



Projet Opnet - 2002

Etude et comparaison des protocoles Slotted ALOHA et CPCH pour l'accès à un réseau UMTS

HESSEL Bénédicte
ARLOT Julien
DEROMAS Arnaud
LEFEVRE Eric
MORIN Laurent

Responsable Khaldoun ALAGHA

TABLE DES MATIERES

1. Introduction	4
2. Présentation des protocoles étudiés	5
2.1. Survol de la technologie UMTS	5
2.2. Les protocoles d'accès	6
2.2.1. généralités.....	6
2.2.2. le mécanisme FDMA.....	6
2.2.3. le mécanisme TDMA.....	7
2.2.4. Le mécanisme CDMA	7
2.3. Le protocole ALOHA pur.....	8
2.4. Le protocole slotted ALOHA	9
2.5. Le protocole CPCH/W-CDMA (Common Packet Channel).....	11
3. OPNET Modeler	12
3.1. OPNET, c'est quoi ?	12
3.2. La boîte à outils d'OPNET	12
3.2.1. Les systèmes (devices)	12
3.2.2. Les protocoles.....	13
3.3. Conception du réseau : La logique d'OPNET	13
3.4. Programmer le comportement des processus	15
3.5. Collecte des résultats	15
3.6. Présentation des résultats	16
4. La modélisation	17
4.1. Généralités pour la modélisation	17
4.1.1. Variables promues.....	17
4.1.2. Modélisation du temps.....	17
4.1.3. Modélisation des messages transmis (paquets).....	18
4.2. Le Node model du projet	19
4.3. Modélisation des émetteurs (clients)	19
4.3.1. Automate du générateur.....	19
4.3.2. Etat INIT	20
4.3.3. Etat GENERATE	20
4.3.4. Etat STAT_WRITE	21
4.3.5. Etat STOP	21
4.4. Modélisation de la base Slotted ALOHA	22
4.4.1. Automate de la cellule Slotted ALOHA	22
4.4.2. Etat INIT	22
4.4.3. Etat IDLE	22
4.4.4. Etat NEXT_SLOT	22
4.4.5. Etat PACKET_ARRIV	24
4.4.6. Etat STOP	25
4.5. Modélisation du protocole CPCH.....	25
5. Simulations	26
5.1. Scenarii de test	26
5.2. Statistiques collectées	26
5.2.1. Pourcentage de paquets reçus / paquets générés	26
5.2.2. Paquets reçus suivant l'inter arrivée.....	27
5.2.3. Pourcentage de paquets reçus / paquets générés	27
5.2.4. Temps d'attente moyen par paquet	28

5.2.5. Nombre moyen de préambules par paquet.....	29
5.2.6. Pourcentage de paquets perdus / paquets générés.....	29
5.2.7. Temps d'attente moyen par paquet.....	30
6. Conclusion.....	31
7. Glossaire.....	32

1.Introduction

Le but de ce projet est de comparer les protocoles d'accès aléatoire Slotted ALOHA / mode CDMA (utilisé dans le GSM depuis le 1^{er} janvier 2000) et CPCH / mode W-CDMA (utilisé dans le futur réseau cellulaire UMTS). L'objectif est d'essayer de déterminer quel protocole paraît être le plus robuste face à la montée en charge des clients simultanément connectés à une cellule.

Pour comparer ces deux protocoles, une modélisation sous l'environnement Opnet permettra de simuler le fonctionnement de l'accès à l'UMTS.

Ce document est composé des 6 chapitres suivants:

- Introduction: Présentation du présent rapport.
- Présentation des protocoles étudiés: rappel des protocoles abordés.
- L'outil Opnet: Présentation de l'outil utilisé pour la simulation.
- La modélisation: Description du paramétrage Opnet.
- La simulation: Description des simulations réalisées et présentation des résultats.
- Conclusion.

2.Présentation des protocoles étudiés

2.1.Survol de la technologie UMTS

UMTS (Universal Mobile Telecommunication System) est la technologie qui va équiper la prochaine génération de mobiles. Contrairement aux prévisions qui étaient un peu trop optimistes, les premiers mobiles et services UMTS (dont la téléphonie n'est qu'un aspect parmi d'autres) ne devraient pas être déployés avant 2003-2004.

L'UMTS permet des améliorations substantielles par rapport au GSM, notamment :

- elle rend possible un accès plus rapide à Internet depuis les téléphones portables, par un accroissement significatif des débits des réseaux de téléphonie mobile.
- elle améliore la qualité des communications en tendant vers une qualité d'audition proche de celle de la téléphonie fixe.
- elle permet de concevoir une norme compatible à l'échelle mondiale, contrairement aux technologies actuelles (les normes utilisées aux Etats-Unis et au Japon ne sont pas toutes compatibles avec le GSM) ;
- elle répond au problème croissant de saturation des réseaux GSM, notamment en grandes villes.

L'UMTS devrait supporter des débits supérieurs à 144Kb/s et pouvant aller jusqu'à 2 Mb/s (en zone urbaine, avec une mobilité réduite). Ces débits sont largement supérieurs au débit du GSM (9,6kb/s) et au téléphone fixe (56kb/s).

Techniquement, l'UMTS utilise une technique de transport issue de l'ATM (le protocole AAL2) et repose sur la commutation cellulaire. Le protocole AAL2 a été défini pour permettre à l'ATM de transporter efficacement les applications temps réel à très bas débit et à débit variable.

L'UMTS gère un ensemble de canaux montants (uplink) depuis l'équipement mobile et descendants (downlink) depuis la base.

Le réseau d'accès à l'UMTS emprunte l'interface radio que se partagent déjà tous les utilisateurs. Ceci n'est pas sans engendrer un certain nombre de collisions, sous la forme d'interférences. L'utilisation de protocoles d'accès permet d'éviter ces conflits d'accès. On peut notamment citer le protocole ALOHA.

Une variante de ce protocole qui supporte mieux un nombre important d'utilisateurs, le Slotted ALOHA est utilisé dans tous les systèmes de communication sans fil dont l'UMTS.

Pour améliorer la fiabilité de l'envoi des messages par l'accès aléatoire, l'UMTS a défini un protocole permettant de gérer les collisions : protocole CPCH (Common Packet Channel). Il consiste à rendre la station de base ordonnatrice de l'accès aléatoire qui doit émettre un paquet.

2.2. Les protocoles d'accès

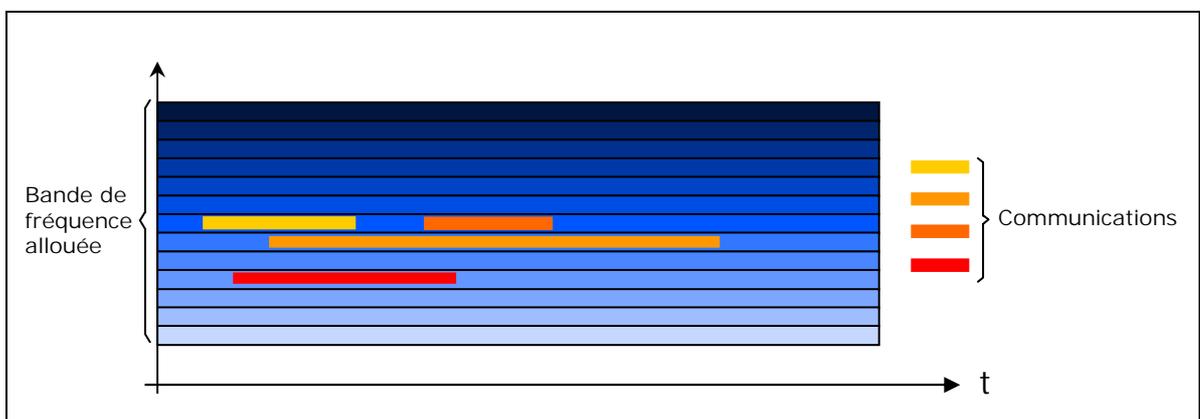
2.2.1. généralités

Les protocoles d'accès au support réglementent l'accès au support physique et sa durée d'utilisation afin de gérer les conflits potentiels. Ces protocoles jouent un rôle important dans les réseaux à diffusion, où tous les utilisateurs sont susceptibles d'émettre et de recevoir à n'importe quel moment. Les protocoles d'accès sont basés sur deux mécanismes d'allocation : statique et dynamique. Un mécanisme d'allocation statique attribue un canal de communication de façon définitive, tandis que le mécanisme dynamique est capable de s'adapter à l'environnement.

Les principales techniques d'accès au support sont les modes FDMA, TDMA et CDMA que nous présentons maintenant.

