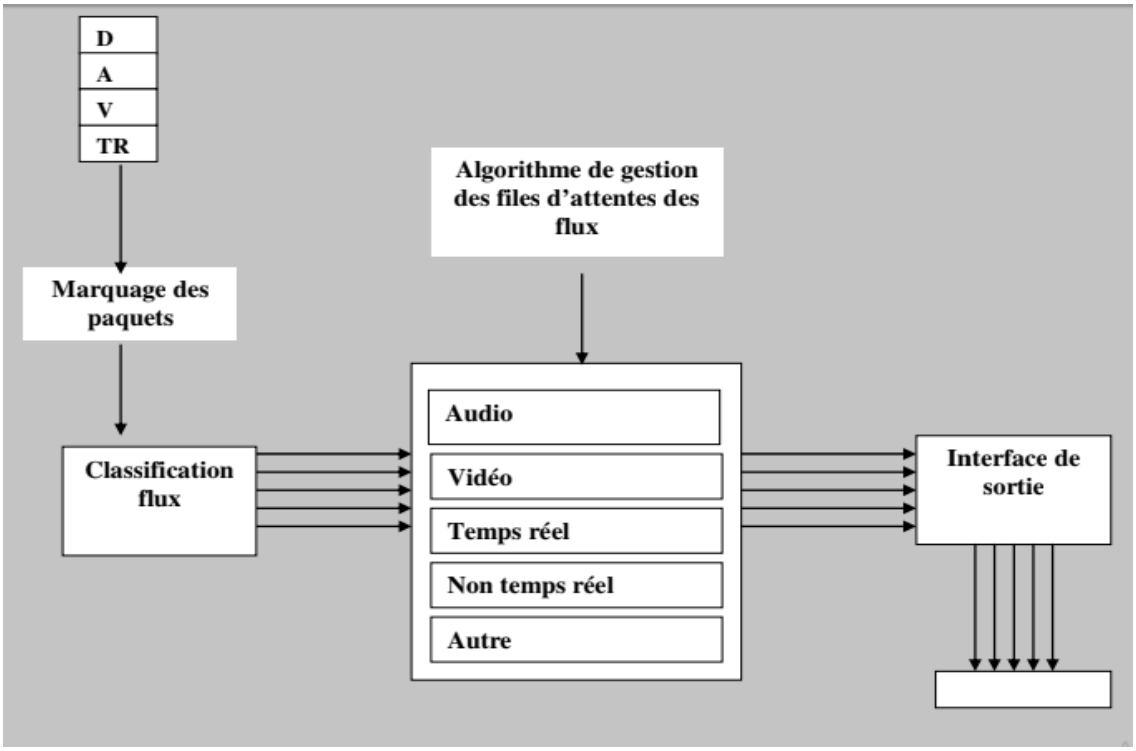

Figure 27 Processus de gestion des files d'attentes

L'algorithme exécuté pour servir les différentes files est le suivant :

- Tant qu'il y a des paquets dans la file "high" alors router le paquet, sinon passer à la file inférieure,
- Tant qu'il y a des paquets dans la file "medium" alors router le paquet, sinon passer à la file inférieure,
- Tant qu'il y a des paquets dans la file "normal" alors router le paquet, sinon passer à la file inférieure, router le paquet dans la file "low", puis réitérer.

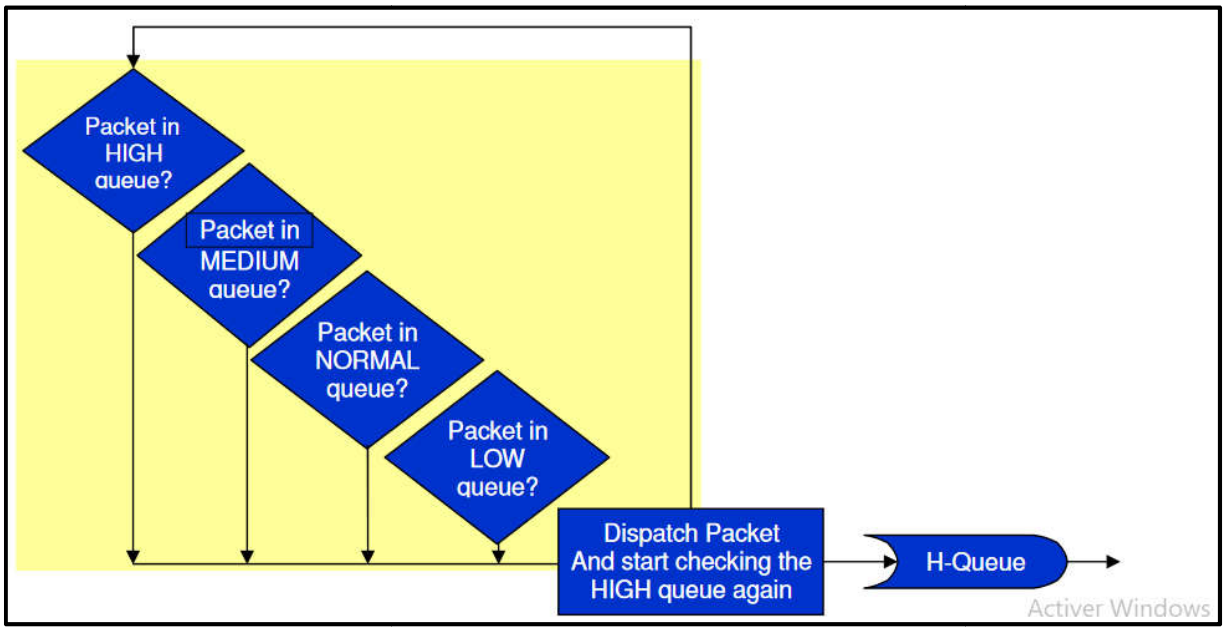

Figure 28 Gestion de files en mode PQ

Avec le Priority Queuing on commence à avoir des possibilités de qualité de service intéressantes, outre les avantages suivant:

- Simplicité, rapidité et faible cout au niveau de l'OS du routeur.
- Les priorités absolues permettent de privilégier de manière absolue un trafic par rapport à un autre.

II. Les inconvénients du mode de gestion Priority Queuing

- L’ordonnancement statique (classifier les paquets en entrée) n'est pas toujours facile à faire (quels traficsdans quelles files ?)(15).
- Les priorités absolues et l'algorithme décrit ci-dessus peuvent causer des congestions artificielles dues àdes problèmes de "famine" (Trop de paquets à ordonnancer dans la file "high")(15).
- Il n'y a seulement que 4 niveaux de priorité, comment faire alors pour classifier plus de 4 classes de trafic(15) ?

Le problème de famine cité ci-dessus se pose lorsque l'ordonnancement statique effectué par l'administrateur va aiguiller trop de paquets dans la file "high" (la priorité la plus grande) par rapport aux autres files. Dansce cas, le routeur ne routera que les paquets de la file "high" en délaissant ceux des autres files, d'où la"famine" pour les autres files. Il faudra donc s'assurer de ne pas "faire trop passer" de paquets dans la file"high"(15).

Conclusion

Le choix d'un algorithme de planification approprié est essentiel pour garantir la qualité de service des applications en temps réel tel que les messages d'urgences. Les réseaux à commutation de paquets ont déployé une variété d'algorithmes statiques et dynamiques, ou une combinaison des deux, pour répondre aux divers besoins des applications en temps réel.

Dans un réseau à commutation de paquets, la manière classique d'accorder des privilèges de service d'application sous des contraintes de temps consiste à donner au service d'application une priorité plus élevée et à planifier les threads en concurrence pour l'accès au média en fonction de la priorité. Il s'agit d'un ordonnancement à priorité fixe. Accès urgent au support de transmission par ordre de priorité. Il s'agit d'un ordonnancement à priorité fixe. Bien que cette technique soit toujours efficace pour améliorer la qualité de service temporelle des flux prioritaires, elle peut être coûteuse pour les flux de priorité inférieure, mais elle n'est pas compatible avec une utilisation réelle en raison de la famine induite par l'arrivée de paquets heuristiques.

