

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS**  
CENTRE REGIONAL DE TOULOUSE

En vue d'obtenir

**Le DIPLOME d'INGÉNIEUR**

en

**Informatique**

par

Frédéric Camps  
CNRS/LAAS

**Modélisation d'une bibliothèque de sources de  
trafic pour des réseaux IP Multi-Services**

[fcamps@laas.fr](mailto:fcamps@laas.fr)

Juin 2005



## *Résumé*

L'évolution des services offerts au Internaute connaît une croissance très rapide, il apparaît régulièrement de nouveau protocole multimédia que se soit dans le domaine de la téléphonie mobile ou des réseaux IP plus classiques. Ces nouveaux services tentent aussi de respecter une qualité de service nécessaire pour le bon usage des technologies multimédia.

Dans ce cadre, la modélisation fiable des flux Internet est devenue un enjeu important pour le dimensionnement des réseaux et leur configuration mais aussi pour comprendre les phénomènes de congestion que l'on peut observer dans les réseaux à grande ou petite échelle. La modélisation d'une source de trafic consiste à trouver un modèle qui permet de générer des dates de création de paquets d'une certaine taille. La date de création de paquet et la taille sont définies par des lois de probabilité mathématique. Les paramètres du modèle peuvent être obtenus à partir des spécifications ou alors à partir d'une observation d'un réseau. Dans ce dernier cas, l'étude statistique des observations permet de définir un modèle proche de la réalité.

Une source de trafic est définie selon trois niveaux : Session, Application et Paquet. Chaque niveau caractérise le comportement de la source de trafic. Une source de trafic utilise un protocole de transport TCP et/ou UDP, ces protocoles ont une influence sur le comportement de la source.

\*\*

**Mots clé :** source de trafic, protocole multimédia, trafic multimédia, source unitaire, source agrégé, modélisation, simulation, statistique, test statistique, algorithme d'ajustement, CODEC audio et vidéo

\*\*

**Keywords :** source of traffic, multi-media protocol, multi-media traffic, unitary source, aggregated source, traffic modeling, simulation, statistical test, algorithm of adjustment, audio and video CODEC



## Remerciements

*J'adresse mes vifs remerciements à Jean-Marie Garcia, Responsable du groupe RST/LAAS et Directeur de la Société QoS-Design, qui m'a étroitement associé à ses activités de recherche depuis quelques années maintenant.*

*Mes remerciements à H. Hassan, pour son aide précieuse dans le développement de l'interface avec le simulateur événementiel DHS et la spécification des sources génériques de trafic.*

*Mes remerciements aussi aux équipes RST et QoS Design, pour leur disponibilité et le savoir qu'ils ont su partager et particulièrement à Eric Thibault pour son soutien théorique en statistique et son inébranlable enthousiasme.*

*Mes sincères remerciements aussi, a ceux que j'aurais injustement oubliés.*



# Table des Matières

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>2</b>	<b>BIBLIOTHÈQUE DE SOURCES DE TRAFICS.....</b>	<b>3</b>
2.1	APPROCHE GÉNÉRALE.....	3
2.2	BIBLIOTHÈQUE DE SOURCE DE TRAFIC.....	3
2.3	SESSION : COMPORTEMENT DES ARRIVÉES.....	4
2.3.1	<i>Modèles généraux d'arrivée de session.....</i>	<i>5</i>
2.3.2	<i>Modèles généraux de durée des sessions.....</i>	<i>5</i>
2.4	NIVEAU APPLICATION : COMPORTEMENT DES APPLICATIONS.....	5
2.5	NIVEAU PAQUET : ÉMISSION DE PAQUET.....	6
2.6	CONCLUSION.....	6
<b>3</b>	<b>PROTOCOLE DE TRANSPORT.....</b>	<b>7</b>
3.1	TCP.....	7
3.2	UDP.....	7
3.3	RÉPARTITION DE UDP ET TCP.....	8
<b>4</b>	<b>CARACTÉRISATION DES TRAFICS.....</b>	<b>10</b>
4.1	TRAFIC TEMPS RÉEL.....	10
4.1.1	<i>Sources audio.....</i>	<i>10</i>
4.1.2	<i>Sources vidéo.....</i>	<i>10</i>
4.1.3	<i>Autres sources temps réel.....</i>	<i>11</i>
4.2	TRAFIC NON TEMPS RÉEL.....	11
4.3	TRAFIC DE SIGNALISATION.....	12
<b>5</b>	<b>TRANSPORT DES DONNÉES TEMPS RÉEL.....</b>	<b>13</b>
5.1	RTP.....	14
5.1.1	<i>Relais RTP.....</i>	<i>14</i>
5.1.2	<i>Bande passante.....</i>	<i>14</i>
5.2	RTCP.....	15
5.2.1	<i>Bande passante consommée.....</i>	<i>16</i>
5.2.2	<i>Taille des paquets.....</i>	<i>16</i>
5.2.3	<i>Inter-arrivée.....</i>	<i>16</i>
5.3	RTSP : REAL TIME STREAMING PROTOCOL.....	16
5.4	MÉCANISME DE RÉOLUTION DES PERTES.....	18
5.5	CONCLUSION.....	18
<b>6</b>	<b>MODÈLE DE SOURCE TEMPS RÉEL.....</b>	<b>19</b>
6.1	SOURCES AUDIO.....	19
6.1.1	<i>Modèles d'une source Audio.....</i>	<i>19</i>
6.1.2	<i>CODEC pour les sources Audio.....</i>	<i>23</i>
6.2	SOURCES VIDÉO.....	27
6.2.1	<i>La compression vidéo.....</i>	<i>27</i>
6.2.2	<i>Modèles de trafic vidéo.....</i>	<i>28</i>
6.3	CONCLUSION.....	33
<b>7</b>	<b>MODÈLE DE SOURCE NON TEMPS RÉEL.....</b>	<b>34</b>
7.1	HTTP.....	34

7.1.1	Modélisation d'une page web.....	34
7.1.2	Comportement HTTP.....	35
7.1.3	Caractéristiques du trafic.....	35
7.1.4	Modèle de trafic.....	37
7.2	TRAFIC HTTP DANS LES RÉSEAUX 2.5 ET 3 G.....	40
7.2.1	Trafic HTTP dans les réseaux GPRS.....	41
7.2.2	Trafic HTTP dans les réseaux UMTS.....	41
7.3	FTP.....	43
7.3.1	Modèles du trafic FTP dans les réseaux LAN - GPRS.....	43
7.3.2	Modèles du trafic FTP dans les réseaux UMTS.....	43
7.4	EMAIL.....	44
7.4.1	Modèles du trafic Email dans les réseaux LAN - GPRS.....	44
7.4.2	Modèles du trafic Email dans les réseaux UMTS.....	45
7.5	WAP (WIRELESS APPLICATION PROTOCOL).....	45
7.5.1	Structure du réseau.....	46
7.5.2	Caractéristiques.....	46
7.5.3	Modèles de trafic.....	47
<b>8</b>	<b>MODÈLE DE TRAFIC AGRÉGÉ.....</b>	<b>48</b>
8.1	AGRÉGATION DES SOURCES DE TRAFICS AUDIO.....	48
8.1.1	Modèle MMPP-N+1.....	48
8.1.2	Superpositions des sources de trafic.....	49
8.2	AGRÉGATION DES SOURCES DE TRAFICS VIDÉO.....	50
8.2.1	Agrégation de modèles M/G/∞.....	50
8.2.2	Superpositions des sources M/G/∞.....	51
8.3	AGRÉGATION DES SOURCES DE TRAFICS DONNÉES.....	51
8.3.1	Propriétés d'autosimilarité.....	51
8.3.2	Modèles de génération de trafic autosimilaire.....	53
8.4	GESTION DES ARRIVÉES D'APPELS.....	54
8.5	CONCLUSION.....	55
<b>9</b>	<b>EDITEUR GÉNÉRIQUE DE SOURCE DE TRAFIC.....</b>	<b>56</b>
9.1	NIVEAU SESSION.....	57
9.2	NIVEAU APPLICATION.....	58
9.3	NIVEAU PAQUET.....	60
9.4	RÉSEAU DE TEST.....	61
9.5	STOCKAGE DES SOURCES.....	62
9.6	DISTRIBUTION STATISTIQUE.....	63
9.6.1	Distribution mixée.....	63
9.6.2	Distribution tronquée.....	64
9.7	TEST ET VALIDATION.....	65
9.7.1	Source unitaire : HTTP1.1.....	65
9.7.2	Source unitaire audio G711.....	68
9.8	DÉBIT MOYEN D'UNE SOURCE UDP.....	71
<b>10</b>	<b>ANALYSE STATISTIQUE DES SOURCES DE TRAFIC.....</b>	<b>73</b>
10.1	ESTIMATEURS STATISTIQUES.....	73
10.1.1	Estimateurs des quatre premiers moments.....	73
10.1.2	Corrélation d'une source de trafic.....	74
10.1.3	Estimation de la PDF et CDF empirique.....	75
10.1.4	Test d'ajustement.....	76
<b>11</b>	<b>EVALUATION D'UNE SOURCE DE TRAFIC.....</b>	<b>89</b>



11.1	MÉTHODE DES MOINDRES CARRÉS.....	89
11.2	L'ALGORITHME E/M.....	90
11.2.1	<i>Distribution Exponentielle.....</i>	92
11.2.2	<i>Distribution normale.....</i>	92
11.2.3	<i>Distribution lognormale.....</i>	93
11.2.4	<i>Distribution inverse-gaussienne.....</i>	94
11.2.5	<i>Distribution gamma.....</i>	95
11.2.6	<i>Distribution de Pareto.....</i>	96
11.2.7	<i>Distribution de Weibull.....</i>	97
11.2.8	<i>Condition initiale.....</i>	99
11.2.9	<i>Condition d'arrêt.....</i>	99
11.2.10	<i>Interface graphique.....</i>	100
11.2.11	<i>Validation.....</i>	101
11.2.12	<i>Découverte d'un échantillon inconnu.....</i>	107
11.2.13	<i>Conclusion.....</i>	108
<b>12</b>	<b>SYNTHÈSE.....</b>	<b>109</b>
<b>13</b>	<b>BIBLIOGRAPHIE.....</b>	<b>111</b>
<b>14</b>	<b>GLOSSAIRE.....</b>	<b>113</b>



# 1 Introduction

---

L'évolution des services offerts par les réseaux de télécommunications (surtout Internet) a mis en valeur la nécessité de la Qualité de Service QoS pour satisfaire les besoins des clients.

L'absence d'une modélisation fiable des flux Internet pose un défi à l'estimation de la QoS. Les opérateurs, les ISP ont besoin d'outils de simulation qui intègrent des modèles réels, des différents flux qui représentent le trafic transitant sur les réseaux de télécommunications, pour pouvoir dimensionner leurs réseaux de la manière la plus optimale possible.

Cette modélisation est confrontée à la diversité des applications déployées et leurs débits. Une source de trafic peut être très simple avec un débit constant CBR, ou exponentiel. Mais elle peut aussi avoir un comportement complexe et produire des rafales de paquets qui influent fortement sur l'occupation des ressources du réseau.

La simulation d'une source de trafic consiste à trouver un modèle qui permet de générer des dates de création de paquets d'une certaine taille suivant le type d'application. Les modèles de trafic sont définis par des lois de probabilité. Dans cette approche mathématique, sont différenciés trois niveaux: Session, Application et Paquet. Le comportement du niveau Application dépend de l'application en question.

Les applications sont très différentes dans leurs besoins. Elles peuvent imposer des contraintes temporelles dures ou « flexible ». Souvent, on associe ces applications avec le protocole de transport UDP, le protocole de transport non fiable. Une deuxième catégorie regroupe les applications non temps réel qui sont en général de type donnée, associées au protocole de transport TCP, le protocole de transport fiable avec acquittement.

L'objectif de ce document est de présenter les principaux modèles de trafic afin de concevoir une bibliothèque générale de sources de trafics qui regroupe les principales applications déployées sur les réseaux fixes et mobiles: voix, vidéo et données. Cette bibliothèque sera ensuite intégrée au simulateur DHS développé par le LAAS/CNRS et la start-up QoS Design qui détient une licence exclusive du brevet.

La deuxième partie de ce document présente l'analyse des sources de trafic avec des méthodes statistiques.

Ce document comporte 11 chapitres principaux :

**Chapitre 2** : Ce chapitre décrit une approche générale de la modélisation avec la présentation des différentes couches d'un modèle de trafic.

**Chapitre 3** : Dans ce chapitre, on présentera les deux protocoles de transport UDP et TCP et leurs comportements.

**Chapitre 4** : Ce chapitre présente la notion de trafic temps réel et non temps réel, avec notamment les applications multimédia audio et vidéo très utilisées par la communauté Internet.

**Chapitre 5** : Ce chapitre décrit les protocoles temps réels les plus utilisés notamment pour les flux multimédia : RTP, RTCP, RTSP. Ce chapitre traite aussi des mécanismes de résolution de perte qui sont intégrés au protocole ou alors qui sont parfois des trafics supplémentaires en parallèle du trafic principal vidéo ou audio.

**Chapitre 6** : Nous présenterons les sources de trafic temps réel et plus particulièrement les trafics audio et vidéo qui nécessitent des traitements rapides pour leur génération et leur transport.

**Chapitre 7** : Ce chapitre est consacré aux applications de type non temps réel : http, ftp, mail.

**Chapitre 8** : Les modèles agrégés permettent de simuler des quantités importantes de trafics, ces modèles sont très importants dans le cadre des simulations à très grande échelle, chose que ne permet pas l'utilisation d'une source unitaire.

**Chapitre 9** : Après avoir fait le bilan sur le comportement des sources unitaires et agrégées, nous aborderons l'aspect générique des sources de trafic et comment les représenter. Ce chapitre, présente comment générer une source unitaire, comment générer une source agrégée. Nous présenterons brièvement l'outil développé dans le cadre de l'étude et les résultats obtenus pour quelques sources unitaires.

**Chapitre 10** : L'analyse statistique d'une source de trafic est une approche fondamentale pour caractériser les trafics qui sont véhiculés sur un réseau mais aussi pour anticiper les évolutions des architectures et la configuration de chaque élément d'un réseau.

**Chapitre 11** : L'évaluation d'une source de trafic permet de proposer une solution analytique d'une source observée. La solution analytique est construite autour d'algorithmes complexes, la validation de la solution passe par l'utilisation d'outils statistique.

## **2 Bibliothèque de Sources de trafics**

---

### **2.1 Approche générale**

On cherche à modéliser une source de trafic en trois niveaux :

- Niveau Session : pour modéliser l'arrivée de tous les clients qui se connectent au système pour utiliser un certain type d'application.
- Niveau Application : selon l'application demandée, la densité d'informations au cours du temps peut être différente.
- Niveau Paquet : c'est le niveau le plus élémentaire où on génère des paquets qui correspondent aux informations émises par l'application.

### **2.2 Bibliothèque de source de trafic**

Le but de la bibliothèque de trafic est de simuler des trafics d'applications, nous pouvons distinguer trois types de trafics :

- entre un client et un serveur,
- en point à point (par exemple échange de donnée entre deux ordinateurs),
- un trafic de fond (trafic de service : ARP, RARP, OSPF, ...)
- trafic multicast et broadcast (non traité dans ce document)

#### **Entrée :**

Pour générer un trafic, il faut connaître les deux extrémités qui communiquent et le type de trafic. Le simulateur DHS fournit une matrice de trafic qui renseigne les générateurs de trafic de la bibliothèque.

#### **Sortie :**

En sortie la bibliothèque génère des paquets, dont les caractéristiques sont définies par types d'applications (HTTP, E-mail, FTP, VoIP, Vidéo ...).

## 2.3 Session : comportement des arrivées

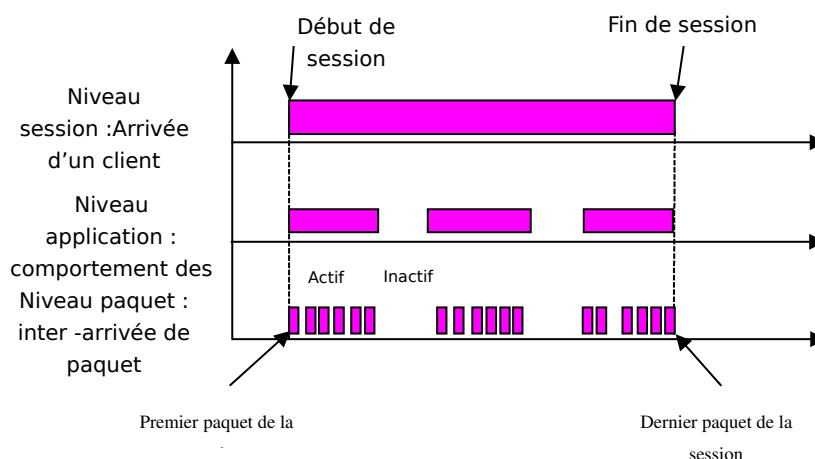


Figure 1 - Trafic unitaire

Ce niveau décrit le comportement d'une population de clients au nœud d'accès (routeur sur un LAN, la station de base dans un réseau radio). L'instant d'arrivée d'un client (début de nouvelle session) dans un système peut être décrit par une distribution d'inter-arrivée de temps. Exemple : un utilisateur se connecte sur un serveur WEB. Le client quitte ensuite le système.

Le comportement des clients peut dépendre de période de temps ou d'autres facteurs :

- Tranche horaire,
- Jour,
- Semaine,
- Tarifs des communications.

### **2.3.1 Modèles généraux d'arrivée de session**

Les modèles généraux de comportement des arrivées de client dans le système sont les suivants :

- Modèle à nombre de session constant : le nombre de client est fixé et ne change pas,
- Modèle de Poisson (ou processus de naissance et de mort) : ce modèle représente un taux d'arrivée moyen  $\lambda$ , sans mémoire, indépendant des arrivées précédentes, chaque client à une durée de vie  $\mu$ ,
- Modèle de Poisson tronqué ou modèle d'Erlang : taux d'occupation moyen des ressources du système.

### **2.3.2 Modèles généraux de durée des sessions**

#### **Application temps réel :**

La durée d'une session est caractérisée par une distribution qui dépend du type d'application. Par exemple pour le trafic téléphonique, on peut supposer que la durée moyenne d'une communication est de trois minutes et que le temps est distribué exponentiellement. On suppose que le temps de communication (temps de conversation + temps de commutation) obéit à une loi exponentielle de moyenne  $D=1/\mu$ . La probabilité que la durée d'un appel soit comprise entre  $t$  et  $t+dt$  est donnée par :  $\mu \cdot e^{-\mu t}$ .

#### **Application non temps réel :**

Pour ce type de source, le temps de communication ne peut être imposé par la source puisqu'il n'y a pas de séquence temporelle à respecter et que le temps inter-paquet est régulé par TCP. On modélisera donc ce type de source par la quantité d'information à transmettre (nombre de paquets à émettre) la durée de la session dépendra de la réponse du réseau et non de la source elle-même.

## **2.4 Niveau application : comportement des applications**

Dans la littérature, le niveau application est parfois appelé « niveau connexion ». Ce niveau caractérise le comportement des applications. Nous pouvons distinguer trois grandes classe de trafic : les données, la vidéo et l'audio. Le niveau application peut être défini par des modèles stochastiques :

- Processus ON / OFF dans un processus conversationnel (HTTP, voix sur IP, ...) : la période ON représente la période d'activité avec des émissions de paquets, la période OFF est inactive.
- Appel permanent pendant toute la durée de session (transfert de fichier FTP, http, ...)

- Processus d'événements corrélés M/G/ $\infty$  (vidéo, LAN) : Ce type de processus permet de bien représenter les corrélations existantes entre des images vidéo successives telles que dans les films, le trafic de vidéoconférence. Ce processus est également utilisé pour simuler les trafics LAN.
- Agrégation de comportement conversationnel, processus MMPP-M (téléphonie) : pour générer N source de trafic simultanément, on peut soit utiliser N sources identiques, soit utiliser des modèles agrégés. Les modèles agrégés nécessitent moins de temps de calcul.

## **2.5 Niveau paquet : émission de paquet**

Le niveau paquet est caractérisé par la distribution des inter-arrivées de paquets et la distribution de la taille des paquets. Les inter-arrivées peuvent être par exemple constante ou exponentielle. L'émission de paquet se fait pendant la période active décrite par le niveau application.

## **2.6 Conclusion**

Nous pouvons distinguer deux catégories d'applications. Les applications temps réel (Audio, Vidéo et système de commande) et les applications non temps réel (Données). Nous avons présenté dans ce chapitre une approche générale pour modéliser le trafic généré par ces deux catégories d'applications. Cette approche hiérarchique présente trois niveaux : Session, Application et Paquets. Nous avons montré qu'un modèle générale peut s'appliquer sur le niveau Session avec un processus de Poisson pour modéliser l'arrivée des clients. Le niveau Application est plus complexe et dépend de la densité d'information générée par l'application. Le niveau Paquet est le plus élémentaire et s'intéresse à la génération des paquets de la source de trafic considérée.



## Protocole de transport

---

L'objectif étant de spécifier une bibliothèque de trafic, il est nécessaire de connaître le comportement des protocoles de transport. Les protocoles de transport les plus utilisés sont TCP et UDP, ils permettent d'acheminer des données de différente nature en proposant des services différents.

Le protocole IP est un protocole de couche 3 qui permet la transmission des paquets de bout en bout sans garantie d'ordre ni de perte.

Les trames UDP et TCP sont encapsulées par la couche IP sous-jacente. Elles peuvent être de longueur variable, la couche IP se charge de leur fragmentation et de leur réassemblage de façon transparente.

### 2.7 TCP

TCP/IP (*Transmission Control Protocol on IP*) est un protocole de transport utilisé à grande échelle sur Internet. C'est un protocole de niveau 4 (transport) qui assure un transfert bidirectionnel de données, de façon fiable et sans erreur, avec contrôle et retransmission des données effectuées aux extrémités de la liaison.

Le contrôle d'erreur de TCP se fait par échange d'information entre deux extrémités. Ces échanges nécessitent de la bande passante et augmentent les temps de traitement (surtout en cas de perte). Les protocoles comme FTP (*File Transfert Protocoles*), telnet, HTTP, SMTP utilisent TCP/IP car les pertes ne sont pas acceptables dans ce type de protocole (TCP assure les retransmissions en cas de perte). Les protocoles qui ont des contraintes de délais ne peuvent pas utiliser TCP.

TCP a un débit particulièrement variable : le mécanisme de transmission de données essaie sans cesse d'augmenter le débit d'émission. Cette augmentation entraîne des pertes et permet à la source d'ajuster le débit par un système d'autorégulation.

### 2.8 UDP

UDP (*User Datagram Protocol*) est un protocole de communication de niveau 4 (transport) sans connexion : cela signifie qu'il n'y a aucune garantie qu'une trame UDP émise arrive à destination. UDP et IP ne proposent pas de QoS.

UDP est principalement utilisé dans les réseaux locaux (dans lesquels la probabilité de perte d'un paquet est faible). Quelques services utilisent UDP : les applications comme NFS (*Network File System*), NIS (*Network Information System*). Le protocole UDP est également utilisé pour transporter les données autorisant des pertes mais ayant des contraintes fortes sur les délais. Par exemple le protocole RTP (*Real-Time Protocol*) s'appuie sur UDP pour transporter des données de type communication audio (VoIP ou vidéo).

## 2.9 Répartition de UDP et TCP

Des mesures sur le trafic IP [15] ont conduit aux répartitions suivantes pour les deux protocoles essentiels de l'Internet que sont TCP et UDP.

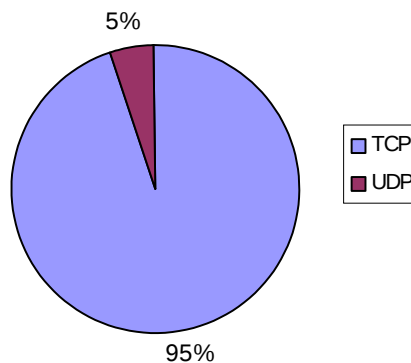


Figure 2 - Utilisation TCP et UDP

Ces pourcentages montrent que le trafic TCP est prépondérant. Cette répartition est cependant transitoire car les nouveaux services multimédias proposés par les opérateurs seront en partie véhiculés par UDP (jeux, peer-to-peer, ...).

Protocole IP	en octet	en Paquet	En flux
TCP	95%	90%	80%
UDP	5%	10%	18%
autres (Ipv6, IP-in-IP, ICMP,...)	-	-	2%

ICMP constitue le troisième pourcentage en terme de trafic derrière TCP et UDP.

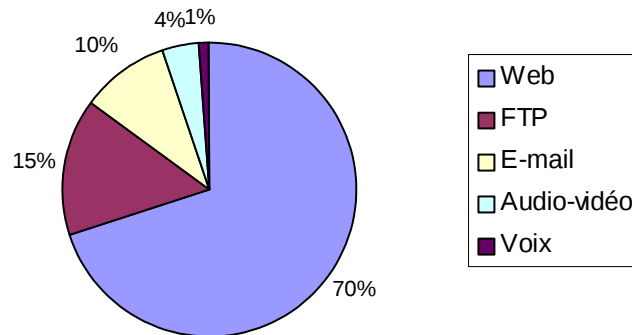


Figure 3 - Répartition des Applications

Des mesures sur le type d'applications utilisées sont représentées sur le schéma précédent. Les résultats montrent clairement que le trafic WEB est le trafic prédominant avec 70%. La tendance est à l'augmentation du trafic HTTP avec l'augmentation de la vitesse de connexion.



## **3 Caractérisation des trafics**

---

### **3.1 Trafic temps réel**

Dans les applications temps réel, c'est la source qui définit (en fait elle impose) la séquence des paquets et leur inter-arrivée. Les paquets arrivent à destination en respectant le profil dynamique de tous les paquets de la session. En fait on autorise certaines modifications (sur le délai de bout en bout, la gigue ou les pertes de paquet) mais très faibles sur ce type de source. Ces contraintes permettent au décodage à la destination de pouvoir recréer le signal original sans perte d'informations perceptible (qualité permanente de l'audio ou de la vidéo respectée).

On associera à ce type de source des contraintes de bornes sur le délai, la gigue et les pertes afin de garantir une certaine qualité de service spécifique à ce type de source. On parle de SLA : Service Level Agreement.

Dans les trafics temps réel, nous distinguons les catégories suivantes :

- les sources audio : voix, real-audio ...
- les sources vidéo : Vidéo téléphonie, Vidéo-conférence, ...
- les sources de trafic d'application temps réel : système de commande,...
- Jeux

#### **3.1.1 Sources audio**

La voix est au départ un signal analogique qui subit un processus de numérisation. Le signal numérique résultant a un débit assez variable suivant le CODEC utilisé. Le débit résultant pour un CODEC MIC classique est 64Kbps et peut descendre jusqu'à 6 Kbps.

Les applications de type Audio ont un débit faible par rapport aux autres types d'applications (vidéo et données). Les pertes de paquets sont autorisées dans certaines limites. Par contre les applications Audio sont très sensibles aux délais et à la gigue.

#### **3.1.2 Sources vidéo**

Plusieurs applications sont apparues grâce à la numérisation de la Vidéo (Télévision numérique, Vidéo à la demande, Vidéo téléphonie, Vidéo-conférence, DVD ...). Contrairement aux applications audio qui ont un débit assez faible de 64 Kbps au maximum (15 Kbps en moyenne), la numérisation de la vidéo produit un débit qui est typiquement supérieur à 100 Mbps. La transmission de la vidéo numérique demande donc une bande passante très large. Les CODEC permettent de réduire efficacement le débit de la vidéo numérique à quelques Mbps au lieu de quelques centaines de Mbps pour la vidéo non compressée.

Les applications Vidéo requièrent des ressources importantes sur les réseaux fixes même après la compression, les réseaux de mobiles actuels 2G et 2.5G (GSM et GPRS) ne peuvent pas supporter ce genre d'application. La troisième génération UMTS autorise des débits allant jusqu'à 2 Mbps par utilisateur ce qui rend les applications vidéo possibles sur les mobiles.

### **3.1.3 Autres sources temps réel**

Il existe d'autres types de source temps réel tels que les systèmes de commande. Les sources de trafics de ces systèmes ont des contraintes temporelles strictes.

## **3.2 Trafic non temps réel**

Dans ce type d'application, la source n'impose pas un débit (temps réel) mais un besoin. Il s'agit en général de transfert de données, pour lesquels on ne tolère aucune perte mais pour lesquels on s'autorise des délais très variables. Aucune qualité de service de type temps réel n'est garantie bien que l'on veuille tout de même assurer certains débits en moyenne ou en crête. Actuellement ces applications sont toutes les applications reposant sur TCP :

- HTTP (WEB),
- FTP,
- SMTP (mail).

La caractéristique de TCP, qui génère du trafic dit "élastique", est de faire varier sans cesse le débit de la source:

- le débit varie d'un seul paquet émis à plusieurs paquets émis en rafale suivant la taille de la fenêtre de congestion de TCP,
- le débit peut varier brusquement lorsqu'il y a détection d'une perte de paquet et réémission des paquets perdus,
- le débit peut varier soudainement en cas de non réponse du destinataire,
- le débit varie en fonction du délai aller retour (RTT : Round Trip Time) qui est donné par le temps que met un paquet pour atteindre sa destination plus le temps que met l'acquittement pour revenir à la source.

### **3.3 Trafic de signalisation**

Le trafic de signalisation est peu étudié mais présente toujours un trafic que l'on appelle trafic de fond, il est composé de :

- broadcast Ethernet
- paquet de gestion et d'erreur ICMP
- information de routage (OSPF, RIP, IP-MPLS ...)
- administration (telnet, ssh, snmp, ftp)
- sonde de réseau : information de trafic, détection pour la sécurité  
...

On l'estime la plupart du temps en fonction du volume globale du réseau, soit quelques pourcents.





## 4 Transport des données temps réel

---

Le transport des sources audio et vidéo est réalisé avec des protocoles de transport temps réel. Le contrôle de ces sources entraîne parfois l'utilisation de protocole de signalisation, ceci peut être vu comme un trafic supplémentaire qui charge les réseaux.

