

Présentation d'un Outil de Simulation Orienté Objet Des Réseaux, NS2

Didi Fedoua
Faculté des Sciences de l'Ingénieur
Département d'Informatique
Université Abou Bekr Belkaid de Tlemcen
BP 230 Tlemcen
Tel : 213 43 28 56 86/89
Fax : 213 43 28 56 85
E-mail : f.dididi@mail-univ.tlemcen
fedouadidi@yahoo.fr

Résumé

Le réseau de télécommunications multimédia est un outil nécessaire au développement des PME / PMI que ce soit pour mettre les ressources en réseau et bénéficier du partage des ressources comme l'imprimante, le scanner, les bases de données, les fichiers, ..., et/ou accéder au réseau INTERNET et spécialement au web qui est un outil indispensable pour la bonne marche des PME. Se faire connaître (en créant un site web), vendre et acheter (le business), ou tout bêtement pour faire des opérations bancaires sans avoir à se déplacer et faire de la gestion efficace de n'importe quel endroit grâce aux réseaux sans fil, sont autant de choses nécessaires aux PME.

Cependant, il est important et judicieux de simuler le résultat de l'investissement d'un réseau d'entreprise et un accès à INTERNET, avant son déploiement, en raison de l'évolution continue et rapide de nouveaux systèmes de communication (réseaux locaux, réseaux industriels, Internet, réseaux sans fil, mobilité ...). Dans cet article, nous présentons un outil de simulation, se composant de trois parties, la première donne des notions générales sur l'évaluation des performances, la deuxième délivre une brève présentation des modules du Network Simulator NS, et la dernière partie donne quelques exemples d'utilisation de NS et nous finissons par une conclusion.

Mots clés : *simulation, réseau informatique, performance, topologie, qualité de service, charge, débit, ...*

Introduction

Face à la complexité des réseaux de communication, la simulation a été et reste toujours l'outil privilégié pour évaluer les performances des réseaux et pour étudier leur comportement. Les différents composants du réseau (source, lien, noeud, protocole, ...), sont modélisés en termes d'instructions qui sont interprétées par une entité de coordination, le simulateur. Devant la complexité des problèmes concernés et la grande diversité des architectures rencontrées, de nombreuses solutions sont envisageables pour un même problème. Chacune des solutions envisagées doit alors être testée, évaluée, caractérisée et critiquée avant d'envisager son implémentation concrète dans un réseau. Il faudrait alors étudier le comportement du système afin de comprendre et de régler les éventuels problèmes qui pourrait affecter le système. Cependant, pour réaliser l'évaluation de performance d'un système il faut disposer d'outils adéquats. Il s'agit de représenter le système et ses paramètres d'entrée à l'aide d'un programme adéquat. Chose qui peut se faire avec un programme de simulation des réseaux nommé NS2 (Network simulator 2). Cet outil peut représenter aussi bien des petits que de grands réseaux, avec beaucoup de vraisemblance, supportant tous les protocoles d'INTERNET et toutes ses couches.

La simulation nous permet d'effectuer ces tests à moindres coûts.

La simulation permet, en principe, d'étudier les modèles de réseau avec n'importe quel niveau de détail. On peut facilement modifier les conditions de fonctionnement du réseau et comparer les mesures d'intérêt d'un scénario à l'autre. Pour un utilisateur, la simulation a plus de crédibilité que les modèles analytiques car elle est plus proche du système réel et nécessite moins de simplifications et quasiment pas d'hypothèse spécifique. Il existe plusieurs simulateurs

de réseau dont OPNET et QNAP,... mais on pense que NS allie beaucoup d'avantages à peu d'inconvénients. On procède généralement à la simulation pour évaluer un nouveau système ou bien pour des raisons de coûts d'évaluation par des mesures réelles. La simulation permet en plus de visualiser les résultats sous forme de graphes faciles à analyser et à interpréter. Si on néglige l'étape de l'évaluation et on passe directement à l'implémentation réelle et la mise en oeuvre du système, les coûts peuvent être sévères en cas de non-conformité du système avec les exigences temporelles requises pour les applications.

