■Thèse

Préparée au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

En vue de l'obtention du Doctorat de l'Université Paul Sabatier - Toulouse III

Ecole doctorale : EDSYS
Spécialité : Systèmes Informatiques

Par **Hassan HASSAN**

Modélisation et analyse de performances du trafic multimédia dans les réseaux hétérogènes

■Thèse

Préparée au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

En vue de l'obtention du Doctorat de l'Université Paul Sabatier - Toulouse III

Ecole doctorale : EDSYS
Spécialité : Systèmes Informatiques

Par **Hassan HASSAN**

Modélisation et analyse de performances du trafic multimédia dans les réseaux hétérogènes

Soutenue le 18 décembre 2006 devant le jury :

Président	Christian BES
Directeur de thèse	Jean-Marie GARCIA
Rapporteurs	Demetres KOUVATSOS Zhili SUN
Invités	Gérard AUTHIE Olivier BRUN

À la mémoire de mon père

À toute ma famille

Remerciements

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique (LAAS-CNRS), dirigé au cours de mon séjour par M. Malik Gallab, que je tiens à remercier cordialement pour son accueil. J'adresse également mes plus sincères remerciements à Mr. Gérard Authié, responsable du groupe RST, pour son suivi lors des deux premières années de thèse et ses précieux conseils sur ce manuscrit, et à Mr André Monin responsable du groupe MRS lors de ma troisième année de thèse pour sa disponibilité.

Je tiens à remercier très chaleureusement Mr Jean-Marie Garcia, Directeur de Recherche au LAAS, pour avoir accepté de m'encadrer lors de ces travaux de thèse et, bien sûr, pour ses nombreux conseils et questions qui me font toujours chercher un peu plus loin.

J'adresse mes sincères remerciements à Mr Demetres Kouvatsos, Professeur à l'université de Bradford, et Mr Zhili Sun, Professeur à l'université de Surrey, pour avoir accepté d'être rapporteurs de ces travaux.

J'adresse mes remerciements à Mr Christian Bes, Professeur à l'université Paul Sabatier, d'avoir accepté de présider le jury et à Mr Olivier Brun, Chargé de Recherche au LAAS, pour sa participation au jury ainsi que pour ses précieux conseils et critiques lors de ces trois années de thèse.

Je remercie également Mr David Gauchard, Ingénieur de Recherche au LAAS, pour son aide technique indispensable lors de cette thèse et Mr Eric Thibault, Ingénieur de recherche à QoS Design, d'avoir guidé mes premiers pas dans le domaine des statistiques.

Tous mes remerciements à mes amis et collègues de travail (Wael, Charles, François, Urtzi, Frédéric, Cédric …) pour leurs encouragements aux moments difficiles.

Je remercie aussi toutes les personnes du LAAS qui m'ont permise d'avoir un cadre de travail agréable (Mme B. Ducrocq, Service II, Service Documentation). Ils ont tous contribué à m'offrir de bonnes conditions matérielles de travail.

Enfin, je souhaite remercier sincèrement tous mes proches : Nazha, Maha, Maya, Naya, Ali, Salwa, Jafar pour leur présence et leur soutien. Un grand merci !

Contents

Part I

Présentation des Travaux de Thèse

Chapter 1

Présentation des Travaux de Thèse

1.1 Introduction

Les réseaux de télécommunications ont évolué énormément depuis l'introduction de l'Internet. Les services offerts aux utilisateurs ont été diversifiés. Ainsi en plus des simples services voix offerts par les réseaux téléphoniques traditionnels à commutation de circuits, les réseaux à commutation de paquets proposent des applications multimédia avec des services audio, vidéo et données intégrés. Néanmoins ce type de services a soulevé la problématique de la Qualité de Service (QdS) des applications multimédia. Surtout les applications temps réel de type audio et vidéo, qui ont des besoins stricts en terme de délai et de gigue.

Les opérateurs de télécommunications sont amenés à évaluer la QdS des applications proposées aux utilisateurs avant de les déployer sur leurs réseaux. Cette évaluation est souvent menée par des études de simulation des réseaux à très grande échelle. La fiabilité des résultats obtenus repose essentiellement sur la fiabilité des modèles utilisés pour les applications ainsi que pour les équipements réseaux.

Dans cette thèse nous adressons la problématique de la QdS des applications multimédia dans les réseaux hétérogènes. Nous nous intéressons principalement à la modélisation du trafic issu des applications multimédia. D'abord, nous proposons une plateforme générique et hiérarchique pour la modélisation du trafic. Cette plateforme est implémentée dans outil de simulation qui est utilisé pour les simulations au cours de cette thèse. Ensuite nous modélisons le trafic Internet et les applications multimédia. Puis nous proposons un modèle performant pour le protocole de transport réactif TCP (Transmission Control Protocol) qui est le protocole dominant sur Internet. Enfin nous adressons la problématique de QdS au niveau application par extension de l'architecture du protocole SIP (Session Initiation Protocol) et l'utilisation des techniques d'évaluation de la bande passante équivalente. Les résultats de nos travaux sont publiés dans [HGCB06, HGB06d, HGB06c, HGB06a, HG06, HG05, HGB06b, HGB06f, HGB06e]

1.2 Réseaux Hétérogènes Multiservices

L'évolution des réseaux de télécommunication ces dernières années a changé la perception des services de communication par les utilisateurs. Alors que les réseaux téléphoniques traditionnels (à commutation de circuits) étaient limités aux services de type voix, Internet et son interface le Web ont permis l'émergence des réseaux multiservices. Les réseaux multiservices transportent tous types d'applications:Voix, Vidéo et Données à travers le même support (à commutation de paquets) utilisant le protocole IP (Internet Protocol). Toutefois la diversité des applications se traduit par une diversité des besoins en terme de QdS. Le support IP est un support de service au mieux (best-effort) qui ne peut pas garantir la QdS des applications multimédia. Heureusement la communauté de l'Internet a fait introduire des mécanismes innovants pour le control et la gestion de la QdS qui permettent conjointement avec les protocoles de transport (TCP, RTP, ...) de transporter des applications multimédia sur un support IP.

Récemment le support sans fil des communications a révolutionné les réseaux de télécommunication. De ce fait, les utilisateurs ont accès aux services multimédia à travers des terminaux sans fils (Téléphone mobiles, PDA, ...). La diversité des réseaux d'accès filaires et sans fils, ont fait apparaître une tendance de convergence vers un support commun de télécommunication: c'est le support IP. Ce support est le mieux adapté à répondre aux besoins des applications en terme de QdS dans un environnement hétérogène. De plus, de nouveaux protocoles de transport et de signalisation sont apparus pour mieux gérer la QdS des applications multimédia, par exemple le protocole SIP (Session Initiation Protocol).

Les réseaux de télécommunications de nos jours (et de demain) sont fortement hétérogène et offrent plusieurs services en même temps. Les réseaux hétérogènes multiservices convergent vers un support unique (le support IP). Cette convergence nécessite un important effort de standardisation entre constructeurs et concepteurs de réseaux. Dans la suite de ce paragraphe nous présenterons brièvement les différents réseaux d'accès sans fils qui forment cet environnement hétérogène avec les prédispositions pour garantir la QdS prévues par chaque réseau. Puis nous présenterons les mécanismes déployés sur les réseaux IP pour garantir la QdS sur un support best-effort.

1.2.1 Réseaux d'accès

1.2.1.1 GSM et GPRS

Les systèmes mobiles de la deuxième génération 2G comme le GSM (Global System for Mobile communications) sont conçus pour la téléphonie mobile, autrement dit pour des communications en mode circuit à faible débit. Pour améliorer l'efficacité du transfert de données, une évolution majeure du GSM est normalisée sous le nom de GPRS (General Packet Radio Service). Fondée sur l'interface radio du GSM, mais développant un partage de ressources dynamique adapté au trafic sporadique, le GPRS introduit une architecture réseau en mode paquet. L'association des services données avec le GPRS est souvent considérée comme un système 2,5 G, intermédiaire entre les systèmes 2 G (GSM, etc.)

et 3 G (UMTS, etc.).

Le GPRS réutilise les éléments du GSM pour le sous-ensemble radio. Mais son réseau cœur (Core Network) diffère sensiblement en introduisant de nouveaux éléments et en utilisant des protocoles spécifiques. Pour acheminer efficacement les paquets des mobiles vers les réseaux paquet publics, tels que X.25 ou IP, le GPRS met en place un réseau fixe à commutation de paquets constitué de routeurs. Aux frontières de ce réseau paquet, on distingue deux types de routeurs, dotés de fonctions particulières: le SGSN (Serving GPRS Support Node), côté sous-système radio, et le GGSN (Gateway GPRS support Node), côté réseau de données public.

Entre le GGSN et le SGSN, les données utilisateur sont simplement encapsulées par le protocole GTP (GPRS Tunnelling Protocol). Le réseau de transport est un simple réseau IP. On peut se trouver ainsi dans la situation où, entre le mobile et son correspondant, des paquets IP se retrouvent encapsulés dans des paquets IP du réseau cœur GPRS entre le GGSN et le SGSN. Une session est établie après attachement au réseau par l'activation d'un contexte PDP (Packet Data Protocol). Ce contexte PDP permet de rendre le mobile visible à l'extérieur du réseau de l'opérateur mobile, en lui associant, par exemple, une adresse reconnue du réseau extérieur : adresse IP, X.121, etc.

Pour exprimer la qualité de service, le mobile dispose d'un ensemble de classes de paramètres: la priorité du service, sa fiabilité, les délais tolérés, le débit moyen et enfin le débit pic des informations transmises. Tous ces paramètres sont négociés entre le GGSN et le mobile lors de l'activation du contexte PDP. En revanche, les moyens pour garantir la qualité de service négociée, c'est-à-dire la définition des stratégies d'allocation de ressources, sont à la charge de l'opérateur, ou du constructeur, mais ne sont en aucun cas standardisés. Trois classes de priorité, haute, normale et basse, sont définies pour différencier les services en cours. Elles caractérisent la précédence du service requis par l'utilisateur et permettent au réseau d'identifier les données à supprimer, par exemple, en cas de congestion du réseau, les services de classes de priorité basse étant interrompus les premiers.

1.2.1.2 UMTS et IMS

Une majeure évolution du système GSM/GPRS est le passage vers les réseaux de troisième génération dit UMTS (Universal Mobile Telecommunications System). On note une évolution importante du débit théorique qui passe de 160 Kbps maximum pour le GPRS à 2 Mbps prévu par l'UMTS. Dans la première spécification du système UMTS définit par le 3GPP connue sous le nom R4 (ou R99), les deux mondes circuits et paquets restent séparés. Ainsi le système comporte deux cœurs de réseaux: un cœur à commutation de circuits CS-CN et un autre à commutation de paquets PS-CN. La partie à commutation de circuits garde les mêmes fonctionnalités du système GSM pour le transport de la voix. On note surtout que la signalisation est basée sur le système SS7. Alors que la partie à commutation de paquets prend en charge le transport des applications multimédia.

Bien que le système R4 reste simple et cohérent avec le GSM, il impose aux opérateurs de garder deux réseaux séparés. Le coût du maintien et d'exploitation est assez élevé. L'idée de rapprocher les deux parties paraît très profitable. La solution est de supprimer

la partie à commutation de circuits et aller vers un monde tout IP dans lequel toutes les applications Voix, Vidéo et Données passent par un réseau à commutation de paquets de type IP : c'est le système R5. Il n'y pas de cœur de réseau à commutation de circuits dans R5. Un seul cœur à commutation de paquets par lequel passe toutes les sessions multimédia. L'IMS (IP Mutlimedia Subsystem) fait son apparition dans cette architecture. Les appels téléphoniques ne disposent plus de circuits. Ils doivent passer par le réseau IP. Ceci impose un interfaçage (réalisé par l'IMS) entre les réseaux téléphoniques traditionnels et le nouveau cœur IP. Le système de signalisation traditionnel SS7 est remplacé par le protocole SIP de l'IETF. Les classes de services de l'UMTS sont définit dans le Tableau **2.1**.

1.2.1.3 LAN sans fil

Les réseaux locaux sans fil (WLAN : Wireless LAN) présentent de nombreux avantages sur les réseaux câblés classiques, en termes de productivité, simplicité et coût. La norme 802.11 pour les réseaux sans fil est l'une des technologies les plus répandues dans le monde. Deux configurations sont possibles pour les WLAN:

- Ad-hoc (mode de base) : Cette configuration est identique à sa cousine filaire (à l'exception des câbles). Deux stations ou plus peuvent communiquer sans point d'accès. Lorsque deux ou plusieurs stations composent un réseau de base, on parle de configuration IBSS (Independent Basic Service Set).
- Clients/serveur (mode infrastructure) : Cette configuration est formée de plusieurs stations connectées à un point d'accès agissant comme une passerelle vers un réseau filaire.

Deux méthodes d'accès sont proposées par la norme 802.11 : Distributed Coordination Function (DCF) et Point Coordination Function (PCF). DCF et PCF font partie de la spécification de la couche MAC qui s'occupe de la gestion de l'accès de plusieurs stations à un support partagé. La QdS proposée par DCF est de type best-effort. Le mode PCF prévoit une différenciation de service entre les stations mais malheureusement n'a jamais été implémenté par les constructeurs (peu efficace). L'extension de la norme 802.11e vient corriger cette situation et propose deux mécanismes plus adapté aux besoins des applications en termes de QdS : Enhanced DCF (EDCF) et Hybrid Corrdination Function(HCF). Les classes de service proposé par 802.11e sont résumées dans le Tableau **2.3**.

1.2.2 Réseaux Multiservices IP

L'Internet permet le déploiement d'applications multimédia ayant des exigences spécifiques en terme de QdS. Certains services comme les services vocaux ont besoin d'un faible délai point à point et d'une faible gigue. D'autres comme les trafics de données

nécessiteront des faibles taux de pertes ou d'erreurs, sans retransmission, avec éventuellement une certaine garantie de bande passante pour les trafics critiques comme les données transactionnelles.

Pour pouvoir garantir la QdS des flux transportés, il va donc falloir utiliser des mécanismes permettant de traiter de manière différenciée les différentes catégories de trafic dans les organes du réseau ainsi que des protocoles de signalisation de la QdS pour pouvoir allouer des ressources en fonction des besoins des applications.

On peut distinguer deux grandes problématiques pour la gestion de la QdS dans un réseau IP: La gestion des phénomènes de congestion et l'ordonnancement (Scheduling) des paquets.

- La gestion des phénomènes de congestion: Il s'agit un point fondamental pour garantir la QdS des flux. En effet, les mécanismes de gestion des QdS n'auront un impact effectif que lorsque le réseau sera chargé. Il existe deux grandes approches pour la gestion des congestions: les méthodes réactives et les méthodes préventives. La philosophie du contrôle réactif est d'accepter un maximum de connexions. Lors de la congestion d'un équipement réseau (mesurée en terme de pertes, de délais et de remplissage des tampons), les sources réduisent leurs débits. Ainsi, dans le modèle best-effort (modèle "au mieux") du réseau Internet actuel, la gestion du phénomène de congestion est faite de manière réactive par le mécanisme de fenêtre glissante (adaptation du débit) du protocole TCP. Toutefois, ce contrôle n'est pas adapté pour pouvoir offrir des garanties de QdS par exemple à des flux audio et vidéo temps réel: ils ne peuvent pas adapter leur débit. A l'inverse, le contrôle préventif consiste à prendre des mesures "à priori", afin de minimiser l'apparition du phénomène de congestion et/ou pour qu'il n'affecte pas les services garantis. Ainsi, les techniques de contrôle d'admission ou de mise en forme des trafics (Policing, Shaping) permettent de réduire la fréquence des congestions tandis que les mécanismes de gestion de tampon (RED, WRED, etc.) permettent de respecter les QdS des services les plus prioritaires en cas de congestion.

- L'ordonnancement (Scheduling) des paquets est aussi un mécanisme fondamental pour garantir la QdS des flux transportés. Ceci est évident si on considère des flux hétérogènes: les rafales de certaines connections pouvant perturber le trafic temps réel même s'il n'y a pas congestion. Ainsi, bien que la mise en oeuvre d'ordonnancements autres que FIFO (First in, First out: Premier Entré, Premier Sorti ou PAPS) soit difficile sur des routeurs à très haut débit, les équipementiers réseau fournissent de plus en plus des mécanismes sophistiqués réalisant un ordonnancement prenant en compte les classes de trafic (WRR, WFQ, etc).

Ces différents mécanismes de contrôle de congestion et d'ordonnancement des paquets sont présents dans toutes les architectures développées pour le contrôle de la QdS dans les réseaux IP. Historiquement, la première architecture qui a été proposée associe, comme ATM, une QdS à chaque flux que le réseau transporte. Il s'agit du modèle IntServ (Integrated Services: Service intégré). Pour des raisons de scalabilité principalement,

la communauté Internet a également proposé un modèle appelé DiffServ (Differentiated Services: Service différencié) dans lequel la QdS est associée à une agrégation de flots. Ce regroupement est appelé "classe de service". Ces deux modèles de réseaux IP-Multiservices peuvent être utilisés avec le protocole de commutation MPLS (Multi Protocol Label Switching) qui permet un traitement très rapide des paquets sur les routeurs du coeur du réseau en remplaçant la fonction de routage IP par une fonction de commutation, beaucoup plus adaptée au débit des réseaux de transport actuels. Le protocole MPLS encapsule ainsi le protocole IP.

1.2.3 Problématique

Il est évident que le déploiement de nouvelles applications multimédia sur les réseaux hétérogènes multiservices nécessite des études d'évaluation de performances. Ces études permettront aux opérateurs de dimensionner leurs réseaux de façon à ce que les contraintes imposées par les applications multimédia soient respectées. On parle généralement de SLA (Service Level Agreement) a respecter. Ceci étant de plus en plus important avec la convergence vers un support unique à commutation de paquets de type IP. Dans cette thèse nous allons adresser le problème de la modélisation du trafic issu des applications multimédia. La modélisation fiable du trafic influence directement les résultats obtenus par les études de simulations. De plus nous traitons l'efficacité de la modélisation quand il s'agit de la simulation des réseaux à très grande échelle.

1.3 Plateforme Générique pour la Modélisation du Trafic

Les opérateurs de télécommunications ont besoin d'outils de simulations qui intègrent des modèles réels, des différents flux qui représentent le trafic transitant sur leurs réseaux, pour pouvoir les dimensionner de la manière la plus optimale. Toutefois, 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 ou exponentiel. D'autre part une source de trafic peut produire des rafales de paquets qui influent fortement sur les ressources réseaux. La simulation d'une source de trafic consiste à trouver un modèle qui permet de générer des dates de création de paquets suivant le type de l'application. Le trafic généré doit avoir les mêmes propriétés statistiques (distribution de taille de paquets, distribution d'inter-arrivées, corrélations, ...), et la même performance dans un réseaux que le trafic modélisé.

Cette problématique est généralement traitée de deux façons différentes. D'abord on considère la modélisation de traces de trafics (éventuellement des traces IP). Dans cette approche, des mesures sont faites sur les liens réseaux et des traces comprenant les tailles de paquets et leurs dates de création sont fournies. La modélisation consiste à trouver un modèle mathématique capable de reproduire un trafic conforme à la trace. Dans cette approche on ne prend pas en compte la nature des paquets présents dans la trace ni la structure des applications qui les produisent. Alors que dans la deuxième approche on

se place à la source et on modélise le comportement de la source (ou l'application) et le processus de la génération des paquets en fonction du protocole de transport.

Dans les deux approches les besoins en termes d'outils d'analyse statistique et d'analyse de performances sont pratiquement les mêmes. La diversité des applications multimédia et la diversité des propriétés statistiques qu'on peut trouver dans les traces nous ont motivé à proposer une plateforme unique et générique pour modéliser les applications multimédia et les sources de trafic. Cette plateforme générique, baptisée Traffic Source Modeller (TSM), vise à proposer un modèle flexible, générique, et hiérarchique des sources de trafics (et des applications multimédia). Dans les paragraphes suivants nous allons détailler le modèle générique qui représente le noyau de cet outil.

1.3.1 Modèle Générique

Le modèle générique distingue deux catégories d'applications selon les contraintes temporelles imposées. Les applications temps réel, souvent associées au protocole de transport UDP (User Datagram Protocol), et les applications non temps réel associées au protocole de transport TCP (Transmission Control Protocol). Dans les applications temps réel, c'est la source qui définit (en fait elle impose) la séquence des paquets et donc les dates d'inter-arrivées. Les paquets arrivent à destination en respectant un profil dynamique pour 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 paquets) mais très faibles sur ce type de sources. Ces limites permettent à la destination de bien recréer le signal original sans perte d'informations perceptible.

Dans les applications non temps réel, la source n'impose pas le débit. Il s'agit en général de transfert de données, pour lesquels on tolère des délais très variables mais sans aucune perte. Ces applications qui reposent sur TCP génèrent un trafic dit "élastique" avec les caractéristiques suivantes:

- 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 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.

En se basant sur les observations précédentes, le modèle générique que nous proposons cherche à représenter une source de trafic en trois niveaux:

- Niveau Session: Pour modéliser l'arrivée de tous les clients qui se connectent au système, lancent une application, utilisent cette application pendant un certain temps, puis se déconnectent.

- Niveau Activité: Pour modéliser selon l'application demandée, la densité d'informations au cours du temps. Ce niveau décrit d'une façon détaillée la succession des périodes d'activités et inactivités de l'application.
- Niveau Paquet: C'est le niveau le plus élémentaire où on génère des paquets avec des tailles et des inter-arrivées spécifiques à l'application et au protocole de transport utilisé (UDP ou TCP). (Voir Figure **3.3**)

Notre approche repose sur la généralisation du modèle ON-OFF. Le but est de décrire d'une façon générique et hiérarchique les applications multimédia quelque soient leurs niveaux de complexité, comme une succession de périodes d'activité et d'inactivité à travers le temps. La généricité du modèle réside dans la capacité du modèle à détailler une période d'activité ON en sous période ON, permettant ainsi de rendre le modèle plus précis. La hiérarchisation en trois niveaux offre une souplesse dans la description des applications et la séparation entre le les deux modes de transport (UDP et TCP) rend possible la prise en compte du contrôle en boucle fermée du protocole TCP. Il important de noter que chaque source est définie comme étant un ensemble de flux décrits eux mêmes comme une succession de période ON et OFF. De plus il possible de synchroniser des périodes entre différents flux pour modéliser le comportement des protocoles complexes comme RTCP et SIP ou certaines activités déclenchent d'autres.

1.3.2 Outils Statistiques

Les sources de trafics décrites avec l'outil TSM sont utilisées dans les simulations de réseaux pour des fins d'évaluation de performances. Toutefois une caractérisation du trafic généré par les sources est souvent nécessaire. Cette caractérisation sert à déterminer le profile des paquets générés pour proposer des modèles agrégés plus simples et plus efficaces en simulation. Un ensemble d'outils statistiques a été développé et annexé au logiciel TSM pour permettre une caractérisation facile du trafic généré. Ces outils comprennent des évaluateurs des moment (moyenne, variance, ...), estimateur de fonctions d'autocorrélation (via l'algorithme Levenberg-Marquardt), estimateur de distributions (via l'algorithme Expexctaion Maximization), estimateur du coefficient de covariance cumulée, estimateur R/S du paramètre de Hurst, ...

1.4 Modélisation du Trafic IP

Depuis la révélation par Paxson et Floyd [PF95] de l'inadéquation du processus de Poisson pour la modélisation du trafic des réseaux LAN et WAN, plusieurs études ont mis en évidence l'existence de différents types de corrélations dans le trafic Internet et plus particulièrement des corrélations à long terme. Toutefois, la diversité des applications déployées sur Internet ainsi que la nature hétérogène des réseaux d'accès (filaire et sans fil) font évoluer les caractéristiques du trafic Internet. On peut distinguer deux grands catégories d'approches pour la modélisation du trafic IP: la modélisation au niveau flux et la modélisation au niveau paquet. Dans l'approche de modélisation au niveau flux

les paquets sont regroupés en se basant sur un ensemble d'informations caractérisent un flux. Ainsi on considère un flux comme étant un flot de paquets avec les mêmes adresses IP source et destination, les mêmes numéros de ports source et destination et le même numéro de protocole. Bien que l'approche flux soit très bien adaptée à la modélisation de trafic comme un outil de gestion de ressources réseau globales, elle ne permet pas d'étudier la congestion du réseau au niveau paquets. Ainsi une deuxième approche très largement utilisée consiste à modéliser le trafic au niveau paquet soit par un processus de comptage de paquet, soit par un processus d'agrégats d'octets (ou de paquets) entrant dans le réseau dans un intervalle de temps. Vue le nombre de paquets dans les traces IP, le processus de comptage de paquets rend l'estimation longue et coûteuse et par conséquent peu utilisable. On lui préfère généralement le processus d'agrégats d'octets. Cependant, la raison principale qui motive ce choix est que les corrélations qu'on a mises en évidence dans le trafic Internet étaient principalement sur les agrégats d'octets. De ce fait, de nombreux modèles mathématiques ont été étudiés pour essayer de capter les corrélations d'agrégat d'octets.

Nous modélisons le trafic IP en se basant sur la méthode d'agrégats d'octets, ou la trace IP est divisée en tranche de temps constantes (dites slots) et on cherche à modéliser la quantité d'information qui entre dans le réseaux par slots. Pour cela nous utilisons le modèle $M/G/\infty$ comme modèle support et des distributions mélanges pour modéliser les distributions de taille de paquets et de taille de slots

1.4.1 Modèle $M/G/\infty$

Le modèle $M/G/\infty$ représente un processus d'occupation d'une file d'attente avec des clients qui arrivent selon une loi de Poisson de paramètre λ (génération de paquet avec inter-arrivées exponentielles), une loi de service G (définie par sa moyenne et sa variance) et un nombre infini de serveur.

Le processus d'occupation noté X_n avec $n = 0, 1, 2, ...$ représente le nombre de clients dans le système au début de l'intervalle temporel $[n,\ (n+1)[$. Le processus du serveur occupé résultant $(X_n)_{n\geq 0}$ est corrélé mais n'est pas stationnaire en général. Si on note $\sigma_{n,i}$ la durée de service du i^{ieme} client dans le système à la date n, nous devons choisir les paramètres initiaux comme suit pour que le processus $M/G/\infty$ démarre dans le régime stationnaire :

- X_0, qui est le nombre de client dans le système à la date $n = 0$, soit distribué selon une loi de Poisson avec un paramètre $\lambda E(\sigma)$. λ étant le paramètre de la loi M du processus$M/G/\infty$.
- $\forall i \geq 1$, la variable aléatoire $\sigma_{0,i}$ est indépendante et identiquement distribuée avec une fonction de probabilité :

$$P(\sigma_{0,i} = k) = \frac{P(\sigma \geq k)}{E(\sigma)} \tag{1.1}$$

Selon ces conditions initiales le processus d'occupation $(X_n)_{n\geq 0}$ vérifie les propriétés suivantes :

- $\forall n \geq 0$, la variable aléatoire X_n est distribuée selon une loi de Poisson avec le paramètre $\lambda E(\sigma)$.
- La structure de corrélation associée avec le processus $(X_n)_{n\geq 0}$ est entièrement définie par la fonction de probabilité σ (associé à la loi générale G du processus$M/G/\infty$). En effet la fonction d'autocorrélation est donnée par :

$$\rho(k) = \frac{P(\sigma \geq k)}{E(\sigma)} \tag{1.2}$$

Il s'en suit que:

$$\sum_{k=0}^{+\infty} \rho(k) = \frac{1}{2} + \frac{E(\sigma^2)}{2E(\sigma)} \tag{1.3}$$

Ainsi le processus $(X_n)_{n\geq 0}$ peut générer un comportement SRD (Short Range Dependence) ou LRD (Long Range Dependence) selon la valeur de $E(\sigma^2)$. Autrement dit, le processus d'occupation $(X_n)_{n\geq 0}$ est SRD si $E(\sigma^2) < \infty$ ou bien LRD si $E(\sigma^2) = \infty$.

Inversement le choix de la structure de corrélation permet de caractériser la distribution du temps de service. Ainsi nous avons :

$$P(\sigma = k) = \frac{\rho(k+1) - 2\rho(k) + \rho(k-1)}{1 - \rho(1.1)} \tag{1.4}$$

Cette propriété permet la définition de la loi de service du processus $M/G/\infty$ en fonction de la fonction d'autocorrélation estimée de la taille des données sur un intervalle de temps. Le modèle de la fonction d'autocorrélation dépend de la structure de la corrélation existante. Nous estimons ces corrélations par une fonction d'autocorrélation $\rho(k)$ mélange de deux fonctions d'autocorrélation SRD et LRD définit comme suit :

$$\rho(k) = \alpha.\rho_1(b_1, k) + (1 - \alpha).\rho_2(b_2, k) \tag{1.5}$$

b_1 et b_2 sont deux constantes strictement positives et $\rho_1(k) = e^{-b_1 * \sqrt{k}}$pour la composante SRD et $\rho_2(k) = (k+1)^{-b_2}$ pour la composante LRD.

En outre, la distribution marginale des tailles de données observées sur des "slots" de temps est en général mieux estimée avec une distribution mélange $f(x)$ définit par $f(x) = p * f_1(x) + (1-p) * f_2(x)$. Dans l'étude expérimentale menée une distribution mélange de type Gamma + Lognormal s'avère très bien adaptée pour estimer la distribution marginale de taille des données :

$$f(x) = p * \log n(\mu, \sigma) + (1-p) * gamma(\alpha, \beta) \tag{1.6}$$

1.4.2 Validation et Performances

Nous utilisons des traces relevées sur Internet pour valider le modèle. Nous évaluons les modèles estimés statistiquement en comparant la distribution de tailles de données formées à partir des traces réelles à celles générées par le modèle. Le modèle réussit parfaitement à capter la distribution de tailles de données et leurs corrélations. Toutefois, il faut évaluer le modèle dans un environnement de simulation. Dans ce but nous choisissons d'injecter le trafic généré par le modèle ainsi que le trafic de la trace dans une file d'attente avec service déterministe (interface d'un routeur) et nous mesurons le taux de perte observé dans la file ainsi que la charge moyenne de la file. Les résultats de la simulation du modèle dans une file à service déterministe sont trop optimistes par rapport à ceux obtenus par l'injection de la trace dans la même file. Bien que la distribution des tailles de données soit excellente et que le débit moyen généré soit satisfaisant, la charge observée dans la file ainsi que le taux perte sont optimistes dans le modèle par rapport à la réalité.

1.4.3 Analyse et Amélioration du Modèle

Une analyse des distributions de taille de paquets et d'inter-arrivée de paquets à l'intérieur des slots a mis en évidence une dépendance entre la distribution de taille de paquets et la taille du slot considéré. En fait, la moyenne des tailles de paquets varie en fonction de la taille du slot. Ainsi on constate que cette moyenne augmente avec la taille du slot. Cette variation a bien évidemment un impact sur la loi d'inter-arrivée de paquets et sur la performance du trafic généré.

Nous avons développé une nouvelle approche pour le modèle $M/G/\infty$ permettant de prendre en compte cette variation lors de la transformation des tailles de slots en paquets. Ainsi les volumes de données correspondants aux slots de temps sont regroupés en fonction de leurs tailles. Les distributions des tailles de paquets et les distributions des lois d'inter-arrivées sont estimées par groupe de slots. Les résultats obtenus avec cette approche sont bien meilleurs au niveau de la performance réseau. Cette approche est applicable à tous les modèles de trafic par agrégat d'octets (FGN, FARIMA, ...).

Le modèle de trafic agrégé développé dans cette partie de la thèse peut servir à modéliser tout type de trace IP homogène ou hétérogène.

1.5 Models de Trafics pour les Applications Multimédia

La modélisation des traces de trafics ou du trafic agrégé ne prend pas en compte le comportement de chaque source présente dans la trace. Toutefois il est plus judicieux parfois de modéliser le trafic par type d'application quand c'est possible. De tels models peuvent servir directement dans la génération de la matrice de trafic lors d'une simulation. Surtout que ce type de modèle peut être utilisé dans des études de caractérisation

sur le trafic résultant de la superposition d'applications homogènes. En effet, pour certains types d'applications la superposition d'un nombre important de sources de trafics peut conduire à des modèles de trafic plus simple et plus efficaces dans les simulations. Dans les sections suivantes nous présentons des modèles unitaires et agrégés par type d'applications : Audio, Vidéo et Données.

1.5.1 Applications Audio

Les applications audio sont caractérisées par une alternance entre des périodes de paroles et des périodes de silence. L'information audio est codée et encapsulée dans des paquets à taille fixe qui sont transmis à intervalles constants. L'efficacité de codage et la prise en compte des périodes de silence font que le débit d'une application audio peut varier entre 5 Kbps et 64 Kbps en fonction du codec utilisé.

1.5.1.1 Modèle

Une application audio peut être modélisé par une processus markovien à deux états ON et OFF avec un taux d'émission de paquets constant $\lambda = \frac{1}{T}$ durant la période ON (T est l'inter-arrivée de paquet). Les mesures montrent que le taux de transition entre les deux états est distribué exponentiellement. (Voir Figure **5.3**)

1.5.1.2 Caractérisation du Trafic Audio et Modèles Agrégés

Les applications audio se caractérisent par une distribution de taille de paquet constante selon le type du codec et une inter-arrivée constante durant la période ON. Le processus d'arrivée de paquets peut se calculer à partir du modèle markovien ON-OFF présenté précédemment. Soit T l'inter-arrivée fixe durant la période ON et soit T_{ON} (T_{OFF} resp) la moyenne de la durée de la période ON (OFF resp). Deux inter-arrivées sont possibles dans cette configuration : une inter-arrivée T avec une probabilité $p = \frac{n-1}{n}$ (n est le nombre de paquets générés durant une phase ON + OFF) et une inter-arrivée $T + T_{OFF}$ avec une probabilité $1 - p$. Ainsi la fonction d distribution cumulée (CDF) des inter-arrivées de paquets peut s'écrire :

$$1 - F(x) = \begin{cases} 1 & 0 \leq x \leq T \\ (1-p)e^{-\frac{x-T}{T_{OFF}}} & x \geq T \end{cases} \tag{1.7}$$

On s'intéresse à cette même distribution dans le cas de la superposition de N application homogènes. La fonction CDF des inter-arrivées de paquets dans ce cas prend la forme:

$$1 - F_N\left(\frac{x}{N}\right) = \begin{cases} (1 - \lambda\frac{x}{N})^{N-1} & 0 \leq x \leq T \\ \frac{(1-p)^N e^{-\frac{x-NT}{T_{OFF}}}}{\left(\frac{T}{T_{OFF}}+1-p\right)^{N-1}} & x \geq T \end{cases} \tag{1.8}$$

On voit bien que lorsque $N \to \infty$ nous avons $1 - F_N\left(\frac{x}{N}\right) \to e^{-\lambda x}$. Ceci revient à dire que l'on peut statistiquement remplacer la superposition de N application homogènes par une loi exponentielle équivalente quand N tends vers l'infini. Nous avons vérifié que pratiquement cette propriété et valable pour un nombre N limité de sources simulés. Ainsi pour $N = 40$ nous avons pu constater que la distribution équivalente tend vers une loi exponentielle.

En utilisant cette propriété nous proposons un modèle agrégé qui remplace la superposition de N différent types d'applications audio. Ainsi pour un nombre $n_i,\ i = 1...N$ d'applications par type de codec i et $\lambda_{ia,i},\ i = 1..N$ débit d'inter-arrivée moyen par type de codec i, nous générons une loi exponentielle équivalente avec un débit Λ_{ia} donné par:

$$\Lambda_{ia} = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\lambda_{ia,i}}{n_i} \tag{1.9}$$

Les paquets sont générés selon une distribution discrète Ps de taille de paquets donnée par :

$$\begin{aligned} Ps &= \{Ps_i,\ i = 1...N\} \\ \Pr(Ps = Ps_i) &= \frac{n_i \cdot \lambda_{ia,i}}{\sum_{i=1}^{N} n_i \cdot \lambda_{ia,i}} \end{aligned} \tag{1.10}$$

1.5.1.3 Limitation de Performance de L'approximation Exponentielle

Bien que le modèle agrégé exponentiel pour la superposition d'application audio soit valide statistiquement, il faut valider la performance du modèle en environnement réseaux. Pour cela le trafic généré par le modèle agrégé ainsi que le trafic superposé sont injectés dans une file d'attente de type $G/D/1/K$ représentant une interface de routeur à débit constant. La performance du modèle et du trafic superposé est mesurée en fonction de la charge de la file.

Les résultats montrent que le modèle agrégé a une performance très proche du trafic superposé pour les charges petites et moyennes. Toutefois, il y a une déviation importante entre le modèle et le trafic superposé à partir d'un certain seuil de charge. Cette déviation s'explique en comparant le coefficient de covariance cumulée du trafic (ou l'indice de dispersion des inter-arrivées : IDI) dans les deux cas comme c'est montré sur la Figure **5.8**. En effet, le trafic exponentiel ne présente pas de covariance cumulée, par contre le trafic superposé montre une augmentation nette de la covariance cumulée. Ceci explique la dégradation en performance du trafic réel par rapport au trafic exponentiel équivalent.

Pour remédier à ce problème nous proposons un modèle agrégé pour le trafic audio en cas de forte charge basé sur l'approximation d'un processus MMPP-N (Markov Modulated Poisson Process - N states) par un processus MMPP-2. L'idée est de diviser l'espace d'état de dimension N qui représente N sources ON-OFF en deux et utiliser un processus MMPP-2 équivalent à deux états seulement. L'estimation des états équivalents est faite par équivalence sur l'indice de dispersion des comptes (IDC) des sources.

La validation du modèle MMPP-2 équivalent montre que la covariance cumulée générée par le processus équivalent suit bien celle du trafic réel. Par conséquent les

performances en simulation sont plus proches de la réalité en cas de fortes charges (voir Figure **5.10**).

Il faut noter finalement que l'approximation exponentielle de trafic est très importante car elle offre une solution analytique pour les simulations de réseaux à grande échelle. Cette propriété n'est pas toujours valable avec d'autres modèles en réseaux de files d'attentes, d'où l'importance de cette approximation.

1.5.2 Applications Vidéo

Le trafic vidéo est issu de différents codecs et présente des caractéristiques très différentes en fonction de la séquence vidéo codée. Ainsi on ne peut pas associer un modèle à un codec vidéo. La majorité du trafic vidéo est codé par l'algorithme de codage MPEG qui a plusieurs versions MPEG1, MPEG2, MPEG4 et MPEG7. Le codage MPEG repose sur la notion du GoP (Groupe of Pictures), ainsi un groupe d'image utilisant les redondances spatiales et temporelles qui existent dans les séquences vidéo est généré toute les 12/25 sec. Cette propriété sera utilisée pour modéliser le trafic vidéo en utilisant les GoPs comme entités de trafic à modéliser.

1.5.2.1 Modèles

Le modèle sélectionné pour modéliser la vidéo c'est le modèle $M/G/\infty$ que nous avons exposé lors de la modélisation du trafic IP. La notion du slot utilisé dans le trafic IP est remplacée par le GoP. En effet, le trafic vidéo est divisé naturellement en slots de temps à cause de la génération en GoP de durée constante. Toutefois une différence majeure avec le cas du trafic IP réside dans la taille constante des paquets et la génération régulière des paquets à l'intérieur des GoPs. Ce qui simplifie le processus de modélisation par rapport au trafic IP.

1.5.2.2 Caractérisation du Trafic Vidéo et Modèles Agrégés

Nous utilisons des traces de films vidéo pour tester le modèle vidéo proposé. Une estimation de la distribution des tailles de GoPs ainsi que la structure de corrélation entre les tailles des GoPs est effectuée de la même façon que ce qui est fait pour le trafic IP. Le trafic généré par le modèle présente les mêmes caractéristiques que le trafic réel et sa performance est très proche dans une simulation en environnement réseaux. La bonne performance du modèle est due principalement à la régularité des tailles de paquets et leurs dates d'inter-arrivées dans les GoPs (ce qui n'est pas le cas pour le trafic IP).

Le profile du trafic résultant de la superposition de N sources vidéo est aussi étudiée. Le but est de vérifier si l'hypothèse exponentielle peut être considérée aussi dans ce cas. Les résultats montrent que le comportement du trafic généré est très variable en fonction du codec MPEG1, MPEG2 ou MPEG4 considéré. Les corrélations persistent avec un nombre plus important de sources selon le codec, mais finissent par converger vers une loi exponentielle. Ce résultat est valable pour les petites et moyennes charges de trafic. Malheureusement le phénomène de covariance cumulée dégrade la performance du modèle

sous fortes charges. Une modélisation du trafic vidéo agrégé est alors possible en utilisant le modèle $M/G/\infty$ sur la trace résultante. Ceci revient à la même méthodologie utilisée pour le trafic IP en général. Malheureusement, cette solution limite la modélisation analytique du trafic vidéo sous fortes charges.

1.5.3 Applications Data

Les applications data concernent généralement le transfert de fichiers. Les fichiers peuvent être de simples pages web, des mails électroniques (email), ou de grands fichiers (par exemple FTP), etc. Le point commun entre toutes ces applications est que le transfert n'a pas généralement de contraintes type temps réel (par exemple délai ou gigue) mais plutôt des contraintes de fiabilité (aucune pertes) et du débit nominal assuré. Etant donné que les réseaux IP sont des réseaux sans connexion avec aucune garantie de réception ni d'ordre de livraison. La fonction de fiabilité est généralement assurée avec des protocoles applicatifs : principalement TCP. Toutefois le protocole TCP est un protocole avec contrôle en boucle fermée et par conséquent le profile de génération de paquet ne peut pas être déterminé à la source car il dépend de la réponse du réseau. Cette propriété rend la modélisation des applications data plus sophistiquée car elle nécessite la reproduction du comportement du protocole TCP en fonction du réseau (ou du processus de pertes de paquets).

1.5.3.1 Modèles

En plus du protocole de transport TCP, les applications data partage un comportement similaire basé une alternance de périodes d'activité (transfert de fichiers) et de périodes d'inactivité (interprétation ou lecture par l'utilisateur). Ces périodes sont dénommées généralement ON et OFF respectivement. L'étude des applications data revient à déterminer les paramètres des périodes ON et OFF correspondants à l'application. La génération des paquets se fait généralement par le protocole TCP.

1.5.3.2 Caractérisation et Modèles Agrégés

Afin de déduire des modèles simples pour le trafic data agrégé (résultant de la superposition des application data) il est important de caractériser le trafic agrégé en fonction des paramètres qui influent sur le contrôle en boucle fermé de TCP. Ceci concerne principalement le processus de pertes. Dans le cas du trafic UDP les caractéristiques du trafic agrégé restent stables en fonction des pertes (absence de boucle de retour, par exemple le trafic audio et vidéo). Par contre, nous avons observé que les propriétés statistiques du trafic TCP agrégé (surtout la corrélation à long terme et la sporadicité) sont variables en fonction du taux de perte.

L'observation précédente montre que l'utilisation de processus agrégé (par exemple$M/G/\infty$) n'est pas adéquate dans le cas du trafic TCP. Deux solutions pour remédier à ce problème ont été adoptées. Premièrement : simplifier la superposition des applications data tout en conservant la simulation du protocole TCP. Deuxièmement: proposer un modèle

plus efficace pour la simulation de TCP qui permet de conserver sa dynamique en boucle fermée tout en étant plus adapté aux simulations dans les réseaux à grande échelle. C'est le modèle différentiel de TCP que nous allons exposé dans le paragraphe suivant. Mais d'abord nous allons présenter la première solution qui vise à simplifier le processus agrégé.

Modèle ON-OFF équivalent

Le modèle ON-OFF équivalent est basé sur l'agrégation de périodes d'inactivité OFF (souvent trop longues) dans les applications individuelles par des périodes OFF plus courtes. Ceci permet de diminuer le nombre de processus à superposer considérablement, tout en conservant les propriétés statistiques du trafic résultant.

Le principe est très simple : On remplace la superposition de N processus ON-OFF simples avec $T_{OFF} \gg T_{ON}$ par M processus ON-OFF équivalent sous les conditions suivantes :

- Le débit résultant dans les deux cas doit être équivalent.
- Le nombre de périodes active ON en moyenne est équivalent dans les deux cas.

On rappelle que les périodes ON ne sont pas agrégés. Ceci permet de calibrer le nombre de périodes ON actives en moyenne en fonction de la durée de la période OFF et du nombre de processus superposés. On voit logiquement que plus T_{OFF} est petit moins le nombre de processus nécessaire est important d'où le gain avec le processus équivalent. On rappelle que la moyenne des périodes ON actives lors de la superposition de N processus ON-OFF est donnée par (moyenne d'une loi binomiale) :

$$N * P_{ON} = N * \frac{T_{ON}}{T_{ON} + T_{OFF}} \tag{1.11}$$

La validation de ce modèle montre que les propriétés statistiques du trafic agrégé sont bien respectées. Toutefois ce modèle reste limité quand il s'agit d'un nombre très élevé de processus à superposer. Le processus équivalent permet de simplifier l'exécution mais le nombre de processus équivalents nécessaires reste aussi élevé. Le modèle différentiel de TCP que nous allons proposer dans le paragraphe suivant offre une solution meilleure et plus efficace à ce problème.

1.6 Modélisation Analytique Différentielle de TCP/IP

Internet est basé sur le protocole IP. Un protocole sans connexion, très simple qui permet une communication sans garantie de réception ni d'ordre de livraison. TCP/IP est le protocole le plus utilisé sur Internet qui permet de fiabiliser les communications en offrant un contrôle en boucle fermé sur la transmission des paquets. Ainsi tout paquet perdu sera retransmis et les paquets sont livrés dans le bon ordre de transmission. En effet TCP/IP assure plusieurs fonctions :

- Contrôle flux de bout en bout.

- Contrôle d'erreur.
- Contrôle de congestion

Le contrôle de flux de bout en bout garanti que la source n'émet pas plus de paquets que la destination peut en recevoir. Ceci est assuré par l'annonce de la valeur de tampon (buffer) du côté du récepteur avant toute communication.

Le contrôle d'erreur assure la fiabilité de transmission par un système d'acquittement pour les paquets bien reçus. Un paquet non acquitté ou mal acquitté (triple ACK) est synonyme d'erreur de transmission qui déclenche un mode de retransmission pour les paquets perdus.

Enfin le contrôle de congestion permet à la source de réduire son débit afin d'éviter la congestion. Ceci est effectué par une fenêtre glissante qui désigne à chaque instant le nombre de paquet que peut émettre la source et qu'on appelle : fenêtre de congestion CWND. Pratiquement la source possède une valeur de crédit qui évolue en fonction de cette fenêtre.

Ces fonctions assurent une utilisation optimisée de la bande passante et un bon partage entre plusieurs flux qui transitent sur un même chemin tout en assurant une transmission fiable des données.

1.6.1 Description de TCP/IP

Pour assurer les fonctions décrites précédemment. TCP/IP utilise trois modes de transmission :

- Slow Start : Dans ce mode TCP/IP découvre la bande passante disponible en augmentant sa fenêtre de congestion CWND exponentiellement (et par conséquent son crédit d'émission) jusqu'à qu'une congestion ait lieu (une erreur ou une perte).
- Fast Retransmit : Dans ce mode TCP/IP retransmet les paquet perdus lors de la congestion un par un.
- Congestion Avoidance : Dans ce mode TCP/IP stabilise le taux d'émission tout en essayant de découvrir plus lentement la bande passant disponible.

La machine d'état simplifié de TCP/IP est présentée sur la Figure **6.2**.

1.6.2 Modèle différentiel

Le principe de la modélisation différentielle est de décrire l'évolution du débit fluide d'une source TCP/IP en fonction du temps pour chaque mode d'opération, ainsi que l'évolution des paramètres utilisés par l'algorithme TCP/IP avec des équations différentielles. Nous supposons que chaque valeur décrite par une fonction f vérifie l'équation suivante :

$$f(t + \Delta t) = f(t) + \dot{f}(t) * \Delta t \tag{1.12}$$

Avec:

$$\dot{f}(t) = K, \forall t \in [t, t + \Delta t] \tag{1.13}$$

Dans chaque nœud le débit fluide est transformé en nombre de paquets équivalent. La décision de l'occurrence d'une perte est prise au niveau des nœuds en fonction des tailles de buffers, ensuite les délais nécessaires à la source pour se rendre compte de la congestion sont calculés analytiquement.

Le débit d'une source TCP peut être calculé approximativement à chaque instant par le rapport entre la taille de la fenêtre de congestion et le temps aller-retour noté généralement RTT (Round Trip Time).

$$\lambda = \frac{CWND(t)}{RTT(t)} \tag{1.14}$$

Le modèle différentiel fournie une expression analytique différentielle de la valeur de $CWND(t)$ et $RTT(t)$ pour chaque mode d'opération de TCP/IP. Le passage d'une équation différentielle à une autre se fait par des événements de contrôle qui sont liés à la détection d'une perte lors d'une transmission. L'ensemble de ces équations est résumé dans le Tableau **6.1**.

Il est important de noter que le modèle a été élaboré dans le cas mono-source dans un premier temps. Ensuite l'extension du modèle dans le cas multi-source a été fait avec une séparation entre les différent flux qui sont propagés. Toutefois, pour détecter les pertes dans un nœud un seul débit global peut être considéré pour évaluer si le nombre de paquets dépasse la capacité du buffer. De plus à la sortie de ce nœud le débit doit être partagé entre les différents flux et les pertes doivent être affectées. Un système de partage a été mise en place en se basant sur le débit instantané de chaque source et la charge résiduelle dans le buffer. L'ensemble de ces équations est présenté dans le Tableau **6.1**.

1.6.3 Validation en mono et multi-sources

Nous avons testé le modèle différentiel de TCP/IP en mode mono-source pour valider le comportement transitoire du modèle. Les tests montrent que le modèle différentiel reproduit convenablement les modes d'opération de TCP. De plus le débit moyen et le nombre de pertes sont très proche du modèle événementiel utilisé comme modèle de référence. Une légère déviation entre le nombre de paquets perdus en modèle différentiel est tout de même à remarquer. Ceci est dû principalement à la propagation fluide des débits alors que les pertes sont détectées sur un nombre de paquets discret dans les buffers.

Le modèle multi-source offre lui aussi une bonne performance globale. Toutefois, le taux de pertes relatif est plus important en présence de plusieurs flux, tout en respectant un débit moyen très comparable au modèle événementiel. Le modèle différentiel de TCP/IP offre une bonne alternative plus adaptée aux réseaux à très grande échelle que le modèle événementiel souvent utilisé dans les simulator réseaux.

1.7 La Qualité de Service au Niveau Session avec SIP

La Qualité de Service des applications multimédia peut être améliorée en utilisant des techniques au niveau application. Les modèles de trafic que nous avons présenté jusqu'à maintenant servent à des évaluer la QdS au niveau paquets. Pour évaluer la QdS au niveau application (ou session) nous utilisons la notion de la bande passante équivalente. Cette notion permet d'utiliser les techniques anciennes des réseaux à commutation de circuits (réseaux téléphoniques) avec les applications multimédia. Toutefois, il faut un cadre qui permet de contrôler une activité utilisateur entre deux points. Heureusement, ceci est désormais possible grâce à l'introduction d'un nouveau protocole de signalisation: SIP (Session Initiation Protocol). L'architecture de SIP permet d'établir une session multimédia entre deux utilisateurs à travers un support à commutation de paquets. Nous utilisons une architecture étendue de SIP qui permet, outre que la signalisation du début et de la fin d'une session, à allouer la bande passante équivalente ou ordonnancer les sessions TCP pour mieux utiliser la bande passante disponible.

1.7.1 Architecture SIP et DiffServ

L'architecture du protocole SIP distingue entre les clients et les serveurs. Le serveur Proxy plus particulièrement joue un rôle important dans l'échange des messages utilisés pour établir une session entre deux utilisateurs. L'architecture étendue de SIP utilise le serveur Proxy comme un serveur d'ingénierie de trafic en plus de ses fonctions de signalisation. Ainsi le serveur Proxy permet de négocier les paramètres de QdS d'une session avant de l'établir. Dans un environnement DiffServ, ce rôle peut inclure l'acheminement des demandes d'allocation de bande passante via le protocole COPS ou l'affectation dynamique des classes de services (Voir Figure **7.3**).

Nous proposons d'implémenter ces fonctions au niveau du serveur Proxy pour contrôler les sessions multimédia au niveau application. Toutefois, ces techniques nécessitent l'élaboration de méthodes qui permettent de calculer la bande passante équivalente des flux multimédia.

1.7.2 Bande Passante Equivalente

La notion de la bande passante équivalente permet de traiter des flux multimédia comme des circuits. Le profile de génération de paquets à l'intérieur des flux ainsi que les caractéristiques du réseau sont prises en compte lors de ce calcul. En général on calcule la bande passante équivalent d'un flux multimédia en fonction de la loi d'arrivée des paquets constituant le flux (caractéristiques source), la loi de service des paquets (caractéristiques serveur), taille du buffer et le taux de pertes maximum autorisé sur les paquets.

Un flux régulier consomme moins de bande passante qu'un flux sporadique et sa bande passante équivalente est moins importante. De plus des buffers très grands permettent de mieux amortir les pics de trafic et la bande passante équivalente dans ce cas est plus

petite. Dans la littérature on trouve deux branches majeures pour le calcul de la bande passante équivalente. La bande passante par la formule de Kelly [Kel96] et la bande passante par les formules analytiques. Nous avons choisi la deuxième méthode car elle permet un calcul temps réel de la bande passante équivalent et peut être utilisé dans le cadre d'un environnement contrôlé par SIP. Nous proposons de calculer la bande passante équivalente par un système de type $G/D/1/K$.

1.7.3 Ordonnancement des Sessions

Le but de l'ordonnancement des sessions est de trouver une solution à un problème connu des sessions TCP. En effet il a été montré que les connexions TCP courtes souffrent énormément des pertes en présence des connexions TCP longues de type FTP par exemple. L'interaction entre les connexions TCP courtes et longues a un effet très négatif sur les courtes connexions. Il est préférable d'affecter les connexions courtes à une classe de service prioritaire par rapport aux longues connexions, de façon à ce qu'elles soient privilégiées. En réalité les connexions longues seront légèrement affectées par cette priorité. Ceci se traduit par des délais négligeables sur la durée totale du transfert. Par contre ce mécanisme permet aux connexions courtes d'avoir moins d'erreurs de transmission et par conséquent être plus performantes. Les techniques traditionnelles pour mettre en œuvre ce mécanisme sont généralement basées sur une modification des entêtes de paquets TCP pour mesurer la durée d'une connexion. Ensuite on peut changer dynamiquement la classe de service d'une connexion lorsque sa durée dépasse un certain seuil.