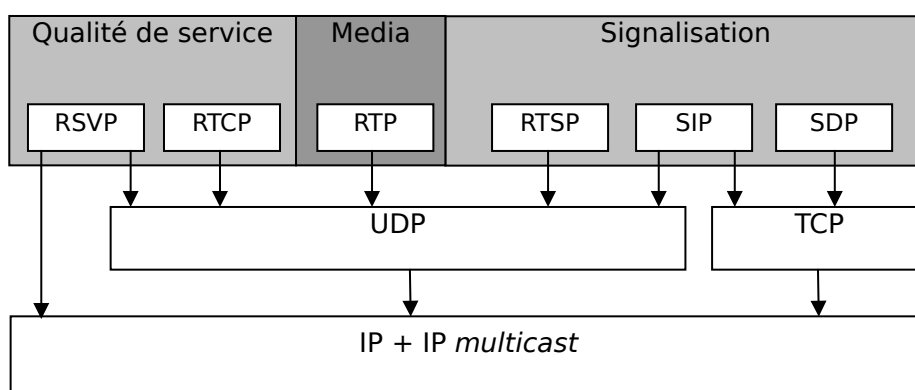


Figure 4 - Transport des données temps réel

Les chapitres suivants présentent les protocoles de transport pour les trafics temps réel et les protocoles de signalisation associés.

On distingue trois grands types d'application multimédia :

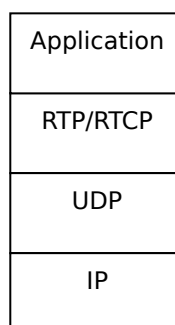
- Diffusion par flot continu de fichiers audio ou vidéo. L'utilisateur a la possibilité d'arrêter la diffusion en cours, de la reprendre, retourner en arrière, avancer en mode accéléré, exactement comme avec un magnétoscope. La littérature désigne souvent ce type de diffusion par l'expression VOD pour Video-On-Demand.
- Diffusion partielle par flot de données audio ou vidéo en direct (Live dans la littérature anglophone).
- Audio et/ou vidéo interactive en temps réel (visioconférence).

## 4.1 RTP

Les briques de base pour le transport temps-réel d'information sont :

- RTP (RFC1889) :Real Time Protocol
- RTCP : Real Time Control Protocol

RTP est un protocole qui fournit des fonctions pour le transport sur un réseau point à point de données temps réel, tels le son et la vidéo, avec des services de diffusion point à point ou de diffusion partielle. RTP ne gère pas la réservation de ressources et ne garantit pas de qualité de service temps-réel. Le transport des données utilise un protocole de contrôle (RTCP, Real Time Control Protocol) pour permettre le monitoring de manière extensible des données diffusées sur un grand réseau multicast et pour fournir des fonctionnalités minimales de contrôle et d'identification. RTP et RTCP ont été définis de manière indépendante des couches de transport et réseau sous-jacentes.



**Figure 5** - RTP dans les couches réseaux

### 4.1.1 Relais RTP

Dans le cas de visioconférence ou une partie des participants est connecté en bas débit et une autre partie des participants est connecté en haut débit, le relais (mixer) RTP permet à chaque participant d'utiliser la bande passante à sa disposition sans pénaliser le groupe de participant avec le débit le plus faible.

### 4.1.2 Bande passante

La gestion de la bande passante n'est pas fournie par le protocole RTP/RTCP mais par le niveau application qui est chargée d'utiliser la bande passante de façon équitable avec les autres applications réseau. RTP fournit au niveau application les outils nécessaires pour calculer la bande passante et déterminer les problèmes de congestion.

## 4.2 RTCP

Le protocole RTCP permet de contrôler en temps réel l'état d'une session RTP. Les paquets RTCP ne comportent pas de données multimédia mais transportent des rapports qui contiennent des statistiques utiles à l'application. Parmi ces statistiques, on trouve le nombre de paquets envoyés, le nombre de paquets perdus et la variation des délais (jitter). Les spécifications de RTP ne définissent pas la manière d'utiliser ces informations, leur utilisation est laissée au libre choix des développeurs. Elles peuvent servir par exemple à modifier les taux de compression en cours de session. Les données RTCP sont transportées en utilisant le même mécanisme de distribution que celui des données. Le protocole utilisé doit donc permettre le multiplexage des paquets de données et de contrôle, par exemple, avec RTP, en utilisant deux ports UDP distincts.

Les applications de visioconférence utilisent RTCP pour synchroniser les sources audio et vidéo en utilisant les informations de temps transportées dans les paquets RTCP

Le protocole RTCP assure quatre fonctions principales :

- La première des fonctions est d'assurer un feed-back sur la qualité de la distribution des données. Cette information peut être utilisée pour le contrôle des codages adaptatifs (changements de taux de compression) mais aussi pour identifier des problèmes dans la distribution. Le fait que les rapports de réception soient envoyés à tous les participants permet à un participant connaissant des problèmes de déterminer si le problème est local ou global. De la même manière, ce feed-back permet de surveiller le réseau pour des acteurs n'appartenant pas forcément à la vidéoconférence comme par exemple les prestataires de services.
- RTCP fournit pour les sources de flux un identificateur persistant, de niveau transport, appelé CNAME pour canonical name. Le CNAME sert à associer différents flux émanant d'un même émetteur, par exemple pour synchroniser des flux audio et vidéo.
- Puisque tous les participants envoient leur rapport à tous les autres, les problèmes d'extensibilité ont été abordés. Ainsi, chaque participant évalue le nombre de participants et adapte la fréquence d'envoi des rapports de manière à ne pas engorger le réseau.
- RTCP fournit aussi des informations nécessaires pour un contrôle minimal des sessions comme par exemple l'identifiant d'un participant pour l'afficher dans l'interface utilisateur. RTCP peut ainsi être un moyen pratique de transmettre des informations à tous les participants mais il n'est pas nécessaire que tout passe par lui.

#### **4.2.1 Bande passante consommée**

Notons que si la quantité de trafic RTP envoyé sur le réseau multicast n'augmente pas avec le nombre de receveurs, celle du trafic RTCP augmente linéairement avec le nombre de receveurs. Pour pallier ce problème, RTCP modifie la fréquence d'envoi des paquets RTCP en fonction du nombre de participants. RTCP tente de limiter sa consommation à **5%** de la bande passante de la session.

#### **4.2.2 Taille des paquets**

La taille du paquet RTCP est variable

- la taille peut varier pendant les phases de communication :
  - packet sender report : max = 640 bits (à confirmer),
  - round-trip time computation : max = 352 bits,
  - source description : max = 160 bits.

A la taille des données RTCP il faut rajouter l'encapsulation UDP et IP.

#### **4.2.3 Inter-arrivée**

L'intervalle de temps entre deux paquets RTCP est de 5 seconds minimums pour éviter les rafales de paquets RTCP qui alors occuperait plus **de 5% de la bande passante** pendant un bref instant.

L'intervalle de temps de **5 secondes** varie aléatoirement dans l'intervalle [0.5,1.5] pour éviter une synchronisation des trafics de signalisation. Le premier paquet émis après la connexion à une session est également envoyé selon un temps aléatoire qui représente la moitié d'un intervalle de temps entre deux paquets RTCP.

### **4.3 RTSP : Real Time Streaming Protocol**

RTSP RFC2326 (<http://www.realnetworks.com/devzone/library/rtsp>) est un protocole applicatif pour contrôler le transfert de données temps réel mis au point conjointement par l'Université de Columbia, Real Networks et Netscape Corp. Il fournit un cadre extensible pour une livraison à la demande, contrôlée, de données temps réel telles la vidéo/audio. Les sources des données peuvent être des flux live (spectacle en direct) ou enregistrés. Ce protocole est étudié pour contrôler plusieurs sessions de transfert de données temps réel, fournit les moyens de choisir un canal de communication comme UDP, multicast UDP ou TCP et fournit un moyen de choisir parmi les mécanismes de livraison de RTP.

RTSP agit comme une télécommande du serveur de données multimédia via le réseau.

L'ensemble des flots à contrôler est défini par une description de présentation.

Il n'y a pas de notion de connexion RTSP à proprement parler ; à la place, le serveur entretient une session libellée par un identificateur. Une session RTSP n'est en aucun cas liée à un type de connexion de transport tel TCP, au contraire durant une session RTSP, un client peut ouvrir ou fermer plusieurs connexions de transport au serveur pour effectuer des requêtes RTSP. Un protocole de transport sans connexion comme UDP peut aussi être utilisé.

Le protocole supporte les opérations suivantes.

- Collecte d'un objet multimédia depuis un serveur. Le client peut demander un descripteur de présentation via HTTP ou autre. Si la présentation est multi-diffusée (multicast), le descripteur de présentation contient les adresses et ports à utiliser ;
- Invitation d'un serveur multimédia à une conférence. Un serveur multimédia peut être invité à une conférence soit pour jouer un objet multimédia, soit pour enregistrer tout ou partie de la présentation ;
- Addition d'un objet multimédia dans une présentation. Plus particulièrement pour les présentations en direct, il est utile que le serveur puisse indiquer au client que de nouveaux objets multimédia sont disponibles.

Les requêtes RTSP peuvent être gérées par les proxies, les tunnels et les caches de la même manière que les requêtes HTTP/1.1.

Tout comme ftp, RTSP est un protocole "out-of-band". Lors d'une connexion ftp entre deux machines, deux connexions TCP sont créées sur des ports différents, l'une transporte les données, l'autre des informations de contrôle. La connexion de contrôle est dite canal "out-of-band" tandis que l'autre est appelée canal de données.

La figure 5 représente un exemple simple de transactions RTSP. Le navigateur Web demande en premier lieu un descripteur de présentation au serveur Web. Le descripteur de présentation peut contenir une ou plusieurs références à des objets multimédia ainsi que des commandes de synchronisation. Chaque référence à un objet multimédia commence par la méthode de l'URL, rtsp ://. Le fichier descripteur de la présentation est renvoyé au navigateur Web via HTTP en réponse à sa requête. Le navigateur invoque le visionneur ad hoc en fonction du champ content-type : de la réponse. Le visionneur et le serveur échangent alors une série de messages RTSP. Le visionneur envoie en premier lieu une requête de type "SETUP", à laquelle le serveur répond par une réponse de type "SETUP". Le visionneur envoie une requête "PLAY", par exemple pour de l'audio de basse qualité, le serveur réplique par une réponse "PLAY". Le serveur de flot envoie alors le fichier audio de basse qualité dans le canal de données. Si plus tard, le visionneur envoie une requête "PAUSE", le serveur répond avec la réponse

"PAUSE" et le flot de données est stoppé. Pour terminer une session, le visionneur envoie une requête "TEAR DOWN" à laquelle le serveur réplique par une réponse de même type.

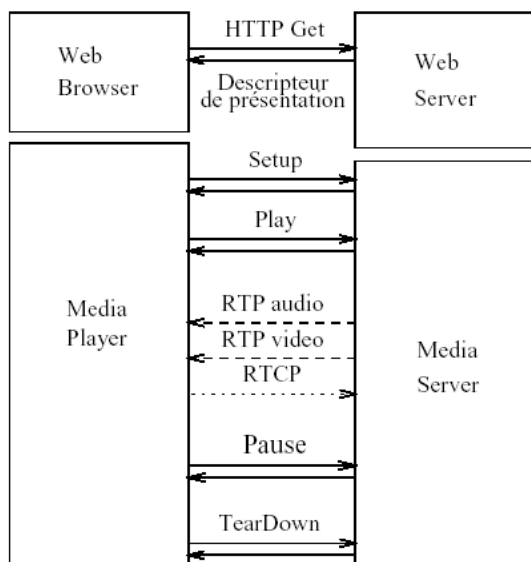


Figure 6: RTSP

## 4.4 Mécanisme de résolution des pertes

Dans le cas du transport de l'audio sur Internet les mécanismes suivants sont mis en œuvre :

- contrôle à boucle fermée ARQ : dans le cas de l'audio
- FEC (système de résolution de perte intégré dans le protocole de codage)

## 4.5 Conclusion

RTP permet de transporter les données temps réel avec des LSA strictes ou molles. RTCP est associé à RTP pour réguler la source et donner des informations sur les performances : délais, gigue, perte.

## 5 Modèle de source temps réel

---

### 5.1 Sources Audio

On peut associer les sources audio avec deux types d'applications : Conversation (Téléphone, Audio-Phonie...) et Lecture (Audio-Conférence, mail-voice,...)

La voix est numérisée avec des CODEC (CODEur/DECodeur) utilisant deux types de quantification : scalaire et vectorielle. Dans la quantification scalaire on ne prend pas en compte la corrélation qui peut exister entre les échantillons (par exemple MIC avec 64 Kbps). Cette quantification est simple mais produit un débit important. La quantification scalaire prédictive réduit le débit en codant la différence entre les échantillons au lieu des échantillons (par exemple ADPCM avec 32 Kbps).

Dans la quantification vectorielle on utilise la forte corrélation qui existe entre certains échantillons. Ces échantillons sont regroupés en vecteurs afin d'être quantifiés ensemble. Les débits résultant peuvent descendre jusqu'à 5-6 Kbps.

Le tableau suivant résume les différents CODEC audio utilisés.

Tableau 1 : différents CODEC audio utilisés

CODEC Audio	G.711	G726/727	G728	G729	G723.1
Débit Kbps	64	32	16	8	5.3-6.3

Avant d'expliquer le fonctionnement de chacun de ces CODEC nous allons présenter les modèles de source de trafic utilisés pour représenter les sources de trafic audio.

#### 5.1.1 Modèles d'une source Audio

##### Processus ON-OFF

Le trafic voix au niveau source se caractérise par une période active (temps de parole) suivie par une période inactive (temps d'écoute). Pendant la période active la source émet des paquets à intervalles réguliers de longueur T (temps de

paquetisation). Par exemple le temps de paquetisation d'une source ADPCM (32 kbps) est de 250 ms, c-à-d un octet tout les 250 ms.

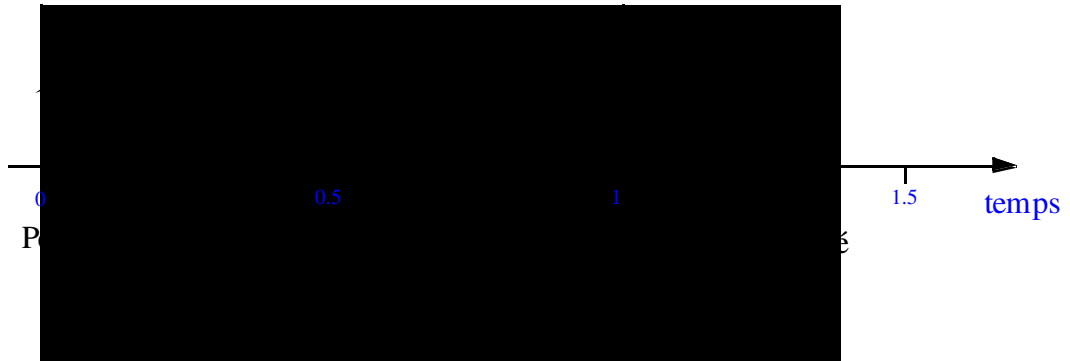


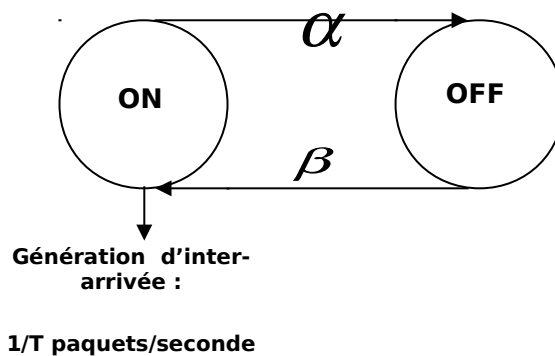
Figure 7 – Modèle d'une source voix

Les modèles parus dans la littérature pour les sources voix utilisent en général des lois exponentielles pour représenter la durée des périodes active et inactive. Une étude faite par Shuang Deng [21] a montré que l'approximation exponentielle des différentes périodes n'est pas précise. Il est difficile de statuer sur l'unicité d'un modèle pour caractériser les différentes applications Audio.

Les durées des périodes actives et inactives sont en général estimées par des lois exponentielles (exemple :  $f(x) = \lambda \exp(-\lambda x)$ ) indépendantes de paramètres respectifs  $\alpha$ ,  $\beta$ . En général une source voix est représentée par un modèle ON-OFF Markovien qui se caractérise par trois paramètres :

- La durée moyenne d'une période d'activité :  $T_{on}=1/\alpha$  (le paramètre de la loi exponentielle de la période d'activité), ce qui représente la moyenne de la loi exponentielle.
- La durée moyenne d'une période d'inactivité :  $T_{off}=1/\beta$  (le paramètre de la loi exponentielle de la période d'inactivité).
- Le taux d'émission constant durant une période d'activité :  $\lambda = 1/T$ , avec  $T$  = inter-arrivée des paquets en fonction du débit du CODEC.

Figure 8 - Processus ON-OFF





Probabilité du passage on à off :  $P(on \rightarrow off) = \frac{\alpha}{\alpha + \beta}$ , passage off à on :

$$P(off \rightarrow on) = \frac{\beta}{\alpha + \beta}$$

Probabilité d'être dans l'état ON :  $P(on) = \frac{1/\alpha}{T_{on} + T_{off}} = \frac{1/\alpha}{1/\alpha + 1/\beta} = \frac{\beta}{\beta + \alpha}$

Probabilité d'être dans l'état OFF :  $P(off) = \frac{1/\beta}{T_{on} + T_{off}} = \frac{\alpha}{\alpha + \beta}$

Dans l'étude de Deng [21] les mesures sur le trafic téléphonique et le trafic audio dans les réseaux à commutation de paquet a permis de donner des valeurs à ces paramètres. On distingue dans cette étude entre deux catégories d'applications audio : Conversation et Lecture. Ces modèles de conversation et de Lecture diffèrent du modèle classique (Téléphonie traditionnelle) comme le montre les résultats suivants :

Tableau 2 : Période ON/OFF audio

Type	Ton	Toff
<b>Classique</b>	0.352 seconde	0.650 seconde
<b>Conversation</b>	7.24 secondes	5.69 secondes
<b>Lecture</b>	3.23 secondes	0.41 seconde

La différence entre ces deux modèles et le modèle classique peut s'expliquer par la sensibilité aux silences des CODEC utilisés. Le CODEC de Deng [21] est moins sensible aux silences que le CODEC ADPCM par exemple. De même on remarque que l'activité dans le modèle de lecture est plus fluide que dans la conversation ce qui se traduit par des périodes de silences très courtes.

On peut calculer la valeur moyenne de  $\bar{\lambda}$  du débit en paquets par seconde sur l'ensemble des états de la manière suivante :

$$\lambda = P_{on} \lambda = \lambda \frac{T_{on}}{(T_{on} + T_{off})} = \lambda \frac{\frac{1}{\alpha}}{(\frac{1}{\alpha} + \frac{1}{\beta})} = \frac{\beta}{(\beta + \alpha).T} \quad (1)$$

A noter finalement qu'afin d'assurer une bonne qualité d'écoute, il faut que le temps de transport de la voix entre l'émetteur et le récepteur soit inférieur à 150ms, avec une tolérance pouvant aller jusqu'à 600 ms pour un réseau satellite. Lorsque le débit devient très faible 5-8 kbps, le temps de paquets devient très long ce qui amène la couche AAL2 (en ATM) par exemple à réduire ce temps

en empaquetant 30ms d'information dans une cellule (une partie de la cellule sera vide ou utilisée par d'autres sources).

Tableau 3 : Temps de paquétisation

CODEC Audio	G.711	G726/727	G728	G729	G723.1
T ms	6	12	24	30	30

### Modèle IPP

Le processus IPP (Interrupted Poisson Process) permet également de modéliser la voix. La différence avec le modèle ON-OFF du paragraphe précédent est que l'émission des paquets suit une loi de Poisson pendant la période ON.

Ainsi le modèle IPP est caractérisé par trois paramètres :

- La durée moyenne d'une période d'activité :  $T_{on}=1/\alpha$
- La durée moyenne d'une période d'inactivité :  $T_{off}=1/\beta$
- Le taux d'émission durant une période d'activité :  $\lambda =1/T$  paramètre de la loi de Poisson.

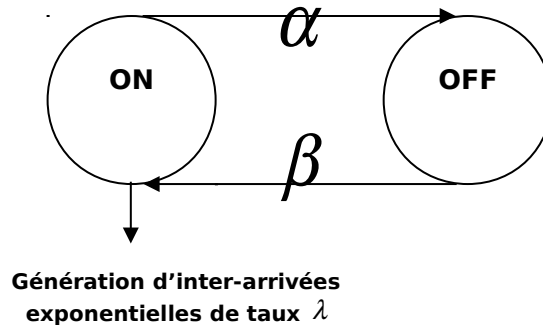


Figure 9 - processus IPP

Bien qu'il existe plusieurs méthodes d'approximation de processus ON-OFF par un processus IPP, on peut prendre les mêmes valeurs pour  $\alpha$ ,  $\beta$  et  $1/T$  pour  $\lambda$ . L'avantage avec ce modèle est que l'on sait en déduire un modèle d'agrégation exact.

## **5.1.2 CODEC pour les sources Audio**

### **5.1.2.1 CODEC RealAudio**

Le model RealAudio provident du trafic du programme RealAudio (trafic qui provident de serveur de musique, radio, concert). La principale partie de ce trafic est véhiculée par UDP (60-80%) ce qui limite les congestions.

Le débit moyen de RealAudio est de l'ordre de 16 kb/s à 20 kb/s. La taille de paquet typique est de 293 octets pour 16kb/s et 495 octets pour 20 kb/s.

Un modèle typique est représenté par un processus ON-OFF: une source envoie une rafale de 5-6 paquets avec une inter-arrivée de 0.03 s (la moyenne de la période On est de 0.2 s), la rafale est suivie par une période OFF de 1.8 s. Le temps d'inter-arrivée pendant la période ON a une variance très faible et peu être considéré comme déterministe.

#### **RealAudio Low Model**

Le modèle "basse qualité" caractérise le flux RealAudio avec un débit de 16 kb/s. Le processus ON-OFF est caractérisé par les paramètres suivants :

- Temps moyen de conversation :  $\bar{T}_{on} = 0.2$
- Temps d'attente (écoute) :  $\bar{T}_{off} = 1.8$
- Le débit constant des paquets pendant la période ON :

$$\lambda = \bar{D} \frac{T_{ON} + T_{OFF}}{T_{ON}} = 16 * \frac{2}{0.2} = 68.26 \text{ paquet / s}$$

- La taille constante des paquets : 293 + 40 (entête IP/UDP/RTP) octets

#### **RealAudio High Model**

Le modèle "basse qualité" caractérise le flux RealAudio avec un débit de 20 kb/s. Le processus ON-OFF est caractérisé par les paramètres suivants :

- Temps moyen de conversation :  $\bar{T}_{on} = 0.2$
- Temps d'attente (écoute):  $\bar{T}_{off} = 1.8$
- Le débit constant des paquets pendant la période ON :

$$\lambda = \bar{D} \frac{T_{ON} + T_{OFF}}{T_{ON}} = 20 * \frac{2}{0.2} = 50.5 \text{ packets / s}$$

- La taille constante des paquets : 293 + 40 (entête IP/UDP/RTP) octets

### **5.1.2.2 CODEC Audio G723.1**

Le CODEC G.723.1 (ITU Recommandation de 1996) autorise deux modes d'encodage : le codage « basse qualité » à 5.3 Kb/s et le codage « haute qualité » à 6.6 kb/s.

Le CODEC « basse qualité » représente une seule voix, il utilise l'algorithme ACELP pour la réduction du débit.

Le CODEC « haute qualité » représente une seule voix, il utilise l'algorithme MP-MLQ (Multi Pulse-Maximum Likelihood Quantization).

L'utilisation typique de ces CODEC est la téléphonie sur IP sur les réseaux paquet et les vidéo-conférences.

### **5.1.2.3 CODEC Audio G728**

Le G.728 représente une seule source de voix associée au CODEC G.728, ITU G.728 (ITU Recommandation in 1992). Ce CODEC standard utilise l'algorithme de compression « Low-Delay Code Excited Linear Prediction (LD-CELP) » pour réduire le débit jusqu'à 16kb/s, son utilisation classique est la voix sur IP dans les réseaux paquets.

### **5.1.2.4 CODEC Audio G729**

Le G729 représente une seule source voix associée au CODEC G.729, ITU-T G.729 (ITU Recommandation de 1996) qui utilise une structure conjuguée d'une prédiction linéaire du code algébrique (CS-CELP) et un algorithme de réduction du débit jusqu'à 8Kv/s. Ce CODEC est typiquement utilisé pour la voix sur IP dans les réseaux paquet comme Internet. Ses caractéristiques sont les suivantes :

Tableau 4 : G729

<b>Taux moyen d'émission</b> $\lambda$	<b>Durée moyenne d'une inter-arrivée (<math>1/\lambda</math>)</b>	<b>Variance inter-arrivées</b>
11.7 paquets/s	8.53e-02 s	6.88e-02

La génération de flux audio se fait avec un processus On-Off, dont les caractéristiques moyennes sont données dans le tableau ci-après :

Tableau 5 : ON-OFF G729

<b>Type Audio</b>	<b>Période On (Ton)</b>	<b>Période Off (Toff)</b>
Modèle classique	0.352 sec	0.650 sec

Durant la période d'activité, l'émission de paquets se fait toutes les trente millisecondes (caractéristique de paquetsisation du CODEC G729).

### **5.1.2.5 CODEC Audio G711**

Le CODEC G711 est plus ancien que le CODEC G729 et ne possède aucun algorithme de compression ce qui conduit à un nombre de paquets moyens plus important que le CODEC G729. La voix est codée sur 8 bits à une fréquence de 8 KHZ, ce qui donne un débit de 64 kbits/sec. Ce CODEC est typiquement utilisé dans les réseaux téléphoniques publics et privés.

La génération de flux audio se fait comme avec le CODEC G729, c'est à dire avec un processus On-Off de même caractéristique :

Tableau 6 : G711

<b>Taux moyen d'émission <math>\lambda</math></b>	<b>Durée moyenne d'une inter-arrivée (<math>1/\lambda</math>)</b>	<b>Variance inter-arrivées <math>\sigma</math></b>
29.273 paquets/s	3.41608e-02	2.83169e-02

Durant la période d'activité, l'émission de paquets se fait toutes les douze millisecondes (caractéristique de paquetsisation du CODEC G711).

Tableau 7 : ON-OFF G711

<b>Type Audio</b>	<b>Période On (Ton)</b>	<b>Période Off (Toff)</b>
Modèle classique	0.352 sec	0.650 sec

### **5.1.2.6 CODEC Audio G726**

Le CODEC G726 est intermédiaire entre le CODEC G729 et le CODEC G711. Les échantillons sont corrélés dans le temps. Un algorithme de type ADPCM est utilisé pour la compression. Le débit résultant est de 32 kb/s. Une utilisation typique du G726 est la téléphonie avec TCP/IP sur les réseaux paquet (VoIP).

Tableau 8 : G726

<b>Taux moyen d'émission <math>\lambda</math></b>	<b>Durée moyenne d'une inter-arrivée (<math>1/\lambda</math>)</b>	<b>Variance inter-arrivées</b>
21.95 paquets/s	4.55466e-02	3.75329e-02

La génération de flux audio se fait comme avec le CODEC G729, c'est à dire avec un processus On-Off de mêmes caractéristiques :

Tableau 9 : ON-OFF G726

Type Audio	Période On (Ton)	Période Off (Toff)
Modèle classique	0.352 sec	0.650 sec

Durant la période d'activité, l'émission de paquets se fait toutes les seize millisecondes (caractéristique de paquetsisation du CODEC G726).

Les CODEC G711, G726, G729 sont utilisés pour coder la voix. On peut utiliser un processus IPP ou ON-OFF pour les représenter. Les paramètres de ces CODEC sont résumés dans le tableau suivant:

Tableau 10 : Paramètres des CODECS audio

CODEC	1/α	1/β	T paquetsi- sation	Débit (état ON) (Octet/s)	Taille Paquet (octet )	$\bar{D}$ (Kbits/s )
G711	1/0.352	1/0.650	0.012	64000	136	31.1
G726	1/0.352	1/0.650	0.016	32000	104	17.8
G729	1/0.352	1/0.650	0.030	8000	70	6.4
G723 low	1/0.352	1/0.650	0.030	5300	60	5,5
G723 high	1/0.352	1/0.650	0.030	6300	64	5,9
RealAudio low	1/0.2	1/1.8	-	16000	323	-
RealAudio high	1/0.2	1/1.8	-	20000	323	-

Les valeurs sont calculées de la façon suivante (exemple G711) :

- Durée de l'état ON :  $\alpha = 1/Ton = 1/0.352$  s

- Durée de l'état OFF :  $\beta = 1/Toff = 1/0.650$  s

- Durée d'émission pendant ON :  $\lambda = 1/T = 1/0.012$  s pendant la période ON.

- Débit moyen pendant la période ON :

$$\lambda = \frac{Ton}{Ton \cdot Toff \cdot Tpaquetisation} = \frac{0.352}{(0.352 + 0.650) \cdot 0.012} = 29.3s^{-1}$$

- Débit moyen du processus ON/OFF :  $\bar{D} = \lambda \cdot (TaillePaquet) = 29.3 \cdot 136 \cdot \frac{8}{1024} = 31.1kb/s$

## 5.2 Sources Vidéo

### 5.2.1 La compression vidéo

La vidéo est numérisée de manière périodique soit in-line pour des applications type vidéo conférence soit out-line pour des applications type vidéo streamée. Les CODEC vidéo compressent l'information afin de réduire le volume de données à transmettre. Le codage vidéo utilise deux types de redondance spatiale et temporelle sur les séquences vidéo. La redondance spatiale repose sur la corrélation qui existe sur les pixels adjacents, alors que la redondance temporelle utilise la forte corrélation entre les images successives. Le tableau suivant résume les CODEC vidéo et leurs débits :

Tableau 11 - Débit CODEC vidéo

<b>CODEC Vidéo</b>	M-JPEG	MPEG-1	MPEG-2	MPEG-4	H261	H263
<b>Débit Mbps</b>	8-10	1.5	4-10	0.005-0.064	N*0.064	0.030

Le JPEG (Joint Photographic Expert Group) qui est la principale norme de compression spatiale utilise seulement la redondance spatiale dans une image, et le M-JPEG représente une séquence d'images indépendantes JPEG.

Le H261 permet de transporter des applications comme la vidéo-conférence et la vidéo-téléphonie sur des lignes 64 kbps du réseau B-ISDN, en utilisant la redondance spatiale et temporelle entre les images dans une séquence vidéo.

Dans le cas de MPEG un cycle de compression a été introduit en groupant un nombre d'image dans un Group Of Picture (GOP).