1. Evaluation des Performances [1]

Pour réaliser une évaluation de performance on doit disposer de deux éléments :

- ❖ **Un système** : c'est l'objet pour lequel on estime le comportement.
En général un système informatique est considéré comme étant un ensemble de ressources partagées entre différentes tâches. Et comme paramètre du système on peut saisir la présence de temps d'attente pour l'accès à ces ressources partagées.
- ❖ **La charge** : (*workload* en anglais). La charge du système représente généralement le trafic en entrée, qui pourrait être l'ensemble de messages servis par une machine du réseau, le nombre de tâches à exécuter par un processeur.

La charge est habituellement exprimée par des lois probabilistes (Poisson, Exponentielle, ...).

La performance étudiée peut prendre plusieurs formes selon la nature du système à étudier.

On peut s'intéresser par exemple au : Débit nominal si on évalue les performances d'un réseau sans fil, au temps de réponse pour un système temps réel

1.1. Eléments de base

Pour évaluer le comportement d'un système informatique, on doit retenir les points suivants :

- **Définir votre objectif** : Dimensionnement d'un réseau, étudier le comportement d'un serveur en cas de surcharge, comparer deux protocoles d'accès pour le réseau sans fil ...
- **Identifier les facteurs** : En déterminant les facteurs qui agissent sur le Système (paramètres d'entrée)
- **Définir vos métriques** :
 - Déterminer les paramètres de performances
 - Déterminer la chose à évaluer.
 - Déterminer le débit effectif

2. Network Simulator [2, 3,4]

NS-2, Network Simulator, est aujourd'hui le simulateur de réseau probablement le plus utilisé par la communauté scientifique des réseaux. Il s'agit d'un simulateur à événements discrets, fruit de la collaboration entre l'université.

NS est un outil de recherche très utile pour la conception et la compréhension des protocoles. Il est orienté objets. Il est devenu aujourd'hui un standard de référence en ce domaine. C'est un logiciel dans le domaine public disponible sur Internet. Son utilisation est gratuite. Le logiciel est exécutable tant sous Unix que sous Windows. NS-2 permet à l'utilisateur de définir un réseau et de simuler les communications entre les noeuds. Le simulateur utilise le langage orienté objet OTCL dérivé de TCL pour la description des conditions de simulation sous forme de script. Dans le script l'utilisateur fournit la topologie du réseau, les caractéristiques des liens physiques, les protocoles utilisés, le type de trafic généré par les sources, les événements, les communications qui ont lieu etc. La figure suivante montre les couches de NS par rapport au modèle OSI

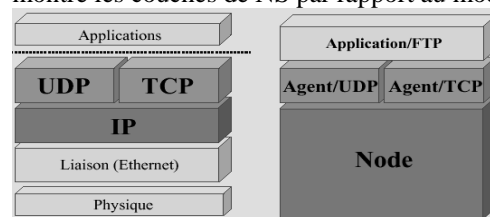


Figure 2.1 Les couches de NS

Si le script écrit en OTCL permet une utilisation (édition, modification des simulations) facilitée du simulateur, les routines sont elles écrites en C++ pour avoir une plus grande puissance de calcul. Un grand nombre de classes sont prédéfinies et mettent en oeuvre plusieurs types de protocoles, de files d'attente, de sources et algorithmes de routage.

Les résultats de la simulation sont sauvegardés dans des fichiers trace qu'on peut interpréter avec des outils comme xgraph ou tracegraph ou gnuplot.

Par ailleurs, le simulateur permet la création d'un fichier d'animation (d'extension .nam), permettant de visualiser la simulation sur l'interface graphique NAM.

L'utilisation de ces outils et l'exploitation des résultats requièrent certaines compétences:

- Connaissance du C++ et maîtrise des concepts de la programmation orientée objet pour l'édition des routines qui composent le simulateur.
- Connaissance du langage script OTCL pour la description des simulations et le prétraitement des données.
- Maîtrise de l'outil indispensable à l'extraction des données des fichiers.

- Capacité à faire des calculs et à produire des graphes à partir des fichiers produits, à l'aide de **Matlab** par exemple.