En utilisant l'architecture SIP, nous pouvons utiliser des mesures sur la durée de la session entière et prendre les décisions au niveau application. En fait, ce genre de mesures de durée sont faites pour des fins de facturations et peuvent être utilisées pour l'ordonnancement dynamique des sessions. Deux types de mesures ont été testées : la durée d'une session et le volume de données échangés lors d'une session. Il s'avère que la quantité de données est un meilleur critère pour l'ordonnancement compte tenu du fait que les sessions TCP peuvent contenir de longues périodes d'inactivité (session Web par exemple).

1.7.4 Réservation de la Bande Passante

La technique présenté dans paragraphe précèdent sert uniquement à ordonnancer les sessions TCP. Toutefois, une meilleure isolation entre les flux peut améliorer la qualité de service sur tous les flux. C'est l'objective de la deuxième technique qui vise à réserver la bande passante équivalente aux flux multimédia les plus exigeants en terme de QdS à travers un système de files d'attente de type WFQ (Weighted Fair Queuing).

Le serveur Proxy chargé d'établir les sessions multimédia, négocie entre les deux parties de la communication les caractéristiques des sessions à établir. Cette description peut être échangée en utilisant le protocole SDP (Session Description Protocol), un protocole généralement associé au protocole SIP. En utilisant ces informations le serveur Proxy peut initier un calcul de la bande passante équivalent pour le flux multimédia correspondant. La valeur calculée sera utilisée pour déterminer les poids à donner au

système de files d'attentes WFQ. Le principe est de régler les poids pour réserver la bande passante équivalente nécessaire au flux prioritaire, alors que les autres flux (moins prioritaires) partagent la bande passante résiduelle.

La réservation de la bande passante équivalente montre des résultats plus performants que l'ordonnancement des sessions mais au prix d'un nombre plus important de paramètres à échanger lors de l'initiation des sessions. Il faut noter que les deux mécanismes peuvent coopérer. Ainsi on réserve la bande passante équivalente pour les sessions avec les paramètres négociés. Toutefois, on peut facilement déclasser une session qui ne respecte pas les valeurs déclarées pour le calcul de la bande passante équivalente en utilisant l'algorithme d'ordonnancement. Finalement le calcul de la bande passante équivalente peut être utilisé aussi pour mettre en œuvre un mécanisme de contrôle d'accès pour ne pas surcharger le serveur Proxy.

1.8 Conclusion

Ces travaux concernent la modélisation et l'analyse de performances du trafic et des applications multimédia dans les réseaux hétérogènes. Le trafic IP agrégé et les applications audio, vidéo et données sont étudiés. Cette étude nous conduit à proposer un modèle générique et hiérarchique pour la représentation des sources de trafics multimédia qui permet de décrire les applications multimédia d'une façon simple, précise et générique. Le modèle générique est implémenté et constitue le noyau d'un outil de modélisation et simulation des sources de trafics. Une caractérisation du trafic IP issu d'applications multimédia est conduite en utilisant les modèles développés avec cet outil. Particulièrement, la problématique de la modélisation des sources de trafics agrégées est adressée, et des modèles agrégés simples sont déduites pour la superposition des sources de trafics audio, vidéo et données. Le trafic agrégé de type TCP présente des propriétés statistiques variables en fonction du taux de pertes de paquets sur le réseau à cause du contrôle en boucle fermée imposé par TCP. Un nouveau modèle analytique du protocole TCP basé sur la théorie différentielle du trafic est ensuite proposé. Ce modèle permet une représentation fiable du trafic TCP tout en étant très performant sur les réseaux à grande échelle. Finalement, une extension de l'architecture du protocole SIP est présentée afin de permettre une gestion de la qualité d service au niveau session. Les mécanismes proposés reposent sur l'ordonnancement des sessions et l'allocation de la bande passante par des approches d'évaluation de bande passante équivalente. Cette dernière technique rend possible l'utilisation des formules d'Erlang dans les réseaux à commutation de paquets.

Part II

Multimedia Traffic Modelling and Performance Analysis in Hetergeneous Networks

Chapter 1

Introduction

Since two decades numeric data transport technologies know a constant evolution. Internet and its front end interface (the Web) offered new horizons for the deployment of new services. Hence, with only voice services offered by circuit switching networks (e.g Public Switched Telephone Network (PSTN) and Global System for Mobile Communications (GSM)), services progressed rapidly towards multimedia applications where voice, video, and data are used by most of the modern applications. Thus, applications like visiophony, videoconferencing and instant messaging are deployed on wireline and wireless networks interchangeably. This flexibility is primarily due to the evolution from circuit switching to packet switching networks. Indeed, application content is encapsulated into packets and transported on different networks. Communication protocols and application layer take in charge the reliable transport and reassembly of packets so that users can use their multimedia applications just like traditional telephones.

The convergence of telecommunication networks towards a packet-switched core with multiple access networks (wireline and wireless) has big advantages for both end users and telecommunication operators. Users by now can use different technologies to access their information via the same applications whether they are using their work laptops, home PCs, personal mobile phones... On the other hand, telecommunication operators deal with all multimedia applications as flows of packets transiting via the same core network without the need to host different signalling systems and different switching technologies. However, multimedia applications are very different in nature. Some applications like voice and video have strict real time constraints due to signal reconstitution problems. Thus, a voice packet arriving very late could not be used while a high loss rate on video packets may result in a very bad video quality. Indeed, the notion of quality of service (QoS) has emerged with the evolution towards packet switched networks as a universal core network. Basically, sophisticated multimedia services can be deployed on packet switched networks, if operators can fulfil QoS constraints of the deployed multimedia services. This is normally referred to as Service Level Agreement (SLA).

Telecommunication operators need to evaluate the performance of new multimedia services by conducting performance evaluation studies on their networks. In this context, stochastic modelling and simulation techniques are suitable to provide telecommunication operators with reliable tools to evaluate the performance of their services and networks.

However, reliable performance evaluation studies require reliable multimedia traffic models. Besides, large scale networks require also efficient simulation techniques, in order to achieve performance evaluation studies in reasonable time. Reliability and efficiency of multimedia traffic modelling is confronted primarily with the diversity of multimedia applications and their complexity. Traffic profile (or packets generation profile) is tightly linked with the user behaviour, transport technology and transport protocols. A reliable multimedia application model must consider all of these parameters.

Traffic and multimedia application modelling issues are essential topics that we will handle throughout this thesis. In order to deal with traffic modelling methodically, we suggest a generic hierarchical framework allowing simple and generic modelling of multimedia applications regardless their complexity. This framework is designed and implemented to offer maximum flexibility in describing most of multimedia applications. The framework represents the main part of TSM (Traffic Source Modeler) a powerful traffic modelling and performance analysis tool. This tool works in tandem with the DHS (Distributed Hybrid Simulator) conceived and implemented by LAAS-CNRS. Using the generic framework, audio, video and data application models are implemented and evaluated. However, some applications have complex behaviours resulting in sophisticated application models. Indeed, the detailed description of application behaviours is necessary when evaluating the performance of one single application. Meanwhile, when superposing large number of applications, the specific behaviour of one application is not important. Moreover, the complexity of the application model may result in long simulation times while not providing significant additional accuracy on simulation results. In this thesis we study the superposition of multimedia application traffic and suggest simple aggregated models. Hence, we provide simple traffic models that may substitute a large number of homogeneous multimedia applications. However, no simple aggregate traffic models do exist for all application types, especially when packets are transported by Transmission Control Protocol (TCP). The closed-loop nature of this protocol induces a dynamic packet generation profile, and by consequence a dynamic correlation structure because of retransmission mechanisms of TCP. Reliable models of TCP applications require the simulation of TCP algorithm. This may be problematic when large number of TCP based applications are evaluated in the context of large scale network. Therefore, we suggest a new simulation technique for TCP using differential traffic theory [GGB+01]. The goal is to preserve the transient behaviour of TCP while reducing simulation complexity by propagating rates rather than packets. We achieve this by introducing a differential analytical model for TCP. The suggested model executes faster than traditional event-driven implementations of TCP while respecting the transient behaviour of this protocol.

The different techniques used for traffic modelling and simulation are used primarily in performance evaluation studies on packet switching networks. Planning decisions can be taken based on these studies. However, multimedia application traffic can be handled at the session level to achieve planning decisions. In this way, old powerful techniques from circuit switching domain can be used with new packet switching flows. This is motivated primarily by the introduction of Session Initiation Protocol (SIP) which is a

session set up and release protocol adopted by the 3rd Generation Partnership Project (3GPP) for mobile networks. In this thesis, we suggest a session management framework based on SIP in order to deal with packet flows as circuits. Session scheduling mechanisms and equivalent bandwidth estimation techniques are used to enhance QoS on multimedia flows.

The thesis is organized as follows:

In **Chapter 2**, we give an overview of Heterogeneous and Multiservice Networks and their QoS mechanisms. This chapter provides a general review of wireless access networks with their predefined QoS classes as well as QoS mechanisms implemented in IP multiservice networks.

Chapter 3 is devoted to describe a generic hierarchical framework for multimedia traffic modelling along with statistical analysis and estimation tools. It constitutes the functional description of the traffic modelling tool developed during this thesis. A more detailed description of the modelling tools is provided in **Appendix A**. The material of this Chapter is published in [HGCB06].

Chapter 4 handles IP traffic modelling issue using time slot (or window based) technique with generalized $M/G/\infty$ input process model. Limitations on time slot technique are showed out while justifications and solutions to enhance the performance are provided. Our results apply to other window based traffic models. Results of this Chapter are published in [HGB06d, HGB06c].

Chapter 5 addresses multimedia application modelling techniques, where audio, video and data applications are studied and aggregated models for the superposition of homogeneous applications are derived. It provides also a characterization study of aggregated UDP and TCP traffic. In fact, the results we obtain on TCP traffic justify the necessity of a new efficient TCP traffic model. Our results are published in [HGB06a, HG06, HG05].

Chapter 6 introduces a differential analytical model of TCP/IP, mixing differential equations with control events. The aim of this model is to provide efficient simulation of TCP/IP for large scale network simulations by propagating fluid rates rather than data packets. Although, the suggested model concerns TCP/IP, its concept can be generalized to any closed-loop and reactive control mechanism. It shows specifically the importance of differential modelling and simulation techniques for reactive systems. The material of this Chapter is published in [HGB06b].

Finally, in **Chapter 7** we extend the SIP signalling architecture to achieve session based QoS via session scheduling and bandwidth allocation on a per-flow basis. Session or flow level QoS mechanisms using SIP are very promising as SIP is adopted for next generation mobile networks and will be dominant in future networks. Results of this Chapter are published in [HGB06f, HGB06e].

We conclude this manuscript with general remarks and future work perspectives.

In **Appendix A**, we present in more details the traffic modelling tool (TSM) implemented arround the generic framework, and in **Appendix B** we provide a complete list of estimated video traffic models.

Chapter 2

Heterogeneous Multiservice Networks

2.1 Introduction

The evolution of telecommunication networks during last decade changed the way services are perceived by end users. While telephony networks offered only voice services, new multimedia services are deployed extensively on packet networks. In particular, the Internet and the World Wide Web (WWW) introduced the notion of multiservice networks where voice, video and data are transported altogether on the same network and are accessed almost by the same way. This is achieved due to innovative Quality of Service (QoS) management and control mechanisms which are deployed over connectionless best-effort IP networks.

Recently, wireless connectivity made a revolution in telecommunication networks. Users by now can access multimedia applications via wireless terminals, although wireless medium has big limitations compared with the wireline medium. This diversity of access networks formed a heterogeneous environment where different technologies are interconnected by the same core IP network. A heterogeneous network [WHM01] could be seen as a common core network that deals with all network functionalities and operates as a single network. Thus, radio access networks handle functions related to radio access technology only (mobility management, user authentication, ...). However, all services transit by the same core IP network whatever their origin or destination. Future networks will deliver services that can operate between mobile, wireless, broadband, or circuit-switched access technologies. This involves supporting new signalling protocols such as Session Initiation Protocol (SIP) and next-generation architectures such as IP Multimedia Subsystems (IMS). One of the major obstacles for heterogeneous networks is the convergence of different access networks, which requires a standardization effort and business commitment to support it.

Heterogeneous multiservice networks will be dominant in the future. Different access networks will cooperate to provide different types of services. Of course technological challenges are numerous but great steps have already been done towards this ambition.

In this chapter, we present an overview of heterogeneous multiservice networks. We focus on new access networks especially wireless technology (Mobile networks and Wireless LANs). Besides we present the QoS management mechanisms used to fulfil Service Level Agreement (SLA) on best effort IP networks (i.e. Internet). It is important to note that satellite communications are a fundamental part of this multiservice heterogeneous environment. However, satellite networks present different constraints from terrestrial networks especially due to the high latency over communication links. Satellite communications will not be addressed in this thesis.

2.2 Access Networks

Access networks are the first contact of customers with telecommunication networks. Although traditional access networks knew big enhancements towards higher bit rates, the main revolution in access networks last years was brought by the wireless access, making of wireless communications a style of life.

While services offered by the GSM (Global System for Mobile communications) were focused on the circuit-switching domain with voice applications mainly, the evolution towards GPRS (General Packet Radio Service), EDGE (Enhanced Data rates for GSM Evolution), UMTS (Universal Mobile Telecommunication System) and more recently IMS (IP Multimedia Subsystem) offer a wide range of multimedia applications via new packet-switching architectures. Particularly, the third generation of mobile networks 3G (UMTS) passed by several releases. While the R4 release of UMTS concentrates its innovation in the radio side with the incorporation of Wideband Code Division Multiple Access (WCDMA), other releases of UMTS R5 and R6 prepared the core network for the transition to the 'All IP' networks. In 'All IP' networks all communications pass by an IP core network where old and new QoS mechanisms guarantee SLA for real time applications.

However, expanding transmission rates requires continuous innovation in the radio domain. Radio Technology promises up to 20 Mbps with High-Speed Packet Downlink Access (HSPDA) and intelligent or Adaptive Antennas (AA). Indeed, UMTS aims to offer end-to-end IP transport for both Radio and Core network. Moreover transmission rates will be pushed up to 100 Mbps with new radio technologies. Therefore, it seems reasonable to think that UMTS and its seamless complementary access technologies (e.g. WLAN and Bluetooth) will serve as the broad band platform for future mobile network evolution.

The wireless data technology can be summarized as follows [PMKN04]:

- 14.4 kilo bits per second (kbps) allows GSM data calls with a rate of 14.4 kbps per time slot, resulting in a 50% higher data throughput compared to the current maximum speed of 9.6 kbps.

- High Speed Circuit Switched Data (HSCSD) aggregates symmetrically or asymmetrically several circuit channels, e.g. 28.8 kbps for two time slots (2 + 2) or 43.2 kbps for three time slots (3 + 1).

- GPRS enables GSM with Internet access at high spectrum efficiency by sharing time slots between different users. It affords data rates of over 100 Kbps to a single user while offering direct IP connectivity.
- EDGE modifies the radio link modulation scheme from Gaussian Minimum Shift Keying (GMSK) to Phase Shift Keying with 8 phases (8-PSK), thereby increasing by three times the GSM throughput using the same bandwidth. EDGE in combination with GPRS (E-GPRS) may deliver single-user data rates of over 300 kbps.
- UMTS as Third Generation (3G) wireless technology utilizes a wide-band Code Division Multiple Access (CDMA) or Time Division CDMA (TD-CDMA) transceiver. Starting with channel bandwidths of 5 MHz it will offer data rates up to 2 Mbps.

In the following sections we give an overview of wireless access networks, besides an overview of the wireless local area networks (WLAN).

2.2.1 GSM

GSM is one of the most successful mobile telecommunication networks around the world. Although Data applications have been included in the GSM design, operators were not interested in these applications in the early days. Data applications must provide reliable connectivity with a guaranteed throughput for each connection. Deploying such services, like email and web browsing, could be expensive in the circuit switching architecture of GSM for both the user and the network operator. Figure **2.1** depicts the GSM public land mobile network (PLMN) architecture [BVE99]. MS denotes a GSM mobile station. The Base Transceiver Station (BTS) area of coverage forms what is called a cell: the basic unit of a cellular system. The control of several BTSs is achieved via the Base Station Controller (BSC) in a base station subsystem (BSS). Traffic produced by MSs in cells is routed through the mobile switching center (MSC), while a dedicated gateway mobile switching center (GMSC) handles connections towards the fixed networks (e.g., Public Switched Telephone Network - PSTN).

The home location register (HLR), the visited location register (VLR), the authentication centre (AUC), and the equipment identity register (EIR) are the used data bases for call control and network management activities.

Unfortunately, the circuit switching architecture of GSM is not compatible with multimedia and data applications. Indeed, the notion of QoS is neither invoked with the 2G mobile systems. However, the basic components of the GSM system are the main components of other mobile communication systems (GPRS, UMTS, ...). Meanwhile, new components were introduced to handle packet switching instead of circuit switching. The goal was to move from a circuit-switching domain to a packet switching domain. This was achieved by introducing some GPRS extensions at first, before moving to the 3G mobile network.

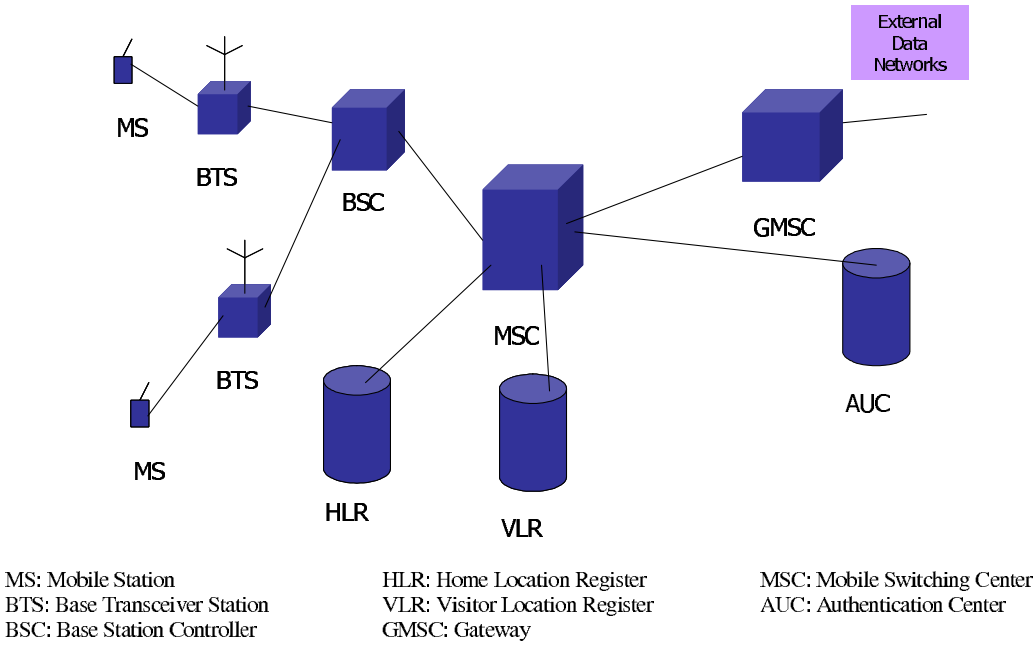


Figure 2.1: GSM Architecture

2.2.2 GPRS

GPRS was introduced into the GSM architecture by adding new classes of network nodes, called GPRS support nodes (GSN) [GSM95, BVE99]. In particular, two nodes were introduced to achieve the delivery of data packets between MSs and external packet data networks (PDN): a serving GPRS support node (SGSN) and a gateway GPRS support node (GGSN). While SGSN is responsible for the delivery of data packets within a service area, GGSN acts as an interface between the GPRS backbone network and the external PDNs.

Many tasks are carried out by SGSN including packet transfer and mobility management (attach/detach and location management). On the other hand, GGSN converts GPRS packets into the appropriate packet data protocol format and sends them out on the corresponding PDN.

There is an IP-based GPRS backbone network connecting GSNs. Indeed, packets are encapsulated and tunnelled using the GPRS Tunnelling Protocol (GTP). There are two kinds of GPRS backbones [BVE99]: Intra-PLMN backbone networks connect GSNs of the same PLMN and Inter-PLMN backbone networks connect GSNs of different PLMNs. Figure **2.2** depicts two intra-PLMN backbone networks of two PLMNs connected with an inter-PLMN backbone.

GPRS introduces the notion of Packet Data Protocol (PDP) context, which describes the characteristics of the session. It contains the PDP type (e.g., IPv4), the PDP address assigned to the mobile station, the requested QoS, and the address of a GGSN that serves as the access point to the PDN. Using the PDP context, the QoS requirements

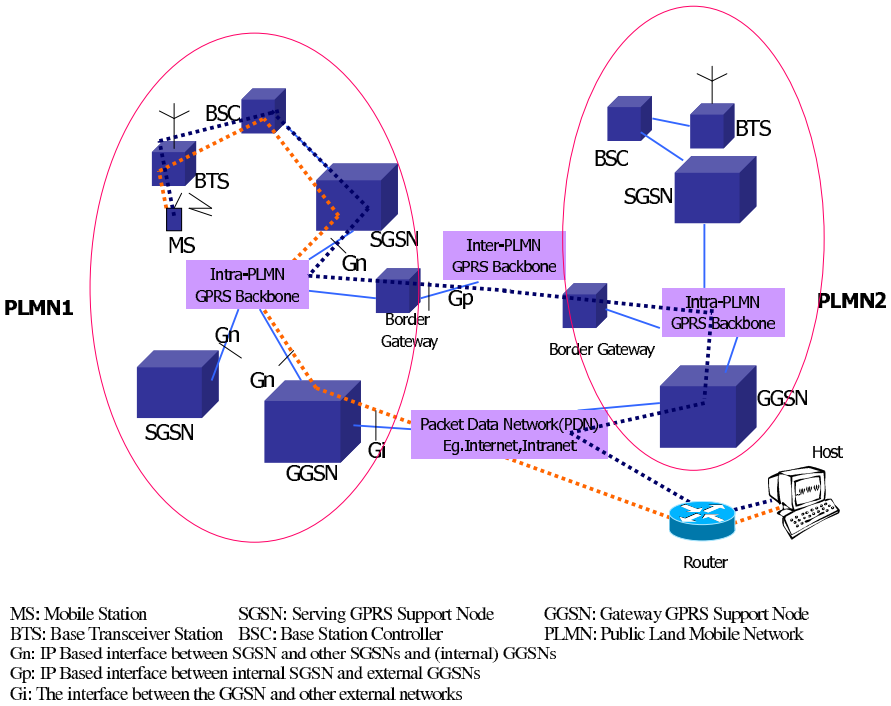


Figure 2.2: Inter and Intra PLMN Structure in GPRS Network

of multimedia applications (real time and non real time) could be specified. GPRS allows defining QoS profiles using the parameters service precedence, reliability, delay, and throughput [GSM98, BVE99].

- Three levels of priority: high, normal, and low are defined by the service precedence parameter. Those levels determine the precedence between services.
- The transmission characteristics needed by each application are defined by reliability parameter. GPRS defines three reliability classes for the maximum values for the probability of loss, duplication, mis-sequencing, and corruption of packets.
- The maximum mean delay of an end-to-end transfer time between two MSs is defined by the delay parameters. The delay includes all delays within the GPRS network.
- Finally, the throughput parameter specifies the maximum/peak bit rate and the mean bit rate.

Using QoS classes, QoS profiles for each session are negotiated, depending on the QoS demand and the current available resources. Generally, the billing of the service is based on the transmitted data volume, the type of service, and the chosen QoS profile. With this configuration, GPRS can be seen as a wireless extension of the Internet and the mobile user can access the Internet directly.

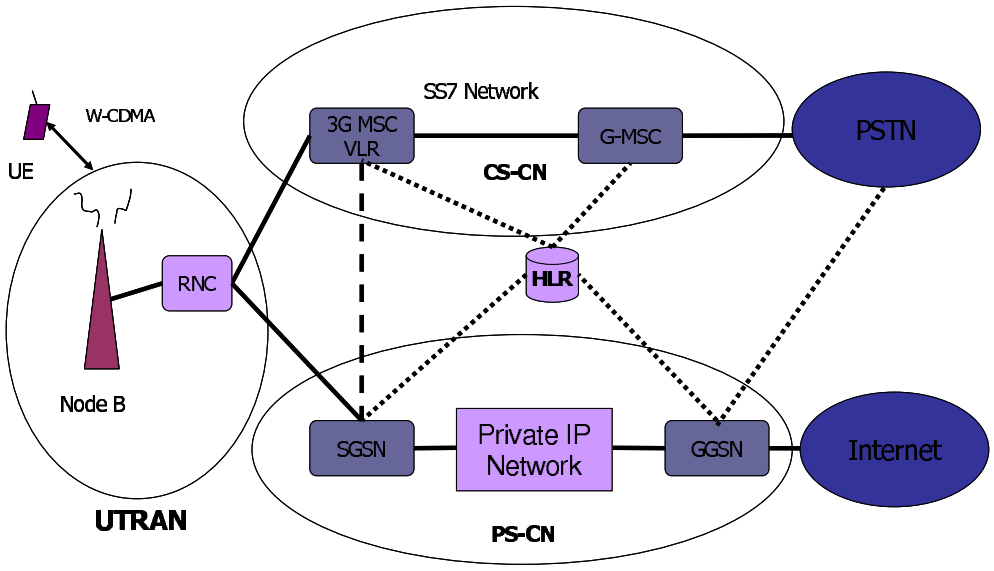


Figure 2.3: UMTS R4 Architecture

2.2.3 UMTS

The low communication bit rate deployed in GPRS made the best effort service the only service available to GPRS users, whereas many service classes could be defined. Indeed, theoretically GPRS would allow up to 128 kbps but in reality not more than 56 kbps is reached. The 3G mobile network (UMTS) gives the solution by offering higher bit rates reaching 2Mbps per user. Moreover, the UMTS architecture evolved to a well established PS (Packet Switching) domain. Although, the first 3GPP proposition for 3G mobile networks known as the R4 release was seeking compatibility between old GSM-GPRS architecture and the UMTS, it was quickly observed that keeping the old equipments with the new packet switching equipments is more expensive than moving to a unique core with both domains. Indeed, this was the major motivation behind the introduction of the IMS to support old and new communication services.

Thus UMTS is supposed to have three domains [BEF+02]:

- Circuit-switched domain (CS): Based on the GSM standard.
- Packed switched domain (PS): Defined in 3GPP standards.
- IP Multimedia subsystem (IMS): Introduced in the release 5 of 3GPP standards.

Figure **2.3** depicts the UMTS R4 architecture.

UE (User Equipment) in UMTS stands for MS in GSM, Node B stands for BTS while RNC (Radio Network Controller) stands for BSC. Finally, the Universal Terrestrial Radio Access Network (UTRAN) consists of the RNC, Node B and the air interface.

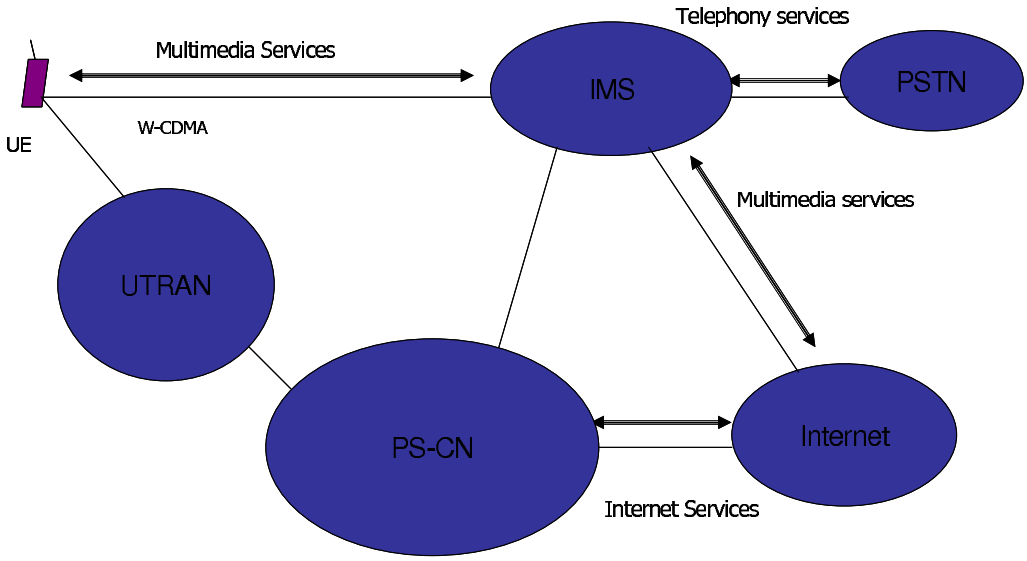


Figure 2.4: UMTS R5 Architecture

Besides the two core domains CS and PS the R4 conserves two signalling systems inside the same network: SS7 for legacy voice service (like GSM) and IP based signalling systems for new multimedia services. Fortunately, the R5 proposition for the UMTS system unifies the CS and PS domains and introduces the IMS. The architecture of R5 UMTS network is depicted on Figure **2.4**.

The UMTS QoS architecture is based on a layered bearer service architecture [BEF+02]. A Bearer Service (BS) is defined as a type of telecommunication service used for the transmission of signals between access points. Each BS provides individual services on a specific layer, using services offered by the layers below. A network BS describes how a given network provides QoS. Indeed, BS offers network QoS with clearly defined characteristics and functionality, from end-to-end per service.

Traffic classes or QoS classes of the UMTS system take into account the constraints imposed by the air interface. In order to offer robust QoS mechanisms UMTS introduces four QoS classes [BEF+02] as in Table **2.1**.

Table 2.1: QoS Classes in UMTS

QoS Class	**Delay**	**Application**
Conversational	<< 1 sec	Real time (telephony, video telephony)
Interactive	Around 1 sec	Real time (video conferencing, Telnet)
Streaming	< 10 sec	Traditional real time (video streaming)
Background	Can be >10 sec	Non real time (Data application)

The delay sensitivity is the major factor of service differentiation. It ranges from the very delay-sensitive conversational class to the delay-insensitive background class.

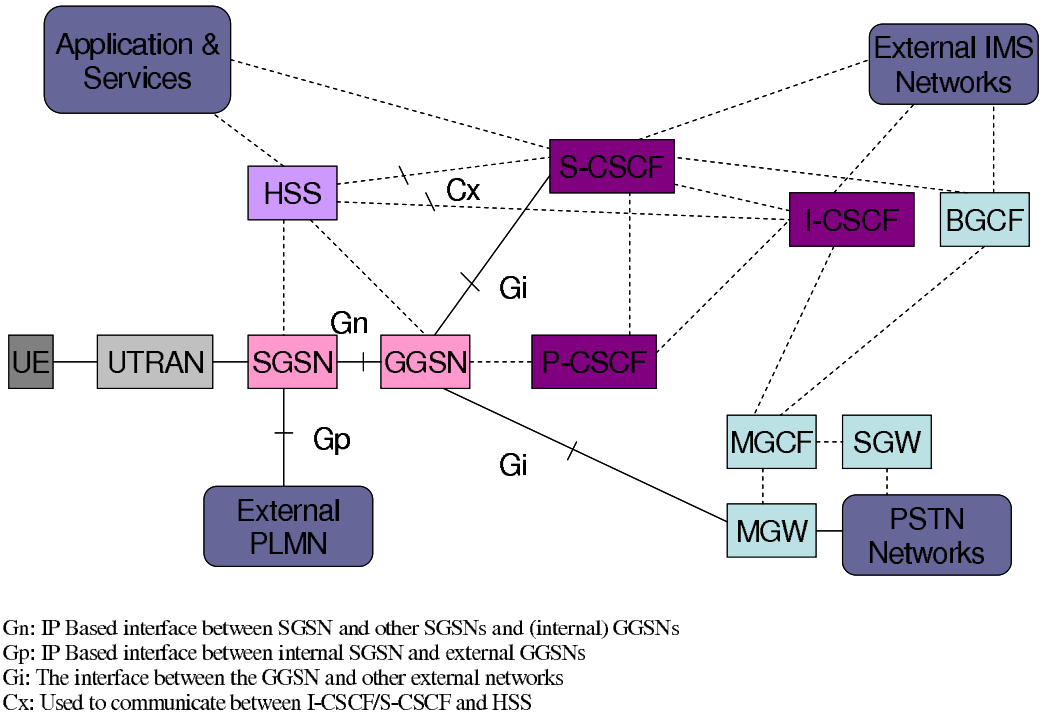


Figure 2.5: IMS Architecture

2.2.4 IMS

Unifying both CS domain and PS domain in the UMTS system was proposed to manage efficiently all kind of traffic that may be present on the network. Meanwhile, there are still separation between real time traffic handled by the CS domain and best effort traffic handled by the PS domain. In order to offer both real time and non-real time services on the PS domain, QoS requirements must be accomplished by the PS domain.

Choosing IP to transport real time and non real time services leads to conceive a common control plane based on IP. This control layer called IP Multimedia Subsystem (IMS) in the R5 of UMTS takes in charge all the signalling of multimedia sessions so that the PS Domain can afford QoS mechanisms for real time traffic.

The main goal of the IMS is to separate the transport in the PS domain from multimedia session control. We can summarize the main principles of the IMS by the following:

- Separation of transport and session control.
- Both signalling and packets transport are achieved in the PS Domain.
- Universal session control achieved by Session Initiation Protocol (SIP).

The main entities constituting the IMS are depicted on Figure **2.5**.

Besides the main components of a UMTS system (SGSN, GGSN and UTRAN) a new Call State Control Function (CSCF) is introduced. This new entity can play several

roles Proxy CSCF (P-CSCF), Serving CSCF (S-CSCF) or as an Interrogating CSCF (I-CSCF). Other entities are also used to support the internetworking with external CS networks. These are essentially gateways to legacy networks. Table **2.2** summarizes the role of these entities.

Table 2.2: IMS Entities

Entity	**Role**
S-CSCF	Handling the session states
I-CSCF	The contact point within an operator's network
P-CSCF	Proxy server representing the first contact point within the IMS core network
HSS(Home Subscriber Server)	Master Database
MGCF(Media Gateway Control Function)	Signaling plane gateway
MGW (Media Gateway)	Transport plane gateway
SGW (Signaling Gateway)	Translating signaling mechanisms to IP
BGCF (Breakout Gateway Control Function)	Selecting the PSTN network for internetworking

The IP core network used in the IMS makes it easy to define several QoS profiles and exchanging QoS demands. On longer term, IMS will enable a smooth migration to the All-IP architecture.

2.2.5 Wireless LAN

Mobile networks provide high wireless bandwidth to users without geographical limits. However, some applications can be deployed on small networks in a limited geographical area without the need of a well established infrastructure. This is achieved by Wireless Local Area Networks (WLANs) [CWKS97] as an alternative to the high installation and maintenance costs incurred by wired LAN infrastructures.

The wireless medium for local area networks is described by two standards: the European Telecommunications Standards Institute (ETSI) High-Performance European Radio LAN (HIPERLAN) and the IEEE 802.11 WLAN. Essentially, these two standards cover only the physical layer and medium access control (MAC) sub-layer of the open systems interconnection (OSI) seven-layer reference model. However, the IEEE 802.11 WLAN is gaining more acceptance as a worldwide standard. Thus, we will be limited to this standard and its extensions to offer QoS for real time applications.

From an architecture point of view two major variants of WLAN are deployed: Infrastructure WLAN and Independent WLAN. The main difference between these two architectures is in the access control imposed on stations forming in a WLAN. Generally a group of stations in a WLAN are referred to as a Basic Service Set (BSS) in IEEE 802.11 standard. The concept of a WLAN is to let stations in a BSS communicate with

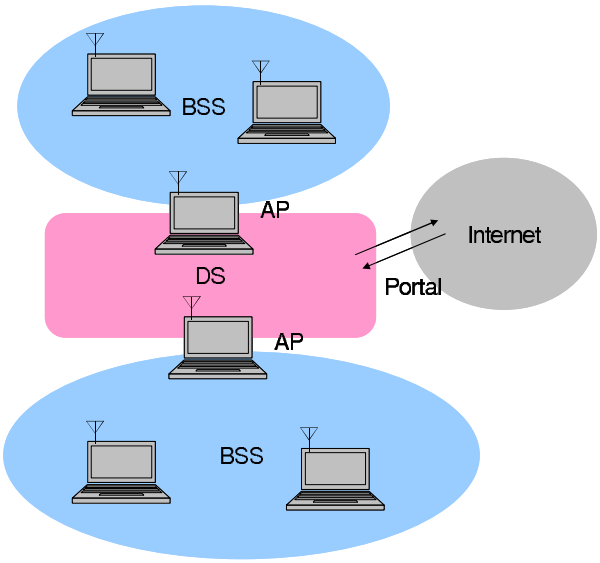


Figure 2.6: IEEE 802.11 WLAN Architecture

all other stations in a BSS. However, in Infrastructure WLAN all traffic transits by a delegated Access Point (AP), while in Independent WLAN also referred to as Ad-Hoc WLAN, there is no AP and traffic transits from source to destination directly. Of course the ad-hoc mode is very interesting as no infrastructure needs to be defined in order to start communications. However, this implies diverse routing and transmission difficulties that are the motivation behind several researches. The presence of an AP allows channelling all traffic through a centralized point but charges one station to take the role of a maestro to coordinate transmissions between all stations. In both cases, there is a limited geographical area covered by the BSS generally known as the Basic Service Area (BSA), which is analogous to a cell in a cellular communications network.

Figure **2.6** illustrates a simple Extended Service Set (ESS) developed with two BSSs, with a common Distribution System (DS), and a portal access to the Internet.

The IEEE 802.11 standard supports two data transfer modes: asynchronous services and synchronous services (or time bounded services). End-to-end delay is the main criterion to distinguish between two service types. File transfers which are delay insensitive are achieved by asynchronous data transfer mode, while real time services such as voice and video services are very sensitive to delay and require synchronous transfer mode.

The Medium Access Control MAC procedures determine the transfer mode in WLANs. Thus, we differentiate two different MAC schemes to transport asynchronous and synchronous services: the Distributed Coordination Function (DCF), very similar to best effort delivery of the data in packet networks and the point Coordination Function (PCF) primarily used for real time traffic transmission. While in wired LANS the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used, in WLANs stations use the Multiple Access with Collision Detection (CSMA/CD) for the DCF scheme as stations are unable to listen to the channel for collisions while transmitting. On the other hand, the PCF scheme is a centralized polling scheme controlled by an access point (AP).

PCF performance is judged very poor and it was not implemented commercially.

It was very quickly understood that with only DCF and PCF no QoS could be achieved on WLANs and this has motivated the elaboration of new standard IEEE 802.11e in order to enhance QoS by new coordination mechanisms. With EDCF (Enhanced DCF) and HCF (Hybrid Coordination Function), the IEEE 802.11e aims to support both IntServ (Integrated Service Architecture) and DiffServ (Differentiated Service Architecture). Thus IEEE 802.11e allows defining up to 8 classes of traffic ranging from best effort (Data) to real time (Video and Audio). The deployement of real-time applications with IEEE 802.11e is in its earliest steps, and many issues still need to be handled. WLAN traffic classes are listed in Table **2.3**.

Table 2.3: QoS Classes in IEEE 802.11e

Priority	**Access Category**	**Service Classes**
0	0	Best Effort
1	0	Best Effort
2	0	Best Effort
3	1	Video Probe
4	2	Video
5	2	Video
6	3	Voice
7	3	Voice

2.2.6 Conclusion

Providing QoS to multimedia applications is an important issue in wireless access networks. In fact this is the major motivation behind the introduction of new technologies with higher bit rates and new traffic classes to differentiate applications. The convergence towards the IP core network put into evidence the necessity to suggest more coherent traffic classes and QoS mechanisms. This issue requires an important standardization effort between different manufacturers of different technologies. However, the existing QoS mechanisms deployed on the Internet have already proved their robustness and reliability, and they can be used in future heterogeneous networks. In the second part of this chapter we examine those QoS mechanisms in more details.

2.3 IP Multiservice Networks

Multimedia applications have different constraints in terms of Quality of Service (QoS). While audio and video (real time) applications require small delay and small variations on end-to-end delay (jitter), non real time applications (e.g. file transfers) are tolerant with time constraints and very strict with losses and transmission errors. Non real time applications or data applications may also require a warranty of minimal bandwidth for

transactional data transfer. This diversity in QoS constraints constitutes a challenge when transporting multimedia applications on best-effort networks.

QoS guarantee for transported flows requires mechanisms allowing differentiated handling of different traffic categories as well as signalling protocols for resource allocation according to QoS demands of applications. From a QoS management point of view we can distinguish two broad categories:

First, congestion management which itself has two big trends: reactive and preventive methods. In reactive methods network accepts a maximum of connections and when congestion occurs, sources reduce their rates. It is the main mechanism behind best-effort networks in which Transmission Control Protocol (TCP) reduces its rate by a sliding window algorithm. Conversely, preventive control seeks to take measures a priori to minimize the risk of congestion. Thus, admission control techniques and traffic policing and shaping mechanisms allow reducing congestion frequency on the network while buffer management techniques like Random Early Drop (RED) and Weighted Random Early Drop (WRED) offer guaranties for priority flows to respect the QoS imposed by their Service Level Agreement (SLA).

Second, packets scheduling which is an essential mechanism to guarantee the QoS of transported flows. It concerns mainly heterogeneous flows when certain flows surges may disturb real time traffic even in the absence of congestion. Although the implementation of other scheduling mechanisms than FIFO (First In First Out) is difficult in very high speed routers, network equipments provide more and more sophisticated mechanisms to realize packet scheduling taking into account traffic classes such as Weighted Round Robin (WRR), Weighted Fair Queuing (WFQ), Class Based WRR (CBWRR), Class Based WFQ (CBWFQ), etc...

Those different mechanisms of QoS and traffic management are present in all architectures developed for providing QoS in IP networks. Historically, the first QoS architecture proposed for the IP Networks was the IntServ (Integrated Services) model, in which a QoS is affected to each flow. At the same time, for scalability reasons the Internet community has developed the DiffServ (Differentiated Services) model, in which QoS is associated with flow aggregation. Thus, the notion of class of service has emerged. Both models for IP multiservice networks can be used with the Multi Protocol Label Switching (MPLS) protocol, which offers very fast packet switching on routers in the core network replacing the IP routing by a switching function.

In the following sections, we present major mechanisms for QoS and traffic management in IP networks. Then we give an overview of IntServ and DiffServ architectures, and finally, we take a look at the MPLS and its extended version GMPLS.

2.3.1 QoS Management Mechanisms

QoS management is a set of policies and mechanisms that allow a network to efficiently satisfy a diverse range of service requests. Admission control, scheduling, buffer management, are all considered as forms of QoS management. Indeed, QoS management schemes must be able to prevent and recover from network congestion. Major QoS management techniques are the followings:

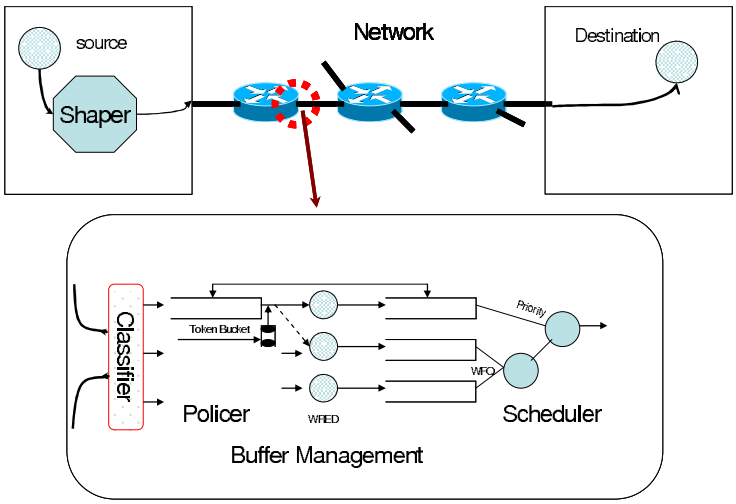


Figure 2.7: General Architecture of IP Network with QoS Support

Admission Control: It is the process of deciding what resources to allocate for a new incoming call. In other words it is an evaluation that network must perform before accepting a new connection. As a result it limits the number of flows to be admitted into the network such that each individual flow obtains the desired QoS.

Bandwidth allocation: This issue concerns bandwidth sharing and dynamic bandwidth allocation. The purpose is to adapt network resources to the varying data rate services that may be demanded by users, and by consequence the variable amount of bandwidth used at any one time.

Traffic shaping and policing: The aim of the these processes is to smooth traffic entering the network, since smoothed traffics are simpler to manage. Meanwhile, shapers always use buffers which may incur additional delays.

Traffic scheduling: Traffic schedulers arbitrate between packets that are ready for transmission on the link. Several scheduling mechanisms are available in multiservice networks such as the Weighted Round-Robin (WRR), the Weighted Fair Queuing (WFQ) and Priority Queuing (PQ) schemes.

Buffer Management: To prevent buffer overflow occurs under heavy loads, decisions must be made about which packets have to be discarded in order to maintain stability in the network. Different mechanisms exist for buffer management ranging from tail drop mechanism, in which all incoming packets are discarded once a queue becomes full to WRED in which a per-class of service dropping algorithm is used.

We will address the admission control and bandwidth allocation issues in detail in **Chapter 7** when using an extended SIP over DiffServ domain architecture for managing sessions' QoS. In the next sections, we give an overview of traffic shaping, policing, scheduling and buffer management mechanisms.

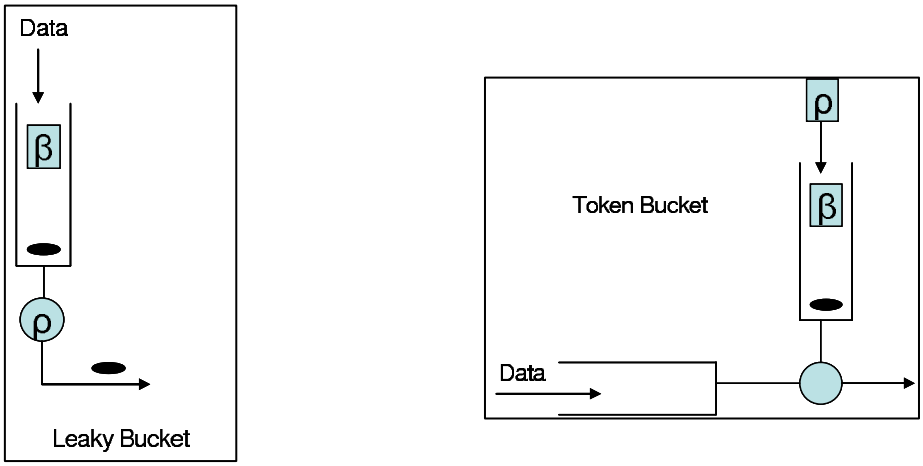


Figure 2.8: Leaky and Token Bucket

2.3.1.1 Traffic Shaping

Traffic shaping provides a mechanism to control the amount of traffic being sent into the network and the rate at which the traffic is being sent. It also may be necessary to identify traffic flows at the ingress point (the point at which traffic enters the network) with granularity that allows the traffic shaping mechanism to separate traffic into individual flows and shape them differently.

Two major methods for shaping traffic exist: a leaky-bucket implementation and a token-bucket implementation.

Leaky-Bucket Implementation Leaky-bucket is used to control the rate at which traffic is sent to the network. A leaky bucket provides a mechanism by which bursty traffic can be shaped to present a steady stream of packets to the network. The concept is that each flow has its own leaky bucket (see Figure **2.8**). During data transmission packets are placed in the buffer to be sent on the network at the fixed rate ρ. The buffer size β limits the number of packets waiting for transmission. As a result flows accumulating packets more than the buffer capacity will be seen eliminating the excess.

The main effect of leaky bucket is to transform traffic bursts into a regular packet flow emitting each $1/\rho$ time unit. While the big advantage of this mechanism is its simple implementation and description of traffic via the rate ρ, it wastes a lot of resources when it deals with highly variable traffics (buffer allocation according to peak rate).

Token-Bucket Implementation Another method of providing traffic shaping and ingress rate control is the token bucket. The token bucket differs from the leaky bucket substantially. Whereas the leaky bucket fills with traffic and steadily transmits traffic at a continuous fixed rate when traffic is present, traffic does not actually transit the token bucket. In the token bucket, the rate ρ is the one at which tokens are placed in the bucket of size β. If the bucket is full new arriving tokens are eliminated. While the leaky bucket transforms traffic bursts into regular flows, token bucket allows some

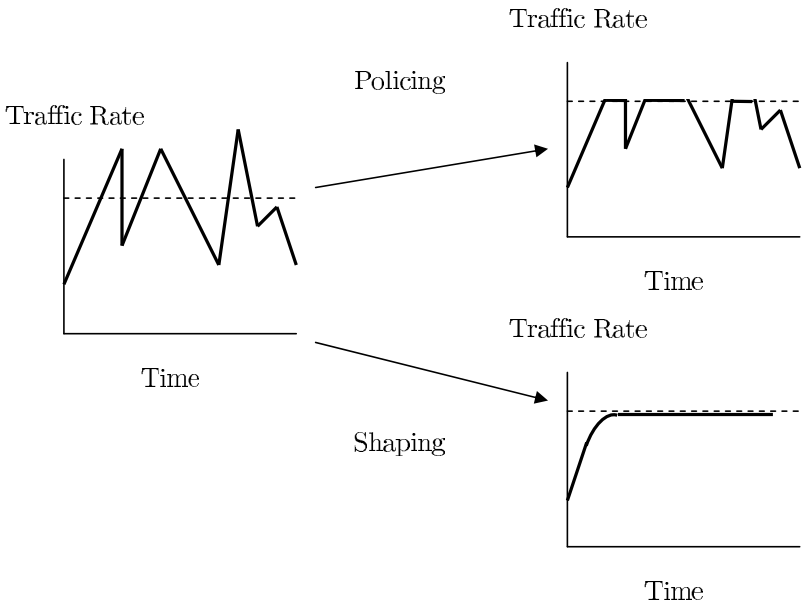


Figure 2.9: Traffic Policing versus Traffic Shaping

limited traffic bursts. Thus during a time period τ the token bucket guarantees that no more than an amount of data equivalent to $(\beta + \tau\rho)$ tokens is transmitted. Besides, the token bucket eliminates tokens not data, allowing the network to manage transmission buffers. The main drawback of token bucket is that the traffic policing function is more difficult to implement than in the leaky bucket. Moreover, it may allow quiet flows to have a big number of unused tokens, and by consequence consume a lot of bandwidth suddenly. That is why it is sometimes proposed to regulate the transmission rate by a leaky bucket of rate $C > \rho$.

2.3.1.2 Traffic Policing

Traffic shaping and policing may work together but they are different. In general, when the traffic rate exceeds the allowed maximum rate, excess traffic is remarked (or dropped). The result is an output rate that appears as a saw-tooth with crests and troughs. Indeed, traffic policing may propagate bursts. On the other hand, traffic shaping retains excess packets in a buffer and then schedules the excess for later transmission over increments of time. The result of traffic shaping is a smoothed packet output rate.

2.3.1.3 Packet scheduling

The idea behind Packet scheduling is to adapt the order of packet transmission in buffers according to the QoS constraints of the corresponding flows. The main quality of a packet scheduler is to achieve a good isolation of flows, easy and scalable implementation and minimum service time for packets. In general three types of scheduling are used: Fixed

priority scheduling, Generalized Processor Sharing (GPS) paradigm based scheduling and time slot based scheduling.

Fixed priority scheduling: Fixed priority schedulers serve packets according to their priority. When a high priority packet is present, low priority packets must wait. This kind of schedulers offers the minimum possible delay for high priority packets. Nevertheless, the big drawback is that low priority packets are not isolated from high priority ones. However, this algorithm is very simple and executes efficiently.

GPS based scheduling: GPS based schedulers [Tou98] achieve differentiated sharing of the bandwidth with guaranteed minimum for each session. Bandwidth sharing is achieved using a weight system per class of service. One flow may get temporarily more than the bandwidth share assigned to it if no other packets from other flows are waiting. The complexity of this kind of schedulers is function of the number of flows n, which make it unusable in the IntServ architecture when n corresponds to flows transported by the network. However, n has a reasonable value in the DiffServ architecture (aggregated flows) which is the main domain of application of this scheduler. Thus for $1...n$ sessions the associated positive weights $(\phi_i)_i = 1...n$ follow the GPS service discipline if the service (W) received by two sessions i and j in the time interval $[t_1, t_2]$ verifies:

$$\frac{W_i[t_1, t_2]}{\phi_i} = \frac{W_j[t_1, t_2]}{\phi_j} \tag{2.1}$$

Practically, for a given bandwidth B, the bandwidth share B_i assigned to each session is given by:

$$B_i = \frac{\phi_i}{\sum_{i=1}^{n} \phi_i} B \tag{2.2}$$

Time slot based scheduling: In this type of schedulers, time is divided into slots with constant or variable durations. These slots are assigned to sessions according to a round robin scheme. From a QoS point of view, these schedulers are less efficient than GPS ones (especially for delay and jitter), but they are simpler to implement.

2.3.1.4 Buffer Management

Generally, when a buffer gets into overflow, all packets are discarded. This is the simplest queuing management mechanism: the tail drop. When all incoming packets conform to basic Poisson arrivals with exponential inter-arrival times, tail drop discarding provides a good means of utilizing a queue when the load exceeds the capacity. However, Internet traffic is inherently bursty with highly variable throughput. The queues located at routers and switches must take into consideration the abrupt bursts of data.