Il y a trois types d'images dans un GOP :

- les images I (dites de référence) qui ne sont compressées que spatialement (permettant le rafraîchissement de l'information),
- les images P qui utilisent la compression spatiale et temporelle unidirectionnelle passé (sur les images I ou P)
- et les images B qui sont codées en fonction des redondances spatiales et temporelles passées et futures.

Un Groupe d'Images GOP, est exprimé en fonction de N le nombre total des images entre deux images I et M le nombre des images B entre deux images P.

Exemple N= 12 et M=2 : GOP=I B B P B B P B B P B B

## 5.2.2 Modèles de trafic vidéo

### 5.2.2.1 MMPP-M modélisation d'une source vidéo à partir du débit

Le modèle MMPP-M peut être utilisé pour représenter des séquences vidéo très régulière qui peuvent correspondre par exemple à des séquences de vidéo-téléphonie et vidéo-conférence. Les séquences doivent être très régulières dans le sens ou elles ne contiennent pas de changement abrupt de scène. On peut dire dans ce cas-ci que le débit est assez régulier.

Le débit dans ces conditions est une fonction de temps  $\lambda(t)$  caractérisé par: le débit moyen  $E(\lambda)$ , la variance  $\sigma^2$  du débit et la fonction de corrélation  $\rho(\tau) \approx e^{-\alpha\tau}$ .

Le processus MMPP-M peut approcher la fonction  $\lambda(t)$  par une fonction  $\bar{\lambda}(t)$ .

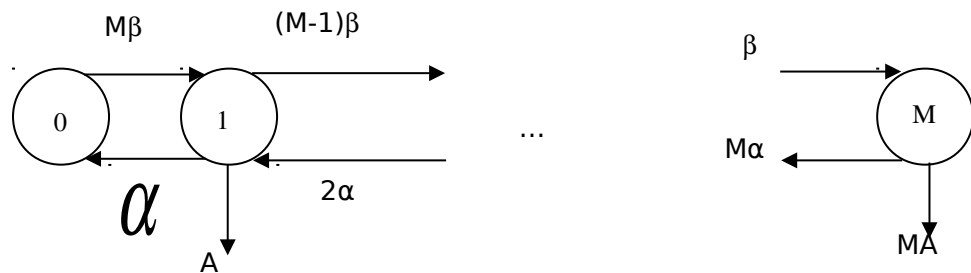


Figure 10 - MMPP-M

M = nombre de source -1

$$P(\bar{\lambda}(t) = kA) = \frac{M!}{k!(M-1)!} P_{on}^k (1 - P_{on})^{M-k} \quad (2) \text{ probabilité du débit moyen pour } kA \text{ source}$$

$$E(\bar{\lambda}) = MA P_{on} \quad (3) \text{ débit moyen}$$

$$\text{var}(\bar{\lambda}(t)) = MA^2 P_{on} (1 - P_{on}) \quad (4) \text{ variance du débit}$$

$$\bar{\rho}(\tau) = e^{-(\alpha+\beta)\tau} \quad (5) \text{ fonction de corrélation}$$

$$\text{Avec : } P_{on} = \frac{\beta}{\alpha+\beta} \quad (6) \text{ probabilité d'être dans la période ON}$$

Les paramètres  $\alpha$ ,  $\beta$  et A peuvent être calculés en égalisant les moyennes, variances et corrélations associées à  $\lambda(t)$  et  $\bar{\lambda}(t)$ .



$$\alpha = a / (1 + \frac{E^2(\lambda)}{M\sigma^2})$$

(7)  $\alpha$  = probabilité de passer d'un état  $M$  à  $M-1$ ,

$$\beta = a - \alpha$$

(8)  $\beta$  = probabilité de passer d'un état  $M$  à  $M+1$

$$A = \frac{\sigma^2}{E(\lambda)} + \frac{E(\lambda)}{M}$$

(9) taux d'émission constant pendant la phase d'activité

### 5.2.2.2 Modèle M/G/∞

Pour représenter la corrélation présente dans la distribution des temps d'inter-arrivées de certaines sources de trafic vidéo, on peut utiliser le processus d'occupation du système M/G/∞ [19][22]. Le modèle M/G/∞ représente un processus d'occupation d'une file d'attente avec des clients qui arrivent selon une loi de Poisson (génération de paquet avec une inter arrivée exponentielle), une loi de service G (définie par sa moyenne et sa variance), un nombre infini de serveur.

Le processus d'occupation noté  $X_n$  représente le nombre de clients dans le système au début de l'intervalle temporel  $[nT, (n+1)T[$ , ou  $T$  est l'échelle de temps représentant le temps de génération d'une image (1/25 secondes) ou d'un GOP (Groupe d'Image).

Le processus  $X_n$  est stationnaire et corrélé. La corrélation peut être à courte ou à long dépendance (SRD/LRD).

Evidement la taille des séquences générées par le modèle  $X_n$  ne correspond pas forcément aux tailles des données réelles (Images et GOP). La séquence générée  $X_n$  sera distribuée suivant une loi de poisson de paramètre  $\lambda E(\sigma)$  avec  $\sigma$  le temps de service de la distribution G et  $\lambda$  est le paramètre de Poisson de la loi M, d'où la nécessité de faire une transformation qui respecte la corrélation mais donne la loi désirée de la taille. Cette transformation est définie comme ci-dessous :

Avec :

$$\begin{aligned} Y(n) &= f(X_n) \\ f(x) &= F^{-1}(F_p(x)) \end{aligned} \quad (10)$$

avec  $F_p$  la fonction de répartition associé a la distribution de la séquence  $X_n$ , et  $F^{-1}$  est la fonction de répartition inverse de la loi qu'on désire obtenir. La corrélation temporelle est conservée par la transformation du processus ( $X_n$ ) en ( $Y_n$ ).

Concernant les sources vidéo, il a été mis en évidence trois types de corrélations différentes. Ces formes de corrélation sont :

**Modèle de source temps réel**

$$\rho(k) = e^{-bk} \text{ avec } b > 0 \text{ (SRD)} \quad (11)$$

$$\rho(k) = e^{-b\sqrt{k}} \text{ avec } b > 0 \text{ (SRD)} \quad (12)$$

$$\rho(k) = (k + 1)^{-b} \text{ avec } b > 0 \text{ (LRD)} \quad (13)$$

où k représente le pas de corrélation.

Toutefois, il est très difficile d'identifier un unique modèle par type de séquence vidéo, car chaque séquence possède des corrélations qui sont très variables.

La taille des Images ou des GOP peut être une loi classique comme Lognormal, Gamma, Weibull, Pareto, ou une loi normale. Des mélanges de distribution peuvent aussi être utilisés. La généralisation est difficile à faire.

Le tableau suivant montre les paramètres de génération de trois CODEC utilisant le processus M/G/∞

Tableau 12 - paramètres du générateur M/G/∞ pour la vidéo

Source Vidéo	CODEC	Time Slot	Corrélation	Frame ou GOP Size Distribution Byte
MJPEG	M-JPEG VBR	Frame 1/25s	$e^{-0.03\sqrt{k}}$	<b>Gamma</b> Mean=12550 Var=1.255e+07
MPEG1	MPEG 1 VBR	GOP 12/25s	$e^{-0.35\sqrt{k}}$	<b>LogNormal</b> Mean=141350 Var=2.1544e+09
MPEG2	MPEG 2 VBR	Frame 1/25s	$e^{-0.055\sqrt{k}}$	<b>LogNormal</b> Mean=21015 Var=1.1443e+08
MPEG4_Film_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.154\sqrt{k}}$	<b>Gamma+LogNormal</b> Mean=9265 Var=2.9235e+07
MPEG4_Film_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.1597\sqrt{k}}$	<b>Gamma+LogNormal</b> Mean=14860 Var=9.151e+07
MPEG4_Film_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.1722\sqrt{k}}$	<b>Gamma+LogNormal</b> Mean=41968 Var=5.1139e+08
MPEG4_Cartoon_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.3733\sqrt{k}}$	<b>Gamma+LogNormal</b> Mean=13826 Var=3.3858e+07
MPEG4_Cartoon_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.341\sqrt{k}}$	<b>Gamma+LogNormal</b> Mean=25385 Var=9.5858e+07

**Modèle de source temps réel**

MPEG4_Cartoon_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.3095\sqrt{k}}$	<b>Gamma+LogNormal</b> Mean=78154 Var=4.2495e+08
MPEG4_Sport_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.106k}$	<b>Gamma+LogNormal</b> Mean=11258 Var=2.8204e+07
MPEG4_Sport_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.108k}$	<b>Gamma+LogNormal</b> Mean=18835 Var=9.179e+07
MPEG4_Sport_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.104k}$	<b>Gamma+LogNormal</b> Mean=49958 Var=5.7125e+08
MPEG4_ParkingCamera_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.0038k}$	<b>Normal+Normal</b> Mean=7983 Var=9.2239e+05
MPEG4_ParkingCamera_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.0032k}$	<b>Normal+Normal</b> Mean=14186 Var=4.8892e+06

**Tableau 13 - paramètres du générateur M/G/∞ pour la vidéo**

Vidéo Source	CODEC	Time Slot second	Corrélation	Size Distribution Byte
MPEG4_ParkingCamera_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.003k}$	<b>Normal+Normal</b> Mean=47295 Var=3.7749e+07
MPEG4_News_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.0778k}$	<b>Gamma+LogNormal</b> Mean=43674 Var=5.5207e+08
MPEG4_Talk_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.024k}$	<b>Gamma+LogNormal</b> Mean=32792 Var=1.053e+08
MPEG4_OfficeCamera_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.2537k}$	<b>Normal+Normal</b> Mean=5406.5 Var=2.1443e+05
MPEG4_OfficeCamera_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.2176k}$	<b>Normal+Normal</b> Mean=6658.8 Var=7.159e+05
MPEG4_OfficeCamera_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.2256k}$	<b>Normal+Normal</b> Mean=23738 Var=5.0029e+06
H263_Film_16 (16kbps)	H263 CBR	Frame 1/25s	$(k+1)^{-0.9196}$	<b>Gamma+LogNormal</b> Mean=428.0631 Var=6.0393e+04
H263_Film_64 (64kbps)	H263 CBR	Frame 1/25s	$(k+1)^{-0.519}$	<b>Normal+Normal</b> Mean=1124.4 Var=1.5475e+05

**Modèle de source temps réel**

H263_Film_256 (256kbps)	H263 CBR	Frame 1/25s	$(k + 1)^{-0.6343}$	<b>Normal+Normal</b> Mean=4536.9 Var=2.4194e+06
H263_Cartoon_16 (16kbps)	H263 CBR	Frame 1/25s	$e^{-1.2036k}$	<b>Gamma+LogNormal</b> Mean=560.0774 Var=1.9425e+05
H263_Cartoon_64 (64kbps)	H263 CBR	Frame 1/25s	$e^{-7.535k}$	<b>Normal+Normal</b> Mean=1081.1 Var=1.5879e+05
H263_Cartoon_256 (256kbps)	H263 CBR	Frame 1/25s	$e^{-1.1288k}$	<b>Normal+Normal</b> Mean=4246.3 Var=2.3814e+06
H263_News_16 (16kbps)	H263 CBR	Frame 1/25s	$(k + 1)^{-3.03}$	<b>Gamma+LogNormal</b> Mean=437.5441 Var=7.212e+04

Tableau 14 - paramètres du générateur M/G/∞ pour la vidéo

Vidéo Source	CODEC	Time Slot	Corrélation	Size Distribution Byte
H263_News_64 (64kbps)	H263 CBR	Frame 1/25s	$e^{-0.2302\sqrt{k}}$	<b>Normal+Normal</b> Mean=1228.5 Var=1.8851e+05
H263_News_256 (256kbps)	H263 CBR	Frame 1/25s	$0.535e^{-0.0205k}$	<b>Normal+Normal</b> Mean=4875.1 Var=2.903e+06
H263_ParkingCamera_16 (16kbps)	H263 CBR	Frame 1/25s	$e^{-13.52k}$	<b>Normal+Normal</b> Mean=414.2058 Var=2.0203e+03
H263_ParkingCamera_64 (64kbps)	H263 CBR	Frame 1/25s	$e^{-15.505\sqrt{k}}$	<b>Normal+Normal</b> Mean=1585.4 Var=7.415e+03
H263_ParkingCamera_256 (256kbps)	H263 CBR	Frame 1/25s	$e^{-12.4736\sqrt{k}}$	<b>Normal+Normal</b> Mean=6318.1 Var=3.7874e+04
H263_OfficeCamera_16 (16kbps)	H263 CBR	Frame 1/25s	$e^{-15k}$	<b>Gamma+LogNormal</b> Mean=444.9297 Var=1.0494e+05
H263_OfficeCamera_64 (64kbps)	H263 CBR	Frame 1/25s	$0.896e^{-1.092\sqrt{k}}$	<b>Normal+Normal</b> Mean=1565.7 Var=4.1747e+04
H263_OfficeCamera_256 (256kbps)	H263 CBR	Frame 1/25s	$(k + 1)^{-1.0095k}$	<b>Gamma+LogNormal</b> Mean=6192.6 Var=4.2687e+05

H263_Sport_16 (16kbps)	H263 CBR	Frame 1/25s	$e^{-0.3568\sqrt{k}}$	<b>Gamma+LogNormal</b> Mean=653.3933 Var=1.5125e+05
H263_Sport_64 (64kbps)	H263 CBR	Frame 1/25s	$e^{-0.4663\sqrt{k}}$	<b>Normal+Normal</b> Mean=902.674 Var=1.0692e+05
H263_Sport_256 (256kbps)	H263 CBR	Frame 1/25s	$e^{-0.4476\sqrt{k}}$	<b>Normal+Normal</b> Mean=3392.3 Var=1.5872e+06

### 5.3 Conclusion

Nous avons présenté dans ce chapitre les modèles de trafic pour les sources de trafic temps réel. Les deux grandes catégories de trafic temps réel sont les applications audio et les applications vidéo.

Les modèles de trafic pour les applications audio sont des modèles stochastiques de type ON-OFF. Nous avons présenté deux modèles : le processus ON-OFF avec débit constant et le modèle IPP (Interrupted Poisson Process) avec débit exponentiel. Bien que l'avantage de IPP sur le processus ON-OFF est de pouvoir en déduire un modèle mathématique agrégé, il a été montré par des simulations sur DHS (Distributed Hybrid Server) que la superposition des CODEC de type ON-OFF converge à priori beaucoup plus rapidement vers un processus de Poisson que la superposition de CODEC de type IPP.

La deuxième catégorie de trafic temps réel est la vidéo. Les applications vidéo sont très nombreuses. Nous avons distingué deux types d'applications : Les applications à débit régulier qui peuvent être modélisées par un processus MMPP (Markov Modulated Poisson process) et les vidéo à débit plus variable qui génèrent un trafic corrélé (M-JPEG et MPEG) qui peuvent être modélisées par un processus d'occupation d'un file de type M/G/∞. Le modèle M/G/∞ est intéressant car il est possible d'en déduire un modèle mathématique agrégé.

L'étude des sources vidéo nous montre la nécessité d'utiliser des mixages de distribution.



## 6 Modèle de source non temps réel

---

### 6.1 HTTP

Le trafic Web (HTTP) est le trafic majoritaire sur Internet (70%). Il y a eu plusieurs études de modélisation sur le trafic HTTP. La modélisation du trafic HTTP nécessite un modèle de la structure d'une page web, et du comportement de HTTP.

Nous allons présenter les différents modèles retenus. Nous rappelons que la structure des modèles proposés suit l'approche générale présentée dans le Chapitre 2. Cette approche comporte trois niveaux, le premier est le niveau de la session qui modélise l'arrivée de tous les clients dans le système (une distribution de Poisson [13]), le deuxième est le niveau Application qui représente la densité d'information au cours du temps et le niveau le plus élémentaire le niveau paquet.

#### 6.1.1 Modélisation d'une page web

Une page web est composée d'image, de texte et d'objet multimédia. Une page est caractérisée par :

- la taille de l'objet principal de la page : la page HTML elle-même,
- la taille des objets hypertextes à télécharger (image, multimédia, ...): liens à partir de la page principale,
- le nombre d'objet hypertextes à télécharger.

Plusieurs études ont été menées sur des serveurs web pour donner les distributions des différents documents. Les distributions proposées sont de type sous-exponentiel. On présente ici deux serveurs (inria.fr et clark.net) étudiés par Niclausse [23] :

Tableau 15 - Distribution des pages WEB www.inria.fr

Variable	Distribution	Moyenne	Ecart type
Taille Page Web	Lognormale(8.43, 2.31)	19 kbyte	124 Kbyte
Document HTML	Lognormale(7.97, 0.76)	4.6 kbyte	11 Kbyte
Images	Lognormale(7.45, 4.04)	12 kbyte	35 kbyte
Autres Documents	Inverse Gaussienne (49450, 1017)	49 kbyte	348 kbyte

Tableau 16 - Distribution des pages WEB www.clark.net

<b>Variable</b>	<b>Distribution</b>	<b>Moyenne</b>	<b>Ecart type</b>
<b>Taille Page Web</b>	Weibull(0.65, 0.0014)	39 kbyte	86 Kbyte
<b>Document HTML</b>	Lognormale(7.8, 1.44)	5.8 kbyte	21.3 Kbyte
<b>Images</b>	Lognormale(8.25, 4)	13.5kbyte	24kbyte
<b>Autres Documents</b>	Lognormale(8.25, 4)	49 kbyte	258 kbyte

Une étude intéressante faite par Jalpa [24] montre qu'un mélange de trois distributions (Inverse Gaussienne, Lognormal et Géométrique) permettent de caractériser les documents. Bien sûr, suivant le serveur, les paramètres de chaque distribution varient ainsi que le poids associée à chaque distribution.

### **6.1.2 Comportement HTTP**

Le comportement de téléchargement dépend de la version HTTP utilisée par le navigateur. Dans la version 1.0, HTTP ouvre une connexion pour chaque objet dans la page ce qui n'est pas très efficace. Alors que dans la version 1.1 HTTP permet d'utiliser une seule connexion HTTP pour télécharger plusieurs objets, et autorise le téléchargement en parallèle avec plusieurs connexions HTTP. En version 1.1, les connexions persistantes sont autorisées afin d'améliorer les performances. Les connexions persistantes ne dépendent pas uniquement du client. Les serveurs HTTP ont plutôt tendance à limiter les connexions persistantes afin de pouvoir accepter des nouvelles connexions et également améliorer les performances du serveur HTTP (les sessions HTTP persistantes utilisent des ressources en mémoire). Les relais HTTP (proxy) limitent également les connexions persistantes afin de pouvoir accepter de nouvelles connexions.

### **6.1.3 Caractéristiques du trafic**

Le protocole HTTP suit le comportement ON/OFF [2] :

Tableau 17 - unité des périodes

<b>Période</b>	<b>Expression</b>
<b>ON</b>	En K octets : téléchargement des données
<b>OFF actif</b>	En seconde : Affichage (généralement inférieur à 1 s), cette période n'est pas toujours prise en compte dans les modèles.
<b>OFF inactive</b>	En seconde : Lecture des données (30 s)



**Période ON** : elle commence dès qu'une ouverture de connexion HTTP (GetRequest) est demandée par l'utilisateur (click sur un lien, une image ou première connexion sur un site). La première requête TCP est générée par l'utilisateur, les autres objets de la page sont ensuite téléchargés par le protocole. Les données sont alors transférées selon la taille des données et la bande passante. La durée de cette période dépend du volume de données à télécharger. La période ON est exprimée en Kb.

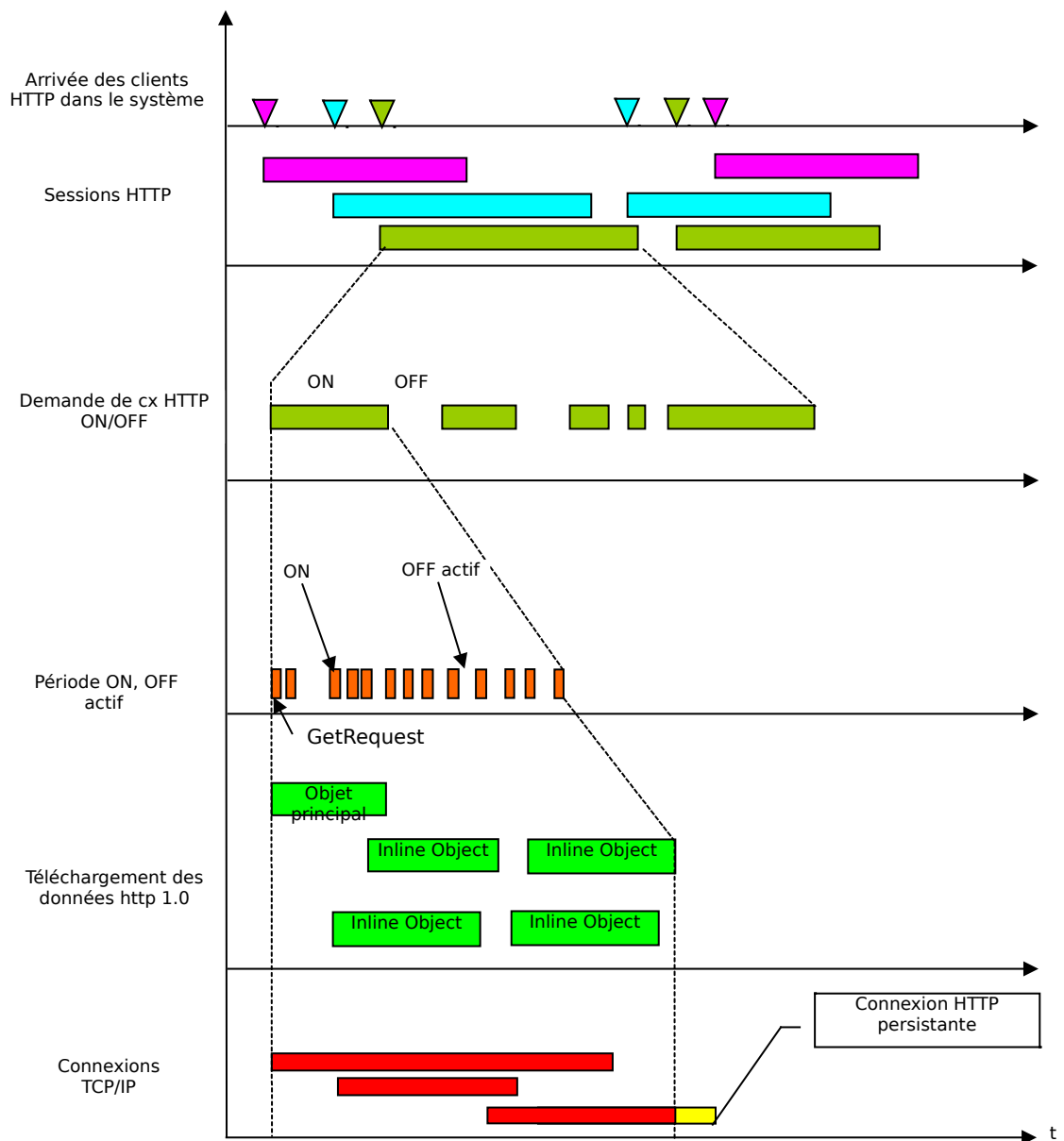


Figure 11 - Comportement HTTP

**Période OFF active** : elle représente le temps nécessaire à l'interprétation des données par le terminal (en moyenne inférieur à 1s). Pour les composants embarqués, cette période peut être plus importante (jusqu'à 30 secondes). Cette période représente aussi le temps entre deux requêtes de l'utilisateur mais elle n'est pas toujours prise en compte dans les modèles de trafic. La queue de la distribution est d'autant plus lourde (distribution de Pareto, Weibull) que les pages contiennent davantage d'objet graphique que de texte.

**Période OFF** : Tous les objets ont été téléchargés et l'utilisateur lit les informations.

#### 6.1.4 Modèle de trafic

Modèle de trafic ON/OFF pour HTTP. Les différentes périodes sont regroupées en tableau afin de faire la comparaison des distributions selon les modèles. Pour maintenir la cohérence, il est nécessaire de ne pas dissocier les périodes d'un même modèle.

Tableau 18 - Arrivée des sessions HTTP

<b>Modèle</b>	<b>Distribution inter-session</b>
[9]	Poisson
[11]	Poisson
[12]	Poisson
[14]	Poisson
[16]	Pareto

Tableau 19 - Période ON en Kb\*

<b>Modèle</b>	<b>Distribution</b>	<b>Paramètres</b>
[2][3]	Pareto	$\lambda = 1000$ (taille minimum fichier), $\alpha = 1.1-1.5$
[4]	Weibull	$a = e^{4.4} \cdot e^{4.6}$ , $\lambda = 0.91-0.77$
[5]	Pareto	$\lambda = 100$ (ou plus), $\alpha = 1.0-1.3$
[12]	Pareto	$\alpha = 1.5 - 1.7$ , $\beta = 3000-31000$
[14]	Pareto	Number of packet per page, $\alpha = 1.1$ , $K = 2.27$

Tableau 20 - Période OFF en seconde\*

<b>Modèle</b>	<b>Distribution</b>	<b>Paramètres</b>
[2][3]	Pareto	Kw=1 sec, aw=1.5
[4]	Pareto	Kw=30 sec, aw=0.9 - 0.58
[5]	Pareto	Kw=30 sec, aw=1.5
[10]	Weibull	Mean 39.5s, Median 11s, Stdev 92.6
[11]	-	moyenne 1000s-1900s median 15s
[12]	Gamma	25-35 s, stdev 133-147s
[14]	Geometric	Mean 120
[16]	Weibull	$\alpha=1.46$ , $\beta=0.38$

\* : Les paramètres (Pareto, Weibull), sont souvent exprimés différemment selon les sources. Il est toujours possible de retrouver la moyenne et la variance.

Tableau 21 - OFF actif en seconde

<b>Modèle</b>	<b>Distribution</b>	<b>Paramètres</b>
[2][3]	Weibull	a=0.328, b=1.46
[4]	Weibull	a=1.5, b=0.5
[5]	Weibull	-

Tableau 22 - Nombre de page par session

<b>Modèle</b>	<b>Distribution</b>	<b>Paramètres</b>
[12]	Lognormal	moyenne 22-25, dev 78-166
[14]	Geometric	Mean 5

Tableau 23 - Taille des requêtes web en K octets

<b>Modèle</b>	<b>Distribution</b>	<b>Paramètres</b>
[10]	Lognormal	Mean 360 byte, Median 344 byte, Stdev 106.5 byte
[11]	Bimodal	250B, 1k mean 320B median 344 B

D'autres modèles ON/OFF tiennent compte de la distribution des objets à télécharger lorsqu'un l'utilisateur fait une requête à partir de son navigateur WEB. Ces modèles donnent les caractéristiques de téléchargement pour la période ON.

Tableau 24 - Modèles avec prise en compte des objets « main » et « inline »

<b>Paramètre du Modèle</b>	<b>Période ON en K (requête de page web)</b>		
	Taille objet principal en Kb	Taille objet en ligne en Kb	Nombre d'objets en ligne en Kb
<b>Source</b>			
[10]	Lognormal Moyenne 10 kB Médiane 6 kB Var 25 kB	Lognormal Moyenne 7.7 kB Médiane 2 kB Var 126 kB	Gamma Moyenne 5.5 kB Médiane 2 kB Var 11.4 kB
[11]	Pareto $\alpha=0.85 - 0.97$ Médiane 2 -2.4kB	Pareto $\alpha=1.12- 1.39$ Médiane 1.2-2kB	Pas de distribution Moyenne 2.8, 3.2 Médiane 1
[16]	Pareto $\alpha=1$ k=1000		Pareto $\alpha=2.43$ k=1

## 6.2 Trafic HTTP dans les réseaux 2.5 et 3 G

Les modèles présentés jusqu'à maintenant pour le trafic HTTP sont tous fondés sur des mesures effectuées sur des réseaux LAN. Le comportement des utilisateurs, le type d'applications déployées sur les réseaux de mobiles sont différents des réseaux fixes à cause de la grande différence en terme de bande passante et de coût de communication qui est plus cher dans les réseaux mobiles [7, 8]. Nous allons présenter dans la suite les modèles proposés dans la littérature pour le trafic HTTP sur les réseaux de mobiles.