Le simulateur est conçu principalement pour le monde de l'Internet. Il permet de simuler le comportement des protocoles standard de l'Internet tels que TCP, IP et permet d'étendre le simulateur aux nouveaux protocoles de l'Internet (routage, transport, application, mobilité, le sans fil) et aux nouvelles architectures de qualité de service (IntServ, DiffServ, MPLS, RED).

Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage unipoint ou multipoint, des protocoles de transport, de session, de réservation, des services intégrés, des protocoles d'application comme HTTP. De plus le simulateur possède déjà une palette de systèmes de transmission (couche 1 de l'architecture TCP/IP), d'Ordonnanceurs et de politiques de gestion de files d'attente pour effectuer des études de contrôle de congestion.

La liste des principaux composants actuellement disponibles dans NS par catégorie est:

Application	Web, ftp, telnet, générateur de trafic (CBR, ...)
Transport	TCP, UDP, RTP, SRM
Routage	Statique, dynamique (vecteur distance) et routage multipoint (DVMRP, PIM)
Gestion de file d'attente	RED, DropTail, Token bucket
Discipline de service	CBQ, SFQ, DRR, Fair queueing
Système de transmission	CSMA/CD, CSMA/CA, lien point à point

Tableau 2.1 Liste des composants disponibles dans NS

2.1. Pourquoi deux langages?

Si NS utilise deux langages c'est parce qu'il est capable de procéder de deux manières. D'une part, le simulateur doit être efficace dans la manipulation de bytes, de paquets et d'en-têtes et dans l'application d'algorithmes sur de grandes quantités de données. Dans ce cas, l'utilisation du C++ s'impose. D'autre part, pour l'utilisateur souhaitant changer rapidement ses scénarios de simulation, OTCL offre une bonne solution. L'inconvénient de cette dualité est le dédoublement des objets et l'implémentation des fonctions qui doit veiller à la juste interaction entre les deux parties. Pour que l'utilisateur puisse suivre l'évolution d'une variable elle doit exister à la fois en C++ et en OTCL (liaisons de variables, voir la figure suivante).

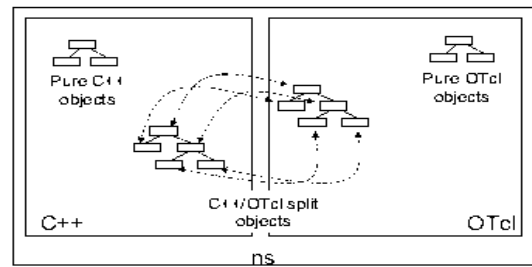


Figure.2.2 La dualité entre OTCL et C++

La création d'une nouvelle classe suppose bien évidemment que l'on respecte cette dualité (Figure.2.2).

2.2. Architecture et composant du réseau [3]

Dans Cette section nous présentons les composants de NS, les classes de base utilisables pour définir l'architecture et la topologie du modèle. Les classes Node et Link servent à la composition de la topologie du réseau.

Elles modélisent les noeuds et les arcs d'un graphe. La figure suivante présente la hiérarchie des classes OTCL de NS. Cette présentation permet de mieux comprendre les différentes composantes d'un réseau dans NS.

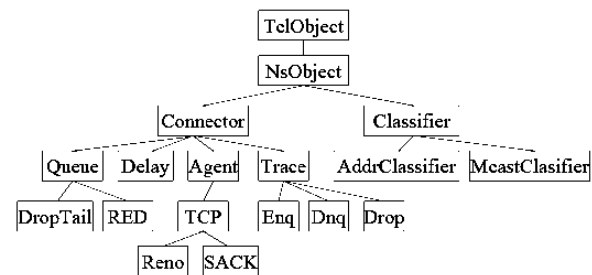


Figure 2.3 Les classes OTCL de NS

La racine de la hiérarchie est la classe TclObject qui est à la base de tous les objets OTCL de la librairie de NS (planificateur, composants réseau, objets NAM). La classe NsObject est une descendante de la classe TclObject et la base de tous les objets qui manipulent les paquets (noeuds, liaisons, files). Cette classe est encore subdivisée en deux sous-classes Connector et Classifier. Les objets offrant une seule sortie possible pour les paquets se trouvent dans la classe Connector et les autres dans la classe Classifier.