Several buffer management algorithms were proposed alternatively. Most of them are based on Random Early Drop (RED) algorithm. RED was designed to avoid TCP

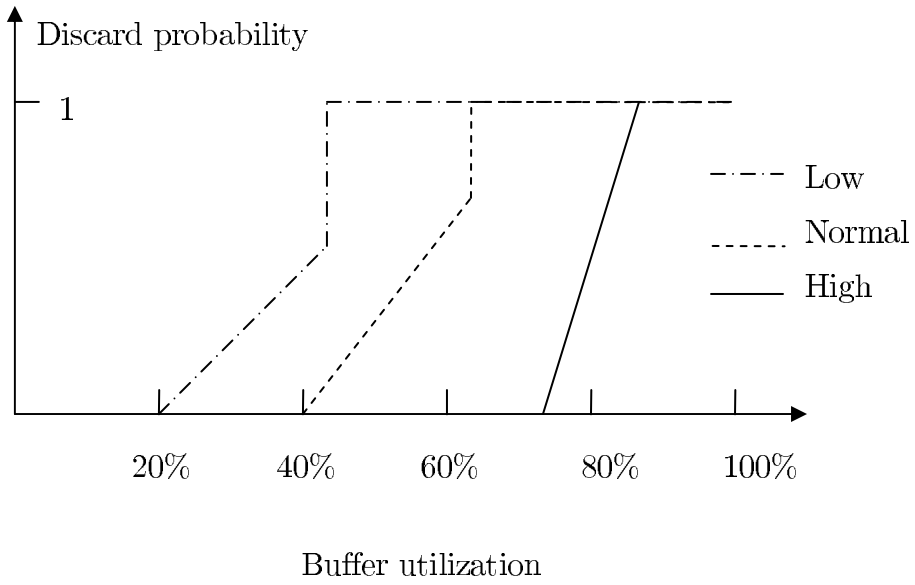


Figure 2.10: WRED Algorithm

synchronization between connections. Initially, RED discards or marks packets probabilistically according to queue filling to avoid reaching overflow conditions. RED uses a monotonically increasing function for marking or dropping packets. It is a linear function used to determine whether a packet will be accepted or not to the queue when k packets are currently present in the system. There is a set of parameters used to determine the dropping probability $d(k)$. Indeed, those parameters create the drop function, and they are defined hereafter:

- Min_{th} : Minimum Queue Fill for RED Dropping.
- Max_{th} : Maximum Queue Fill for RED Dropping (Normally set to K, the maximum queue size).
- Max_p : Maximum Probability of RED Dropping.

For instantaneous queue size k, $d(k)$is given by:

$$\begin{aligned} &d(k) = 0 \; if \; k < Min_{th} \\ &d(k) = 1 \; if \; k > Max_{th} \\ &Otherwise: \\ &d(k) = Max_p \cdot \left(\tfrac{k - Min_{th}}{Max_{th} - Min_{th}}\right) \end{aligned} \tag{2.3}$$

In multiservice networks Weighted RED is used instead of RED. WRED (see figure **2.10**) is a variant of RED in which drop parameters are defined on a per-class basis, allowing a differentiated treatment for each class of service.

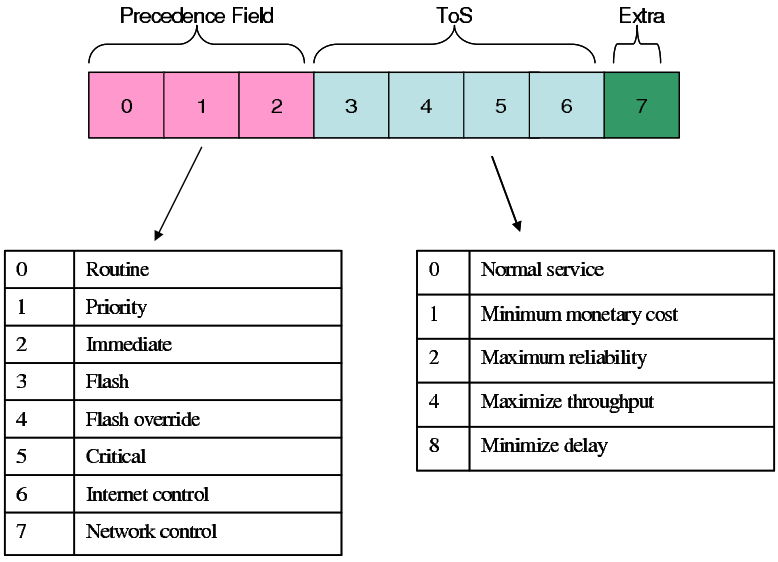


Figure 2.11: Type-of-Service Field

2.3.2 IntServ

Since IP protocol does not support any QoS, the Internet community developed two architectures to support QoS on IP best-effort networks: IntServ and DiffServ. The Integrated Services (IntServ) architecture suggests an end-to-end QoS solution. Basically, it supports two categories of applications:

- Real-time applications with latency sensitive traffic requiring guaranteed service.
- Best-effort applications.

In order to guarantee QoS requirements IntServ uses resource allocation policy. Resource allocation information is transported by Reservation Protocol (RSVP) messages and may end by a success or failure answer.

IntServ uses the Type-of-Service Field (ToS) of the IP header. This field contains 8 bits as illustrated on Figure **2.11**. Eight different classes can be defined at the ingress point using the precedence bits (3 bits). Precedence bits are used by network nodes to forward packet along the packet's route based on its class. Although the ToS field has 4 bits to classify packets, only 5 values are defined.

There are many problems with IntServ model. Indeed, only relative priorities are allowed by the precedence field. Moreover, practically only few routers make use of the ToS field. However, the main problem with the IntServ is scalability. The per-flow management of QoS is very powerful, but only efficient with a small number of flows n. It requires a lot of ressources when n becomes very big.

2.3.3 DiffServ

The IntServ model was soon replaced by a simpler architecture model, the DiffServ architecture [BBC+98]. The key idea is to classify and possibly condition the traffic entering the network at the ingress point of the network. This is achieved by assigning different behaviour aggregates (BAs) to traffic. BAs characterize each class of traffic and are identified by a unique Differentiated Services Code Point (DSCP). However, in the core network, each DSCP is associated with a Per-Hop Behaviour (PHB) and packets are forwarded according to their PHB.

At the ingress point (border entry) of a network the conditioner applies PHBs and marks the traffic. The choice of the corresponding PHB depends on policy criteria defined previously. Then traffic is forwarded according to the PHB marking. Indeed, each node offers different management for each PHB. Finally, at the egress point of the network (border exit) traffic is unmarked.

DSCPs correspond to different service levels that are used generally to classify applications traffic. Two main traffic classes are defined by PHB:

- Expedited Forwarding (EF): It has a single DiffServ value, specified to describe the highest level of aggregate quality of service. Minimum delay and jitters are only allowed. The local policy defines traffic profile to be respected, otherwise traffic is discarded. The DSCP for EF is 101110.

- Assured Forwarding (AF): It has 12 DSCPs resulting from four different classes jointly with three drop precedence priorities. The local policy defines traffic profile to be respected. However, AF traffic exceeding the traffic profile is only demoted not discarded. Table **2.4** lists the usual values used for AF DSCPs.

Table 2.4: DSCPs for AF Classes

Drop Precedence	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Low Drop	AF11= 001010	AF21= 010010	AF31= 011010	AF41= 100010
Medium Drop	AF12= 001100	AF22= 010100	AF32= 011100	AF42= 100100
High Drop	AF13= 001110	AF23= 010110	AF33= 011110	AF43= 100110

Finally, we should note that there is also a best-effort forwarding class defined by the Default Forwarding (DF) traffic class, for which no preferential handling is prescribed.

The DiffServ architecture defines units called DiffServ domains. Inside domains we distinguish between edge routers and core routers. The simplicity and scalability of DiffServ architecture is due to the separation of functionalities between edge and core routers. While classification and conditioning functions are implemented at the edge routers, core routers only forward packets according to their traffic class. Core routers have no need to store any flow specific information to achieve their work.

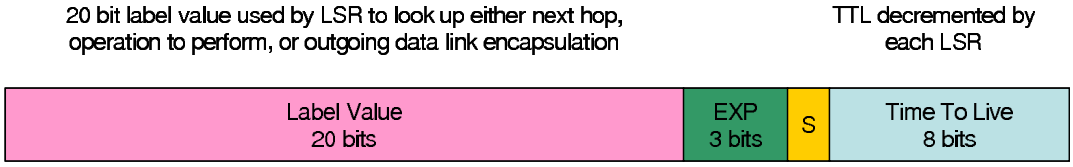


Figure 2.12: Generic MPLS Label

2.3.4 MPLS

Multi-Protocol Label Switching (MPLS) [RVC01] combines the functionalities of Layer-2 (switching) and Layer-3 (routing) in order to provide traffic engineering capabilities. Using the switching technology enhance considerably scalability and interoperability issues. It allows mapping the Layer 3 traffic into a Layer 2 connection-oriented transport much like the switching used by ATM and Frame Relay. Switching is based on added labels containing routing information specific to each IP packet. Using these labels virtual paths can be defined by routers for each class of traffic.

Functionally, the MPLS standard uses only 3 bits from the precedence bits of the IP header (EXP bits), and it can only address eight possible PHBs. Whereas, DSCPs of the DiffServ architecture can have up to 64 possible values (see Figure **2.12**). As a consequence, not all DSCPs can have their own QoS while being switched by Label Switching Routers (LSR) in the network. DiffServ traffic classes are aggregated and switched by group of aggregates.

Virtual connections used by MPLS to forward IP packets are called label switched paths (LSPs). Establishing LSPs is done by the Label Distribution Protocol LDP and RSVP-TE protocol using routing tables of the Layer 3. Hence, a logical connection oriented network is formed above the connectionless IP network, and new traffic engineering procedures could be applied to enhance traffic QoS and delivery.

2.3.5 GMPLS

The MPLS architecture offered a first step towards the integration of many heterogeneous technologies such as Dense Wavelength Domain Multiplexing (DWDM) and The Synchronous Digital Hierarchy (SDH). Indeed, the IETF searched to apply the techniques used in the MPLS architecture in optical networks. First, a Multi Protocol Lambda Switching (MPλS) architecture was suggested using the packet transmission wavelength to define packet labels. However, MPλS was quickly abandoned and gave its place to a more generalized framework which is: Generalized MPLS (GMPLS) in which, MPLS and MPλS appear as special cases.

GMPLS seeks to enhance the performance of heterogeneous networks by reducing operational costs and better integrating packet switched domains with optical/synchronous domains. As a consequence, there will be a common control plane handling all network devices equally, and all legacy switching mechanisms: MPLS label switching, MPλS lambda switching and space/time switching of TDM interfaces will be unified into a generalized label switching mechanism that needs to be specified. An example of GMPLS

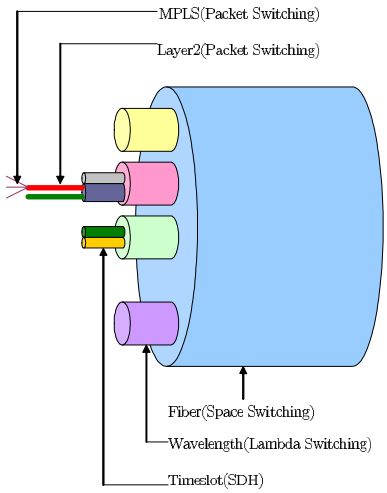


Figure 2.13: Example of GMPLS Hierarchy

hierarchy is shown on Figure **2.13**.

Further details about the GMPLS architecture and its extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control can be found in [Man04, MP04].

2.4 Conclusion

The interconnection between different telecommunication networks with different access technologies has evolved to a heterogeneous environment. Heterogeneous networks define many challenges to QoS guarantee for real time applications. Many compatibility issues will appear as QoS classes are very different from one network to another. Fortunately, the core network is converging towards an All-IP network and new technologies like GMPLS will unify control plane and packet forwarding techniques. However, the introduction of new multimedia services with very strict requirements implies QoS guarantees. Telecommunication operators should verify the good operation of their networks. Basically, this will be achieved by performance evaluation studies involving accurate models for both network components and multimedia services deployed over these networks. There will be an increasing need for modelling and simulation tools over large scale networks, and consequently reliable and efficient modelling of networks and multimedia traffic. In the next chapters we focus on multimedia traffic modelling with different aspects: IP traffic modelling, single multimedia applications modelling, aggregated multimedia applications modelling, transport protocols modelling. Finally, we explore the extension of the SIP signalling protocol functionalities to provide QoS on a per-flow basis.

Chapter 3

Generic Framework for Traffic Modelling

3.1 Introduction

Traffic modelling is a key tool for performance evaluation and resource provisioning in networks. Traffic models support efficient network dimensioning procedures and traffic management functions; they assist in characterizing traffic behaviour for Quality of Service (QoS) ends and help to estimate the resource utilization. A clear understanding of the nature of traffic in the target system and subsequent selection of an appropriate traffic model are critical to the success of the network modelling and performance evaluation process.

The traffic process can be described in terms of the characteristics of a number of objects, including packets, bursts, flows and sessions, depending on the time scale of relevant statistical variations. The preferred choice for modelling purposes depends on the object to which traffic controls are applied. Conversely, when designing traffic controls it is necessary to bear in mind the facility of characterizing the implied traffic object. Whatever the traffic description method used, a traffic model should be able to characterize the network dynamics with an acceptable level of accuracy.

Traffic modelling is a vast and complicated subject. We distinguish between two methodologies of traffic modelling: Traffic modelling based on traffic traces (Trace modelling) and traffic modelling based on source behaviour (Source modelling). The difference between the two methods is in the level at which the traffic modelling process is taking place.

Trace modelling overpasses the particularities of each application and handles the aggregated traffic modelling issue. Thus, traffic is measured at network backbones while trace files with detailed packet information are generated. Then a statistical analysis of the trace is achieved and aggregated traffic models are proposed to generate the traffic. Trace modelling (or IP Traffic Modelling) will be the object of **Chapter 4**.

On the other hand, Source modelling characterizes the behaviour of applications at the source level. Indeed, Source modelling is very dependent on application type (Audio,

Video, Data), transport protocol (TCP, UDP, RTP,...) and communication medium (Wireline, Wireless). New multimedia applications deployed like Voice over IP (VoIP) and Visiophony etc ... introduced new coding mechanisms. Audio and Video CODECs (COder/DECoder) produce different traffic profiles to respect specified Service Level Agreements (SLAs). Furthermore, transport protocols may influence traffic profiles, especially when Transmission Control Protocol (TCP) is used because of its closed-loop control mechanisms. Source modelling (or Multimedia Application Modelling) will be the object of **Chapter 5**.

Whether we have to deal with source modelling or trace modelling issues, the needs in terms of simulation, design and analysis tools are very similar. This motivates us to search a global approach to allow traffic modelling and evaluation in a generic way without the need to construct a modelling structure for each traffic model. In this Chapter we will present a generic framework designed and implemented to allow maximum flexibility when designing traffic models. First, we discuss the motivation of our approach along with a brief overview of transport protocols. Then we present the structure of the generic framework for traffic modelling. Finally, we present the statistical analysis methods and tools that will be used along this thesis to characterize traffic and estimate model parameters.

3.2 Motivation

New services offered by telecommunication networks, emphasized the importance of Quality of Service (QoS) for multimedia applications. Operators need simulation tools with real models of different flows that may be present on their networks. In fact, this issue is fundamental as it allows to study off-line the network behaviour and the influence of different parameters on QoS perceived by users.

Stochastic modelling approaches in simulation/emulation tools are suitable to handle the diversity of Internet flows. Although a traffic source may be very simple with a constant or exponential bit rate, it may also produce bursts of packets. For example, on Figure **3.1** we show four traffic generation profiles with the same average throughput. The simulation of a traffic source consists of finding a model to generate the packet creation dates with defined sizes according to the application type. Besides, user behaviour and transport protocols must be considered when designing traffic models.

We distinguish between two categories of applications according to temporal constraints. Real time applications, often associated with the User Datagram Protocol (UDP), and non real time applications associated with Transmission Control Protocol (TCP). In real time applications, packet inter-arrival times are defined by the source. Thus, packets arrive to the destination respecting predefined constraints. Usually, some modifications are allowed on packet delay and jitter according to the QoS profile requested by the service (usually known as the Service Level Agreement (SLA)). Respecting the SLA allows the destination restoring the original signal without loss of information.

On the other hand, the source in non real time applications, cannot define the packet rate or generation profile. It concerns in general data transfers, in which variable packet

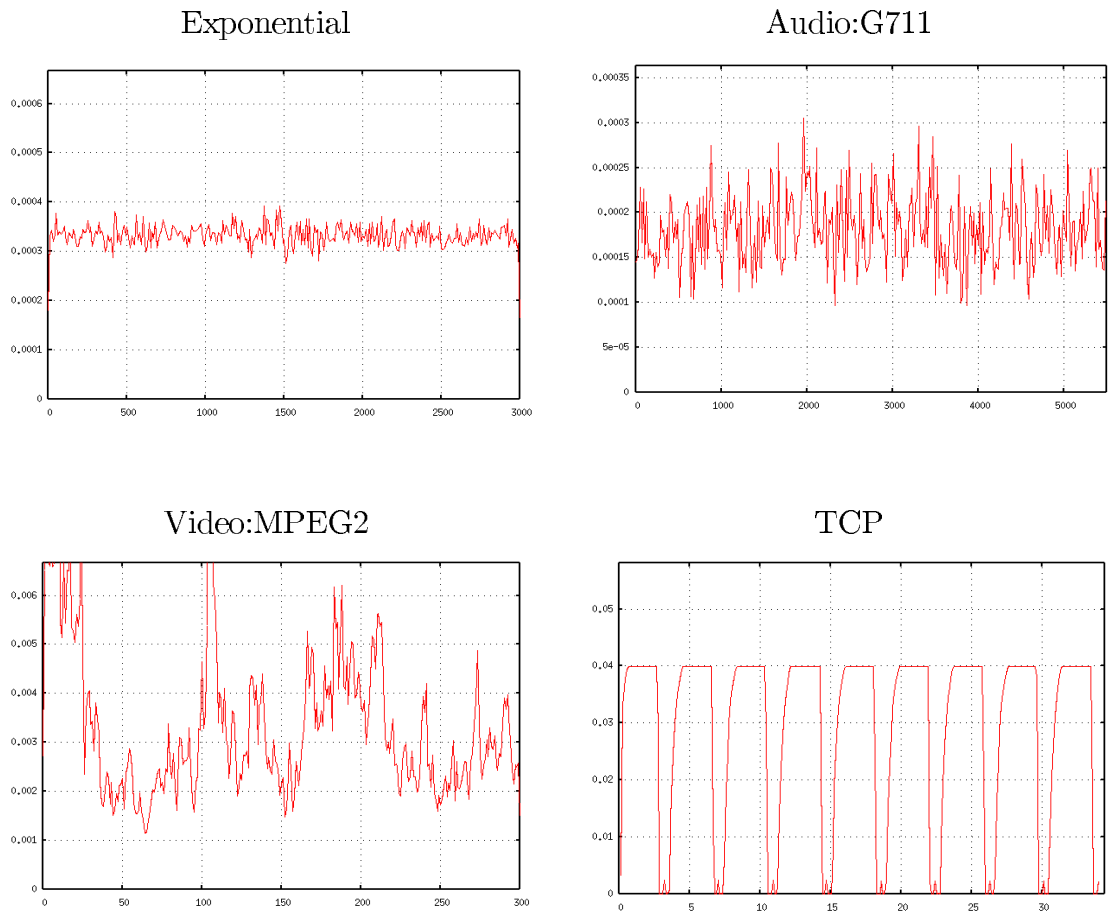


Figure 3.1: Diversity of Packet Generation Profiles in Multimedia Applications

delay and jitter are tolerated while losses are not allowed. In this case, QoS is not expressed in the same terms as for real time applications (delay, jitter ...), but rather as assured nominal bit rates. This kind of applications relying on TCP generates "Elastic" traffic with the following characteristics:

- The packet rate varies from one packet to several packets per time-unit according to the congestion window of TCP.
- The packet rate changes suddenly when the source detects the loss of one packet and retransmits the lost packets.
- The packet rate depends on the Round Trip Time (RTT) observed on the network. RTT refers to the time needed by the packet to arrive to its destination plus the time needed by its acknowledgement (ACK) to return to the source.

The major difference between these two categories of applications is in the packet transmission profile regarding the network conditions. The packet rate of UDP applications is independent of the network and is determined by the source, while the packet rate of TCP applications varies according to network conditions.

In order to handle the diversity of applications deployed on the Internet as well as the wide number of protocols used to achieve packet transport, we propose a hierarchical representation of multimedia applications. Our approach is based on a three level architecture of multimedia applications: Session, Activity and Packet. Our goal is to provide a framework for traffic modelling capable of representing most of applications deployed on the Internet. The framework makes it possible to simulate and evaluate multimedia applications in a generic way. Furthermore, the framework can host mathematical traffic models for testing and performance evaluation issues. This is particularly interesting for trace based models in which the user behaviour (or application structure) is not relevant.

3.3 Transport Protocols

The deployment of new multimedia applications over best-effort networks was possible due to advances in coding technologies as well as transport protocols assisting multimedia applications in managing packets transmission procedure. Hence, besides the two basic transport protocols UDP and TCP a wide range of multimedia protocols was introduced. Most of these protocols are based on TCP and UDP. However, they add new functionalities that ease the communication between end users and help in respecting SLAs. In this section, we give a brief overview of transport protocols from a traffic modelling point of view.

3.3.1 User Datagram Protocol (UDP)

UDP is a connectionless communication protocol of the transport layer (of the OSI standard). Basically, there is no guarantee that any UDP sent packet arrive to destination.

UDP offers no quality of service other than the one offered by the underlying Internet Protocol (IP). However, some extra information is added to the Header of the UDP packets to specify the source and destination ports that can be used to differentiate connections or services between two end-points.

UDP is used basically in local area networks (where loss probability of IP packets is very small). Besides, the connectionless mode of transport used by UDP is very useful for transporting real time applications as we will see in the RTP section later.

3.3.2 Transmission Control Protocol (TCP)

TCP/IP (Transmission Control Protocol on IP) is a connection based protocol of the transport layer (of the OSI standard). TCP achieves bidirectional, in-order, reliable transmission of packets. It implements congestion control and packet retransmission mechanisms in a closed-loop algorithm.

TCP is used by many other protocols from the simplest (telnet and rlogin), File transfer Protocol (FTP), to the most complex HyperText Transfer protocol (HTTP) and Simple Mail Transport Protocol (SMTP). TCP implements several mechanisms to control and validate data before delivering it to overlying layers:

- TCP transfers packets to the IP layer by segments in order to optimize resource utilization.
- A timeout timer is associated with each segment upon transmission. If no Acknowledgment is received before its expiration the segment is considered as lost and retransmitted.
- Each segment has its own checksum. A segment with an invalid checksum upon arrival is considered as lost.
- TCP rearranges segments and delivers them in order. Recall that IP does not guarantee any reception order.
- Reception buffers are limited on both ends. TCP assures that data sent by one end does not exceed the reception buffer size of the other end.

TCP operation relies essentially on the loss detection mechanisms. However, new access networks (particularly wireless networks) operate on high loss rate medium making the standard loss detection mechanisms of TCP inadequate. For example, a lost ACK on a noisy wireless channel may be interpreted as normal congestion which is completely wrong and may induce useless reactions of the TCP algorithm. Some recommendations introduced by the RFC3481 [LGK03] can help to enhance the performance of TCP on the wireless medium. TCP will be explained in more details in **Chapter 6** as we present the differential analytical model of TCP.

The closed-loop packet retransmission and congestion control mechanisms of TCP makes it inappropriate for real time applications. Recall that real time applications have

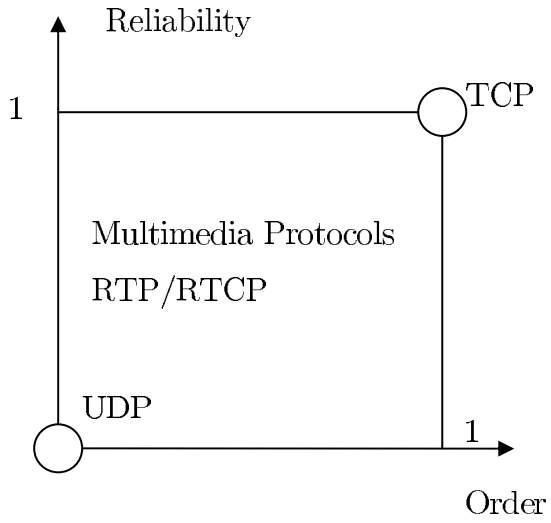


Figure 3.2: Multimedia Protocols

strict SLAs (maximum loss rate, maximum delay and maximum jitter) which can not be respected when using TCP. Therefore, new multimedia protocols were introduced allowing loose congestion control mechanisms resulting in reliable non elastic traffic as we will see in the next section.

3.3.3 Real Time Protocol (RTP)

RTP is the standard protocol used over the Internet for transporting multimedia applications (see RFC 1889 [SSFJ98] for complete specification). It contains two parts: Data Transport achieved by RTP itself and Data Control achieved by Real Time Control Protocol (RTCP). The data part of RTP protocol takes care of real time properties including: loss detection, timing, identification and security. While RTCP support source identification, audio and video gateways as well as the synchronization of different media streams. From a functional point of view RTP and RTCP offers an alternative to TCP for real time applications where small loss rates are allowed. Indeed, the elastic traffic generated by TCP does not comply with real time constraints required by audio and video applications. Therefore, a loose notion of reliable packet transmission was introduced by RTP/RTCP (see Figure **3.2**) in which information about congestion are communicated between end users without packet acknowledgment and retransmission mechanisms. However, the source takes into account this information to reduce packet rate while respecting the SLA of real time traffic. Thus, a communication must be stopped if the congestion information exchanged reveals a lack of resources to assure the required QoS. Nevertheless, from a modelling point view RTP/RTCP is based on UDP and RTP/RTCP packets can be specified as UDP packets with appropriate Headers.

3.3.4 Other Multimedia Protocols

Besides the three basic transport protocols described previously, there are many multimedia protocols developed to enhance multimedia content delivery over the Internet. Such as the Streaming Control Transmission Protocol (SCTP), Real Time Streaming Protocol (RTSP), Session Initiation Protocol (SIP), etc. These protocols offer advanced controls to transfer multimedia content that can be considered as application level functionalities (much like HTTP). However, they are based on UDP or TCP on the packet level and are modelled accordingly.

This preliminary investigation of multimedia protocols underlines the importance of separating application/traffic models based on the packet transmission technology (or protocol). This idea is adopted in our framework and a three level representation of application/traffic models is provided. Using this structure, we model user populations, application activity behaviour, and packet generation profile separately allowing a flexible modelling of multimedia application/traffic sources.

3.4 Generic Framework for Traffic Modelling

The generic framework targets multimedia application modelling as well as IP traffic modelling. Two issues were considered when designing the tool: The packet transport protocols and the application behaviour. From a traffic modelling point of view we have two types of packet generation profiles: deterministic with UDP packets and closed-loop controlled with TCP packets. On the other hand, multimedia applications on packet switched networks share a common behavioural structure composed of active and idle periods. Thus, despite the diversity of multimedia applications and traffic profiles they may produce, they can be modelled as a generalized ON-OFF process. Indeed, audio, video and data applications are composed of a succession of active and idle periods. According to the complexity of modelled applications we may have a simple periodic succession of ON and OFF periods or a multilevel non-periodic structure of ON-OFF processes. In order to take into account all possible behaviours we suggest a hierarchical representation of multimedia applications (and traffic sources in general). Thus, multimedia application behaviour can be recreated with three levels: Session, Activity and Packet.

- Session Level: It models the arrival of clients connecting to the system in order to use a given application.
- Activity Level: It models the density of exchanged information accross the time. This could be different depending on the application structure.
- Packet Level: It is the basic level at which packets are generated according to information exchanged during the application.

We describe in the following sections the role of each of these three levels.

3.4.1 Session Level

A Session level models the arrival process of clients into the system. It is the behaviour of the population of users at the access node (router on a LAN, Base station in radio networks ...). Two cases can be considered for client arrival process:

- Constant number of sessions: the number of connected users is fixed and does not change.
- Random number of sessions: sessions are established randomly and sessions' durations are application dependent.

Of course when the number of sessions is constant there is no arrival process to define, and session durations are considered illimited. The constant number of sessions model is very important to evaluate the performance of a system under constant number N of applications. On the other hand, the random number of sessions model involves two processes: the session arrival process and the session duration process.

The session arrival process can be of any type, but most common models are:

- Poisson model: It is the most popular model used to represent the client arrivals. This model represents an average rate of exponential arrivals λ, without memory, independent of previous arrivals.
- Truncated Poisson model (or Erlang model): Clients arrivals are still considered exponential but only a maximum number of active sessions is allowed. It is generally expressed in terms of blocking probability per type of session. This model is widely used in PSTN networks and still has applications in multiservice networks.

The session duration process depends on the transport protocols. In real time applications, the duration of a session is characterized by a distribution depending on the type of application. While in non real time applications, the communication time is controlled by TCP. In fact, in such case the source cannot determine the temporal sequence of packet inter-arrivals. This type of sources is better modelled by the quantity of information to transmit. The duration of the session depends on the network response and not on the source itself.

3.4.2 Activity Level

The density of information of an application is specific to each application. To describe this property, the activity level represents the application as a set of parallel flows. Each flow is composed of a succession of active and idle periods. The main behaviours are listed below:

- ON-OFF Behaviour in a conversational process (HTTP, VoIP...) where ON period represents the active period during which packets are transmitted and OFF period represents idle time.

- ON behaviour during all the session time, such as FTP file transfers. This behaviour may be used with complex processes like correlated event processes, in order to define aggregated traffic models.

Active periods (denoted ON) can represent any kind of activity, using the probability distribution of packet inter-arrivals and sizes. ON periods can also describe more complex behaviours by subdividing the ON period itself to sub-ON periods. The Web model, for example, can be described by simple ON period representing the page downloading time followed by an OFF period representing the reading time. ON and OFF periods are grouped in one entity called Pattern. In this case the pattern is the Web page and the occurrence of the pattern defines the repetition of the Web pages. However, the Web model can detail the activity in the ON period. Hence, the page downloading period can be divided into sub-ON periods to reflect the presence of inline objects (image, applet, video ...). See **Chapter 5** for more details.

The activity level handles another specific characteristic of multimedia applications which is parallel flows. Actually, different parallel flows may be present during application life time. Those flows concern main application activity and signalling protocols flows. For example, a voice or video application may be transported by RTP and controlled by RTCP.

3.4.3 Packet Level

This is the basic level of the hierarchical model. At this level one defines the way packets of the application will be generated and transported. If the UDP protocol is used then one determines the packet inter-arrival and size distributions. The packet inter-arrivals may follow any kind of distributions, while packet sizes can only follow truncated versions of distributions to respect packet size limits on the networks (e.g Internet, Ethernet ...). If TCP is used, packet inter-arrivals are determined by the TCP automata according to the reception rate of ACKs. Figure **3.3** illustrates the three levels.

The generic framework represents the core of a complete tool conceived for the design and the evaluation of multimedia traffic sources in network environment called Traffic Source Modeler (TSM). A more detailed description of the tool is given in **Appendix A**.

3.4.4 Packet Rate Estimation

Some multimedia applications are very complex. As a consequence, the corresponding application models conceived with the generic framework could be very sophisticated. Several flows with real time and non real time traffic patterns alternating could be present. For example in new chat sessions, end users exchange text, audio and video data. Hence, a simple evaluation tool for a new application model is to determine its average packet rate. Unfortunately, the packet rate estimation is not possible with TCP flows. As we know TCP has an elastic behaviour that depends on network congestion, thus the estimation of packet rate in this case is problematic. To overcome this difficulty

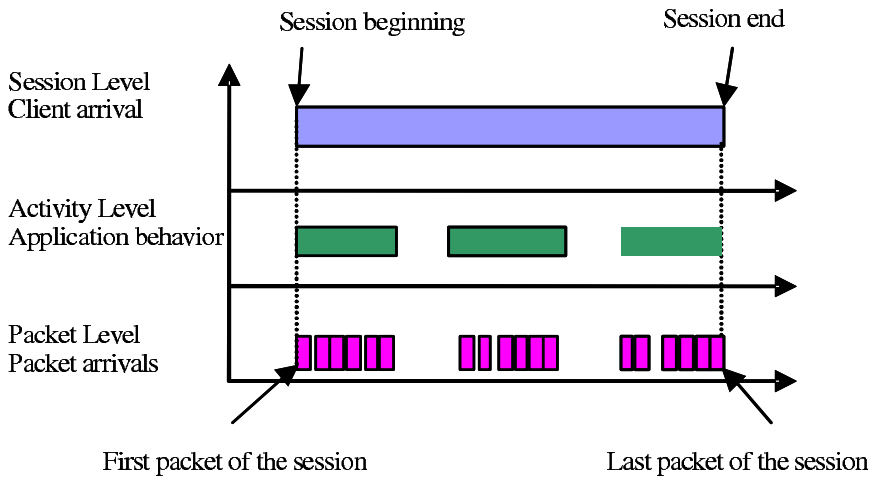


Figure 3.3: Three Level Description of Multimedia Applications

we suppose that the TCP source rate is bounded by terminal packet rate on which the multimedia application will be deployed. Indeed, this assumption is justified because TCP sources stabilize after the transient phase on a nominal packet rate normally equal to the terminal packet rate. Of course, packet rate estimation in this case is approximate but still useful to gain insight on possible packet rate generated by the application when deployed on the network. In the following we present the estimation of the average packet rate of a multimedia application conceived with the framework.

Generally, application models have different flows (streams). Each flow has patterns composed of ON and OFF periods. Consider:

- The number of streams is N_s.
- The number of repetitions of pattern m in stream s is $N_{m,s}$.
- The duration of pattern m in stream s is $T_{m,s}$.
- The packet rate of the period p in pattern m in stream s is $\lambda_{p,m,s}$.
- The duration of period p in pattern m in stream s is $T_{p,m,s}$.
- The period p file size (when it applies) in pattern m in stream s is $Q_{p,m,s}$.
- The average packet size during period p in pattern m in stream s is $P_{p,m,s}$.
- The packet inter-arrival during period p in pattern m in stream s is $IA_{p,m,s}$.
- The packet rate of stream s is λ_s.
- The packet rate of pattern m in the stream s is $\lambda_{m,s}$.

We want to estimate the packet rate (denoted λ) of a multimedia application described by the previous parameters.

We have:

$$\lambda = \sum_{s=1}^{N_s} \lambda_s \tag{3.1}$$

Also:

$$\lambda_s = \frac{\sum\limits_{m \in M} \lambda_{m,s} * N_{m,s} * T_{m,s}}{\sum\limits_{m \in M} N_{m,s} * T_{m,s}} \tag{3.2}$$

And:

$$\lambda_{m,s} = \frac{\sum\limits_{p \in P} \lambda_{p,m} * T_{p,m}}{\sum\limits_{p \in P} T_{p,m}} \tag{3.3}$$

Where $\lambda_{p,m,s} = \frac{1}{IA}$ and IA is the average packet inter-arrival in period p of pattern m.

The value of $\lambda_{p,m}$ can not be determined in the case of TCP based traffic sources, as it is function of network congestion. It is replaced by the nominal value of access terminal packet rate. The duration of a TCP period is given by:

$$T_{p,m,s} = \frac{Q_{p,m,s}}{P_{p,m,s} \star \lambda_{p,m,s}} \tag{3.4}$$

The packet rate during period p depends on its type, OFF periods have null value for packet rate:

$$\lambda_{p,m,s} = \begin{cases} \lambda_{ON} & ON \\ 0 & OFF \end{cases} \tag{3.5}$$

Using the previous formulas we can give an estimation of the average packet rate produced by a multimedia application described by the framework. The estimated packet rate is considered as a characteristic of the multimedia application model and it is saved with its description for later use.

3.5 Simulation and Statistical Tools

Models described by the framework are used with Distributed Hybrid Simulator (DHS) [GGB+01] developed in LAAS-CNRS. DHS is an IP-MPLS network simulator, based on differential traffic theory and hybrid simulation. DHS can simulate continuous models combined with discrete event-driven models. In fact, the same core can simulate a complete network in analytical mode, event-driven mode or hybrid mode. In the last mode some flows are continuous while others are event driven.

Traffic resulting from models described with the framework is generated in the event driven mode and used in any network topology for performance evaluation ends. Although, application models are simulated generally in the event driven mode, they can also be combined with other analytical existing models via Hybrid simulation. It makes it

possible to evaluate the performance of a new multimedia application when multiplexed with thousands of analytical flows for example, very quickly.

However, estimating traffic models and characterizing superposed and trace traffic requires the development of many statistical tools. We are going to present these tools along with fundamental definitions that will guide our characterization and estimation study throughout this thesis. Finally, we note that all of these tools were implemented in a statistical module appended to the TSM tool.

3.5.1 Definitions

In this section we will give the definition of self-similarity, short range dependence (SRD), and long range dependence (LRD) as major properties observed in Internet traffic. Besides, many traffic models rely on long range dependent processes and the Hurst exponent [Hur51]. In all the following definitions we consider stationary processes.

Definition 1: LRD Process

A discrete stationary process $\{X_l, \quad l \in \mathrm{N}\}$ with autocorrelation function $\rho(k)$ is said LRD if the following relation holds:

$$\sum_k \rho(k) = \infty \tag{3.6}$$

A nonsummable autocorrelation function is synonym of correlations on large time scales.

Definition 2: SRD Process

A discrete stationary process $\{X_l, \quad l \in \mathrm{N}\}$ with autocorrelation function $\rho(k)$ is said SRD if the following relation holds:

$$\sum_k \rho(k) < \infty \tag{3.7}$$

Definition 3: Self-Similarity

A continuous process $(Y(t))_{t\geq 0}$ is said exactly self-similar of Hurst exponent $H(0 < H < 1)$ if $\forall c > 0$:

$$Y(t) \stackrel{L}{=} c^{-H} Y(ct) \tag{3.8}$$

Assuming that $\stackrel{L}{=}$ stands for finite dimension distribution equality. Self-similarity induces similar behaviour of the process on different time-scales.

Definition 4: Discrete self-similarity

Let $\{X_l, \quad l \in \mathrm{N}\}$ be a self-similar process. We denote $(X_k^m)_{k\in\mathrm{N}}$ its aggregated process of order m defined as follows:

$$X_k^{(m)} = \frac{1}{m}(X_{(k-1)m+1} + ... + X_{km}) \tag{3.9}$$

The $(X_l)_{l\geq 0}$ process is said self-similar of H exponent if $\forall m$ integer:

$$(X_k^{(m)} = m^{1-H}(X_{(k-1)m+1} + ... + X_{km}))_{k\in\mathrm{N}} \stackrel{L}{=} (X_l)_{l\in\mathrm{N}} \tag{3.10}$$

When a process $\{X_l, \quad l \in \mathrm{N}\}$ is self-similar of Hurst parameter $\frac{1}{2} < H < 1$, the autocorrelation function of the process $\rho_X(k)$ and the autocorrelation function of its aggregated process $\rho_{X^{(m)}}(k)$ are related as follows:

$$\rho_{X^{(m)}}(k) = \rho_X(k) = \frac{1}{2}\left\{(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right\}, \quad \forall m, k \geq 0 \tag{3.11}$$

The last equality is difficult to verify in general, that is why the notion of second-order self-similarity is introduced.

Definition 5: Second-order self-similarity

A stationary process $\{X_l, \quad l \in \mathrm{N}\}$ is said self-similar of second order if $\forall m$ integer:

$$\rho_{X^{(m)}}(k) = \rho_X(k) \tag{3.12}$$

In time series, second order self-similarity describes the property that the correlation structure (ACF) of a time-series is preserved irrespective of time aggregation. Second-order self-similar processes are extensively used to model long-range dependent processes. While the concepts of self-similarity and long range dependence are often used interchangeably in the literature, they are not equivalent. Although second-order self-similarity usually implies long range dependence (i.e nonsummable ACF), the reverse is not necessarily true. In addition not all self-similar processes are long-range dependent (e.g, Brownian motion).

Definition 6: Asymptotic self-similarity

A stationary process $\{X_l, \quad l \in \mathrm{N}\}$ is said asymptotically self-similar if:

$$\lim_{k\to\infty} \rho_{X^{(m)}}(k) = \frac{1}{2}\left\{(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right\} \tag{3.13}$$

The notion of asymptotic self similarity is used with some traffic models that can not produce exact self-similar paths.

3.5.2 Analysis Tools

Traffic analysis tools are numerous, the most simple ones are moment estimators (average, variance, skewness, ...) as well as empirical probability distribution functions (Density PDF and Cumulated CDF). However, based on the definitions given in the previous section, it is clear that the most important is to characterize the correlation nature that may exist in the traffic (correlation between packet inter-arrivals primarily). Correlation structure may be memoryless, short range or long range. Naturally, the autocorrelation function is the best tool to characterize the correlation structure of a time series. However, it is well known that the LRD property can also be characterized by the Hurst exponent. Besides, we will use the Index of Dispersion of Inter-arrivals (IDI) as it better reflects the influence of cumulated covariance in time series. Assuming that probability distribution functions and moment estimators are very known, we present only the Hurst exponent and the IDI index.

3.5.2.1 Hurst Exponent

One robust statistic for describing long range correlations in time series was invented by Hurst [Hur51] to describe reservoir capacity and later generalized by Mandelbrot [MW69]. The analogy between water reservoirs, where water comes in at irregular intervals, but leaves at a constant rate is analogous to a queuing system where packets arrive at irregular intervals and are processed at the rate of the server. An analysis known as "Rescaled-Range Analysis," is used to study long range correlations. Range is defined as the span of maximum and minimum fluctuations about the mean over some time window. As defined before, the range R is expressed in terms of the average influx of a quantity x over a time period τ:

$$E(x)_\tau = \frac{1}{\tau}\sum_{t=1}^{\tau} x(t) \tag{3.14}$$

Let $X(t,\tau)$ be the accumulated quantity as:

$$X(t,\tau) = \sum_{u=1}^{t} (x(u) - E(x)_\tau) \tag{3.15}$$

Then the range R over the time period τ is defined as

$$R(\tau) = \max_{1\le t\le\tau} X(t,\tau) - \min_{1\le t\le\tau} X(t,\tau) \tag{3.16}$$

Let S be the standard deviation, which scales the range and is defined as:

$$S(\tau) = \sqrt{\frac{1}{\tau}\sum_{t=1}^{\tau} (x(t) - E(x)_\tau)^2} \tag{3.17}$$

The ratio R/S is very significant because, Hurst has proved the following relationship between time period and the ratio R/S (a is a constant):

$$R/S = (a\tau)^H \tag{3.18}$$

The Hurst exponent, H, measures self-similarity of the incoming events. Values of $H > 0.5$ indicate long range correlations and values of $H < 0.5$ indicate anti-correlations. When $H = 0.5$ there are no correlations, that is, the sequence of events is completely random. The value of H could be estimated by the least mean squares or by the linear regression log / log of R/S on τ.

Some studies underline the fact that some non-stationary processes can behave themselves in ways similar to LRD processes. A basic test for stationary consists of splitting the dataset into two halves, and estimating the Hurst exponent independently for each half. If the two estimates yield comparable values we can say that the process is stationary, otherwise H is varying with time and the process is non-stationary.

There are several other methods to measure the Hurst exponent of a time series. The Variance Time Plot and the Wavelette estimator are among the widely used ones. Indeed,

the evaluation of this parameter is a research subject itself. Some studies mention that the Hurst exponent can fluctuate according to the time scale chosen for its evaluation. We recommend the study published in [STP+04] for a complete discussion about the subject. However, we will be limited to the R/S method as it gives one of the best average estimations of this parameter.

3.5.2.2 Index of Dispersion of Inter-arrivals

The Index of Dispersion of Inter-arrivals (IDI) [CL66] for a sequence of inter-arrivals $\{T_k, k \geq 1\}$ assumed to be stationary is the k-squared coefficient of variation $\{c_k, k \geq 1\}$ defined by:

$$c_k^2 = \frac{k.Var(S_k)}{[E(S_k)]^2} \tag{3.19}$$

With $S_k = T_1 + T_2 + ... + T_k$ denotes the sum of k consecutive inter-arrivals. Indeed, c_k^2 measures the cumulative covariance normalized by the square of the mean among k consecutive inter-arrivals.

The IDI is a simple measure of traffic burstiness and more reliable than the Hurst exponent, from a performance evaluation point of view. Indeed, the Hurst exponent estimation is achieved on the number of packets (or bits) entering the network during a constant slot of time. As a consequence, the covariance that may exist between individual packets arrivals are lost and can not be taken into consideration. Indeed, the cumulated covariance of packet inter-arrivals have dramatic consequences on queue performance when its value is greater than 1 (Poisson arrivals have $c_k^2 = 1, \forall k \geq 1$) as we will see in **Chapter 5**.

3.5.3 Estimation Tools

Traffic models require estimating traffic parameters. In most cases it concerns distribution fitting problems to determine the distribution of packet sizes, packet inter-arrivals, data slots sizes etc... Besides, basic distributions (e.g. exponential, normal, lognormal, ...) some parameters are better fitted by a mixture of distributions (especially quantities dealing with sizes in bytes). We suggest estimating the probability density function by mixture distribution $y(x)$ defined as follows:

$$y(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \cdots + \alpha_n f_n(x) \tag{3.20}$$

Where $\{\alpha_i, i = 1 \cdots n\}$ are positive weights such that $\sum_{i=1}^{n} \alpha_i = 1$. We use the EM (Expectation-Maximization) algorithm to estimate distribution parameters by the maximum likelihood approach [DLR77].

On the other hand, function fitting problems useful to estimate the Autocorrelation function are better solved using non-linear regression methods. We use the Levenberg-Marquardt algorithm to estimate the correlation structure of traffic traces.

3.5.3.1 EM Algorithm

The EM algorithm provides an efficient method to estimate a parametric model of parameter α using an incomplete observation data set y. The algorithm uses a hidden variable h that it is supposed to provide a complete data set (y, h) with the incomplete observation data set y. In other words, we suppose the existence of a joint distribution $p(y, h|\alpha)$ that is simpler to estimate. The idea of EM is to estimate a maximum likelihood of the complete data set $p(y, h|\alpha')$ using the current estimation of α'. This maximum likelihood will be used to estimate the new parameter α. Formally, this could be expressed as:

$$Q(\alpha, \alpha') = E_h(\log p(y, h|\alpha)|y, \alpha') \tag{3.21}$$

When estimating a mixture distribution as defined in equation (3.20). The problem is formulated as follows:

We have n known distributions $f_1(x), f_2(x), \cdots, f_n(x)$ with an incomplete observation data set $x = x_1, x_2, \cdots, x_T$, and:

$$\left\{ f_\alpha(x) = \sum_{i=1}^{n} \alpha_i f_i(x) \; with \; \alpha_i \in [0, 1] \; and \; \sum_{i=1}^{n} \alpha_i = 1 \right\} \tag{3.22}$$

We estimate $\hat{\alpha} = \arg\max_\alpha \log f_\alpha(y)$

The hidden variable in this case is the state $s\,(s \in [1, n])$ of the model f_α at the prediction instant. The auxiliary function is expressed as follows:

$$Q(\alpha, \alpha') = \sum_y \tilde{f}(y) \sum_{i=1}^{n} f_{\alpha'}(s = i|y) \log f_\alpha(y, s = i) \tag{3.23}$$

$\tilde{f}$ is the empirical distribution and:

- α_i is the initial probability of being in state i,
- $f_\alpha(y, s = i) = \alpha_i \times f_i(y)$ is the joint probability of being in state i and generate y,
- and $f_{\alpha'}(s = i|y) = \frac{\alpha_i f_i(y)}{\sum_i \alpha_i f_i(y)}$ is the probability of being in state i knowing that y is the current observation.

EM tries to maximize the auxiliary function (on α) under the constraint: the sum of coefficients equals to 1. Using a normalisation factor λ we can write:

$$\frac{\delta}{\delta \alpha_i} \left[Q(\alpha, \alpha') - \lambda \left(\sum_i \alpha_i - 1 \right) \right] = \underbrace{\frac{1}{\alpha_i} \sum_\alpha \tilde{f}(y) f_{\alpha'}(s = i|y) - \lambda}_{C_i} = 0 \tag{3.24}$$

And then $\alpha_i = \frac{C_i}{\sum_i C_i}$. The EM algorithm in this case can be summarized as follows:

- Initialize α'_i to any values under the condition $\sum_{i=1}^{n} \alpha'_i = 1$

- Repeat until convergence:
  - Expectation-step: Evaluation of C_i.
  - Maximization-step: for each i, $\alpha_i \leftarrow \frac{C_i}{\sum_i C_i}$.

C_i could be seen as the expected number of times the model i is used to generate the observation. It is very intuitive! We should note that, the maximization step depends on the distribution type. Solutions for most used probability density functions can be found in [Vil00].

We use the EM algorithm to estimate the different parameters and weights of different distribution mixtures suggested to fit traffic model parameters. The main advantage of the EM algorithm is that it converges towards a local maximum starting from any initial point. On the other hand, its convergence speed may be very slow. We use an accuracy criterion to stop the iterations of the algorithm. The distribution fit results are generally very positive.

3.5.3.2 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm [Lev44, Mar63] is used to solve numerically the minimization problem in least squares curve fitting. In many cases the Levenberg-Marquardt algorithm can find a solution even when it is initialized far from the final minimum at the cost of slower convergence than traditional algorithms (e.g Gauss-Newton).

Let g be a function of two vectors x and P. The sum of the squares of the deviations $f(P)$ could be expressed like this:

$$f(P) = \left\langle (g(x, P) - y)^2 \right\rangle \tag{3.25}$$

Where $\langle . \rangle$ stands for the calculated average over a set of couples (x, y). The algorithm is iterative and seeks to find the vector P that minimizes the function $f(P)$ based on a set of real measured values y.

The algorithm seeks to calculate the vector P_i as a function of vector P_{i-1} at iteration i, so that $f(P_i)$ tends to a local minimum of f. Indeed, a quadratic approximation $\hat{f}$ of f is calculated based on a linear approximation $\hat{g}$ of g around the point P_{i-1}. This approximation is not efficient unless function g is really linear around point P_{i-1}. Otherwise, very bad results are obtained. This motivated Levenberg to use this approximation only in the regions where g is quasi linear, otherwise a gradient descent is used instead. Later Marquardt work focused on quickly switching to gradient descent to avoid a big number of iterations in linear regions. This combined algorithm is known as Levenberg-Marquardt algorithm. The algorithm implementation details can be found in [EM78] where more technical information are provided.

Practically, this algorithm converges with small number of iterations. However, for each iteration the number of operations is proportional to N^3, where N is the size of vector P. As a consequence, this algorithm is usually limited to problems with small number of parameters to optimize.

This algorithm will be used to estimate the correlation structure of traffic traces based on autocorrelation function models for short range and long range dependent correlations where the number of parameters to optimize is very limited (maximum three).

3.6 Conclusion

In this chapter we presented a generic and hierarchical framework for traffic modelling, based on the generalization of the ON-OFF behavioural model. The model can describe most of the applications deployed on the Internet throughout a detailed description of the application behaviour. Its generic structure allows modelling applications of higher complexity with several parallel flows. We implemented this framework in a simulation tool for performance evaluation studies. Throughout this thesis we use the framework to describe multimedia applications: Audio, Video and Web. A complementary analysis and estimation module was implemented. It is used to analyze statistically traffic sources and to estimate traffic model parameters. The framework offers a complete workbench for estimating, modelling and evaluating multimedia traffic sources.

Chapter 4

IP Traffic Modelling

4.1 Introduction

Since Paxson and Floyd [PF95], reported the failure of Poisson process in modelling Wide Area Network (WAN) traffic, long-range dependence of Internet traffic was widely revealed (e.g. [LTWD94]). Many approaches were explored to model Internet traffic. We can classify these different approaches into two broad categories: Flow level approaches and Packet level approaches. Flow level approach focuses on traffic modelling at the connection level. In this case, we group IP packets into connections or into more generic "flows" and we model the flow arrival process and the fluctuation of active flows. On the other hand, Packet level approach has two different trends: point processes and aggregate count processes. In point process, traffic may be seen as single arrivals of discrete entities (packets, cells, etc). It can be mathematically described as a point process [Çin75], consisting of a sequence of arrival instants $t_1, t_2, \ldots, t_n$. There are two equivalent descriptions of point processes: counting processes and inter-arrival time processes. A counting process $N(t)$ is a continuous-time, non-negative integer-valued stochastic process, where $N(t) = \max\{n, t_n \leq t\}$ is the number of (traffic) arrivals in the interval $\{0, t\}$. An inter-arrival time process is a non-negative random sequence $A(n)$, where $A(n) = t_n - t_{n-1}$ is the length of the time interval separating the n_{th} arrival from the previous one. However, given the huge number of packets involved in any network traffic, this would result in huge data sets. A a consequence, aggregate count process, denoted $X_\Delta(k)$ is generally preferred. The aggregate count process $X_\Delta(k)$ consists of the number of packets (or bytes) lying within the k_{th} slot of size $\Delta > 0$ and whose time stamps lie between $k\Delta \leq t_i < (k+1)\Delta$.

A recurrent theme relating to traffic modelling in broadband networks is the traffic "burstiness". Burstiness is present in a traffic process if the arrival points $\{t_n\}$ appear to form visual clusters; that is, $\{A_n\}$ tends to give rise to runs of several relatively short inter-arrival times followed by a relatively long one. The main sources of burstiness are due to the shapes of the probability distribution and autocorrelation function of $\{A_n\}$. There is no single widely-accepted measure of burstiness. Some of the commonly-used mathematical measures are: the ratio of peak rate to mean rate, the Index of Dispersion

Of Inter-arrivals (IDI), and the Hurst exponent (see **Chapter 3** for definitions).

In this chapter, we model IP traffic by aggregate byte count process (a window-based approach). We construct traffic models for some Internet traffic traces and we evaluate the generated traffic both statistically and in a network environment. In particular, we analyze the poor performance of the generated traffic when injected in queuing system. Then we introduce proposals to enhance the packet generation process. The proposed modifications are valid also for similar traffic models based on the aggregate byte count process approach.

4.2 Internet Traffic Modelling

Long range correlations in Internet traffic put into evidence the importance of capturing the correlation structure and probability distributions of traffic components in order to obtain reliable models. A wide range of mathematical models were developed to generate Internet traffic including fractional Gaussian noise [Pax97], autoregressive processes [LTWD94], chaotic maps [ESP94], fractional autoregressive integrated moving average process [AM95], random midpoint displacement [LEWW95], spatial renewal process [TDL98], wavelet transformation [Fla92] and others ...

Those algorithms differ in their computational complexity and implementation flexibility. We focus on models based on aggregate byte count processes, where $X_\Delta(k)$ denotes the byte count entering the network by unit of time (generally called *slot*). Hence, a good model must capture well the autocorrelations of $X_\Delta(k)$ as well as the probability distributions associated with $X_\Delta(k)$ and packet sizes.

In the following sections we give a brief overview of Fractional Gaussian Noise (FGN) and Fractional Autoregressive Integrated Moving Average (FARIMA) process as well as an empirical method based on the aggregation of heavy-tailed ON-OFF processes. Then we present the $M/G/\infty$ input process to model IP traffic, and we focus on its versatile properties and flexibility.