Nous avons présenté le mode de gestion de files d'attentes Priority Queuing qui peut être utilisé pour gérer les différents flux du protocoles SCTP en raison de son usage est que certaines données sont importants que d'autres, la file d'attente prioritaire s'assure que les objets les plus importants sont récupérés en premier, toutefois cela peut signifier que les objets de faible priorité languissent dans la file d'attente ne jamais être enlevé.

Chapitre 04: Simulation Et Résultats

Introduction :

La simulation permet de tester de nouveaux protocoles à faible coût et de prévoir les défis futurs afin que vous puissiez mettre en œuvre la technologie qui répond le mieux à vos besoins.

OMNeT++ [http://www.omnetpp.org/] est une plateforme de simulation, modulaire, open source, orienté objet et à événements discrets écrit en C++. Elle offre un IDE basé sur Eclipse, un environnement d'exécution graphique, et une foule d'autres outils.

La simulation doit d'aborder les différents concepts, son environnement de travail, ainsi que les outils aidant à bâtir un modèle de simulation et de l'exploiter.

Dans ce présent chapitre nous allons essayer de présenter le simulateur OMNeT++ que nous allons utiliser pour évaluer les performances de notre proposition. Nous allons, tout d'abord, éclaircir certaines fonctionnalités du simulateur OMNeT++. La simulation proprement dite. Celle-ci comprend plusieurs phases :implémentation, scénario de simulation, et autres mesures tel que la charge et la mobilité.

I. Présentation du simulateur OMNeT++ (Objective Modular Network Testbed in C++) :

OMNeT++ modélise des schémas basés sur des événements discrets (Klaus, Mesut et James, 2010) où les opérations au sein du système sont comparées à un Événements maintenus par l'émulateur à l'aide d'une file d'attente et classés en fonction de leur temps d'exécution. Chaque événement s'exécute à un moment précis, ce qui entraîne Un changement dans l'état du système. Puis simulez l'événement en exécutant file d'attente et profiter de la stabilité du système entre deux événements consécutifs pour que Possibilité de passer directement d'un événement à un autre(34).

Il est également important de noter que OMNeT++ est un logiciel libre, il dispose d'une licenceOuverte à des fins académiques. Sa principale responsabilité est de créer un système fiable etRivaliser avec des simulateurs spécialisés et axés sur la recherche tels que NS2 ainsi que des émulateursOPNET commercial dont la licence est très coûteuse.

OMNeT++ ne fournit pas de modèles de simulation prêts à l'emploi. Cependant, il fournit uneEnvironnement puissant alimenté par tous les outils nécessaires, y compris les cœurs de simulation,bibliothèques, éditeurs graphiques et outils d'analyse pour créer cesComposants de simulation.

Les modèles de simulation sont ensuite développés par plusieurs individus et groupes de Recherche, indépendante d'OMNeT++, chacune suit son propre cycledévelopper. Ils sont généralement regroupés dans des packages appeléscadre. Parmi eux, on peut citer principalement le Framework INET, qui couvre un large éventail deMettre en œuvre divers protocoles tels que le protocole Internet (TCP,UDP, IPv4, IPv6, ...), protocoles de couche liaison (Ethernet, PPP, IEEE 802.11, ...) etIl existe également des Framework Castalia et MiXiM pour les réseaux mobiles et fixes sansdocument(34).

Ces cadres évoluent constamment et gagnent de plus en plus en fiabilité etrichesse, ce qui fait d'OMNeT++ dans le temps un certain nombre de groupes de recherche universitaires, d'institutions de recherche à but non lucratif, Travaillant également avec des sociétés telles qu'IBM, Intel, Cisco, Thales et Broad om (Vargaet Honig, 2008)

1. Structure d’un modèle OMNeT++ :

Une façon de construire un modèle OMNeT++ consiste à imbriquer un ensemble de modules. Utilisant des messages pour communiquer les uns avec les autres, le système a une structure hiérarchique selon les découvertes de Varga et Hornig en 2008.

Contenant des sous-modules pouvant également contenir des sous-modules supplémentaires, le niveau le plus élevé de la hiérarchie est représenté par le module système. Il existe de nombreux niveaux dans ce système, chacun avec différents sous-modules.Est une hiérarchie illimitée, ne l'oubliez pas.Appelés modules composés, modules qui contiennent des sous-modules(34).

Les modules simples constituent le niveau le plus bas de la hiérarchie.Ses algorithmes sont contenus dans les composants actifs, qui constituent le modèle.En utilisant le langage C++, les utilisateurs mettent en place des implémentations.Les instances du type de module incluent des modules simples et composés, chacun avec ses propres caractéristiques distinctives. Ces modules sont définis à travers une variété de critères.Pour créer des composants pour d'autres modèles, l'utilisateur doit fournir une description du modèle(34).

L'obtention du module système implique de traiter des types de modules de plus en plus complexes. Enfin, ce module est atteint et il représente l'aboutissement de ce processus.

Le module qui forme le réseau comprend tous les sous-modules dans une disposition sporadique.

Les modules simples forment ce que l'on appelle le niveau inférieur de la hiérarchie.

Les algorithmes qu'ils contiennent constituent les parties fonctionnelles du modèle.

En utilisant le langage C++, les utilisateurs peuvent implémenter certaines fonctionnalités.

Le type de module se compose de modules simples et composés, tous deux définis par un ensemble de lignes directrices.En tant que blocs de construction pour d'autres entités, la description du modèle sert l'utilisateur. Pour obtenir le module système, nous explorons d'abord des types de modules plus complexes. le module réseau, sans connexions, comprend l'ensemble de ses sous-modules.

(Voir figure 30.) à l'extérieur, en plein air.

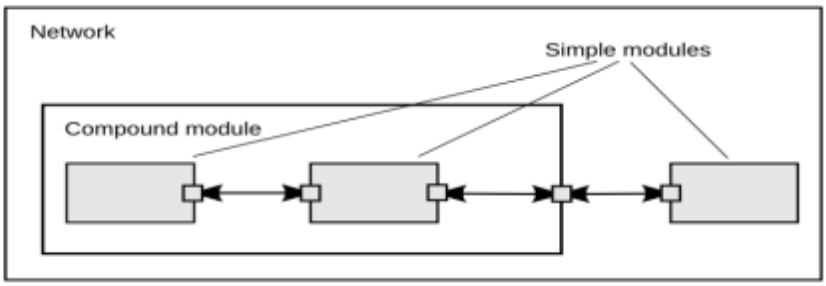

Figure 29 Hiérarchie d'un modéle OMNeT++ Tirée de Varga et Hornng(2008)

Il n'y a pas de différence lorsque le type de module est utilisé comme bloc de construction

Entre modules simples ou composés. Cela permet aux utilisateurs de diviser de manière transparente un module en plusieurs modules plus simples pour obtenir des modules composites ou au moins au lieu de cela, réimplémentez la fonctionnalité du module composite pour obtenir un seulModules simples sans affecter les utilisateurs du type de module.

Dans le modèle OMNeT++, les modules communiquent à l'aide de messages, sauf l'attribut d'horodatage peut contenir plusieurs autres champs d'informations, tels qu'adressent source, adresse de destination, etc.

Les modules simples envoient généralement ces messages via des ports (enOMNeT++ sous le nom gate), mais aussi directement aux modules objets etPar exemple, en cas de connexions temporaires.

Ces ports représentent des interfaces d'entrée/sortie. Un port d'entrée et un port de sortiePeut être lié par des connexions créées dans un seul niveau de hiérarchie

Un module, c'est-à-dire une connexion entre ou entre deux sous-modules d'un module composé

Un sous-module et un module qui le compose. Différents niveaux de connexion

Les hiérarchies ne sont pas autorisées, car elles constituent un inconvénient à la réutilisationComposants du modèle.

Tout en respectant la hiérarchie de ses modèles, les messages transitent parUne série de connexions entre des modules simples. Cela se fait en toute transparenceModules composites.

Plusieurs propriétés peuvent être affectées à une connexion, par exemplePropagation, débit et taux d'erreur. Un type de connexion appelé canal peutSpécifiez ensuite pour corriger ces propriétés et faciliter leur réutilisation pour d'autres connecter.