2.2.2. le mécanisme FDMA

Le mode FDMA consiste à effectuer un découpage fréquentiel du spectre. On obtient ainsi plusieurs porteuses contenues dans la bande de fréquence allouée. Chacune de ces porteuses reçoit une communication (et une seule de manière simultanée). En règle général, le canal physique utilise deux bandes de fréquence (une montante et une descendante).



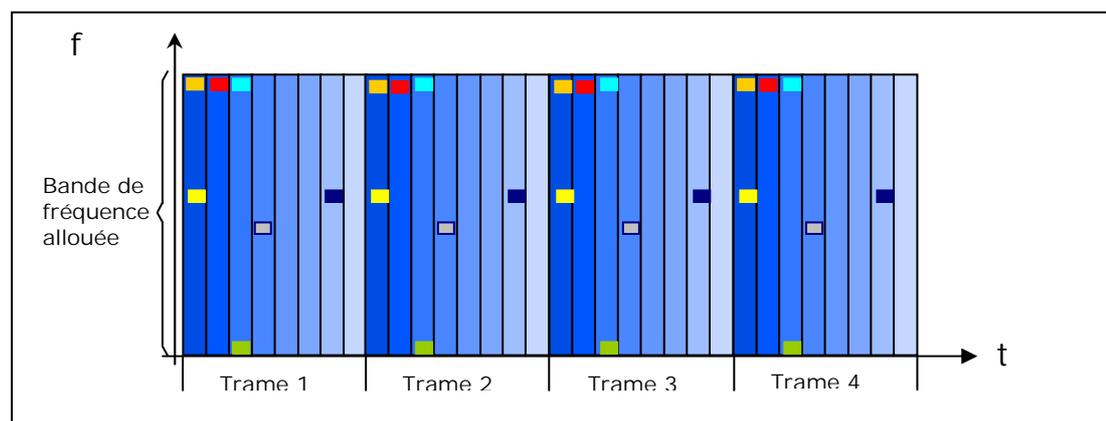
Mode d'accès TDMA

Ce mécanisme était utilisé sur le téléphone BI-BOP. Il présente certains désavantages :

- chaque utilisateur monopolise une porteuse et ce même lors des phases de silence.
- si une gamme de fréquence présente des zones d'ombre, le client perd sa communication.

2.2.3. le mécanisme TDMA

Ce mode d'accès fonctionne sur un partage de la ressource physique dans le temps : chaque porteuse est découpée en éléments temporels appelés Slots. La trame TDMA est définie comme étant égale à huit slots. Lorsqu'un client accède au réseau, un slot lui est assigné (porteuse fixe et numéro de slot, de 1 à 8, dans la trame). Ce client conserve ce slot sur cette fréquence durant toute la durée de sa communication.



Mode d'accès TDMA

Lorsque deux clients émettent simultanément sur le même slot (même porteuse et même numéro de slot dans la trame), il y a collision, les paquets sont perdus et devront être réémis.

Cette technique est bien plus intéressante que le FDMA en terme de taux d'utilisation du canal (on fait passer 8 fois plus de clients sur une fréquence qu'en FDMA).

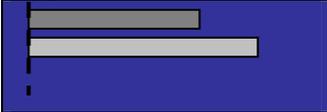
Elle a été utilisée par le GSM jusqu'en 2000 (aujourd'hui, dans la nouvelle norme GSM, les paquets propres à une communication changent de fréquence à chaque trame, ce qui assure une meilleure qualité de service moyenne : chaque utilisateur profite de l'ensemble de la gamme de fréquence).

2.2.4. Le mécanisme CDMA

Ce mécanisme permet d'optimiser l'utilisation de la bande passante par deux améliorations :

- en multiplexant par code les messages qui occupent simultanément un canal. En adjoignant un code d'identification aux trames d'un message, il est possible de les combiner en un seul signal. En réception, on pourra extraire de ce signal les différents messages dans la majeure partie des cas avec un algorithme antagoniste de décombinaison par code. Le codage / extraction des messages s'appuie sur des techniques de mathématiques combinatoires (matrices de Walsh Hadamard, « gold sequences »).

Par rapport à TDMA, CDMA permet donc d'éviter la plupart des collisions, à l'exception du cas où deux messages sont émis strictement en même temps, auquel cas il y aura collision et donc réémission. Il n'y a donc plus de notion de partage de la bande passante (fréquentiel ou temporel) entre les utilisateurs. Chacun a d'une certaine manière toute la bande à sa disposition.

Type de collision	TDMA	CDMA
	collision	collision (mais paquets pas forcément tous perdus, voir ci dessous *)
	collision	pas de collision

- en émettant avant le message un petit paquet dit « préambule de puissance » suivant le protocole ALOHA pur avec une puissance d'émission initiale de p watts. A chaque échec de la transmission du préambule, celui-ci est réémis mais avec une puissance incrémentée (dans la limite du nombre de tentative octroyées, s'il est dépassé il y a échec de la transmission). Lorsque le préambule est acquitté, le message est à son tour transmis à la puissance atteinte. En cas de collision entre messages, le message de plus forte puissance sera transmis (contrairement à l'ALOHA pur) et le paquet plus faibles sera réémis sans pour autant augmenter sa puissance. Bien entendu, il faut qu'un paquet soit significativement plus puissant que les autres (delta de 2 pas de puissance) pour pouvoir être « distingué » des autres. Ainsi dans le type de collision 1 du tableau ci dessus, un message plus puissant que l'(les) autre(s) ne sera pas perdu.

2.3. Le protocole ALOHA pur

Fonctionne avec TDMA (pas / peu de mécanisme d'évitement de collision).

En Hawaïen, « Aloha » signifie « Bonjour ». Ceci résume le fonctionnement de ce protocole quelque peu barbare qui consiste pour le client à envoyer son message sur le canal sans se soucier de savoir si celui est déjà occupé à transmettre ou non des données. Si l'émetteur ne reçoit pas d'acquiescement pour sa transmission durant un intervalle de temps donné, il réémet celui-ci de

la même manière. Si au delà d'un certain nombre de tentative aucun acquittement n'a été reçu, il y a échec de la transmission (« réseau saturé »).

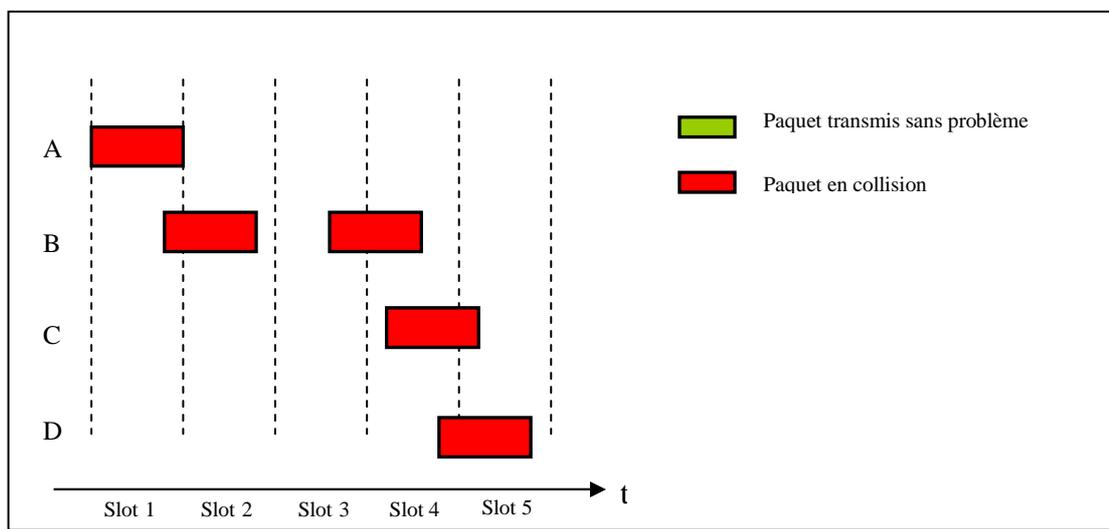
2.4. Le protocole slotted ALOHA

Ce protocole hérite du protocole ALOHA. Il l'améliore en discrétisant le temps en éléments appelés « slots », intervalles d'une durée de 577 µs.