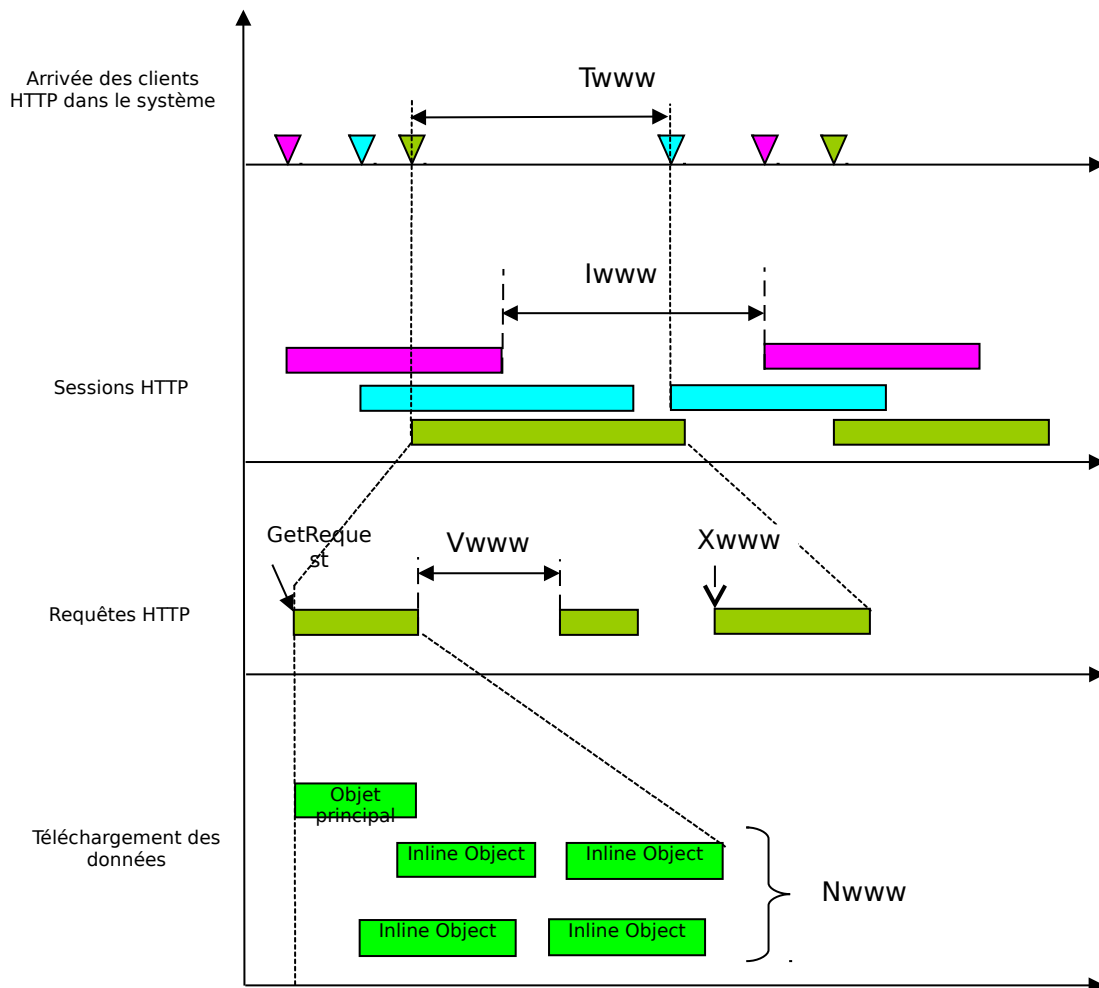


Figure 12 - Modèle http 2.5 et 3G

### 6.2.1 Trafic HTTP dans les réseaux GPRS

Le trafic WWW résultant à la station de base est caractérisé par les variables aléatoires suivantes :

**T<sub>www</sub>** : le temps entre session (distribution de Poisson),

**I<sub>www</sub>** : le temps entre deux sessions WWW (moyenne 1000-1900 secondes, médiane 15 s),

**X<sub>www</sub>** : le nombre de requêtes web par session WWW,

**V<sub>www</sub>** : le temps de visualisation de la requête web,

**M<sub>www</sub>** : la taille de l'objet principal,

**N<sub>www</sub>** : le nombre d'objets en ligne,

**O<sub>www</sub>** : la taille d'un objet en ligne,

**R<sub>www</sub>** : la taille de GetRequest.

Les distributions présentées dans le travail de Choi [10] sont les suivantes :

Pour la variable V<sub>www</sub>, une distribution de Weibull tronquée à 15 minutes (une nouvelle session est commencée si le temps de visualisation dépasse les 15 minutes).

Tableau 25 - Modèles LAN empiriques pour GPRS [10]

<b>Variable</b>	<b>Loi</b>	<b>Moyenne E</b>	<b>Ecart-type <math>\sigma</math></b>
<b>V<sub>www</sub></b>	Weibull	39.5 s	92.6s
<b>N<sub>www</sub></b>	Gamma(0.24, 23.42)	5.55	11.4
<b>M<sub>www</sub></b>	Lognormal(1.31, 1.41)	10 kB	25 kB
<b>O<sub>www</sub></b>	Lognormal(-0.75,2.36)	7.7 kB	126 kB
<b>R<sub>www</sub></b>	Lognormal(5.84,0.29 )	360 B	106 B
<b>X<sub>www</sub></b>	Lognormal(1.8,1.65)	25 pages	100 pages

La taille moyenne de la requête web est d'environ 50 kB.

### 6.2.2 Trafic HTTP dans les réseaux UMTS

L'étude [1] propose un modèle de trafic UMTS fondée sur l'observation d'un réseau LAN (IP + routeur Dial-in). Ce modèle de trafic est particulièrement intéressant car il prend en compte les bandes passantes de l'UMTS. Un modèle de trafic agrégé basé sur BMAP (Batch Markovian Arrival Process) est également proposé.

Pour le temps inter-arrivée au niveau session, une distribution lognormal a été dérivée avec  $\mu=0.6061$  et  $\sigma^2 = 7.5330$ . Les tableaux ci-après montrent les distributions selon trois niveaux:

Tableau 26 - Distribution du volume de session HTTP en Kb [1]

<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 kbps</b>
Lognormal( $\mu,\sigma^2$ )	(5.1613 ;13.8210)	(5.2943 ;14.1839)	(5.4197 ;14.4979)

Tableau 27 - Distribution des inter-arrivées des connexions et volume de donnée [1]

		<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 kbps</b>
<b>HTT P</b>	Inter-arrivée des sessions en seconde	Lognormal( $\mu,\sigma^2$ )	(-0.3278;4.4807)	(-0.6907;4.8483)	(-1.1997;5.3259)
	Volume de données Kb	Lognormal( $\mu,\sigma^2$ )	(2.8651;12.6098)	(2.8072;13.0871 )	(2.7888;13.4888)

Tableau 28 - Distribution des inter-arrivées des paquets en seconde [1]

	<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 kbps</b>
<b>HTTP</b>	Lognormal( $\mu,\sigma^2$ )	(-2.7597;3.5448)	(-3.2588;3.8720)	(-3.9599;4.3342)

**Conclusion** : Les différents modèles de trafics HTTP utilisent des distributions relativement proches, certains modèles permettent de définir plus finement le comportement du trafic dans la période ON avec la prise en compte des OFF inactifs. L'ensemble des modèles s'accorde pour l'utilisation des distributions à queue lourde (Heavy-Tailed) à cause des comportements LRD. La définition de modèle de trafic GPRS est faite de façon empirique ce qui ne garantit pas un modèle fiable (le nombre de page lu semble peut probable)

Le modèle UMTS pour la bande passante 2Mb/s n'est pas disponible car les données ne sont pas encore très fiables [1].

## 6.3 FTP

Le comportement des sessions FTP est déduit de mesure sur un réseau LAN. Les modèles GPRS et UMTS ont été déduits de façon empirique. Le modèle est de type ON/OFF :

- **Période ON** : téléchargement
- **Période OFF** : période de non activité

### 6.3.1 Modèles du trafic FTP dans les réseaux LAN - GPRS

Tableau 29 – Période ON/OFF en seconde pour LAN, GPRS [2]

	<b>LAN / GPRS</b>
<b>ON</b>	Pareto $0.9 < \alpha < 1.1$
<b>OFF</b>	Weibull

### 6.3.2 Modèles du trafic FTP dans les réseaux UMTS

Tableau 30 – Distribution du volume de session FTP en Kb [1]

<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 kbps</b>
Lognormal( $\mu, \sigma^2$ )	(5.1613 ;13.8210)	(5.2943 ;14.1839)	(5.4197 ;14.4979)

Tableau 31 - Distribution des inter-arrivées des connexions et volume de donnée [1]

		<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 KBPS</b>
<b>FTP</b>	Inter-arrivée des sessions en seconde	-	-	-	-
	Volume de données Kb	Lognormal( $\mu, \sigma^2$ )	(3.3433 ;13.9698)	(3.3020 ;14.7826)	(3.3435 ;15.3229)



Tableau 32 - Distribution des inter-arrivées des paquets en seconde [1]

	<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 kbps</b>
<b>FTP</b>	Lognormal( $\mu, \sigma^2$ )	(-3.0691 ; 3.8055)	(-3.2656 ; 3.9347)	(-3.4160 ; 4.0337)

## 6.4 Email

L'Email est l'application la plus utilisée après le WEB. Les études sur les caractéristiques du trafic Email sont moins importantes et peu de modèles sont disponibles. Le comportement du trafic est du type ON/OFF.

**Période ON** : transmission de l'Email.

**Période OFF** : aucun Email n'est transmis, lecture, rédaction d'un Email, réponse à un Email.

### 6.4.1 Modèles du trafic Email dans les réseaux LAN - GPRS

Tableau 33 - Modèles LAN et empirique pour GPRS

	<b>ON - taille en K octet</b>	<b>OFF en seconde</b>
<b>[17]</b>	Cauchy $f(x) = \left\{ \pi\beta \left[ 1 + \left( \frac{x-\alpha}{\beta} \right)^2 \right] \right\}^{-1}$ $\beta=1.0, \alpha=0.8$	-
<b>[2]</b>	Weibull $F_e(x_e) = \begin{cases} 1 - e^{-ek_1 x_e^{c_1}} \\ 1 - e^{-ek_2 x_e^{c_2}} \end{cases}$ $C1=1.2 - 3.2 (m=2.04)$ $C2=0.31 - 0.46 (m=17.64)$ $K1= 14.0 - 21.0 (m=17.64)$ $K2=2.8 - 3.4$	Pareto $K_e=30 - 60 \text{ s,}$ $\alpha_e=0. - 1.5$

## 6.4.2 Modèles du trafic Email dans les réseaux UMTS

Modèle dépendant de la bande passante :

Tableau 34 - Distribution du volume de session Email en Kb [1]

<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 kbps</b>
Lognormal( $\mu, \sigma^2$ )	(5.1613 ;13.8210)	(5.2943 ;14.1839)	(5.4197 ;14.4979)

Tableau 35 - Distribution des inter-arrivées des connexions et volume de données [1]

		<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 kbps</b>
<b>Email</b>	Inter-arrivé des sessions en seconde	Pareto(k, $\alpha$ )	(14.4360 ;2.1345)	(15.1334 ;2.1254)	(16.0229 ;2.1223)
	Volume de données Kb	Lognormal( $\mu, \sigma^2$ )	(3.2677 ;13.2369)	(3.2799 ;13.5579)	(3.3084 ;13.8518)

Tableau 36 - Distribution des inter-arrivées des paquets en seconde [1]

	<b>Distribution</b>	<b>64 kbps</b>	<b>144 kbps</b>	<b>384 kbps</b>
<b>Email</b>	Lognormal( $\mu, \sigma^2$ )	(-3.3678 ;2.4220)	(-3.7152 ;2.6410)	(-4.1058 ;2.8902)

## 6.5 WAP (Wireless Application Protocol)

Le WAP (Wireless Application Protocol) est une couche de protocole identique dans sa structure à TCP/IP. La technologie WAP a été conçue pour fonctionner avec toutes les technologies sans fil (GSM, GPRS, UMTS, Bluetooth). Cette technologie prend en compte les ressources limitées du terminal (CPU, énergie, affichage) grâce à une architecture qui simplifie largement l'affichage. Les terminaux WAP se connectent aux serveurs au travers d'un relais WAP qui joue le rôle de passerelle. Lorsqu'un terminal WAP se connecte à un serveur WEB HTTP, la passerelle fait un encodage des informations HTTP vers le terminal en simplifiant les données.

### 6.5.1 Structure du réseau

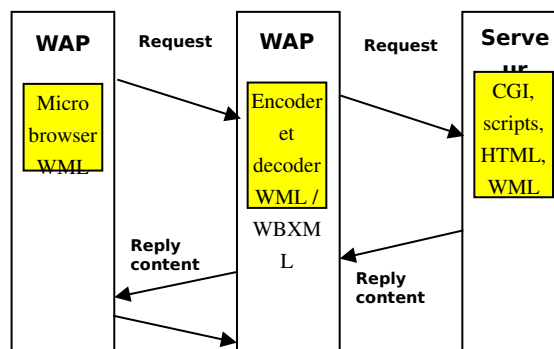


Figure 13 - Architecture WAP

Les serveurs WML offrent des pages WEB qui n'ont pas besoin d'être encodées. Les pages WML possèdent des structures simplifiées : réduction des objets graphiques et multimédias, le texte est également simplifié. La taille des pages est donc considérablement réduite. Le facteur de réduction est estimé à 40%. Les terminaux WAP peuvent communiquer dans le mode connecté (TCP) et non connecté (UDP). Le mode connecté est utilisé par la majeure partie des applications développées autour du WAP, dans la structure de la pile IP ce mode correspond à la couche WTP (Transaction Layer). Les applications WAP sont classées, les applications qui utilisent TCP/IP (WTP) sont de classe 2.

### 6.5.2 Caractéristiques

Une session WAP peut être résumée de la manière suivante :

- demande de page (ou « deck »),
- taille des paquets en upload et download,
- temps de lecture avant une nouvelle requête pour une nouvelle page,
- temps de réponse du réseau (ne dépend pas de l'utilisateur).

Le WAP offre différentes applications :

- Email : différents services WAP offrent la possibilité de lire et d'écrire des E-mail,
- News : services d'accès aux informations (article de presse, ...),
- M-Commerce : achat en ligne,
- Commun : trafic de plusieurs types.

### 6.5.3 Modèles de trafic

La structure du modèle WAP est proche du modèle HTTP mis à part la taille des requêtes HTTP. Dans un terminal mobile, les objets « inline » dépendent de la capacité du terminal, ce qui influe sur la distribution. Selon [6], la distribution des objets « inline » à télécharger est de type lognormal. Les modèles suivants ont été définis à partir de mesure sur un réseau WAP, sur le lien download.

Tableau 37 - Distribution période WAP [18]

	Distribution
<b>OFF</b>	géométrique
<b>GetRequest</b>	exponentielle
<b>Taille de la page</b>	lognormal

Tableau 38 - Paramètres des périodes WAP [18]

Paramètre	Moyenne	Ecart type
<b>News</b>		
OFF en seconde	11.4	16.5
Taille du paquet Get Request en octet	82.1	26.2
Taille de la page en octet	638.4	389.9
<b>E-Mail</b>		
OFF en seconde	10.0	22.2
Get Request en octet	112.0	57.7
Taille de la page en octet	582.9	260.0
<b>M-Commerce</b>		
OFF en seconde	10.7	13.1
Get Request en octet	84.3	16.5
Taille de la page en octet	641.0	342.9
<b>Commun</b>		
OFF en seconde	13.8	116.8
Get Request en octet	108.2	84.7
Taille de la page en octet	511.0	368.0

## 7 Modèle de trafic agrégé

Pour la superposition de N source, on peut soit utiliser N sources identiques indépendantes, soit utiliser des modèles agrégés. Cependant ceux-ci ne permettent pas de tenir en compte du comportement des sources réelles de trafic. En effet, il est intéressant de considérer les fluctuations du nombre de sources actives dans le temps.

Nous allons présenter dans ce chapitre les modèles agrégés connus pour les sources de trafic unitaires que nous avons vu dans le chapitre précédent.

### 7.1 Agrégation des sources de trafics Audio

Nous disposons des modèles IPP et ON-OFF pour représenter les flux de type voix unitaire dans l'Internet. L'intérêt du modèle IPP est de pouvoir l'agréger mathématiquement en processus MMPP-M. Le modèle ON-OFF représente cependant mieux la source réelle et son caractère déterministe pendant la période ON (la période dans laquelle les données digitalisées sont empaquetées et émises à intervalles réguliers).

#### 7.1.1 Modèle MMPP-N+1

Le processus MMPP-N+1 (Markov Modulated Poisson Process : processus de poisson modulé par un processus de Markov à N+1 état), est utilisé pour représenter la superposition de N (constant) sources homogènes et identiques de type IPP.

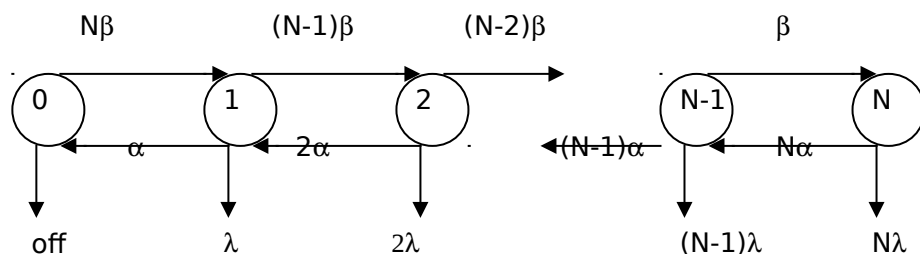


Figure 15 - Modèle MMPP-N+1

Les paramètres  $\alpha$ ,  $\beta$  et  $\lambda$  sont définis de la manière suivante :

-La durée moyenne d'une période d'activité :  $T_{on}=1/\alpha$  .

-La durée moyenne d'une période d'inactivité :  $T_{off}=1/\beta$  .

-Le taux d'émission durant une période d'activité :  $\lambda =1/T$  paramètre de la loi de Poisson.

La probabilité d'être dans l'état  $i$  et les caractéristiques moyennes du processus sont données par :

$$\pi_i = P(\bar{\lambda} = i\lambda) = \frac{N!}{i!(N-i)!} P_{On}^i (1-P_{On})^{N-i} \quad (16)$$

$$E(\bar{\lambda}) = N\lambda P_{On} \quad (17)$$

$$Var(\bar{\lambda}) = N\lambda^2 P_{On}(1-P_{On}) \quad (18)$$

Avec :

$$P_{On} = \frac{\beta}{\beta + \alpha} \quad (19)$$

On peut utiliser un processus MMPP-2 pour approcher un processus MMPP-N+1. l'état UL(UnderLoad) du processus MMPP-2 est associé à l'ensemble des états 0 à M du processus MMPP-N+1, tandis que l'état OL (OverLoad) est associé aux états M+1 à N.

Plusieurs méthodes existent pour déterminer M. Connaissant M, un calcul possible des paramètres du processus MMPP-2 est donné par :

$$\lambda_{ul} = \lambda \sum_{i=0}^M i \frac{\pi_i}{\pi_{ul}} \quad \lambda_{ol} = \lambda \sum_{i=M+1}^N i \frac{\pi_i}{\pi_{ol}} \quad (20)$$

$$\pi_{ul} = \sum_{i=0}^M \pi_i \quad \pi_{ol} = \sum_{i=M+1}^N \pi_i \quad (21)$$

### 7.1.2 Superpositions des sources de trafic

Théoriquement, la superposition d'un nombre très grand de sources conduit à un processus de Poisson. Dans une étude réalisée au LAAS-CNRS par simulation événementielle [20], nous avons constaté que le processus de Poisson constitue une approximation satisfaisante pour les CODEC audio à partir d'un nombre assez faible de sources de trafic (à partir de 15 sources).

### 7.1.2.1 CODEC Audio G729

Le CODEC G729 peut être modélisé par un processus ON-OFF ou un processus IPP. Des résultats d'étude montrent que dans le cas où le CODEC G729 est modélisé par un processus ON-OFF. On peut à partir de 15 sources approcher l'agrégat des flux G729 par un flux exponentiellement distribué. En dessous de 10 sources, les délais et la charge de l'agrégat de sources G729 sont nettement supérieurs à celui de la source exponentielle.

Dans le cas d'un CODEC G729-IPP la convergence vers une distribution exponentielle est beaucoup plus lente que dans le cas de l'utilisation du processus ON-OFF (au moins 100 sources sont nécessaires). On rappelle que le processus IPP est un modèle facile à agréger mathématiquement avec le Modèle MMPP-N+1.

### 7.1.2.2 CODEC Audio G711 et G726

Des résultats comparables à ceux du CODEC G729 ON-OFF ont été constatés pour les CODEC G711 ON-OFF et G726 ON-OFF. La superposition de 15 sources peut être représenté par une source exponentielle de même débit, tant en terme de distribution des inter-arrivées qu'en terme de charge engendrée dans une file d'attente à service exponentiel.

## 7.2 Agrégation des sources de trafics Vidéo

Nous avons vu que le trafic vidéo peut être modélisé soit par le processus MMPP-N+1 pour le trafic régulier (une propriété qui ne concerne qu'un nombre limité de sources de trafic vidéo) soit par le processus M/G/∞ qui représente bien la corrélation engendré par le trafic vidéo M-JPEG et MPEG. Nous allons présenter le modèle mathématiques d'agrégation du modèle M/G/∞ ainsi que les résultats statistiques d'agrégation de ce modèle avec DHS.

### 7.2.1 Agrégation de modèles M/G/∞

Nous allons nous intéresser à l'agrégation du modèle M/G/∞. Une propriété importante du modèle M/G/∞ est d'être stable par agrégation. En effet, si l'on considère N modèles M/G/∞ définis par :

$\lambda_i$  : Paramètre de la loi de poisson.

$F_i(x)$  : Fonction de répartition associée au temps de service  $\sigma_i$ .

L'agrégation de ces modèles est aussi un processus M/G/∞ défini par :

$$\lambda = \sum \lambda_i \quad (22)$$

$$F(x) = \sum_{i=1}^N \frac{\lambda_i}{\lambda} F_i(x) \quad (23)$$

ou  $F(x)$  représente la fonction de répartition associée à  $G$ .

Cette caractérisation peut donc permettre de générer directement un flux vidéo agrégé à l'échelle image ou GOP.

### **7.2.2 Superpositions des sources M/G/ $\infty$**

L'étude faite au LAAS-CNRS avec le simulateur DHS [20] montre que dans le cas des CODEC vidéo la superposition de sources M/G/ $\infty$  (non LRD) tend rapidement (à partir de 10 sources) vers une distribution exponentielle.

## **7.3 Agrégation des sources de trafics Données**

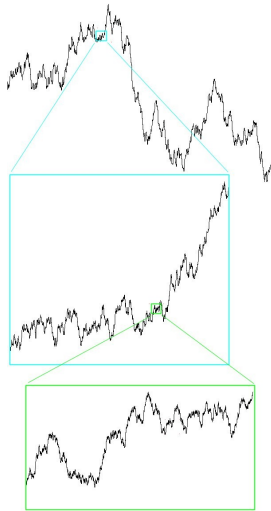
Différentes études effectuées sur des réseaux LAN ou WAN, ont mis en évidence le caractère autosimilaire et LRD (long range dependency) de leur trafic.

Les traces étudiées représentent le nombre de paquets ou de bits entrant dans le réseaux en fonction du temps. La propriété d'autosimilarité du trafic signifie qu'une trace générée a le même comportement en effectuant un zoom, quel que soit l'échelle de temps choisie. Dans cette partie, nous allons étudier la propriété d'autosimilarité ainsi que les modèles de génération de trafic autosimilaire. D'autres modèles de trafic de données agrégés pour les réseaux de mobiles seront aussi présentés.

### **7.3.1 Propriétés d'autosimilarité**

L'autosimilarité d'un objet renvoie au fait que l'objet peut être décomposé en sous unité, puis en sous unité, qui possèdent les mêmes propriétés statistiques que l'objet global.





Dans cet exemple de source de trafic, si l'on observe la source à différente échelle de temps, on remarque des propriétés statistiques similaires, quelle que soit l'échelle d'observation.

Exemple de représentation de l'autosimilarité

*Définition :* Soit  $(X_n, n \in \mathbb{N})$  un processus stationnaire. On note  $(X_k^m)_{k \in \mathbb{N}}$  son processus agrégés d'ordre  $m$  défini comme suit :

$$X_k^{(m)} = \frac{1}{m} (X_{(k-1)m+1} + \dots + X_{km}) \quad (24)$$

le processus  $(X_n)_{n \geq 0}$  est dit autosimilaire de paramètre  $H$  si  $\forall m$  entier :

$$X_k^{(m)} = m^{1-H} (X_{(k-1)m+1} + \dots + X_{km})_{k \in \mathbb{N}} \stackrel{L}{=} (X_n)_{n \in \mathbb{N}} \quad (25)$$

Le paramètre de Hurst  $H$  associé à un processus autosimilaire, représente intuitivement le « Zoom » temporel et en amplitude.

La corrélation d'un processus autosimilaire s'exprime aussi par rapport au paramètre de Hurst. En effet, la fonction d'autocorrélation associée à un processus autosimilaire  $(X_n)_{n \geq 0}$  de paramètre  $H$  est de la forme :

$$\rho_{X_n}(k) \approx_{k \rightarrow \infty} H(2H-1)k^{2H-2} \quad (26)$$

On en déduit qu'un processus  $(X_n)_{n \geq 0}$  est à dépendance longue (LRD) si  $\frac{1}{2} < H < 1$  (le paramètre d'autocorrélation est égal à  $\beta = 2 - 2H$ ).

On remarque qu'il suffit donc d'estimer le paramètre de Hurst,  $H$ , d'une trace si l'on veut déterminer ses caractéristiques d'autosimilarité et de corrélation.

Plusieurs méthodes statistiques existent pour estimer le paramètre de Hurst  $H$ . Les trois méthodes des plus utilisées sont *Variance-time plot*, *R/S plot*, *méthode des ondelettes*.

Les différentes études effectuées montrent que le choix du paramètre Hurst tel que  $0.7 \leq H \leq 0.9$ , permet de bien représenter le comportement du trafic agrégé au niveau paquet ou bit.

### **Corrélation à long terme (LRD) :**

Un exposant  $H$  supérieur à 0.5 révèle des phénomènes de persistance, c'est à dire que l'évolution d'une série temporelle tend à suivre des tendances. Les séries persistantes ont une mémoire à long terme, c'est à dire qu'il existe une corrélation à long terme entre les événements actuels et les événements futurs.

### **Corrélation à court terme (SRD) :**

A l'inverse, un exposant inférieur à 0.5 révèle un phénomène d'anti-persistance. Dans ce cas les accroissements successifs tendent à être négativement corrélés. Une augmentation de la variable tend à être suivi d'une diminution, et vice-versa.

## **7.3.2 Modèles de génération de trafic autosimilaire**

Il existe plusieurs modèles pour représenter le trafic autosimilaire. Nous présentons ici deux modèles, le modèle FGN (Fractional Gaussian Noise) et le modèle  $M/G/\infty$ .

### **7.3.2.1 Le modèle FGN (Fractional Gaussian Noise)**

Ce modèle permet de générer du trafic autosimilaire et LRD à partir des caractéristiques suivantes :

- Moyenne  $m$  et variance  $\sigma^2$  associées à la taille des échantillons,  $X_n$ , générées à l'instant  $n$ .
- Corrélation associée aux données définie par le paramètre de Hurst  $0.5 < H < 1$ .

Le processus FGN représente la taille des données par une loi normale  $N(m, \sigma^2)$ . Le processus FGN représente bien un trafic autosimilaire mais pose tout de même quelques problèmes au niveau algorithmique. L'algorithme de FGN utilise la Transformée de Fourier Inverse ce qui empêche une représentation générative. En effet l'algorithme dépend de la quantité  $N$  des échantillons que l'on veut générer. Les données sont calculées en un seul bloc de taille  $N$ , ce qui ne permet pas une implémentation indépendante du temps.

### 7.3.2.2 Le modèle M/G/∞

Le modèle M/G/∞ permet de générer des données dont la taille peut être choisie parmi les distributions classiques (Normale, Lognormale, Pareto, ...). Le comportement des corrélations peut être aussi très différent (SRD ou LRD) suivant la loi de service. Par exemple pour générer un trafic LRD il suffit de prendre une loi de Pareto de paramètre  $1 < \beta < 2$ .

D'un point de vue algorithmique ce modèle permet une implémentation générative. Par contre il ne peut pas générer du trafic exactement autosimilaire.

Dans le cas où le temps de service suit une loi de Pareto il peut générer un trafic asymptotiquement autosimilaire. Alors pour un trafic autosimilaire LRD avec ( $0.5 < H < 1$ ) on peut par exemple utiliser un modèle M/G/∞ avec une loi de service Pareto avec le paramètre  $\beta = 3 - 2H$ .

### 7.3.2.3 Auto-similarité de source unitaire superposée

Une source de trafic unitaire superposée peut reproduire un comportement auto-similaire. Ce comportement a été observé de façon expérimentale avec une source HTTP exécutée dans le simulateur DHS. Ce comportement est discuté par M.E Crovella et A. Bestavros[26].

## 7.4 Gestion des arrivées d'appels

Le processus de naissance et de mort modélise correctement les arrivées et les départs de sources qui correspondent à des appels d'abonnés à des services particuliers (téléphonie-vidéo, ...), arrivant à des intervalles distribués exponentiellement de taux moyen  $\lambda$ , et restant en communication pendant une durée exponentielle de taux moyen  $\mu$ , avec un taux de mort proportionnel au nombre de source actives

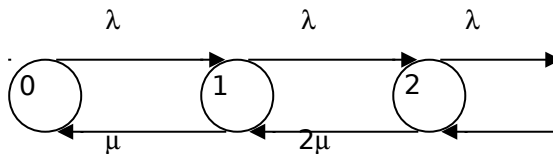


Figure 14 - Processus de naissance et de mort

Chaque état représente le nombre de sources actives simultanément. Chaque source génère un taux moyen  $r$  de paquets par secondes. Le modèle de files d'attente correspondant est M/M/∞. Si  $N(t)$  représente le nombre de sources de

trafics dans le système à l'instant  $t$ . Le nombre moyen  $\bar{N}$  de sources dans l'état stationnaire et le taux moyen  $\bar{r}$  de génération sont données par  $\bar{N} = \lambda / \mu$  et  $\bar{r} = \bar{N}r$ .

Les paramètres d'entrées  $\lambda$  et  $\mu$  d'un tel modèle peuvent provenir de statistiques faites en métrologie. En effet, les paramètres du système sont complètement déterminés en connaissant le nombre moyen de communications simultanées  $N(t)$  et la durée moyenne  $1/\mu(t)$  entre chacune d'elles.

En réalité, le nombre de sources actives simultanément ne peut pas excéder une certaine valeur  $C$ . Il faut donc pouvoir limiter le nombre d'états du processus de naissance et de mort. On utilise alors un processus d'Erlang B représentant une file d'attente de type  $M/M/C/0$ . Le nombre moyen de sources dans l'état stationnaire ainsi que le taux moyen de génération sont donnés par les relations suivantes :

$$\bar{N} = \rho(1 - \prod c) \quad \text{et} \quad \bar{r} = \bar{N}r \quad (14)$$

avec

$$\rho = \frac{\lambda}{\mu} \quad \text{et} \quad \prod_j = \frac{\rho^j / j!}{\sum_{i=0}^c \frac{\rho^i}{i!}} \quad (15)$$

## 7.5 Conclusion

Les modèles agrégés présentés dans ce chapitre sont des modèles mathématiques qu'on trouve classiquement dans la littérature.

Le modèle agrégé pour le modèle IPP (Interrupted Poisson Process) utilisé pour le trafic audio est le modèle MMPP-N+1. Nous avons vu que le modèle MMPP-N+1 peut être approché par un modèle MMPP-2 avec deux états UL (UnderLoad) et OL (OverLoad).

Le modèle  $M/G/\infty$  utilisé pour modéliser le trafic vidéo peut être agrégé avec un modèle  $M/G/\infty$  ou en processus de Poisson, avec des paramètres déduits des paramètres des modèles unitaires.

Finalement nous avons examiné la propriété d'autosimilarité qui caractérise le trafic agrégé de certains réseaux LAN de type donnée. Nous avons présenté des modèles pour générer ce type de trafic : le modèle FGN (Fractional Gaussian Noise) et le modèle  $M/G/\infty$ .