Enfin, des paramètres peuvent être associés à chaque module, ce qui permettra de construire sa topologie, par exemple en précisant le nombre de nœuds. Ils s'appliquent également à prend du code C++ implémentant un module simple comme entrée(34).

Ces paramètres peuvent être de type nombre, chaîne ou XML. Valeurspar défaut peuvent leur être attribués, s'ils ne sont pas attribués, ils seront utilisésaprès.

De plus, un paramètre peut avoir le qualificatif volatile, ce qui provoque une relecture l'expression qui apparaît à chaque fois que vous la lisez. C'est pour simulation, auquel cas il simule une distribution de nombres aléatoires.

2. Le langage de description de réseau (NED) :

Décrire la topologie des modèles OMNeT++ en utilisant le langage NED(Description du réseau) (Klaus, Mesut et James, 2010). Il permet aux utilisateurs de déclarerDes modules simples, connectez-les et assemblez-les en modules composites. il préciseAttribuez à nouveau le mot-clé réseau au simulateur pour simuler le module composite final(34).

NED est un langage déclaratif. Comme mentionné précédemment, il est basé sur une structureen couches, avec son fonctionnement orienté composants réduit la complexité de mise en œuvre des modules et facilite le déploiement plusieurs bibliothèques telles que INET et MiXiM.

Il a également plusieurs autres améliorations à sa flexibilité etce qui les rend idéales pour les grands projets (Varga et Hornig, 2008). Nous pouvons citer :

• **Héritage :**permet de sous-classe les modules et les canaux, il est donc possible

Ajoutez d'autres membres (paramètres, ports et sous-modules) selon vos besoins. Il permet également ajuster la structure ou le comportement des modules dérivés en modifiant la structure ou le comportement des modules dérivésparamètres existants.

• **Interface :** Parfois, vous devez déterminer le type exact d'un module ou d'un canal en paramètres lors de la configuration du réseau. Ces interfaces peuvent être réservéesespaces nécessaires dans le code où les modules ou les canaux doivent être utilisés. Ces genres bien entendu, l'objet concret doit implémenter l'interface à surcharger.

• **Packages :** La mise en place de grands projets nécessite souvent l'organisation de documentsNED se trouve dans un répertoire différent. Le support NED est proche de langage

Java. Simulele noyau pouvant trouver la racine de (s) les packages qui utilisent la variable NEDPATH (similaire à CLASSPATH en Java).

Ce vous permet essentiellement de mieux organiser vos fichiers NED et surtout de réduire les conflits noms entre différents modèles.

• **Métadonnées :** de nouvelles propriétés peuvent être ajoutées, sous la forme métadonnées, presque tous les attributs du langage NED (modules, canaux, ports,paramètres, packages, ...). Ceux-ci ne sont pas directement utilisés par le noyausimulé, mais peut fournir des informations supplémentaires à l'environnementexécution et modules pendant la simulation ; par exemple, ils sont utilisés pour spécifie l'unité de mesure de certains paramètres (ex : paramètre post-délai les attributs @unit(us) doivent être exprimés en microsecondes). ça existe aussi en module d'animation et de représentation graphique (spécification des icônes,info bulles, ...) et dans les outils d'édition et d'invite de commande qui les utilisentcomme moyen d'aide(34).

De plus, il est important de mentionner que le langage NED a des représentations XML équivalentes qui permettent la conversion en fichiers XML et vice versa sans perte de données. Cela permet d'automatiser le traitement de ces fichiers dans une troisième langue dans le sens où des informations pertinentes peuvent être extraites, des fichiers NED peuvent être facilement reconstruits, transformés, voire générés à partir d'informations stockées dans des bases de données(34).

Notez également que les fichiers NED peuvent être modifiés en mode graphique ou texte via l'IDE OMNeT++.

Enfin, NED ne définit que la structure du modèle sans tenir compte du comportement du modèle, laissant un sous-ensemble de paramètres ouvert. Ce comportement est corrigé à l'aide d'une simple implémentation modulaire C++. Les paramètres tirent leurs valeurs d'un fichier INI nommé configuration.

3. Programmation des modules simples :

OMNeT++ fournit un environnement de développement intégré (IDE) avec tous les outils nécessaires pour créer, exécuter et déboguer des programmes (Klaus, Mesut & James, 2010).

Le comportement de chaque module simple est implémenté à l'aide de classes C++. Pour ce faire, l'utilisateur doit dériver la classe de base cSimpleModule, redéfinir ses fonctions virtuelles, et enfin enregistrer la nouvelle classe via la macro Define_Module().

L'interaction entre les différents modules se fait souvent par le biais de messages. Ils sont utilisés pour transmettre des informations utiles entre les niveaux d'application. Il est également utilisé comme déclencheur d'événement, car le noyau de simulation ne fait pas la distinction entre les messages et les événements.

La gestion des messages est effectuée dans la fonction handleMessage(cMessage *msg) d'une classe qui implémente un module simple. Chaque module peut alors décider d'envoyer immédiatement le message traité à l'aide de la fonction send(cMessage *msg) ou de s'envoyer un message à lui-même à l'aide de la fonction scheduleAt(simtime_t time, cMessage *msg) pour retarder la transmission d'un laps de temps qu'il y a.

Les attributs de la classe de base cMessage vous permettent de stocker plusieurs informations telles que le nom, la longueur, le type et d'autres champs qui transmettent des informations supplémentaires telles que l'heure d'arrivée, le port de destination et l'heure du dernier envoi.

Vous pouvez également ajouter de nouveaux attributs aux messages en étendant la classe de base. Ceci est fait automatiquement par l'outil OMNeT++ opp-msgc, qui génère les fichiers C++ nécessaires à partir des fichiers de message (.msg).

Un autre aspect fondamental d'OMNeT++ est la modélisation des piles de protocoles réseau. En pratique, les couches réseau sont souvent implémentées sous forme de modules qui échangent des messages via plusieurs mécanismes. D'une part, grâce aux opérations d'encapsulation/décapsulation implémentées par la classe cMessage, et d'autre part, grâce à l'attachement de champs d'informations auxiliaires. À un objet appelé Control Information. Ensuite, lorsque les messages sont reçus au niveau de l'audience, ils sont ignorés et traités. Par exemple, lorsqu'un datagramme IP descend vers la couche Ethernet, le champ d'informations de contrôle associé peut contenir l'adresse MAC de destination. Dans le sens inverse, ce champ peut contenir une adresse IP ou une connexion TCP au niveau supérieur(34).

Parmi les principales fonctions virtuelles qui doivent être remplacées lors de la mise en œuvre du modèle OMNeT++ figurent les fonctions d'initialisation et d'arrêt. C'est le premier qui se produit pour la plupart des opérations d'initialisation, en particulier les membres de données qui prennent en charge les mécanismes d'initialisation multi phase. Cela se fait lorsque le code d'initialisation du

module dépend d'un autre module qui a déjà été initialisé. Dans ce cas, le processus d'initialisation se fait de manière incrémentale en plusieurs étapes. Cette solution est plus propre que les approches traditionnelles d'autres simulateurs tels que OPNET et NS qui diffusent globalement les événements d'initialisation au début de la simulation.

La transmission des paramètres du fichier NED au code C++ du module s'effectue dans la fonction d'initialisation à l'aide de la méthode par().

Pour les fonctions de sortie, elles sont souvent utilisées pour stocker les résultats des statistiques scalaires à la fin d'une simulation réussie.

Il est à noter que la communication par messages n'est pas toujours la solution la plus efficace, notamment pour les modules fortement couplés. Si tel est le cas, il serait plus logique d'utiliser des appels de méthode directs, c'est-à-dire des appels de fonction publics simples de modules. Cependant, cela nécessite deux opérations supplémentaires : trouver le module appelé et enregistrer la méthode d'appel auprès du moteur de modélisation.