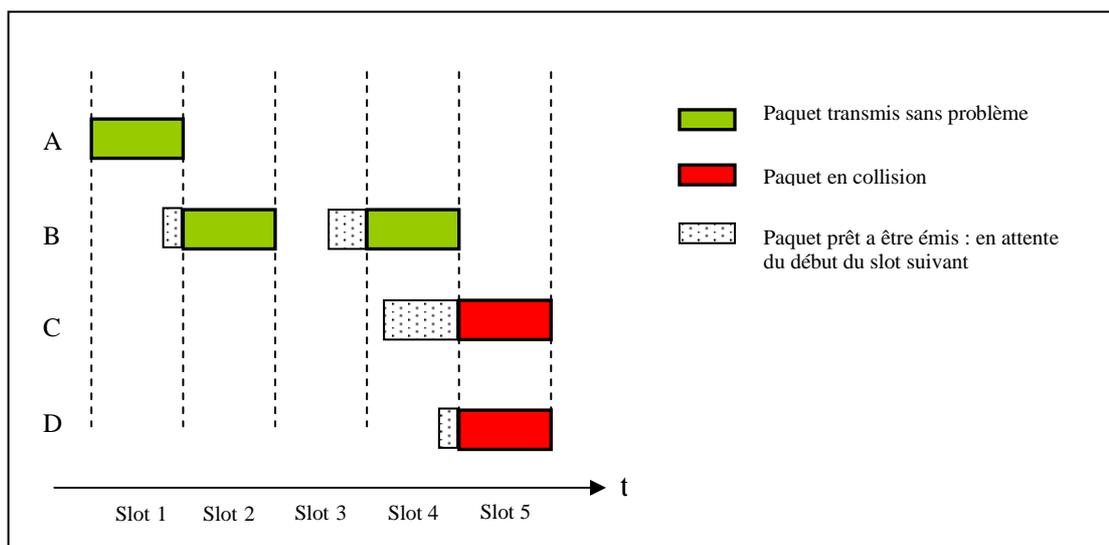
La discrétisation du temps implique que les émetteurs et la base avec laquelle ils communiquent, sont synchronisés sur une même horloge externe : chaque station sait exactement lorsqu'un slot commence.

Deux paquets n'entrent en collision que s'ils sont prêts à être émis dans le même slot. En fonction du type de transmission sélectionnée (CDMA...), le traitement des collisions diffère.

Différence entre ALOHA et Slotted ALOHA



ALOHA



slotted ALOHA

2.5. Le protocole CPCH/W-CDMA (Common Packet Channel)

Ce protocole sera utilisé par le système UTRAN pour un des deux canaux montants.

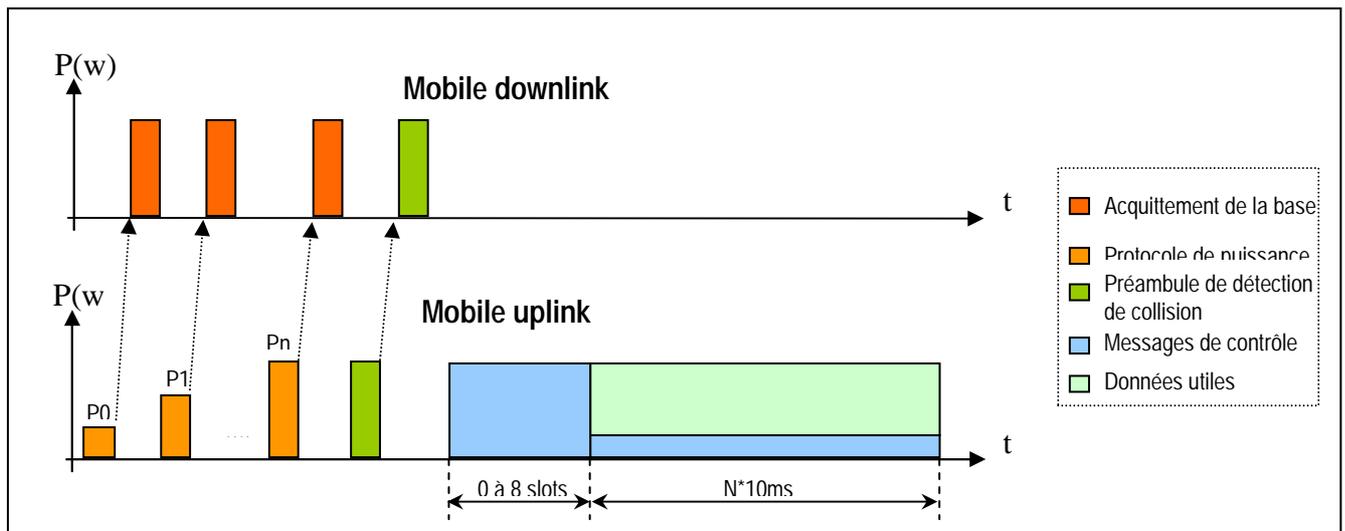
Il utilise W-CDMA pour l'accès, qui n'est autre qu'une version améliorée de CDMA fonctionnant pour une bande passante plus large (5 Ghz au lieu de 1.83Ghz).

Le mécanisme d'ajustement de la puissance du signal (propre à CDMA) est identique à celui adopté dans le protocole Slotted ALOHA / CDMA : lorsqu'il y a collision entre deux paquets, on compare la puissance des deux paquets. On distingue 2 cas :

- Un des deux paquets à une puissance suffisamment forte par rapport à celle de son concurrent pour être discriminée, il est reçu par la base et son concurrent augmente sa puissance d'un pas et réémet son préambule sur échéance d'une temporisation aléatoire.
- Les deux paquets ont des puissances trop proches pour être discriminés, on décide d'augmenter la puissance des deux protagonistes d'un pas et de réémettre les préambules de puissance sur échéance d'une temporisation aléatoire (propre à chacun).

Une fois la puissance optimale du signal déterminée, le mobile envoie un préambule de détection de collision qui lui permet de réserver le canal sur lequel il se trouve (sous réserve d'un acquittement de la cellule). Ce préambule de détection est agrémenté d'un niveau de priorité. Si deux préambules de détection de collision sont mis en concurrence, c'est le plus prioritaire qui l'emporte. A priorité égale, on applique à nouveau le principe de discrimination par le niveau de puissance, à la différence que le plus faible est détruit (les deux en cas de puissance équivalente).

Ce mécanisme de réservation assure que les données transmises par le mobile ne seront jamais en collision. Or la taille d'une zone de données étant de très loin supérieure à celle des préambules, on assure une bien meilleure utilisation de la bande passante du support.



Accès CPCH/W-CDMA

3.OPNET Modeler

3.1. OPNET, c'est quoi ?

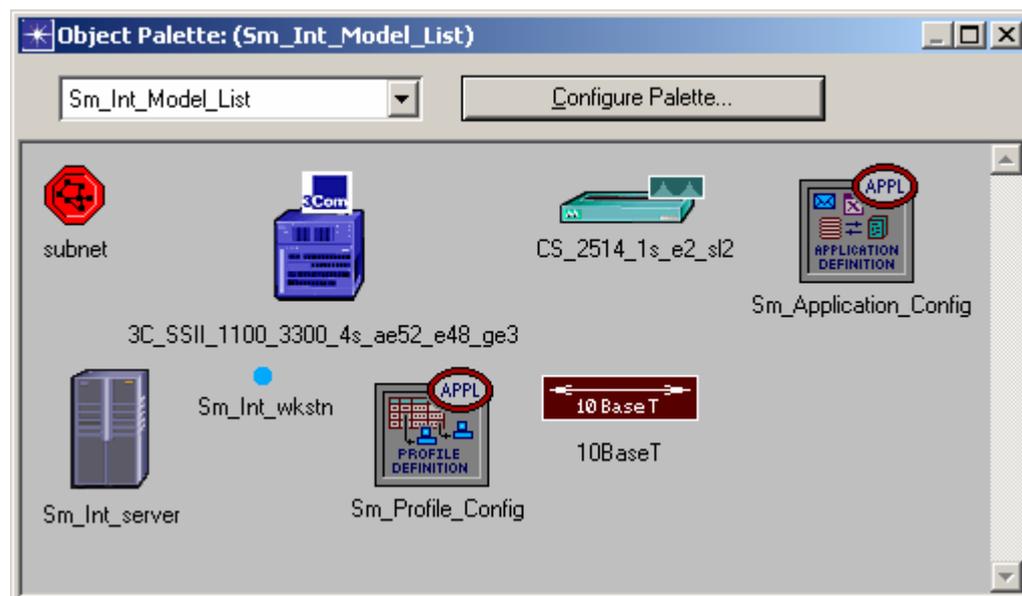
OPNET Modeler est un simulateur permettant de modéliser le fonctionnement d'un réseau pendant la phase de conception. Conçu et commercialisé par la société OPNET, OPNET Modeler est utilisé par de nombreuses entreprises d'ingénierie réseau dans le monde entier.

L'interface graphique et la modélisation orientée objet permettent de reproduire la structure réelle du réseau et de ses composants afin de coller à la réalité de façon intuitive. Pratiquement tous les types de réseau existants sont représentés. L'avantage du simulateur est de détecter les problèmes qui surviendront en exploitation dès la conception du réseau, et donc de pouvoir tester les différentes solutions permettant d'y remédier.