4.2.1 Fractional Gaussian Noise

Fractional Gaussian Noise (FGN) is a stochastic process that is the formal derivative of a fractional Brownian motion. FGN is gaussian process that is indexed by the Hurst parameter H, for which an LRD property is observed for H values belonging to $[0.5, 1]$. The algorithm of generating FGN process is based on Paxson work [Pax97]. The idea is to form complex numbers based on an approximation of the spectral function characterizing the FGN process and then use the Inverse Fourrier Transform to generate samples having the same spectral density. The particularity of the FGN process is that it represents data volumes (in bytes or in packets) by a normal law $N(\mu, \sigma^2)$.

The inputs of the model are H, the desired Hurst exponent, and n, the desired (even) number of observations in the synthesized sample path. The FGN algorithm is quite fast and allows quick data generation (complexity is of $n \log n$ order for n samples). The data samples generated represent very well a self-similar process. On the other

hand, the use of the Inverse Fourrier Transform makes the algorithm dependent of the samples number n to generate, and the implementation can not be independent of time. Besides, the data size distribution is fixed (Normal law), and the correlation function is fixed also $\rho(k) \approx_{k\to\infty} H(2H-1)k^{2H-2}$. Indeed, the FGN model can capture only the long range correlations while it is often matter to estimate both short range and long range correlation coexisting in traffic. Those limitations make the FGN incapable of representing different types of correlation or different types of data sizes distributions that may be present in the IP traffic.

4.2.2 Fractional Autoregressive Integrated Moving Average process

Fractional Autoregressive Integrated Moving Average (FARIMA) processes are another class of models used to model correlations in Internet traffic [Ber94]. An $farima(P,d,Q)$ process $\{X(k), \quad k \in \mathrm{Z}\}$ is defined via two polynomials of order P and Q and a fractional integration D^{-d}, of order $-\frac{1}{2} < d < \frac{1}{2}$, as:

$$X(k) = \sum_{p=1}^{P} \phi_p X(k-p) + D^{-d}(\varepsilon(k) - \sum_{q=1}^{Q} \theta_q \varepsilon(k-q)) \tag{4.1}$$

Where the $\varepsilon(k)$ are independent, identically distributed random variables, referred to as innovations, with zero mean and σ_ε standard deviation. The fractional integration is given by:

$$D^{-d} = \sum_{i=0}^{\infty} b_i(-d)B^i \tag{4.2}$$

Where B is the backward operator $B\varepsilon(i) = \varepsilon(i-1)$, and $b_i(-d) = \frac{\Gamma(i+d)}{\Gamma(d)\Gamma(i+1)}$. It is proved that for $d \in (0, 0.5)$ the process is LRD. In this process the first term contribution accounts for short range correlations and the fractional integration of order d accounts for long range correlations.

Garrett and Willinger in [GW94] discuss an algorithm for generating sample paths from a fractional ARIMA process, which are asymptotically self-similar. The algorithm has the very attractive property of being exact, but its running time is $O(n^2)$ for generating n points, which is quite slow.

4.2.3 Empirical ON-OFF Aggregation

Consider an alternating renewal process $R(t)$ in which the ON and OFF periods have durations from a "heavy-tailed" (e.g. Pareto) distribution. Let S_n be the process constructed by multiplexing n independent instances of the $R(t)$ process, where $S_n(t)$ is the number of $R(t)$ processes that are in ON periods at time t. Then S_n is asymptotically (as n approaches ∞) a self-similar process. This method has been proved theoretically by Taqqu, Willinger and Sherman [TWS97].

Let $1 < \alpha_1 < 2$ (resp. $1 < \alpha_2 < 2$) be the parameter associated with the heavy tail of the ON period (resp. the OFF period). It has been proved that the superposition of the corresponding homogeneous ON-OFF process result in a Fractional Gaussian Noise of H exponent given by:

$$H = \frac{3 - \alpha_{\min}}{2}, \quad \alpha_{\min} = \min(\alpha_1, \alpha_2) \tag{4.3}$$

The most common example is to take ON-OFF process with $Pareto(\lambda, \beta)$ laws for ON and OFF periods. In this case we obtain self-similar traffic of $H = \frac{3-\beta_{\min}}{2}$, $\beta_{\min} = \min(\beta_1, \beta_2)$.

The main difficulty when using a simulation of S_n for synthesizing a self-similar process, is that one must trade off speed of computation (low n) against the degree of agreement with a true self-similar process (asymptotically high n). In [TWS97] the approximate self-similar traces were obtained with $n \geq 500$. Another ambiguous point is the choice of heavy-tailed parameters. For example in the case of $Pareto(\lambda, \beta)$ distribution, the simulated Hurst exponent depends only on the β parameter associated with the queue tail while the Pareto law depends also on the λ parameter associated with the minimum value taken by the distribution. In this method, the choice of this parameter is left for the user in function of the quality of simulated self-similar traffic! However, this model is important for the study of superposed ON-OFF processes as we will see in **Chapter 5**.

4.2.4 $M/G/\infty$ Input Process

The $M/G/\infty$ input model represents the occupation process of a discrete-time queue in stationary regime. The clients (bits, bytes, packets or frames in the video flow context) arrive according to a Poisson process and are served (by infinite number of servers) with the general service distribution G.

Let σ be N-valued random variable distributed according to the probability function G. We denote X_n, for $n = 0, 1, 2, ...$, the number of clients at the beginning of time slot [n, n+1[and by $\sigma_{n,i}$ the service duration for the i_{th} client in the system at time n. The busy server process $(X_n)_{n\geq 0}$ is correlated but in general is not a stationary process. To start the process in the stationary regime, we must choose the initial parameters as follows:

- X_0 the number of clients in the system at time $n = 0$, is distributed according a Poisson law with parameter $\lambda E(\sigma)$.
- $\forall i \geq 1$, the random variables $\sigma_{0,i}$ are independent and identically distributed with the probability function:

$$P(\sigma_{0,i} = k) = \frac{P(\sigma \geq k)}{E(\sigma)} \tag{4.4}$$

In fact, with these initial conditions, the occupation process $(X_n)_{n\geq 0}$ verifies the following properties:

- $\forall n \geq 0$, the random variables X_n are distributed according a Poisson law with parameter $\lambda E(\sigma)$.
- The correlation structure associated with $(X_n)_{n\geq 0}$ process is completely determined by σ. Indeed, the autocorrelation function is given by:

$$\rho(k) = \frac{P(\sigma \geq k)}{E(\sigma)} \tag{4.5}$$

And

$$\sum_{k=0}^{+\infty} \rho(k) = \frac{1}{2} + \frac{E(\sigma^2)}{2E(\sigma)} \tag{4.6}$$

The process $(X_n)_{n\geq 0}$ can produce SRD and LRD behaviours depending on the value of $E(\sigma^2)$. If $E(\sigma^2) < \infty$ (resp. $E(\sigma^2) = \infty$) the process is SRD (resp. LRD).

Conversely, the choice of a correlation structure allows characterizing the service time distribution as the following equation is verified by the $M/G/\infty$ queuing system:

$$P(\sigma = k) = \frac{\rho(k+1) - 2\rho(k) + \rho(k-1)}{1 - \rho(1)} \tag{4.7}$$

This last property is particularly interesting. Indeed, from the autocorrelation function of a time series (data slots), we can define the service distribution G.

In fact, the probability distribution of X_n is Poisson with parameter $\lambda_p = \lambda E(\sigma)$. However, to capture distribution of a real sequence, the Poisson distribution must be transformed into the estimated slot size distribution (recall that although slot durations are constant the corresponding data sizes are variable). When developing our traffic model an explicit example of this transformation will be detailed. However, numerical experimentations show that this non-linear transformation has a negligible impact on the original correlation structure.

The input $M/G/\infty$ model associated with a time series is thus characterized by the following parameters:

- Time scale or time slot.
- Slot size distribution.
- Autocorrelation function.
- Packet size distribution

Some statistical studies (e.g. [KM98]) suggest the following empirical autocorrelation function models:

- $\rho_0(k) = e^{-b*k}$ Markov
- $\rho_1(k) = e^{-b*\sqrt{k}}$ SRD
- $\rho_2(k) = (k+1)^{-b}$ LRD

Where k is the lag between data slots and b a positive real parameter. We generalize the autococorrelation models by using mixture correlation functions (see next section).

Let X_t be the number of customers in the system at time t, it is proved that $\{X_t\}$ is asymptotically self-similar. The big advantage of this model is in its flexibility. Any type of correlations SRD or LRD could be reproduced and any data slot size distribution could be represented easily.

The $M/G/\infty$ process model has a medium mathematical and computational complexity. It can be used to model correlations in aggregated traffic as well as in single traffic (e.g. video). We use the $M/G/\infty$ process to model correlated traffic throughout this thesis.

4.3 Modelling IP Traffic Using M/G/∞ Process

We use an aggregate byte count process based on the $M/G/\infty$ input process to model IP traffic. Traffic models of two Internet traffic traces are derived and estimated. Generated traffic is validated statistically and by simulation in network environment.

4.3.1 Traffic Trace Presentation

We realized our study on two Internet traffic traces: a) the first trace was captured on an OC-48 link from CAIDA project [CAI03]. It was collected on CAIDA monitor located at a SONET OC48 (2.5 Gbps) link that belongs to MFN, a US Tier 1 Internet Service Provider (ISP). b) The second trace was collected on the WIDE link held by the MAWI (Working Group Traffic Archive) for the WIDE project [MAW06]. The WIDE backbone link is a trans-Pacific 100Mbps link. The description of both traces is shown in Table **4.1**.

Table 4.1: Statistics of Traffic Traces (CAIDA and MAWI)

Statistics	**CAIDA**	**MAWI**
	24-4-2003 (15:55-16:00)	9-1-2006(14:00-14:15)
Bytes	3436.4M	2862.16M
Packets	7M	7.2M
Mean Rate	91.64Mbps	26.7Mbps
TCP Packets	91%	78.18%
UDP Packets	8%	13.01%

Figures **4.1** and **4.2** illustrate the percentage of protocol bytes in each trace

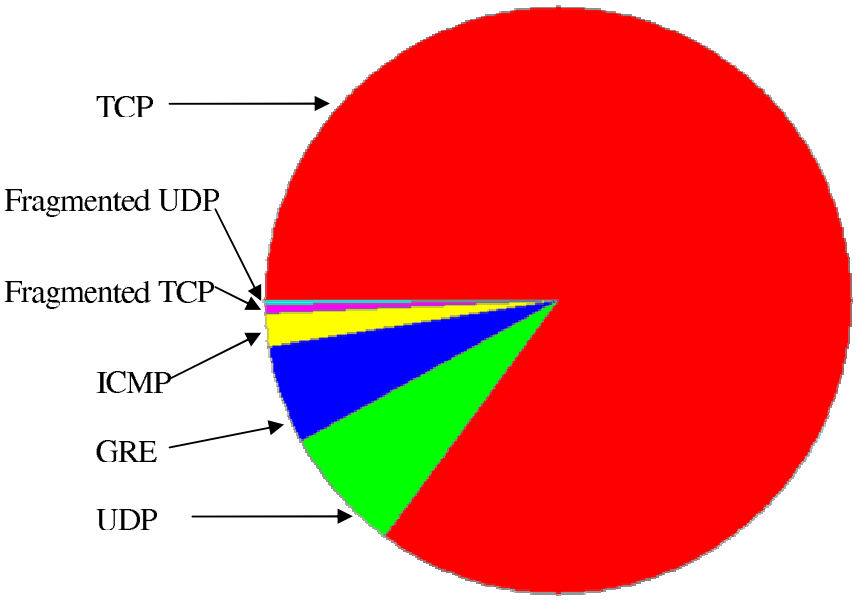


Figure 4.1: MAWI Trace Byte Per Protocol Pie Chart

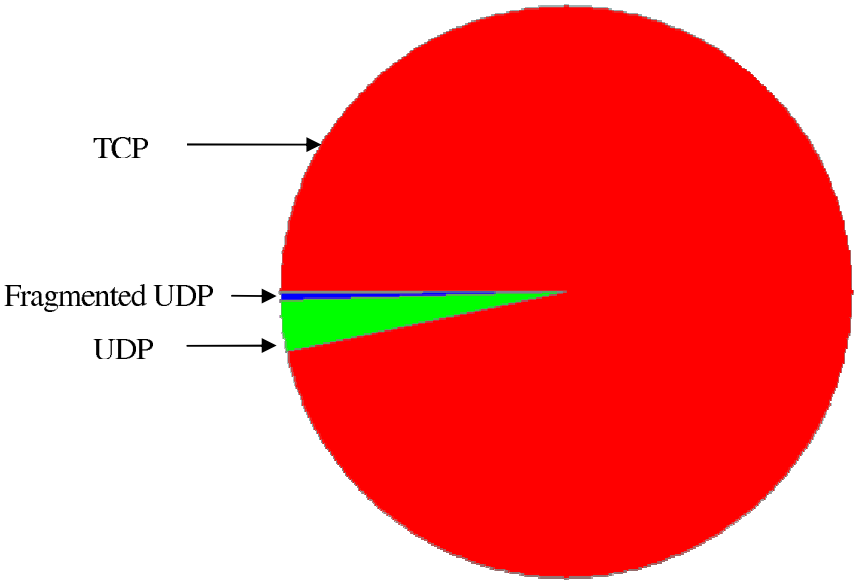


Figure 4.2: CAIDA Trace Byte Per Protocol Pie Chart

We estimate the traffic model parameters on a time series representing the amount of bytes entering the network per unit of time (slot). Of course, the same approach is applicable using packet counts instead of byte counts (in which the number of packets in a slot is estimated rather than the byte counts). However, with variable packet sizes the aggregate byte count process is more suitable to achieve accurate estimation of the correlation structure.

4.3.2 Traffic Model

Using the $M/G/\infty$ process we target a general case in which correlations may be both SRD and LRD and slot size distribution is a mixture distribution. Thus, we introduce a mixed correlation function $\rho(k)$ defined as

$$\rho(k) = \alpha.\rho_1(b_1, k) + (1 - \alpha).\rho_2(b_2, k) \tag{4.8}$$

$\rho_1(k) = e^{-b_1 * \sqrt{k}}$ represents the SRD component and $\rho_2(k) = (k+1)^{-b_2}$ represents the LRD component. b_1 and b_2 are positive numbers and α is a weighing factor.

Experimentations show that the slot size distribution is most adequately captured by a mixture distribution. Particularly, the Lognormal-Gamma mixture distribution is very suitable to Internet traces. Thus, the Probability Distribution Function (PDF) of slot sizes could be expressed as:

$$f(x) = p * \log n(\mu, \sigma) + (1 - p) * gamma(\alpha, \beta) \tag{4.9}$$

Recall that $M/G/\infty$ process generates Poisson distributed slot sizes. Hence, we need to achieve the transformation from Poisson distribution to Lognormal-Gamma mixture distribution.

Let $F(x)$ be the Cumulative Distribution Function (CDF) of Lognormal-Gamma distribution, we seek the function G that verifies:

$$F(x) = G(F_{pois}(x)) \tag{4.10}$$

Where F_{pois} is the CDF of the Poisson distribution with parameter $\lambda_{pois} = \lambda E(\sigma)$. The Lognormal-Gamma distribution is defined by the following PDF:

$$f(x) = p * f_1(x) + (1 - p) * f_2(x) \tag{4.11}$$

With p the weight parameter and $f_1(x)$ (resp. $f_2(x)$) the PDF of the Lognormal distribution (resp. Gamma distribution).

Let U be a uniform random variable within the interval[0 1]. Then, the function $G(x)$ is calculated with the following algorithm:

$$
\begin{array}{l}
If\ p < U\ Then \\
G(x) = F_1^{-1}(x) \\
Else \\
G(x) = F_2^{-1}(x)
\end{array}
$$

Where:
$F_1^{-1}(x) = e^{\delta\Phi^{-1}(x)+\mu}$ is the inverse Lognormal CDF.
And:
$F_2^{-1}(x) = \frac{1}{\lambda}\Gamma^{-1}(a, \Gamma(a,0)(1-x))$is the inverse Gamma CDF.
With: $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$ and $\Gamma(a,x) = \int_{x}^{+\infty} e^{-t} t^{a-1} dt$.

4.3.3 Model Parameters Estimation

The aggregate count process approach introduces an important parameter which is the aggregation level. It concerns the duration of the slot on which byte counts are realized. To examine the influence of this parameter we evaluate the contribution of each component of the mixed correlation function (SRD and LRD) at different byte aggregation levels. This is expressed by the value of parameter α of equation (4.8). Results are shown in Table **4.2**.

Table 4.2: SRD and LRD Correlation Versus Aggregation Level

Slot (ms)	α **(CAIDA)**	α **(MAWI)**
0.1	0.92	0.99
1	0.85	0.96
10	0.63	0.87
100	0.14	0.36

We can clearly see that the LRD component depends on the slot duration. Thus, for long slot durations we observe stronger LRD component. Moreover, we see a clear difference between the two traces. The CAIDA trace has a higher overall aggregation level than the MAWI trace as it concerns data collected on 2.5 Gbps link versus only 100 Mbps link for the MAWI trace. As a consequence, for the same slot duration we have less important SRD component in the CAIDA trace than in the MAWI trace (e.g. for 1 ms slot 0.85% of correlation is SRD in the CAIDA trace versus 0.96% SRD in the MAWI trace). In fact, the slot duration influences directly the correlation observed on slot sizes (or data entering network per time slot). However, with a mixture autocorrelation function the model estimates well the correlation structure regardless the time slot size.

The aggregate byte counts approach models slots not packets. However, slot sizes must be transformed into packets in order to be simulated in a network. During this transformation, packet sizes must respect the probability distribution of packets in the trace. We note that packet inter-arrival distribution inside slots could not be specified,

as it changes from one slot to another. That is why it is generally assumed that packet inter-arrivals during slots are uniform or exponential.

We have estimated the model parameters for both traces using a slot of 1 ms for the CAIDA trace and a slot of 10 ms for the MAWI trace. Parameters are shown in Table **4.3** (refer to equations (4.8) and (4.9)). Packet size distribution is estimated by an empirical discrete distribution. Table **4.3** shows also the mean and variance of the empirical packet size distribution for both traces.

Table 4.3: Traffic Models Parameters

Trace	**Correlations**			**Data Slot Sizes**					**Packet Size Distribution**	
	α	b_1	b_2	p	μ	σ	α	β	Mean	Variance
CAIDA (Slot=1ms)	0.85	2.95	0.58	0.378	9.42	0.36	3.54	2.9212E+3	512.3	3.987419E+5
MAWI (Slot=10ms)	0.87	13.7	0.52	0.144	10.33	0.2	9.27	3.6608E+3	405.3	3.404803E+5

We construct traffic models based on estimated parameters. Notice that we estimate packet size distribution over the whole trace. Furthermore, we use an exponential distribution for packet inter-arrivals inside slots. The corresponding traffic generators are implemented and are used to validate the generated traffic against traffic traces.

4.3.4 Validation

4.3.4.1 Statistical Validation

The goal of statistical validation is to make sure that the generated traffic slots present the same statistical properties as the trace traffic slots. Figure **4.3** shows the CDF of slot sizes for both trace and model (for both traces). The results are excellent in terms of fitting as the model reproduces correctly the probability distribution of slot sizes.

We compare also the packet size distribution for both trace and generated traffic. Figure **4.4** shows the CDF of packet sizes for the CAIDA trace. The packet size distribution over the whole trace is very well respected (A good fitting was also obtained in the case of MAWI trace).

Finally, we compare the correlation structure of both trace and generated traffic. Figure **4.5** shows the autocorrelation function of slot sizes for the CAIDA trace (similar results were obtained in the case of MAWI trace). The autocorrelation function is well captured by the mixed (SRD, LRD) autocorrelation function model. The generated traffic presents similar correlations as in traffic trace.

The traffic model captures very well the statistical properties of the traffic trace. However, the traffic generated by the model should be evaluated in a network environment to measure its performance versus traffic trace.

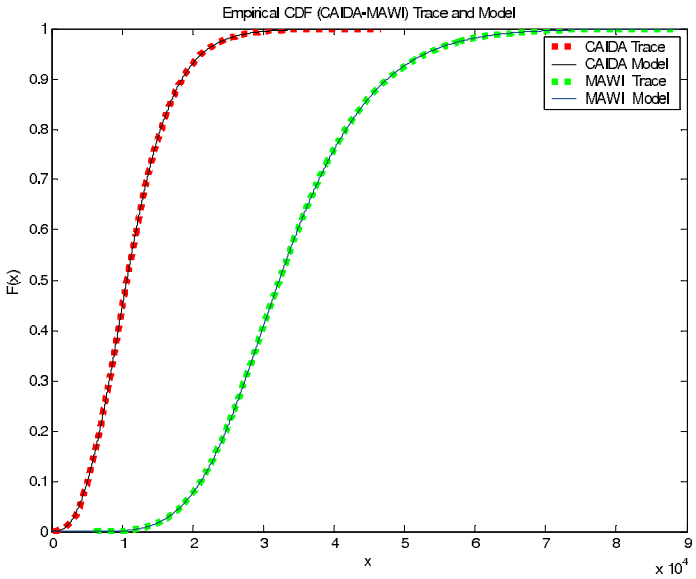


Figure 4.3: CDF of Slot Sizes (Trace and Model) for Both Traces

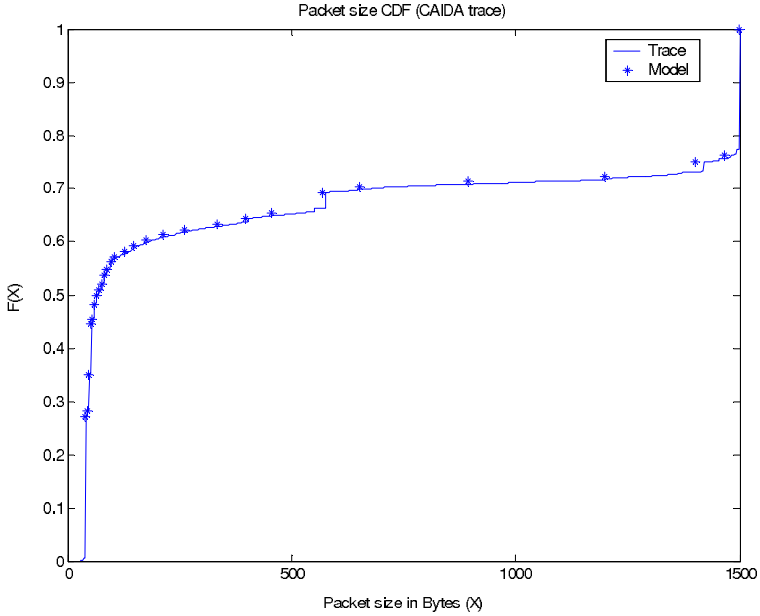


Figure 4.4: CDF of Packet Sizes (CAIDA Trace)

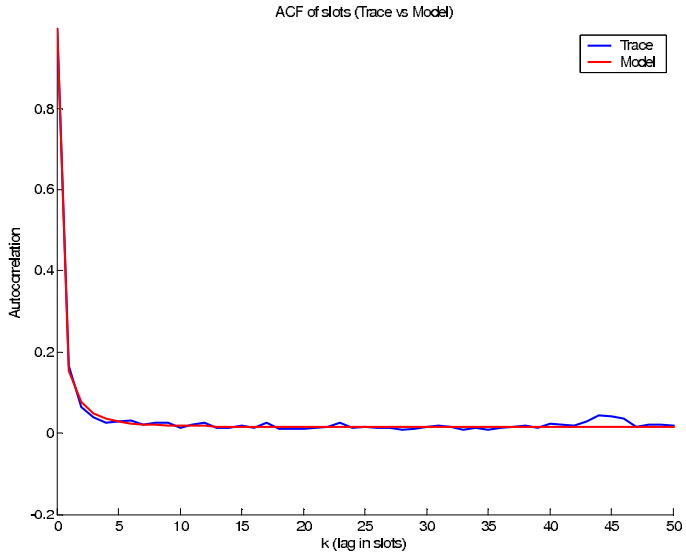


Figure 4.5: ACF of Slot Sizes (CAIDA Trace)

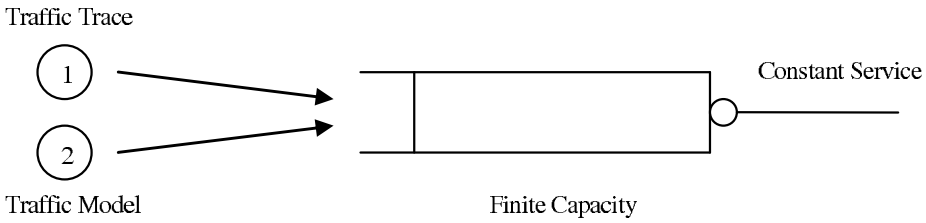


Figure 4.6: Simulation Network For Performance Validation

4.3.4.2 Performance Analysis

Performance analysis deals with the generated traffic behaviour in a queuing system compared to original traffic. Thus, we generate packets by the traffic model and we inject them into a $G/D/1/N$ queuing system (Figure **4.6**). Indeed, the service is constant and the queue capacity is finite N, but the resulting queuing system is not really a $G/D/1/N$ one, because packet sizes are not constant. However, for the purpose of our test, we want to evaluate the traffic in a router interface equivalent system. In such interface the service is constant (determined by the link bandwidth) and that is exactly what we do. The value of N is taken 64 packets for all next simulations.

We evaluate the average queue length and loss rate observed when injecting packets from traffic trace and traffic model into the queue. The test is achieved under high utilization rate of $\rho = 0.9$, $\left(\rho = \frac{Input\ Rate}{Output\ Rate}\right)$ for both traces. Results are shown in Table **4.4**.

Although the average rate is well estimated in both cases, the traffic model gives more optimistic results than real traffic concerning the loss rate and the average queue length. That means when using traffic generated by the model to evaluate the performance of a

Table 4.4: Load in Packets and Loss Rate for $\rho = 0.9$

Stats	**CAIDA** $\rho = 0.9$		**MAWI** $\rho = 0.9$	
	Trace	Model	Trace	Model
Loss rate %	5.2	2.3	5.31	2.1
Queue Load (Packets)	24.4	18.1	25.6	17.7
Mean Rate (Kbps)	90865.5	91265.8	25983.8	25269

network, the model may underestimate the burstiness of traffic and wrong decisions may be made about resource provisioning. This result is important as it predicts the failure of similar models in performance evaluation studies. It shows clearly that generating the correlation structure of time slots is not sufficient to guarantee a reliable performance of traffic when used in network simulation studies.

4.3.4.3 Analyzing the Results

Enhancing the performance of the traffic model implies understanding the possible reasons of the optimistic performance of the traffic model. Thus, we investigate packet sizes and packet inter-arrivals inside slots. Figure **4.7** depicts the empirical distribution of packet inter-arrivals during one slot compared to the presumed exponential distribution used in the model. We notice a difference between trace packet inter-arrivals in a slot and the presumed exponential inter-arrivals used in the model. A possible explanation of this difference is the traffic composition. In fact, the traffic in both traces is composed mainly of TCP packets (see Table **4.1**). TCP protocol transmits packets by bursts, which generate packet inter-arrivals of big variance, and consequently could hardly be approximated by an exponential law. Meanwhile, the exponential approximation is a practical solution from implementation point of view as it is very difficult to estimate packet inter-arrivals for each slot.

In the same way, we investigate the relation between packet size distribution and slot size. In fact, the model considers that packet size distribution is the same for all slots regardless their size; meanwhile slots have the same duration but not the same size. Figure **4.8** depicts the evolution of average packet size versus slot size. We can see that the average packet size increases with slot size in the traffic trace while the average packet size is almost the same for all slots in the traffic model. Indeed, we apply the same packet size distribution for all slots regardless their size and this is clearly wrong.

4.3.5 Modified Traffic Model

In order to improve the model, we suggest an empirical approach in which packet size distribution and packet inter-arrival distribution are estimated per slot size. Hence, we divide the slots into N groups, and we estimate the empirical packet size distribution for each group of slots. Besides, we calculate the mean and variance of packet inter-arrivals for each group of slots.

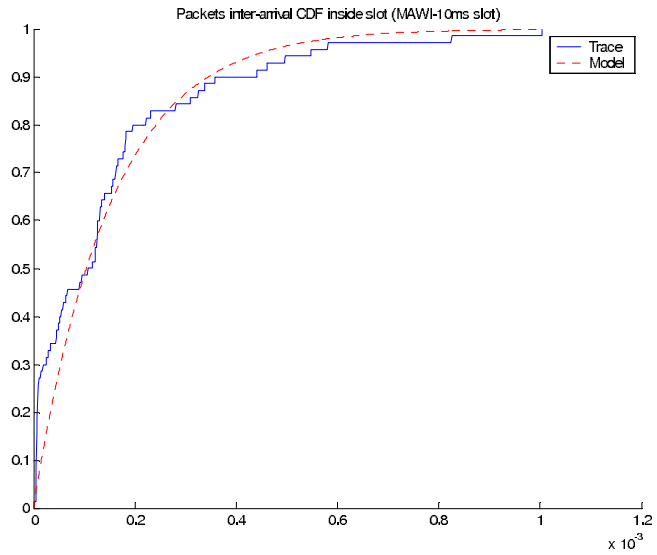


Figure 4.7: CDF of Packet Inter-arrivals (during one slot)

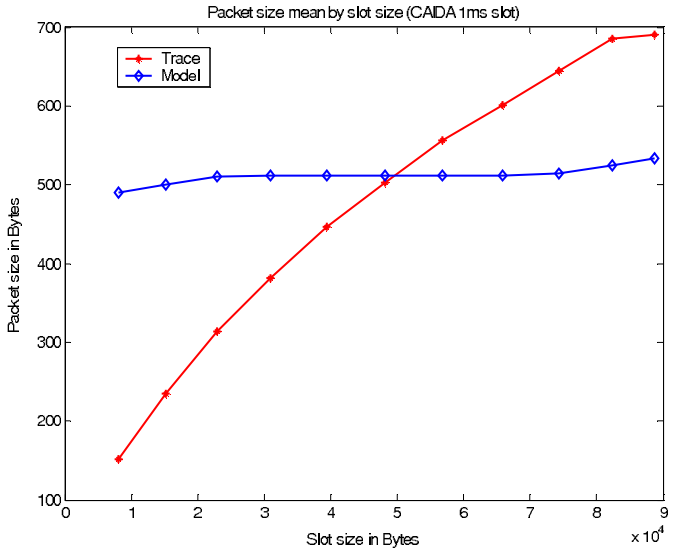


Figure 4.8: Average Packet Size versus Average Slot Size (CAIDA Trace)

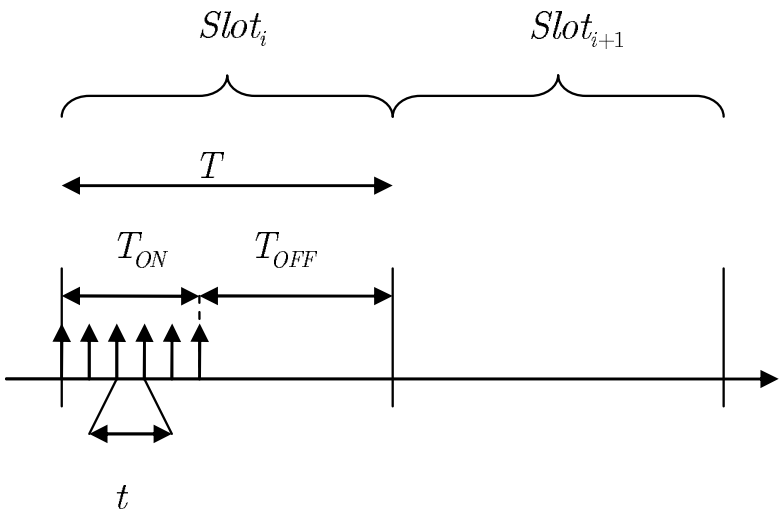


Figure 4.9: Suggested Packet Generation Process Inside Slots

Let us denote PS the packet size distribution over the whole trace, and $\{PS_i,\ i = 1, ..., N\}$ the packet size distribution per slot size group. We have:

$$E(PS) = \sum_{i=1}^{N} \frac{E(PS_i)}{N} \tag{4.12}$$

Similarly, let IA be the packet inter-arrival distribution over the whole trace, and $\{IA_i,\ i = 1, ..., N\}$ the packet inter-arrival distribution per slot size group, then:

$$E(IA) = \sum_{i=1}^{N} \frac{E(IA_i)}{N} \tag{4.13}$$

PS and PS_i are estimated by discrete distributions directly from the trace. On the other hand, we evaluate the average and variance of packet inter-arrivals for each group of slots and we generate packet inter-arrivals with distributions having the same average and variance. Two heuristics are proposed: first, we approximate the packet inter-arrivals inside slots by a Pareto distribution of the same average and variance, and second we generate packets according to an ON-OFF process (see Figure **4.9**). In the later, we consider that packet inter-arrivals are constant (t) during ON period while there are no packet arrivals during OFF period. The duration of ON and OFF periods are calculated with respect to measured average and variance of packet inter-arrivals per slot size group.

According to the proposed model, we have:

$$E(\lambda_{IA_i}) = \frac{1}{t} * \frac{T_{ON}}{T_{ON} + T_{OFF}} \tag{4.14}$$

$$\begin{aligned} Var(\lambda_{IA_i}) &= E(\lambda^2_{IA_i}) - (E(\lambda_{IA_i}))^2 \\ &= \left(\frac{1}{t}\right)^2 * \left(\frac{T_{ON}}{T_{ON}+T_{OFF}}\right) - \left(\frac{1}{t} * \frac{T_{ON}}{T_{ON}+T_{OFF}}\right)^2 \end{aligned} \tag{4.15}$$

Note that $T_{ON} + T_{OFF} = T$ is constant by construction (slot duration), and after some manipulations, we obtain:

$$T_{ON} = \left(\frac{(E(\lambda_{IA_i}))^2}{Var(\lambda_{IA_i}) + (E(\lambda_{IA_i}))^2}\right) * T \tag{4.16}$$

$$\lambda_{IA_i} = \frac{1}{t} = \left(\frac{Var(\lambda_{IA_i}) + (E(\lambda_{IA_i}))^2}{E(\lambda_{IA_i})}\right) \tag{4.17}$$

Once the values of T_{ON} and t are estimated, the generation process is completely specified. We summarize in the following the complete estimation process as well as the generation process:

1. Divide the trace into slots of equal durations T.
2. Estimate the correlation structure of the time series corresponding to the size of slots.
3. Estimate the probability distribution of slot sizes.
4. Group slots into N sets by size.
5. For each set of grouped slots:
   - Estimate the empirical packet size distribution.
   - Calculate the mean and variance of packet inter-arrivals.

The generation process in the modified model is identical to the previous case. In fact, the $M/G/\infty$ model will generate slot sizes respecting both probability distribution and correlation structure in the estimated traffic trace. The difference concerns only the way slot sizes are transformed into packet sizes with specified inter-arrivals. The following steps will take place after the generation of a size slot:

1. Determine the set to which belongs the generated slot.
2. Generate packets according to the packet size distribution of this set of slots.
3. Generate packet inter-arrivals respecting the estimated values per slot size, using either a Pareto distribution or an ON-OFF generation way.

Table 4.5: Load in Packets and Loss Rate for $\rho = 0.9$ (Modified Model)

Stats	**CAIDA** $\rho = 0.9$		**MAWI** $\rho = 0.9$	
	Trace	Model	Trace	Model
Loss rate %	5.2	4.9	5.31	5.1
Queue Load (Packets)	24.4	23.9	25.6	24.2
Mean Rate (Kbps)	90865.5	91292	25983.8	25311

4.3.5.1 Validation

The modified traffic model has the same statistical behaviour as the standard model. In fact, the modifications concerns only packet generation inside slots not the slot generation process. In Table **4.5** we show the simulation results using the modified model with the ON-OFF packet generation inside slots.

The modified model performs better in network simulation. The average queue length is almost the same in both cases. Similar results were obtained when using the Pareto distribution for packet inter-arrivals (not presented here). It seems that the model performance is influenced by the average and variance of packet inter-arrivals more than the distribution choice.

In fact, the accuracy of the model is achieved at the cost of estimating more parameters (N packet size and packet inter-arrival distributions, N equals to 10 in our case). This may slow down the execution of the corresponding traffic generator which must be considered as another factor in designing efficient traffic models.

However, the packet size distribution and packet inter-arrival distribution during slots is a problematic issue in similar aggregate traffic models (FGN, FARIMA ...). In fact, many models succeed to capture correctly slot size correlations and probability distributions. Unfortunately, the model performance in a network environment is always optimistic as it does not take into consideration the variation of packet size distribution and packet inter-arrival distribution in function of the slot size.

4.4 Conclusion

In this Chapter, we presented a traffic modelling methodology using the $M/G/\infty$ process with byte count process. The model shows great flexibility in capturing traffic correlations by mixing SRD and LRD correlations. Statistical evaluation of the model gives good results. On the other hand, the evaluation in network environment by simulation shows optimistic results comparing to traffic trace. We proposed to enhance the model by modifying packet inter-arrival distribution as well as packet size distribution in function of slot size by empirical tuning. The modified model performs better at the cost of higher computational complexity.

It is important to note that packet size distribution and packet inter-arrival distribution inside slots is a problematic issue in byte count process models. Indeed, capturing slot size correlations and probability distributions is not sufficient. Models must take

into consideration the variation of packet size distribution and packet inter-arrival distribution in function of slot size. This appears to be a dominant factor in the reliability of the generated traffic.

Chapter 5

Traffic Models for Multimedia Applications

5.1 Introduction

Deployment of new multimedia applications and services require reliable performance evaluation studies involving reliable and efficient traffic models. In this chapter, we focus on multimedia application modelling and characterization. Indeed, we distinguish between three categories of multimedia applications: Audio, Video and Data applications. Audio and video application are characterized by the CODECs (COder/DECoder) used to encode the digitized information signal. The generated traffic is transported by User Datagram Protocol (UDP) in general. Conversely, data applications concern files transfer activity where the information is already digital and the transport is achieved by Transmission Control Protocol (TCP).

The application behaviour is studied for each group of the above mentioned applications and a behavioural application model is provided and implemented using the Generic Framework for Traffic Modelling (refer to **Chapter 3**). Once the application model is defined a characterization study of traffic resulting from the superposition of N identical applications is achieved (Figure **5.1**). In fact, the resulting aggregated process could be simpler than the single application model, and it may be approximated by a Poisson process in some cases. This is an interesting feature in large scale network simulations as Poisson processes can be evaluated analytically. Finally, we validate our models and techniques by statistical analysis and network simulations.

5.2 Audio Applications

The voice is an analog signal, which should be digitized in order to be transmitted over packet networks. To be able to reproduce the original waveform, the sampling rate must be at least double that of the highest frequency in the original signal (Shannon's Law). Each sample must then be quantized using scalar, predictive scalar or vector methods.

In the scalar quantization, the samples are coded independently. A typical scalar

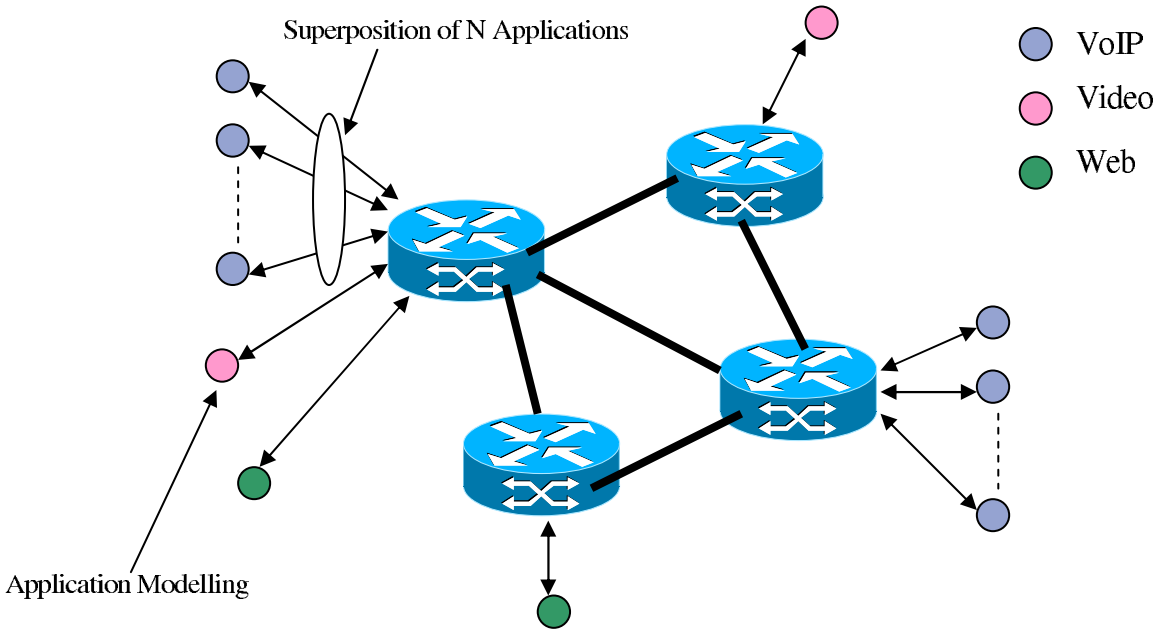


Figure 5.1: Multimedia Application Modelling

quantization method is the Pulse Code Modulation (PCM) in which each sample is encoded independently on 8 bits with a frequency of 8 kHz. A bit rate of 64 Kbps is then obtained. However, to reduce the bit rate, the predictive scalar quantization was introduced. It is based on the Adaptive Differential Code Modulation (ADPCM) algorithms. In this technique, one uses the high correlation that exists between subsequent samples. Predictive methods allow encoding only the difference of amplitude between a sample and its predicted value. Consequently, this quantization allows reducing the bit rate down to 16 Kbps.

Finally, in vector quantization, one uses the high correlation, which exists between several samples. These samples are gathered (vectors) in order to be quantized together. This quantization is based on the algorithms of CELP (Code Excited Linear Prediction) type and allows reducing the bit rate down to 5 Kbps. The typical applications of this speech coder are telephony over packet networks, like VoIP and the speech component in the videoconferencing.

5.2.1 Audio Application Modelling

The previous standard voice-encoding schemes have a fixed bit rate and a fixed packetization delay. The source is characterized by a stream of fixed size packets during talkspurts (ON Period) and no arrivals during silences (OFF period) as illustrated on Figure **5.2**. The packets with encoded voice are transported over the Internet by UDP while the packets with error control information are transported by TCP.

The most common used audio codecs for VoIP applications are listed in Table **5.1**.

Table 5.1: Common Audio Codecs

CODEC	**Description**	**Packet Size (Byte)**	λ_{ON}**(P/Sec)**
G711	ITU Recommendation in 1988 using Pulse Code Modulation (PCM).	136	83.4
G726	ITU Recommendation in 1990 using an Adaptive Differential Pulse Code Modulation (ADPCM).	104	62.5
G728	ITU Recommendation in 1992 using the Low-Delay Code Excited Linear Prediction (LDCELP).	88	41.7
G729	ITU Recommendation in 1996 using the Conjugate-Structure Algebraic Code Linear Prediction (CS-CELP).	70	33.4
G723.1 Low	ITU Recommendation in 1996 using Algebraic Code Excited Linear Prediction (ACELP).	60	33.4
G723.1 High	ITU Recommendation in 1996 using Multi Pulse-Maximum Likelihood Quantization (MPMLQ).	64	33.4
RealAudio Low	The average sending rate is typically 16 Kbps with 293 bytes packet sizes (without header) [MH00]	333	68.26
RealAudio High	The average sending rate is typically 20 Kbps with 495 bytes packet sizes (without header) [MH00]	535	50.5

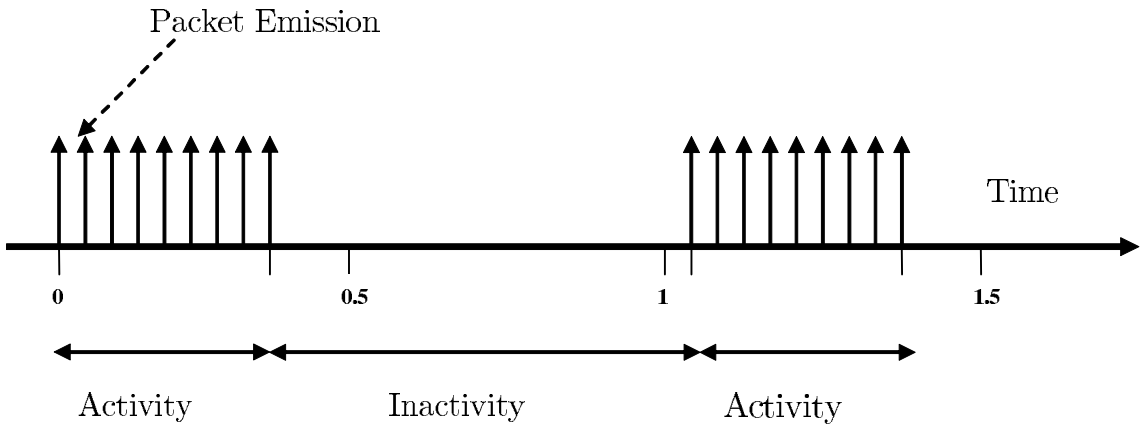


Figure 5.2: Behaviour of Single Audio Source

5.2.1.1 ON-OFF Model

Analyzing the voice traffic at the source level shows that it can be characterized by a succession of active periods (talkspurt or ON) followed by idle periods (silence or OFF). During the ON period, the source sends packets at regular intervals of length T (Packetization time). The duration of active and inactive periods is generally estimated by independent exponential laws of respective parameters α and β (refer to [Bra68, HL86, SW86]). A voice source may be viewed as a two state birth-death process with birth rate β and death rate α. The model is characterized by the following parameters:

- The mean duration of the ON period $T_{ON} = \frac{1}{\alpha}$. α is the parameter of the exponential law of the active period ON.
- The mean duration of the OFF period $T_{OFF} = \frac{1}{\beta}$. β is the parameter of the exponential law of the idle period OFF.
- The constant packet rate during the ON period: $\lambda = \frac{1}{T}$.
- The constant packet size: Payload + 40 (header IP/UDP/RTP) Bytes.

Figure **5.3** illustrates the ON-OFF model of a single audio source.

The average bit rate associated with an ON-OFF process is given by the following formula:

$$\bar{\lambda} = \frac{T_{on}}{(T_{on} + T_{off})T} = \frac{\beta}{(\beta + \alpha)T} \tag{5.1}$$

The value of $\bar{\lambda}$ depends very much on the voice codec. In general, T is chosen between 12ms and 30ms in order to ensure a short packetization delay and a non-negligible payload size per packet.

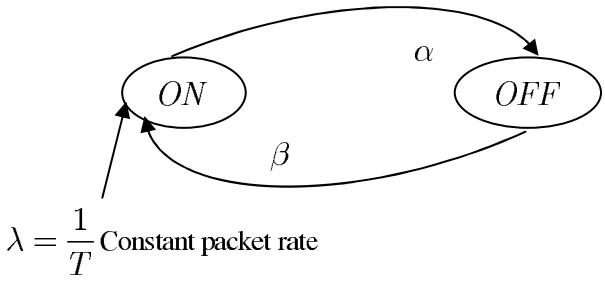


Figure 5.3: Audio ON-OFF Model

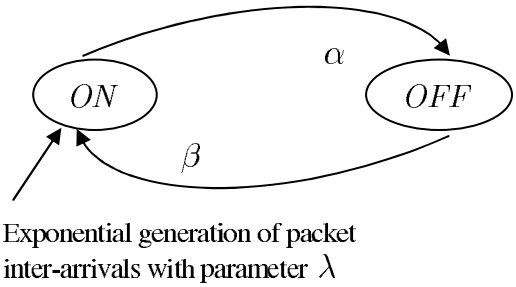


Figure 5.4: IPP Process

5.2.1.2 IPP Process

The two state model presented in the previous section has another variant with exponential arrival rate of packets during the ON period instead of constant arrival rate. The model is known under the name of Interrupted Poisson Process (IPP) which is particular case of the more general Markov Modulated Poisson Process (MMPP) with only two states. It has the same parameters as the ON-OFF model defined before with exponential packet transmission during the ON period of average $\lambda = \frac{1}{T}$.

The theoretical importance of the IPP process is in its simple mathematical aggregation into MMPP process (see MMPP process in section **5.3.3**). Unfortunately, the approximation of constant packet inter-arrivals by exponential ones is wrong as it does not correspond to packet generation profile of audio sources.

5.2.2 Characterization of Audio Application Traffic

The ON-OFF model used for Audio applications is characterized by exponential period durations, constant packet sizes and constant packet inter-arrivals. Deng [Den95] measured the duration of ON and OFF periods for telephonic and audio traffic in packet commutation networks. Thanks to his work we have numeric values for talk and silent periods. In fact, we distinguish between two categories of audio applications: Reading and Conversation. Actually values are different from classical telephonic traffic as shown in Table **5.2**.

The difference between these models could be explained by sensibility to silence in the

Table 5.2: Period Durations by Audio Application Type

Type	T_{ON} **Sec**	T_{OFF} **Sec**
Classical	0.352	0.65
Conversation	7.24	5.69
Reading	3.23	0.41

used CODECs. Moreover, activity in the Reading model is more fluid than conversation which results in very short silent periods.

The duration of one audio call is considered to be 3 min on average. Values of packet sizes and packet inter-arrivals depend on used codecs. All packets are transported by UDP. For the purpose of our study we use three types of codecs: G729, G711 and RealAudio Low with the three different categories of audio applications: Classical, Conversation and Reading. Numerical values of the studied models are listed in Table **5.3**.

Table 5.3: Audio Application Models

Application	**CODEC**	T_{ON} **Sec**	T_{OFF} **Sec**	**Packet Size Byte**	**IA ms**	$\overline{\lambda}$ **Kbps**
G729C	G729	0.352	0.65	70	30	6.3
G729V	G729	7.24	5.69	70	30	10.2
G729R	G729	3.23	0.41	70	30	16.2
G711C	G711	0.352	0.65	136	12	31.1
RAL	RealAudio Low	0.2	1.8	333	14.6	18.3

5.2.2.1 Homogeneous Superposition of Audio Applications

We are interested in specifying the packet inter-arrival process for an audio application especially when superposing N homogeneous audio applications. Let T be the fixed inter-arrival time between two packets during the ON period, then $n = \frac{T_{ON}}{T}$ is the number of transmitted packets during ON period. There are two possible packet inter-arrivals: T (with probability $p = \frac{n-1}{n}$) and $T + T_{OFF}$ (with $1 - p$ probability).

The tail of packet inter-arrival CDF from one source could be written:

$$1 - F(x) = \begin{cases} 1 & 0 \leq x \leq T \\ (1-p)e^{-\frac{x-T}{T_{OFF}}} & x \geq T \end{cases} \tag{5.2}$$

Sriram and Whitt [SW86] have proved that the tail of packet inter-arrival CDF from the superposition of N homogeneous sources could be written as:

$$1 - F_N\left(\frac{x}{N}\right) = \begin{cases} (1 - \lambda\frac{x}{N})^{N-1} & 0 \leq x \leq T \\ \frac{(1-p)^N e^{-\frac{x-NT}{T_{OFF}}}}{\left(\frac{T}{T_{OFF}}+1-p\right)^{N-1}} & x \geq T \end{cases} \tag{5.3}$$

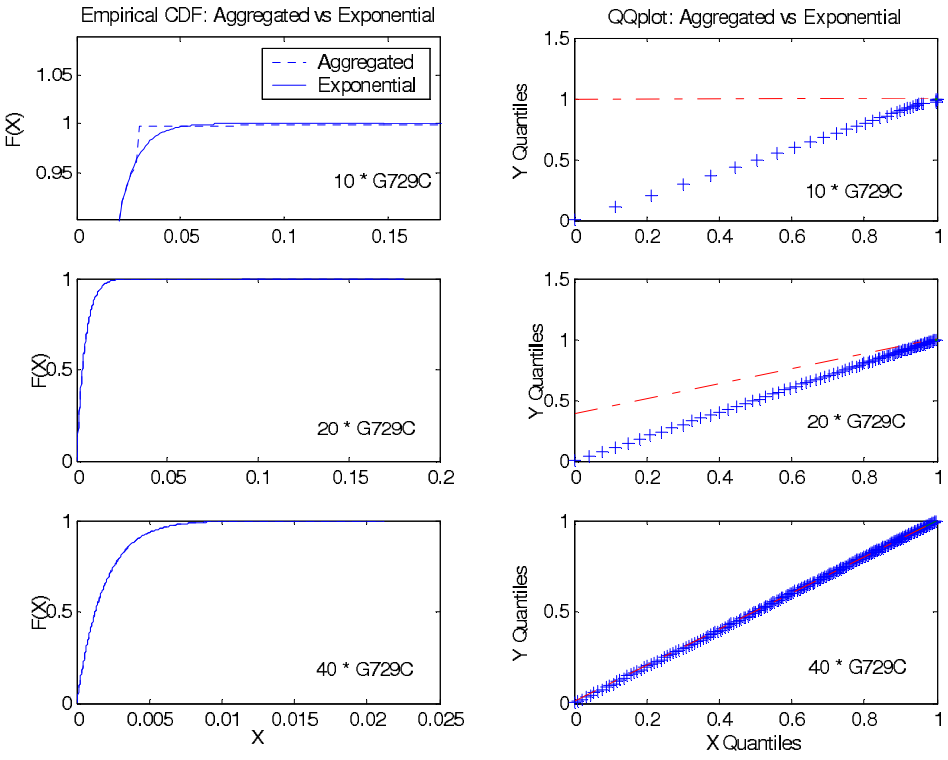


Figure 5.5: Packet Inter-arrival Distribution for G729C Audio Applications

It is clear that as $N \to \infty$ we have $1 - F_N\left(\frac{x}{N}\right) \to e^{-\lambda x}$, thus the superposition of infinite number of homogeneous audio applications tends to an exponential distribution of packet inter-arrivals. The question is: do we really need to superpose an infinite number of audio processes to consider that the resulting inter-arrival process is an exponential one? The answer is No, from a practical point of view we find that the superposition of a limited number of homogeneous audio applications tends quickly to exponential distribution of packet inter-arrivals.

First, we use the G729 codec with three audio categories: Classical, Conversation and Reading (denoted G729C, G729V and G729R). We generate the traffic corresponding to N connections of each audio category and we examine the packet inter-arrival distribution. Indeed, packet size distribution is simple in this case as we have constant packet size. We compare the packet inter-arrival distribution with equivalent exponential distribution for three values of N: 10, 20 and 40 parallel connections. Results concerning the G729C codec are shown on Figure **5.5**. We obtained similar results for the two other applications G729R and G729V (not represented here).