Plusieurs méthodes sont disponibles pour trouver le module appelé par emplacement. Dans le cas le plus courant, lorsque le module appelé est dans le même module composite que l'appelant, les deux méthodes getParentModule() et getSubModule() de la classe cModule sont : Utilisé pour renvoyer un pointeur vers le module appelé. Sinon, vous devez utiliser la méthode cSimulation getModuleByPath(const char *path) pour identifier le module appelé avec un chemin absolu.

Le pointeur vers le module appelé pointe initialement sur un objet de type cModule ou cSimulation. La fonction check_and_cast<>() se charge de forcer le typage vers le type actuel.

Pour enregistrer la méthode appelante auprès du moteur de simulation, vous devez utiliser l'une des deux méthodes publiques décrites ci-dessus, Enter_Method() ou Enter_Method_Silent(). Cela détermine si l'appel est visible dans l'interface utilisateur graphique (GUI) et d'autres raisons de changement de contexte temporaire.

Les bibliothèques INET utilisent largement ce mode d'appel pour accéder à des modules tels que RoutingTable, InterfaceTable et NotificationBoard sur l'hôte.

OMNeT++ fournit également un ensemble puissant de fonctionnalités de débogage. Les actions et les événements qui se produisent pendant l'exécution du modèle peuvent être affichés sous forme de texte ou d'animations annotées à l'aide de l'interface graphique Tkenv. Vous pouvez également utiliser la macro WATCH() pour surveiller les modifications en temps réel de diverses variables de module.

Des classes telles que cOutVector, cStdDev, cDoubleHistogram et cLongHistogram sont utilisées pour stocker et calculer des statistiques de base.

Enfin, concernant la relation entre les fichiers NED et les implémentations de modules C++, OMNeT++ compile les fichiers NED en code C++, puis les lie dans l'exécutable du simulateur. Pouvoir créer dynamiquement des modules composites est avantageux. Grâce au code NED compilé, les sous-modules et les liaisons internes peuvent être créés automatiquement.

L'importance de cet aspect réside dans le fait qu'OMNeT++ étend la portée des scénarios de modélisation pour comprendre les topologies qui peuvent changer à l'exécution et ainsi répondre à de nouvelles questions d'optimisation. Filet.

4. Bibliothèques :

On peut classer les bibliothèques en OMNeT++ en deux catégories : une première regroupant les bibliothèques de modèles, la deuxième englobe les bibliothèques de simulation.

4.1. Bibliothèques de modèles :

L'importance de cet aspect est qu'OMNeT++ prend en charge les scripts de modélisation pour comprendre les topologies qui peuvent changer au moment de l'exécution et fournir des réponses aux nouvelles questions d'optimisation. Filet.

La bibliothèque de modèles fait partie d'un projet multidomaine plus vaste, souvent appelé cadre, et fournit une large gamme de modèles de protocole, de modèles d'application et de source de trafic, ainsi que d'autres composants réutilisables qui suivent le paradigme modulaire OMNeT++. Chaque Framework suit son propre cycle de publication indépendant d'OMNeT++. Parmi les OMNeT++ les plus connus on peut citer le Framework INET et les Framework mobiles (MiXiM, Castalia, …)(34).

a) INET Framework :

INET est considéré comme la bibliothèque de modèles standard OMNeT++. Il est basé sur le package IPSuite développé à l'origine à l'Université de Karlstuhe et maintenu avec des modifications et de nouveaux modèles par l'équipe OMNeT++(34).

Comprend des modèles de protocole pour les suites TCP/IP actuelles (IPv4, IPv6, TCP, SCTP, UDP,...), des modèles de couche de liaison pour les réseaux filaires et sans fil (Ethernet, PPP,

IEEE802.11,...), MPLS il y a . . Modèles avec signalisation RSVP et LDP, prise en charge de la mobilité et certains autres protocoles et composants.

Ses modules sont organisés en packages eux-mêmes organisés selon la hiérarchie du modèle OSI (par exemple inet.applications, inet.transport, etc.).

D'un point de vue architectural, INET adhère au concept modulaire d'OMNeT++.

Les protocoles sont exprimés sous forme de modules simples, les interfaces externes sont décrites par des fichiers NED et les comportements sont implémentés à l'aide de classes C++. Les nœuds sont construits en construisant quelques modules simples(34).

D'autres modules (qui n'implémentent pas de protocoles) sont utilisés pour effectuer des tâches spécifiques pendant la simulation :

Côté hôte, on trouve un module **InterfaceTable** qui contient les tables d'interface réseau (eth0, wlan0, ...), les tables de routage **RoutingTable** et **RoutingTable6** pour IPv4 et IPv6 respectivement, et le module NotificationBoard pour faciliter la communication entre les différents modules.

Au niveau de la couche réseau, on citera le module FlatNetworkConfigurator utilisé pour attribuer des adresses IP à différents nœuds et configurer le routage statique, le module **ScenarioManager** pour gérer les expériences de simulation et la planification des événements, et le module **ChannelControl** requis pour la simulation sans flux. Cela aide à garder une trace des nœuds dans les zones d'interférence avec d'autres nœuds.

En ce qui concerne les interactions entre divers éléments, INET gère la communication entre différentes couches de protocole via un processus d'encapsulation/décapsulation utilisant des informations de contrôle sous forme d'objets attachés aux messages pour transmettre des informations supplémentaires à la couche suivante.

Le mode d'appel direct est souvent utilisé pour relier d'autres modules en état de liaison. Celle-ci est assurée par le module **NotificationBoard**qui sert d'intermédiaire entre les modules pour lesquels des événements apparaissent et les modules qui s'intéressent à ces événements. Son fonctionnement est basé sur le concept de publication/abonnement où les modules peuvent s'abonner à des catégories de changements (par exemple, l'état d'une table de routage change et un canal de communication est libéré). Lorsque l'un des changements se produit, le module hôte (par exemple, table de routage, couche physique) notifie le module **NotificationBoard**, et le module **NotificationBoard** diffuse les informations à tous les modules de cette catégorie de changement.

INET est maintenant un cadre important pour OMNeT++. De par la richesse du modèle et la réutilisabilité de ses composants, il a été largement adopté par la communauté OMNeT++ et a pu servir de base à plusieurs extensions, comme CoRE4INET, une extension INET qui implémente le protocole **TTEthernet**.

b) Les frame works de mobilité :

On distingue actuellement deuxFramework : le cas Talia et MiXiM. Conçu pour les réseaux mobiles et sans fil (réseaux de capteurs, BAN, ad hoc, réseaux véhiculaires, etc.) et généralement réservé aux équipements réseaux embarqués de faible puissance.

Fournit des modèles pour la propagation radio, l'estimation des interférences, la consommation d'énergie des émetteurs/récepteurs sans fil et les protocoles MAC sans fil.

5. Quelques avantages d'OMNET++ :

Architecture modulaire permettant l'intégration de nouveaux modèles ;

•Utilisation du C++ (et récemment du C#) pour le développement du noyau ;

•Les classes de base du simulateur peuvent être étendues et personnalisées ;

•Conception de modèles se rapprochant de la réalité ;

II. Topologie et Scénarios Simulés :

Les scénarios choisis obéissent à la propriété single-homed ou multi homed selon les simulations. Chaque noeud établit une association avec un autre noeud en envoyant et recevant des paquets SCTP sur plusieurs streams.
Le trafic pris en considération peut être un flux continu comme FTP ou un service web.
Nous avons essayé que le trafic généré soit le plus varié possible **:** une page peut contenir des objets de différent type (texte, image, séquence vidéo). Ceci était bien possible avec la loi de Pareto(15).