2.3. Les Agents [4,5]

On peut assimiler les agents de NS à la couche transport du modèle OSI.

La classe agent fournit des méthodes utiles au développement de la couche transport et à d'autres protocoles du plan de signalisation ou de gestion. Cette classe est à la fois dans l'interpréteur et dans le simulateur. C'est la classe de base pour définir des nouveaux protocoles dans NS. Ceux sont les agents

qui construisent les fonctions pour générer ou détruire les paquets. Ceux sont les points terminaux du réseau qui reçoivent ou délivrent les paquets des applications. Elle fournit l'adresse locale et de destination, l'interface à la classe Application. A chaque agent est attribué un port. L'adresse d'un agent se compose du numéro de son noeud et de son port.

Actuellement NS comporte de nombreux agents citons: UDP, protocoles de routage, différentes versions de TCP, RTP, etc.

Ns offre la possibilité de ramasser un certain quantième de données statistiques sur le développement de la simulation et de les conserver dans un fichier "trace". Le code des fonctions de trace est respectivement dans `trace.cc` et `tclcl/tracedvar.cc`. Le fichier de trace est un fichier texte structuré en lignes.

Ce fichier saisira tous les événements arrivés lors de la simulation.

En traitant ce fichier de très grande taille on pourra en extraire l'information souhaitée: calcul du débit aux noeuds, évaluation des pertes, etc. Voici un extrait d'un tel fichier.

1	2	3	4	5	6	7	8	9	10	11	12
+	8.5	0	1	Tcp	1000	-	0	2	3	4562	102
-	8.5	0	1	ack	40	-	1	2	3	4586	103
D	8.8	2	3	Tcp	1000	-	1	2	3	4596	99
R	8.9	2	3	tcp	1000	-	1	2	3	4596	253

Tableau 2.2 Fichier Trace obtenu après simulation

Chacune des lignes correspond à un événement survenu à un paquet. Voici le détail de ce que contiennent ces colonnes. En commençant par la colonne de gauche.

1. Action effectuée sur le paquet. Un "+" signifie que le paquet est reçu dans une file, un "-" signifie que le paquet quitte la file, un "d" signifie que le paquet est jeté et un "r" signifie que le paquet est réceptionné par un agent.
2. Instant où l'action est effectuée.
3. Noeud de départ du lien concerné.
4. Noeud d'arrivée du lien concerné.
5. Type de paquet.
6. Taille du paquet en bytes.
7. Flags.
8. Identificateur.
9. Agent de départ.
10. Agent d'arrivée.
11. Numéro de séquence.
12. Identificateur unique pour chaque paquet.

2.4. Interprétation de la Simulation [4]

2.4.1. Le Simulateur

La simulation est formée, vérifiée et utilisée par l'interface approvisionnée par la classe OTcl Simulator. Cette classe joue le rôle d'API. Elle ne se trouve que dans l'interpréteur. Un script de simulation débute toujours par créer une instance de cette classe par la commande:

```
set ns_ [new Simulator]
```

2.4.2. Ordonnanceur

Il est déterminé dans le fichier `scheduler.{h,cc}`. L'Ordonnanceur a le rôle de favoriser l'événement le plus proche en terme de temps, de faire les traitements, de faire améliorer le temps simulé et d'accélérer l'événement suivant etc. Les événements sont disposés dans un échéancier. Les événements se traite un par un. Si certains événements nécessitent un traitement simultané, ils sont exécutés ensuite mais au même instant en terme de temps simulé. On parle dans ce cas du quasi parallélisme

2.5. Le Nam (Network Animator) [4]

Nam est un outil d'animation basé sur Tcl/TK permettant de visualiser le mouvement des paquets.

La première étape pour employer le *nam* est de produire le dossier `trace`. Le dossier `trace` devrait contenir l'information de topologie, par exemple les noeuds, les liens, et des traces de passage des paquets.