## **8 Editeur générique de source de trafic**

---

L'étude des différentes sources de trafics nous permet maintenant de déterminer un niveau suffisamment générique pour reproduire non pas une source de trafic réel mais un comportement statistique. L'éditeur doit pouvoir générer une source de trafic avec les propriétés suivantes :

- Une source unitaire :
  - o Reproduction d'une source déjà connue
  - o Création d'une nouvelle source
- Création d'une source agrégée :
  - o Reproduction d'une source déjà connue
  - o Création d'une nouvelle source
- Distribution du poids des paquets
- Distribution des inter-arrivées des paquets (sauf pour TCP)
- Source LRD, SRD
- Distribution mélangée et tronquée
- Transport des données UDP, TCP
- Exécution de flux en série (HTTP1.0) et parallèle (HTTP1.1)

L'éditeur de source est réalisé en Java ce qui permet une plus grande portabilité, les modules d'exécution sont réalisés en C++ pour des raisons de performance.

## 8.1 Niveau session

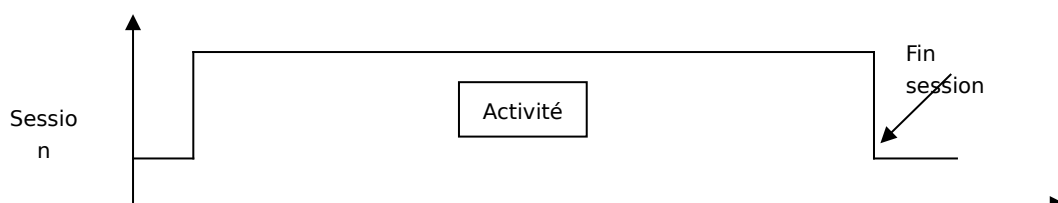


Figure 17 -Editeur niveau session

Nous observons deux cas pour la durée d'une session :

- **Session driven** : durée ou taille de la session , une fois que la durée ou le temps indiqué est atteint la session se termine, la taille ou la durée est fonction de fonction de distribution avec une moyenne et une variance.
- **Stream driven** : La session prend fin lorsque les éléments qui composent la source ont fini leur cycle, ce cycle est composé de période de temps et/ou de période de données

### Caractéristique de la source :

- le nom de la source
- le débit maximum (une source de trafic a un débit limité en fonction des interfaces physiques : carte réseau, modem, câble ...)

### Classification des Sessions :

- **Applications TCP** : pour les applications TCP, la durée moyenne ne peut pas être connue d'avance, elle dépend du l'état du réseau, des stations, des logiciels.
- **Pour les applications téléphoniques** la durée moyenne est connue, on peut donc spécifier la durée moyenne.
- **Applications UDP** : les sources UDP fixe leur débit, dans ce cas la durée moyenne peut être fixée car le temps d'envoi d'une quantité de donnée est déterministe.
- **Applications mixte TCP/UDP** : le temps de la session n'est pas connu donc seul « stream driven » est possible.

Début  
session

activité

Fin  
session

## Editeur générique de source de trafic

- **Applications en boucle fermée** : pour les protocoles particuliers dont nous n'avons pas fait d'études et qui présentent des fonctionnements en boucle fermée alors il faut utiliser « stream driven » à cause de l'élasticité possible.

Tableau 39 - Résumé des cas possibles de session

	Session driven	Stream driven
Application TCP		X
Application téléphonique (CODEC fixé)	X	
Application UDP	X	
Applications mixtes TCP/UDP		X
Application en boucle fermée		X

## 8.2 Niveau application

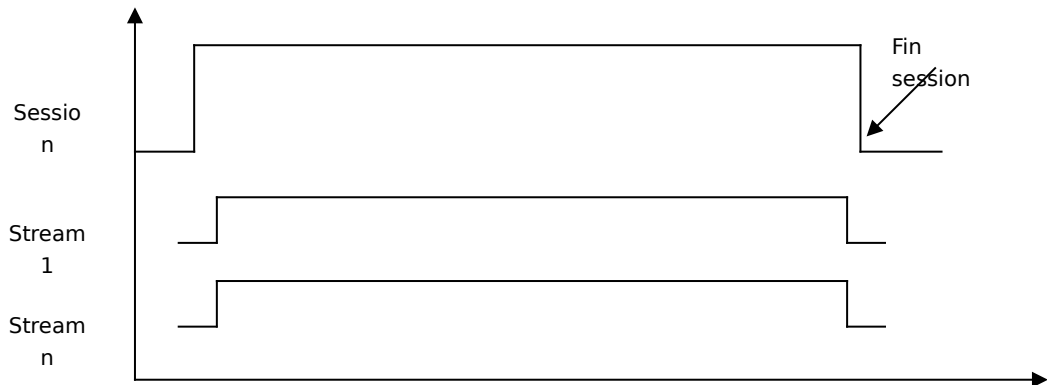


Figure 18 -Editeur niveau application

Le niveau application est défini par le comportement des flux qui composent le trafic. Nous appellerons « stream » les flux principaux qui caractérisent une source, par exemple un trafic RTP est composé d'un stream de données et d'un stream de synchronisation.

### Stream :

- Le stream est contenu dans une session,
- Flux qui composent une session,

- Un trafic peut être composé de plusieurs stream,
- Un stream peut être inclus ou exclus du volume total de données d'une source de trafic,
- Un stream peut être synchronisé par rapport à un autre stream,
- Un stream peut être forward (source -> destination) ou backward (destination -> source),
- Un stream peut être interrompu par le niveau session.

**Motif** : Nous appellerons motif un ensemble de périodes ON et OFF qui se répète dans le temps.

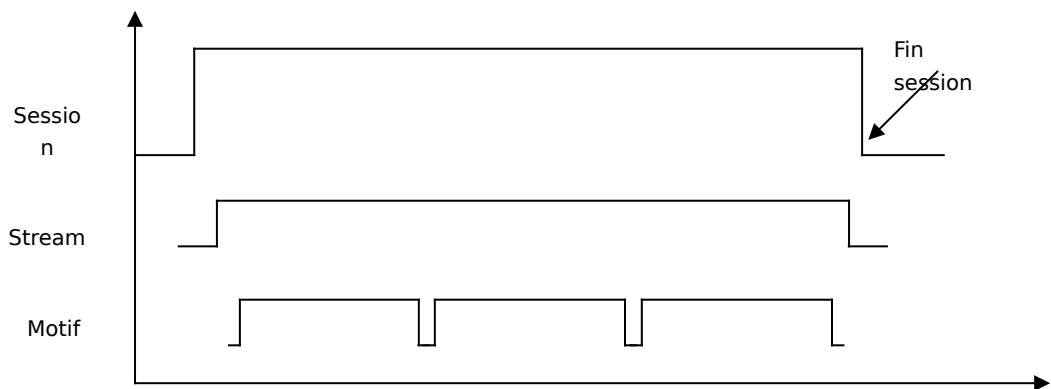


Figure 19 -Editeur niveau application

**Motifs :**

- Un motif est contenu dans un stream
- Un motif peut avoir une seule période ON avec et une période OFF égale à zéro.
- Le nombre de répétition d'un motif peut être constant ou aléatoire suivant une distribution. Dans le cas de session driven, le nombre de répétition est limité par la taille ou la durée de la session.
- Un motif peut être synchronisé

### 8.3 Niveau paquet

Le niveau paquet est le dernier niveau, il représente le moment où l'application envoie des données sur le réseau. Le niveau paquet est représenté par des « périodes » d'activité : émission de paquet et des périodes d'inactivité : période d'attente.

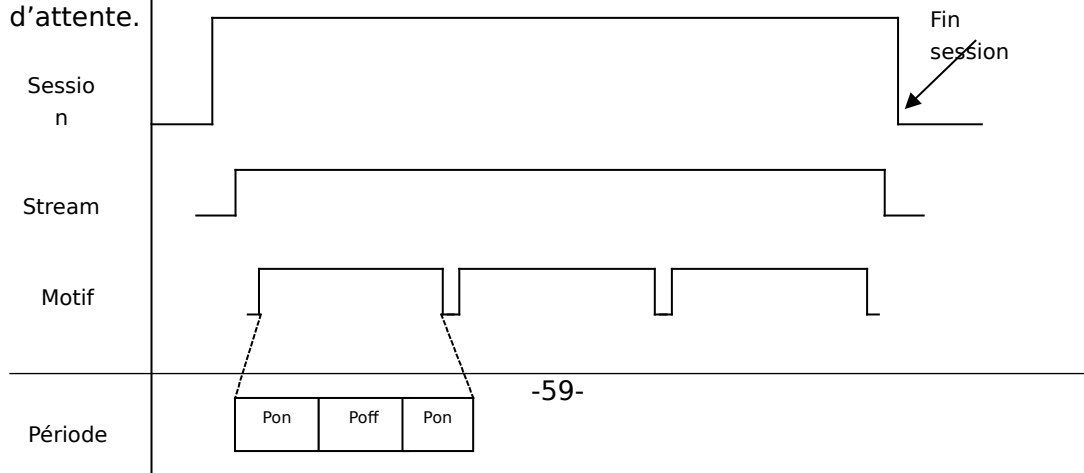




Figure 20 -Editeur niveau paquet

**Périodes :**

- Une période est contenue dans un motif
- Une période peut être définie par sa taille ou sa durée (période simple).
- Une période peut comporter des sous-Motifs (période complexe).
- Pour chaque période simple il faut définir le protocole de transport avec la taille et les inter-arrivées des paquets.
- Pour chaque période complexe il faut définir les sous-motifs
- **Période OFF** : l'application n'est pas active, cette période est définie par : distribution, moyenne et variance.
- **Période ON** : présentent un comportement plus complexe. Une période ON peut être décomposé de succession de période ON et OFF.

Une application peut pendant sa période d'activité (ON) utiliser TCP et/ou UDP. Dans le cas de TCP la période est exprimée en une taille de données à transmettre car TCP réagit en fonction du réseau (on est pas maître du débit). Pour UDP la période est exprimée en temps car le comportement est déterministe.

## **8.4 Réseau de test**

L'exécution d'une source de trafic doit se faire au travers d'un réseau dans le cas d'une source TCP car ce protocole fonctionne en boucle fermée et à besoin d'un retour sous forme d'acquittement.

Pour simuler la source un réseau de test a été crée avec DHS :

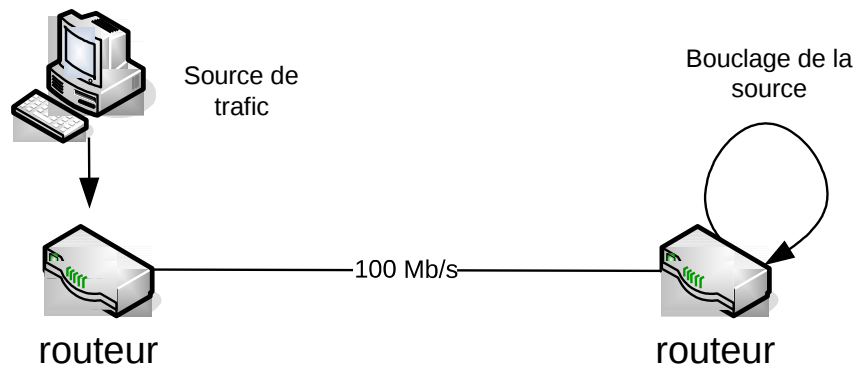


Figure 21 -Réseau de test

## 8.5 Stockage des sources

Les sources sont stockées dans une base de données MySQL, dont voici le schéma :

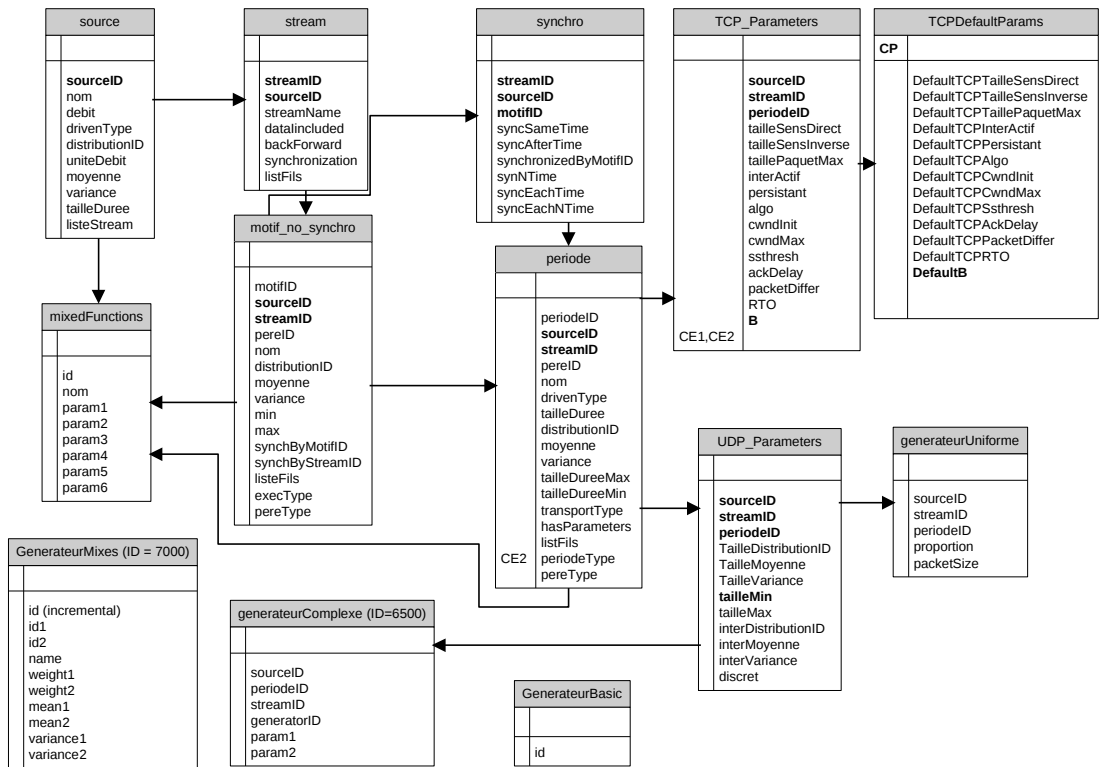


Figure 22 - Format de la base de données des sources de trafic - en version 3

L'utilisateur peut sauvegarder et réutiliser les sources de trafic.

## 8.6 Distribution statistique

### 8.6.1 Distribution mixée

Dans la littérature et dans nos expérimentations, nous observons que les sources de trafic peuvent être caractérisées par des mélanges de distributions statistiques. L'éditeur de trafic doit donc proposer à l'utilisateur la possibilité de composer des mélanges de distributions. Soit  $X$  la distribution mélange, dans ce mélange chaque distribution possède un poids  $w$  avec :

$$\text{poids mélange} = \sum_{i=1}^n w_i = 1$$

*Moyenne équivalente du mélange :*

$$E(X) = \sum_{i=1}^n w_i E(X_i)$$

*Variance équivalente du mélange :*

$$\begin{aligned} E(X^2) &= \sum_{i=1}^n E(X^2 | X = X_i) P(X = X_i) \\ &= \sum_{i=1}^n E(X_i^2) P(X = X_i) \\ \rightarrow E(X^2) &= \sum_{i=1}^n (\sigma_i^2 + m_i^2) w_i \end{aligned}$$

ou :

-  $w_i$  : poids du mélange

-  $\sigma_i$  : variance du mélange

-  $m_i$  : moyenne du mélange

$$\sigma^2 = \left( \sum_{i=1}^n X_i (\sigma_i^2 + m_i^2) \right) - m^2 \quad \mathbf{d'ou} \quad \sigma^2 = \left( \sum_{i=1}^n X_i (\sigma_i^2 + m_i^2) \right) - m^2$$

Mise en œuvre dans l'éditeur :

La composition des mélanges est fixée à deux mélanges car c'est le cas le plus utilisé. Les mélanges sont enregistrées dans une base de données et sont ensuite utilisables pour la génération de taille de paquet et des inter-arrivées.

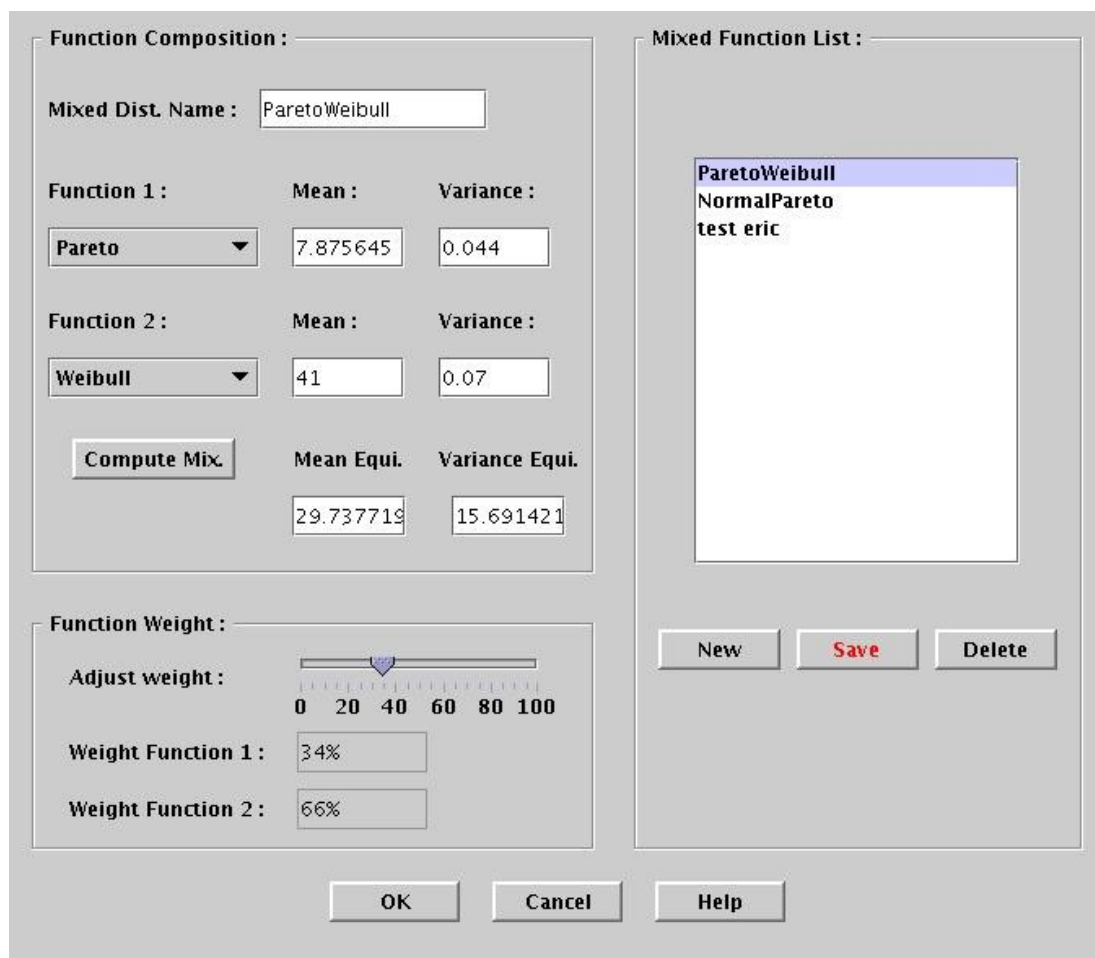


Figure 23 - Mélange de distribution dans l'éditeur

### 8.6.2 Distribution tronquée

L'utilisation d'une distribution sur tout son domaine de définition n'est pas vraiment de sens dans les cas suivants :

- génération d'une inter-arrivée négative,
- génération d'une taille de paquet qui suit une loi donnée n'a pas de sens dès que la taille des données dépasse la taille d'un paquet UDP ou TCP (même si TCP et UDP peut être fragmenté).

L'idée est donc de tronquer les distributions entre une valeur Min et Max autour de la moyenne. L'éditeur propose le choix entre une distribution :

- classique,
- tronquée,
- mélange de distribution.

## 8.7 Test et validation

### 8.7.1 Source unitaire : HTTP1.1

Le choix de http pour le test permet de valider le concept proposé, c'est le model de trafic le plus complexe que nous avons vu dans la littérature en terme de comportement. Le model se présente comme suit :

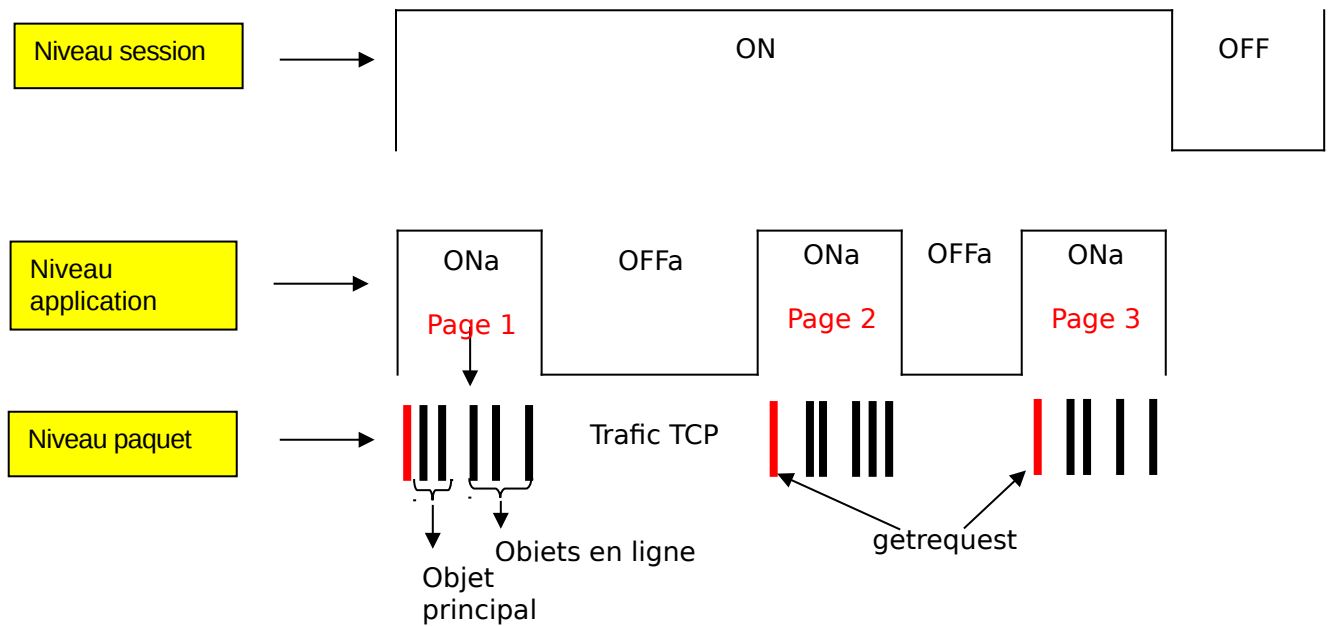


Tableau 40 - Mélange de distribution dans l'éditeur

Niveau	Distribution / Description	Type
<b>Session</b>	Choix du processus	Inter-arrivée
<b>Application</b>	<b>ON</b> : téléchargement des pages Taille getrequest en octet : lognormal (360 ; 106,5) Taille objet principal en Ko: lognormal (10 ;25) Taille objet en ligne en Ko : lognormal (7,7 ;126) Nb objet en ligne : gamma (5,5 ; 11,4)	Volume de données
	<b>OFF</b> : lecture page web En seconde: Weibull (39.5, 92.6)	Temps
<b>Transport</b>	<b>TCP/IP</b>	Emission paquet

La complexité vient de la structure des « objets en lignes » qui se répète selon la loi Gamma. Cela veut dire que la période doit être composée d'un sous-motif piloté par le motif supérieur, d'où l'intérêt de la notion de motif et sous-motif :

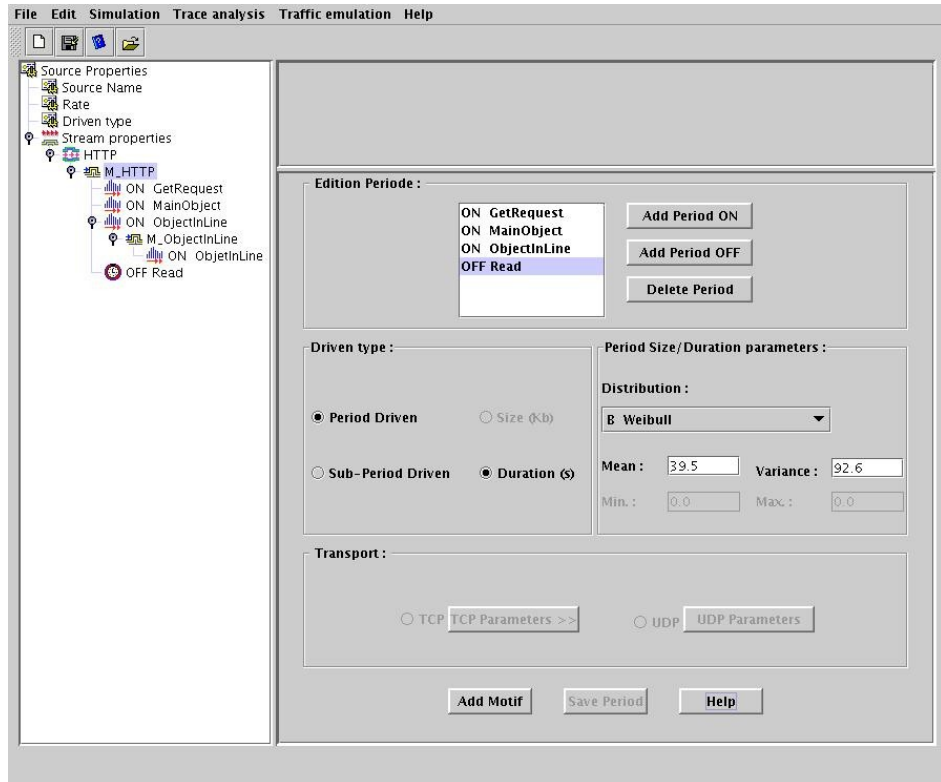


Figure 25 - Source de trafic HTTP

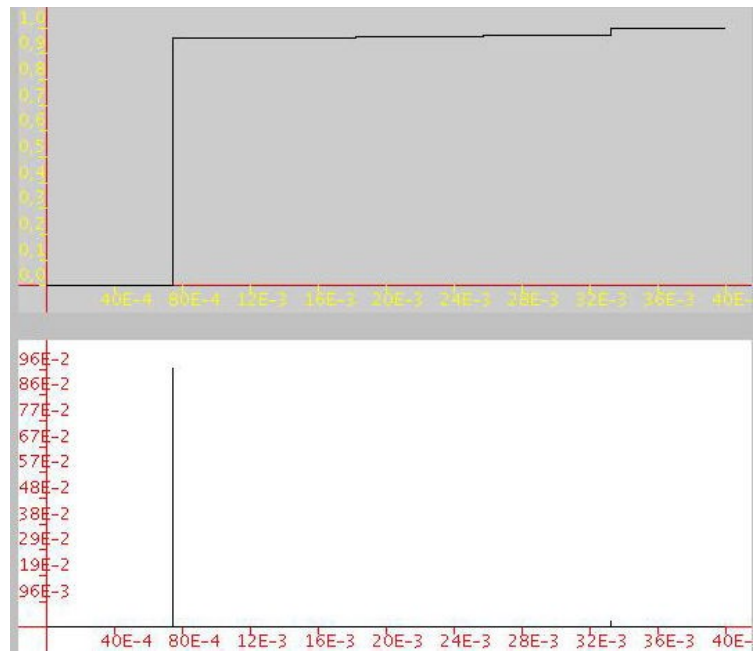


Figure 26 - CDF et PDF -Inter-arrivée des paquets HTTP

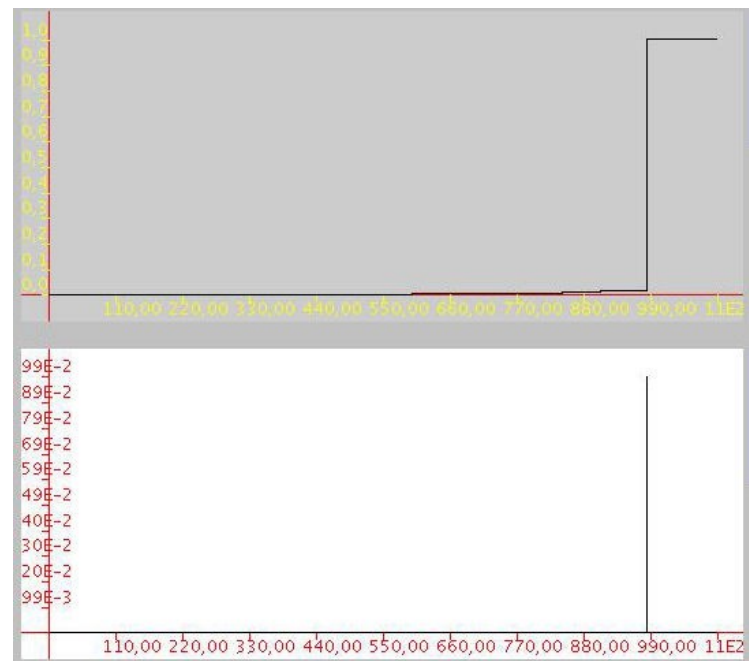


Figure 27- CDF et PDF -Taille des paquets



## 8.7.2 Source unitaire audio G711

### 8.7.2.1 Caractéristique de la source

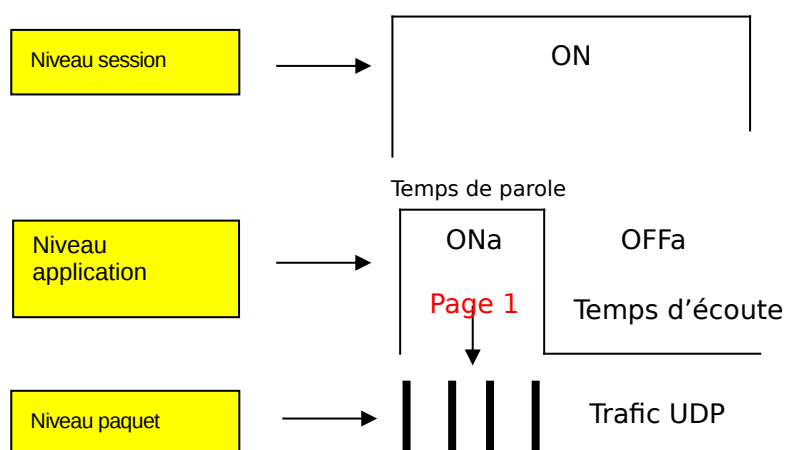


Tableau 41 - Mélange de distribution dans l'éditeur

Niveau	Distribution Description	Type
<b>Session</b>	Choix du processus	Inter-arrivée
<b>Application</b>	<b>ON</b> : temps de parole En seconde : Exponentiel (0.352)	Temps
	<b>OFF</b> : temps d'écoute En seconde: Exponentiel (0.65)	Temps
<b>Transport</b>	<b>UDP</b> : taille des paquets constant (136) IA constant(0.012)	Emission paquet

#### 8.7.2.2 PDF et CDF des inter-arrivées et taille de paquet

La PDF et la CDF des inter-arrivées montrent bien un Dirac à 0.012 seconde ce qui confirme bien les données saisies dans l'interface graphique, On en déduit le bon fonctionnement pour la génération d'une source audio simple.