1. Les principales phases d'une simulation OMNeT++ sont les suivantes :

Phase d'initialisation : dans cette phase, l'environnement de simulation est configuré et L'état initial de la simulation est configuré. Cela inclut la création des modules réseau, l'initialisation des paramètres et des variables et la configuration de la planification des événements. Dans le code de simulation donné, la phase d'initialisation est gérée par la fonction initialize() du module MyNetworkNode. Les files d'attente (emergencyQueue et regularQueue) sont initialisées et la première génération de message est planifiée.
Phase de traitement des événements : une fois la phase d'initialisation terminée, la simulation passe à la phase de traitement des événements. Dans cette phase, les événements planifiés sont exécutés un par un en fonction de leur heure planifiée. Dans le code de simulation donné, la phase de traitement des événements est gérée par la fonction handleMessage(cMessage* msg) du module MyNetworkNode. Cette fonction est responsable du traitement des messages entrants et de la génération de nouveaux messages. Si le message entrant est un message personnel (événement planifié), un message aléatoire est généré et la fonction de traitement appropriée (processEmergencyMessage() ou processRegularMessage()) est appelée en fonction du type de message.
Phase de fin : la simulation continue de traiter les événements jusqu'à ce qu'il n'y ait plus d'événements programmés dans la file d'attente d'événements. Une fois tous les événements traités, la simulation entre dans la phase de terminaison. Au cours de cette phase, toutes les tâches de nettoyage ou de finalisation nécessaires sont effectuées. Dans le code de simulation donné, la phase de terminaison n'est pas explicitement indiquée, mais elle implique généralement la libération de toutes les ressources allouées et l'impression des statistiques ou des résultats finaux.

Il est important de noter que les simulations OMNeT++ sont pilotées par les événements, ce qui signifie que la simulation progresse en traitant les événements et en planifiant de nouveaux événements en fonction de la logique de simulation. Les phases décrites ci-dessus sont exécutées de manière répétée jusqu'à ce que la simulation se termine.

2. Analyse des Résultats De La Simulation

Dans ce qui suit, nous allons comparer le SCTP standard et le SCTP avec priorité PQ. En effet, les simulations ont montré qu'il y a une différence entre les deux modèles.
L'algorithme de priorité choisi a introduit un gain notable en termes de taux de livraison et du délai de transmission.

2.1. Influence du type de trafic engendré

Pour nos simulations nous avons joué sur la taille et le contenu d'une page web. En effet, le contenu de cette page est vu comme une suite d'objets de nombre et de tailles variable la taille d'un objet est prise comme une variable qui suit la loi de Pareto II, de moyenne et forme variable.
les messages d'urgences sont présenté sous la forme d'une page web simple.
Ceci était avec les commandes (15):
set objSize [new RandomVariable/ParetoII]
$objSize set avg_ 10 (variable de 10 à200 selon le cas)
$objSize set shape_ 1.2
Le nombre de streams utilisés est le même que celui des degrés de priorité.

a. Taux de Livraison :

La figure 31 montre le taux de livraison de paquets(PDF) pour un flux FTP.

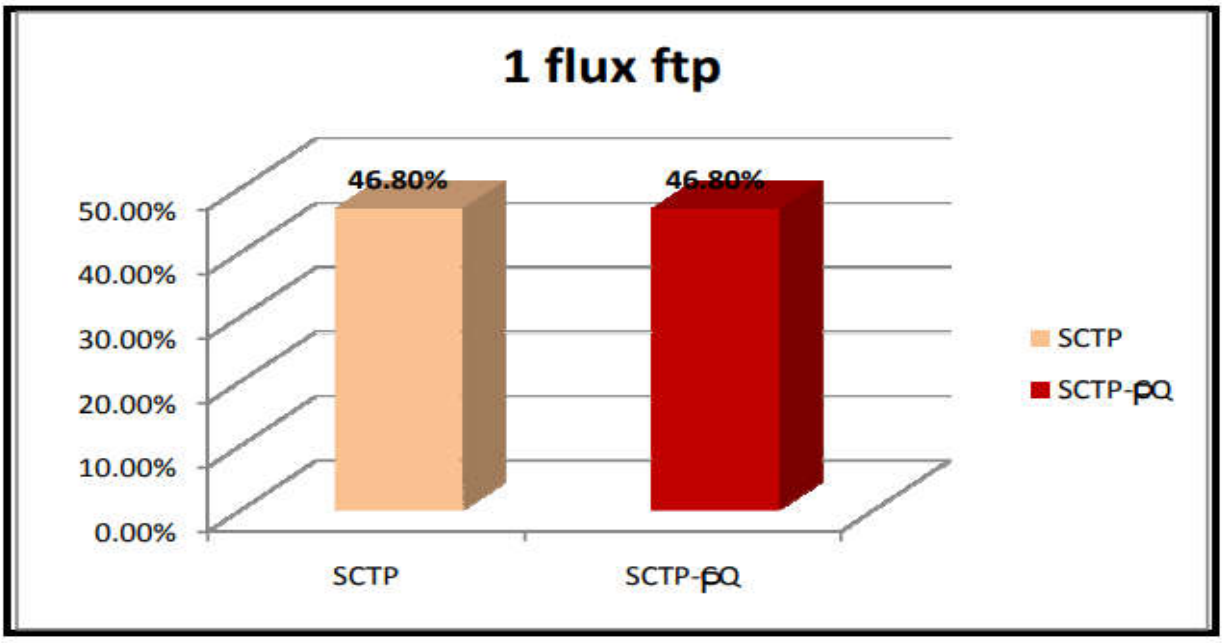

Figure 31 : influence du mode de gestion des files d'attentes Priority Queuing pour un trafic FTP

Le comportement du protocole SCTP dans le cas standard et dans le cas d'ajout du mécanisme priority Queuing est le même pour un trafic FTP.
Ce résultat est bien prévisible puisque le FTP utilise un seul flux de donnée (un seul stream).

Cette étape nous a aidés à conclure que l'implémentation du mécanisme Priority Queuingn'influe pas sur lesfonctionnalités de base de SCTP.
La figure 32 montre la variation du taux de livraison suite à une variation de la taille d'une page web.

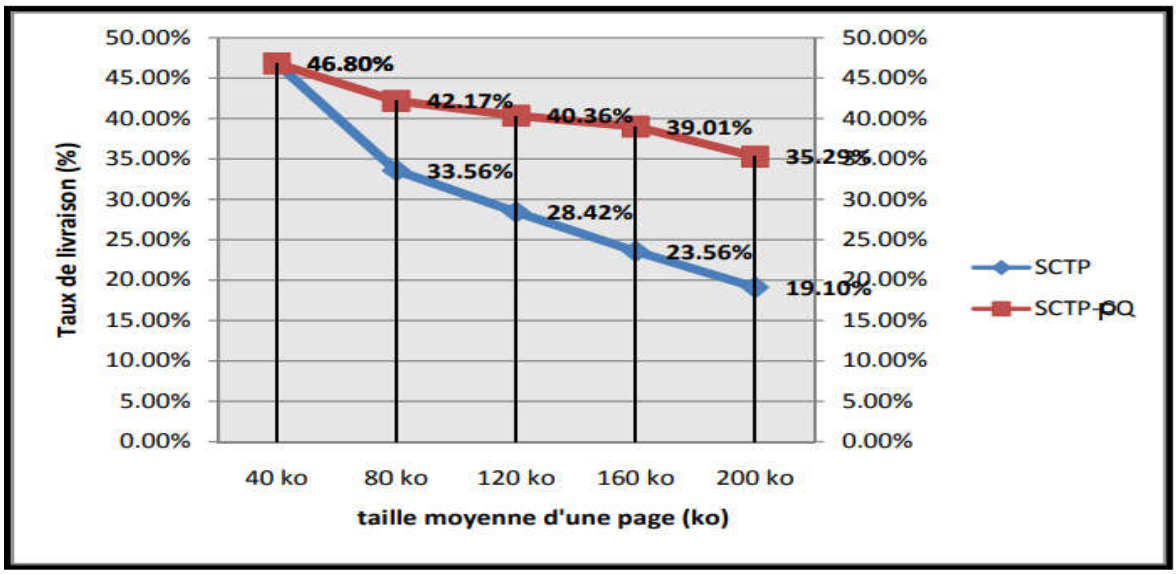

Figure 30 : influence du mode de gestion des files d'attentes Priority Queuing sur variation du taux de livraison

Dans les deux cas, SCTP standards et SCTP-PQ, le taux de livraison diminue avec l'augmentation de la taille des données à transmettre. En effet plus le client ouvre un page web qui comporte des objets de tailles plus grandes, plus des paquets seront perdue c-à-dire la diminution du taux de livraison résulte une augmentation du taux de perte.
Mais il reste toujours que les résultats fournis par SCTP avec Priority Queuing sont meilleur que celles fournis par SCTP.

b. Délai de transmission :

Nous présentons dans ce qui suit la variation du délai moyen de transmission des paquets en fonction dela taille d'une page web. Celle-ci doit être le plus variée que possible pour voire l'influence de l'ajout dePriority Queuing sur la gestion des files d'attente de Stream 's.
Dans ce cas nous avons utilisés dans notre scénario deux nœuds mobile qui se connecte à un nœud fixe(un serveur http par exemple).