Our results show that with limited number of audio traffic sources (40 in this case), the superposed traffic tends to exponential distribution. The QQplot for 40 audio connections shows that the distribution of packet inter-arrivals is very well captured by an exponential distribution. Thus, we can model the superposition of $N \geq 40$ G729C connections by an equivalent exponential generator. Let λ_{ia} be the average packet inter-arrival rate, N the number of Audio connections, P_s the packet size, and then the packet inter-arrival rate

Λ_{ia} for the equivalent model is calculated as follows:

$$\Lambda_{ia} = N\lambda_{ia} \tag{5.4}$$

And the average rate is given by:

$$\Lambda = \Lambda_{ia} * P_s \tag{5.5}$$

5.2.2.2 Heterogeneous Superposition of Audio Applications

The simple aggregated model obtained for the superposition of homogeneous audio applications is very efficient. We generalize the model to the superposition of different audio codecs with different packet sizes. Thus, the superposition of N different audio codecs with $n_i,\ i = 1...N$ connections per codec and $\lambda_{ia,i},\ i = 1..N$ average inter-arrival rate for codec i can be modelled by an equivalent exponential law with average inter-arrival rate Λ_{ia} calculated by:

$$\Lambda_{ia} = \sum_{i=1}^{N} n_i \lambda_{ia,i} \tag{5.6}$$

Packets are generated according to a discrete distribution Ps constructed as follows:

$$\begin{aligned} Ps &= \{Ps_i,\ i = 1...N\} \\ \Pr(Ps = Ps_i) &= \frac{n_i \cdot \lambda_{ia,i}}{\sum_{i=1}^{N} n_i \cdot \lambda_{ia,i}} \end{aligned} \tag{5.7}$$

Assuming that Ps_i is the packet size corresponding to codec i.

Numerical validation We superpose 10 connections of each of the audio applications listed in Table **5.3**. We evaluate packet size distribution and packet inter-arrival distribution and then we construct an equivalent aggregated model for the superposed traffic. Figure **5.6** depicts packet size distribution of superposed traffic, while Figure **5.7** depicts packet inter-arrival distribution. The obtained results validate the exponential approximation for the superposition of heterogeneous audio applications.

5.2.2.3 Performance Limits of the Exponential Approximation

So far the exponential approximation for the superposition of homogeneous and heterogeneous audio applications was verified only by distribution fitting. The suggested traffic model needs to be evaluated in network environment as well. For this purpose we inject N superposed audio sources into a queuing system of deterministic service denoted $G_{ON-OFF}/D/1$. Then we evaluate the average total number of clients (packets) X present in the system (including the packet being served) to show the influence of connections' number N as well as traffic intensity ρ. We compare our results with $M/D/1$ system (the exponential approximation case).

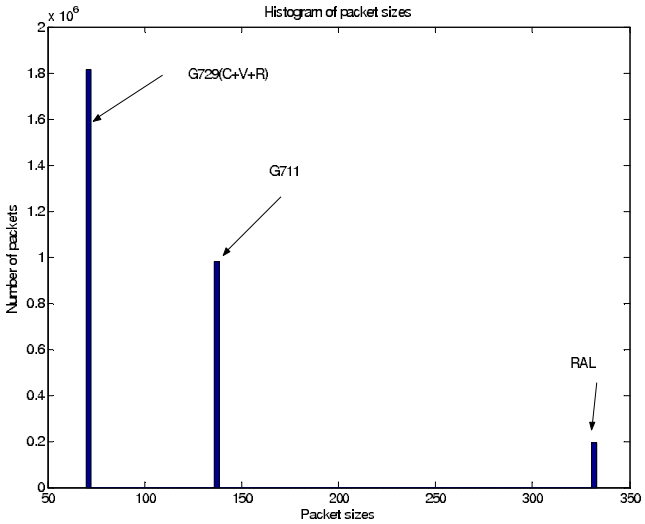


Figure 5.6: Packet Size Distribution For Heterogeneous Audio Traffic

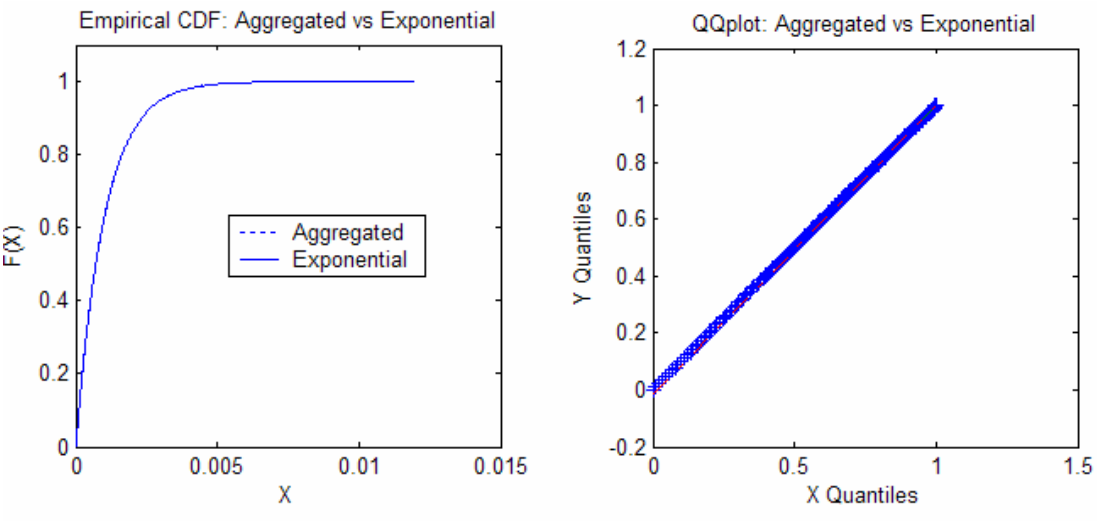


Figure 5.7: Packet Inter-arrival Distribution for Heterogeneous Audio Traffic

First, we evaluate the average queue length as a function of connection's number N under constant traffic intensity ρ using the G711C audio application model ($G711_{ON-OFF}/D/1$) with the following parameters:

- N : Number of superposed processes.
- X : Total number of clients in the system.
- ρ : Traffic intensity $\rho = \frac{\lambda}{\mu}$. with λ as the input rate and $D = \frac{1}{\mu}$.

Recall that the average queue length in the case of $M/D/1$ is given by [Kle75]:

$$X_{M/D/1} = \frac{\rho\left(1 - \frac{\rho}{2}\right)}{1 - \rho} \tag{5.8}$$

Table 5.4: Queue length Comparison: Exponential vs Superposed (Constant ρ)

N	$X_{M/D/1}$	$X_{G711_{ON-OFF}/D/1}$	ρ	λ Kbps
10	3.5	41	0.86	311
20	3.5	36.2	0.86	622
30	3.5	28.9	0.86	933
40	3.5	26.5	0.86	1244
50	3.5	24.13	0.86	1555
100	3.5	11	0.86	3110
500	3.5	3.74	0.86	15550

Results in Table **5.4** show a significant difference between the exponential model and the superposed process. Although the exponential approximation is valid from 40 connections statistically, the average queue load generated by superposed traffic is much higher than the one generated by the model under the same traffic intensity (26.5 packets vs only 3.5 packets under $\rho = 0.86$).

We investigate the influence of traffic intensity by fixing the number of connections (40 in this case) and varying the traffic intensity. Results are shown on Table **5.5**.

Table 5.5: Queue length Comparison: Exponential vs Superposed (Constant N)

ρ	$X_{M/D/1}$	$X_{G711_{ON-OFF}/D/1}$	N
0.86	3.5	26.5	40
0.7	1.516	1.997	40
0.65	1.254	1.259	40
0.58	0.98	0.984	40
0.46	0.656	0.659	40
0.3	0.36	0.36	40
0.12	0.13	0.13	40

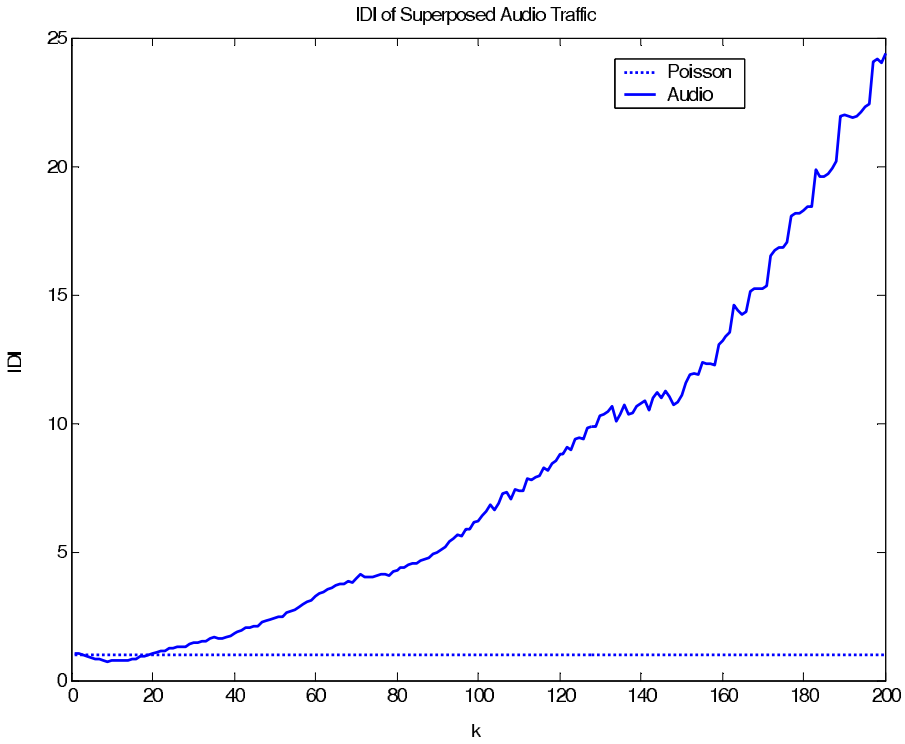


Figure 5.8: Evolution of IDI for Superposed Audio Traffic vs Equivalent Poisson Traffic

Experimental results indicate that the exponential approximation is adequate below a certain traffic intensity ρ^* ($\rho^* = 0.65$ in this case), and wrong above.

In order to understand the reasons of this poor performance under high traffic loads we investigate the correlation structure of traffic, by analyzing the Hurst exponent H and the index of dispersion for inter-arrivals IDI.

5.2.2.4 Correlation Analysis of Audio Traffic

We analyze the superposed traffic resulting from audio applications in order to characterize the nature of the existing correlations. We calculate the average Hurst parameter value using an R/S analysis of self-similarity. Results show that the superposed audio traffic presents strong correlations expressed by a strong average value of Hurst parameter. For example, using the superposed heterogeneous audio traffic with three codecs of section **5.2.2.2**, we obtain an average Hurst parameter value of 0.87. Although the statistical distribution of packet inter-arrivals could be approximated by exponential law some strong correlations exist at larger time scales.

The IDI index allows a better understanding of the phenomenon. Figure **5.8** shows the evolution of the squared coefficient of variation for the superposed audio (VoIP) traffic vs Poisson traffic. The IDI of a Poisson process is constant and equals to one. However, the IDI for superposed audio traffic is increasing with k which means that the k-squared coefficient of variation (i.e. the IDI) of inter-arrivals is not stable with time like Poisson traffic. The impact of this cumulated covariance appears at larger time

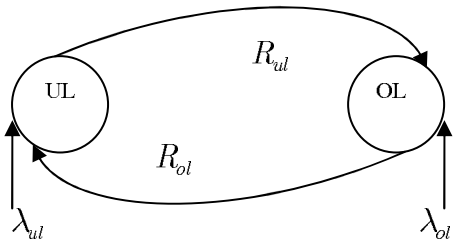


Figure 5.9: Equivalent MMPP-2 Process

scales, causing a poor performance in network simulations. Although superposed audio traffic shows exponential behaviour statistically the cumulated inter-arrivals covariance deteriorates traffic performance in queues rapidly compared to Poisson traffic.

5.2.2.5 Audio Traffic Under Heavy Loads

The exponential approximation is adequate for light to medium queue loads. Notice that the exponential model is very interesting as it allows analytical evaluation of traffic performance in queuing networks (Bit rate, Delay, Jitter ...).

However, for heavy traffic loads another approximation should be used. Shah-Heydari and Le-Ngoc [LNSH98] suggest that the superposition of N homogeneous (identical) ON-OFF processes with parameter β related to the inactivity period, parameter α related to activity period and $\lambda = \frac{1}{T}$ peak rate in the active period, could be approximated by an MMPP-2 process (see Figure **5.9**).

The goal is to determine an M value used to split the state space into two parts: UL (UnderLoad) associated with states $\{1, 2, ..., M\}$ and OL (OverLoad) associated with states $\{M+1, ..., N\}$, with $M = \lfloor NP_{on} \rfloor$. Indeed, M corresponds to the state of average transmission and is such that $M < N$. The transmission rates λ_{ul} and λ_{ol} associated with the two states UL and OL are given by:

$$\begin{aligned} \lambda_{ul} &= \lambda \sum_{i=0}^{M} i \frac{\pi_i}{\pi_{ul}} \\ \lambda_{ol} &= \lambda \sum_{i=M+1}^{N} i \frac{\pi_i}{\pi_{ol}} \end{aligned} \tag{5.9}$$

With:

$$\begin{aligned} \pi_{ul} &= \sum_{i=0}^{M} \pi_i \\ \pi_{ol} &= \sum_{i=M+1}^{N} \pi_i \end{aligned} \tag{5.10}$$

And:

$$\pi_i = \frac{N!}{i!(N-i)!} P_{on}^i (1 - P_{on})^{N-i} \tag{5.11}$$

The transition rates r_{ul} and r_{ol} are determined from the Index of Dispersion of Counts (IDC) of correlation associated with arrival process, $N(0, t)$, which counts the number

of packets in interval $[0, t]$.

$$IDC(t) = \frac{Var(N(0,t))}{E(N(0,t))} \tag{5.12}$$

We note $IDC(\infty) = \lim_{t\to\infty} IDC(t)$.

By equalizing the index $IDC(\infty)$ associated with the superposed process (N sources) with the index $IDC(\infty)$ associated with MMPP-2 process, we obtain the transition rates r_{ul} and r_{ol}. The index $IDC(\infty)$ is calculated by representing the source as a renewal process.

$$IDC(\infty) = \lim_{t\to\infty} \frac{Var(N(0,t))}{E(N(0,t))} = \frac{1-(1-\alpha T)}{(\alpha T+\beta T)^2} = a_\infty \tag{5.13}$$

For an MMPP-2 process the index $IDC(\infty)$ is given by:

$$IDC(\infty) = \lim_{t\to\infty} \frac{Var(N(0,t))}{E(N(0,t))} = 1 + \frac{2(\lambda_{ul}-\lambda_{ol})^2 r_{ul} r_{ol}}{(r_{ul}+r_{ol})^2(\lambda_{ul} r_{ol} + \lambda_{ol} r_{ul})} \tag{5.14}$$

Also by equalizing the average transmission rate we get:

$$NP_{on}\lambda = \lambda_{ul}\frac{r_{ol}}{r_{ol}+r_{ul}} + \lambda_{ol}\frac{r_{ul}}{r_{ol}+r_{ul}} \tag{5.15}$$

Finally we have:

$$r_{ol} = \frac{2(\lambda_{ol}-NP_{on}\lambda)^2(NP_{on}\lambda-\lambda_{ul})}{(\lambda_{ol}-\lambda_{ul})NP_{on}\lambda(a_\infty-1)} \tag{5.16}$$

$$r_{ul} = \frac{2(\lambda_{ol}-NP_{on}\lambda)(NP_{on}\lambda-\lambda_{ul})^2}{(\lambda_{ol}-\lambda_{ul})NP_{on}\lambda(a_\infty-1)} \tag{5.17}$$

Numerical validation In order to validate the MMPP-2 approximation process we compare the IDI of MMPP-2 process versus the IDI of the superposed audio (ON-OFF) traffic (Figure **5.10**).

Tests with MMPP-2 process are positive. The model captures well traffic correlations. The MMPP-2 approximation follows well the correlation structure of superposed audio traffic. Meanwhile, the superposed traffic still has stronger correlations expressed in stronger values of IDI.

Finally, we evaluate the average queue length (average number of packets X) of both inputs MMPP-2 and N superposed ON-OFF sources in a queuing system with deterministic service (Table **5.6**). Here also we obtain good results.

The MMPP-2 process represents an improved approximation of superposed audio traffic under heavy traffic loads. Of course, it can also be used under light loads, but no analytical evaluation of QoS parameters for traffic could be achieved when queuing networks are considered. That is why the exponential approximation still has a great benefit.

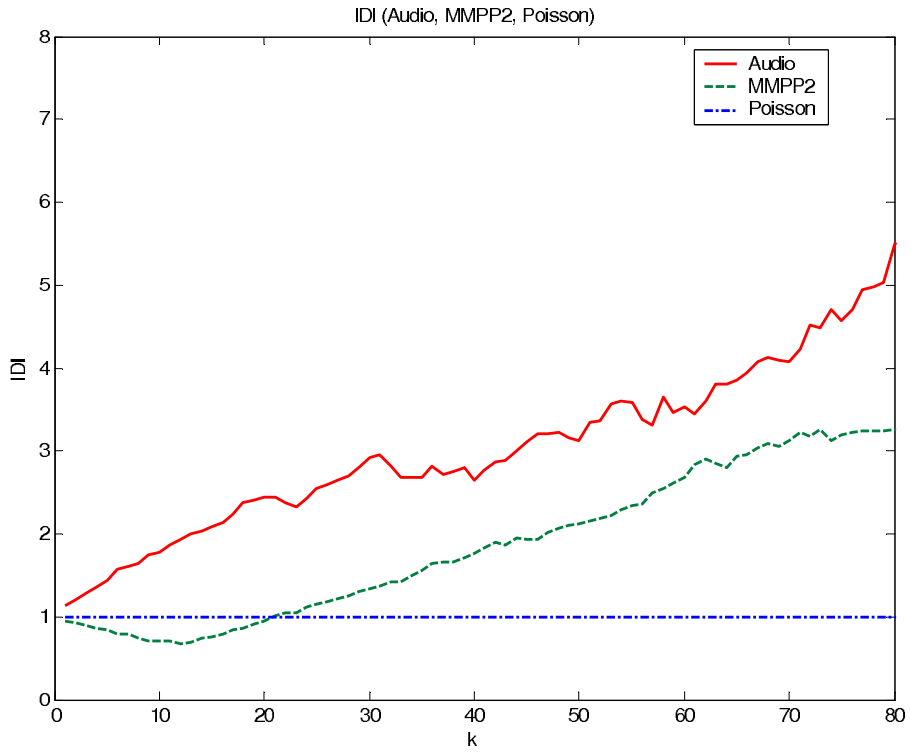


Figure 5.10: Evolution of IDI for Superposed Audio Traffic vs Equivalent MMPP-2 Traffic

Table 5.6: Queue length Comparison MMPP-2 vs Superposed $ON - OFF$ Audio Processes (Variable N, Constant ρ)

N	$X_{M/D/1}$	$X_{MMPP-2/D/1}$	$X_{G711_{ON-OFF}/D/1}$	ρ
10	3.5	36	41	0.86
20	3.5	31.9	36.2	0.86
30	3.5	23.2	28.9	0.86
40	3.5	21	26.5	0.86
50	3.5	17	24.13	0.86
100	3.5	8	11	0.86
500	3.5	3.6	3.74	0.86

5.2.2.6 Load Threshold Estimation

We have aggregated audio traffic models for light and heavy traffic loads, but still we need to know under which load ρ^* we can consider the traffic load light or heavy from a traffic modelling point of view. Our goal in this section is to provide a first order approximation of this traffic load threshold ρ^* estimated from superposed audio traffic characteristics.

We use average load criteria to evaluate this threshold. Recall that the average load of $M/D/1$ queue system (the exponential model case) is given by:

$$X_{M/D/1} = \frac{\rho(2-\rho)}{2(1-\rho)} \tag{5.18}$$

We use a first order approximation of superposed input audio traffic by $GI/D/1$ queue system, for which the average load is given by [Whi83]:

$$X_{GI/D/1} = \rho + \frac{\rho^2 c_a^2}{2(1-\rho)} \tag{5.19}$$

The superposed audio traffic has a squared coefficient of variation for inter-arrivals given by [SW86]:

$$c_a^2 = w c_1^2 + 1 - w \tag{5.20}$$

While c_1^2 and w are given by:

$$\begin{array}{l} c_1^2 = \frac{1-p^2}{(T/T_{OFF}+1-p)^2} \\ w = \frac{1}{1+4(1-\rho)^2(n-1)} \end{array} \tag{5.21}$$

$p = \frac{n-1}{n}$ is the probability of generating one packet and $n = \frac{T_{ON}}{T}$ is the number of generated packets.

We search values of ρ^* under which the two average queue loads of $M/D/1$ and $GI/D/1$ have a non significant relative error ε:

$$\left| \frac{X_{M/D/1} - X_{GI/G/1}}{X_{M/D/1}} \right| < \varepsilon \tag{5.22}$$

This equation is solved numerically. Figure **5.11** depicts the ρ^* values for the G711 audio application with two different values of ε: 1% and 5%.

However, compared with simulation results the estimated values of ρ^* using this approximation are pessimistic. For example, for 500 connections the estimated $\rho^* = 0.44$ with 5% relative error while simulations show that till $\rho^* = 0.76$ the exponential approximation is accepted for the same relative error. This could be explained by the rough approximation of packet arrivals in superposed audio traffic with an GI arrival process.

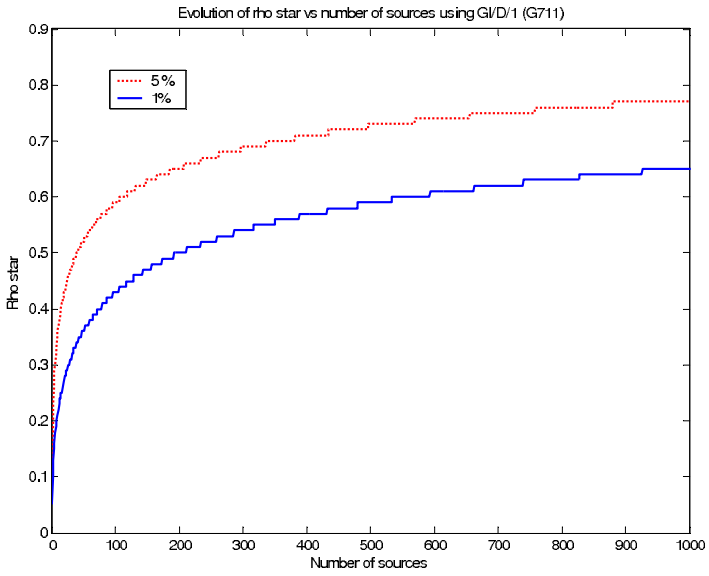


Figure 5.11: Evolution of ρ^* vs Number of Audio Sources (G711)

5.3 Video Applications

Video applications such as digital television, video conferencing, video on demand, streaming video, ... requires video signal digitization. When the analog video signal is converted into a digital signal, the bit rate obtained is typically higher than 100 Mbps. The transmission of uncompressed digital video requires the allocation of very large bandwidth in telecommunication networks. For instance, the bit rate associated with a digital television signal is estimated at 216 Mbps. In order to reduce the necessary bandwidth to the different video applications, several video codecs were developed.

The compression technologies associated with different codecs are based on spatial or/and temporal redundancies of video sequences. Indeed, they make use of the spatial redundancy, which exists between neighbouring pixels in a same image, as well as the temporal redundancy, which is due to the correlation between successive video frames. An overview of the principal video compression standards (ITU and MPEG) is given in **Appendix B**. In the next section we present the most important coding scheme: MPEG.

5.3.1 MPEG Coding

MPEG stands for Moving Pictures Experts Group, an International Standards Organization group formed to standardize audio and video compression. There are many MPEG standards: MPEG1, MPEG2 and MPEG4 and more recently MPEG7.

The MPEG standard defines three pictures encoding: Intraframe coding or I-Pictures (no prediction), P-Pictures, incorporating motion prediction from the previous video image and, B-Pictures (bi-directional prediction) which include motion prediction one frame ahead as well as from the previous frame. MPEG encoders generate a sequence of

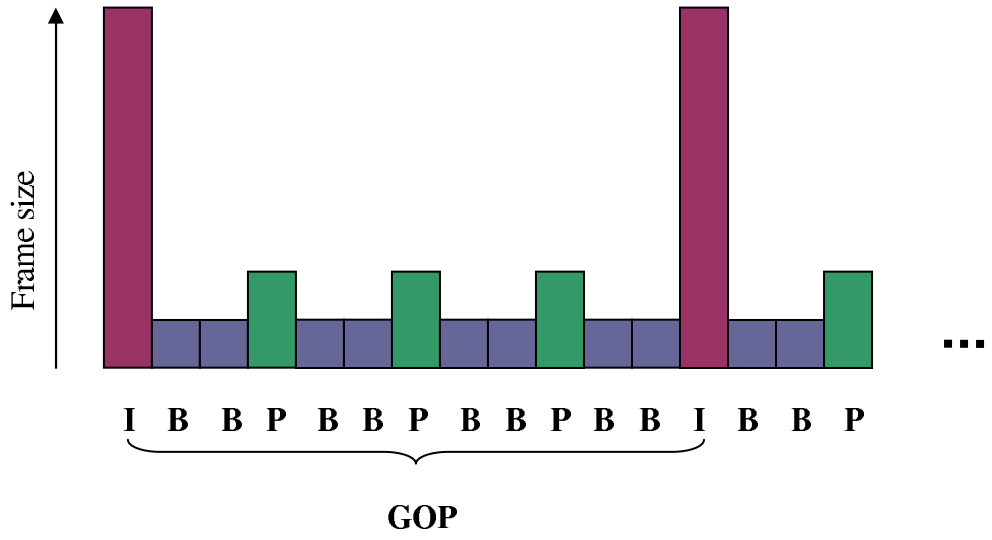


Figure 5.12: GOP Structure in MPEG Coding

frames (I, P and B) according to a cyclic frame pattern, which is referred to as a Group Of Pictures (GOP). GOP is usually determined by its length N the number of images between two I frames and M the number of B images between two P frames. Usually the numerical values are $N = 12$ and $M = 2$ as illustrated on Figure **5.12**.

Table **5.7** summarizes the bit rate associated with MPEG video codecs.

Table 5.7: Bit Rate Associated with MPEG Video Codecs

Video Codec	**MJEPG**	**MPEG1**	**MPEG2**	**MPEG4**
Bit Rate Mbps	8-10	1-1.5	4-10	0.005-4

5.3.2 Video Packet Size

Unlike voice, video has a very high and extremely variable packet rate with a much higher average Maximum Transmission Unit (MTU). The packet size cannot be larger than the path MTU, which is defined to be the minimum of the MTUs along all the used links from the source to the destination. Packets exceeding the path MTU are fragmented and reassembled at the source and destination nodes. For Ethernet, a classical value of MTU is 1500 bytes. This gives us a maximum payload (UDP + RTP headers + RTP payload) of 1480 bytes for IPv4, and 1460 bytes for IPv6 (IP header of 40 bytes).

Video traffics are transported using an application layer framing protocol, RTP (Real-time Transport Protocol). RTP, together with its payload format profiles, recommends encoding all RTP payloads into one more RTP packets that don't exceed the MTU of the transport layer. A typical size of UDP packet containing encoded video is 1000 Bytes.

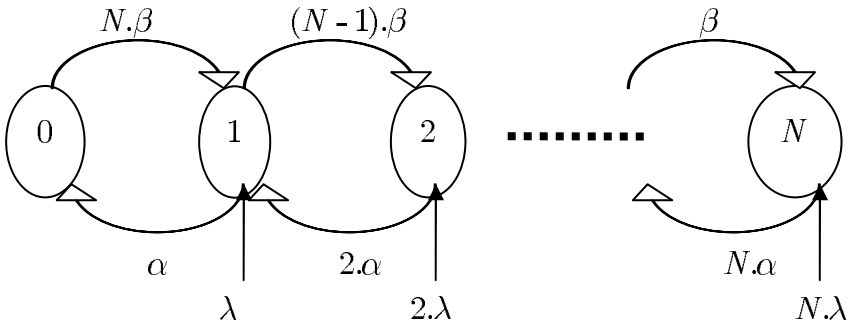


Figure 5.13: MMPP-N+1 Process

5.3.3 Video Application Modelling

Although video traffic could have extremely variable bit rates, some applications like videoconferencing and video streaming work with very low bit rates. We differentiate between regular video streams and variable bit rate video streams. In the following we present traffic models for both cases.

5.3.3.1 The MMPP Model

Regular video streams (without abrupt change of scene) such as videoconferencing or emissions over internet encoded with the H261 codec can be adequately modelled by a Markov Modulated Poisson Process MMPP [MAS+88].

An MMPP is a Poisson process whose rate is controlled by a finite-state Markov chain (Figure **5.13**). This MMPP Process with N+1 states is characterized by (Q, Λ) where Q is the infinitesimal generator associated with the state process (in our case, it's a birth-death process) and Λ the transmission rates associated with each state of the Markov chain.

$$Q = \begin{pmatrix} -N\beta & N\beta & 0 & 0 & & \\ \alpha & -(N-1)\beta-\alpha & (N-1)\beta & 0 & & \\ 0 & 2\alpha & -(N-2)\beta-2\alpha & (N-2)\beta & & \\ & & \vdots & & & \\ & & \vdots & (N-1)\alpha & -(N-1)\alpha-\beta & \beta \\ & & & 0 & N\alpha & -N\alpha \end{pmatrix} \tag{5.23}$$

And

$$\Lambda = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & \lambda & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & N\lambda \end{pmatrix} \tag{5.24}$$

The state N represents the maximal rate of coding. The characteristics associated with the bit rate are calculated by the following formula:

- The steady-state mean of instantaneous bit rate:$\bar{\lambda}(t) = N\lambda\frac{\beta}{\alpha+\beta}$
- The steady-state variance of instantaneous bite rate: $var(\lambda(t)) = N\lambda^2\frac{\alpha\beta}{(\alpha+\beta)^2}$
- The steady-state autocorrelation function of instantaneous bit rate: $\rho(k) = e^{-(\alpha+\beta)*k}$

The MMPP process model can not capture adequately all kind of correlations that may exist in traffic (its correlation function is markovian $\rho(k) = e^{-(\alpha+\beta)*k}$). As a consequence it is not suitable for modelling traffic presenting short or long range correlations. That is why other models are preferred to model high variable bit rate videos presenting short or long range correlations.

5.3.3.2 The Transform-Expand-Sample Model

Transform-Expand-Sample (TES) models provide a modelling approach that captures probability distributions and autocorrelation function of empirical records simultaneously. The empirical TES methodology assumes that some stationary empirical time series are available. It aims to construct a model satisfying the following requirements, simultaneously: the model's probability distribution should match its empirical counterpart and the model's leading autocorrelations should approximate their empirical counterparts up to a reasonable lag.

The derivation of TES models is performed in two phases. First, a correlated sequence, with uniform probability in [0,l], (also called background TES process) is formed as follows:

$$\begin{aligned} U_n^+ &= \begin{cases} U_0 & n = 0 \\ \left\langle U_{n-1}^+ + V_n \right\rangle & n > 0 \end{cases} \\ U_n^- &= \begin{cases} U_n^+ & n \quad even \\ \langle 1 - U_n^+ \rangle & n \quad odd \end{cases} \end{aligned} \tag{5.25}$$

Where U_0 is a random variable uniformly distributed in [0,l]. $\{V_n\}$ is a sequence of independent and identically-distributed (i.i.d) random variables independent of U_0, called the innovation sequence (with probability density f_v). The operator <> denotes the modulo-1 operation which for every real x is defined by:

$$\langle x \rangle \;=\; x - \max\{\text{integer}(n) : n \leq x\} \tag{5.26}$$

Both U_n^+ and U_n^- can generate lag-1 autocorrelations in the range $[0, 1)$ and $[-1, 0)$. In the second phase, synthetic sample data, called the foreground sequence, which resembles the real sample, may be derived from the background sequence using an inversion technique. This inversion technique allows the transformation of any uniform random variable to one with arbitrary distribution.

The TES modelling technique has a low computing complexity and can be applied for fast simulation. On the other hand, it requires high programming complexity. More

details about the specific application of the TES model in video traffic modelling can be found in [MS92].

5.3.3.3 The $M/G/\infty$ Process

Several studies [GW94, KM98, Ros95, Thi02] showed that video streams (e.g. MPEG) exhibit a significant correlation between frames or GOPs. The MPEG traffic is divided accross time by GOP structure. The correlations are in part due to the periodic generation of GOPs. The idea is to recreate the observed generation using a correlated pattern generator. The $M/G/\infty$ process (see **Chapter 4**) is very suitable to recreate correlations between time slots.

In MPEG setting, each slot represents the time needed for generating a frame (typically 1/25s or 1/30s) or a Group of Pictures (GOP) (typically 12/25s, 15/30s, 15/25s or 15/30s) depending on the codec that is used. The input $M/G/\infty$ model associated with a sequence of video is thus characterized by the following parameters:

- Time scale or time slot: Frame or GOP level.
- Frame or GOP size distribution: Mixture distribution.
- Correlation type: Markov, Short-Range Dependence or Long-Range Dependence.

The particularity of $M/G/\infty$ is that it is stable by aggregation. Consider N $M/G/\infty$ models defined by:

- λ_i : Poisson law parameter.
- $F_i(x)$: Distribution function associated with service time σ_i.

The aggregation of these models results in a $M/G/\infty$ process defined by:

$$\begin{aligned} \lambda &= \sum_{i=1}^{N} \lambda_i \\ F(x) &= \sum_{i=1}^{N} \left(\frac{\lambda_i}{\lambda}\right) F_i(x) \end{aligned} \tag{5.27}$$

With $F(x)$ representing the distribution function associated with G.

Due to its versatility and powerful representation of correlations, we use the $M/G/\infty$ process to model video traffic using the same estimation techniques introduced in **Chapter 4**. However, MPEG traffic is naturally divided into slots because of the periodic GOP generation structure.

5.3.4 Characterization of Video Application Traffic

We model MPEG traffic using the $M/G/\infty$ process. Our models are based on statistical estimation of different video traces. These traces are very well described in [TNG00].

Video traffic can be modelled at GOP or frame level. We estimate the correlation structure of video traces based on markovian($\rho_0(k) = e^{-b*k}$), short range ($\rho_1(k) = e^{-b*\sqrt{k}}$) or long range ($\rho_2(k) = (k+1)^{-b}$) correlation models and their combinations. Besides, we estimate the probability distributions of GOP or frame sizes by mixture distributions. Packet sizes are constant of 1000 bytes.

A wide range of video models were estimated and a complete library of MPEG video models was implemented (The estimation procedure is the same as the one used for IP traffic traces in **Chapter 4**). The reader may refer to **Appendix B** for the complete list of estimated video traffic models. For the purposes of our characterization study, we were limited to three traffic models for MPEG1, MPEG2 and MPEG4 codecs (Table **5.8**). Similar results were obtained with other models.

Table 5.8: Video Traffic Models

Codec	**Time Slot**	**ACF**	**Frame or GOP Size Distribution (Byte)**
MPEG1 (Dino)	GOP 12/25s	$e^{-0.35\sqrt{k}}$	LogNormal Mean=141350 Var=2.1544e+09
MPEG2 (Wizard of Oz)	Frame 1/25s	$e^{-0.055\sqrt{k}}$	LogNormal Mean=21015 Var=1.1443e+08
MPEG4 (Die Hard: High)	GOP 12/25s	$e^{-0.1722\sqrt{k}}$	Gamma+Lognormal Mean=41968 Var=5.1139e+08

5.3.4.1 Video Models Validation

Using the estimated parameters presented in the previous section we construct traffic generators corresponding to each one of the three video traces. We validate the generated traffic versus traces both statistically and in a simulation environment. Figure **5.14** depicts the autocorrelation structure of GOPs for both MPEG4 video trace and MPEG4 video model. The generated GOPs present similar correlations as in video trace. For this particular video trace the correlation is SRD with an autocorrelation function $\rho(k) = e^{-0.1722\sqrt{k}}$.

The probability distribution of GOP sizes for this MPEG4 video trace is captured by a Gamma+Lognormal mixture distribution. Figure **5.15** depicts the CDF of GOP sizes for both video trace and video model. The statistical fitting is very good.

We evaluate the performance of generated video traffic in a queuing system of deterministic service. Recall that packet size distribution is constant with 1000 Bytes. On the other hand, packet inter-arrival distribution is uniform inside GOPs. Table **5.9** lists the results of the performance analysis of video traffic model versus video trace under high load of $\rho = 0.9$.

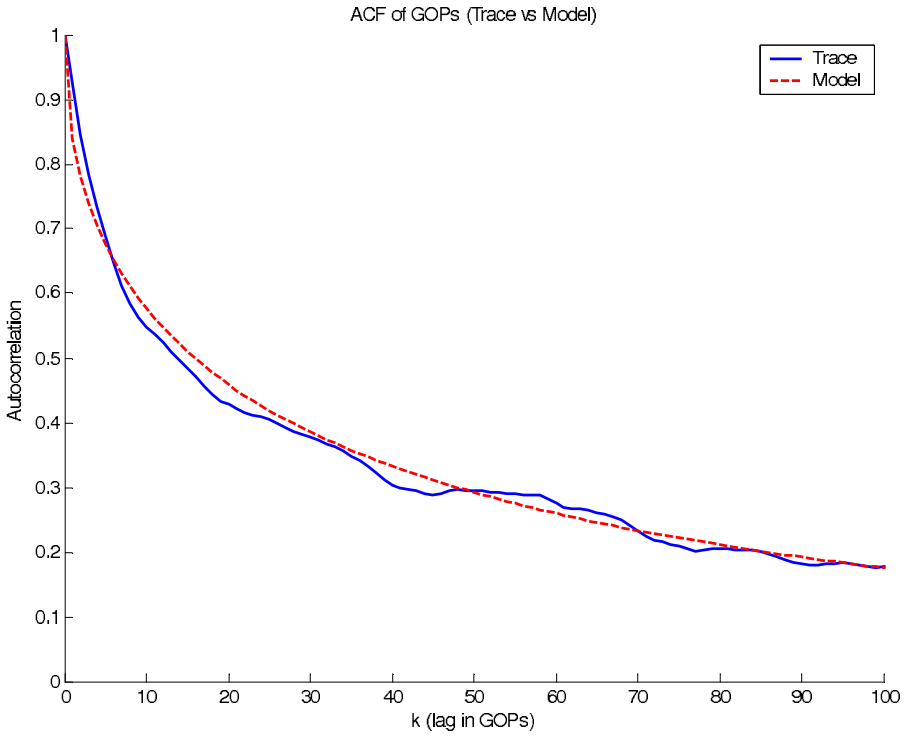


Figure 5.14: Autocorrelation function of GOPs for MPEG4 (Trace vs Model)

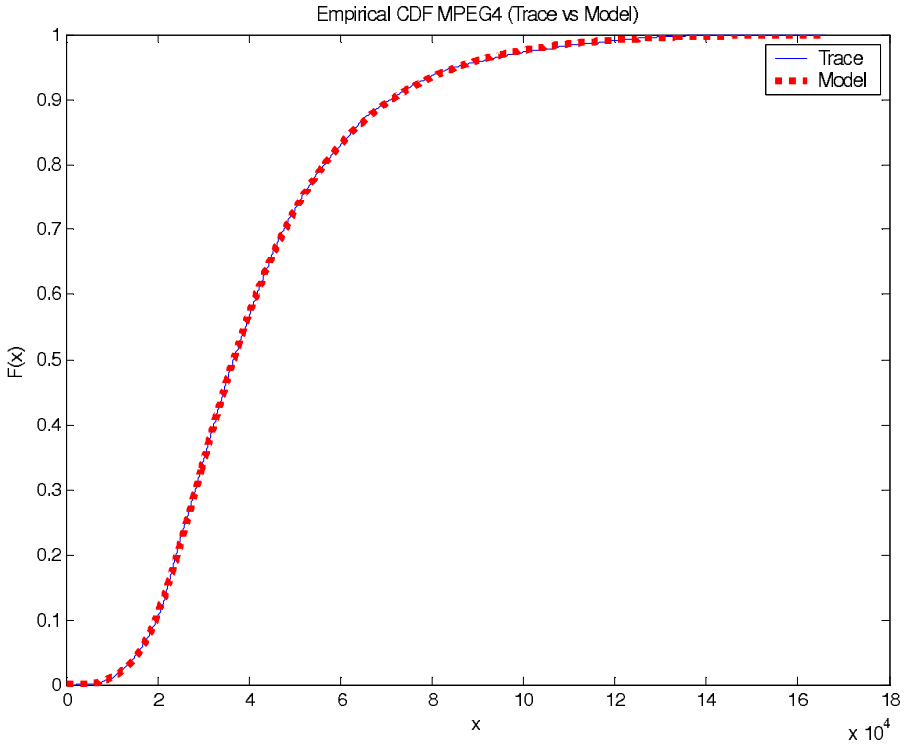


Figure 5.15: CDF function of GOPs for MPEG4 (Trace vs Model)

Table 5.9: Load in Packets and Loss Rate for $\rho = 0.9$

Stats	**Trace**	**Model**
Loss rate %	3.5	3.3
Queue Load (Packets)	14.1	13.7
Mean Rate (Kbps)	734	730

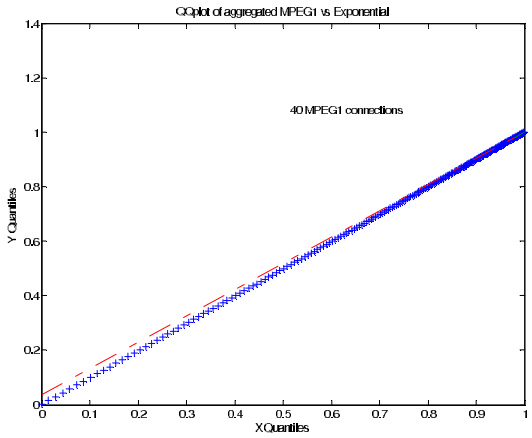
Results show a good performance of the video model. It is important to note that video traffic has constant packet sizes of 1000 bytes, which explains the good performance of the model compared to IP traffic case (where packet sizes are variable). Validation tests are good globally. Generated traffic respects very well the dynamic profile of video trace. In the following sections, we use these traffic generators to characterize the superposition of video applications traffic.

5.3.4.2 Homogeneous Superposition of Video Applications

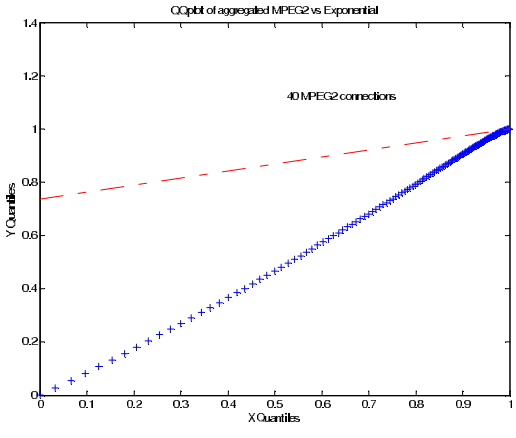
Using previous traffic models we study the superposition of N connections of each codec. In particular, we want to know whether the exponential approximation can be used in the case of superposed video traffic and under which conditions.

We superpose N MPEG connections with increasing number N. We find that packet inter-arrivals of superposed traffic tend to exponential distribution, with slower convergence than for audio applications. However, there are differences in the convergence speed between MPEG1, MPEG2 and MPEG4 codecs. MPEG1 and MPEG4 traffics converge more quickly to exponential distribution than MPEG2 traffic. On Figure **5.16** (**A,B** and **C**) we show the QQplot of MPEG codecs traffics for 40 superposed connections each. We show also the IDI value in each case (Figure **5.16-D**). For only 40 MPEG connections the exponential approximation is not valid for all codecs. Particularly, MPEG2 superposed traffic presents significant difference from the exponential distribution comparing to MPEG1 and MPEG4. Actually, the exponential distribution was validated on MPEG1 for more than 80 connections and on MPEG4 for more than 90 while for MPEG2 it requires more than 160 connections. These differences could be explained by the values of corresponding IDI index for each codec. The cumulated covariance of packet inter-arrivals is much stronger for MPEG2 traffic than for MPEG1 and MPEG4. This result was confirmed by the estimation of the Hurst exponent on the three resulting traffic traces. We obtained H=0.77 for MPEG1, H=0.86 for MPEG4 and H=0.94 for MPEG2.

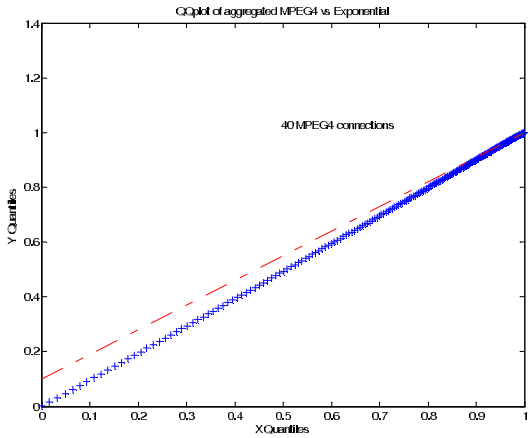
Although the exponential distribution is valid for MPEG coded video traffic at higher number of connections N, the same problem of cumulated packet inter-arrivals covariance must be considered. Indeed, the exponential approximation does not hold under heavy traffic loads in network simulation environment. Therefore we may model the aggregation of N MPEG connections by $M/G/\infty$ process, for which we estimate the correlation structure of the resulting superposed traffic as well as the probability distribution for the considered time slots as we have done for IP traffic in **Chapter 4**.



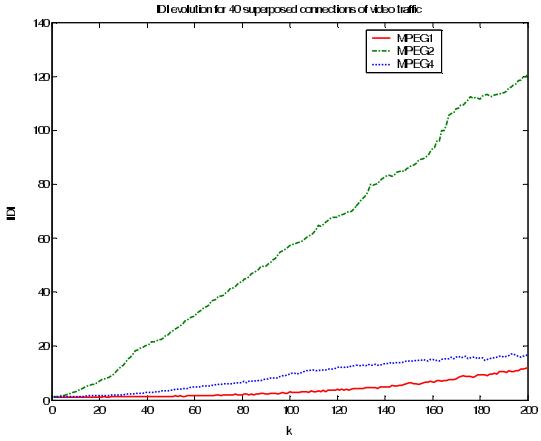
A: QQplot for Superposed MPEG1 vs Exponential



B: QQplot for Superposed MPEG2 vs Exponential



C: QQplot for Superposed MPEG4 vs Exponential



D: Evolution of IDI for Superposed MPEG Traffic

Figure 5.16: Superposed MPEG Traffic Characterization

5.4 Data Applications

Data traffic refers in general to traffic transported by TCP/IP. It concerns mainly the Web activity but includes also File Transfer Protocol (FTP), e-mail .... The HyperText Transfer Protocol (HTTP) is widely used to transfer Web pages over the Internet. Indeed, the HTTP traffic represents 75% of the TCP/IP flows on the Internet. Two HTTP protocols are currently available: HTTP 1.0 [BLFF96] and HTTP 1.1 [FGM+97].

A typical Web page consists of a Hypertext document with links to other objects that make up the whole page. The file containing an HTML document is referred to as a main object and the objects (image, sound, icon, ...) linked from the Hypertext document are referred to as in-line objects. In the standard version of HTTP 1.0, the browser opens multiple TCP connections to download objects of a web page. Thus, the browser opens consecutively parallel connections for in-line objects after the first connection for the main object. On the other hand, in Version 1.1 of HTTP, one TCP connection is used to load the web page, the requests are pipelined and the objects are downloaded back-to-back on the same connection.

5.4.1 Data Application Modelling

The client-server interaction in data applications in general and web application in particular is best modelled by a succession of activity and idle periods. User behaviour and the type of application determine the characteristics of active and idle periods. We will present in the following sections behavioural traffic models for major data applications.

5.4.1.1 Web Traffic

Web applications are typically ON-OFF processes (see [CL99] for example). The ON period represents the activity associated with a page downloading whereas the OFF period represents a thinking time or reading time of the user. Meanwhile, the ON-OFF process can range from very simple to very complex according to the level at which application details are modelled. Thus, a Web page can be modelled as a whole ON period or as a succession of ON and OFF periods representing in-line objects.

We introduce the notion of active ON periods (for user requests) and passive ON periods (for object downloading), and active OFF periods (for time needed by the browser to interpret the HTML code) and passive OFF periods (for viewing or thinking time taken by the user). Using the previous definitions a general Web traffic model can be used as illustrated on Figure **5.17**.

Active ON periods are very short in general (few hundreds of bytes). Passive ON periods concern file downloading and their durations are completely controlled by the underlying transport protocol (TCP). On the other hand, active OFF period refers to interpretation, formatting or displaying time of an object and is less than 1 sec in general, while passive OFF periods are more than 30 sec which is the minimum time needed to read a Web page. Of course Web traffic models may not consider all of these parameters.

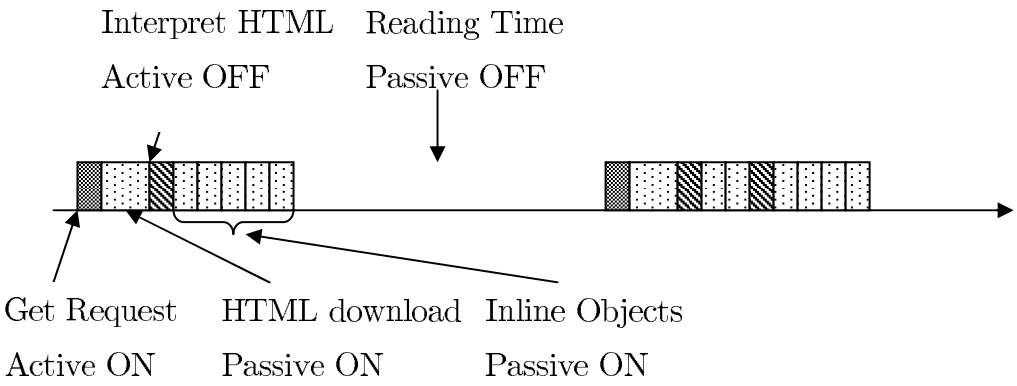


Figure 5.17: Generalized Web Application ON-OFF Model

Many Web models may be limited to the page level with a simple one level ON-OFF model.

Notice that FTP applications concern big file transfer activity. In the FTP application model, only the ON period is significant. Indeed, FTP can be modelled by an ON-OFF process with OFF period of 0 sec.

5.4.1.2 Email Traffic

Email application is one of the most used on the Internet. The data volume corresponding to this traffic is not very important compared to Web and FTP traffic. The Email traffic can be modelled by an ON-OFF process, if we consider the interaction between email client and email server. Practically, email traffic can be seen as small file transfer activity where only the ON period corresponding to message transfer is considered. Several studies proposed empirical distributions for the size of ON period in email applications. We give here for example the proposed distribution by Brasche and Walke [BW97] for the size of ON period corresponding to email application on GPRS networks.

$$f(x) = \left\{ \pi\beta \left[1 + \left(\frac{x-\alpha}{\beta} \right) \right]^2 \right\}^{-1} \tag{5.28}$$

With $\beta = 1.0$ and $\alpha = 0.8$

5.4.1.3 WAP Applications

Wireless data traffic models are of the same nature as wireline data traffic models. In fact the ON-OFF process is also valid to represent user activities in the wireless domain. The Wireless Application Protocol (WAP) is intended to provide Internet access to low rate wireless connections. In the WAP architecture data is organized in Decks instead of pages. Decks are smaller in size and contain hyperlinks to other decks. The WAP traffic can be adequately modelled by an ON-OFF process.

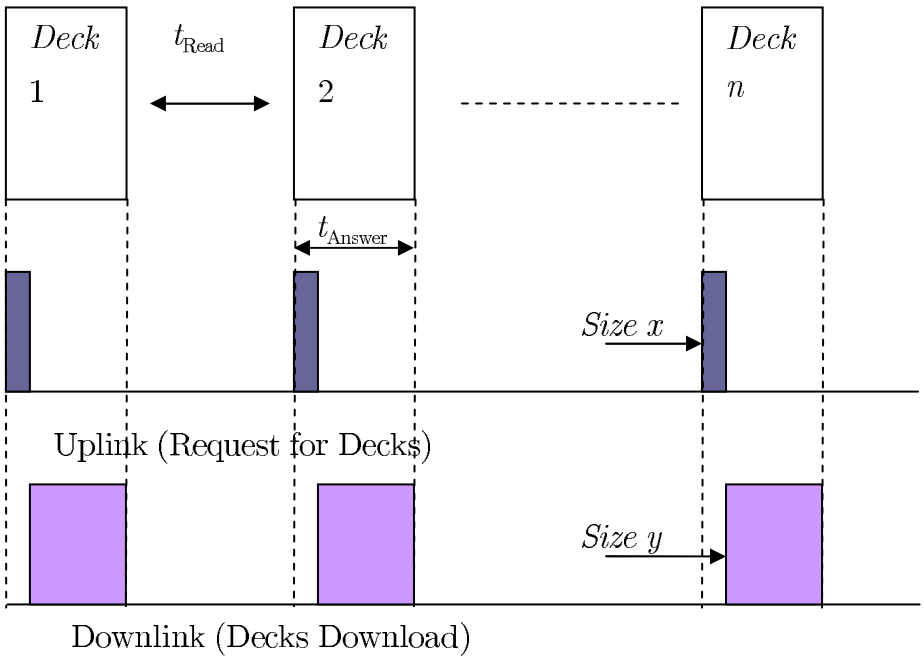


Figure 5.18: WAP Application Model

A WAP session consists of several requests for decks performed by the user. The sequence chart of a session is depicted in Figure **5.18**. WAP sessions are totally described by:

- The requests for a number of decks, n.
- The packet size in uplink and downlink, size x and size y.
- The time needed by the user before requesting the next deck, t_{Read}.
- The response time of the network, t_{Answer}.