De même, la taille des paquets est fixée à 136 octets, ce que nous retrouvons dans la PDF et la CDF de la taille des paquets.

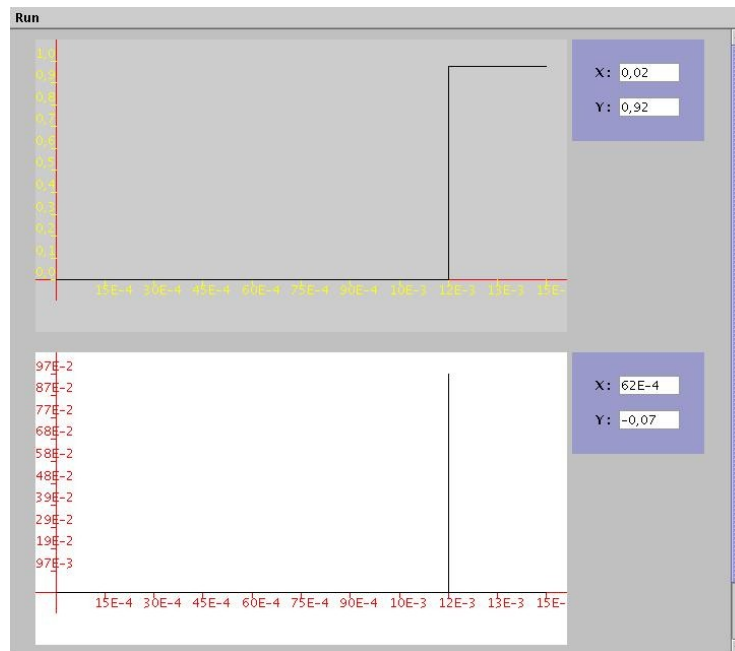


Figure 28 - CDF et PDF inter-arrivée des paquets G711

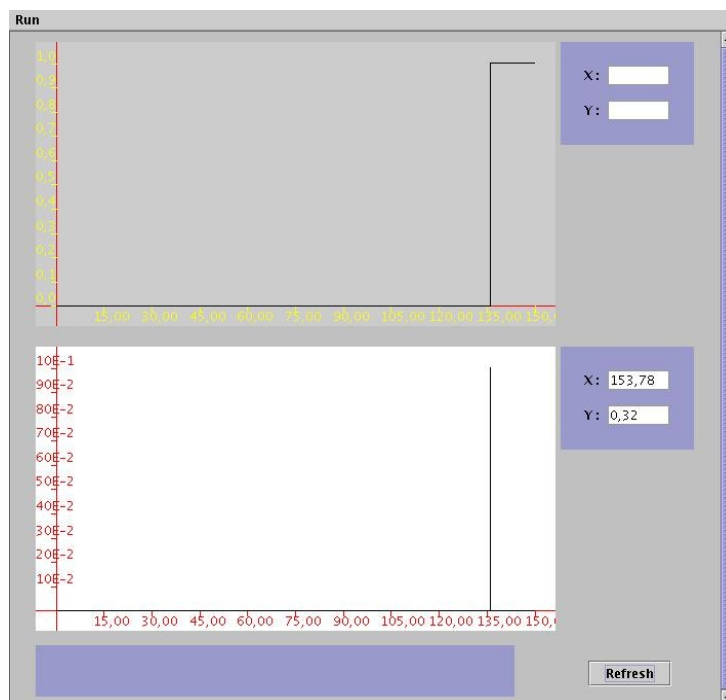


Figure 29 - CDF et PDF taille des paquets G711

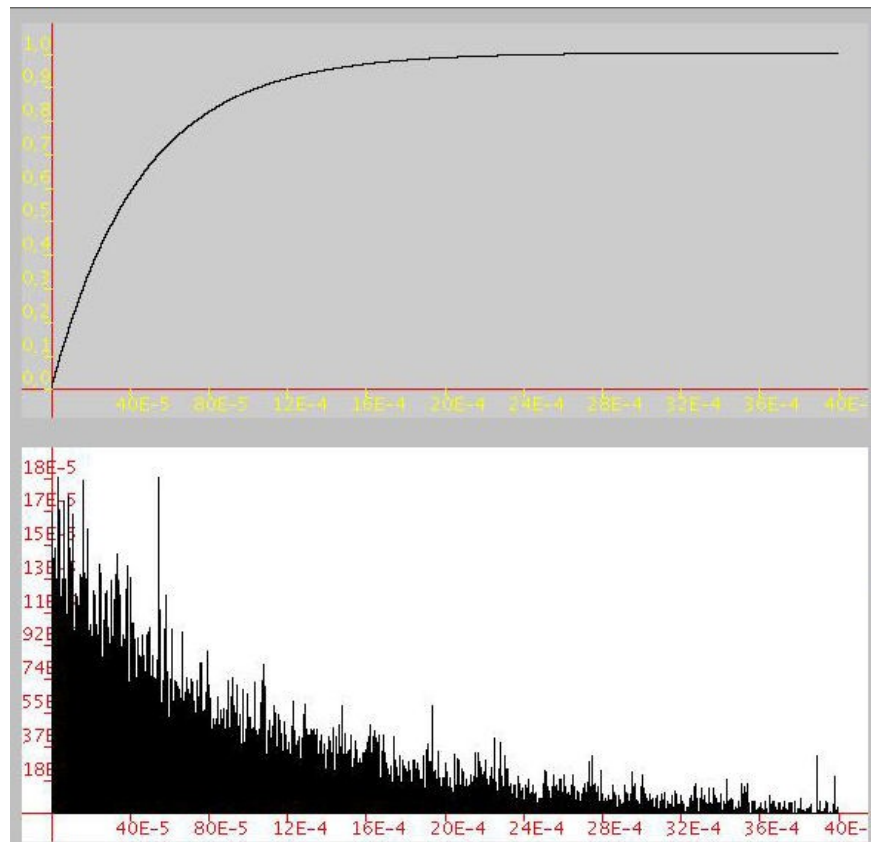
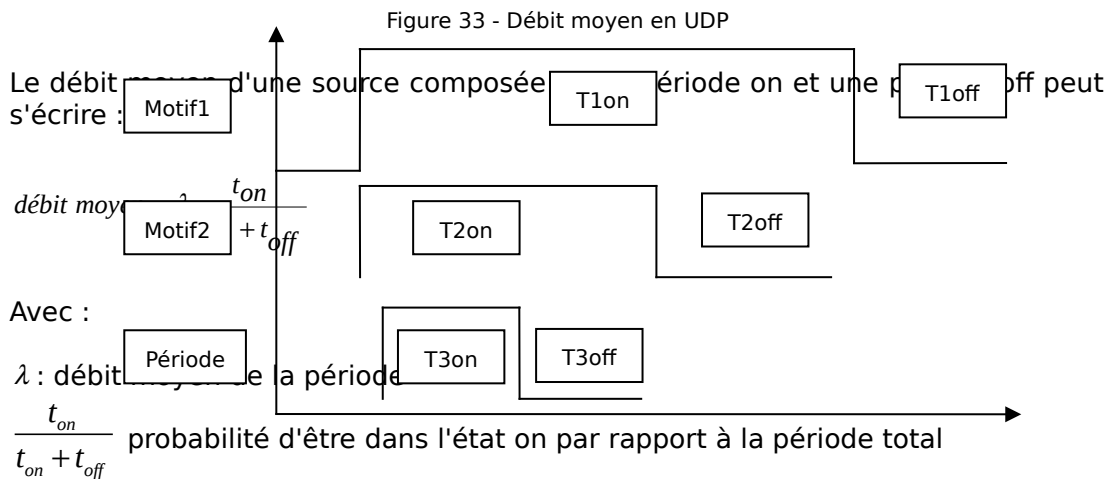


Figure 30 - CDF et PDF-Inter-arrivée pour cent sources unitaires G711

La superposition de plusieurs sources G711 montre visuellement le caractère exponentiel des inter-arrivées. Cela suggère que nous pouvons approximer la distribution des inter-arrivées par une loi exponentielle, ce comportement est bien connu et largement publié dans la littérature.

## 8.8 Débit moyen d'une source UDP

Une source de trafic UDP composée de générateur connus pour les inter-arrivées et pour la taille des paquets permet de connaître facilement la moyenne et la variance de la source.



Pour l'ensemble on peut effectuer le calcul en descendant :

$$\text{Pour le motif 1 : } d_1 = \lambda_{1on} \frac{T_{1on}}{T_{1on} + T_{1off}}$$

$$\text{Pour le motif 2 : } d_2 = \lambda_{1on} = \lambda_{2on} \frac{T_{2on}}{T_{2on} + T_{2off}}$$

$$\text{Pour la période 3 : } d_3 = \lambda_{2on} = \lambda_{3on} \frac{T_{3on}}{T_{3on} + T_{3off}}$$

$$\text{D'ou : } d_2 = \lambda_{3on} \frac{T_{3on}}{T_{3on} + T_{3off}} \frac{T_{2on}}{T_{2on} + T_{2off}}$$

$$\text{On peut maintenant calculer } d_1 : d_1 = \lambda_{3on} \frac{T_{3on}}{T_{3on} + T_{3off}} \frac{T_{2on}}{T_{2on} + T_{2off}} \frac{T_{1on}}{T_{1on} + T_{1off}}$$

$\lambda_{3on}$  est connu puisque c'est le débit moyen de la source exprimé en k/s pendant la période ON



## 9 Analyse statistique des sources de trafic

---

Une source de trafic peut être composée de plusieurs générateurs connus avec pour UDP le choix de générer des tailles de paquet et des inter-arrivées et pour TCP le choix de pouvoir générer des tailles de paquets uniquement puisque les inter-arrivées sont gérées par le protocole lui-même. Les mélanges des distributions peuvent aboutir à une source de trafic complexe dont l'analyse nécessite des outils statistiques.

### 9.1 Estimateurs statistiques

#### 9.1.1 Estimateurs des quatre premiers moments

Les quatre premiers moments donnent des indications sur le contenu d'un échantillon de trace. Ils seront utilisés pour les tests d'ajustement et pour l'affichage d'information à l'utilisateur. Voici les équations pour chaque moment (*série de Taylor*) :

*La moyenne, moment d'ordre 1* : la moyenne représente le premier moment,  $X_i$  représente un point de l'échantillon

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

*La variance, information sur la dispersion de l'échantillon* :

$$v = \sigma^2$$

*L'écart type, information sur la dispersion de l'échantillon* :

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}$$

*Moment d'ordre 3, permet de définir l'asymétrie de la distribution, proche de 0 pour une distribution symétrique* :

$$m_3 = \frac{1}{n} \sum_{i=1}^n X_i^3$$

Moment d'ordre 4 :

$$m_3 = \frac{1}{n} \sum_{i=1}^n X_i^4$$

### 9.1.2 Corrélation d'une source de trafic

La corrélation d'une source de trafic permet d'identifier des séquences répéter dans la source de trafic. Ces séquences ou motifs peuvent être répétés selon une distance plus ou moins importante. La corrélation appelée aussi auto-corrélation est exprimée de la façon suivante :

$$\rho = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y} = \frac{1/n \sum_i (x_i - \bar{y})(x_i - \bar{y})}{\sqrt{\text{var}(x)} \sqrt{\text{var}(y)}} \rightarrow \text{si on calcul la corrélation sur une série double}$$

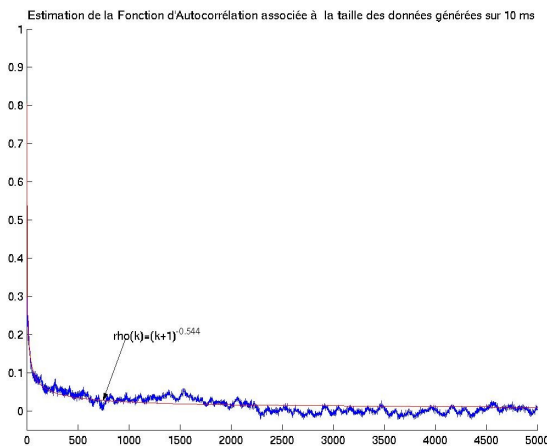
$$\rho_k = \frac{\text{cov}(x_i, x_{i+k})}{\sigma_x \sigma_x} = \frac{1/n \sum_i (x_i - \bar{x})(x_{i+k} - \bar{x})}{\text{var}(x)} \rightarrow \text{avec } \sqrt{\text{var}(x)} = \sqrt{\text{var}(y)}, \text{ sachant : écart type} = \sqrt{\text{variance}}$$

$\text{var}(x) = 1/n \sum_i (x_i - \bar{x})^2$  alors

$$\rho_k = \frac{\text{cov}(x, x_k)}{\sigma_x \sigma_x} = \frac{1/n \sum_i (x_i - \bar{x})(x_{i+k} - \bar{x})}{1/n \sum_i (x_i - \bar{x})^2}$$

d'ou  $\rho(k) = \frac{c_k}{c_0} \rightarrow c_0 = 1/n \sum_i (x_i - \bar{x})^2$  (on retrouve, la formulation de  $\frac{\text{cov}(x, x)}{\sigma_x \sigma_x}$ )

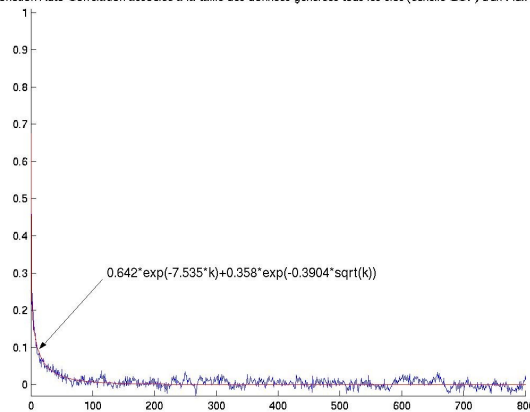
$C_k$  est un estimateur de l'auto-corrélation d'une série  $X_i$ . Il est possible alors tracer la courbe :  $\rho(k) = \frac{C_k}{c_0}$  qui donne une indication sur le caractère d'auto-corrélation.



Voici un exemple qui représente les corrélations de la taille des données générées tous les 10ms à la sortie du réseau de Jussieu, on voit une corrélation de type LRD.



on de la Fonction Auto-Correlation associée à la taille des données générées tous les 0.5s (échelle GOP) d'un Flux Cartoon H



Exemple d'un film vidéo avec une corrélation associée à la taille des GOP d'un flux vidéo H263 à débit constant de 64kbps. Corrélation de type SRD

### 9.1.3 Estimation de la PDF et CDF empirique

D'un point de vu pratique le résultat du simulateur doit être affiché pour comprendre le comportement de la source. Dans ce but un utilitaire d'affichage a été développé dans l'éditeur pour afficher la PDF et la CDF.

Dans le cas discret, la PDF est construite sur le principe suivant :

$$f(x) = \Pr[X = x]$$

Dans le cas discret, la CDF est construite sur le principe suivant :

$$F(x) = \sum_{i=1}^n f(x_i)$$

La représentation graphique est effectuée selon le principe d'un espace de calcul noté  $U$  vers un espace graphique  $U'$ . Le principe est que l'on doit maintenir les mêmes proportions.

Les équations de proportionnalité peuvent être écrite indépendamment pour  $x$  et  $y$  :

$$\frac{x - x_{\min}}{x_{\max} - x_{\min}} = \frac{u - u_{\min}}{u_{\max} - u_{\min}} \quad \text{et} \quad \frac{y - y_{\min}}{y_{\max} - y_{\min}} = \frac{v - v_{\min}}{v_{\max} - v_{\min}}$$

Pour  $u$  et  $v$  on obtient alors :

$$u = u_{\min} + \frac{u_{\max} - u_{\min}}{x_{\max} - x_{\min}}(x - x_{\min}) \quad \text{et} \quad v = v_{\min} + \frac{v_{\max} - v_{\min}}{y_{\max} - y_{\min}}(y - y_{\min})$$

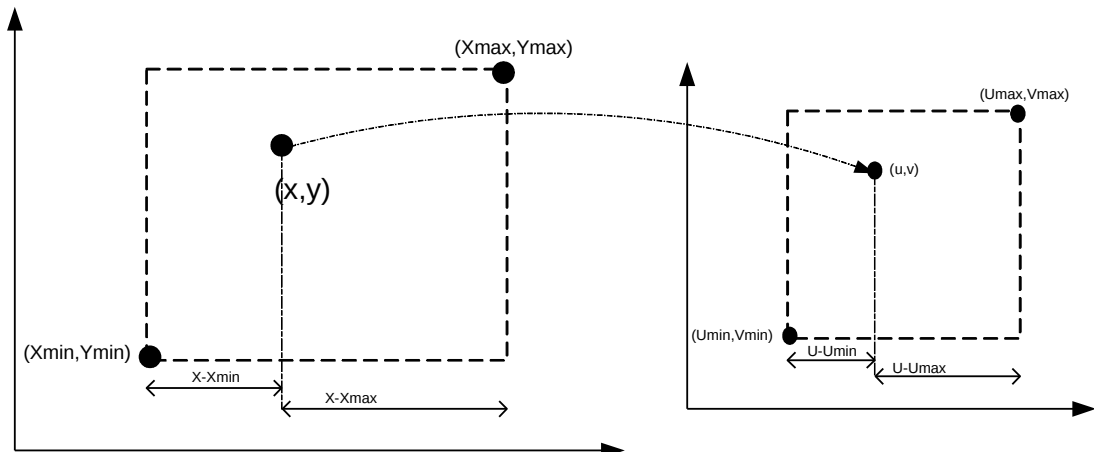


Figure 31 - Projection de  $(x,y)$  vers  $(u,v)$

### 9.1.4 Test d'ajustement

Les tests d'ajustement permettent de juger l'adéquation entre un modèle réel et un modèle théorique, les tests de comparaisons d'échantillons sont utilisés pour comparer deux ou plusieurs échantillons.

Dans le cas des sources de trafics, nous identifions la distribution associée à mesure réelle sur un réseau, ou à une source de trafic créée avec l'éditeur de trafic et exécutée dans le simulateur DHS. Plusieurs tests peuvent être utilisés pour juger de la concordance entre une distribution théorique et une distribution réelle :

- Le test le plus utilisé est celui de Pearson, plus connu sous le nom de test du chi-deux, noté :  $\chi^2$ ,
- Le test de Kolmogorov-Smirnov,
- Le test de Cramer-von Mises.

Pour mettre en œuvre un test d'ajustement, il faut :

- prélever un échantillon suffisamment important de la population étudiée ( $\geq 10000$ )

- définir une variable de décision  $D$  permet de mesurer les écarts entre la distribution théorique et la distribution empirique  $F^*$  de l'échantillon.

Pour définir la concordance des deux distributions, on doit :

Définir les hypothèses  $H_0$  et  $H_1$ ,

- $H_0$  : les observations suivent une distribution théorique spécifiée  $F = F_0$
- $H_1$  : les observations ne suivent pas une distribution théorique spécifiée  $F \neq F_0$

Enoncer une règle de décision :

- on rejette l'hypothèse  $H_0$  si la valeur calculée  $d$  est supérieure à une valeur  $d_0$  n'ayant qu'une probabilité  $\alpha$  (niveau de signification) d'être dépassée par la variable  $D$ ,
- sinon, on garde l'hypothèse  $H_0$  et on considère que la distribution théorique spécifiée peut décrire le phénomène étudié, c'est à dire  $F = F_0$ .

#### 9.1.4.1 Kolmogorov Smirnov

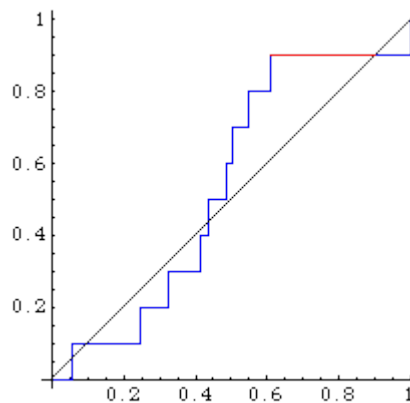


Figure 32 - Test Kolmogorov-Smirnov

#### **Principe :**

Le test de Kolmogorov-Smirnov compare la fonction de répartition uniforme avec la fonction de répartition de l'échantillon empirique. L'idée est de calculer la distance maximale (en rouge sur le graphique) entre les fonctions théoriques et empiriques. Si cette distance dépasse un certain seuil, on dira que l'hypothèse  $H_0$  est rejetée.

Le test de Kolmogorov-Smirnov est un test d'ajustement à une loi continue, qui prend en compte l'ensemble de l'échantillon. L'hypothèse nulle est :

$H_0$  : La loi a pour fonction de répartition  $F_0$

Où  $F_0$  est la fonction de répartition d'une loi continue donnée.

**L'idée est la suivante** : si l'hypothèse  $H_0$  est correcte, alors la fonction de répartition empirique  $\hat{F}$  de l'échantillon doit être proche de  $F_0$ . La fonction de répartition empirique est définie par :

$$\hat{F}(x) = \begin{cases} 0 & \text{pour } x < X_{(1)} \\ \frac{i}{n} & \text{pour } X_{(i)} \leq x \leq X_{(i+1)} \\ 1 & \text{pour } x \geq X_{(n)} \end{cases}$$

Où les  $X_{(i)}$  sont les statistiques d'ordre de l'échantillon (valeurs de l'échantillon rangées par ordre croissant). En d'autres termes,  $\hat{F}(x)$  est la proportion d'éléments de l'échantillon qui sont inférieurs ou égaux à  $x$ .

On mesure l'adéquation de la fonction de répartition empirique à la fonction  $F_0$  par la distance de Kolmogorov-Smirnov, qui est la distance de la norme uniforme entre fonctions de répartitions :

$$D_{KS} = (F_0, \hat{F}) = \max_{i=1, \dots, n} \left\{ \left| F_0(X_{(i)}) - \frac{i}{n} \right|, \left| F_0(X_{(i)}) - \frac{i-1}{n} \right| \right\}$$

Sous l'hypothèse  $H_0$ , la loi de la statistique  $D_{KS}(F_0, \hat{F})$  ne dépend pas de  $F_0$ , car les images de  $X_{(i)}$  par  $F_0$  sont des variables aléatoires de loi  $u(0,1)$ . Mais la fonction de répartition de  $D_{KS}(F_0, \hat{F})$  n'a pas d'expression explicite simple et doit



Les valeurs critiques sont largement publiées sous forme de tableau. Néanmoins des valeurs approchées sont disponibles à partir d'une taille d'échantillon  $n > 100$  :

Tableau 42 - valeurs critiques Kolmogorov-Smirnov

$$D_n = \sup |F_n^*(x) - F(x)|$$

valeur de  $d_n$  telles que  $P = P(D_n < d_n)$

---

P=0.8	P=0.9	P=0.95	P=0.98	P=0.99
$1.073 / \sqrt{n}$	$1.223 / \sqrt{n}$	$1.358 / \sqrt{n}$	$1.518 / \sqrt{n}$	$1.629 / \sqrt{n}$

---

**Limite d'utilisation :**

- Ce test s'applique seulement aux distributions continues.
- Il tend à être plus sensible au centre qu'à la queue de la distribution.
- La limitation la plus sérieuse est que la distribution doit être entièrement définie. C'est-à-dire, si l'échelle et la pente sont estimées à partir des données, la région critique de l'essai de K-S n'est plus valide. Elle doit typiquement être déterminée par la simulation. En raison des limitations ci-dessus, beaucoup d'analystes préfèrent employer le test d'ajustement Anderson-Darling. Cependant, le test d'Anderson-Darling est seulement disponible pour quelques distributions spécifiques.



NULL HYPOTHESIS H0: DISTRIBUTION FITS THE DATA  
 ALTERNATE HYPOTHESIS HA: DISTRIBUTION DOES NOT FIT THE DATA  
 DISTRIBUTION : Normale WEIGHT : 0.09419460687977466  
 DISTRIBUTION : logNormal WEIGHT : 0.9058053931202236  
 KOLMOGOROV-SMIRNOV TEST STATISTIC = 0.0017448867523415482  
 ALPHA LEVEL CUTOFF CONCLUSION  
 10% 0.012230000000000001 ACCEPT H0  
 5 % 0.013580000000000002 ACCEPT H0  
 1 % 0.01629 ACCEPT H0

CRAMER-VON MISES MISES GOODNESS-OF-FIT TEST  
 NULL HYPOTHESIS H0: DISTRIBUTION FITS THE DATA  
 ALTERNATE HYPOTHESIS HA: DISTRIBUTION DOES NOT FIT THE DATA  
 DISTRIBUTION : Normale WEIGHT : 0.09419460687977466  
 DISTRIBUTION : logNormal WEIGHT : 0.9058053931202236  
 CRAMER-VON MISES MISES TEST STATISTIC = 0.08855084946798761  
 ALPHA LEVEL CUTOFF CONCLUSION  
 10% 0.3473 ACCEPT H0  
 5 % 0.46136 ACCEPT H0  
 1 % 0.74346 ACCEPT H0

Les valeurs critiques sont publiées sous forme de tableau. Néanmoins des valeurs approchées sont disponibles à partir d'une taille d'échantillon  $n > 1000$  :

Tableau 43 - valeurs critiques Cramer-von Mises

valeur de  $z$  telle que :  $1 - \alpha = P(nw^2 < z)$

P=0.8	P=0.9	P=0.95	P=0.99
0.24124	0.34730	0.46136	0.74346

**Limite d'utilisation :**

Le test de Cramer-Von Mises a les mêmes applications que le test de Kolmogorov. La différence entre ces deux tests réside dans le fait que pour le test de Kolmogorov seul l'écart maximum entre la distribution empirique et la distribution d'ajustement entre en considération alors que la statistique de Cramer-Von Mises prend mieux en compte l'ensemble des données en calculant une distance moyenne des écarts. Le test de Kolmogorov est donc beaucoup plus sensible à l'existence de points aberrants dans un échantillon que le test de Cramer-Von Mises. On pense généralement que ce dernier test est plus puissant, mais cela n'a pas été démontré théoriquement.



### 9.1.4.3 Anderson-Darling

**Principe :**

Le test Anderson-Darling est utilisé pour tester si un échantillon de données provient bien d'une population avec une distribution spécifique. C'est une modification du test de Kolmogorov-Smirnov qui donne plus de poids à la queue de la distribution que le test de Kolmogorov-Smirnov. Le test de Anderson-Darling utilise des distributions spécifiques pour calculer des valeurs critiques. Ceci à l'avantage d'avoir un test plus précis mais aussi le désavantage de devoir calculer les valeurs critiques de chaque distribution.

Actuellement, les tables des valeurs critiques sont disponibles pour les distributions : normal, lognormal, exponentiel, Weibull (valeurs inconnues pour les mixages d'où l'impossibilité de son utilisation dans le cadre d'un mélange de distribution).

$$A^2 = -n - \frac{1}{n} \sum_{i=1}^n \{(2i-1)\ln(Z_i) + (2n+1-2i)\ln(1-Z_i)\} \text{ ou } Z_i = F(x_i)$$

Les valeurs  $Z_i$  sont supposées ordonnées dans une suite croissante

$$Z_1 < Z_2 < \dots < Z_n.$$

A noter que pour une distribution donnée, la statistique de Anderson-Darling doit être multipliée par une constante (qui dépend de la taille de l'échantillon). Ces constantes ont été publiées par Stephens Anderson.

**Test sur un échantillon de 10000 :**

1) ANDERSON-DARLING TEST STATISTIC VALUE =	5.826050
ADJUSTED TEST STATISTIC VALUE	= 5.849208
2.) CRITICAL VALUES:	
90 % POINT	= 0.6560000
95 % POINT	= 0.7870000
97.5 % POINT	= 0.9180000
99 % POINT	= 1.0920000

**Limite d'utilisation :**

Le test d'Anderson-Darling peut être utilisé pour vérifier les distributions suivantes :

- normal
- log-normal

- Weibull
- exponential
- Distribution logistique (non implémenté).

#### **9.1.4.4 Test du Khi2**

##### **Principe :**

La loi du  $\chi^2$  a été découverte par K. Pearson. On démontre que lorsque les  $n$  variables normales centrées, réduites  $X_1 \dots X_n$  sont liées par  $p$  relation, la somme de leurs carrés suit encore une loi du  $\chi^2$ , mais cette fois à  $\nu = n - p$  degrés de liberté.

Cette propriété peut être utilisée pour critiquer un ajustement comme celui proposé dans l'exemple. En effet, le problème qui se pose est de savoir si les écarts entre distribution observée et la loi théorique peuvent être jugés faibles. Par exemple, lorsque la distribution observée est celle d'un échantillon pris dans une population nombreuse, on peut se demander si les écarts entre distribution observée et loi théorique sont dus uniquement aux fluctuations d'échantillonnage. De façon plus générale, si la distribution observée concerne l'ensemble de la population, on peut se demander si elle suit ou non la loi théorique.

Le test du  $\chi^2$  s'applique à toute loi théorique, on démontre en effet que si l'on calcul pour chaque classe statistique :

$$\frac{(n_i - np_i)}{np_i} = \frac{(O - T_i)^2}{T_i}$$

Avec  $n_i = O_i$  : effectif observé,  
 $np_i = T_i$  : effectif théorique.

La somme suit une loi du  $\chi^2$ , la variable de décision est la suivante :

$$D^2 = \sum_{i=1}^n \frac{(O_i - T_i)^2}{T_i}$$

D'un point de vue pratique, ce test s'applique à des effectifs observés et non à des fréquences. Les classes doivent avoir un effectif minimum de 5 ou même 10.

La règle de décision est la suivante :

- On rejette l'hypothèse  $H_0$  si la valeur de la statistique  $D^2$  obtenue à partir de l'échantillon est supérieure à une valeur n'ayant qu'une probabilité  $\alpha$  d'être supérieure à  $\chi^2$  considérée.
- Sinon, on garde l'hypothèse  $H_0$  et on considère que la distribution spécifiée est acceptable.