3.2. La boîte à outils d'OPNET

3.2.1. Les systèmes (devices)

OPNET Modeler fournit une librairie de modèles de matériels disponibles dans le commerce. Ces matériels peuvent être des routeurs, des commutateurs, des stations de travail, et des générateurs de trames.



exemple de matériels réseau proposés via une palette d'objets

3.2.2. Les protocoles

OPNET Modeler fournit des modèles de protocoles des différentes couches ISO, telles que http, TCP, IP, Ethernet, ATM, 802.11, etc...

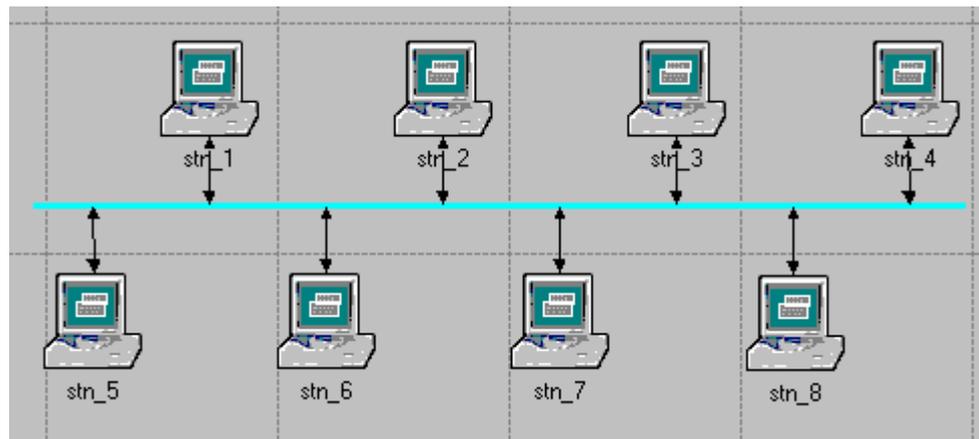
3.3. Conception du réseau : La logique d'OPNET

La modélisation du réseau se construit de façon hiérarchique. Il existe 4 niveaux hiérarchiques (level) dans OPNET (du plus général au plus restreint).

- Project
- Network
- Node
- Process

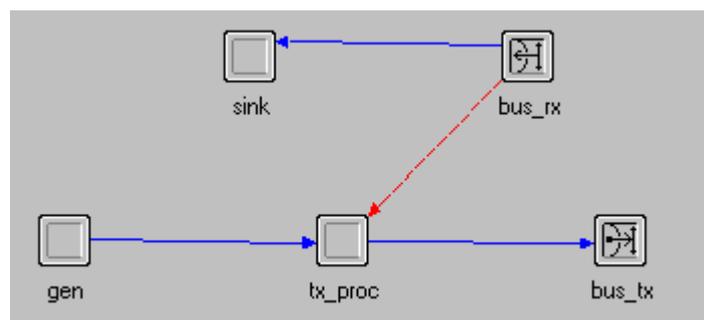
Project level : c'est le niveau de conception le plus élevé. On peut regrouper plusieurs Network Models dans un Project. Cela peut être intéressant si on souhaite comparer les avantages et inconvénients de deux solutions d'ingénierie distinctes.

Network level : c'est à ce niveau qu'est représenté le réseau dans son ensemble. Un réseau étant vu comme un ensemble de nœuds (nodes) éventuellement reliés par des liens (bus). Eventuellement, parce que dans la présente étude sur l'UMTS un seul node a été utilisé.



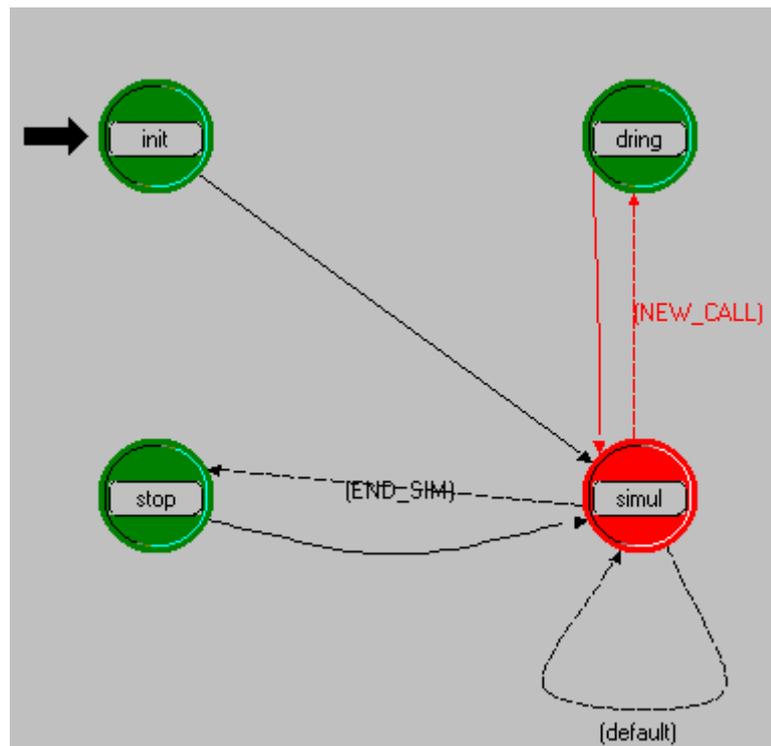
Un réseau Ethernet modélisé sous OPNET Modeler

Node level : le node level montre l'organisation des différentes machines à état et la façon dont ils communiquent via des bus. Le Node Level permet de mettre en œuvre divers objets de type « générateur de paquets », « Queue », « émetteur point à point », « bus », etc... On peut concevoir ses propres objets au niveau Process level.



Exemple de node model sous OPNET Modeler

Process level : Un processus, ou process, est représenté comme une machine à états. Chaque état peut être dans l'état ouvert (couleur verte) ou fermé (couleur rouge). L'entrée dans un état ouvert est immédiatement et automatiquement suivi de la sortie de cet état. Par contre on ne sort d'un état fermé que lorsqu'il advient un événement. Un événement (event) provoque le passage d'un état à un autre.



représentation d'un "process" sous forme de machines à états finis

3.4. Programmer le comportement des processus

Le process de simulation est appelé Simulation Kernel. Il fait appel aux Kernel Procedures.

Les process utilisateurs définis dans le process model font appel aux Kernel Procedures par l'intermédiaire du langage C ou C++. Ces procédures permettent de définir le comportement des process, de définir selon quelles lois statistiques les paquets sont émis dans le réseau et de mettre à jour des statistiques.

OPNET Modeler ne comporte pas de compilateur mais utilise celui de Microsoft Visual Studio.

3.5. Collecte des résultats

Les données statistiques sont stockées dans des fichiers particuliers appelés scalar files. Un scalar file comporte un ensemble d'attributs représentant les résultats statistiques qui nous intéressent. Les attributs possèdent une étiquette à laquelle on fait référence lors de l'appel à la primitive d'écriture dans le scalar file, *op_stat_scalar_write()*. Lors du paramétrage de la simulation, il faut préciser sur quel scalar file les données statistiques seront stockées.

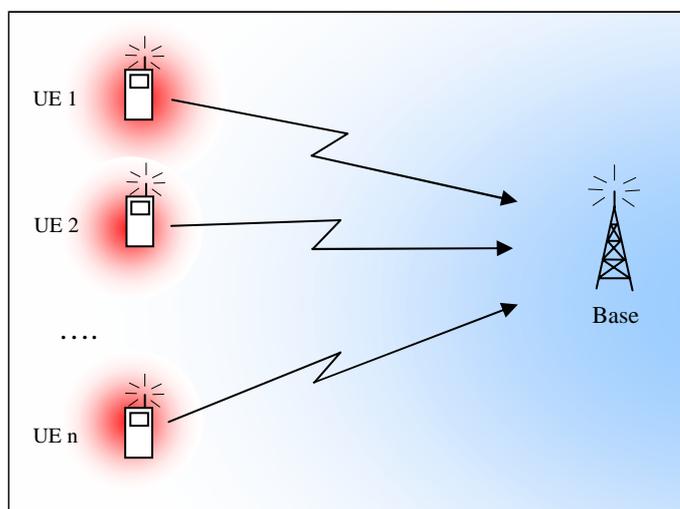
Précisons en passant qu'il est possible de faire exécuter plusieurs scénarii dans une simulation et qu'on associe un scalar file par scénario. Autrement dit, une simulation peut générer plusieurs scalar files. De plus, on peut enrichir les scalar files au fur et à mesure que les scénarii sont exécutés.

3.6. Présentation des résultats

Pour visualiser les résultats, on ouvre un scalar file et on choisit quels attributs sont analysés et leur position dans le graphe, abscisse ou ordonnée. De plus, les données peuvent être exportées pour être utilisées dans un tableur comme Excel.