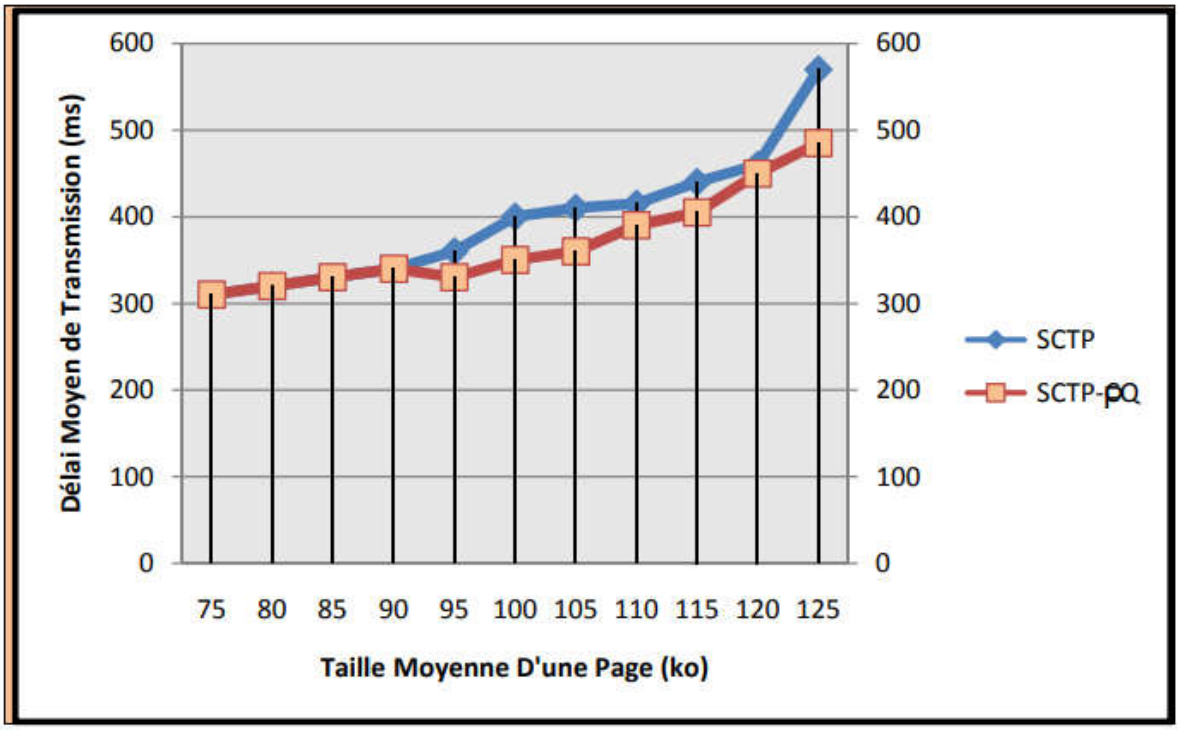

Figure 31 influence du mode de gestion des files d'attentes Priority Queuing pour un trafic varié

Les résultats fournis par la figure 33 permettent de déduire que :

Les deux versions du protocole SCTP ont un comportement similaire. En effet, pour une taille moyennede page inférieur à 90 ko, les délais moyens ont des valeurs presque identiques et ne dépasse pas 350 ms.

Dès que la taille moyenne de la page dépasse les 100 ko, le Priority Queuing intervient dans le délai detransmission. Ceci peut être expliqué par le fait qu'à de telles valeurs les objets sont de plus en plusvariés, et que ce mode de gestion de files d'attentes aide à répartir la bande passante entre les différents flux, comme il permet de diminuer le temps de réponse.

Après avoir étudié le comportement de deux noeuds mobiles, nous avons ajouté d'autres nœudsafind'observer la variation du délai moyen de transmission en fonction de la densité du réseau.

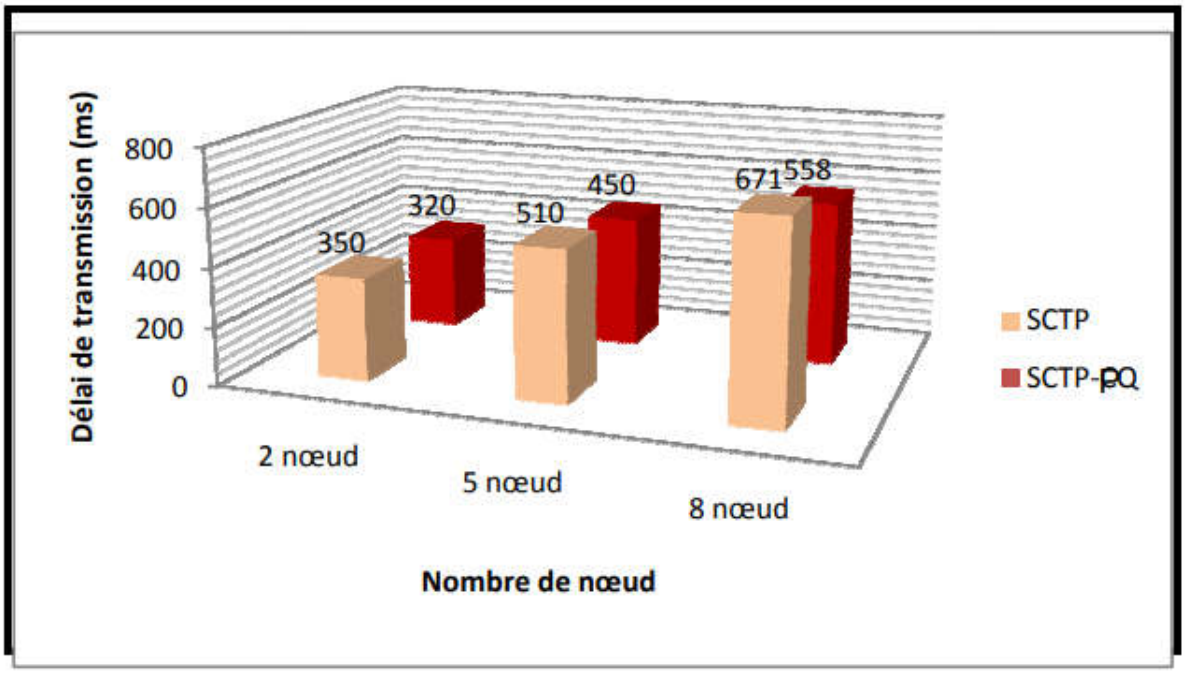

Figure 32 variation du délai de transmission en fonction de la densité du réseau

Dans la figure 34nous remarquerons que Priority Queuing réduit les délais de transmission, nous remarquerons aussi une croissance du délai de transmission en fonction du nombre de noeud, nous justifions cette croissance par l'augmentation du trafic échangé ce qui provoque une saturation des files d'attentes, (et dans le cas réel une augmentation du taux de perte surtout pour les paquets qui ont un délai d'expiration), mais il reste toujours que les résultats de SCTP-PQ sont meilleurs que celles de SCTP,

2.2. Influence de la vitesse du mouvement des nœuds :

Dans ce cas nous avons utilisés pour nos simulations 4 noeuds mobiles et 2 hôtes fixes afin de simuler un environnement à densité moyenne.

a. Taux de Livraison :