5.4.1.4 Numerical Values

In the literature we find very different numerical values for the ON-OFF models representing data applications (Web, WAP, ...). The difference is justified by the diversity of data sets used to estimate parameters as well as networks used to collect data. However, the general tendency observed reveals a heavy tailed distribution of file sizes (or ON periods). Of course, no particular estimated data set can be generalized to model data applications on all networks. Indeed, traffic modellers may use their own collected data sets. Using the estimation tools of the Generic Framework for Traffic Modelling presented in **Chapter 3**, one can estimate easily numerical values of data application parameters based on the behavioural model presented before. Refer to [BW97, SLTG00, KLL01, SFB01] for numerical values examples. In the next section, we will focus on simulation issues concerning ON-OFF applications transported by TCP (data applications).

5.4.2 Characterization of Data Application Traffic

Traffic characterization of audio and video applications can be done at the source level while results are valid for all network configurations. This is because audio and video traffics are usually transported by UDP. Conversely, data applications are generally transported by TCP and the elastic behaviour of TCP changes the characteristics of traffic in function of network congestion conditions. As a consequence, the network congestion factor (or packet loss rate) must be considered in data traffic characterization. In the following sections we analyze the factors that influence traffic LRD property, and then we study the influence of packet loss rate on LRD behaviour.

5.4.2.1 Impact of file size distribution on LRD behaviour

Data traffic concerns file transfer in most cases. Many studies (e.g. [PKC96, WPT98, KC03]) related traffic LRD property to file size distribution on Web servers. It has been argued that the heavy tailed nature of file size distribution on Web servers has an important impact on the resulting self-similarity of traffic (and by consequence traffic burstiness).

We investigate the relationship between file size distribution and traffic self-similarity. We choose three types of distributions for file sizes: Pareto as a heavy tailed distribution, Normal and Exponential as non heavy-tailed distributions (see Table **5.10**). The traffic corresponding to each type of these Web sessions is characterized separately. Indeed, we form three separate flows by generating Web sessions corresponding to each of previous Web models with Poisson arrival process of sessions. Each session ends when all files within the session are transmitted. Session arrival rate is taken 0.33 Session/sec.

Table 5.10: Web Session Models

HTTP Source Model	**Pages Number Distribution**	**Pages Number Mean**	**Pages Number Var**	**Distribution ON OFF**	**ON Mean KB**	**ON Var KB**	**OFF Mean Sec**	**OFF Var Sec**
W1	Normal	10	5	Pareto	200	400	30	60
W2	Normal	10	5	Normal	200	400	30	60
W3	Normal	10	5	Expo	200	200	30	30

Using a network of two routers: Source and Destination (see Figure **5.19**), we generate traffic corresponding to each flow using TCP New Reno implemented from end to end in the simulator. The link bandwidth between the two routers is B Mbps, where B is a parameter of the simulation. A constant link delay of 10 ms is considered with a buffer size of 64 packets at each router. LRD behaviour is characterized by estimating the Hurst exponent using the R/S method on the resulting traffic trace per flow.

First, we evaluate the impact of file size distribution on traffic LRD behaviour in a loss free transmission. For this purpose we use a bandwidth link $B = 100Mbps$. The

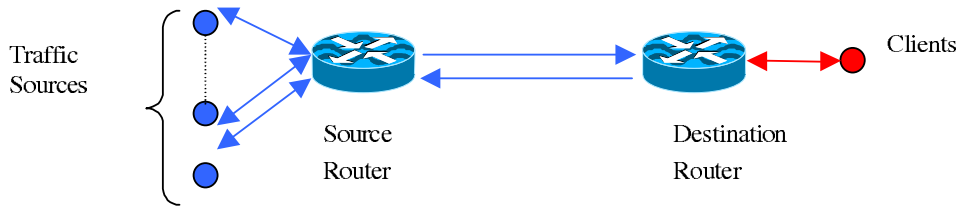


Figure 5.19: Web Session's Simulation Network

impact of packet loss rate on LRD behaviour will be examined later. The simulation results are listed in Table **5.11**.

Table 5.11: Impact of File Size Distribution on LRD Behaviour

Flow	Rate (Kbps)	Loss %	HURST	Link Bandwidth (Mbps)
W1	5294	0	0.81	100
W2	5210	0	0.74	100
W3	5190	0	0.7	100

Although the distributions used in W2 and W3 Web session models are not heavy tailed (Normal and Exponential), the resulting traffic shows a significant Hurst exponent (0.74 and 0.7). It seems that the heavy tailed nature of file size distribution is not the main reason behind LRD behaviour of the superposed traffic. The aggregation of ON-OFF type sessions can produce LRD traffic regardless the type of file sizes or idle time distributions.

5.4.2.2 Impact of packet loss rate on LRD behaviour

We repeat the same experiment as in previous section while reducing the bandwidth of the link B to induce buffer overflows and by consequence packet losses. We estimate the Hurst exponent of the resulting traffic for each flow. We find that the value of H tends to increase with loss rate. For example, the W2 flow exibits a Hurst exponent of 0.74 with 0% packet loss rate, 0.79 with 1.7% packet loss rate, and 0.84 with 4% packet loss rate. It indicates that the Hurst exponent is not stable when loss rate is different from 0%.

In order to explain this behaviour we measured the duration of each session during the simulation. Figure **5.20** depicts the shape of empirical probability distribution function (PDF) of session duration for W2 web session for three packet loss rates: 0%, 1.7% and 4%. We note that W2 session durations are more variant with significant loss rate. Indeed, the normal file size distribution should result in normal session duration distribution. We can clearly see that this distribution tends to heavy tailed one when loss rate is important. The TCP algorithm makes session durations longer because of retransmissions. We note that the session duration is highly variable with increasing loss rate. The high variability of the duration is the key behind the instability of long range correlations of resulting traffic. This behaviour is coherent with Taqqu, Willinger and

Sherman [TWS97] theoretical result about self-similarity: "the superposition of many ON-OFF sources with strictly alternating ON and OFF periods and whose ON periods or OFF periods exhibit high variability (heavy-tailed) can produce aggregate network traffic that exhibits long range dependence or self-similarity".

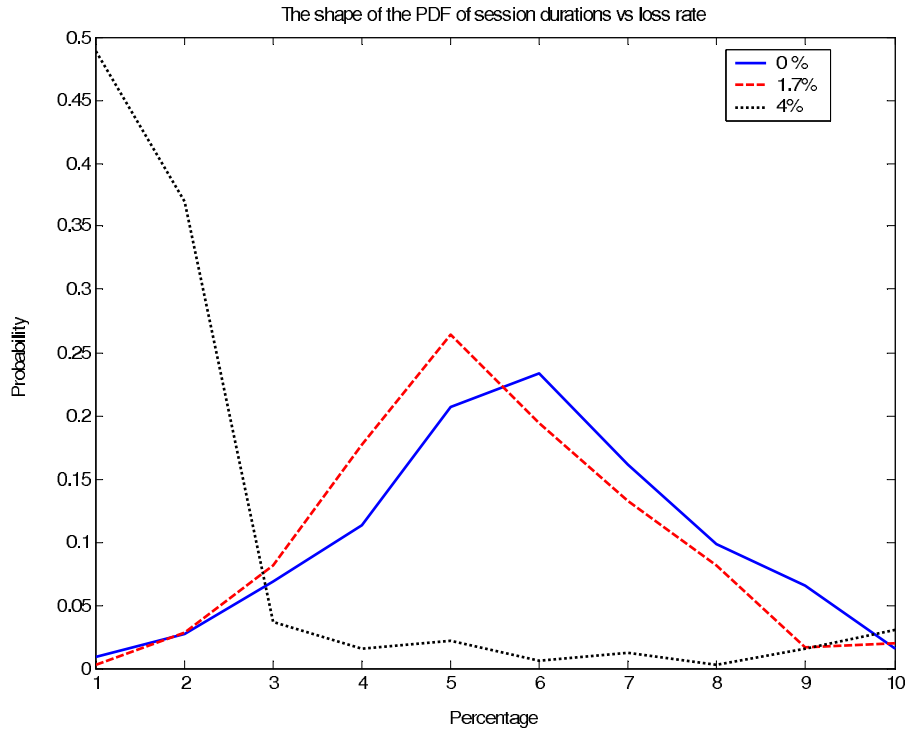


Figure 5.20: The Shape of Empirical PDF of Session Durations in Function of Packet Loss Rate

It seems that the association between LRD property of data traffic and the heavy-tailed nature of file sizes on web servers is not accurate. In fact, when the underlying protocol is UDP, a heavy-tailed file distribution will result in heavy-tailed ON period duration. When it comes to TCP, the situation is very different. File size distribution may be of any kind. The resulting ON period durations may be heavy-tailed because of TCP closed-loop behaviour and by consequence traffic exhibits LRD correlations.

In the last example we were reducing the link bandwidth B to induce losses. Reducing the link bandwidth has a smoothing effect on output traffic. In fact, traffic bursts can not be transmitted as bandwidth offered is limited. However, the characterization of resulting traffic using the Hurst exponent shows instability on this parameter while less bursty traffic was expected. As an alternative we evaluate the IDI of traffic. Results are different in this case (See Figure **5.21**).

The value of IDI decreases when bandwidth is reduced (as expected). In fact, TCP traffic without losses shows the highest IDI value (solid line) where for k=200 we have an IDI value of 500 for loss free traffic, versus 40 (resp. 20) for lossy traffic of 1.7% loss rate (resp. 4% loss rate). Recall that the more important the value of IDI the more

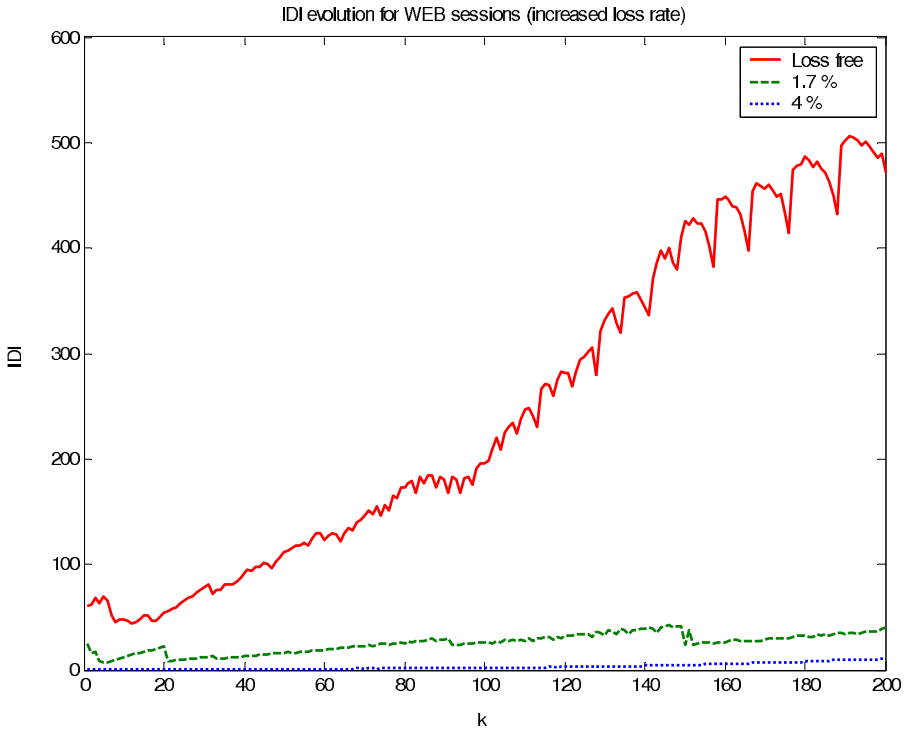


Figure 5.21: Evolution of IDI for Superposed Web Session Traffic

covariance on packet inter-arrivals is cumulated, and the traffic is more bursty. In fact, the IDI characterizes better traffic burstiness.

From a traffic modelling point of view, this analysis showed that the correlation structure in aggregate TCP traffic is not stationary. As a consequence, any proposed model to generate aggregate TCP traffic must take this property into account.

5.4.2.3 Superposed Data Traffic Modelling

Aggregate TCP traffic presents strong correlations as seen in previous section. However, it can not be modelled using usual LRD traffic models (such as $M/G/\infty$), because of the closed-loop behaviour of the underlying transport protocol TCP that influences packet inter-arrival correlations.

Two approaches are proposed to model aggregate Web traffic. First, we consider another simulation technique that allows simulating TCP connections quickly using differential analytical modelling (see **Chapter 6** for more details). Second, we keep the key idea of superposing Web sources (ON-OFF processes), while trying to reduce the number of superposed ON-OFF processes using equivalent ON-OFF process. The second approach will be developed hereafter.

Equivalent ON-OFF process Consider a simple ON-OFF process with $T_{OFF} \gg T_{ON}$ (a typical Web application). We need to generate N simple ON-OFF processes simultaneously (in other words, we superpose N simple Web applications). The goal is to replace

the N simple ON-OFF processes by $M \ll N$ equivalent ON-OFF processes to reduce the computational complexity. However, the traffic generated by the M equivalent ON-OFF processes should preserve the same statistical behaviour and the same performances as the traffic generated by the N simple ON-OFF processes.

The idea is to aggregate the long OFF periods in the simple ON-OFF process into shorter OFF periods in the equivalent ON-OFF process. This should be done under the condition: the average resulting throughput is the same in both cases. Indeed, the aggregation factor of OFF periods and the aggregation factor of superposed processes are calculated under this condition. Consider:

- T_{OFF} is the duration of the OFF period in the simple process.
- T_{OFF}^{Equ} is the duration of the OFF period in the equivalent process.
- A_{OFF} is the aggregation factor of OFF periods.
- A_{Agg} is the aggregation factor of ON-OFF processes.

Then we have:

$$T_{OFF}^{Equ} = \frac{T_{OFF}}{A_{OFF}} \tag{5.29}$$

Notice that the equivalent ON-OFF process have the same ON period characteristcis as the simple ON-OFF process (no aggregation for ON periods). A_{Agg} can be written as:

$$A_{Agg} = \frac{N}{M} \tag{5.30}$$

The average throuput of superposed simple ON-OFF processes and superposed equivalent ON-OFF processes must be equal:

$$\lambda_{Equ} = A_{Agg} * \lambda \tag{5.31}$$

We replace $\lambda = \lambda_{ON} * P_{ON}$ and $\lambda_{Equ} = \lambda_{ON}^{Equ} * P_{ON}^{Equ}$. With:

- λ_{ON} is the throughput during the ON period of the simple ON-OFF process.
- λ_{ON}^{Equ} is the throughput during the ON period of the equivalent ON-OFF process.
- $P_{ON} = \frac{T_{ON}}{T_{ON}+T_{OFF}}$ is the occurrence probability of ON period in the simple ON-OFF process.
- $P_{ON}^{Equ} = \frac{T_{ON}}{T_{ON}+T_{OFF}^{Equ}}$ is the occurrence probability of ON period in the equivalent ON-OFF process.

However, $\lambda_{ON} = \lambda_{ON}^{Equ}$ as we conserve the same ON period while we perform aggregation on OFF periods. So, we get:

$$A_{Agg} = \frac{\lambda_{Equ}}{\lambda} = \frac{P_{ON}^{Equ}}{P_{ON}} \tag{5.32}$$

$A_{Agg} = \frac{N}{M}$ represents the gain we obtain by applying the equivalent ON-OFF process. We can rewrite the last equation as:

$$N * P_{ON} = M * P_{ON}^{Equ} \tag{5.33}$$

This equation can be interpreted like this: The superposition of simple ON-OFF processes generate the same average number of active periods (ON) as the superposition of equivalent ON-OFF processes. Recall $N*P_{ON}$ is the average of a binomial distribution (which is the distribution of ON periods in ON-OFF processes).

The algorithm Practically, we need to determine the following parameters: A_{Agg}, P_{ON}, $P_{ON}^{Equ}$and M in function of N and A_{OFF} in order to substitute N simple ON-OFF processes by M equivalent ON-OFF processes. The algorithm is presented hereafter:

1. Choose the value of A_{OFF} so that $T_{OFF}^{Equ} > T_{ON}$
2. Calculate $P_{ON} = \frac{T_{ON}}{T_{ON}+T_{OFF}}$
3. Calculate $P_{ON}^{Equ} = \frac{T_{ON}}{T_{ON}+T_{OFF}^{Equ}}$
4. Calculate $A_{Agg} = \frac{P_{ON}^{Equ}}{P_{ON}}$
5. Calculate $M = \frac{N}{A_{Agg}}$
6. Generate M ON-OFF processes with the same ON period and equivalent OFF period with T_{OFF}^{Equ} duration

Numerical validation We use the previous algorithm to substitute N superposed ON-OFF processes (using the W1 web model defined in Table **5.10**) by $M \ll N$ equivalent ON-OFF processes. The numerical values are listed in Table **5.12**.

Table 5.12: Aggregated Web Sessions Example

N	A_{OFF}	T_{OFF}^{Equ}	A_{Agg}	M
1000	100	0.3 sec	17.5	57

We generate the N superposed simple processes as well as the corresponding M equivalent processes. We compare the traffic generated in both cases statistically using the IDI index. Then we compare their performance in a queuing system of deterministic service.

The IDI evolution for both simple and equivalent processes traffic is depicted on Figure **5.22**. The statistical evaluation shows a similar behaviour of simple and equivalent processes. The IDI of superposed equivalent processes follows very well the IDI of superposed simple processes.

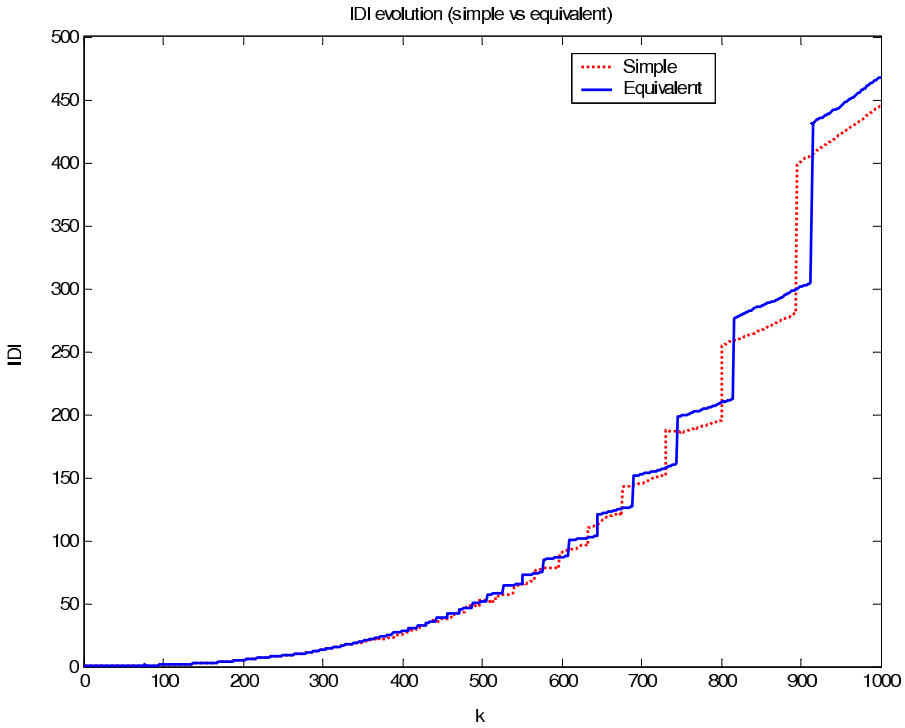


Figure 5.22: Evolution of IDI: Simple vs Equivalent Processes

The performance of the generated traffic is evaluated in a queuing system of deterministic service. Both simple and equivalent traffic performances are compared under high queue load. Results are good and validate the use of the equivalent ON-OFF process (Table **5.13**).

Table 5.13: Performance of the Equivalent ON-OFF Process

Traffic	**Bit Rate (Kbps)**	**Average Load (Packets)**	ρ
Simple	2425	2.4	0.9
Equivalent	2490	1.9	0.9

A good approximation is realized using the equivalent ON-OFF process. The aggregation of OFF periods performs well and can be used to reduce computational complexity when large number of identical Web sessions needs to be simulated (in the previous example we reduced the number of sessions to be simulated from 1000 to 57). However, its performance decreases when very large number of sessions is considered as the number of equivalent processes may still be large. A better solution to the aggregated TCP modelling issue will be detailed in **Chapter 6** when a new differential simulation technique is presented.

5.5 Conclusion

In this Chapter, we presented application models for Audio, Video and Data applications. Application models take into consideration the user behaviour and provide reliable packet generation process according to application type. The detailed application models are used to characterize multimedia applications traffic. Hence, the packet inter-arrival process is studied on single and superposed applications. We found that the exponential approximation of packet inter-arrival process is valid for the superposition of audio applications and MPEG videos under light to medium loads of traffic. This approximation is very convenient as it allows analyzing analytically the performance of superposed traffic in queuing networks. However, we find that this approximation is not valid under heavy loads of traffic and we propose other approximations instead. On the other hand, the superposition of data applications present dynamic correlation structure in function of packet losses. This is due to packet retransmission mechanisms implemented by TCP (the underlying transport protocol). We model the superposition of web applications by an equivalent ON-OFF process. The equivalent ON-OFF process reduces the simulation complexity by aggregating OFF periods. Although the proposed model performs well, its performance is limited when very large number of sources is considered as the number of equivalent processes to superpose may still be very large. In **Chapter 6** we present a more efficient solution via differential simulation technique of TCP/IP. The proposed technique allows simulating TCP/IP sources by fluid rate propagation, while preserving the transient behaviour of TCP sources.

Chapter 6

TCP/IP Differential Analytical Modelling

6.1 Introduction

Transmission Control Protocol on IP (TCP/IP) plays an important role in the Internet. Most end-to-end reliable connections on the Internet are established by TCP/IP. In-order delivery of packets, lost packets retransmission and the efficient use of bandwidth are functionalities implemented in TCP/IP, and they are behind its success. However, the numerous functionalities of TCP/IP resulted in a sophisticated algorithm. Hence, from a traffic modelling point of view, the reliability of TCP/IP generates "elastic" traffic because of packet retransmission mechanisms. No simple traffic models could be used to generate TCP/IP traffic unless the TCP/IP loss process is reproduced.

Event-driven technique is widely used to simulate TCP/IP. Unfortunately, the increase in the number of generated events makes it unsuitable for large scale network simulations. Many other techniques to simulate the behaviour of TCP/IP analytically are proposed in the literature. Most of them are based on analytical stationary approximations of rate and loss process. However, such approaches do not reproduce the transient behaviour of TCP/IP. Our objective is to model TCP/IP analytically to overcome scaling problems while preserving the TCP/IP transient behaviour for more precision. We achieve this using the Differential Traffic Theory [GGB+01].

In this Chapter, we present a differential model for TCP/IP. The model describes precisely the behaviour of TCP/IP by fluid differential equations mixed with control events. Control events pilot the simulation to pass from one differential equation to another. Network nodes are represented by $D(t)/D/1/N$ queues, while $D(t)$ means transient deterministic arrival. Losses and delays are evaluated analytically. The Chapter is organized as follows: in section **6.2**, we give an overview of TCP/IP with its different operation modes. In section **6.3**, we present the differential analytical modelling technique as well as its application to TCP/IP. Finally, in section **6.4**, we validate the model by comparing differential simulation results with event-driven simulations in different topologies.

6.2 TCP/IP Overview

TCP/IP was designed to build over the simple network layer to provide a reliable in-order data delivery service to the application layer. We can summarize the main objectives of TCP/IP protocol by [Pos81, Bra89, Jac88, BA00, Ste97]:

- End-to-end flow control
- Error control
- Congestion control

6.2.0.4 End-to-End Control

The end-to-end control guarantees that the sender does not inject into the network more data than the receiver can hold in its buffer. This is achieved by a window whose value is advertised during connection set-up and it is updated during the connection lifetime if the buffer space at the receiver changes.

6.2.0.5 Loss (or Error) Control

The loss control means that TCP/IP is responsible of information recovery in case of loss. This is done by the retransmission of the lost information. Loss detection in TCP/IP is based on sequence numbers and acknowledgments (ACKs). An ACK carries implicitly the sequence number of the last in-order packet received. The TCP/IP receiver acknowledges the reception of one data packet over two in general. This is called the delay mechanism and its objective is to reduce the volume of generated ACKs. The TCP/IP receiver transmits a duplicate ACK for every out of order packet. Upon the reception of three duplicate consecutive ACKs, the source realizes that a packet has been lost (this is called the Fast Retransmit algorithm). In fact, the Fast Retransmit algorithm was introduced to enhance the performance of TCP/IP. Before the loss detection was only possible via the retransmission timer called *timeout*. This timer is initialized before transmission and recalculated during the connection life time. The expiration of this timer before the reception on an ACK is a loss signal for the sender.

6.2.0.6 Congestion Control

The congestion control means that different sources adapt their transmission rates as a function of the network load. The objective is the good utilization of network resources with fairness between different flows. The congestion control algorithm adopted by TCP/IP is an additive-increase with multiplicative-decrease algorithm. The rate of TCP/IP source is controlled by a window-based approach. This window is called *Congestion Window*. The rate control in TCP/IP is achieved by changing the size of the congestion window. The rate on a TCP/IP connection can be approximated by the window size (Or the number of packets the source has in the network) divided by the round-trip time (RTT).

The injection of an application's traffic is dynamically governed in function of RTT, and effects of congestion in the network as reflected in timeouts and lost packets. RTT estimates are important performance parameters in a TCP/IP exchange, especially when considering a large transfer. If the RTT estimate is too low, packets are retransmitted unnecessarily; if too high, the connection can sit idle while the host waits to timeout. TCP/IP source uses ACKs to estimate the RTT of the connection and sets accordingly a timer when packets are sent. An Exponentially Weighted Moving Average algorithm is used to smooth the oscillations in the RTT.

The average transfer rate, also called the throughput, is the main performance measure that indicates how well a bulk TCP/IP transfer is done.

6.2.1 Notations

All notations given here concern the TCP/IP *NewReno* version. We develop our model and do our simulations based on this version. However, the extension of our model to other versions can be done very easily.

All values unites are in Bytes. We denote:

- **MSS (Maximum Segment Size):** The maximum segment size (MSS) is the largest amount of data in one packet that can not be exceeded regardless how large the current window is.
- **SEQ (Sequence):** SEQ is the address of the next packet to send.
- **ACK (Acknowledgment):** The value of the last received ACK which corresponds to the address of the next packet to be sent.
- **CWND (Congestion Window):** This window corresponds to the maximum data volume present on the network between sender and receiver. It indicates the output rate per RTT.
- **RWND (Receiver Window):** This window corresponds to the maximum data volume that could be held in the receiver buffer.
- **NDUP (Number of Duplicate ACKs):** NDUP is a counter of the number of identical ACKs received by the sender.
- **RTT (Round-Trip Time):** The RTT is the time measure between the moment of sending a packet and the moment of the reception of its corresponding ACK.
- **RTO (Retransmit Time-Out):** The RTO is the estimation of the maximum delay allowed between the sending of one packet and the reception of its corresponding ACK.
- **CREDIT** The *CREDIT* is a calculated value representing the number of packets the source is allowed to transmit. It is given by the following equation:

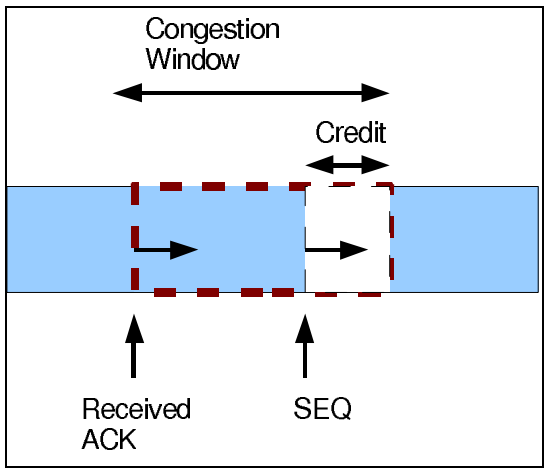


Figure 6.1: Sending Credit and Congestion Window

$$CREDIT = ACK + \min(CWND, RWND) - SEQ + NDUP * MSS \quad (6.1)$$

The establishment of a TCP/IP connection starts always by the transmission of a SYN packet and the reception of the corresponding ACK. This gives the source a credit of two packets to start the transmission of data. Equation (6.1) governs the evolution of the credit on the sender side. To understand the evolution of the credit Figure **6.1** illustrates the credit equation (without considering *RWND* or *NDUP*).

The reception of one ACK increases the sending credit while the transmission of one segment decreases the credit. TCP/IP algorithm uses the reception rate of ACKs to control its sending rate by changing the sending credit and the congestion window accordingly. When the receiver gets a packet, SEQ gives the address of data in the receiver buffer. On the receiver side ACK value is calculated based on the data volume contained in the received packet and the SEQ value.

However, upon packet reception the data could not be processed in the following cases:

- The address of the end of data (SEQ+packet size) is less than ACK at receiver.
- The address of the beginning of received data is bigger than (ACK+RWND) at receiver.

Otherwise, data is placed into the receiver buffer and a new ACK value is calculated. In all cases the next ACK to be sent contains the address of the beginning of next data to be received.

We develop in the following sections the operation modes of TCP/IP. This behavioural description is important to understand the differential analytical model of TCP/IP.

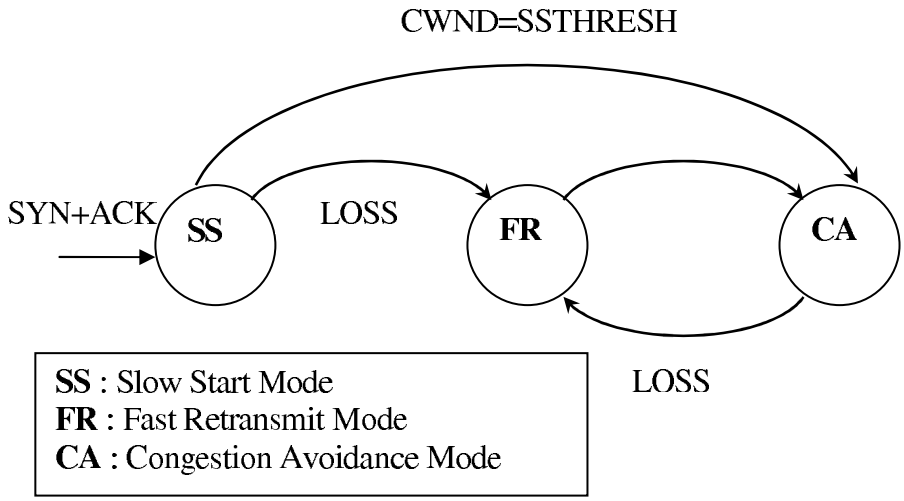


Figure 6.2: Simplified TCP/IP State Transitions

6.2.2 Operation Modes

TCP/IP has three principal operation modes: Slow Start, Congestion Avoidance and Fast Retransmit. The state machine of TCP/IP operation is depicted on Figure **6.2**. In the following sections we explain the operation of each mode as well as the events causing the transition from one state to another.

6.2.2.1 Slow Start

When a new connection is established, the congestion window is initialized to one segment. Each time an ACK is received, the congestion window is increased by one segment. The sender starts by transmitting one segment and waiting for its ACK. When that ACK is received, the congestion window is incremented from one to two, and two segments can be sent. When each of these two segments is acknowledged, the congestion window is increased to four and so forth, providing an exponential growth. Indeed, it is not exactly exponential because the receiver may delay its ACKs. Typically, the receiver sends one ACK for every two segments. The delayed ACK parameter is called b and its value is 2 in general.

During Slow Start TCP/IP sender transmits packets by bursts. The size of bursts is function of the delayed ACK parameter. For each b packets received one ACK is sent from the receiver to the sender. The number of packets sent in one burst follows approximately a geometric series of $\frac{b+1}{b}$ reason. Indeed, the exponential behaviour of packet rate during Slow Start mode of TCP/IP is due to this evolution. However, this exponential behaviour is only an approximation because the actual packet rate is controlled by the service rate of the slowest router on the packets path denoted $\mu_{\min}$. ACKs arrive to the sender at this maximum rate and the increase of the Congestion Window CWND is exponential until the reception rate of ACKs reaches this value where it becomes linear not exponential.

6.2.2.2 Congestion Avoidance

Congestion Avoidance is implemented to deal with lost packets. There are two indications of packet loss: a timeout event and the reception of three duplicate ACKs. Although Congestion Avoidance and Slow Start are independent algorithms with different objectives, they are implemented together and use two variables for each connection: a congestion window (CWND), and a Slow Start threshold size (SSTHRESH).

If CWND is less than or equal to SSTHRESH, TCP/IP is in Slow Start, otherwise TCP/IP is performing Congestion Avoidance. When congestion occurs (indicated by a timeout or the reception of duplicate ACKs), one-half of the current window size is saved in SSTHRESH. Additionally, if the congestion is indicated by a timeout, CWND is set to one segment (i.e., Slow Start).

Congestion Avoidance increments CWND by $\frac{b+1}{CWND}$ segment each time an ACK is received. This is a linear growth of CWND, compared to Slow Start's exponential growth. CWND is increased by one segment at most each RTT.

6.2.2.3 Fast Retransmit

The detection of one loss turns TCP/IP into the Fast Retransmit mode. The sender must retransmit all lost packet at the rhythm of one packet per RTT. This mode turns over when all lost packets are retransmitted. The Fast Recovery function is coupled with the Fast Retransmit mode to enhance the performance of TCP/IP. After the TCP/IP sender finishes retransmitting all lost packets, it does not resume in the Slow Start mode. Instead, it turns into the Congestion Avoidance mode with a CWND half its value before loss detection. Figure **6.3** summarizes the TCP/IP operation modes.

6.3 TCP/IP Differential Analytical Modelling

The suggested differential model relies on the transformation of the credit equation (6.1) into differential equation representing the evolution of TCP/IP fluid rate in the network in function of the reception rate of ACKs and events causing the transition from one mode to another.

6.3.1 Related Models of TCP/IP

Event-driven simulation techniques simulate the generation and propagation of TCP/IP packets. Events corresponding to the creation of each packet are generated and a full state context for each packet is conserved from the Source to the Destination. Event-driven simulators (e.g. NS2 [NS201]) are very precise as the packet life cycle is simulated entirely. The problem arises when the number of packets to simulate increases because the number of generated events will increase with the same proportion. As a consequence event-driven simulation techniques are not adapted to large scale networks.

Fluid models suggest propagating rates instead of packets. The flow rate is approximated based on the rules that govern the transmission process. There are different

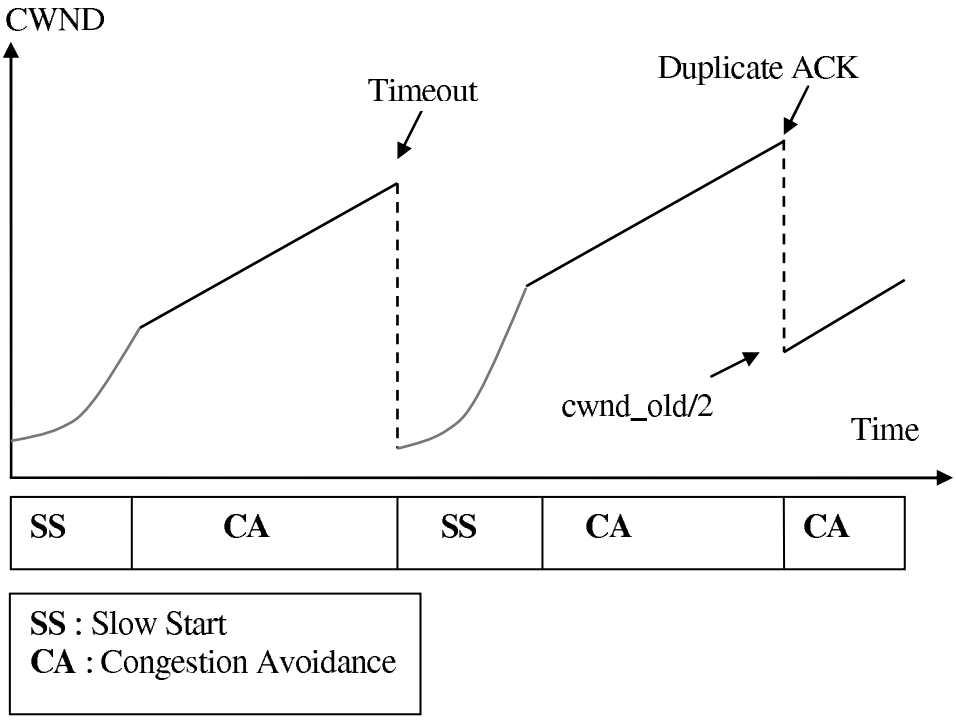


Figure 6.3: TCP/IP Operation Modes

techniques in fluid modelling of TCP/IP. In [AAB00] authors present a fluid model for TCP/IP flow control mechanism. In this model, the rate at which data is transmitted increases linearly in time until a packet loss is detected. At that point, the transmission rate is divided by a constant factor. Losses are generated by random process which is only assumed to be stationary to allow accounting for any correlation and any distribution of inter-loss times. The authors obtain an explicit expression for the throughput of a TCP/IP connection and bounds on the throughput when there is a limit on the congestion window size. The proposed model estimates the loss process and gives stochastic behaviour of TCP/IP, but the transient behaviour can not be obtained.

J.L Gil [Gil05] uses a discrete time Markov chain to model TCP/IP. The presented model does not use a stochastic approach but rather an exact evolution of TCP/IP algorithm using a discrete time markov chain. TCP/IP operation modes and mechanisms are modelled in the mono-source configuration only. Unfortunately, the proposed model does not handle the multi-source or network configuration.

In [MGT99] authors characterize the steady state performance of a TCP flow by stochastic differential evaluation of the congestion window. The proposed model does not rely on a loss process model but rather on a loss event stream arriving at the source. This stream is supposed to be Poisson, based on measurements realized on TCP traces. However, the transient behaviour is not considered in this approach, but a steady state approximation of TCP throughput is given instead.

The differential TCP/IP model we suggest models the TCP/IP algorithm operation

modes in mono-source and multi-source configurations. Besides, it reflects the transient behaviour of TCP/IP while not being as heavy as event-driven implementations of TCP/IP.

6.3.2 Notations

The TCP/IP differential model is based on a set of differential equations for rate in each operation mode of TCP/IP. The differential equations evolve with time. The rate of the TCP/IP source is calculated at each date $t + \Delta t$ as a function of the rate at date t.

Let $\lambda_s(t)$ be the rate of source s at time t, and $N_s(t)$ is the total number of packets transmitted by source. We have:

$$N_s(t) = \int_0^t \lambda_s(t)dt \tag{6.2}$$

Let $\lambda_{s,r}(t)$ be the rate perceived by the receiver of source s at time t, and $N_{s,r}(t)$ is the number of packets arrived to the receiver, then:

$$N_{s,r}(t) = \int_0^t \lambda_{s,r}(t)dt \tag{6.3}$$

Let $\lambda_{s,l}(t)$ be the packet loss rate of source s at time t, and $N_{s,l}(t)$ is the number of lost packets, so we get:

$$N_{s,l}(t) = \int_0^t \lambda_{s,l}(t)dt \tag{6.4}$$

Let us denote $P_s(t)$ the packet loss rate of source s at time t, then we obtain:

$$\lambda_{s,r}(t) = \lambda_s(t) - \lambda_{s,l}(t) = \lambda_s(t).(1 - P_s(t)) \tag{6.5}$$

And

$$P_s(t) = \frac{\lambda_s(t) - \lambda_{s,r}(t)}{\lambda_s(t)} \simeq \frac{N_s - N_{s,r}}{N_s} = \frac{N_{s,l}}{N_s} \tag{6.6}$$

In differential modelling we propagate values from one node to another as function of time. The integration of differential equations is done each Δt (time step) which is considered constant. Propagated values include packet rate, congestion window, Ack rate, ... We assume that all values calculated with a function f verify the following equation:

$$f(t + \Delta t) = f(t) + \dot{f}(t) * \Delta t \tag{6.7}$$

With:

$$\dot{f}(t) = K, \forall t \in [t, t + \Delta t] \tag{6.8}$$

K is a constant. That means we consider the variation of function f is linear during the

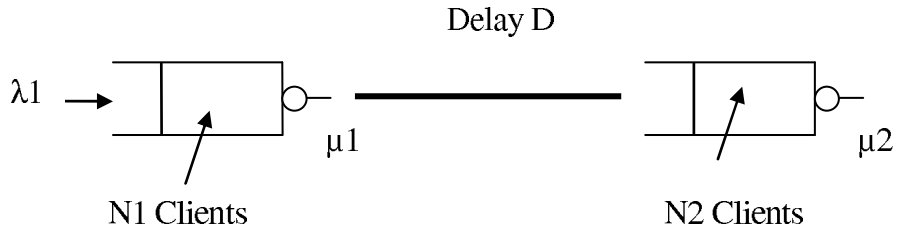


Figure 6.4: Propagation Rules

interval $[t, t + \Delta t]$.

6.3.3 Propagation Rules

The propagation of rates in a network must take into consideration the latency D that exists between different servers due to link delays and server waiting time. On Figure **6.4** we represent two servers with a constant link delay D. The service rate of server1 is μ_1 and the service rate of server2 is μ_2. The source input rate in server1 is λ_1. N_1 and N_2 are the number of clients in queue1 and queue2 (of server1 and server2 respectively).

The number of clients in queue1 evolves according to the following equation:

$$N_1(t + \Delta t) = N_1(t) + (\lambda_1 - \mu_1) * \Delta t \tag{6.9}$$

Server1 can serve $\mu_1 * \Delta t$ data quantity during the integration step. The output rate of server1 is ruled by:

$$\overset{*}{\mu_1} = \begin{cases} \mu_1 & if \quad N_1(t + \Delta t) \geq 0 \\ \lambda_1 & if \quad N_1(t + \Delta t) < 0 \end{cases} \tag{6.10}$$

The output rate of queue1 constitutes the input rate of queue2. The propagation of $\overset{*}{\mu_1}$ to queue2 must be done respecting the latency of the link between two servers. The input rate of queue2 is given by the equation:

$$\lambda_2(t) = \overset{*}{\mu_1}(t - D) \tag{6.11}$$

The integration of differential equation is done each time step Δt. The output rate values must be propagated by Δt. An array of $\left\lceil \frac{D}{\Delta t} \right\rceil$ dimension is created for each server to store the propagated values of its output rate.

6.3.4 Differential Analytical Model of TCP/IP New Reno

In the following sections we develop the differential equations used for each of the operation modes of TCP/IP in both mono-source and multi-source configurations.

6.3.4.1 Slow Start

The number of packets sent by TCP sender in one burst during Slow Start mode follows a geometric series of $\frac{b+1}{b}$ reason. The average packet rate could be approximated at any moment by $\frac{CWND(t)}{RTT(t)}$. Thus, the average packet rate at t_{k+1} (time is measured by RTT units) is related to the average packet rate at t_k by the following equation:

$$\lambda\left((k+1).RTT\right) = \frac{b+1}{b}\lambda(k.RTT) \tag{6.12}$$

Or

$$\lambda_k = \lambda_0 \left(\frac{b+1}{b}\right)^{\frac{k}{RTT}} \tag{6.13}$$

The initial value of λ_0 is given by $\lambda_0 = \frac{b}{RTT}$, so we get:

$$\lambda(t) = \frac{b}{RTT(t)}.e^{\frac{\ln\left(\frac{b+1}{b}\right)}{RTT(t)}.t} \tag{6.14}$$

The derivative from of (6.14) is:

$$\dot{\lambda}(t) = \frac{\lambda(t)}{RTT(t)}.\ln\left(\frac{b+1}{b}\right) \tag{6.15}$$

Equation (6.14) is general and depends only on the value of b (constant equals to 2 in general) and the estimated value of $RTT(t)$. It can be used in multi-source configuration by replacing $RTT(t)$ with $RTT_s(t)$ for each source s. Hence, the relation becomes in multi-source configuration:

$$\lambda_s(t) = \frac{b}{RTT_s(t)}.e^{\frac{\ln\left(\frac{b+1}{b}\right)}{RTT_s(t)}.t} \tag{6.16}$$

$RTT_s(t)$ is estimated separately for each source as we will see later.

The exponential rate given by (6.16) is limited in reality by the reception rate of ACKs and the maximum value of CWND.

ACK Rate Limitation The TCP/IP source rate is controlled by the reception rate of ACKs. In Slow Start mode for each ACK received the source can transmit $b+1$ packets.

The general equation is:

$$\lambda_{r-\max} = (b+1).\lambda_{ack} \tag{6.17}$$

In mono-source configuration it can be simplified considering that $\lambda_{ack} = \frac{\mu_{\min}}{b}$ as one ACK is sent for each b packets, so:

$$\lambda_{r-\max} = \mu_{\min}\frac{b+1}{b} \tag{6.18}$$

In multi-source configuration, the ACK rate is estimated for each source separately. Equation (6.17) becomes.

$$\lambda_{s,r-\max} = (b+1).\lambda_{s,ack} \tag{6.19}$$

CWND Max Limitation TCP/IP source rate increases exponentially in Slow Start mode until a loss occurs. In fact, the TCP/IP source discovers the available bandwidth by increasing its congestion window. If the congestion window CWND value is limited by a maximum value, this increase stops and TCP/IP rate stabilizes on:

$$\lambda_{cwnd-\max} = \frac{CWND_{\max}}{RTT(t)} \tag{6.20}$$

In multi-source configuration we replace $RTT(t)$ by $RTT_s(t)$, and the maximum value becomes:

$$\lambda_{s,cwnd-\max} = \frac{CWND_{\max}}{RTT_s(t)} \tag{6.21}$$

6.3.4.2 Congestion Avoidance

The average rate of TCP/IP in Congestion Avoidance mode is given by:

$$\lambda(t) = \frac{CWND(t)}{RTT(t)} \tag{6.22}$$

In fact the previous equation is a first approximation of the TCP/IP rate in general, expressed by the number of packets transiting between sender and receiver divided by the round trip time. In multi-source configuration we replace $RTT(t)$ and $CWND(t)$ by their values for each source:

$$\lambda_s(t) = \frac{CWND_s(t)}{RTT_s(t)} \tag{6.23}$$

TCP/IP rate is also limited in this mode by the reception rate of ACKs as well as the maximum value of CWND.

ACK Rate Limitation In Congestion Avoidance mode for each ACK received the source can transmit $b + \frac{1}{CWND(t)}$ packet. The general equation is:

$$\lambda_{r-\max} = (b + \frac{1}{CWND(t)}).\lambda_{ack} \tag{6.24}$$

In mono-source configuration it can be simplified considering that $\lambda_{ack} = \frac{\mu_{\min}}{b}$ as one ACK is sent for each b packets, so:

$$\lambda_{r-\max}(t) = \left(1 + \frac{1}{b.CWND(t)}\right).\mu_{\min} \tag{6.25}$$

In multi-source configuration, the ACK rate is estimated for each source separately. Equation (6.24) becomes:

$$\lambda_{s,r-\max}(t) = \left(b + \frac{1}{CWND_s(t)}\right).\lambda_{s,ack} \tag{6.26}$$

CWND Max Limitation If the congestion window $CWND$ value is limited by a maximum value, the TCP/IP rate is bounded by:

$$\lambda_{cwnd-\max} = \frac{CWND_{\max}}{RTT(t)} \tag{6.27}$$

In multi-source configuration we replace $RTT(t)$ by $RTT_s(t)$:

$$\lambda_{s,cwnd-\max} = \frac{CWND_{\max}}{RTT_s(t)} \tag{6.28}$$

6.3.4.3 Fast Retransmit

After the detection of a loss the source turns into Fast Retransmit mode. In this mode packets lost are sent one by one for each ACK received (or one packet per $RTT(t)$). Then the packet rate is given by:

$$\lambda_s(t) = \frac{1}{RTT_s(t)} \tag{6.29}$$

When all lost packets are sent, the source goes into Congestion Avoidance mode with half the value of CWND before the loss.

6.3.4.4 Evolution of CWND

TCP/IP is a window controlled algorithm. The evolution of the congestion window is summarized here:

$$CWND_s(t) = \begin{cases} CWND_s(t) + 1 & SS \\ CWND_s(t) + \frac{1}{CWND_s(t)} & CA \\ 0 & FR \\ \frac{CWND_s(t)}{2} & Loss \end{cases} \tag{6.30}$$

6.3.4.5 ACK Rate

The reception rate of ACKs represents the rate at which ACK packets reach the source. The value of this rate determines the actual evolution of the Congestion Window and by consequence the TCP/IP source rate. Let $\lambda_j(t)$ be the reception rate of ACKs and $\lambda_i(t)$

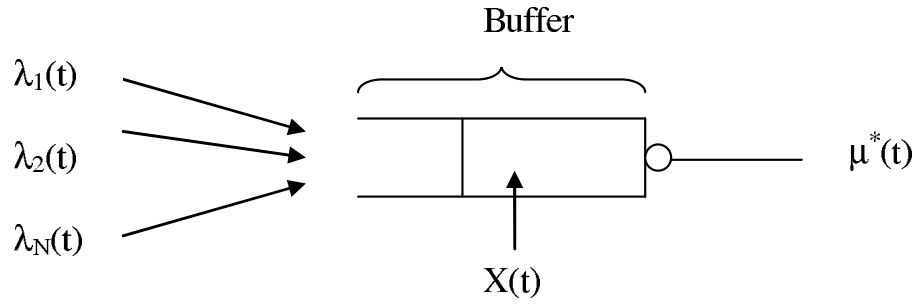


Figure 6.5: Node Model

the TCP/IP source rate, we have the following relation:

$$\lambda_j(t) = \begin{cases} \lambda_i(t) & FR \\ \frac{\lambda_i(t)}{b} & Otherwise \end{cases} \tag{6.31}$$

In the multi-source configuration we replace $\lambda_j(t)$ by $\lambda_{s,j}(t)$ with s is the source index:

$$\lambda_{s,j}(t) = \begin{cases} \lambda_{s,i}(t) & FR \\ \frac{\lambda_{s,i}(t)}{b} & Otherwise \end{cases} \tag{6.32}$$

6.3.5 Network Modelling

TCP/IP operation is controlled by network conditions. The evolution of the Congestion Window and TCP/IP source rate depends on RTT estimation and loss rate at each node. We present in the following sections the estimation of different parameters necessary for the operation of TCP/IP at each node.

6.3.5.1 Node Modelling

Nodes are modelled as D(t)/D/1/N queues. $D(t)$ arrivals are considered because we model the exact evolution of TCP/IP not its stochastic behaviour. Figure **6.5** depicts the queue and the different parameters associated with it:

- $\lambda(t) = \sum_i \lambda_i(t)$ is the total rate entering the node at date t.
- $X(t)$ is the queue load at date t.
- $\mu^*(t)$ the output rate of the queue at date t.
- dt the integration step.

The following fundamental relation must always be verified (at any time t).

$$X(t + dt) = (\lambda(t) - \mu^*(t)).dt + X(t) \tag{6.33}$$

If $X(t+dt)$ is bigger than buffer size, we start the loss processing mode. An overflow has occurred while the source doesn't know about it yet. We calculate first the time needed by the source to realize that a loss has occurred (It is the time necessary to receive triple ACK as prescribed by the TCP/IP algorithm). We call this period Δt_3 referring to the reception of three ACKs. The lost packet is assigned to the source and a loss credit value is initialized to count the number of losses. As the source continues transmitting, more packets will arrive to the node. More losses may actually happen also, before the source knows about the first loss. When the source gets informed about the first loss it passes into Fast Retransmit mode and retransmits all lost packets.

The estimation of Δt_3 time plays an important role in the model, as it determines the number of losses that may be happened. In the case of one source this time is given by:

$$\Delta t_3 = RTT(t) + \frac{3}{\mu_{\min}} \tag{6.34}$$

In multi-source configuration the estimation of this time depends on the $\mu_{\min}$ value for each source. The estimation of $\mu_{\min}$ when the node has multiple entries is detailed hereafter. Consider:

- $\lambda_s(t)$ is the rate of flow s entering the node at date t.
- $X_s(t)$ is the queue load of flow s at date t.
- $\mu_s^*(t)$ the output rate of flow s at date t.

The conservation laws imply:

$$\sum_{s\in S} \mu_s^* = \mu^*, \quad \sum_{s\in S} \lambda_s = \lambda, \quad \sum_{s\in S} X_s = X \tag{6.35}$$

And

$$\forall s \in S, X_s(t+dt) = \left[\lambda_s(t) - \mu_s^*(t)\right].dt + X_s(t) \tag{6.36}$$

S represents the group of sources. Globally, the following relation must be verified:

$$\mu^*(t) \leq \mu \tag{6.37}$$

We define the following coefficient $C_s(t)$ taking into account the residual load of the queue as well as the continuous increment:

$$\begin{array}{l} C_s(t) = X_s(t) + \lambda_s(t).dt \\ \sum\limits_s C_s(t) = C(t) = X(t) + \lambda(t)dt \end{array} \tag{6.38}$$

The number of packets departing during dt can not exceed the total number of packets to be sent:

$$C(t) - \mu^*(t)dt \geq 0 \Leftrightarrow \mu^*(t) \leq \frac{C(t)}{dt} \tag{6.39}$$

The output rate per flow is calculated by:

$$\mu_s^*(t) = \frac{C_s(t)}{C(t)} \times \min\left\{\frac{C(t)}{dt}, \mu\right\} \tag{6.40}$$

The output rate $\mu_s^*(t)$ is used to estimate $\mu_{s,\min}$ which is the value of $\mu_{\min}$ as seen by source s.

$$\mu_{s,\min} = \min\{\mu_s^* \quad \text{on} \quad \text{all} \quad \text{nodes}\} \tag{6.41}$$

Thus, in multi-source configuration, equation (6.34) takes the form:

$$\Delta t_{s,3} = RTT_s(t) + \frac{3}{\mu_{s,\min}} \tag{6.42}$$

The value of $\Delta t_{s,3}$ determines the number of lost packets per source when a buffer overflow happens. Its value depends on the value of the source rate by means of the $C_s(t)$ coefficient. As a consequence the number of lost packets depends on the source rate. Indeed, the loss credit is initialized for each source and the number of lost packets is updated for each source in function of its delay $\Delta t_{s,3}$.

6.3.5.2 RTT Estimation

Estimating the value of RTT is a fundamental step as all rates depend on it. Indeed, the RTT value depends on the path followed by packets. Consider D_i is the delay of the link between router i and router $i+1$, and T_i is the processing delay of one packet in node i. The value of RTT can be obtained as follows:

$$RTT(t) = \sum_{i \in R} T_i(t) + D_i \tag{6.43}$$

With R denoting the group of Routers on the packets path and $T_i(t)$ is given by:

$$T_i(t) = \begin{cases} \frac{1}{\mu_{\min}} & n(t) = 0 \\ \frac{n(t)}{\mu_{\min}} & n(t) > 0 \end{cases} \tag{6.44}$$

$n(t)$ represents the number of packets in queue at date t.