4.La modélisation

La modélisation va consister à simuler la charge induite par plusieurs équipements terminaux (UE) sans fil qui tentent de se connecter sur une base (BTS) en s'appuyant sur deux protocoles d'accès distincts.



4.1. Généralités pour la modélisation

4.1.1.Variables promues

Ci-dessous la liste des valeurs de base qui nous permettent de paramétrer nos simulations.

Ces variables sont dites PROMOTED, ce qui veut dire que nous pouvons saisir leur valeur pour chaque simulation, et ainsi faire varier le contexte de simulation pour mieux comparer Slotted ALOHA à CPCH.

LIBELLE	DEFINIE DANS	DESCRIPTION
Cpch_flag	Queue	Permet la simulation en mode Slotted-ALOHA (0) ou en mode CPCH (1).
Interarrival_Time	Generateur	Valeur moyenne pour le tirage de l'inter temps entre l'émission de paquet.
Waiting_Time	Queue	Constante de temps avant de pouvoir réémettre un paquet.
Max_Retrans_Time	Queue	Borne supérieure de la loi uniforme utilisée pour le tirage du temps aléatoire avant réémission (s'ajoute au Waiting_Time).

4.1.2.Modélisation du temps

L'unité de base de temps simulation est le slot (0,577 μ s pour l'UMTS).

Dans la réalité la synchronisation entre la cellule et l'émetteur doit s'effectuer malgré des temps de propagation du signal qui peuvent varier suivant la distance qui sépare l'émetteur de l'antenne de la cellule.

OPNET Modeler garantit une parfaite synchronisation dans le timing des événements générés.

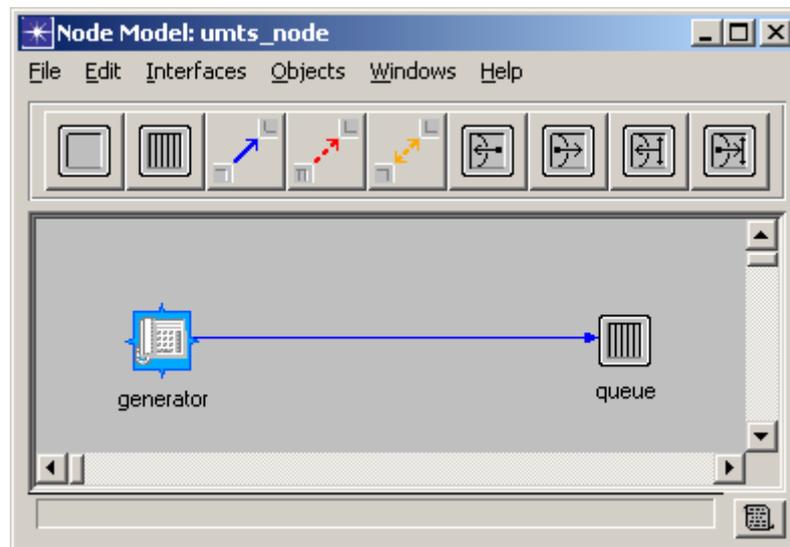
4.1.3. Modélisation des messages transmis (paquets)

Chaque paquet utilisé dans la simulation contiendra les renseignements suivants :

id (32 bits)
emission_time (32 bits)
type (32 bits)
failure (32 bits)
power (32 bits)
priority (32 bits)
size (32 bits)
generated_time (32 bits)

- **id** : identifiant séquentiel du paquet
- **emission time** : date de naissance du paquet : c'est un simple timestamp.
- **type** : préambule de puissance, préambule de collision ou données.
- **failure** : compteur d'échecs ayant débouché sur la réémission du paquet
- **power** : niveau de puissance d'émission du paquet.
- **priority** : niveau de priorité du paquet.
- **size** : taille du paquet.
- **generated time** : date de la dernière entrée en scène du paquet. Cela peut-être simplement sa date d'émission, ou si il a été réémis la date de sa dernière réémission.

4.2. Le Node model du projet



Le Node Model du projet UMTS

Le node « generator envoie les paquets qu'il produit au process « queue » via le bus de type « packet stream ».

4.3. Modélisation des émetteurs (clients)

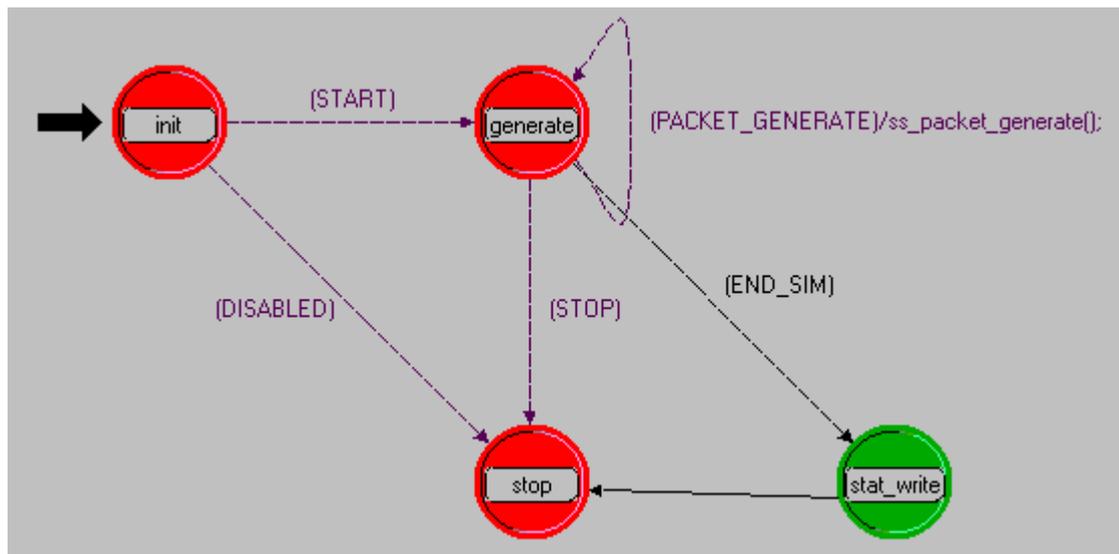
4.3.1. Automate du générateur

Pour garantir une comparaison équitable entre SLOTTED ALOHA et CPCH le générateur est le même pour les 2 types de base (même automate, mêmes séquences aléatoires de paquets envoyées aux bases).

Le logiciel de modélisation générant les événements de manière strictement séquentielle, il n'est pas nécessaire de gérer un processus par émetteur. C'est pourquoi nous ne disposerons que d'un processus pour gérer les n émetteurs. Nous pourrions faire varier le nombre de client, c'est à dire la charge réseau, en modifiant la constante INTERARRIVAL_TIME (promoted).

Le générateur émet des paquets suivant un intervalle de temps qui suit une loi de poisson.

A un instant donné, le nombre d'émetteurs qui émettent simultanément est tiré aléatoirement (loi exponentielle de raison 2).



Représentation sous forme de machines à états finis du node « generator »

4.3.2. Etat INIT

Événement	Traitement
Entrée dans l'état	Initialisation des distributions : <ul style="list-style-type: none"> • poisson pour l'inter arrivée. • uniforme pour le nombre de clients simultanés.

4.3.3. Etat GENERATE

Événement	Traitement
Entrée dans l'état	Armement d'une temporisation variable en nombre de slots (tirée selon une loi de poisson) chargée de générer l'événement PACKET_GENERATE .
Réception de PACKET_GENERATE	<p><u>Exécution de la fonction <i>ss_packet_generate</i> :</u></p> <ul style="list-style-type: none"> • Déterminer le nombre C de client (1..3) qui émettent dans le slot courant. Tirage d'une valeur uniforme entre 1 et 100 : <ul style="list-style-type: none"> ○ Entre 1 et 50 => 1 client. ○ Entre 51 et 80 => 2 clients. ○ Entre 81 et 100 => 3 clients. On simule ainsi des collisions à l'émission même des paquets, dans des proportions entièrement paramétrables. • Déterminer la priorité P du paquet dans le mode CPCH. Tirage d'une valeur uniforme entre 1 et 100 : <ul style="list-style-type: none"> ○ Entre 1 et 50 => Priorité faible. ○ Entre 51 et 100 => Priorité forte. • POUR (les C clients) FAIRE Créer un paquet p avec : <ul style="list-style-type: none"> p.Id=Compteur de paquet; p.failure=0; p.Type=PREAMBULE_POWER;

	<p>p.Priority=P p.Generated_Time = op_sim_time(); p.Emission_Time = op_sim_time(); Envoi du paquet. FINPOUR</p>
--	--