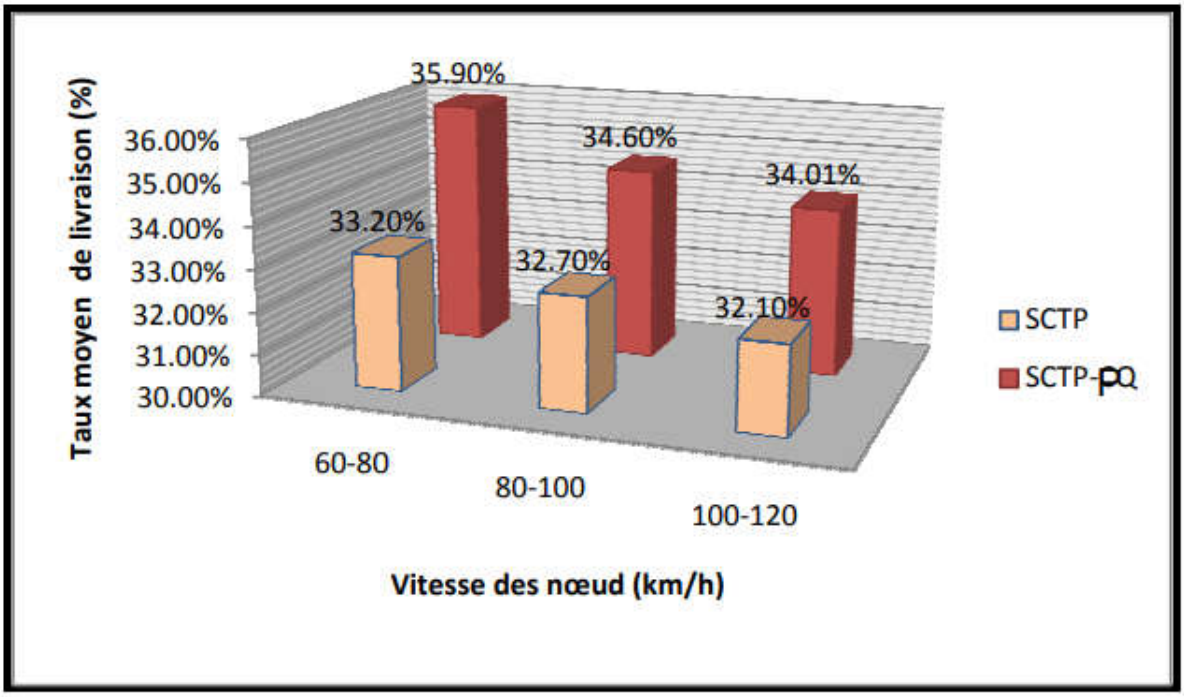

Figure 33 Variation du taux de livraison moyen en fonction de la vitesse des nœuds

Nous remarquons dans la Figure 35 que le taux de livraison diminue avec l'augmentation de la mobilité des nœuds mais avec le support de la mobilité que peut fournir SCTP le problème de déconnexion n'est plus fréquent.

Et ceci apparaît plus avec Mobile SCTP dont sa fonctionnalité principale permettant de maintenir une association active lors d'un changement de réseau IP (la fonction ADDIP), définie dans (35) et permettant de modifier / supprimer/ajouter une adresse IP faisant partie de l'association, sans que celle-là ne soit interrompue.

b. Délai de transmission :

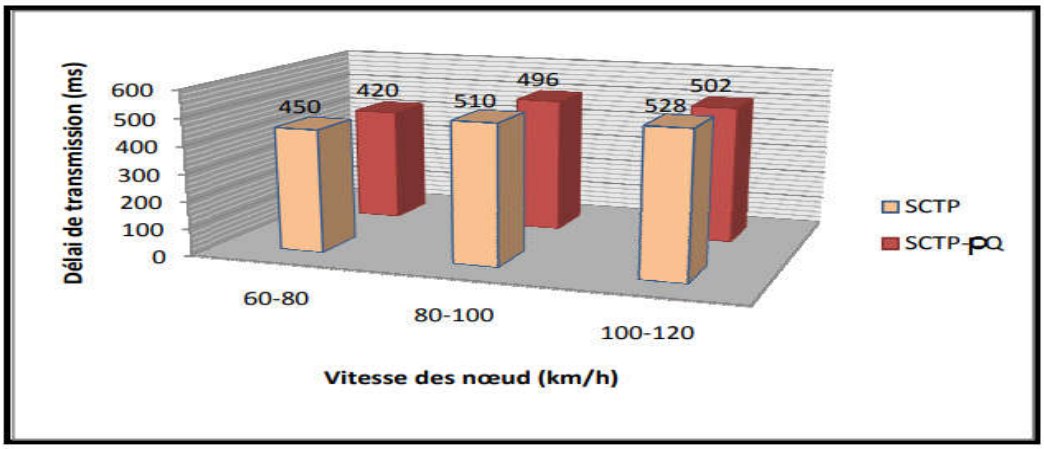

Figure 34 Variation du délai de transmission en fonction de la vitesse des nœuds

Avec la mobilité le délai moyen de transmission se varie selon la vitesse des nœuds
Mais il ne change pas beaucoup il varie entre 420 ms et 530 ms.

Et c'est toujours grâce au support de la mobilité que peut fournir SCTP

Conclusion :

Dans ce chapitre nous avons présenté une comparaison de performances du protocole SCTP dans son implémentation standard et suite à un ajout du mécanisme Priority Queuing, nous avons validé dans un premier volet notre solution à travers l'exemple qui présente l'influence de notre solution sur le taux de livraison d'un trafic FTP,

Nous avons présenté aussi son influence sur la variation du taux de livraison et du délai de transmission dans le cas du transfert d'une page web de taille moyenne, puis nous avons aussi rajouté des nœuds dans notre simulation afin de voir l'influence de la densité du réseau sur le délai de transmission.

A la fin nous avons testé notre solution afin de voire Influence de la vitesse du mouvement des nœuds sur la variation du taux de livraison et du délai de transmission.

D'après nos simulations, il s'est avéré que l'ajout du mécanisme de gestion des files d'attentes Priority Queuing améliore le taux de livraison et réduit les délais de transmission. En effet, le Priority Queuing nous aider à améliorer la QOS du protocole SCTP grâce à ses différentes fonctionnalités.

Conclusion générale

Dans les réseaux VANETs, la plupart des applications de sécurité routière se basent sur la diffusion pour disséminer les messages sur une zone géographique spécifiée.

Nous nous sommes intéressés dans ce mémoire à la dissémination Orientée priorité des messages pour répondre aux différents besoins en qualité de service des multiples applications des VANETs, des solutions de dissémination proposent une adaptation de la dissémination par rapport à l'importance du contenu des messages échangés.
Afin de ne pas supprimer systématiquement tous les nouveaux messages entrants en cas de congestion du réseau.

Cependant, les services de communication actuellement disponibles au niveau de la couche transport ne peuvent pas entièrement répondre à ces exigences complexes de qualité de service (QoS).

Notre projet d'étude est concentré sur l'évaluation des performances Notre approche a été d'incorporer des mécanismes de contrôle de QoS au niveau du protocole SCTP afin de pouvoir l'utiliser pour la dissémination des donnés et surtout la dissémination des messages d'urgences .

Nous avons d'abord présenté des informations générales sur les réseaux sans fil mobiles, qui se répartissent généralement en deux catégories : les réseaux avec infrastructure utilisant un modèle cellulaire, et les réseaux sans infrastructure, ou réseaux ad hoc.

Après la présentation des réseaux mobiles sans fil, nous avons détaillé le domaine des réseaux ad hoc, en présentant les problèmes liés à ce domaine qui sont la gestion de la mobilité et le support de la QoS.

le protocole SCTP est une solution proposée par l'IETF pour faire face au problème de gestion de la mobilité dans les réseaux ad hoc ; La gestion de la mobilité constitue un important défi technique à relever, un protocole de mobilité efficace doit pouvoir empêcher la terminaison forcée de l'appel et permettre l'exécution des applications d'une manière transparente au mouvement de l'utilisateur.

Ceci nécessite le développement de nouveaux concepts et de nouvelles solutions qui peuvent assurer une gestion efficace de la mobilité d'une part et un support de qualité de service QoS d'autre part.