Pearson a montré que la statistique  $D^2$  suit une loi à  $\nu$  degrés de liberté quelle que soit la loi considérée, quand le nombre  $n$  d'observation tend vers l'infini. Le nombre  $\nu$  de degrés de liberté est égal à :

- $(r-1)$  si la distribution théorique est entièrement déterminée, aucun paramètre n'ayant été estimé,
- $(r-1-k)$  si  $k$  paramètres ont été estimés à partir des observations, pour définir complètement la distribution.

( $r$  : Nombre d'intervalle d'égale étendue ou non.)

**Important** : le test du  $\chi^2$  s'applique à des effectifs observés et non à des fréquences. Les classes doivent avoir un effectif minimum de 5 ou même de 10. Pour les tests que nous effectuons, il est nécessaire de réunir plusieurs classes pour satisfaire à cette dernière condition. Pour chaque échantillon il est donc important de vérifier les effectifs et de prévoir une plage de degrés de liberté codée dans le programme autour du degré de liberté initialement prévu.

La distance d'un effectif est la suivante : moyenne de l'échantillon  $\pm 0,3 * \sigma$ ,  $\sigma$  représente l'écart type de l'échantillon.

**Test sur un échantillon de 10000** : test avec un échantillon Pareto Pareto et une approximation exponentielle exponentielle :

```
CHI-SQUARED GOODNESS-OF-FIT TEST
NULL HYPOTHESIS H0:      DISTRIBUTION FITS THE DATA
ALTERNATE HYPOTHESIS HA: DISTRIBUTION DOES NOT FIT THE DATA
DISTRIBUTION : Exponentielle WEIGHT : 0.36070020959100135
DISTRIBUTION : Exponentielle WEIGHT : 0.6392997904089989
CHI-SQUARED TEST STATISTIC      = 2046176.880409321
SAMPLE:
NUMBER OF OBSERVATIONS          = 10000
DEGREES OF FREEDOM              = 12
NUMBER OF PARAMETERS USED       = see E/M algo
ALPHA LEVEL      CUTOFF          CONCLUSION
10%      18.549      REJECT H0
5 %      21.026      REJECT H0
```

1 %      26.217      REJECT H0

Le test rejette l'hypothèse H0, l'échantillon ne peut pas être approximé par un mixage exponentielle exponentielle.

**Limite d'utilisation :** *Ce test s'applique à toutes les distributions connues.*

#### **9.1.4.5 Interprétation des tests**

Nous avons présenté les tests statistiques les plus utilisés. Il s'avère important de bien connaître les limites de chaque test et de ne pas se borner à interpréter juste les résultats bruts.

Il est possible que les tests ne désignent pas un même mélange de distribution comme le plus adéquate, car chaque test est sensible sur des critères différents. Il est possible de définir une métrique qui prenne en compte plusieurs tests et désigner la distribution avec la meilleure métrique comme la plus adéquate pour la distribution aléatoire cherchée [Niclausse, 1999] a défini la métrique suivante :

$$P_j = \frac{1}{n} \sum_{i=1}^n \frac{T_i^j}{\min_k (T_i^k)}$$

n : nombre de test réalisés sur chaque distribution

$T_i^j$  : valeur du i-ème test effectué sur le j-ème distribution

$\min_k (T_i^k)$  : la plus petite valeur du k-ème test effectué sur toutes les distributions

La distribution la plus adéquate sera celle qui minimisera  $P_j$ .

#### **9.1.4.6 Définition des fonctions de répartition**

##### **Loi Normal N(moyenne, variance)**

La fonction de répartition de la loi Normal en fonction d'une moyenne et une variance :

$$F(x) = \Phi\left(\frac{x-m}{\sigma}\right) \text{ avec: } \begin{cases} m = \text{moyenne} \\ \sigma = \text{écart type} \rightarrow \sigma = \sqrt{\text{var}} \end{cases}$$

sachant que:

$\Phi(x) \Rightarrow$  fonction de répartition de la loi  $N(0,1)$

**Loi Lognormal LogN(moyenne, variance)**

$$F(x) = \Phi\left(\frac{\ln x - m}{\sigma}\right) \text{ avec: } \begin{cases} m = \text{moyenne} \\ \sigma = \text{écart type} \rightarrow \sigma = \sqrt{\text{var}} \end{cases}$$

sachant que:

$\Phi(x) \Rightarrow$  fonction de répartition de la loi  $N(0,1)$

**Loi Pareto Pareto(moyenne, variance)**

Si  $x \geq \lambda$

$$F(x) = 1 - \exp\left(-\beta \log\left(\frac{x}{\lambda}\right)\right) \text{ avec: } \begin{cases} \beta = 1 + \sqrt{1 + \frac{\text{mean}^2}{\text{var}}} \\ \lambda = \text{mean} \frac{(\beta - 1)}{\beta} \end{cases}$$

Si non  $F(x) = 0$

**Loi Inverse Gaussienne IGaussienne(moyenne, variance)**

Si  $x \geq 0$

$$F(x) = \Phi\left(\frac{x}{\mu - 1} \sqrt{\frac{\lambda}{x}}\right) + e^{\frac{2\lambda}{\mu}} \Phi\left(-\left(1 + \frac{x}{\eta}\right) \sqrt{\frac{\lambda}{x}}\right) \text{ avec: } \begin{cases} \mu = \text{mean} \\ \lambda = \frac{\text{mean}^3}{\text{var}} \end{cases}$$

Si non  $F(x) = 0$

**Loi de Weibull Weibull(moyenne, variance)**

Pour  $x \geq 0$

$$F(x) = 1 - e^{-\left(\frac{x}{a_1}\right)^{b_1}} \text{ sachant: } \begin{cases} a_1 = \lambda^{\frac{1}{a}} \\ b_1 = a \end{cases}$$

## 10 Evaluation d'une source de trafic

---

Les applications informatiques génèrent des flux d'information dont il est parfois difficile d'évaluer le comportement. Le comportement est d'autant plus difficile à comprendre lorsqu'une même application génère des flux multimédia (http, RTP, RTSP, SIP ...). Les applications ont un impact important sur le comportement dynamique d'un réseau, la connaissance du comportement des flux doit permettre de configurer le réseau en conséquence (gestion des buffers, politique de priorité des files d'attente, débit max des interfaces, explication des congestions de réseau ...).

Dans ce chapitre nous allons présenter des méthodes d'analyse de flux de réseaux et les outils développés, le but principal est d'aider à déterminer les lois statistiques des variables aléatoires d'une observation sur le réseau: les inter-arrivées, la taille des paquets. L'observation peut être faite à partir d'un flux agrégé de flux inconnus ou alors d'un flux en particulier. Le format de trace retenu est celui de NS afin de garder une ouverture de l'outil sur une application connue.

Dans la bibliographie, les méthodes E/M (Expectation/Maximisation) et Levenberg-Marquardt (les moindres carrés), semblent être les plus efficaces pour l'évaluation d'un échantillon et l'approximation d'un modèle analytique que l'on peut utiliser pour rejouer un trafic observé. Nous allons dans ce chapitre présenté la méthode E/M utilisé pour caractériser un échantillon inconnu, ainsi que le principe de l'algorithme de Levenberg-Marquardt.

### 10.1 Méthode des moindres carrés

La méthode de Levenberg-Marquardt est un algorithme d'optimisation non linéaire qui permet d'ajuster la distance entre des données calculées à partir d'une fonction continue et des données observées issue d'un échantillon, en minimisant la distance des moindres carrés :

$$\min \frac{1}{2} \|F(x, xdata) - ydata\|_2^2 = \min \left( \frac{1}{2} \sum_{i=1}^m (F(x, xdata_i) - ydata_i)^2 \right)$$

avec :

*xdata* : vecteur de données en entrée de taille  $m$ ,  
*ydata* : vecteur des données observées de taille  $m$ ,  
 $x$  : coefficients qui ajustent au mieux l'équation

## 10.2 L'algorithme E/M

L'algorithme E/M (Expectation/Maximization [25]) se base sur le principe que l'observation d'un échantillon (des milliers de  $X_i$ ) ne suit pas parfaitement une distribution donnée (normal, Weibull, ...). Les tests statistiques, qui sont d'autant plus précis que l'échantillon est grand, finissent par mesurer cette différence même si la comparaison visuelle donne parfois l'impression d'un ajustement correcte entre le modèle et les données. Il semble alors qu'une seule distribution ne soit pas assez riche pour s'ajuster à l'échantillon. Pour palier à ce problème l'algorithme E/M propose d'utiliser un mélange de densité de probabilité.

Soient :  $f_1(x), \dots, f_k(x)$ ,  $k$  fonction paramétrique de densités de probabilité avec paramètres  $\theta_1, \dots, \theta_k$ .

Nous considérons le mixage :  $p(x) = \pi_1 f_1(x) + \dots + \pi_k f_k(x)$  ou les coefficients (ou poids de mixage)  $\pi_i$  sont positifs et  $\pi_1 + \dots + \pi_k = 1$ .

Soit  $x = (x_1, \dots, x_n)$  un échantillon de  $p(y)$  et soit  $\Psi = (\pi, \theta)$  le vecteur de paramètre à estimer ou  $\pi = (\pi_1, \dots, \pi_k)$  et  $\theta = (\theta_1, \dots, \theta_k)$ . L'algorithme EM génère, à partir d'une approximation initiale  $\Psi^{(0)}$ , une séquence d'estimation  $\{\Psi^{(m)}\}_m$ . Dans chaque itération il y a deux pas,

Pas E : calculer  $E \left[ \log p(y | \Psi) | x, \Psi^{(m)} \right] = Q(\Psi, \Psi^{(m)})$

Pas M : trouver  $\Psi = \Psi^{m+1}$  pour maximiser  $Q(\Psi, \Psi^{(m)})$

Soit :

$$\omega_{ij}^{(m)} = \frac{\pi_j^{(m)} f_j(x_i | \theta_j^{(m)})}{p(x_i | \Psi^{(m)})}$$

$$\omega_i^{(m)} = (\omega_{i1}^{(m)}, \dots, \omega_{ik}^{(m)})$$

$$V(\pi) = (\log \pi_1, \dots, \log \pi_k)$$

$$U_i(\theta) = (\log f_1(x_i | \theta_1), \dots, \log f_k(x_i | \theta_k))$$

Alors :  $Q(\Psi, \Psi^{(m)}) = \sum_{i=1}^n (\omega_i^{(m)}) \mathcal{V}^T(\pi) + \sum_{i=1}^n (\omega_i^{(m)}) U_i^T(\theta)$

En dérivant  $\pi_j^{(m+1)} = \sum_{i=1}^n \omega_{ij}^{(m)} = \frac{1}{n} \sum_{i=1}^n \pi_j^{(m)} \frac{f_j(x_i | \theta_j^{(m)})}{p(x_i | \Psi^{(m)})}$   $Q(\Psi, \Psi^{(m)})$  par rapport à  $\pi_j$

on obtient les poids du pas M :

$$\pi_j^{(m+1)} = \sum_{i=1}^n \omega_{ij}^{(m)} = \frac{1}{n} \sum_{i=1}^n \pi_j^{(m)} \frac{f_j(x_i | \theta_j^{(m)})}{p(x_i | \Psi^{(m)})}$$

$$\frac{dQ(\Psi, \Psi^{(m)})}{d(m+1)} = 0 \Leftrightarrow \frac{d\left(\sum_{i=1}^n w_i^{(m)} V_i(\theta)\right)}{d(m+1)} \Big| \frac{d\left(\sum_{i=1}^n w_i^{(m)} U_i(\theta)\right)}{d(m+1)}$$

le terme  $\frac{d\left(\sum_{i=1}^n w_i^{(m)} V_i(\theta)\right)}{d(m+1)}$  s'annule car le vecteur  $V_i(\theta)$  est cst.

reste le second terme qu'il convient de dérivée selon :  $u'v + uv'$

d'ou:  $\frac{dQ(\Psi, \Psi^{(m)})}{d(m+1)} = 0 \Leftrightarrow \sum_{i=1}^n w_i^{(m)} \log f_i(x)$  cette formulation est ensuite

utilisée pour définir l'étape(m+1) des distributions choisies dans le cadre de cette étude.

Pour l'utilisateur il est plus simple de d'utiliser la moyenne et la variance plutôt que des paramètres de la fonction elle-même. Cela à pour inconvénient de recalculer dans certains cas les paramètres en fonction de la moyenne et la variance et ensuite de faire l'opération inverse lorsque l'algorithme à atteint le point de convergence désirée.

Vu la complexité de l'algorithme, il est préférable de définir un développement générique qui permet de ne pas prendre en compte les spécificités de chaque distribution. D'un point de vu développement la généralité est plus complexe à mettre en œuvre et peut entraîner des pertes de performance à cause des efforts de casting entre les classes. Pour cette raison le développement est générique sauf au moment du calcul ou l'on retrouve les classes avec l'utilisation d'accesseur.

Au moment des calculs les itérations sur l'échantillon ont été factorisé partout ou il apparaît une somme du type :  $\sum_{i=1}^n X_i$ , puis les divisions ont été reportée à la fin des calcul afin de ne pas avoir une perte de précision.



Dans les paragraphes suivants nous développons les distributions utilisées dans le cadre de l'étude des sources de trafics.

### **10.2.1 Distribution Exponentielle**

Fonction de densité

$$f_j(x) = \lambda_j \exp(-\lambda_j x) \quad \text{pour tout } x \geq 0$$

Paramètres de la loi en fonction de la moyenne et la variance

$$mean = \frac{1}{\lambda} \quad var = mean^2$$

Les paramètres de l'étape (m+1)

$$\lambda_j^{(m+1)} = \frac{\sum_{i=1}^n W_{ij}^{(m)}}{\sum_{i=1}^n W_{ij}^{(m)} X_i}$$

Moyenne et variance pour l'étape (m+1)

*Les paramètres mean, var se déduisent facilement :*

$$mean = \frac{1}{\lambda} \quad var = mean^2$$

### **10.2.2 Distribution normale**

Fonction de densité

$$f_j(x) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left(-\frac{(x - m_j)^2}{2\sigma_j^2}\right) \quad \text{pour tout } x$$

Paramètres de la loi en fonction de la moyenne et la variance

$$var = \sigma^2 \quad mean = m_j, \quad var = \sigma^2$$

Les paramètres de l'étape (m+1)

$$\mu_j^{(m+1)} = \frac{\sum_{i=1}^n w_{ij}^{(m)} x_i}{\sum_{i=1}^n w_{ij}^{(m)}} \quad \text{et} \quad \sigma_j^{2(m+1)} = \frac{\sum_{i=1}^n w_{ij}^{(m)} (x_i - \mu_j^{(m+1)})^2}{\sum_{i=1}^n w_{ij}^{(m)}}$$

### 10.2.3 Distribution lognormale

Fonction de densité

$$f(x) = \frac{1}{x\delta_j\sqrt{2\pi}} \exp\left(-\frac{(\log(x) - \mu_j)^2}{2\delta_j^2}\right) \quad \text{pour tout } x > 0, \delta > 0$$

$$\text{mean} = \exp\left(\mu_j + \frac{\delta_j^2}{2}\right); \quad \text{var}^{(m+1)} = \exp(2\mu_j^{(m+1)} + \delta_j^{2(m+1)}) \exp(\delta_j^2 - 1)$$

$$\frac{\text{var}}{\text{mean}^2} = \exp(\delta^2 - 1) \rightarrow \delta^2 = \log\left(1 + \frac{\text{var}}{\text{mean}^2}\right)$$

Paramètres de la loi en fonction de la moyenne et la variance :

$$\mu = \log(\text{mean}) - \frac{1}{2} \log\left(1 + \frac{\text{var}}{\text{mean}^2}\right) \quad \delta^2 = \log\left(1 + \frac{\text{var}}{\text{mean}^2}\right)$$

Les paramètres de l'étape (m+1) :

$$\mu_j^{(m+1)} = \frac{\sum_{i=1}^n \omega_{ij}^{(m)} \log x_i}{\sum_{i=1}^n \omega_{ij}^{(m)}} \quad \delta_j^{2(m+1)} = \frac{\sum_{i=1}^n \omega_{ij}^{(m)} (\log x_i - \mu_j^{(m+1)})^2}{\sum_{i=1}^n \omega_{ij}^{(m)}}$$

Moyenne et variance pour l'étape (m+1) :

$$\text{mean}^{(m+1)} = \exp\left(\mu_j^{(m+1)} + \frac{\delta_j^{2(m+1)}}{2}\right)$$

$$\text{var}^{(m+1)} = \exp(2\mu_j^{(m+1)} + \delta_j^{2(m+1)}) (\exp(\delta_j^2) - 1)$$

### 10.2.4 Distribution inverse-gaussienne

Fonction de densité

$$f(x) = \sqrt{\frac{\gamma_j}{2\pi x^3}} \exp\left(-\frac{\gamma(x - \mu_j)^2}{2\mu_j^2 x}\right) \text{ pour tout } x > 0$$

Paramètres de la loi en fonction de la moyenne et la variance :

$$mean = \mu \rightarrow \mu_j = mean$$

$$var = \frac{\mu_j^3}{\gamma_j} \rightarrow \gamma_j = \frac{mean^3}{var}$$

Les paramètres de l'étape (m+1) :

$$\mu_j^{(m+1)} = \frac{\sum_{i=1}^n \omega_{ij}^{(m)} X_i}{\sum_{i=1}^n \omega_{ij}^{(m)}} \quad \gamma_j^{(m+1)} = \frac{\sum_{i=1}^n \omega_{ij}^{(m)}}{\sum_{i=1}^n \omega_{ij}^{(m)} \frac{(X_i - \mu_j^{(m+1)})^2}{\mu_j^{2(m+1)} X_i}}$$

Moyenne et variance pour l'étape (m+1) :

$$mean = \mu \quad var = \frac{\mu_j^3}{\gamma_j}$$

### 10.2.5 Distribution gamma

Fonction de densité :

$$f(x) = \frac{b^{-a}}{\Gamma(a)} x^{a-1} \exp\left(-\frac{x}{b}\right) \text{ pour tout } x \geq 0, b > 0$$

Paramètres de la loi en fonction de la moyenne et la variance :

$$\text{mean} = ab \rightarrow a = \frac{\text{mean}^2}{\text{var}}$$

$$\text{var} = ab^2 \rightarrow b = \frac{\text{var}}{\text{mean}}$$

$$\text{On a : } \begin{cases} \ln b + \frac{\Gamma'(a)}{\Gamma(a)} = \frac{\sum_{i=1}^n \omega_{ij}^m \ln x_i}{\sum_{i=1}^n \omega_{ij}^m} \\ ab = \frac{\sum_{i=1}^n \omega_{ij}^m x_i}{\sum_{i=1}^n \omega_{ij}^m} \end{cases}$$

$$\text{D'ou : } \begin{cases} b = \frac{1}{a} \frac{\sum_{i=1}^n \omega_{ij}^m x_i}{\sum_{i=1}^n \omega_{ij}^m} \end{cases}$$

Résoudre par dichotomie le paramètre « a »:

$$\text{D'ou : } \begin{cases} \frac{\Gamma'(a)}{\Gamma(a)} - \ln a = \frac{\sum_{i=1}^n \omega_{ij}^m \ln x_i}{\sum_{i=1}^n \omega_{ij}^m} - \ln \left( \frac{\sum_{i=1}^n \omega_{ij}^m x_i}{\sum_{i=1}^n \omega_{ij}^m} \right) \end{cases}$$

(par itération on connaît la partie droite de l'équation, par contre « a » n'est pas connu, il faut alors résoudre f(a)=constante)

(il est nécessaire de multiplier chaque terme par -1 pour obtenir une fonction gamma décroissante pour la résolution par dichotomie)

La fonction décroissante s'écrit :  $\left\{ \ln a - \frac{\Gamma(a)'}{\Gamma(a)} = \ln \left( \frac{\sum_{i=1}^n \omega_{ij}^m x_i}{\sum_{i=1}^n \omega_{ij}^m} \right) - \frac{\sum_{i=1}^n \omega_{ij}^m \ln x_i}{\sum_{i=1}^n \omega_{ij}^m} \right.$

Avec :  $\frac{\Gamma(x)'}{\Gamma(x)} = (\ln \Gamma(x))' = \frac{\Gamma(x+h) - \Gamma(x)}{h}$  puis calculer le paramètre « b » :

$$b = \frac{1}{a} \frac{\sum_{i=1}^n \omega_{ij}^m x_i}{\sum_{i=1}^n \omega_{ij}^m}$$

### 10.2.6 Distribution de Pareto

Fonction de densité :

$$f_j(x) = \beta_j \lambda_j^{\beta_j} x^{-(\beta_j+1)} \text{ pour } x < \lambda \text{ alors } f(x) = 0$$

Paramètres de la loi en fonction de la moyenne et la variance :

$$\beta_j = 1 + \sqrt{1 + \frac{\text{mean}^2}{\text{var}}} \text{ et } \lambda_j = \frac{\text{mean}(\beta_j - 1)}{\beta_j}$$

Les paramètres de l'étape (m+1) :

$$\lambda_j^{(m+1)} = \min(X_i) \text{ et } \beta_j^{(m+1)} = \frac{\sum_{i=1}^n \omega_{ij}^{(m)}}{\sum_{i=1}^n \omega_{ij}^{(m)} (\log x_i - \log \lambda_j)}$$

Moyenne et variance pour l'étape (m+1) :

$$\text{Si } \beta > 2 \rightarrow \text{var} = \frac{\lambda_j^2 \beta_j}{(\beta_j - 1)^2 (\beta_j - 2)} \quad \beta > 1 \rightarrow \text{mean} = \frac{\lambda_j \beta_j}{\beta_j - 1}$$

$$\text{Si } \beta > 2 \rightarrow \text{var} = \frac{\lambda_j^2 \beta_j}{(\beta_j - 1)^2 (\beta_j - 2)}$$

La fonction de Pareto donne dans certains cas une moyenne et variance « négative », cela s'explique car l'expression qui donne la moyenne et la variance est acceptable dans certaines conditions. Lorsque la moyenne et la variance sont négatifs seuls les paramètres de la fonction donnent une indication correcte.

### 10.2.7 Distribution de Weibull

Fonction de densité :

$$f_j(x) = a \lambda x^{a-1} e^{(-\lambda x^a)}$$

Paramètres de la loi en fonction de la moyenne et la variance :

Le paramètre « a » initial est calculer par dichotomie :

a=dichotomie(avg, var);

On en déduit :

lambda = pow((nr\_gamma(1+1/a))/avg, a);

avec :

```
real gen_weibull_c::dichotomie(real avg, real var){
  real k=var/pow(avg,2);
  real leftx, rightx, meanx;

  leftx=0.01;
  rightx=1.0;
  #define INC 100
  #define F(x) ( (nr_gamma(1+2/x))/pow(nr_gamma(1+1/x),2)-1) - k )
  while (F(rightx)> 0) {
    rightx+=INC; }
  while (rightx-leftx >= ZERO) {
    meanx=(leftx+rightx)/2;
    if (F(meanx)> 0)
      leftx=meanx;
    else
      rightx=meanx;
  }
}
```

nr\_gamma : fonction de densité Gamma(x)

Les paramètres de l'étape ( m+1) :

Pour réduire le produit en somme:

$$\log f(x) = \log a + \log \lambda + (a - 1) \log x - \lambda x^a$$

Cette fonction possède deux estimateurs qu'il s'agit de calculer pour l'étape (m + 1):

$$\frac{d \log f(x)}{da} = \frac{1}{a} + \log x - \lambda \log(x) x^a$$

$$\frac{d \log f(x)}{d\lambda} = \frac{1}{\lambda} - x^a$$

Dans l'équation  $Q(\Psi, \Psi^{(m)})$  on peut déduire  $\lambda$  en fonction de  $a$  :

$$\sum_{i=1}^n w_{ij}^{(m)} \left( \frac{1}{\lambda} - x_i^a \right) = 0 \rightarrow \lambda \sum_{i=1}^n x_i^a w_{ij}^{(m)} = \sum_{i=1}^n w_{ij}^{(m)}$$

$$d'ou \lambda^{(m+1)} = \frac{\sum_{i=1}^n w_{ij}^{(m)}}{\sum_{i=1}^n x_i^a w_{ij}^{(m)}} \quad (1)$$

La fonction de coût pour (a):

$$\sum_{i=1}^n w_{ij}^{(m)} \left( \frac{1}{a} + \log x_i - \lambda \log(x_i) x_i^a \right) = 0$$

$$\Leftrightarrow \sum_{i=1}^n w_{ij}^{(m)} + a \sum_{i=1}^n w_{ij}^{(m)} \log x_i - \lambda a \sum_{i=1}^n w_{ij}^{(m)} \log(x_i) x_i^a = 0$$

$$\Leftrightarrow \sum_{i=1}^n w_{ij}^{(m)} + a \sum_{i=1}^n w_{ij}^{(m)} \log x_i - \left( \frac{\sum_{i=1}^n w_{ij}^{(m)}}{\sum_{i=1}^n x_i^a w_{ij}^{(m)}} a \sum_{i=1}^n w_{ij}^{(m)} x_i^a \log x_i \right) = 0$$

Pour chaque étape résoudre cette équation par dichotomie, puis calculer  $\lambda$  avec (1).

Moyenne et variance pour l'étape (m+1) :

$$\begin{aligned} \text{moyenne} &= \lambda^{-\frac{1}{a}} \Gamma\left(1 + \frac{1}{a}\right) \\ \text{var} &= \lambda^{-\frac{2}{a}} \left( \Gamma\left(1 + \frac{2}{a}\right) - \left( \Gamma\left(1 + \frac{1}{a}\right) \right)^2 \right) \end{aligned}$$

### 10.2.8 Condition initiale

Le démarrage de l'algorithme doit suivre quelques règles. :

- L'observation de l'échantillon peut apporter des critères de démarrage. La moyenne et la variance sont deux estimateurs qui permettent une première itération proche de l'échantillon.
- Dans le cas d'un mélange de deux distributions identiques, les conditions initiales doivent être les suivantes :

Soit  $f_1(x)$  et  $f_2(x)$  si  $f_1(x) = f_2(x)$  alors  $\pi_1 \neq \pi_2$ .

- L'échantillon doit respecter le domaine de validité de la fonction de distribution, dans notre cas les valeurs discrètes sont strictement positive (taille de donnée, inter-arrivée de paquet). Avant de lancer un test sur un mélange, il est nécessaire de vérifier le contenu de l'échantillon, cette opération est réalisée à l'échantillonnage avec les critères suivants :

Pour chaque  $X_i$  vérifier si :  $\begin{matrix} \exists X_i < 0 \\ \exists X_i = 0 \end{matrix}$

### 10.2.9 Condition d'arrêt

Le test d'arrêt consiste à calculer la distance entre  $\Psi^{(m)}$  et  $\Psi^{(m+1)}$  :

$$\text{Etape (m)} : \Psi^{(m)} = (\pi_1^{(m)}, \pi_2^{(m)}, m_1^{(m)}, \sigma_1^{(m)}, \delta_2^{(m)})$$

$$\text{Etape (m+1)} : \Psi^{(m+1)} = (\pi_1^{(m+1)}, \pi_2^{(m+1)}, m_1^{(m+1)}, \sigma_1^{(m+1)}, \delta_2^{(m+1)})$$



La distance est représentée par  $dist(\Psi^{(m+1)}, \Psi^{(m)})$  :

$$dist = \sum_{i=1}^n (\Psi_i^{(m+1)} - \Psi_i^{(m)})^2 =$$
$$(\pi_1^{(m+1)} - \pi_1^{(m)})^2 + (\pi_2^{(m+1)} - \pi_2^{(m)})^2 + (m_1^{(m+1)} - m_1^{(m)}) + \dots$$

si la distance est supérieure à epsilon (fixé au départ) alors bouclage de l'algorithme avec :

Paramètre étape E = paramètre étape M puis paramètre étape M = 0

Les tests effectués montrent une convergence rapide même dans le cas d'un choix de plusieurs distributions (n>4).

### 10.2.10 Interface graphique

L'interface graphique permet de choisir N distributions à mélanger parmi les distributions disponibles, pour chaque distribution l'utilisateur peut définir le poids, la moyenne et la variance. Le poids peut être équitablement réparti en utilisant la touche « Dispatch weight ». Enfin le champ « accuracy » permet de donner la précision souhaitée.

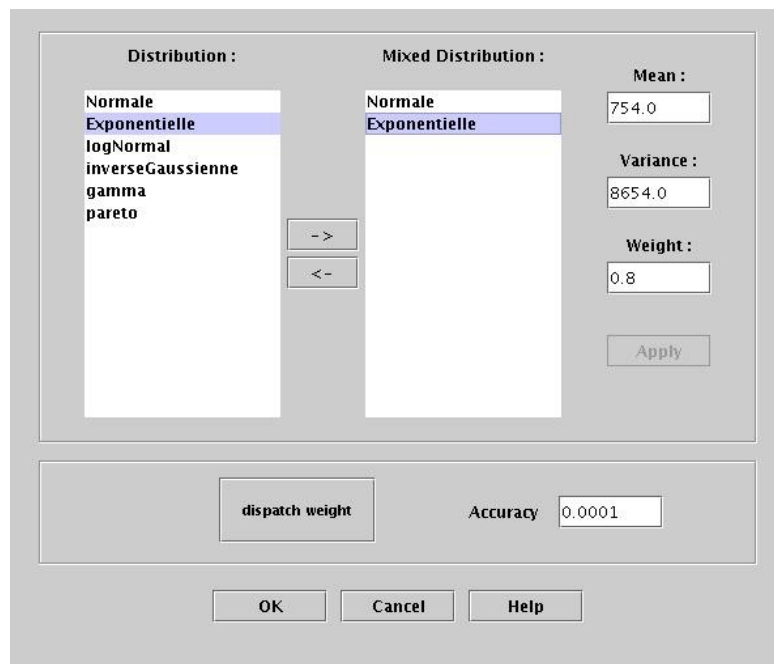


Figure 34 - Interface pour l'algorithme E/M

### 10.2.11 Validation

La phase de validation permet de valider la convergence des mélanges avec des échantillons générés par Matlab. Pour chaque distribution, nous choisissons un mélange de la même distribution. Par exemple pour la loi Normal, nous générons un mélange Normal-Normal avec pour chaque distribution un poids, une moyenne et une variance. Le test doit montrer que E/M retrouve un mélange qui ajuste l'échantillon.