4.3.4.Etat STAT_WRITE

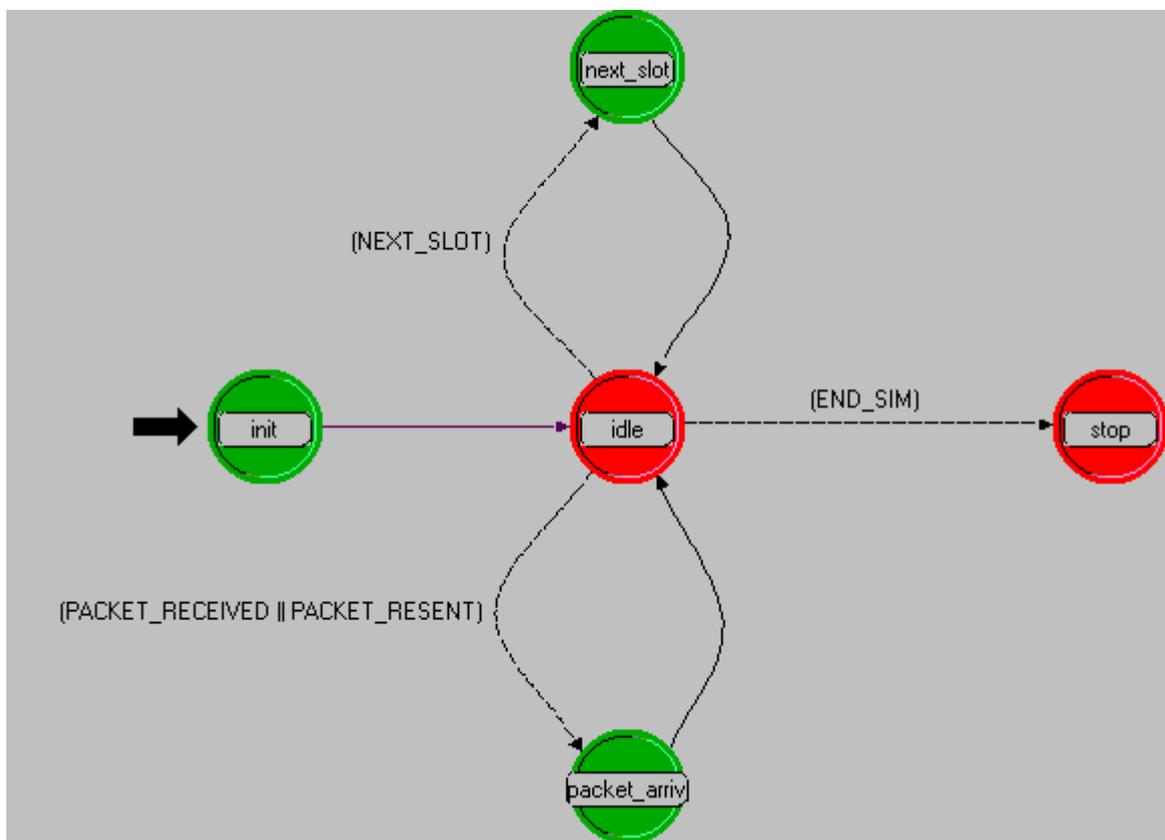
Evénement	de	Traitement
Réception END_SIM		Mise à jour des statistiques de l'émetteur : <ul style="list-style-type: none"> o Valeur de l'inter arrivée o Lambda (=1/ valeur de l'inter arrivée).

4.3.5.Etat STOP

Evénement	de	Traitement
Réception STOP		Restitution des ressources mémoires.

4.4. Modélisation de la base Slotted ALOHA

4.4.1. Automate de la cellule Slotted ALOHA



4.4.2. Etat INIT

Evénement	Traitement
Entrée dans l'état	Initialisation des distributions : <ul style="list-style-type: none"> • uniforme pour le temps de retransmission. Génération de l'événement NEXT_SLOT .

4.4.3. Etat IDLE

Evénement	Traitement
Entrée dans l'état.	Rien à faire....

Génération d'une temporisation (durée d'un slot) à « moi-même ». Sur réception de cette temporisation passage dans l'état Analyse.

4.4.4. Etat NEXT_SLOT

Evénement	Traitement
-----------	------------

<p>Réception de NEXT_SLOT</p>	<p>A la fin d'un slot s'il reste un paquet non mis dans la queue, on considère qu'il a été reçu correctement :</p> <ul style="list-style-type: none"> • <u>Paquet de type PREAMBULE_POWER</u> Si (Cpch_flag = ALOHA) Le paquet devient de type DATA. Tirage d'un temps de réémission aléatoire. Mise à jour du temps d'émission du paquet. Positionne le paquet dans la queue triée. Sinon Le paquet devient de type PREAMBULE_COLLISION. Tirage d'un temps de réémission aléatoire utilisé pour réarmer une temporisation chargée d'émettre un événement de type PACKET_RESENT. Mise à jour du temps d'émission du paquet. Positionne le paquet dans la queue triée. • <u>Paquet de type PREAMBULE_COLLISION (CPCH uniquement)</u> Le paquet devient de type DATA. Temps de réémission égal à 2 utilisé pour réarmer une temporisation chargée d'émettre un événement de type PACKET_RESENT. Mise à jour du temps d'émission du paquet. Positionne le paquet dans la queue triée. • <u>Paquet de type DATA</u> Le paquet est détruit et considéré comme reçu de manière correcte.. <p>Réarmement d'une temporisation chargée d'émettre un événement de type NEXT_SLOT. Passage dans l'état Idle.</p>
--------------------------------------	--

4.4.5. Etat PACKET_ARRIV

Evénement	Traitement
<p>Réception de PACKET_RECEIVED OU PACKET_RESENT</p>	<p>On distingue les trois types de paquets :</p> <ul style="list-style-type: none"> • <u>Paquet de type PREAMBULE_POWER</u> On calcule la différence de puissance du paquet entrant vis à vis du paquet courant. <i>Deux cas à considérer :</i> <ul style="list-style-type: none"> ○ DeltaPuissance ≥ 2 On garde comme paquet courant celui de plus forte puissance. Pour l'autre : <ul style="list-style-type: none"> • Dans la mesure où il n'a pas atteint la puissance maximale, on augmente sa puissance d'un pas. • On lui attribue un temps aléatoire de réémission utilisé pour réarmer une temporisation chargée d'émettre un événement de type PACKET_RESENT. • On le remet dans la queue triée. ○ DeltaPuissance ≤ 2 <ul style="list-style-type: none"> • Dans la mesure où chacun des paquets n'a pas atteint la puissance maximale, on augmente sa puissance d'un pas. • On lui attribue un temps aléatoire de réémission utilisé pour réarmer une temporisation chargée d'émettre un événement de type PACKET_RESENT. • On le remet dans la queue triée. <p>Si (On simule CPCH) Le paquet devient de type PREAMBULE_COLLISION. Tirage d'un temps de réémission aléatoire utilisé pour réarmer une temporisation chargée d'émettre un événement de type PACKET_RESENT. Mise à jour du temps d'émission du paquet. Positionne le paquet dans la queue triée.</p> <p>Sinon Le paquet devient de type DATA. Tirage d'un temps de réémission aléatoire utilisé pour réarmer une temporisation chargée d'émettre un événement de type PACKET_RESENT. Mise à jour du temps d'émission du paquet. Positionne le paquet dans la queue triée.</p> <ul style="list-style-type: none"> • <u>Paquet de type PREAMBULE_COLLISION (CPCH uniquement)</u> Le paquet devient de type DATA. Temps de réémission égal à 2, utilisé pour réarmer une temporisation chargée d'émettre un événement de type PACKET_RESENT. Mise à jour du temps d'émission du paquet. Positionne le paquet dans la queue triée. • <u>Paquet de type DATA</u> Le paquet est détruit et considéré comme reçu de manière correcte.

Passage dans l'état Idle.

4.4.6. Etat STOP

Cet état est atteint sur arrêt de la simulation.

Veille à libérer les ressources mémoire.

NB : la Queue triée consiste à toujours avoir en tête le paquet qui va être réémis en premier.

4.5. Modélisation du protocole CPCH

Voir modélisation slotted-ALOHA (avec variable promoted `Cpch_flag = 1`).