La qualité de service ou qualité de service (QoS) est un concept important dans les réseaux et les télécommunications. En effet, pour que des applications multimédia émergent sur un même réseau, des politiques de QoS doivent être mises en place pour gérer ces différents flux.

Pour fournir ce dernier, un mécanisme de mise en file d'attente est nécessaire pour trouver une solution qui améliore la dissémination des messages d'urgence .

À travers ce mémoire, nous avons été en mesure de comprendre les différentes étapes par lesquelles un projet de recherche doit progresser. Nous avons également acquis une expérience interne et un bon aperçu des méthodes de travail d’un chercheur, avec une connaissance approfondie dans ce domaine.

Bibliographies

1. **H. Labiod.** «Wireless Ad Hoc and Sensor Networks»,. 2010.

2. **S. Kassab, M. Oularbi,.** « élaboration d'un protocole de routage efficace en énergie pour les réseaux de capteurs sans fil », mémoire de fin d'études Pour l'obtention du diplôme d'Ingénieur d'Etat en Informatique,. 2010.

3. **MEERSCHEN, M. VAN DER.** « Hybridation entre les modes ad-hoc et infrastructure dans les réseaux de type Wi-Fi », Mémoire de fin d'études présenté en vue de l'obtention du grade d'Ingénieur Civil Informaticien en Sciences Appliquées, Université Libre de. 2006.

4. **M. Smith., D. Cook, & B. Smith,.** « Dairy Science & Technology », Second Edition , CRC Taylor & Francis Group,. 2001.

5. **ahm2click.**

6. **M. TAHAR ABBES.** « Proposition d'un protocole á économie d'énergie dans un réseau hybride GSM et AD HOC Devant », mémoire doctorat, Université d'Oran, . 2012.

7. **Bernsen, James, and D. Manivannan.** "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification." Pervasive and Mobile computing 5.1 . 2009.

8. **TCHEPNDA, Christian.** Authentification dans les réseaux véhiculaires opérés. Thèse de doctorat. Télécom ParisTech. 2008.

9. **Tchepnda., Christian.** "Authentification dans les Réseaux Véhiculaires Opérés", Thèse de Doctorat, École Nationale Supérieure des Télécommunications Spécialité : Informatique et Réseaux, Paris- France , . 18 Décembre 2008.

10. **Laanaoui, My Driss, and Said Raghay.** "Greedy forwarding mechanism and decomposition areas in urban environment for VANET." 2014 International Conference on Multimedia Computing and Systems (ICMCS). . 2014.

11. [Online] [Cited: 05 10, 2023.] https://www.techtarget.com/whatis/definition/vehicle-to-infrastructure-V2I-or-V2X?amp=1..

12. [Online] [Cited: 05 10, 2023.] / https://blog.rgbsi.com/connection-with-vehicle-to-network-v2n. .

13. **Nguyen, Bach Long,.** "A joint scheduling and power control scheme for hybrid I2V/V2V networks." IEEE Transactions on Vehicular Technology . 2020.

14. **/ Anaya, José Javier, Pierre Merdrignac, Oyunchimeg Shagdar, Fawzi Nashashibi, and José E. Naranjo.** / Anaya, José Javier, Pierre Merdrignac, O "Vehicle to pedestrian communications for protection of vulnerable road users." In 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. . 2014.

15. **Rmichi, Amina.** "Mécanisme De Qualité De Service Pour La Gestion Des Flux Dans Le Réseau Véhiculaire"université de skikda. 2013.

16. **D. Johnson, C. Perkins, and J. Arkko.** D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775 (Proposed Standard). Juin 2004.

17. **PAUN, Laurentiu Sorin.** «Gestion de la mobilité dans les réseaux ambiants» Thèse pour obtenir le grade de DOCTEUR DE L'INPG Spécialité : « Informatique : Systèmes et Communications » préparée au laboratoire LSR – IMAG dans le cadre de l'École Doctorale. *« Mathématiques, Sciences et Technologies de l'Information ».* Novembre 2005.

18. **Narten., S. Thomson and T.** IPv6 Stateless Address Autoconfiguration. RFC 2462 (Draft Standard), . Décembre 1998.

19. **R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L.** «Stream Control Transmission Protocol», RFC2960, Internet Engineering Task Force, . October 2000.

20.**Inwhee Joe and Latha Kant,** «SCTP with an improved cookie mechanism for wireless networks through modeling and simulation», 58th IEEE VTC Fall, vol.4, pp.2559-2563, October 2003.

21. **Randall Stewart, Chris Metz,.** «SCTP New Transport Protocol for TCP/IP», IEEE Internet Computing, vol.5, no.6, pp.64-69, . November-December 2001.

22. **R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L.Zhang, V. Paxson,.** «Stream Control Transmission Protocol», RFC2960, Internet Engineering Task Force, . October 2000.

23. **R. Stewart, Qiaobing Xie,.** «Stream Control Transmission Protocol (SCTP) : a reference guide»,Addison wesley, London . 2002.

24. **Inwhee Joe and Latha Kant.** «SCTP with an improved cookie mechanism for wireless networks through modeling and simulation», 58th IEEE VTC Fall, vol.4, pp.2559-2563, . October 2003.

25. **Shaojian Fu and Mohammed Atiquzzaman.** SCTP: state of the art in research, products, and technical challenges», IEEE Communications Magazine, vol.42, no. 4, . April 2004,.

26. **Shaojian Fu, Mohammed Atiquzzaman and William Ivancic,.** «Evaluation of SCTP for Space Networks», IEEE Wireless Communications, vol.12, no 5, . October 2005, pp. 54-62.

27. **Y. Wang, L Fan, N. Akthar, K. Chew, R. Tafazoli,.** "An Aggregation-based QoS Architecture for Network Mobility", 14th IST Mobile & Wireless Communications Summit. 2005.

28. **Shaojian Fu, Liran Ma, Mohammed Atiquzzaman, Yong-Jin Lee,.** "Architecture and Performance of SIGMA: A Seamless Mobility Architecture for Data Networks'',p3249-3253, IEEE International Conference on Telecommunications ICC, . , Mai 2005.

29. **Pulak K Chowdhury, William Ivancic,.** "SINEMO: An IP-diversity based Approach for Network Mobility in Space'', Second IEEE International Conference on Space Mission Challenges for Information Technology, SMC-IT'06,. Juillet 2006.

30. **Meriem, TALAI.** Etude et implémentation d'algorithme de mobilité de groupe et. 2010.

31. **Thierry Ernest.** "Les Réseaux Mobiles dans IPv6, support nécessaire au Multimédia", Première Conférence Nationale sur le Multimédia Mobile (MCube), . Mars,2004.

32. **V. Basto, V. Freitas,.** "SCTP Extensions for Time Sensitive Traffic'', Fifth International Network Conference, INC2005, . Juillet 2005.

33. **V. K. Madisetti, D. A. Agyriou,.** “Transport Layer QoS Management for Wireless Multimedia Services’’, Soft.Networks Technical Report, . , September 2002.

34. **ABIDI, Dhafer.** ÉTUDE ET SIMULATION DU PROTOCOLE TTETHERNET SUR UN SOUS-SYSTÈME DE GESTION DE VOLS ET ADAPTATION DE LA PLANIFICATION DES TÂCHES À DES FINS DE SIMULATION. MAI 2015.

35. **draft, IETF.** Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, August. 2002.

36. [Online] [Cited: 05 10, 2023.] https://www.tonex.com/training-courses/vehicle-vehicle-v2v-communications-training/. .

37. [Online] [Cited: 05 10, 2023.] https://www.researchgate.net/figure/Representation-of-V2V-and-V2I-technologies-a-V2V-communication-b-V2I-communication_fig2_275220928.

38. **Rangarajan, Sathyanarayanan,.** "V2c: a secure vehicle to cloud Framework for virtualized and on-demand service provisioning." Proceedings of the International Conference on Advances in Computing, Communications and Informatics. . 2012.

39. [Online] [Cited: 05 10, 2023.] https://www.researchgate.net/figure/Network-architectures-in-VANET_fig2_273338827 .