Résultat des tests pour un échantillon  $n=10000$  et une précision  $10^{-4}$ , en respectant le domaine de validité des fonctions de densité.

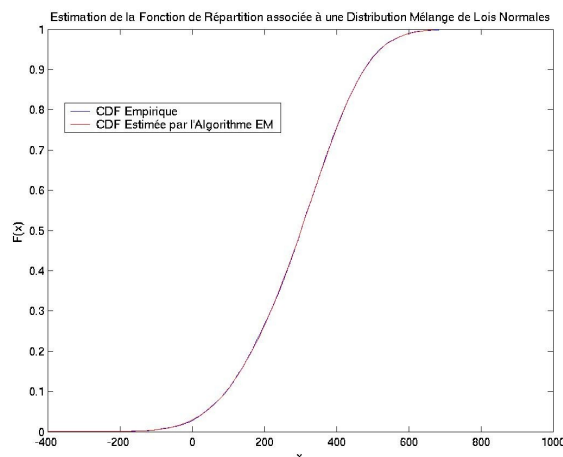
#### Normal - Normal

Généré par Matlab				Estimation E/M			
Généré	Poids	Moyenne	Variance	E/M	Poids	Moyenne	Variance
Normal	0.3	400	250	Normal	0.4217	186.65	15636.61
Normal	0.7	10000	20000	Normal	0.5782	370.37	12011.06 9

#### Test Kolmogorov-Smirnov

1%	5%	10%
$h = 0$	$h = 0$	$h = 0$
$p = 0.9751$	$p = 0.9751$	$p = 0.9751$
$k = 0.0048$	$k = 0.0048$	$k = 0.0048$
$cv = 0.0163$	$cv = 0.0136$	$cv = 0.0122$

P : valeur asymptotique du seuil maximum d'acceptation ; k : distance entre CDF empirique et CDF de  $\hat{F}(x)$  ; valeur tabulée pour les seuils de 1%, 5% et 10%



L'ajustement visuel est très bon, ceci est confirmé par les tests K-S et Cramer.

**Exponentielle - Exponentielle**

Généré par Matlab

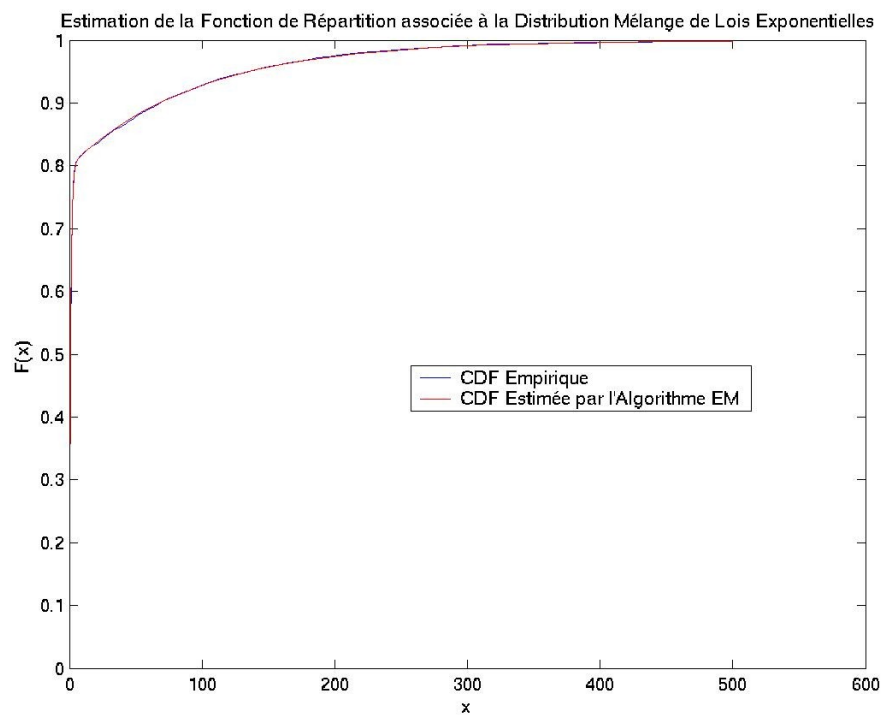
Généré	Poids	Moyenne	Variance
Exp	0.2	100	-
Exp	0.8	1	-

Estimation E/M

E/M	Poids	Moyenne	Variance
Exp	0.7994	0.9927	1.0145
Exp	0.2005	97.7268	1.0470E-4

Test Kolmogorov-Smirnov

1%	5%	10%
h = 0	h = 0	h = 0
p = 0.9299	p = 0.9299	p = 0.9299
k = 0.0054	k = 0.0054	k = 0.0054
cv = 0.0163	cv = 0.0136	cv = 0.0122



L'ajustement est de très bonne qualité.

Inverse Gaussienne - Inverse Gaussienne

Généré par Matlab

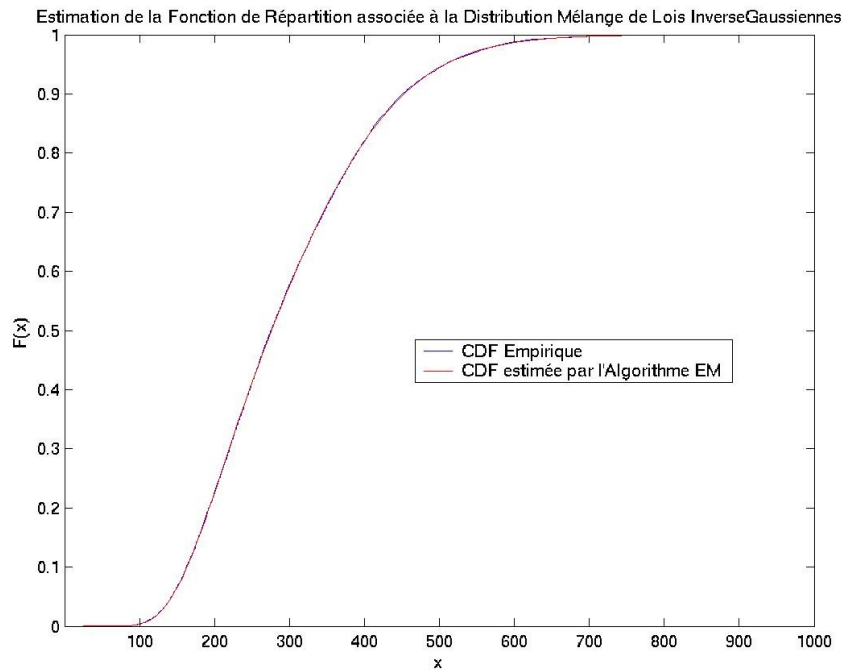
Généré	Poids	Moyenne	Variance
InverseG	0.7	250	7500
InverseG	0.3	400	9700

Estimation E/M

E/M	Poids	Moyenne	Variance
Inverse G	0.6789	244.94	6995.7930
Inverse G	0.3210	400.32	9715.2067

Test Kolmogorov-Smirnov

1%	5%	10%
h = 0	h = 0	h = 0
p = 0.99	p = 0.99	p = 0.99
k = 0.0035	k = 0.0035	k = 0.0035
cv = 0.0163	cv = 0.0136	cv = 0.0122



L'ajustement est de très bonne qualité.

**LogNormal - LogNormal**

Généré par Matlab

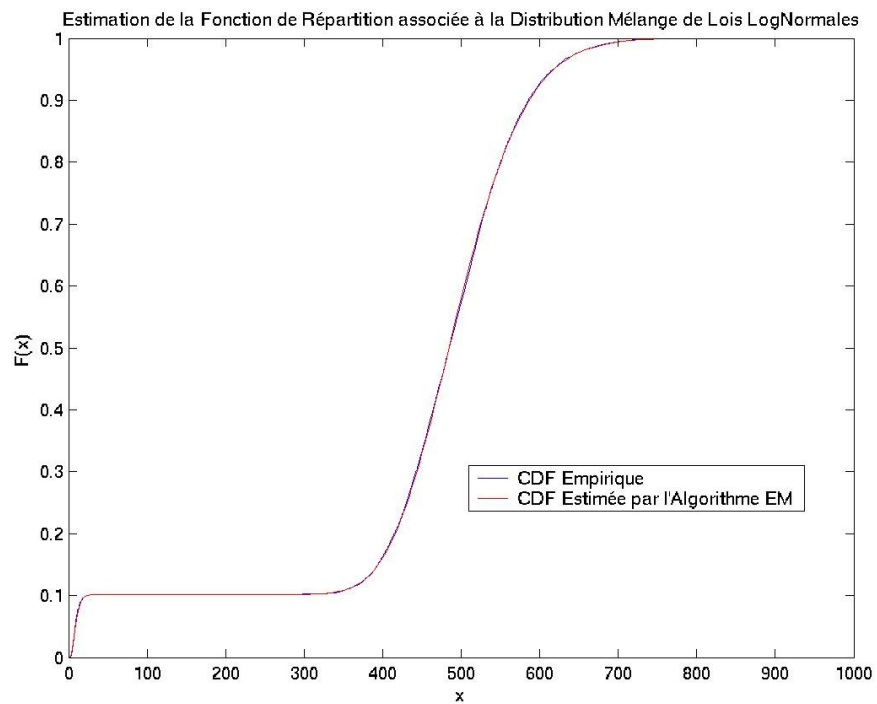
Généré	Poids	Moyenne	Variance
LogNormal	0.1	10	30
LogNormal	0.9	500	5000

Estimation E/M

E/M	Poids	Moyenne	Variance
LogNorma l	0.1019	9.7790	26.3520
LogNorma l	0.8980	499.151 6	4962.580 8

**Test Kolmogorov-Smirnov**

1%	5%	10%
h = 0 p = 0.2464 k = 0.0102 cv = 0.0163	h = 0 p = 0.2464 k = 0.0102 cv = 0.0136	h = 0 p = 0.2464 k = 0.0102 cv = 0.0122



L'ajustement est de très bonne qualité.

**Gamma- Gamma**

Généré par Matlab

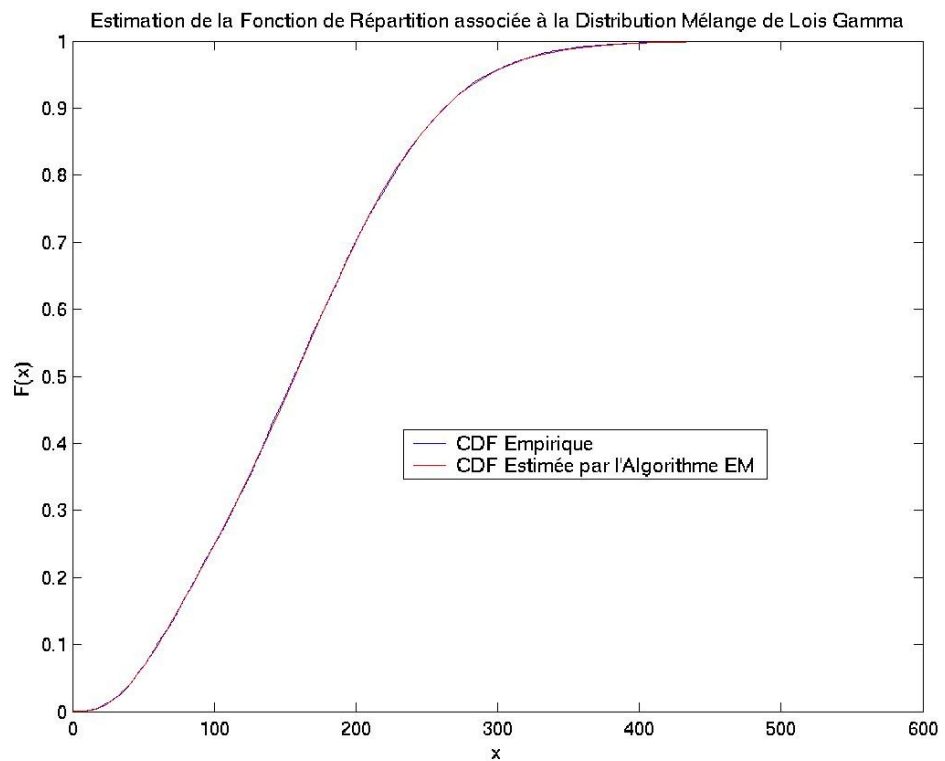
Généré	Poids	Moyenne	Variance
Gamma	0.5	100	3000
Gamma	0.5	200	4000

Estimation E/M

E/M	Poids	Moyenne	Variance
Gamma	0.4850	112.9348	3978.8690
Gamma	0.5149	204.6795	3780.4331

**Test Kolmogorov-Smirnov**

1%	5%	10%
h = 0	h = 0	h = 0
p = 0.94471	p = 0.94471	p = 0.94471
k = 0.005261	k = 0.005261	k = 0.005261
cv = 0.0163	cv = 0.0136	cv = 0.0122



L'ajustement est de très bonne qualité.

Mélange LogNormal, Inverse Gaussienne, Exponentielle

Généré par Matlab

Généré	Poids	Moyenne	Variance
LogNormal	0.2	150	3000
InverseGaussian	0.3	100	3000
Exponentielle	0.5	200	-

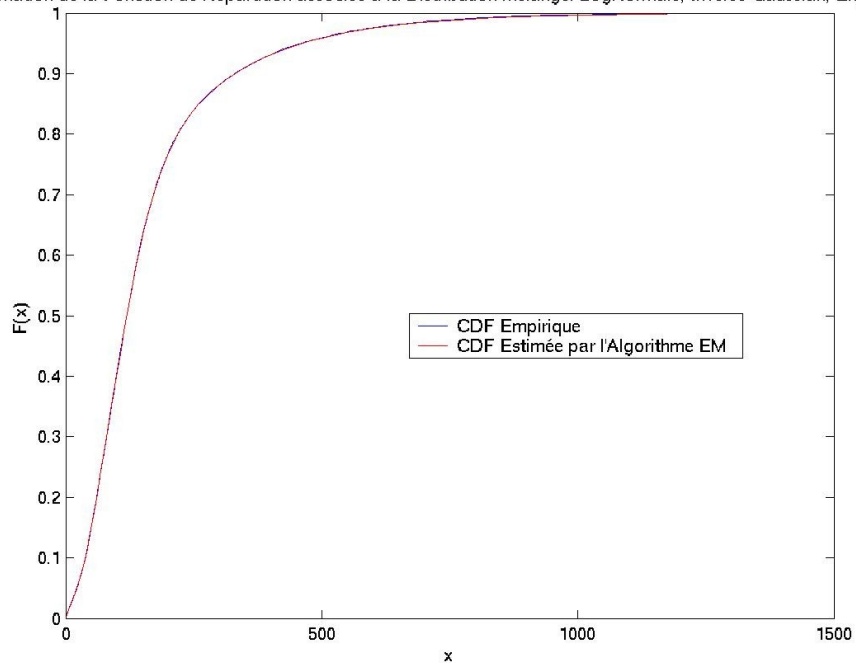
Estimation E/M

E/M	Poids	Moyenne	Variance
LogNormal	0.4016	115.144 4	4863.67 06
InverseGaussian	0.1211	147.113 8	1804.25 73
Exponentielle	0.4771	202.637 6	Mean2

Test Kolmogorov-Smirnov

1%	5%	10%
h = 0	h = 0	h = 0
p = 1.0000	p = 1.0000	p = 1.0000
k = 0.003406	k = 0.003406	k = 0.003406
cv = 0.0163	cv = 0.0136	cv = 0.0122

Estimation de la Fonction de Répartition associée à la Distribution Mélange: LogNormale, Inverse Gaussian, Exponential



E/M retrouve bien le mélange des échantillons, d'ailleurs ce test est le plus significatif dans le cas de découverte d'un échantillon inconnu.

### 10.2.12 Découverte d'un échantillon inconnu

L'estimation d'un échantillon inconnu passe par une phase de test avec plusieurs fonctions mixées :

- Faire un test E/M,
- Vérifier que les tests  $H_0$  soient respectés (on accepte l'hypothèse  $H_0$ ),
- Sélectionner les distributions avec un poids supérieur à 10%, et relancer E/M,
- Si aucun mélange ne convient au sens des tests statistiques, choisir le mélange qui minimise la distance des moindres carrés.

*Résultat des tests pour un échantillon inconnu d'inter-arrivée :*

- $n=10000$
- précision demandée de  $10^{-4}$
- Poids équitablement répartie (0.33, 0.33, 0.34)
- Moyenne :  $4.8822440098021235E-4$  Variance =  $3.382457512424475E-7$

*Etape 1 : la distribution normal ne présente pas d'intérêt, on note le caractère exponentielle de l'échantillon*

Estimation E/M

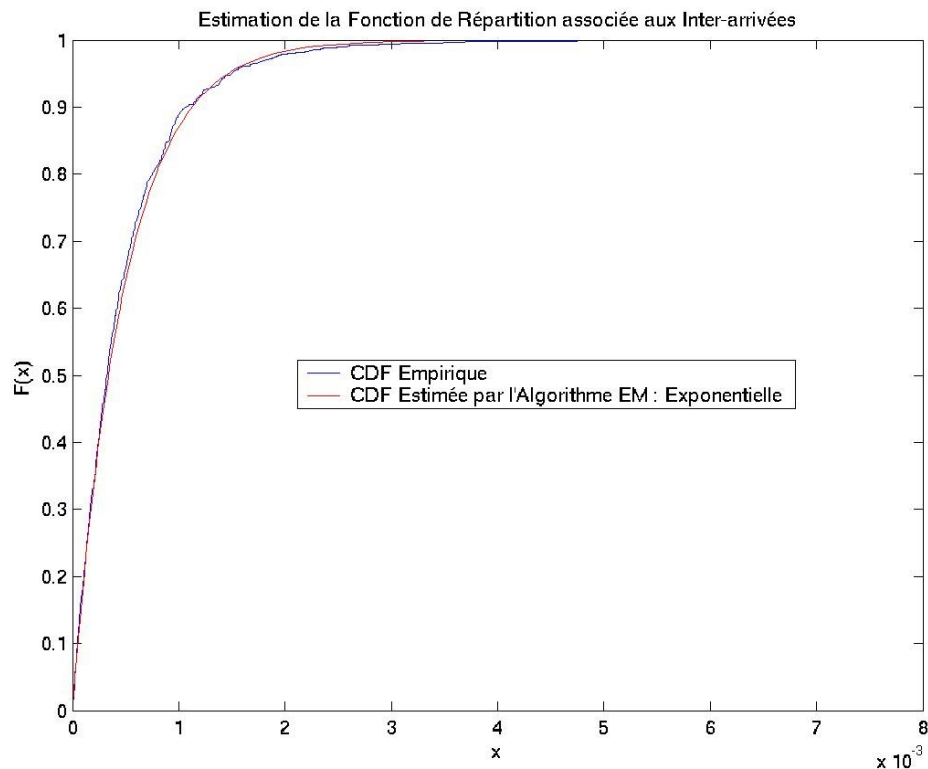
E/M	Poids	Moyenne	Variance
LogNormal	0.2069	8.9536E-4	7.1608E-7
Normal	0.00841	0.001778	1.6459E-7
Exponentielle	0.7846	3.6571E-4	-

*Etape 2 : Finalement l'échantillon est proche d'une exponentielle*

Estimation E/M

E/M	Poids	Moyenne	Variance
Exponentielle	1	4.8825E-4	-





L'étape 2 montre le caractère exponentiel de la distribution, cependant le test K-S ne passe pas à 5%. Néanmoins, c'est la distribution la plus proche proposée par cette méthode.

Test Kolmogorov-Smirnov

5%
h = 1 p = 7.6619e-09 k = 0.0311 cv = 0.0136

### 10.2.13 Conclusion

L'algorithme E/M donne des résultats analytiques proches des observations empiriques mais il est nécessaire d'ajouter les tests statistiques pour contrôler si l'ajustement est correct (K-S, Cramer). Si tous les tests sont négatifs, il faut choisir la distribution la plus proche en calculant la distance des moindres carrés entre  $F(x)$  et  $F'(x)$  et choisir la distribution la plus proche. Les distributions Weibull, Pareto ne permettent pas de trouver un mélange qui s'ajuste correctement. En effet pour l'étape  $(m+1)$  certains paramètres sont fixés sur le  $\min(X_i)$ , mais ce  $\min()$  n'est pas forcément celui de la distribution attendue.

## 11 Synthèse

---

L'Internet actuel a l'ambition d'être le réseau universel par excellence. Sa plus grande lacune est sans conteste l'absence de qualité de service et sa faible performance. L'avènement de nouvelles technologies comme MPLS, Diffserv, permet de concevoir des architectures plus adaptées aux applications multimédias, voix, vidéo à la demande, jeux, application point à point. L'architecture actuelle sans qualité de service (best effort) devrait être remplacée par le concept de différenciation de services (diffserv). Le déploiement et la gestion d'un tel système vont nécessiter le développement de nouveaux systèmes d'administration mais aussi de planification. Dans ce contexte, les modèles de trafic et les outils de simulations vont jouer un rôle de plus en plus important et vont pouvoir répondre aux défis que sont la taille, la complexité du déploiement et de l'utilisation d'un tel réseau à l'échelle planétaire.

Dans le domaine du réseau mobile, des efforts importants sont réalisés sur les modèles de trafic de type multimédia. Les modèles GPRS et UMTS présentés dans ce document montrent cependant que ceux-ci ont été largement conçus en fonction de données de métrologie de réseau LAN (ou dial-in). Des mesures fiables sur de vrais réseaux permettraient de s'affranchir de certaines hypothèses et d'obtenir des modèles de trafic plus fiables. Les paramètres des distributions doivent permettre d'ajuster le comportement des trafics de façon « intuitive » car certains modèles présentent des distributions peu probables (distribution de page WEB sur HTTP en GPRS). A noter que les modèles de trafic UMTS pour une bande passante de 2 Mb/s ne sont disponibles car il existe encore une trop grande incertitude pour le comportement du mobile et des réseaux de support.

Les modèles de trafic pour réseaux fixes semblent mieux maîtrisés, cependant on peut remarquer que les modèles se concentrent sur les applications les plus utilisées : WEB, vidéo, E-Mail, FTP. En considérant l'augmentation exponentielle du trafic des réseaux de données et de la disparité des applications, on peut en déduire qu'il existe peu de modèles de trafic bien maîtrisés. La conception de modèles de trafic génériques peut alors palier le problème de façon temporaire jusqu'à la définition de modèles plus précis. L'étude de la superposition d'une source unitaire permet d'en déduire le flux agrégé, ainsi le modèle analytique peut être utilisé pour simuler un grand nombre de source. Le modèle agrégé permet de réduire le temps de simulation et de réaliser des simulations à grande échelle.

Ce document a présenté des modèles de trafics fondés sur des mesures. La première étape de modélisation d'un modèle de trafic consiste à effectuer des mesures physiques sur un réseau et de définir un modèle général. Dans le cas des opérateurs de télécommunication, l'idéal serait d'avoir un lien entre la métrologie du réseau et la définition d'un modèle de trafic (distribution, moyenne, variance, tranche horaire, comportement des utilisateurs). Cette méthode dynamique permettrait de faire réellement de la planification de réseaux avec une grande précision et permettrait de tenir compte de l'émergence de nouveaux types de trafic sans remettre en cause les modèles de trafics et les développements logiciels.

Enfin, l'analyse statistique des sources de trafic permet de comprendre la complexité des comportements et de produire des modèles analytiques performants pour tester les réseaux in situ ou alors pendant les phases de maquettage.

## 12 Bibliographie

[1]	Alexander Klemm, Christoph Linderman, Marco Lohmann- University of Dortmund -Traffic Modelling and Characterization for UMTS Networks
[2]	Ho J & Zu, Wireless Packet Data traffic profile, Nortel Network, February 1999, Draft Version 0.1.1
[3]	Frost V & Melamed B, Traffic modelling for telecommunication network, IEEE Communication Magazine, 32(3), 70-81, 1994
[4]	Desaulniers-Soucy N & Luoras A, Traffic modelling with universal multifractal, Proceedings of IEEE Globecom'99, Rio Janeiro, Brazil, December 1999, pp 1058-1065.
[5]	Chan MC & Woo T, Next-generation wireless data services: architecture and experience, IEEE Personal communication Magazine, 6(1), 20-30, 1999
[6]	D. Staehle, K. Leibnitz and P. Tran-Gian, Source Traffic Wireless Applications, Report N°261, June 2000, University of Würzburg
[7]	Ho J Zhu Y & Madhavapeddy S, Throughput and buffer analysis for GSM general packet radio services, Proceedings of IEEE WCNC 99, New Orleans, September 1999
[8]	Hong D & Rapport SS, Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures, IEEE Transactions on Vehicular Technology, VT-35, 1986
[9]	Universal Mobile Telecommunication System (UMTS); selection procedure for the choice of radio transmission technologies of the UMTS, Technical Report TR 101 112 v3.2.0; ETSI, April 1998
[10]	Choi, H. and Limb, J. A behavioural Model of web traffic, International conference of networking protocol 99 (ICNP 99), Sep 1999.
[11]	Mah, B. An Empirical Model of HTTP Network Traffic, Proceedings of the IEEE INFOCOM 97, Kobe, April 1997, vol. 2, pages 592-600.
[12]	Reyes-Lecuona, A, Gonzales-Parada, E Casilari, E., and Diaz-Estrella, A. A page-oriented WWW traffic model for wireless system simulations, Proceedings of the 16th International Teletraffic Congress (ITC16), Edinburgh, United Kingdom, June 1999, pp. 1271-1280.
[13]	Vicari, N. Measurement and Modeling of WWW-Session, Institute of Computer Science, University of Wurzburg, Technical Report No 184.
[14]	ITU: US TG 8/1, Radio Communication Study Group, The Radio cdma2000 RTT candidate submission, TR45-5, June 98.
[15]	Etudes INRIA : <a href="http://mission-dti.inria.fr">http://mission-dti.inria.fr</a>
[16]	Barford, P. and Crovella, M. Generating representative workloads for network and server performance evaluation, Proceedings of ACM SIGMETRICS 98, page 151-160, june 1998.
[17]	Brasche, G., Walke, B. Concepts Services and Protocols of the new GSM Phase 2+ General Racket Radio Service, IEEE Communication Magazine, August 1997.
[18]	Peter Stuckmann, Henrick Finck, Thomas Bahls - A WAP Traffic and its appliance for the performance analysis of WAP overs GPRS, RWTH Aachen University of

## **Bibliographie**

---

	Technology, Germany.
[19]	Eric Thibault, Projet Mistral, INRIA -Projet Esquimaux - Etat de l'Art de la modélisation des flux multimédia.
[20]	David Gauchard, Simulation Hybride des réseaux IP-DiffServ-MPLS Multi-services sur Environnement d'Exécution Distribuée, Thèse UPS, CNRS/LAAS
[21]	Deng. S - Traffic Characteristics of Packet Voice. IEEE Journal, 1995, pp.1369-1374.
[22]	Krunz M. Makowski M. Modeling Video Traffic Using M/G/ $\infty$ Input Processes : A compromise between Markovian and LRD Models, IEEE Journal of Selected Areas in Communications, June 1998, pp.733-748.
[23]	Niclausse N. Modélisation, évaluation de performances et dimensionnement du WWW, Thèse, Université de Nice-Sophia Antipolis, Juin 1999
[24]	Liu Z. Niclausse N. Jalpa-Villanueva C et Barbier S.. Traffic Model and Performance Evaluation of Web Servers, Rapport de Recherche INRIA, décembre 1999
[25]	The Expectation Maximisation Algorithm - Frank Dellaert - Février 2002
[26]	Self similarity in World Wild Web Traffic - Evidence and possible causes - M.E Crovella - A. Bestavros

## 13 Glossaire

<b>ADPCM</b>	Adaptive Differential Pulse Code Modulation : CODEC pour la voix.
<b>ADSL</b>	Asynchronous Digital Subscriber Line
<b>CODEC</b>	COder-DECoder : système de codage ou de décodage électronique et/ou logiciel qui permet de convertir un signal analogique en signal numérique et vis versa avec une compression et contrôle de donnée.
<b>Délais</b>	Le délai est le temps de propagation d'un paquet de la source vers la destination.
<b>EGPRS</b>	Enhanced GPRS
<b>FEC</b>	forward error correction
<b>FGN</b>	Fractional Gaussian Noise
<b>FTP</b>	File Transfert Protocol
<b>Gigue</b>	La gigue représente les variations de délai qui perturbent le flux de paquet dans les réseaux locaux ou étendus.
<b>GOP</b>	Group Of Pictures
<b>GPRS</b>	General Packet Radio Service
<b>GSM</b>	Group System for Mobile communications
<b>ICMP</b>	Internet Control Message Protocol
<b>IP</b>	Internet Protocol
<b>IPP</b>	Interrupted Poisson Process
<b>ITU</b>	International Telecommunication Union
<b>JPEG</b>	Joint Photographic Expert Group
<b>LAN</b>	Local Area Network
<b>LRD</b>	Long Range Dependance
<b>LRD</b>	Long Range Dependance
<b>LSP</b>	Label Switched Path
<b>MG00</b>	File d'attente, arrivée Poisson, Service générale Moyenne/Variance, Infinité de serveur
<b>MMPP</b>	Markov Modulated Poisson Process
<b>MPLS</b>	Multi Protocol Label Switching. Protocole de commutation d'étiquettes, qui permet de « tager » les paquets de niveau 2, supprime le routage par la

## Glossaire

---

	communication de paquet, et réalise un routage sur chemin (LSP).
<b>NIS</b>	Network Information System
<b>OSPF</b>	Open Shortest Path First
<b>QoS</b>	Quality Of Services
<b>RIP</b>	Routing Information Protocol
<b>RSTP</b>	Real Time Streaming Protocol
<b>RSVP</b>	Resource ReSerVation Protocol
<b>RTCP</b>	Real Time Control Protocol
<b>RTP</b>	Real Time Protocol
<b>RTT</b>	Round-Trip Time
<b>SDP</b>	Session Description Protocol
<b>SMS</b>	Short Message Service
<b>SNMP</b>	Simple Network Management Protocol
<b>SRD</b>	Short Range Dependance
<b>SRD</b>	Shot Range Dependance
<b>SSH</b>	Secure Shell
<b>TCP</b>	Transfert Control Protocol
<b>UDP</b>	User Datagram Protocol
<b>UMTS</b>	Universal Mobile Telecommunication System
<b>VoIP</b>	Voice IP
<b>WAN</b>	Wide Area Network
<b>WAP</b>	Wireless Application Protocol
<b>WTP</b>	Wireless Transaction Protocol