In multi-source configuration the equations become:

$$RTT_s(t) = \sum_{i \in R} T_{s,i}(t) + D_i \tag{6.45}$$

$$T_{s,i}(t) = \begin{cases} \frac{1}{\mu_{s,\min}} & n_s(t) = 0 \\ \frac{n_s(t)}{\mu_{s,\min}} & n_s(t) > 0 \end{cases} \tag{6.46}$$

6.3.6 Recapitulative

In Table **6.1** we give a recapitulative of the different equations used for mono-source and multi-source differential model. This table serves as a reference for TCP/IP differential model implementation.

The differential analytical model presented in previous sections simulates the behaviour of TCP/IP in mono-source and multi-source configurations (also in a network). Network nodes hold the state of each source represented by: input rate, output rate and load share in the buffer. A sharing mechanism of the output rate of each node is achieved using a proportion coefficient. The proportion coefficient takes into account both the residual source load in the buffer as well as the input rate. A new value of the minimum output rate is calculated per source. Reference dates are evaluated as a function of this minimum output rate per source. RTT estimation is also done per source.

6.4 Validation Tests

Validation tests concern the evaluation of the differential analytical model of TCP/IP in mono-source and multi-source configurations. All results are compared with event-driven simulations to evaluate the accuracy of the differential model. Event driven simulations are undertaken in DHS using the event-driven mode.

In mono-source tests we validate the transient behaviour of one TCP/IP connection, with different operation modes. On the other hand, in multi-source tests we validate the model in terms of average loss rate and average throughput achieved. Finally, a global validation over many tests, expressed in terms of average relative error (ARE), average difference (AD) and standard deviation (σ), is provided for both mono-source and multi-source configurations:

$$ARE = \frac{\sum_{i=1}^{N} \frac{\left|\hat{x_i} - x_i\right|}{x_i}}{N} \tag{6.47}$$

$$AD = \frac{\sum (x - \bar{x})}{N} \tag{6.48}$$

$$\sigma = \sqrt{\frac{N \sum x^2 - \left(\sum x\right)^2}{N^2}} \tag{6.49}$$

6.4.1 Mono-source Validation

We use a simple two nodes network (Source and Destination) with fixed delay link. The goal is to validate the behaviour of mono-source differential model of TCP/IP versus event-driven simulation of TCP/IP. The validation network is modelled by four queues as shown on the Figure **6.6**. This simple network is very representative and allows testing the different parameters of the model easily. The same network will be used also

Table 6.1: Recapitulative of TCP/IP Model (mono-source and multi-source configurations)

Mode	TCP Mono-Source	TCP Multi-Source
Slow Start (rate)	$\lambda(t) = \frac{b}{RTT(t)}.e^{\frac{\ln\left(\frac{b+1}{b}\right)}{RTT(t)}.t}$ $\overset{\bullet}{\lambda}(t) = \frac{\lambda(t)}{RTT(t)}.\ln\left(\frac{b+1}{b}\right)$	$\lambda_s(t) = \frac{b}{RTT_s(t)}.e^{\frac{\ln\left(\frac{b+1}{b}\right)}{RTT_s(t)}.t}$ $\overset{\bullet}{\lambda}_s(t) = \frac{\lambda_s(t)}{RTT_s(t)}.\ln\left(\frac{b+1}{b}\right)$
ACK Limitation	$\lambda_{r-\max} = \mu_{\min}\frac{b+1}{b}$	$\lambda_{s,r-\max} = (b+1).\lambda_{s,ack}$
CWND$_{max}$ Limitation	$\lambda_{cwnd-\max} = \min\left(\frac{CWND_{\max}}{RTT(t)}, \mu_{\min}\right)$	$\lambda_{s,cwnd-\max} = \min\left(\frac{CWND_{\max}}{RTT_s(t)}, b.\lambda_{s,ack}\right)$
Congestion Avoidance (rate)	$\lambda(t) = \frac{CWND(t)}{RTT(t)}$	$\lambda(t) = \frac{CWND_s(t)}{RTT_s(t)}$
ACK Limitation	$\lambda_{r-\max}(t) = \left(1 + \frac{1}{b.CWND(t)}\right).\mu_{\min}$	$\lambda_{s,r-\max}(t) = \left(b + \frac{1}{CWND_s(t)}\right).\lambda_{s,ack}$
CWND$_{max}$ Limitation	$\lambda_{cwnd-\max} = \frac{CWND_{\max}}{RTT(t)}$	$\lambda_{s,cwnd-\max} = \frac{CWND_{\max}}{RTT_s(t)}$
Fast Retransmit (rate)	$\lambda(t) = \frac{1}{RTT(t)}$	$\lambda_s(t) = \frac{1}{RTT_s(t)}$
$CWND(t)$ Evolution	$CWND(t) = \begin{cases} CWND(t)+1 & SS \\ CWND(t)+\frac{1}{CWND(t)} & CA \\ 0 & FR \\ \frac{CWND(t)}{2} & Loss \end{cases}$	$CWND_s(t) = \begin{cases} CWND_s(t)+1 & SS \\ CWND_s(t)+\frac{1}{CWND_s(t)} & CA \\ 0 & FR \\ \frac{CWND_s(t)}{2} & Loss \end{cases}$
ACK Rate	$\lambda_j(t) = \begin{cases} \lambda_i(t) & FR \\ \frac{\lambda_i(t)}{b} & Otherwise \end{cases}$	$\lambda_{s,j}(t) = \begin{cases} \lambda_{s,i}(t) & FR \\ \frac{\lambda_{s,i}(t)}{b} & Otherwise \end{cases}$
RTT Estimation	$RTT(t) = \sum_{i\in R} T_i(t) + D_i$ With $T_i(t) = \begin{cases} \frac{1}{\mu_{\min}} & N(t)=0 \\ \frac{N(t)}{\mu_{\min}} & N(t)>0 \end{cases}$	$RTT_s(t) = \sum_{i\in R} T_{s,i}(t) + D_i$ With $T_{s,i}(t) = \begin{cases} \frac{1}{\mu_{s,\min}} & N_s(t)=0 \\ \frac{N_s(t)}{\mu_{s,\min}} & N_s(t)>0 \end{cases}$
Loss Detection	$\Delta t_3 = RTT(t) + \frac{3}{\mu_{\min}}$	$\Delta t_{s,3} = RTT_s(t) + \frac{3}{\mu_{s,\min}}$
$\mu_{\min}$	$\mu_{\min}$ defined by the network	$\mu_{s,\min} = \min\{\mu_s^*\}$ $\mu_s^*(t) = \frac{C_s(t)}{C(t)} \times \min\{C(t)/dt, \mu\}$ With $\sum_{s\in S}\mu_s^* = \mu^*, \quad \sum_{s\in S}\lambda_s = \lambda \sum_{s\in S} X_s = X$ $X(t+dt) = (\lambda(t) - \mu^*(t)).dt + X(t)$ $C_s(t) = X_s(t) + \lambda_s(t).dt$ $\sum_{s\in S} C_s(t) = C(t) = X(t) + \lambda(t)dt$

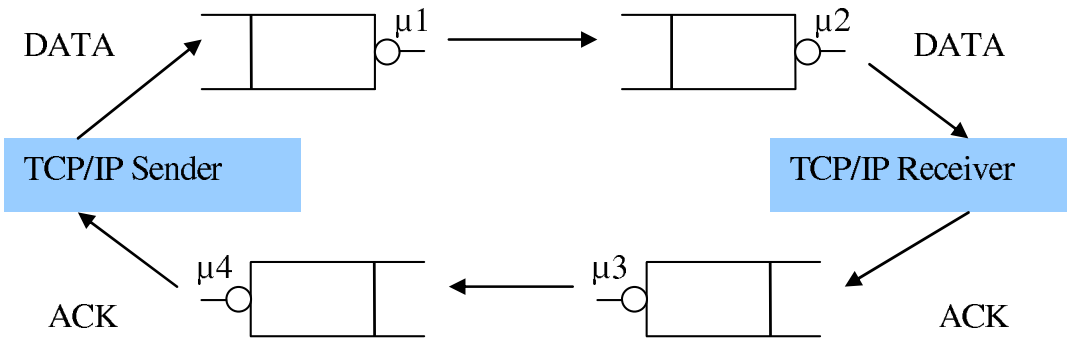


Figure 6.6: Simulated Queues in Validation Network

for multi-source validation tests, where multiple conections are initiated from the TCP sender node. We list in the following the simulation parameters:

- Service rate μ in Bps.
- File size Q in Bytes.
- Link delay L in ms.
- Buffer capacity B in Packets.
- Number of TCP/IP sources N (used for multi-source configuration tests).

6.4.1.1 Operation Modes

The first result set concerns the validation of different operation modes. Based on the parameters listed in Table **6.2** we verify the transition between different operation modes of TCP/IP compared to event-driven simulation.

Table 6.2: Simulation Parameters (Operation Modes)

μ **(Bps)**	B **(Packets)**	L **(ms)**
500000	35	1

On Figure **6.7** we show the evolution of the congestion window via differential and event-driven simulations. The curves show the good estimation of source rate by the differential model (the source rate is controlled by the CWND size).

Loss detection is very important as it impacts the transitions between operation modes and the overall throughput achieved by the source. On Figure **6.8** we show the evolution of lost packets number, and on Figure **6.9** we show the cumulated lost packets number in both differential and event-driven simulations.

There is a little deviation between the differential model and the event-driven simulation. In fact, we count one less lost packet than event-driven simulation. This is due to

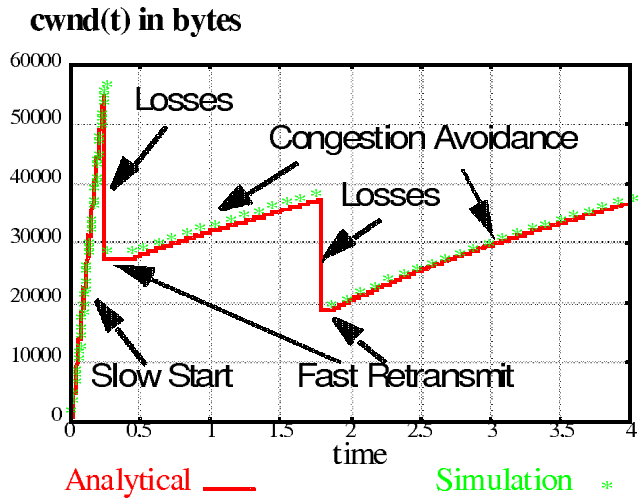


Figure 6.7: Evolution of $CWND(t)$ (Differential vs Event-driven)

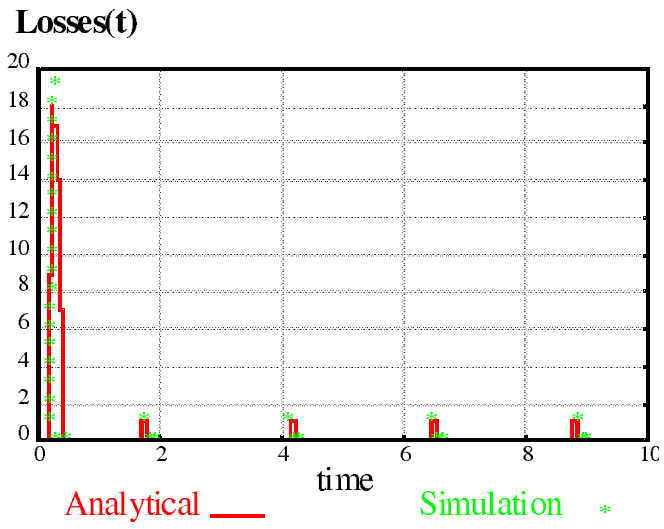


Figure 6.8: Losses Detection (Differential vs Event-driven)

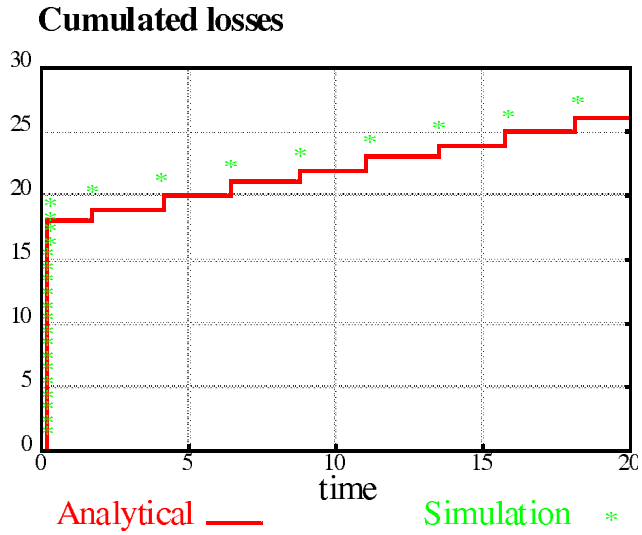


Figure 6.9: Cumulated Losses (Differential vs Event-driven)

two factors: first, we assume the evolution of rate follows a geometric series in Slow Start mode, while this is only an approximation. Second, TCP/IP sends packets by bursts of $b+1$ packets while we are propagating fluid rates.

The curve on Figure **6.10** depicts the evolution of the RTT value. The RTT value is well estimated by the differential model. Recall that the good estimation of RTT guarantees a good estimation of the packet rate.

As the instantaneous rate is difficult to obtain in event-driven simulations we compare the number of sent packets in both cases (Figure **6.11**). We observe a little difference between the differential model and the event-driven simulation. The Fast Retransmit phase lasts less than expected. In fact, as we detect one lost packet less, the retransmission period of lost packets lasts less than in event-driven simulation. This difference is of one RTT the time of one lost packet retransmission.

Globally, we obtain very good results. Packets rate and losses number are evaluated precisely. In Table **6.3** we give the global statistics of our simulations.

Table 6.3: Global Statistics (One Source)

Link Delay 1 ms	**Analytical**	**Simulation**	**Relative Error**
Rate (Packets/s)	488.7	488.14	0.11%
Loss ratio	0.266%	0.276%	3.831%

6.4.1.2 Global Validation

The results showed in previous section point out clearly the precision of the differential model. We present now summarized results on a large number of simulations. Using the same network we vary the value of service rate μ and buffer capacity B. For each couple $\{\mu, B\}$ we vary the size of transferred file.

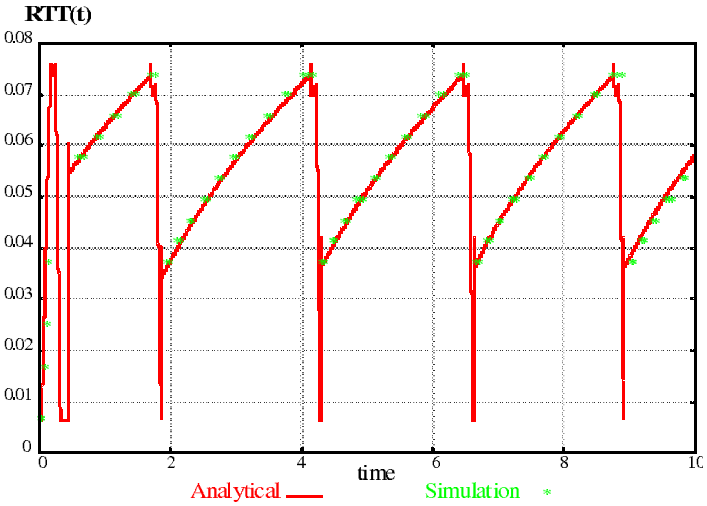


Figure 6.10: RTT Evolution

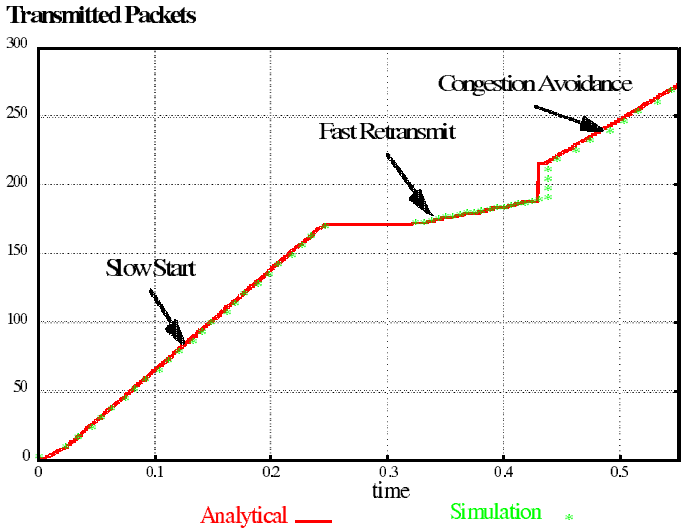


Figure 6.11: Number of Transmitted Packets by The Source

Table 6.4: Global Configuration Parameters

Parameter	Value
μ	{1, 2, 5, 8 , 10, 20, 50, 70, 100}*16384 Bps
B	{10, 20, 30, 40, 50} Packet
Q	{1, 2, 3, 5, 8, 10, 20, 50, 100, 200, 500}*102336 Bytes

We perform simulations using the configuration parameters listed in Table **6.4**. For each configuration couple $\{\mu, B\}$ we evaluate the transmission duration, the source packet rate, the input rate and the output rate (ACK rate) of the receiver as well as the loss rate of the source. For each file size we calculate the relative error for each of the mentioned values comparing with the event-driven simulation. We calculate the average of these relative errors by couple $\{\mu, B\}$. Finally we show the global average of these simulations, resulting by averaging all obtained averages by couple $\{\mu, B\}$. We give also the standard deviation (see Table **6.5**).

Table 6.5: Relative Error (%) with Link Delay 1 ms

Link Delay 1 ms	**Transmission Duration**	**Loss Rate**	**Throughput**
ARE (%)	1.68	7.48	1.63
AD(%)	0.72	3.81	0.59
σ(%)	0.89	4.87	0.76

The global validation shows that the differential model works very well. We have an average relative error less than 5% for the transmission duration and the source rate while only the average relative error of losses is 7.48%.

6.4.2 Multi-source Validation

Using the same network in Figure **6.6**, we inject N sources in the TCP sender node. Configuration parameters are the same as in Table **6.2**. We compare the average loss and average source rate over the N sources in both differential and event-driven simulations. Results for different values of N are listed in Table **6.6**.

Table 6.6: Multi-source Configuration

N	**Loss Rate** %		λ **Kbps (Per source)**	
	Diff	Eve	Diff	Eve
3	0.73	0.86	496.6	494
6	2.17	2.3	499.65	496.5
9	4.23	4.14	503.26	500.6
12	6.7	6.3	507.12	502.9

The average statistics show a good estimation of the differential model to both loss rate and source rate. However, a difference is observed in the loss rate as the number of connections increases. Indeed, the differential model overestimates the number of losses when the number of connections increases. This is mainly due to the loss sharing mechanism.

For a global validation we perform multiple tests using configuration parameters in Table **6.2** with variable number of connections. In Table **6.7** we give the results in terms of relative error.

Table 6.7: Multi-source Global Validation

Relative Error%	Transfer Duration	Loss Rate	Throughput
ARE	1.06	4.5	0.9
AD	0.84	2.7	0.54
σ	1.11	3.9	0.99

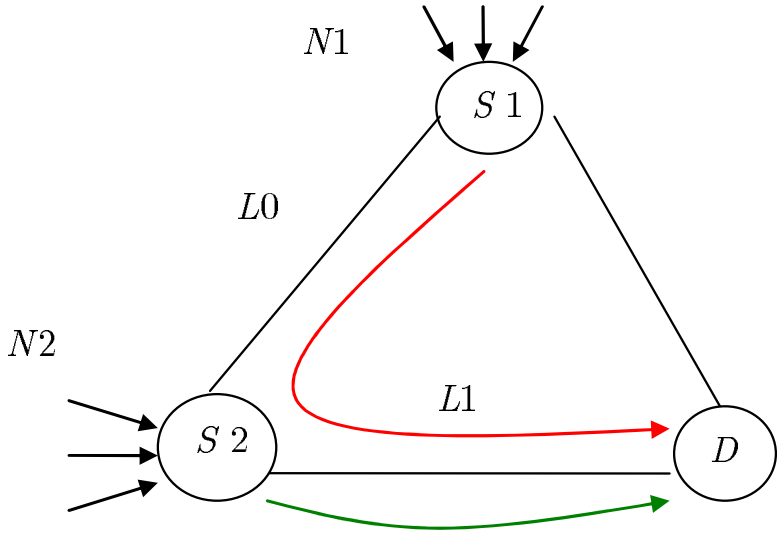


Figure 6.12: Triangle Network

Results show good global behaviour of multi-source model. Relative error is less than 5% for transfer duration, throughput as well as loss rate.

6.4.3 Network validation

So far only two nodes where considered with the same route for all TCP connections. Now, we suggest a simple triangle network with two groups of sources sharing a bottleneck (see Figure **6.12**).

We study two groups of TCP sources $N1$ and $N2$ sharing a bottleneck link $L1$. Sources of $N1$ group have the node $S1$ as originating source node and node D as destination target node transiting via $S2$ node. On the other hand, $N2$ group of sources has the node $S2$ as originating source node and node D as destination target node. Each node is composed of two queue interfaces (one for each direction in, out).

Besides previous configuration parameters, a new parameter concerning the number of sources in each group is taken into consideration. Furthermore, source of $N1$ group are subject to longer route and longer delay $(L0 + L1)$.

We perform simulations using the configuration parameters listed in Table **6.8**, bottleneck delays are $(L0 = 10ms)$ and $(L1 = 10ms)$, $\mu = 500000\ Bps$ and $B = 35\ packets$.

The results show a good overall behaviour for the two groups of sources. Relative error is less than 5% for transfer duration, throughput while loss rate exhibits bigger

Table 6.8: Configuration Parameters (Triangle Network)

Parameter	Values
$N1$	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
$N2$	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
Q	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} * 1000000\ Bytes$

Table 6.9: Triangle Network Global Validation

Relative Error%	Transfer Duration	Loss Rate	Throughput
ARE	2.87	9.1	1.2
AD	1.2	5.3	0.75
σ	1.8	6.1	1.04

values (= 9.1%).

6.4.4 Qualitative Evaluation

The simulation of a TCP/IP packet in event-driven techniques requires at least two events: one for the packet generation and the other for the timeout. Let $\bar{\lambda}$ be the average throughput of simulated flows, and then the average number of packets N generated during the simulation (of real duration T) is given by:

$$N = \bar{\lambda} * T \tag{6.50}$$

The number of events needed by the simulation N_{eve} is approximately:

$$N_{eve} \approx N * 2 \tag{6.51}$$

As the simulation time is proportional to the number of events, we will have longer simulation times with greater average throughputs. On the other hand, the differential technique is based on numerical integration of differential equations by fixed steps. The average rate propagated during the integration step Δt may correspond to one or more packets according to the average throughput. However, the simulation is achieved in M integration steps:

$$M = \frac{T}{\Delta t} \tag{6.52}$$

Notice that M determines the number of iterations needed to achieve the simulation. Practically, for the same time step Δt we have more packets simulated when the average throughput is greater. This will result in an increasing gain in the simulation time when propagated throughputs are bigger compared with event-driven technique, for which more packets and consequently more events are needed.

6.5 Conclusion

In this Chapter, we presented a differential analytical model for TCP/IP. The model is based on differential equations describing TCP/IP rate in different operation modes. Network nodes are modelled as $D(t)/D/1/N$ queues. The model takes into consideration different features including: congestion window, the reception rate of ACKs, loss detection delay, etc...

TCP/IP is a closed-loop protocol as it adapts its transmission rate to network congestion conditions. It makes it very important to model its reactive mechanisms in order to reproduce its transient behaviour. This is achieved by mixing differential equations and control events. In this way, differential equations describe the fluid behaviour and generate control events allowing a dynamic hand over between differential equations. Using differential equations we can evaluate instantaneous rate, delay and packet losses. When a packet loss occurs, the loss date is calculated and a control event is generated. TCP/IP can change then its operation mode according to a well defined state machine.

Different tests were achieved using the differential model in mono-source and multi-source configurations. We obtained good results on different configurations and for numerous tests. The model is very interesting as it allows simulating the transient behaviour of TCP/IP without simulating packets. Only fluid rate is propagated between nodes. When big flows are simulated using event-driven techniques the number of events explodes as it is proportional to the number of packets. In this case the differential technique becomes more efficient as it propagates fluid rates not packets. This property makes the differential model very suitable to large scale network simulations.

It is clear that the performance of the differential model is tightly linked to the choice of the integration step. Smaller values of this step result in more accurate simulations while big values allow faster simulations. Naturally the integration step should not be fixed. An important enhancement of the model resides in a dynamic choice of the integration step according to the operation mode and buffer status. Thus, bigger values can be used for Congestion Avoidance mode when TCP/IP increases its rate slowly and avoids congestion! While smaller values should be used in slow start mode because of the exponential increase in rate. Another factor related to buffer congestion status can also be used. When buffers are partially filled (well dimensioned networks with sufficient bandwidth) bigger integration steps can be used as buffer overflows are not likely to happen. All these refinements can enhance considerably the performance of the model.

The differential modelling methodology proposed in this chapter illustrates very well the use of hybrid mechanisms involving differential equations and control events to model complex and reactive processes precisely and efficiently.

Chapter 7

Session Based QoS with SIP

7.1 Introduction

The convergence to all IP network came with new signaling protocols to handle user sessions. Thus, the Session Initiation Protocol (SIP) was proposed as a signaling protocol to establish and release sessions between end users. SIP is very general and can be used for any kind of sessions in all communication networks. Moreover in the year 2000, SIP was selected by the Third Generation Partnership Project (3GPP) as the call control protocol for the 3G IP-based mobile networks. On the other hand, the successful deployment of MPLS in DiffServ networks delegates label switching as an apt switching technology for the future core networks (especially for its Traffic Engineering (TE) capabilities). Indeed, combining both technologies introduces a new vision of QoS management at the application level. The autonomous structure of SIP makes it possible to manage user sessions as phone calls, from the beginning to the end. However, SIP can achieve much more than signaling the beginning and the end of a communication. In particular, it can host traffic engineering intelligence. Thus, the use of SIP over DiffServ networks allows flexible QoS management as it combines DiffServ facilities with SIP supervision.

In this chapter, we propose a QoS management framework based on SIP over DiffServ environment. Where QoS management mechanisms are implemented and supervised by the SIP proxy server. The proposed mechanisms are: session scheduling based on session duration and/or session data volume, and bandwidth allocation on a per-flow basis using equivalent bandwidth estimation techniques. In section **7.2** we give a brief overview of SIP and its associated Session Description Protocol (SDP), and then in section **7.3** we explain the suggested SIP over DiffServ architecture. In section **7.4**, we present equivalent bandwidth estimation techniques used for bandwidth allocation. Finally, in section **7.5** we present the session based QoS algorithms implemented along with simulation results.

7.2 SIP

Session Initiation Protocol is an application layer control protocol designed and developed by the IETF. The easy implementation, flexibility and good scalability are the main motivations considered while designing this protocol. The specification is available in form of several RFCs, the most important one is RFC3261 [RSC+02] which contains the core protocol specification. The main task of the protocol is to set up and release sessions between end users. The session refers to the activity between sender and receiver when the whole state is maintained during the communication. Communication sessions include Internet telephone calls, multimedia conference sessions, Web sessions, distributed computer game sessions, etc.

The communication itself between devices is achieved by other protocols (often RTP, RTCP and SDP) as the purpose of SIP is to initiate communications only. RTP carries the real-time application data by splitting and encoding data into packets to allow per-packet transport on the Internet. On the other hand, SDP describes and encodes capabilities of sessions. This includes the type of codecs used to encode media in order to facilitate decoding process, maximum allowed bit rates, the transport protocol, etc.

The end-to-end model of SIP complies with the Internet architecture. Indeed, all the intelligence is stored in end devices, including state. This protects from single point failure while preserving scalability in networks. In contrast with Public Switched Telephone Network (PSTN) where state and intelligence are stored in the network while terminals are dump. However, SIP can provide the same functionality as PSTN with the possibility to implement end-to-end services that are hardly configured in PSTN.

Finally, it is clear that the scalability and decentralization of SIP come at the cost of end-to-end message overhead. In fact, SIP is based on HTTP protocol which is widely used on the Web. Actually HTTP can be seen as a signaling protocol also, as web browsers tell HTTP servers about the documents they need. The encoding of message headers in both protocols (HTTP and SIP) have been inherited from RFC822 [Cro82]. This encoding has already showed robustness and flexibility with HTTP.

7.2.1 Architecture

End users in SIP architecture are called User Agents (UAs). A user agent may have two roles:

- User agent client (UAC): A client application that initiates the SIP request.
- User agent server (UAS): A server application that contacts the user when a SIP request is received and that returns a response on behalf of the user.

Generally, a SIP end device can act as both a UAC and a UAS per transaction. The nomination of two roles UAC or a UAS depends on the request initiation direction. Thus, when a user agent sends an INVITE request and receives responses it behaves like UAC. On the other hand, the user agent receiving the INVITE and sending responses

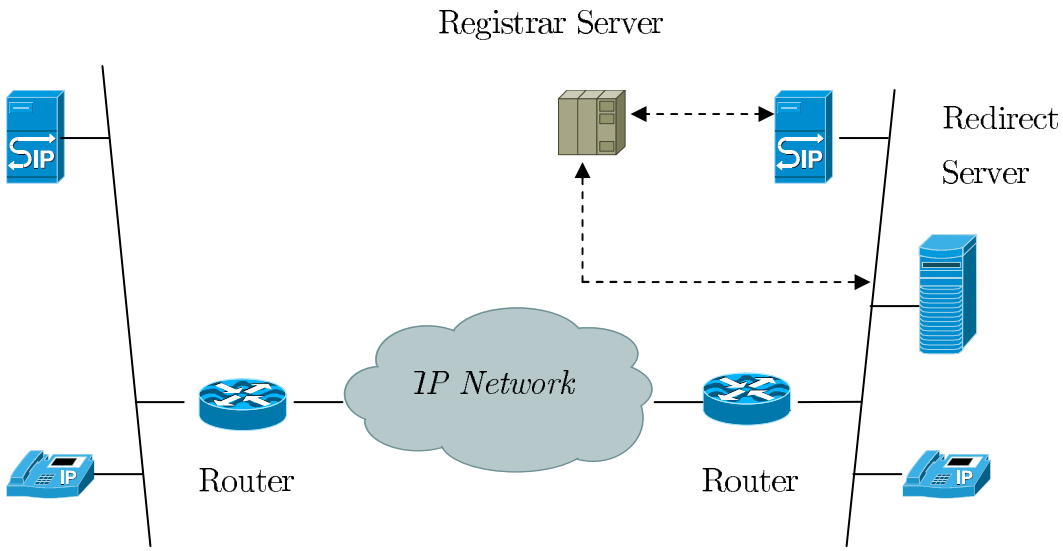


Figure 7.1: SIP Architecture

is considered as UAS. The physical elements of a SIP network fall into two categories: clients and servers. Figure **7.1** illustrates the architecture of a SIP network.

Client in SIP is a general concept. It could be any device initiating sessions (Phones, PCs, Palms, ...). The main SIP servers are:

- Proxy server: The most important element in the SIP architecture, as it constitutes an intermediate device receiving SIP requests from clients and forwarding the requests on clients' behalf. Typically, proxy servers forward SIP messages to other SIP servers in the network. Besides it provides functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.
- Redirect server: Takes care of directing the client to the next hop until the client reaches the destination server and contacts UAS directly.
- Registrar server: Handles UAC registration request for its current location. Generally, Registrar server is co-located in the same physical entity hosting the redirect or proxy server.

7.2.2 SDP

SIP works in tandem with the Session Description Protocol (SDP) that describes multimedia sessions. Session description serves for session announcement, session invitation and other session initiation functionalities. SDP is completely independent of transport protocol. It concerns mainly the format of session description and is designed to work with any transport protocol.

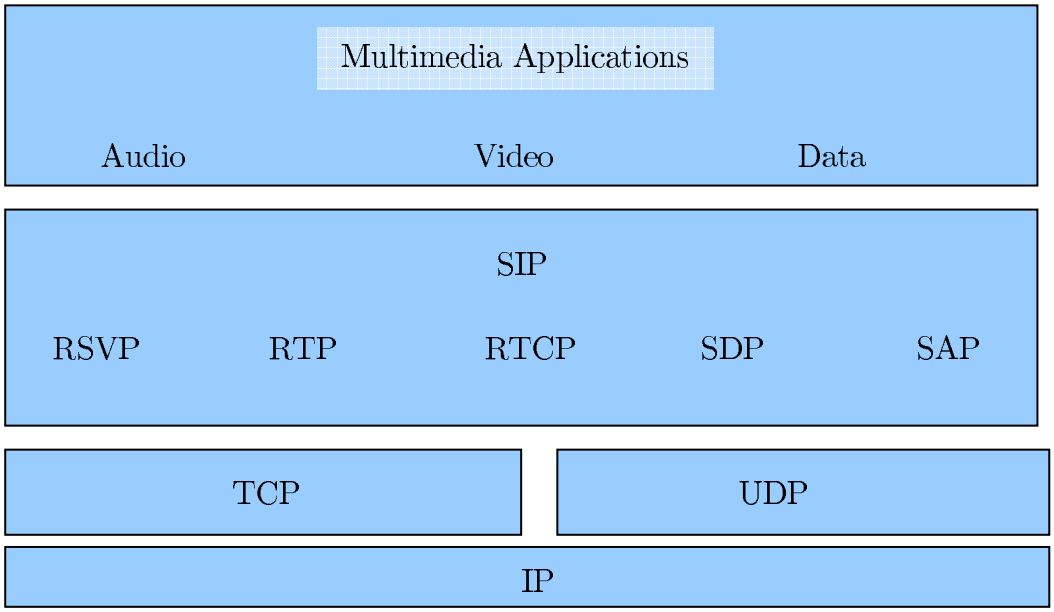


Figure 7.2: Multimedia Protocol Stack with SIP

Many of the SDP messages are sent using Session Announcement Protocol (SAP). These messages are UDP packets with a SAP header and a text payload. The text payload is the SDP session description. Figure **7.2** depicts the position of SIP and SDP in the multimedia protocol stack.

7.3 SIP over DiffServ

Few researches [ZG03, SV02, RLB+05] have addressed the architecture of SIP over DiffServ architecture in IP networks. Zhang and Guy [ZG03] proposed an extension to the Proxy server in the SIP architecture to include Traffic Engineering (TE) capabilities; they call it TE-SIP server. The TE-SIP server uses the messages exchanged during an SIP session to provide TE requests.

In order to use SIP over DiffServ domain, the SIP architecture needs extensions at the Proxy server to include traffic engineering capabilities. The proxy server may use the messages exchanged during an SIP session to provide traffic engineering requests. These requests will be exchanged between the SIP proxy server and the Label Edge Router (LER) by Common Open Policy Service (COPS) protocol messages [SV02]. Indeed, we need to transfer information related to the request of resource by QoS clients and for the allocation of resources by resource allocation servers (e.g., bandwidth broker) in a DiffServ network. Hence, this resource allocation functionality can be added in the COPS framework. Figure **7.3** depicts the proposed architecture for SIP over DiffServ.

TE-SIP server could only negotiate TE sessions with another TE-SIP server otherwise normal SIP session (Without TE extensions) is initiated. The flow of SIP messages is summarized on Figure **7.4**.

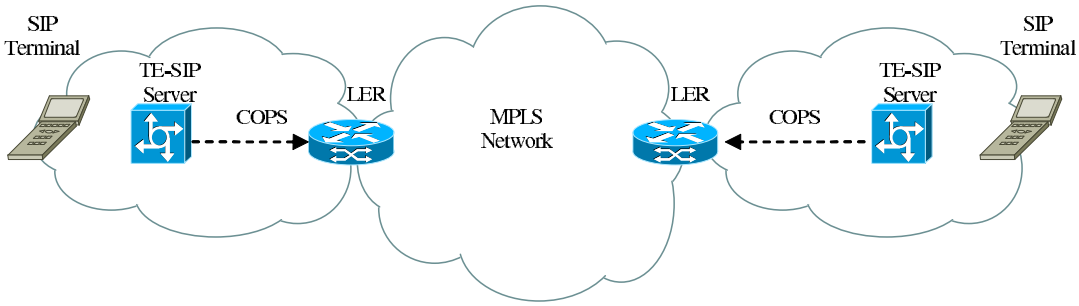


Figure 7.3: SIP over DiffServ Architecture

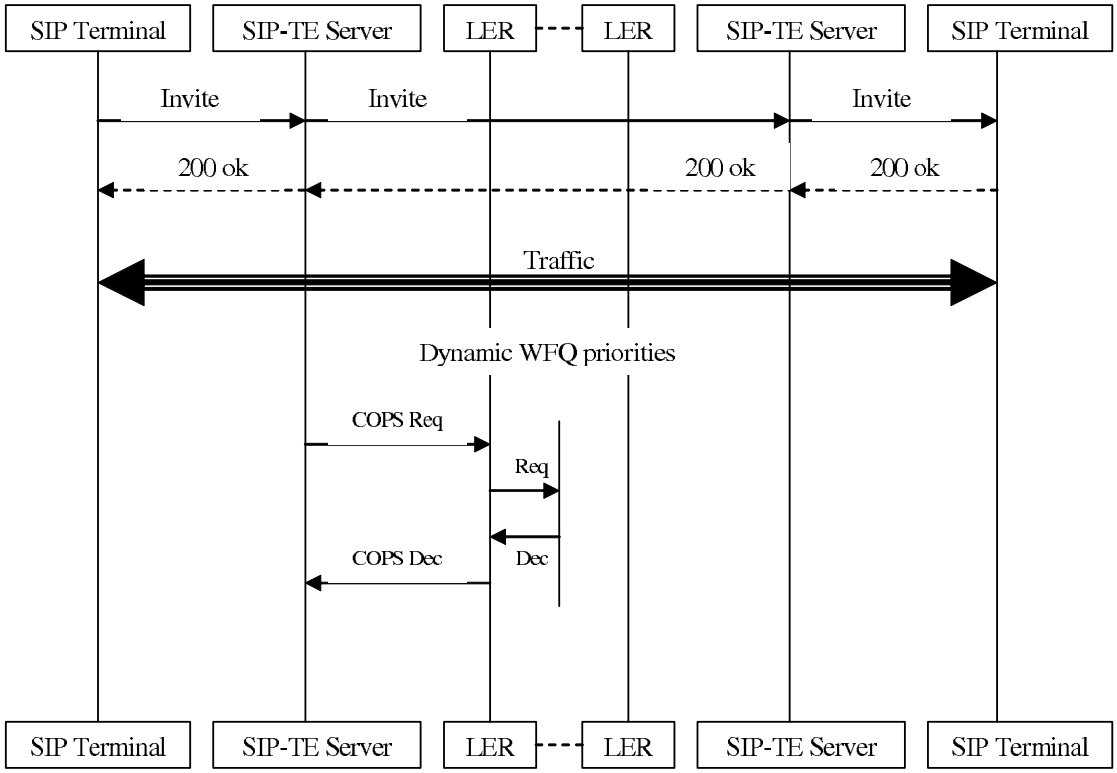


Figure 7.4: SIP over DiffServ Flow of Messages

Assuming that communication issues related to resources management and reservation at the LER are achieved by the COPS framework; SIP can play the role of an application control layer protocol for delivering QoS to user sessions. Generally, we speak of QoS per type of service. Thus, for real time applications we are concerned about end to end time constraints like delay and jitter to guarantee the reconstitution of multimedia signals (voice or video). On the other hand, non-real time applications (or data applications) are less sensitive to time constraints. In both cases real time and non-real time applications, the QoS requirements are grouped together to express an SLA required for one application. A global view per session is important in our context because SIP handles the session establishment and considers the user activity during a session as a whole.

Based on the above presented architecture we want to implement QoS management algorithms at the session level. The proposed algorithms are based on traffic engineering techniques and will be hosted in the SIP proxy server. Thus, the TE-SIP server implements QoS mechanisms on multimedia sessions, based on measurements and a priori estimation of equivalent bandwidth. Indeed, the TE-SIP server measures the duration of sessions and the data volume exchanged during sessions (functions that are generally used for billing purposes), then a dynamic assignment of class of service can be achieved. Besides, the TE-SIP server can also perform flow based equivalent bandwidth estimation, based on information collected about session parameters before initiating it. However, this requires equivalent bandwidth estimation techniques that we will present in the following section.

7.4 Equivalent Bandwidth

Many traffic control strategies rely on the notion of the equivalent (or effective) bandwidth or the resulting connection load on network links. The notion of equivalent bandwidth has been used in the literature and two broad categories of equivalent bandwidth estimation approaches are generally used. The first category is based on Kelly's [Kel96] mathematical definition of equivalent bandwidth for different kinds of traffic. The second category refers to analytical methods based on traditional queuing theory (e.g. [GAN91, Whi93]). The mathematical framework proposed by Kelly relies on large deviation theory to estimate the equivalent bandwidth of a stationary arrival process. On the other hand, analytical approaches hypothesize the traffic models in order to give an approximate expression for the equivalent bandwidths in some cases such as Markov processes.

7.4.1 Equivalent Bandwidth by Kelly's Approach

Frank Kelly introduced in [Kel91] a scalar involving the statistical properties of a single source, the statistical properties of traffic being superposed with the traffic source and the capacity and buffer of a multiplexer. This scalar expresses the equivalent bandwidth which estimates the required resources for the source in order to respect its Quality of

Service (QoS) requirements. For example, consider n_j sources of type j, each having equivalent bandwidth α_j, and J the number of source types. The linear constraint to meet the QoS of all J type sources is defined as follows:

$$\sum_{j=1}^{J} n_j \cdot \alpha_j \leq C^* \tag{7.1}$$

Where C^* is equivalent capacity of the link which depends on different parameters (Link capacity, buffer, QoS, and the statistical properties of the traffic mix). The equivalent bandwidth estimation is based on the asymptotic analysis. It concerns mainly the way the buffer overflow probability decays as a function of some quantity. Two quantities are generally used: The size of the buffer which gives the large buffer asymptotic [EM93, CW95], and the size of the system (link, number of sources, ...) which gives the many sources asymptotic [SG95, CW96]. The many sources asymptotic consider that the buffer per source and the capacity per source are constant. However, the definitions of equivalent bandwidth based on the large buffer asymptotic have been found not accurate. The main explanation of this refers to the gain we have when superposing many sources together and which does not figure in the large buffer asymptotic. Indeed, Kelly [Kel96] tried to include this information into the definition of the equivalent bandwidth through two parameters: space and time. Briefly, the equivalent bandwidth according to Kelly depends on the link's operating point through these two parameters which can be calculated using the many sources asymptotic.

7.4.1.1 Many Sources Asymptotic

We consider J independent source types being superposed in a multiplexer. The number of sources of each type j is defined by $n_j = N\rho_j$, while N represents the total size of the system and $\rho = (\rho_1, \rho_2, ..., \rho_J)$ is the proportion vector by type of source. To comply with this notation the buffer size is defined as $B = Nb$ and the link capacity $C = Nc$ with parameters b (resp. c) corresponding to the buffer (resp. capacity) per source.

Consider the time interval $[0, t]$, and note $X_j[0, t]$ the load produced by a j source. Kelly introduces [Kel96] the equivalent bandwidth of a source of type j, assuming that $X_j[0, t]$ has stationary increments, as:

$$\alpha_j(s, t) = \frac{1}{s \times t} \log E\left[e^{X_j[0,t]}\right] \tag{7.2}$$

s, t are source context parameters defining the system, i.e., the characteristics of the superposed traffic, capacity, buffer... The time parameter t is related to the duration of the busy period of the buffer prior to overflow, while the space parameter s expresses the degree of superposing. The space parameter depends on the ratio between the peak rate of the superposed sources and the link capacity. Thus, the physical interpretation of the space parameter shows that when link capacity is larger than the peak rate of the superposed sources, the space parameter$s$ tends to zero. As a consequence, the term $\alpha_j(s, t)$ tends to the mean rate of the source. On the other hand, when link capacity is not

much larger than the peak rate of the superposed sources, s is big and consequently the term $\alpha_j(s,t)$ tends to the maximum value of $X_j[0,t]/t$ (Note that $X_j[0,t]$ is a random variable).

Practically, it is important to know the buffer overflow probability in presence of superposed traffics. Based on Kelly's definition of the equivalent bandwidth, Courcoubetis and Weber [CW96] demonstrated the sup inf formula that gives the many sources asymptotic for buffer overflow expressed usually as $P(overflow) = e^{-NI+o(N)} \underset{N\to\infty}{\approx} e^{-NI}$ derived from the general form:

$$\lim_{N\to\infty} \frac{1}{N} \log(P(overflow)) = \sup_t \inf_s \left[s.t \sum_{j=1}^{J} \rho_j \alpha_j(s,t) - s(ct+b) \right] = -I \tag{7.3}$$

Usually $P(overflow)$ is denoted as $Q(Nc, Nb, N\rho)$ expressing the probability that in an infinite buffer where $N\rho = (N\rho_1, N\rho_2, ..., N\rho_J)$ sources are superposed and served at $C = Nc$ rate, the queue length exceeds the threshold $B = Nb$. I is generally called the asymptotic rate function.

The QoS constraint on the overflow probability is expressed as $P(overflow) \leq e^{-\gamma}$. However, the effective bandwidth $\alpha_j(s,t)$ provides a measure of resource usage for a particular operating point of the link, expressed through parameters s and t. For example, if a source of type j_1 has twice as much equivalent bandwidth as a source of type j_2, then for this particular operating point of the link, one source of the first type uses twice as much resources than a source of the second type. The asymptotics underlying the above results assume only stationarity of sources.

7.4.1.2 Large Buffer Asymptotic

The many sources asymptotic definition of the equivalent bandwidth takes into account the effects of statistical superposition of traffic sources. Meanwhile, the definition of the equivalent bandwidth based on the large buffer asymptotic considers only the characteristics of the source as well as the QoS constraint. Thus, if we consider the QoS constraint $P(overflow) \leq e^{-\delta B}$ [CW95, EM93, dVW95], where B is the total buffer. Then, the equivalent bandwidth of a source of type j is given by:

$$\alpha_j(s) = \frac{1}{s} \lim_{t\to\infty} \frac{1}{t} \log E\left[e^{sX_j[0,t]}\right] \tag{7.4}$$

In fact, the last equation is the same as (**7.2**) when $t \to \infty$. Thus, the last equation is only accurate for large buffer sizes as the time parameter t becomes large. That is why for finite buffer sizes, a significant miss utilization of link capacity could occur.

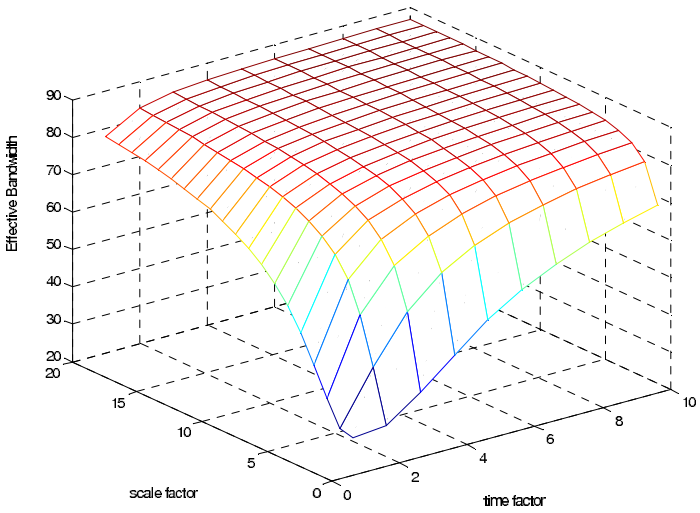


Figure 7.5: The Evolution of ON-OFF VoIP Source Equivalent Bandwidth

7.4.1.3 Equivalent Bandwidth of ON-OFF Sources by Kelly's Formula

The effective bandwidth of an ON-OFF source model (for an individual application) is given by the following equation [Kel96]:

$$\alpha(s,t) = \frac{1}{s.t} \log\left[1 + p.(\exp(s.t.\alpha^*(s,t)) - 1)\right] \tag{7.5}$$

Where $\alpha^*(s,t)$ is the effective bandwidth of the ON state and p is the proportion of time spent in the ON state. The mean and peak of the source are represented by M and h respectively, so $\alpha^*(s,t) = h$ and $p = \frac{M}{h}$. Then the available bandwidth is represented by:

$$\alpha_{M,h}(s,t) = \frac{1}{s \times t} \log\left[1 + \frac{M}{h}.(e^{s.t.h} - 1)\right] \tag{7.6}$$

As an example we show on Figure **7.5** the effective bandwidth for a VoIP traffic source based on G711 codec and classical communication parameters for ON and OFF periods (See **Chapter 5**).

The two dimension graph shows how the equivalent bandwidth varies with time and scale parameters between average and peak rate values. However, determining the time and scale parameter (or the operating point) according to network configuration requires the resolution of the sup inf formula. This may be problematic when such decisions must be taken by the SIP proxy server for bandwidth reservation in real time. Algorithms proposed for the resolution of the sup inf formula do not allow the estimation of the operating point (t, s) analytically.

7.4.2 Equivalent Bandwidth by Analytical Approaches

Analytical approaches focus bit rate generated by sources, and not on the different possible interactions that may occur within the network. In order to characterize the effective bit rate or equivalent bandwidth of a traffic source, we need to select an appropriate model to specify its characteristics in terms of known parameters or metrics. For the purposes of our study, we adopt a two-state model (ON-OFF). Let the following values be associated with one session:

- R_M : Average rate of one connection (Kbps).
- T_{ON} : Average duration of ONperiod (Sec).
- T_{OFF} : Average duration of OFFperiod (Sec).
- R_{ON} : Average rate in ONperiod (Kbps).
- Q_{ON} : Average file size in ONperiod (Kb) (To be used only with TCP based models)
- X : Buffer size.
- E : Target packet loss probability.

We present two methods to estimate the equivalent bandwidth. In the first method we estimate the number of maximum active ON periods using the binomial law. Then we use $GI/D/1/K$ queuing system model to evaluate the equivalent bandwidth of input flows.

7.4.2.1 Equivalent Bandwidth by the Binomial Law

In this approach we consider a constant number of non-synchronized homogeneous ON-OFF processes. We intend to calculate the probability of having N sources transmitting at the same time (which corresponds to the probability of having N ON periods at the same time for N sources). We denote this probability $\Pr(n_{ON} = N)$.

We denote P_{ON} (resp. P_{OFF}) the occurrence probability of ON period (resp. OFF period). Then, we have:

$$P_{ON} = \frac{T_{ON}}{T_{ON} + T_{OFF}} \quad and \quad P_{OFF} = 1 - P_{ON} \tag{7.7}$$

The probability of having i simultaneous ON periods out of N sources is given by a binomial law of order$N$:

$$P_i = \Pr(n_{ON} = i) = \frac{N!}{i!(N-i)!} P_{ON}^i (1 - P_{ON})^{N-i} \tag{7.8}$$

Let us denote p the target probability of losing a connection, then the equivalent bandwidth for N sources respecting that no more than $p * N$ connections are lost is calculated

by:

$$BW_{eq} = J * R_{ON} \tag{7.9}$$

While J is calculated by:

$$\sup\{J\}, \quad \sum_{i=0}^{J} \frac{N!}{i!(N-i)!} P_{ON}^{i} (1 - P_{ON})^{N-i} < 1 - p \tag{7.10}$$

In the case of heterogeneous ON-OFF sources, the equivalent bandwidth is estimated per homogeneous type. Let BW_i be the equivalent bandwidth assigned to a connection of type i (with N_i connections of type i); let I be the number of connection types; the equivalent bandwidth BW for the I connection types is given by:

$$BW = \sum_{i=1}^{I} BW_i \tag{7.11}$$

Note that the equivalent bandwidth calculated by this method is not related to the buffer size. Indeed, it uses a first order approximation of the traffic by its average rate during ON periods. Unfortunately, this is not sufficient as traffic characteristics can not be resumed to the average rate during ON periods. Indeed, the binomial estimation can be used as a first order approximation of the equivalent bandwidth.

7.4.2.2 Equivalent Bandwidth by Renewal Process Approach

The renewal process approach is an approximation of the superposition of ON-OFF processes. In this method we study the superposition of N ON-OFF processes as a $GI/D/1/K$ system. In order to evaluate the equivalent bandwidth of N ON-OFF processes we study the packet loss in $GI/D/1/K$ system. The packet loss probability is calculated as a function of the following parameters:

- K : The buffer size in packets
- ρ : System load with $\rho = \frac{\lambda}{\mu}$ while λ is the aggregate arrival rate of input ON-OFF processes and μ is the average service time.
- c_a^2 : The squared coefficient of variation of the input arrival process.
- c_s^2 : The squared coefficient of variation of the service time process.