3. Quelques exemples de simulation

NS a été utilisé pour simuler toute sorte de réseaux, en particulier les réseaux sans fil WIFI du standard 802.11 qui connaissent une expansion sans précédent et sont de très bon réseaux pour les PME qui veulent se doter d'un réseau informatique ou étendre leur réseau local, pour un coût très abordable et d'une simplicité évidente. On va présenter deux exemples de simulation avec quelques résultats sous forme de graphes. Dans le premier, on simule le réseau WIFI de notre laboratoire, dans le but d'étudier ses performances, dans le deuxième on compare les performances du standard 802.11 sans qualité de service, avec le standard 802.11 e qui lui intègre la qos. La topologie de notre réseau local sans fil est la suivante : 13 stations numérotées de 0 à 12, dont deux points d'accès reliés entre eux par le système de distribution (les stations 6 et 7 respectivement AP1 et AP2 représentés par des carrés sur la figure obtenue par le NAM). Les stations mobiles sont numérotées de 8 à 12 et représentées par les grands cercles, quant aux stations fixes numérotées de 0 à 5, elles sont représentées par les petits cercles et reliées aux points

d'accès par des liaisons filaires. Les flux utilisés sont des datas.

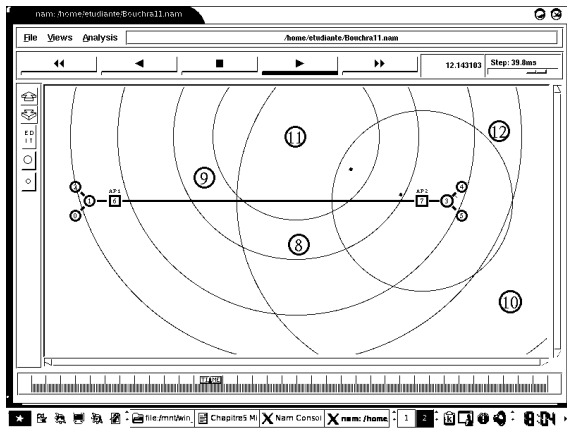


Figure 3.1 Topologie du LAN simulé

Les communications entre stations sont résumées comme suit :

De station s à d	Déb. de tran	Fin de tran	Type de tran	Route
1 à 8	0.02	4.12	TCP	AP1
4 à 11	5.15	20.15	UDP	AP2 puis AP1 à 12 .45s
5 à 12	14.10	35	UDP	AP2
3 à 9	21	35	UDP	AP1

Tableau 3.1 Le trafic généré par la simulation

Nous avons jugé utile de schématiser les métriques suivantes: le nombre total de paquets rejetés, le débit effectif de transmission des paquets et le temps de transmission de bout en bout des données. Les résultats sont donnés par les trois figures qui suivent :

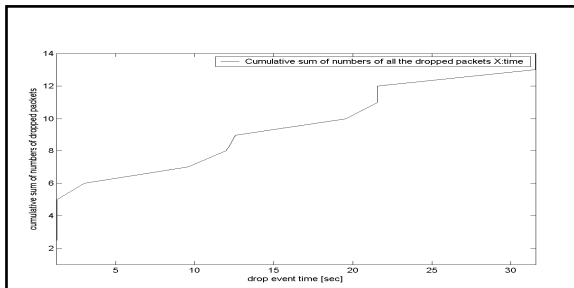


Figure 3.2 Somme cumulative des paquets rejetés

Nous remarquons d'après la figure 3.2 que le nombre de paquets rejetés par les AP augmente de façon continue à partir du moment où deux communications concourent pour l'accès au canal. Chose qui nous indique que si les données étaient de type sensible, ce serait préjudiciable et donc aucune qualité de service n'est supportée.

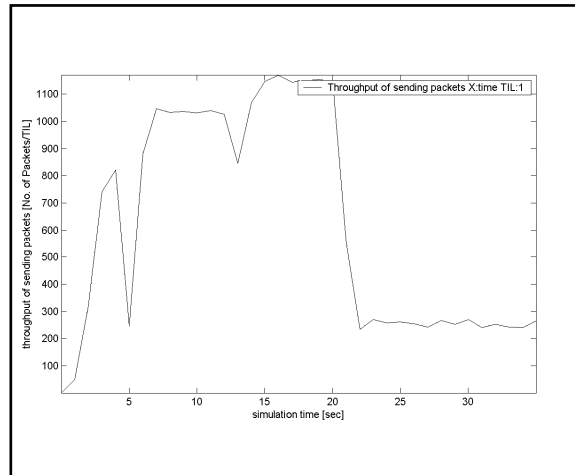


Figure 3.3 Débit effectif des paquets

La figure 3.3 montre un débit qui augmente avec le début des transmissions, se stabilise jusqu'au début d'une deuxième communication qui va entraîner une chute de débit à plus de la moitié, puis retrouve une certaine stabilité dès la fin d'une des deux communications.