5. Simulations

5.1. Scénarii de test

Huit scénarii ont été utilisés pour tester chacun des protocoles. Voici le détail du paramétrage pour chacun d'entre eux :

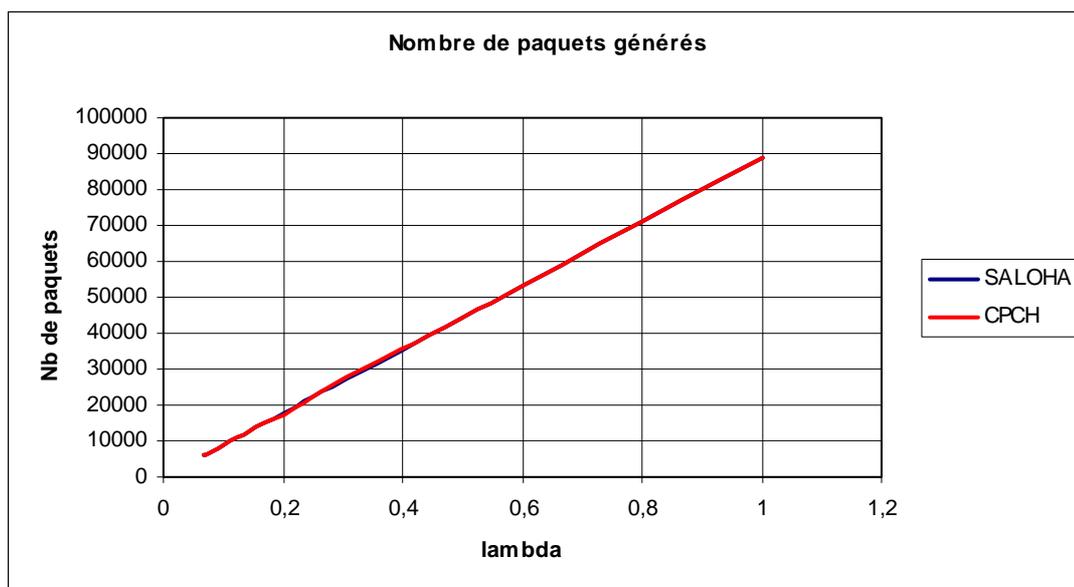
Scenario	Interarrival_Time	Max_Retrans_Time	Waiting_Time
0	1..16	4	4
1	1..16	8	8
2	1..16	12	12
3	1..16	16	16
4	1..16	4	16
5	1..16	16	4
6	1..16	8	12
7	1..16	12	8

Ainsi, nous disposerons de 8 simulations différentes pour chaque variable observée.

5.2. Statistiques collectées

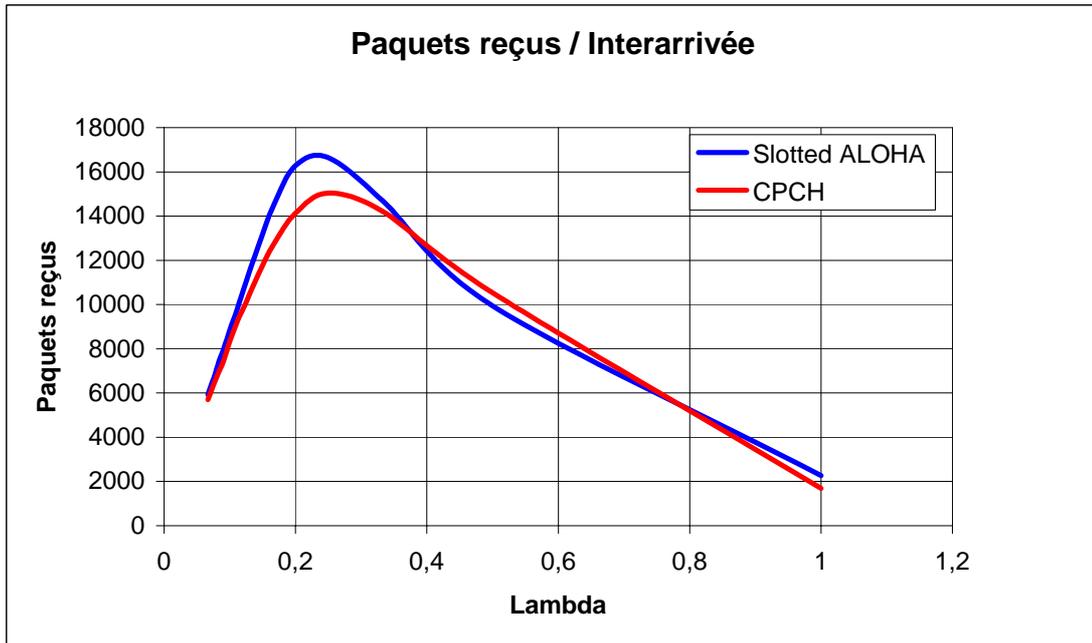
Au cours de l'étude, nous avons observé les paramètres suivants pour comparer les deux protocoles d'accès :

5.2.1. Pourcentage de paquets reçus / paquets générés



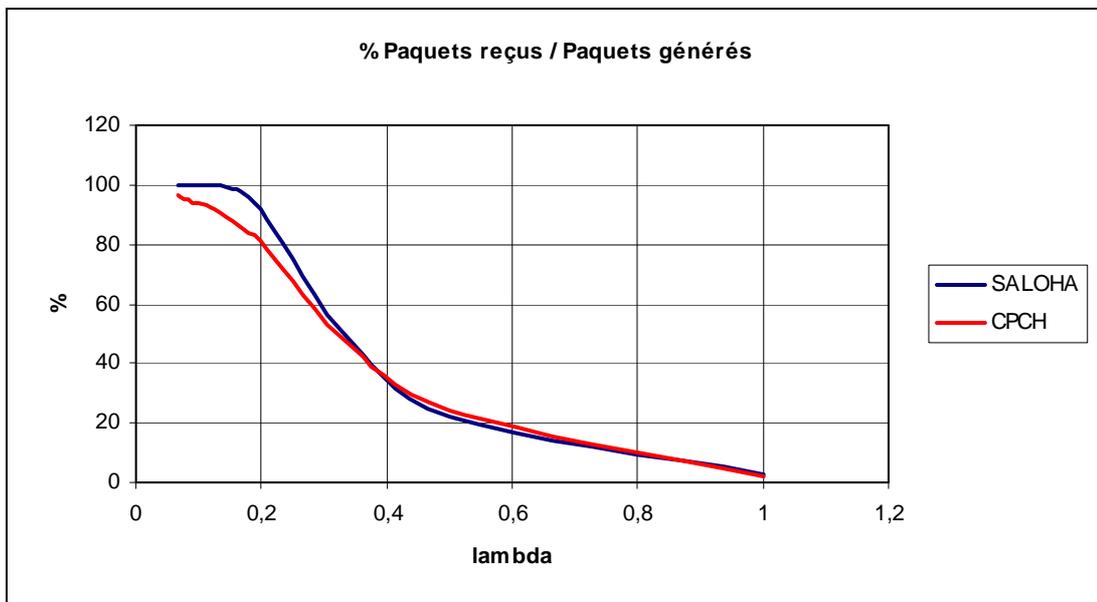
Cette courbe montre que le volume de paquets entre les deux protocoles est comparable. Il informe également sur l'ordre de grandeur de paquets générés en fonction de l'inter arrivée.

5.2.2. Paquets reçus suivant l'inter arrivée



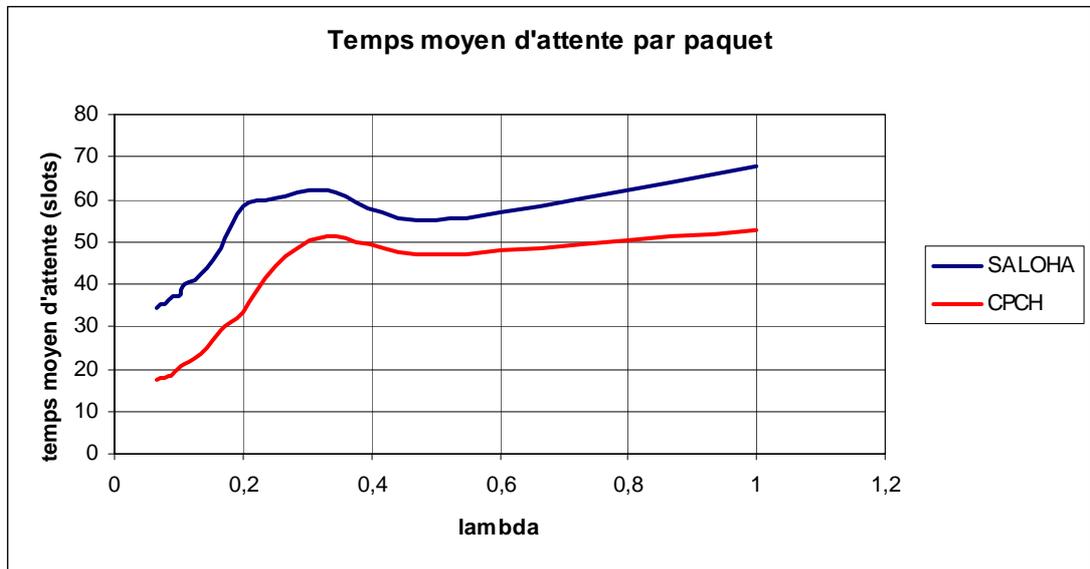
5.2.3. Pourcentage de paquets reçus / paquets générés

$$\% \text{ de paquets reçus} = \frac{NbPaquetsReçus}{NbPaquetsGénérés}$$



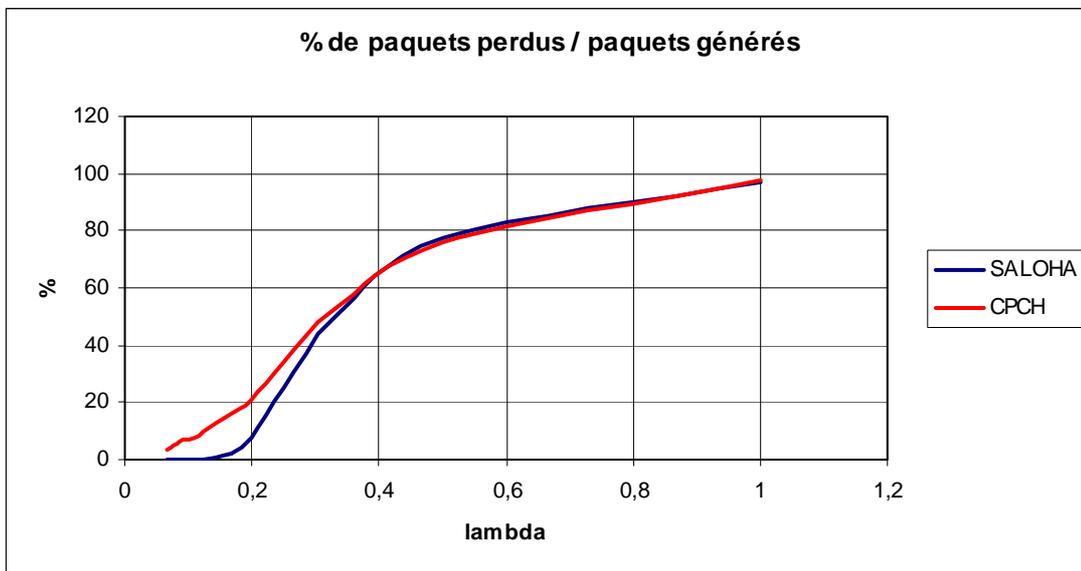
5.2.4. Temps d'attente moyen par paquet

$$\text{Temps d'attente moyen par paquet} = \sum_{\text{PaquetsArrivés}} \left(\frac{\text{PaquetSlotArrivée} - \text{PaquetSlotGénééré}}{\text{NbPaquetsArrivés}} \right)$$



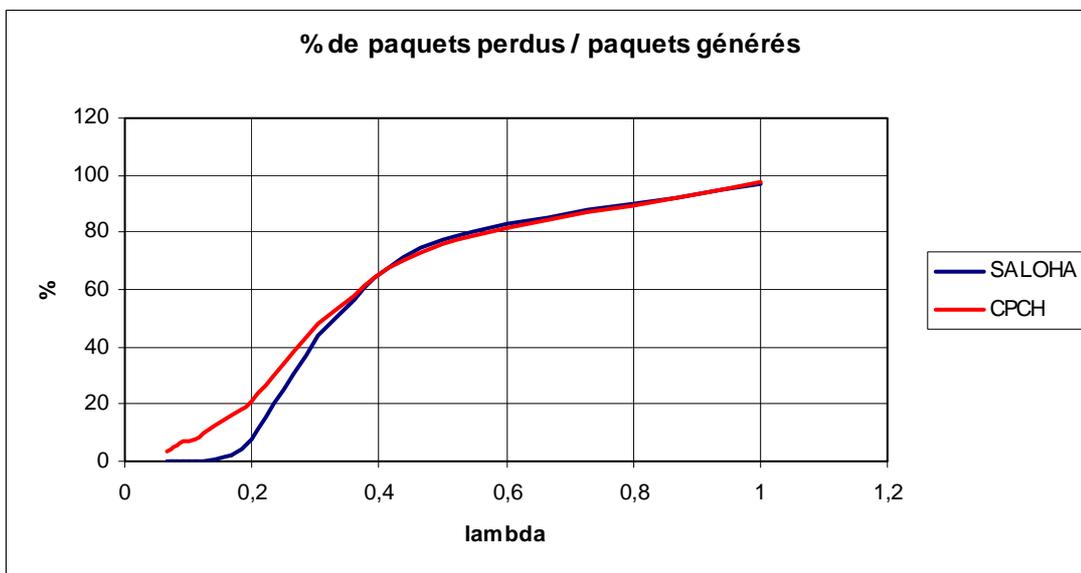
5.2.5. Nombre moyen de préambules par paquet

$$\text{Nb moyen de préambule par paquet} = \frac{\text{NbTotalPréambules}}{\text{NbPaquetsGénérés}}$$

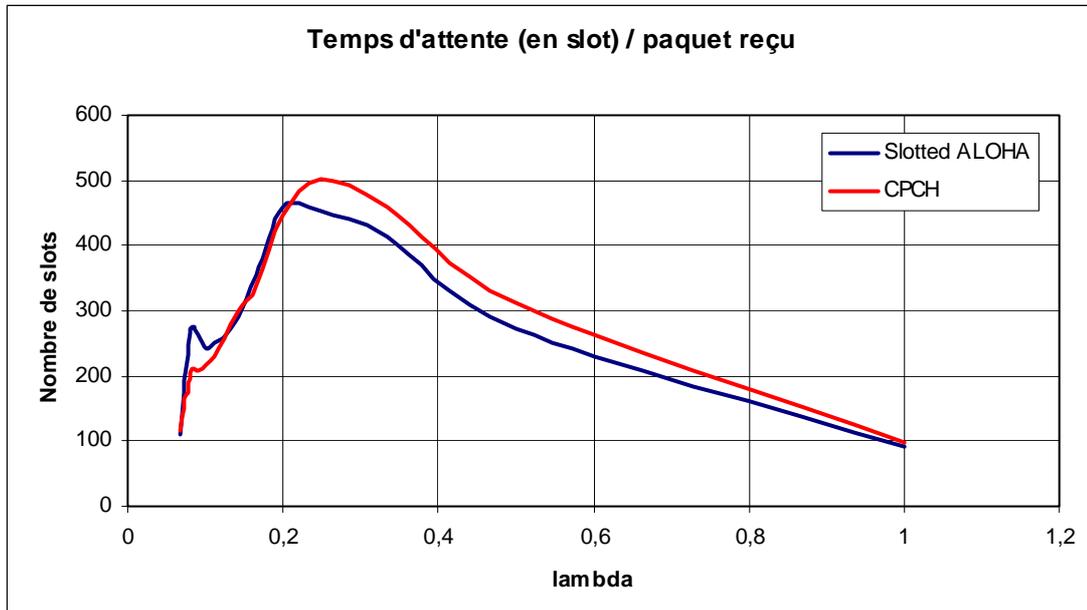


5.2.6. Pourcentage de paquets perdus / paquets générés

$$\% \text{ de paquets perdus} = \frac{\text{NbPaquetsPerdus}}{\text{NbPaquetsGénérés}}$$



5.2.7. Temps d'attente moyen par paquet



6. Conclusion

Les différences entre slotted-ALOHA et CPCH ne sont pas aussi marquées que nous aurions pu l'attendre.

CPCH procure un réel avantage en ce qui concerne les délais de transmission. Ceci s'explique par le fait que les trames de préambule de puissance en collision sont détruites, libérant ainsi de la ressource réseau, tandis que dans ALOHA les trames de données sont réémises.

CPCH semble légèrement moins performant lorsque le réseau est chargé (voir 5.2.2, 5.2.4 et 5.2.5). comme les paquets sont détruits, il a moins de paquets émis avec succès.

La simulation ne donne aucune indication en ce qui concerne les débits. Or a priori, dans ce cas de figure, CPCH devrait être sensiblement meilleur. En effet, les trames de données sont nettement plus importantes en termes de volume que l'ensemble des préambule. ALOHA peut être amené à réémettre des trames de données tandis que CPCH garantit que les trames de données transmises le seront sans collision.

7. Glossaire

AAL2 : ATM Adaptation Layer de type 2

ALOHA : « bonjour » en Hawaïen

ATM : Asynchronous Transfer Mode

BI-BOP : Base Transceiver Station

BTS : Base Transceiver Station

CDMA : Code Division Multiple Access

CPCH : Common Packet Channel

FDMA : Frequency Division Multiple Access

GSM : Global System for Mobile communication

OPNET : Logiciel de simulation de réseau

Slotted ALOHA : Aloha en temps discrétisé

TDMA : Time Division Multiple Access

UE: User Equipment

UMTS : Universal Mobile Telecommunication System

W-CDMA : Wideband-CDMA, même méthode sur une bande plus large (UMTS)