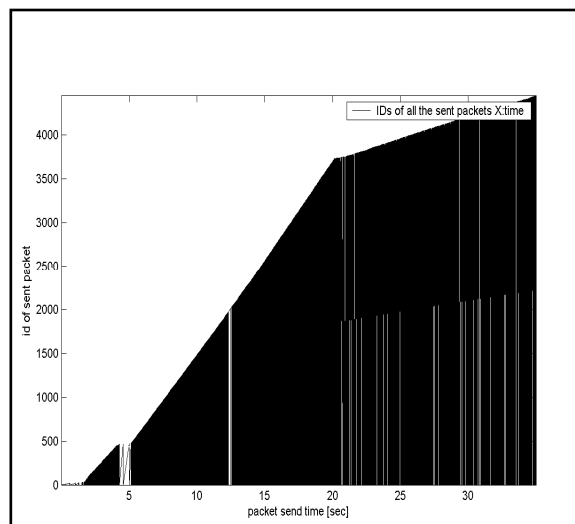


Figure 3.4 Identificateurs des paquets Envoyés

La figure 3.4 représente les numéros des paquets émis pendant le temps de transmission, numéros qui augmentent d'abord très rapidement puis presque se stabilise quant il ne reste qu'une seule communication en cours, chose qui est tout à fait normal.

Dans le deuxième exemple, on compare le protocole DCF du standard 802.11 au protocole EDCF avec qos du standard 802.11 e. On a adopté un réseau à sept nœuds dont un point d'accès portant le numéro zéro et trois types de flux sont générés (audio, vidéo et data). Et les graphes obtenus sont les suivants :

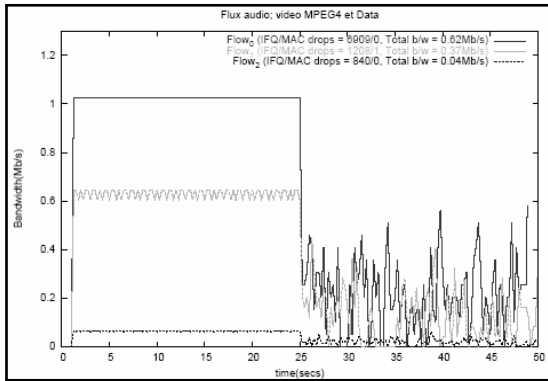


Figure 3.5 DCF répartition de la bande passante entre les trois flux audio, vidéo et data dans le temps

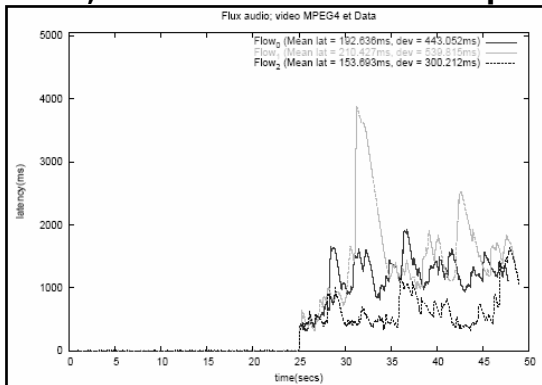


Figure 3.6 DCF latence des flux audio, vidéo et data dans le temps

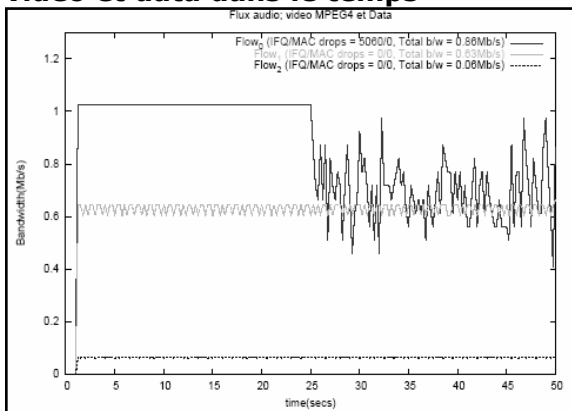


Figure 3.7 EDCF répartition de la bande passante entre les flux audio, vidéo et data dans le temps

A la 25^{ème} seconde de la simulation, nous avons démarré plusieurs sources de trafic dans le but de surcharger le réseau et d'observer le comportement du point d'accès. Le protocole EDCF, qui fait une différenciation des flux avec un mécanisme de

priorité, permet de garantir aux flux prioritaires une bande passante et un faible taux de perte de paquets. Quant à la latence, elle reste acceptable pour les flux prioritaires mais se dégrade pour les flux les moins prioritaires. Ce qui n'est pas le cas du protocole DCF, qui laisse le débit de tous les flux se dégrader et le taux de perte des paquets est vertigineux. Quant à la latence, elle augmente dangereusement pour tous les trafics sans faire de différence entre eux, ce qui est dû à l'existence d'une seule file où l'attente se prolonge.

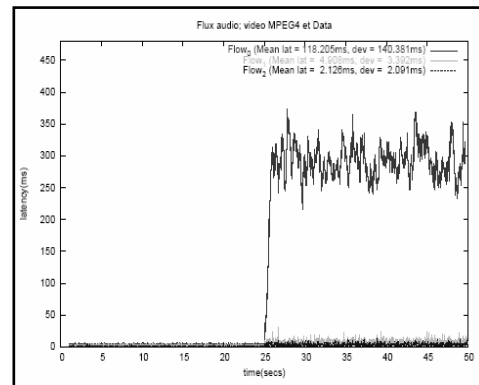


Figure 3.8 EDCF latence des flux Audio, vidéo et data dans le temps

Donc effectivement le protocole EDCF apporte bel et bien une certaine qualité de service aux utilisateurs, mais qui peut être encore amélioré.

Conclusion

Les réseaux informatiques (et en particulier les réseaux sans fil) connaissent une expansion sans précédent, et devant l'évolution des techniques et de la technologie, la simulation permet de tester sans aucun coût ces nouvelles technologies, les nouveaux protocoles mais aussi d'anticiper les problèmes qui pourront se poser dans le futur. Le NS est un logiciel de simulation à événement discret. C'est un logiciel dans le domaine public disponible sur Internet. Son utilisation est gratuite. Pour utiliser NS vous avez besoin d'un ordinateur et d'un compilateur C++. Ns étant un logiciel très évolutif, de nombreux composants se rajoutent chaque année. Il est possible avec NS de simuler un réseau informatique comportant beaucoup de noeuds. Le logiciel est exécutable tant sous Unix que sous Windows. NS est devenu aujourd'hui un standard de référence dans le domaine de simulation, essentiellement des réseaux informatiques.

Référence Bibliographique

- [1] Anis.Koubba « Introduction à l'évaluation des performances des systèmes informatiques et de communication » janvier 2004
- [2] P.Anelli ,E.Horlait « Manuel NS version 2 : Principes de conception et d'utilisation » UPMC-LIP6
- [3] Jean Marie Bonin « Introduction à la simulation d'un réseau à travers NS »
- [4] Polly Huang « NS2 Tutorial » 14 Juin 1999
- [5] Amir.Carta "A Network Topology Presentation Tool" Project Report, UC Berkeley 1993

Les sites Web:

- http://www.cacs.louisiana.edu/~apu7797/utgikar_ns/index.html
- <http://cs.lamar.edu/faculty/osborne/Thesis/index.html>
- <http://www-130.ibm.com/developerworks/>
- <http://www.ibm.com/link/oss.software.ibm.com/redirect.shtml/bluehoc>
- <http://www.inrialpes.fr/planete/mobiwan/>
- <http://canu.informatik.uni-stuttgart.de/mobisim/>
- <http://www.ti-wmc.nl/eurane/>
- <http://www.geocities.com/opahostil/>
- <http://nile.usc.edu/important/>