Our approach is based on packet loss approximation formulas for the $GI/G/1/\infty$ queue presented in [Whi83, NKT91]. Let the number of clients in the queue including the one being serviced, be denoted by M. Whitt [Whi83] expresses the average and the second moment of the number of clients M as:

$$\begin{aligned} E(M) &= \rho + \tfrac{\rho^2(c_a^2+c_s^2)g}{2(1-\rho)} \\ E(M^2) &= E^2(M)(c_M^2+1) \end{aligned} \tag{7.12}$$

g is a weighing factor depending on the value of c_a^2:

$$g = \begin{cases} e^{\left[-\frac{2(1-\rho)(1-c_a^2)^2}{3\rho(c_a^2+c_s^2)}\right]} & c_a^2 < 1 \\ 1 & c_a^2 \geq 1 \end{cases} \tag{7.13}$$

The parameter c_M^2 is defined as:

$$c_M^2 = \frac{Y_1 Y_2}{Y_3} \tag{7.14}$$

Y_1 is the value of $Var(M)$ given by:

$$Var(M) = \lambda E(W) + \rho + \rho^2 c_s^2 + \lambda^2 Var(W) \tag{7.15}$$

W denotes the steady state waiting time before beginning service. The average and the variance of W are given by:

$$\begin{array}{l} E(W) = \frac{\tau\rho(c_a^2+c_s^2)g}{2(1-\rho)} \\ Var(W) = E^2(W)c_w^2 \end{array} \tag{7.16}$$

τ is the mean service time and c_w^2 is the squared coefficient of variation of the waiting time expressed as:

$$c_w^2 = \frac{c_D^2 + 1 - \sigma}{\sigma} \tag{7.17}$$

σ is the delay probability whose value is:

$$\sigma = P(W > 0) = \rho + (c_a^2 - 1)\rho(1-\rho)h \tag{7.18}$$

$$h = \begin{cases} \frac{1+c_a^2+\rho c_s^2}{1+\rho(c_s^2-1)+\rho^2(4c_a^2+c_s^2)} & c_a^2 \leq 1 \\ \frac{4\rho}{c_a^2+\rho^2(4c_a^2+c_s^2)} & c_a^2 \geq 1 \end{cases} \tag{7.19}$$

c_D^2 is the squared coefficient of variation of the conditional delay given that the server is busy. Its value when service time is deterministic is given by:

$$c_D^2 = \frac{2\rho - 1 + 4(1-\rho)}{3(c_s^2+1)^2} \tag{7.20}$$

Finally Y_2 and Y_3 are given by:

$$\begin{array}{l} Y_2 = \frac{1-\rho+\sigma}{\max\{1-\sigma+\rho, 0.000001\}} \\ Y_3 = \max\{\rho + \lambda E(W), 0.000001\} \end{array} \tag{7.21}$$

The maximum is used to avoid division by zero. In order to compute the packet loss when finite buffer is considered the two first moments of the packet loss distribution are not sufficient. The distribution itself is needed, which can be obtained by a continuous distribution fit as shown in [Whi83]. Thus $\Pr(M > x)$ is expressed as a function of c_M^2 value as follows:

Case $c_M^2 > 1.01$

$$\Pr(M > x) = p(e^{-\gamma_1 x} - e^{-\gamma_2 x})$$
$$\text{where } p = \frac{\left(1+\sqrt{\frac{c_M^2-1}{c_M^2+1}}\right)}{2}$$
$$\text{And } \gamma_1 = \frac{2p}{E(M)}, \gamma_2 = \frac{2(1-p)}{E(M)}$$

Case $0.99 < c_M^2 < 1.01$

$$\Pr(M > x) = e^{-\frac{x}{E(M)}}$$

Case $0.501 < c_M^2 < 0.99$

$$\Pr(M > x) = \frac{(\gamma_1 e^{-\gamma_2 x} - \gamma_2 e^{-\gamma_1 x})}{(\gamma_1 - \gamma_2)}$$
$$\text{where } \gamma_1 = \frac{\gamma_2}{\gamma_2 E(M) - 1}$$
$$\text{And } \gamma_2 = \frac{2}{E(M) + \sqrt{2Var(M) - E^2(M)}}$$

Case $c_M^2 < 0.501$

$$\Pr(M > x) = e^{-\gamma x}(1 + \gamma x)$$
$$\text{Where } \gamma = \frac{2}{E(M)}$$

Note that the suggested heuristic estimates the packet loss probability as a function of buffer size, average input rate, average output rate and the squared coefficient of variation for both arrival and service processes (c_a^2 and c_s^2). Specifically, the last two parameters play an important role in estimating the equivalent bandwidth of input traffic.

7.4.3 Erlang Blocking Probability

Till now we were considering constant number N of ON-OFF connections. In the general case we model the flow arrival process at the call level as Poisson process. Each flow is defined by its call arrival rate λ, ON period average rate R_{on}, mean rate R_M, average ON duration T_{ON}. Thus, for N connections with $R_{M,i}$ equivalent bandwidth, $i = 1, ..., N$, the overall equivalent bandwidth BW_{eq} is given by:

$$BW_{eq} = \sum_{i=1}^{N} R_{M,i} \tag{7.22}$$

In the last equation we consider N constant sessions. When Poisson arrival process is considered one can estimate the equivalent number of connections (circuits) for one blocking probability, using Erlang B formula:

$$P(N) = \frac{A^N/N!}{\sum_{i=0}^{N} A^i/i!} \tag{7.23}$$

With $A = \frac{\lambda}{\mu}$.

Hence, for one flow with Poisson arrivals (for sessions), we estimate the equivalent

number of sessions N for one blocking probability B_p and the equivalent bandwidth BW_{eq} is obtained easily by multiplying the equivalent rate of one connection by N.

7.4.4 Numerical Validation

In this section we evaluate the analytical approach of equivalent bandwidth estimation on three types of flows: VoIP, Video and Data. VoIP and Data sessions are modelled as ON-OFF processes with UDP as transport protocol for VoIP and Video, and TCP as transport protocol for Data sessions. Recall that TCP automata reacts to packet losses. Hence, we evaluate the equivalent bandwidth of TCP flows under the hypothesis of loss free transmission. This is justified by the final goal of bandwidth estimation which is allocating resources according to application needs.

Refer to Tables **5.3** , **5.8** and **5.10** (**Chapter 5**) for the definition of G711C, MPEG4 and W1 traffic source models that we will use in this evaluation.

7.4.4.1 Equivalent Bandwidth for G711C VoIP Application

VoIP applications have a common characteristic which is the constant packet size and constant packet inter-arrival time during ON periods. All results concerning the estimation of the equivalent bandwidth of G711C application model hold for other VoIP codec types.

The squared coefficient of variation of service time process in a deterministic service queue for VoIP packets is null ($c_s^2 = 0$) because packet sizes are constant, while the squared coefficient of variation of packet arrival process for N ON-OFF processes is given by [SW86]:

$$c_a^2 = wc_1^2 + 1 - w \tag{7.24}$$

c_1^2 is the squared coefficient of variation of single ON-OFF connection and it is calculated as function of the packet transmission probability p during period ON, constant packet inter-arrival T, T_{OFF} duration:

$$c_1^2 = \frac{1 - p^2}{(T/T_{OFF} + 1 - p)^2} \tag{7.25}$$

$$w = \frac{1}{1 + 4(1 - \rho)^2(N - 1)} \tag{7.26}$$

The average rate is $R_M = P_{ON} * R_{ON}$ and the maximum rate is $R_{ON} = \frac{1}{T}$.

Using the previous formulas we estimate the equivalent bandwidth of constant number of G711C applications with two variable parameters: the buffer size K and the target packet loss probability E. Results are depicted on Figures **7.6** and **7.7**.

We note that the estimated value of the equivalent bandwidth decreases with large buffer size as traffic bursts are more easily absorbed. Besides higher packet loss rate results in lower value of the equivalent bandwidth. The estimated values will be checked in a performance test later.

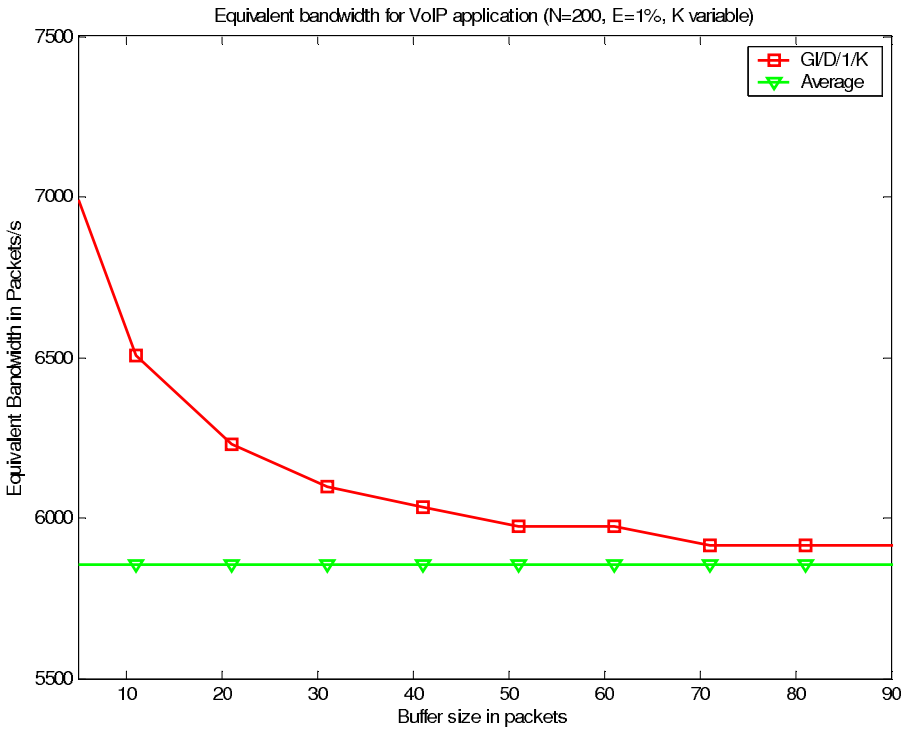


Figure 7.6: G711C Equivalent Bandwidth Estimation (Variable Buffer Size)

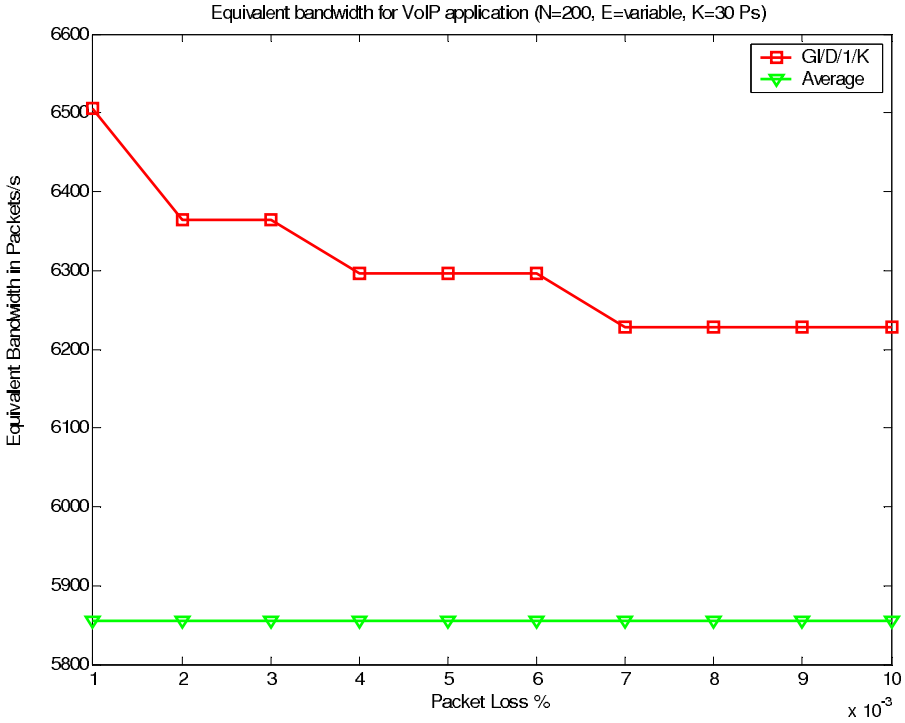


Figure 7.7: G711C Equivalent Bandwidth Estimation (Variable Packet Loss Rate)

7.4.4.2 Equivalent Bandwidth for Web Sessions

The packet arrival process in Web sessions is complicated as it depends on the TCP algorithm. As a consequence, the squared coefficient of variation of packet arrival process can not be estimated analytically. Two solutions to this problem may be proposed: first, we can measure the value of c_a^2 directly on the generated trace. This requires having the generated traffic before evaluating the equivalent bandwidth, which may not be useful when used in a QoS management server (the SIP proxy server). That is why we suggest a second heuristic based on the approximation of packet arrival process during ON periods by a constant process of the same average. Thus, we need to estimate the average rate during the ON period when only the file size is known.

Authors in [SKV01] present a formula to calculate the transfer time when TCP is used on short-lived connections (one ACK per two packets $b = 2$).

$$T(Nb) = RTT\left(\log_{1.57}(Nb) + (f(p, RTT)Nb + 4p\log_{1.57}(Nb) + 20p)\right) + \frac{(10 + 3RTT)Nb}{4(1-p)W_{\max}\sqrt{W_{\max}}} \tag{7.27}$$

The parameters are:

- Nb: The file size in packets.
- $W_{\max}$: The maximum reception window.
- p: The packet loss probability.
- $f(p, RTT) = \frac{2.32(2p+4p^2+16p^3)}{(1+RTT)^3} + \frac{1+p}{10^3 RTT}$

As we are concerned with the ON periods of Web sessions, the file transfer activity is very short compared to the idle period. Thus, this formula is appropriate to our study case. Meanwhile, a major simplification can be done when estimating the equivalent bandwidth with small loss probabilities. Indeed, the contribution of the first term $RTT\log_{1.57}(Nb)$ is dominant, and the equation can be used in its simpler form:

$$T(Nb) = RTT\log_{1.57}(Nb) \tag{7.28}$$

Using the estimated transmission time during the ON period we can estimate the average transmission rate:

$$\lambda_{ON} = \frac{Nb}{T(Nb)}, Nb = \frac{Q_{ON}}{Ps} \tag{7.29}$$

Having the average λ_{ON} we can use the same formulas as for VoIP while considering constant packet inter-arrivals during the ON period (which is only an approximation).

However, the packet service time process is not constant as packet sizes are not constant. However, by using the maximum segment size in the TCP algorithm we generate packets of (MSS+Header= 984+40 =1024 Bytes). As a result we generate 1024 bytes for all packets except the last one (the residual value). Consequently, the squared coefficient of variation of packets service time for Web session can be supposed null ($c_s^2 \approx 0$).

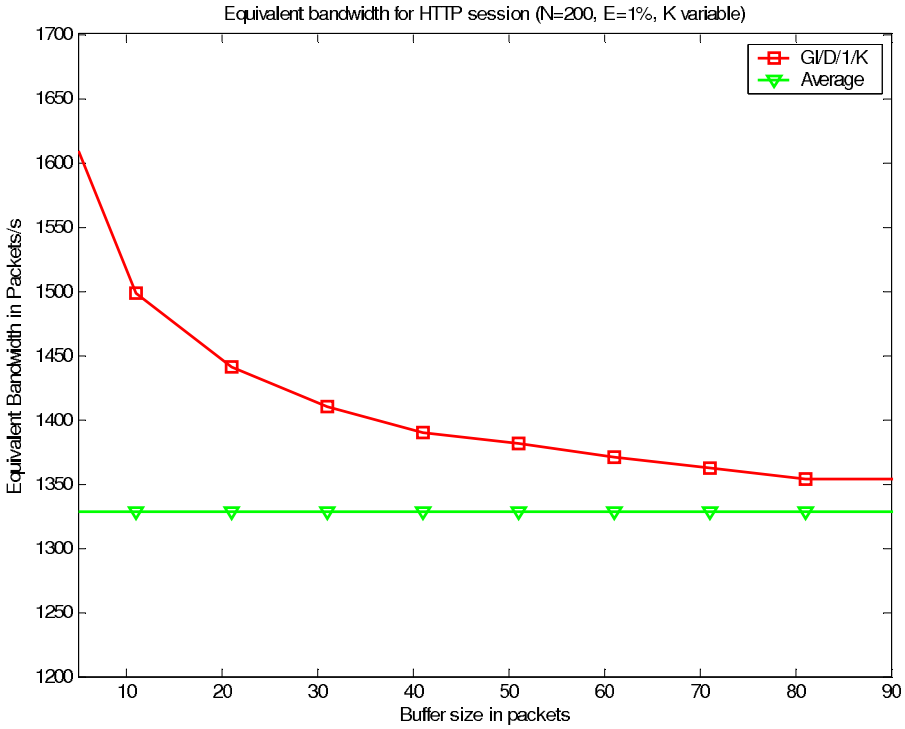


Figure 7.8: HTTP Equivalent Bandwidth Estimation (Variable Buffer Size)

The equivalent bandwidth estimation procedure is similar to the VoIP case, and similar results were obtained (See Figures **7.8**, **7.9**).

The same observation can be made on the equivalent bandwidth for Web sessions as for VoIP applications.

7.4.4.3 Equivalent Bandwidth for MPEG4 Video Application

The MPEG4 Video application traffic has different characteristics from VoIP and Web applications. Indeed, the squared coefficient of variation of packet arrival process can not be estimated analytically, and can not be assimilated to constant packet inter-arrivals as for VoIP applications. In order to estimate the equivalent bandwidth we need to evaluate the MPEG4 traffic c_a^2 value offline on the generated traffic. However, as we use MPEG4 traffic models with constant (1000 bytes) packet sizes (refer to **Chapter 5**), the squared coefficient of variation of service time in a deterministic service queue for Video packets is null ($c_s^2 = 0$).

The equivalent bandwidth estimation procedure is similar to previous cases. We show results only as function of Buffer size (see Figure **7.10**).

7.4.4.4 Performance Validation

We validate the estimated equivalent bandwidth values for VoIP, Video and Web Sessions in network environment. For this purpose, we inject the traffic generated by the three

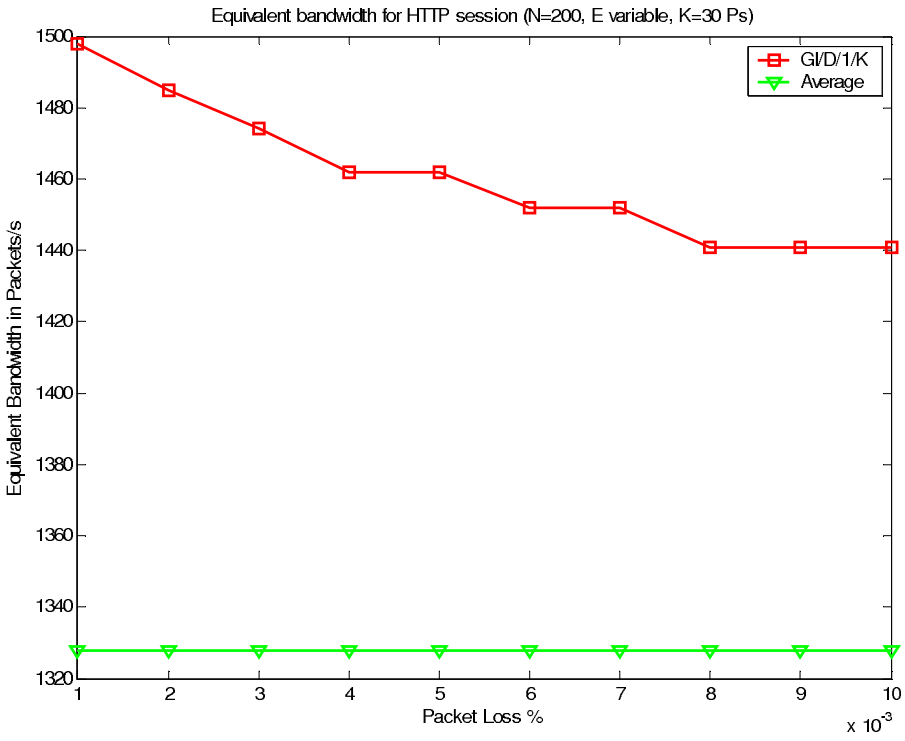


Figure 7.9: HTTP Equivalent Bandwidth Estimation (Variable Packet Loss Rate)

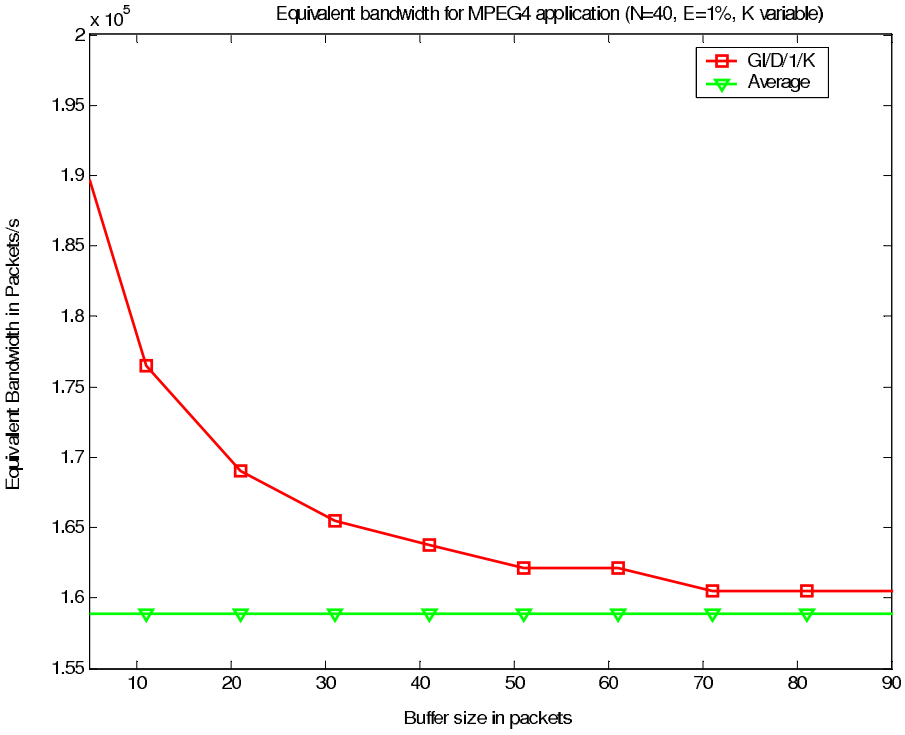


Figure 7.10: MPEG4 Equivalent Bandwidth Estimation (Variable Buffer Size)

types of sessions into a queuing system of deterministic service. The service rate is chosen as a function of equivalent bandwidth. In this test, we evaluate the equivalent bandwidth for 30 packet size buffer at 1% packet loss rate with a variable number of connections. Results for VoIP sessions are listed in Table **7.1**.

Table 7.1: Validation of the Equivalent Bandwidth for G711C Application

N	**EB (GI/D/I/K) Kbps**	**Loss Rate** %
100	3164	1.45
200	6330	1.37
300	9494	1.24
500	15824	1.11
1000	31648	1.03

We note that the $GI/D/1/K$ queue system model underestimates the equivalent bandwidth when the number of connections is small. The observed loss rate is higher than the target value (1.45% instead of 1% for $N = 100$).

Table 7.2: Validation of the Equivalent Bandwidth for Web Sessions

N	**EB (GI/D/I/K) Kbps**	**Loss Rate** %
100	9494	1.9
200	11531	1.5
300	16142	1.38
500	25490	1.09
1000	50980	1.07

In Table **7.2** we show the results for Web sessions traffic. They are similar to VoIP case. Notice that constant packet inter-arrivals approximation during ON periods for Web sessions lead to acceptable results. Although this is not the real behaviour of inter-arrivals, we obtain acceptable loss rate. This approximation allows analytical estimation of the equivalent bandwidth for Web sessions directly.

Similar results where obtained for video traffic. Meanwhile, the squared coefficient of variation of packet arrival process was estimated on the generated video traffic trace.

7.5 QoS Mechanisms with SIP

The goal of this section is to introduce novel mechanisms for QoS management on a session basis. The SIP proxy server will be delegated for the implementation of these mechanisms. The first mechanism relies on the dynamic assignment of the class of service between TCP sessions based on session duration and data volume exchanged during a session. The second mechanism uses the equivalent bandwidth estimation methods to allocate bandwidth per flow. The SIP proxy server is supposed to achieve measures on session durations and data volume exchanged, as well as the equivalent bandwidth estimation based on session parameters exchanged during session initiation phase.

7.5.1 Dynamic Assignment of the Class of Service

Numerous studies show that 80% of internet flows are carried by TCP. It is also shown that 80-90% of the traffic is carried by only 10-20 % of the flows (big file transfers) while 80-90 % of the flows carry only 10-20% of the traffic. It is obvious that TCP requires special attention and particularly the interaction between big and small data transfers must be considered in any QoS provisioning mechanisms. Indeed, several researches dealing with the efficiency of TCP congestion control mechanism in congested networks have been undertaken. However, results show that losses have dramatic consequences on short TCP connections. It was suggested that according higher priority to short TCP connections constitutes a good solution to this problem (e.g. [Aye05]). The question of differentiating long from short TCP connections requires modifications in TCP headers to perform measures on TCP connections (Duration or data volume exchanged) reader can refer to [EV03, PGP04] for some other proposals. In all cases, this issue was always addressed at the connection level. Although, an application level solution to this problem is more appropriate and easier to implement. Indeed, we can consider the user behaviour during all the session as a whole and instead of differentiating short from long connection we distinguish small from big sessions.

Using SIP we can manage user communications at the session level, where assignment of the class of service is based on a session level criterion. This is achieved by supervising mechanisms implemented in the SIP Proxy server in the extended SIP architecture. The main advantage of our approach is that supervising mechanisms relies on measures that are performed usually for billing purposes.

7.5.1.1 Concept of the Dynamic Assignment of Class of Service

The main idea behind session scheduling is to change traffic priority dynamically during communication based on real time measurements. The goal is to minimize the impact of long TCP sessions on both short TCP sessions and real time traffic. Indeed, the TE-SIP server measures in real time the duration of TCP sessions and data volume exchanged during one session. All sessions initiated by the SIP server has the same priority at first. Sessions lasting more than average session duration T_{rs}, or exchanging more than average data volume V_{rs} are automatically declassed into a lower priority class of traffic.

In order to calculate the values of T_{rs} and V_{rs} we introduce the notion of the reference session RS. The reference session RS represents the threshold session activity under which the user session is considered small. The notion of small session may refer to the duration or the data volume of a session. This is quiet different from the connection notion in which long connections are synonym of big file transfers. In fact, TCP sessions may contain long idle periods and the notion of duration may lead to some wrong differentiation between sessions. Once the reference session RS is defined, the theoretical duration and data volume exchanged during reference session RS can be calculated. Let:

- Nbr : Average number of ON periods in a session.
- T_{ON} : Average duration of ON period (Sec).

- T_{OFF} : Average duration of OFF period (Sec).
- R_{ON} : Average rate in ON period (Kbps).
- Q_{ON} : Average file size in ON period (Kb)

In order to calculate the T_{ON} value, we consider the TCP session transmission in the free loss case (the simplified form of TCP connection duration equation (**7.28**)
$T\left(\frac{Q_{ON}}{Ps}\right) = RTT \log_{1.57}\left(\frac{Q_{ON}}{Ps}\right)$ with Ps packet size
The average session duration is given by:

$$T_{rs} = Nbr(T_{ON} + T_{OFF}) \tag{7.30}$$

And the average session data volume is given by:

$$V_{rs} = Nbr.Q_{ON} \tag{7.31}$$

7.5.1.2 Algorithm for the Dynamic Assignment of Class of Service

Here we develop the algorithm based on reference session duration and data volume exchanged. This algorithm is implemented by the TE-SIP server and is applied to all incoming TCP sessions.

1. Calculate the reference session duration T_{rs} and data volume V_{rs}
2. Define two service classes High and Low
3. Accept all incoming TCP sessions with the High service class
4. For all sessions

   If session duration > T_{rs} (or session data volume > V_{rs})

   Declass the session service to the Low class

7.5.2 Resource Allocation

The dynamic assignment of the class of service is a posteriori solution to network congestion and QoS provisioning. Indeed, it minimizes interaction between big and small TCP session based on some threshold. Although it requires less information about TCP sessions, the choice of its threshold may be problematic. Actually, it influences the performance of the algorithm and the overall gain in terms of QoS. On the other hand, the role played by the SIP proxy server can be enhanced using a priori solutions that prevent interaction between big and small TCP sessions. Hence, instead of detecting big TCP sessions after some threshold, users may declare their sessions previously. According to the session type required by the user a different class of service may be assigned and consequently an appropriate QoS is obtained.

7.5.2.1 Concept of Ressource Allocation

Resource allocation requires equivalent bandwidth estimation. The idea is delegate the SIP proxy server to evaluate the equivalent bandwidth of TCP flows per type of session. In order to achieve this estimation the TE-SIP server needs some specific description of initiated sessions. This will be achieved by the SDP protocol associated with SIP.

Assuming a Poisson arrival distribution of client sessions, an equivalent number of sessions N can be estimated by the Erlang B formula for a determined blocking probability. Then an equivalent bandwidth estimation procedure is launched based on session information exchanged during the SDP communication phase. Once the equivalent bandwidth is determined, a bandwidth sharing process is undertaken by Weighted Fair Queuing (WFQ) system at the Edge router. Particularly, weights are chosen as function of equivalent bandwidth and available bandwidth. The goal is to assign the required bandwidth to small TCP sessions, while big TCP sessions share the residual bandwidth. In the following section, we present the algorithm to implement in the TE-SIP proxy server while considering only two flow types. Of course, this approach may be extended to several flows of different types (not only TCP sessions).

7.5.2.2 Algorithm for Ressource Allocation

Consider two TCP sessions: small and big. The goal is to allocate resources for small TCP sessions. The TE-SIP proxy server handles the following parameters:

- Bp The blocking probability.
- D Poisson session arrival rate (Session/sec).
- B Available bandwidth.

And the following session parameters:

- Nbr : Average number of ON periods in a session.
- T_{OFF} : Average duration of OFFperiod (Sec).
- Q_{ON} : Average file size in ONperiod (Kb).

The algorithm is summarized in the following steps:

1. For small TCP sessions calculate the equivalent number of sessions N

$$B_p(N) = \frac{A^N/N!}{\sum_{i=0}^{N} A^i/i!}$$

With

$$A = \frac{\lambda}{\mu}$$

2. Estimate the equivalent bandwidth B_{eq} for N sessions
3. Define two classes of service $High$ for small sessions and Low for Big sessions
4. Adjust the WFQ weights so that:

$$W_S = \frac{B_{eq}}{B}$$

And

$$W_B = \frac{B - B_{eq}}{B}$$

5. Assign small TCP sessions to the $High$ class of service
6. Assign big TCP session to the Low class of service

7.5.3 Call Admission Control for TCP sessions

So far only the issue of minimizing interaction between flows was addressed, first by dynamic assignment of the class of service and then by resource allocation. Indeed, a more natural role can also be assigned to the TE-SIP proxy server, which is Call Admission Control. Whether it concerns small or big sessions, the overhead that may be induced by the initiation of new sessions may be considerable when network fall into congestion.

Thus, the dynamic assignment of the class of service will have no effect if the number of small TCP sessions exceeds system capacity. Moreover, big sessions will endure extremely long time of service due to the system overhead. Indeed, this issue is of particular interest for TCP sessions as the session duration is tightly linked with the loss rate observed on network links. If the number of accepted sessions is larger than system capacity, higher packet loss rate may be observed and longer transmission times are needed (because of TCP retransmission mechanisms). Consider users connecting to the system according to Poisson arrivals with D session arrival rate (Session/sec). Let T be the session duration then the average number of users N present in the system may be obtained by $Little's$ formula:

$$N = D * T \tag{7.32}$$

Notice that $N \underset{T \to \infty}{\to} \infty$.

Hence, the increase of session duration results in increasing number of active sessions in the system. Indeed, the session management overhead for the SIP proxy server will increase rapidly affecting the performance of the server itself.

Call admission control could be associated with the previous proposed algorithms to guarantee normal functioning of the network. This requires the definition of the Call Admission Control Threshold (CAC_{th}) per type of sessions. When resource allocation is performed this threshold is simply defined by bandwidth reserved for one flow (denoted

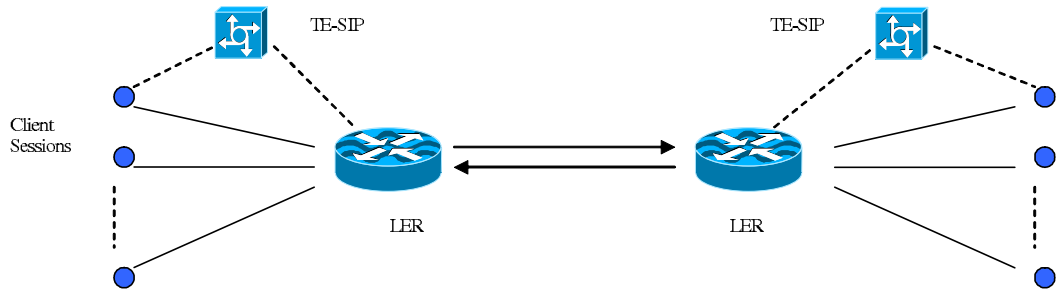


Figure 7.11: SIP Simulation Network

B_{eq}). Indeed, the estimation of B_{eq} rely on the number N of sessions, thus:

$$CAC_{th} \leq N \tag{7.33}$$

The session information used for equivalent bandwidth estimation and allocation is also used by CAC algorithm. Thus, for every new incoming connection the equivalent bandwidth necessary to allow the transmission of the flow is calculated. If the estimated value does not exceed the maximum bandwidth allocated for the flow the connection is accepted otherwise it is rejected.

We note that resource allocation procedure is based on session information exchanged before session initiation. The type of the session declared by the user is determinant for accepting or rejecting his demand. Meanwhile, in the case of wrong session type declaration, low priority sessions may be initiated as high priority ones causing the deterioration of the overall performance of the system. Therefore, it is possible to combine the dynamic assignment of the class of service algorithms to demote wrongly declared sessions to lower class of service as a posteriori validation mechanism.

7.5.4 Numerical Validation

The QoS session mechanisms are tested in a simple network of two nodes representing the two LER routers. The DiffServ domain is modelled by a bottleneck link between two LER routers. Bottleneck delay is of 10 ms (used for the RTT calculation). An SIP Proxy server is charged of initiating different kinds of sessions (see Figure **7.11**).

We consider three types of web sessions (W1, W2, W3), and a VoIP application session (G729). Four flows are generated (one for each type of session) in order to evaluate the suggested QoS mechanisms. Indeed, the G729 flow is only used for performance evaluation ends while data sessions are handled dynamically by the QoS mechanisms. Session parameters are listed in Table **7.3**.

Given the above session parameters we calculate the flows parameters that will be used when applying QoS mechanisms by the SIP proxy server (see Table **7.4**).

Recall that the estimation of TCP session duration is only possible in free loss transmission. This estimation is used to determine the reference session duration (and volume) for the dynamic assignment of the class of services.

Table 7.3: Session Parameters for SIP Tests

Session	Q_{ON} **Mean (KB)**	Q_{ON} **Variance (KB)**	T_{OFF} **Mean (Sec)**	T_{OFF} **Variance (Sec)**
W1	20KB	40KB	10 sec	20 sec
W2	50KB	100KB	20 sec	40 sec
W3	100KB	200KB	30 sec	60 sec
VoIP	**Packet Size (Bytes)**	**Packet Inter-arrival (ms)**	T_{ON} **(sec)**	T_{OFF} **(sec)**
G729	70	30	0.352	0.65

Table 7.4: Flows' Parameters

Flow	D **Session/sec**	**T Sec**	**Blocking Probability**	N **Equ**	B_{eq} **Kbps**
W1	0.1	101.3	0.01 %	21	283.5
W2	0.05	203.9	0.01 %	21	411.6
W3	0.0333	307.8	0.01 %	21	546
G729	0.0556	180	0.01 %	21	130.2

7.5.4.1 Test of the Dynamic Assignment of the Class of Service Algorithm

We consider only two classes of traffics High (Priority 1) and Low (Priority 2). All sessions start transmitting packets in the High traffic class. VoIP sessions stay always in the High traffic class and do not change their class. Declassing sessions concerns only TCP sessions. The goal of our test is to give short Web sessions represented by W1 sessions higher priority on other TCP sessions using dynamic assignment of class of service. In this case, the W1 session is the reference session RS and its transmission time without losses is the reference time (T_{RS}=101.3 sec). The data volume exchanged during the RS session is V_{RS}=200 Kbytes based on W1 session parameters. We compare the results of proposed scheduling mechanisms with FIFO queue system without any priority classes. Results are shown on Table **7.5**.

Results show no big difference in performance when using time based session scheduling. Indeed, if we analyze the activity of W1 sessions we note that idle periods are very long compared to activity periods. Declassing sessions according to the time passed on the network is not profitable in this case. In fact, the communication duration criterion could only be used for FTP type like sessions where there are no idle times.

Conversely, we see that the performance of W1 sessions, G729 sessions has improved considerably in the case of session declassing based on data volume exchanged during a session. In fact, W1 session duration is closer to theoretical value without losses, while G729 sessions endure less packet loss rate.

7.5.4.2 Test of Ressource Allocation Algorithm

Using the same previous sessions, we calculate the weights of WFQ system based on the estimation of the equivalent bandwidth per flow. We consider two WFQ queues

Table 7.5: Flows' Statistics with Different Priorities

Flow		**Loss %**	**Delay ms**	**Average Session Duration sec**
FIFO	W1	3%	155	299
	W2	2.9%	151	401
	W3	2.7%	156	497
	G729	3.5%	170	180
Time Priority	W1	2.9%	152	270
	W2	2.5%	149	395
	W3	2.3%	145	478
	G729	1.6%	165	180
Volume Priority	W1	0.9%	78	105
	W2	2.7%	150	393.9
	W3	3.1%	152	487.8
	G729	1.1%	86	180

in the edge router with two corresponding traffic classes: High and Low. The weights are calculated to allocate the required bandwidth for the High traffic class. The Low traffic class takes the residual bandwidth. Table **7.6** lists the calculated weights based on sessions parameters defined before.

Table 7.6: WFQ Weights for Bandwidth Sharing

Flow	**Eq BW Kbps**	**Traffic Class**	**Weight**	**Bottleneck BW Kbps**
W1+G729	413.7	High	11	1200
W2+W3	957.6	Low	19	1200

Results (see Table **7.7**) are better than the first mechanism. Loss rate and delay on the W1 flow and G729 flow are smaller than previous tests. Especially W1 session average duration is very close to theoretical estimation. Even W2 and W3 sessions perform better compared to the FIFO system case. This is especially due to the better bandwidth utilization and the isolation factor resulting in less interaction between flows.

Table 7.7: Flows Statistics with WFQ

Flow		**Loss %**	**Delay ms**	**Average Session Duration Sec**
BW allocation	W1	0.3%	25	101.9
	W2	2%	145	375
	W3	2.1%	151	437
	G729	0.05%	17	180

To illustrate the robustness of this approach we show on Figure **7.12** the evolution of session durations while the link bandwidth is reduced from 1200 Kbps to 800 Kbps by

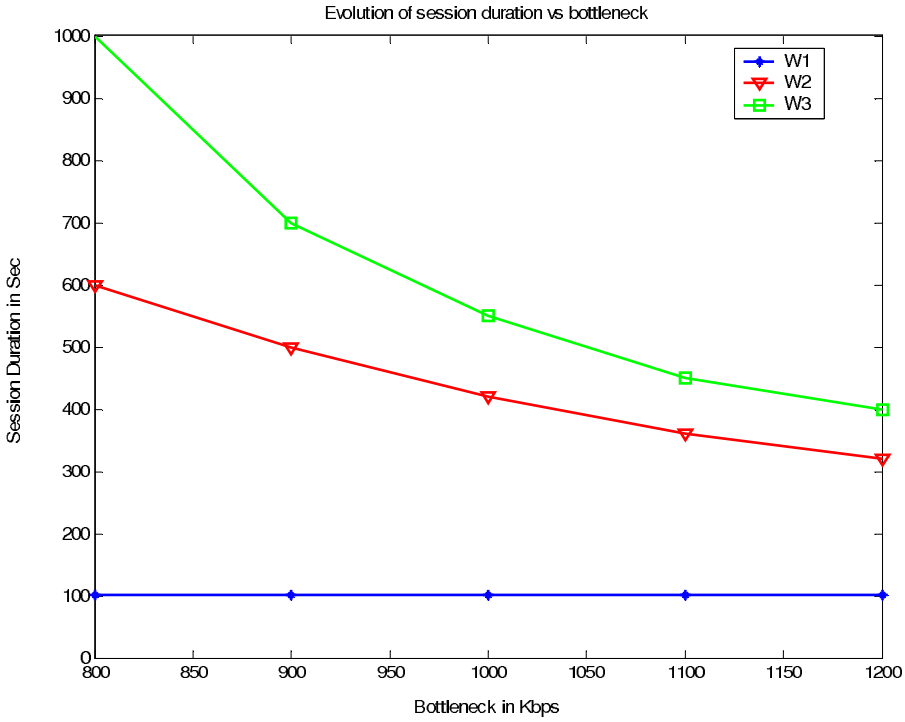


Figure 7.12: Evolution of Session's Durations vs Bottleneck Capacity

100 Kbps step. The TE-SIP proxy server adjusts weights of the WFQ queuing system to guarantee the required bandwidth for W1 and G729 flows while the W2 and W3 flows gets always the residual bandwidth. The curves show that the measured W1 session duration is stable while the measured W2 and W3 session durations increase as the bandwidth is reduced.

The stability of the duration of small TCP session could be seen as a QoS parameter for Web sessions. It has bigger impact in wireless networks where bandwidth is a precious resource, and guaranteed average throughput is an important parameter of the service.

7.6 Conclusion

In this chapter we presented an SIP based framework for QoS at the session level. Using the SIP proxy server with extended architecture over DiffServ domain, we can implement session scheduling mechanisms and bandwidth allocation strategies to minimize the interaction between small and big TCP sessions. The suggested mechanisms rely on session level real time measurements of session duration and data volume exchanged during a session. The required measurements do not cause additional overhead as they are achieved for billing purposes. However, bandwidth allocation requires more specific session description that should be exchanged between the SIP proxy and the UAC using the SDP protocol. Bandwidth allocation is based on equivalent bandwidth estimation

per flow. Indeed, we proposed a $GI/D/1/K$ queue system model to evaluate the equivalent bandwidth using the first and second order moment of packet arrivals and packet service processes. Good approximations of the equivalent bandwidth are obtained using this model, meanwhile the arrival process in some cases is not easy to characterize (Video for instance). Some heuristics to measure the equivalent bandwidth in real time could be considered using iterative algorithms.

In this study we only considered homogeneous flows (per type of session) for the estimation of equivalent bandwidth. In fact, the estimated equivalent bandwidth depends on traffic mixes and may be strongly influenced by the different packet sizes according to application types (and transport protocols). Particularly, in VoIP application packet sizes are very small comparing to Video and TCP traffic resulting in important covariance on the packet service process and the overall performance.

Finally, we would like to mention that the QoS mechanisms proposed are applied on a session basis, and this may result in an extra delay on session initiation process for users. Some aggregated reservation techniques [RLB+05] may be useful in this case to enhance the SIP proxy server response time.

Chapter 8

Conclusion and Prospects

The emergence of new multimedia services on the Internet and the development of new wireless access networks result in complex heterogeneous networks. This heterogeneous environment motivated the convergence towards a unique packet switching core network. The unique core network is certainly the future Internet in which all multimedia services are transported by internet protocol IP, and accessed by users seamlessly regardless access technology. However, one of the most complex aspects of this convergence is how to provide Quality of Service (QoS) to mass multimedia flows transported by IP. Especially, when it concerns real time flows with very strict service level agreements (SLAs) such as VoIP and high definition videos. Indeed, designing such networks requires robust QoS mechanisms as well as modelling and simulation tools capable of evaluating precisely the performance of these mechanisms along with target multimedia applications. Moreover, with large scale networks the efficiency of modelling and simulation techniques is an important factor in the design and evaluation process.

We focused throughout this thesis on multimedia traffic modelling issues in heterogeneous networks. The goal is to provide tools and models for multimedia traffic in order to achieve reliable and efficient performance evaluation studies in heterogeneous networks. Reliability and efficiency of multimedia traffic modelling is confronted with the diversity of multimedia applications and their complexity. The packet generation profile is tightly linked with the user behaviour and transport protocols. Our first contribution is to provide a generic hierarchical framework allowing simple and generic modelling of traffic sources regardless their complexity. Using the generic framework, audio, video and data application are described easily based on a hierarchical representation of the application model. The hierarchical representation includes three levels: Session, Activity and Packet. This framework allows a better understanding of multimedia traffic by suggesting a clear separation between the two main transport protocols on the Internet Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Indeed, the closed-loop behaviour of TCP impacts the traffic generation profile of TCP-based applications and this issue should be handled carefully.

Using the generic framework we addressed two main issues in traffic modelling: IP traffic modelling (or trace based modelling) and multimedia traffic modelling. Trace

based modelling may be applied to any captured traffic trace, in order to define a corresponding traffic generator. Our contribution concerns the generalization of the M/G/∞ process to estimate mixed correlation structure (SRD+LRD) in traffic traces, and mixed probability distribution functions for traffic entities. The proposed M/G/∞ model belongs to window based traffic models category in which traffic trace is divided into constant time slots, and traffic parameters are estimated based on the resulting data series. The generalized M/G/∞ process offers great flexibility in characterizing and producing correlation structure and probability distributions with respect to traffic trace statistics. However, we show out an important performance limitation on window based traffic models when used for IP traffic modelling. It concerns the transformation of the generated data slots into IP packets. We observed optimistic performance of generated traffic compared to traffic trace. In fact, some general assumptions concerning the use of exponential or uniform packet inter-arrival distribution inside slots, and the use of the same packet distribution over all time slots are behind this limitation of performance. We suggested to estimate the packet inter-arrival and packet size distributions on a per-group of slots basis. Our results show that this modification enhances the performance of the model and should be considered in other data slot (or window) based traffic models when used for IP traffic modelling.

Although the proposed M/G/∞ model is appropriate for trace based traffic modelling, some aggregated (or superposed) multimedia traffic may lead to less complex models. Indeed, one of our objectives was to show under which conditions simple approximations such as Poisson process could be used when superposing a high number of identical multimedia applications. Our results show that superposed Audio (or VoIP) applications may be modelled by Poisson process for light to medium traffic loads, while heavy traffic loads may be approximated by an MMPP-2 process. The same result may apply to superposed MPEG video traffics for light to medium traffic loads, although the number of superposed sources has to be greater than in the audio case. On the other hand, superposed TCP applications are more complicated because of the closed-loop behaviour of TCP. Actually, a dynamic behaviour of superposed TCP traffic statistics under variable loss rate is observed which makes it difficult to use usual aggregated traffic models for TCP traffic. Such models do not react to network congestion conditions like TCP. Two solutions were proposed to model superposed TCP traffic: first, we preserve TCP dynamics by superposing TCP sessions, using an equivalent session model in order to reduce the number of simulated sessions. Specifically, we introduce equivalent ON-OFF process that aggregates OFF periods when these ones are very long. Although, the equivalent ON-OFF process gives a good approximation, this solution still shows limitations when large number of flows is considered. That is why a second approach based on differential traffic theory is proposed, in which we substitute TCP event-driven simulation by differential analytical simulation. The goal is to propagate rates instead of packets, while preserving the transient behaviour of TCP. This is achieved by mixing differential equations with state control events. The differential model is developed in mono-source and network configurations, and its implementation shows very good performances. The model captures very well the transient behaviour of one source and the average behaviour

of many sources compared to event-driven simulations. The differential technique is very promising as most of Internet traffic is transported by TCP.

Finally, we addressed the QoS problematic from a session level point of view. Hence, we use the Session Initiation Protocol (SIP) as a supervising protocol in order to guarantee QoS requirements for multimedia flows. Two strategies were proposed: stochastic scheduling of sessions and bandwidth allocation on a per-flow basis. Results obtained with this approach are very good and offers new possibilities for developing a QoS framework based on SIP architecture. SIP is the protocol selected by the 3GPP as the signalling protocol for IP based mobile networks.

Throughout this thesis we studied and analyzed issues related to multimedia traffic modelling and QoS provisioning in heterogeneous networks. Our research pointed out the inadequacy of classical event-driven packet simulation techniques in practice. Analytical simulation techniques are more appropriate for network evaluation tools. There are many perspectives in this direction. One solution is provided by the differential traffic theory [GGB+01] that offers the possibility to replace event-driven techniques by differential and hybrid simulation techniques. Besides, solutions that are based on flow level modelling techniques instead of packet level ones are a good alternative. Indeed, traffic models on flow level are much easier to execute, besides they can be easily combined with session level QoS frameworks such as SIP signalling protocol. However, packet traffic models may still be necessary, that is why simple analytical tractable models should be privileged in order to allow analytical evaluation studies. This could be achieved by Poisson based traffic models (BMAP, MMPP, ...). Although, these models are approximate compared to models like $M/G/\infty$, FARIMA, FGN, ... they may allow analytical evaluation in queuing networks. In fact, the error produced by these approximations may not be as important as estimation errors on traffic matrices for example. This is particularly important in the context of heterogeneous networks where telecommunication operators can not define precisely traffic matrices on their networks and they are forced to use approximate traffic matrix inference procedures. In this case more attention should be paid to the estimation of user populations rather than single user behaviour.

Appendix A

Traffic Source Modeller

A.1 Introduction

In **Chapter 3** we presented a generic framework for multimedia traffic modelling. This framework has been implemented into a more complete modelling and simulation tool called Traffic Source Modeller (TSM). TSM is composed of several collaborating modules used primarily for describing and implementing multimedia application models. Models designed with TSM can be used by a simulation module in order to perform performance evaluation studies.

The TSM software (TSM main window is shown on Figure **A.1**) aims to provide a workbench for traffic modelling and evaluation in IP networks. In TSM we find four main modules:

- Source Modeller
- Trace Analyser
- Simulation
- Traffic Emulator

Basically, the Simulation module provides users with a complete topology and traffic matrix editor. Users can use this module to evaluate the performance of proprietary network topologies in presence of user populations based on predefined traffic models or new specific traffic models. Building specific application models or traffic models is done using the Source Modeller module. Of course, building traffic models require trace analysis and estimation tools in order to specify traffic parameters (packet size distribution, packet inter-arrival distribution, correlation structure, moments ...). All trace analysis functionalities whether it concerns building new traffic models or not are achieved using the Trace Analyser module. Finally, traffic models can be tested in a simulation environment but also they can be used by traffic emulator to generate packets and inject them into real networks.

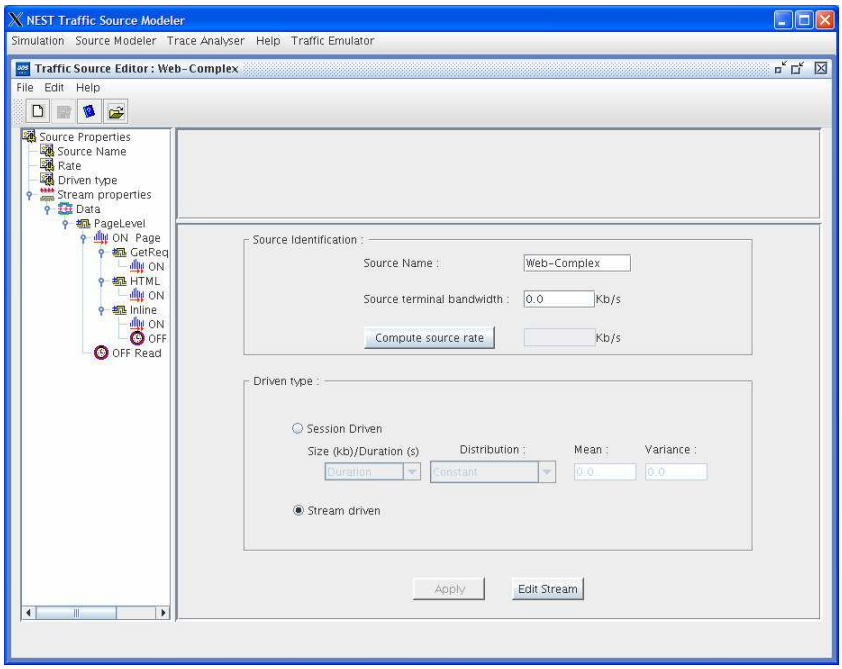


Figure A.1: TSM Main Window

In brief, a user of this framework can make use of the tool to model multimedia applications and evaluate their performances in a network according to the flow chart illustrated in Figure **A.2** .

Of course TSM modules can be used separately to achieve individual tasks. Indeed, the previous flow chart represents the logic in which the tool was designed in order to offer maximum flexibility for users.

In this appendix we will describe the Source Modeller and Trace Analyser modules as they are issued from this thesis. Simulation and Traffic Emulator modules are out of the scope of this thesis.

A.2 Trace Analyser

Trace Analyser module implements estimation and analysis tools that helps the user to characterize a traffic trace and estimate traffic related parameters. We give below details about estimation techniques: EM and Levenberg-Marquardt algorithms (presented in **Chapter 3**).

A.2.1 EM Algorithm

The EM algorithm in a traffic modelling context is especially used for estimating probability distributions (The distribution of slot size, packet size, packet inter-arrival, ...). First, a data series issued from a traffic trace is loaded in the tool and then the user may choose to fit the data series with any desired combination of basic distributions $y(x)$

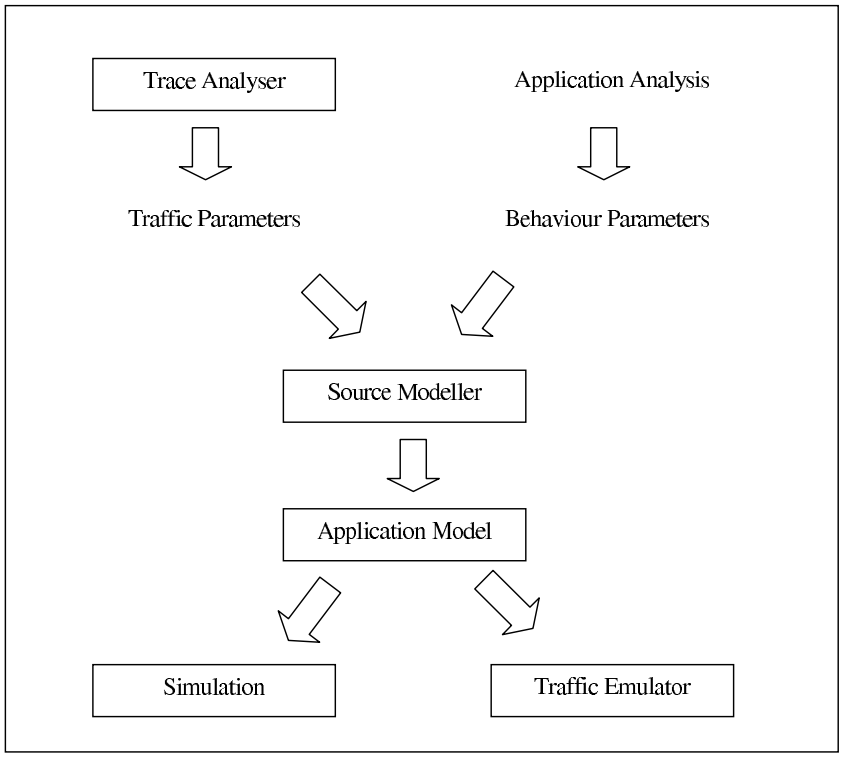


Figure A.2: TSM Operation Flow Chart

(normal, exponential, gamma, lognormal, ...).

$$y(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \cdots + \alpha_n f_n(x) \tag{A.1}$$

Where $\{\alpha_i, i = 1 \cdots n\}$ are positive weights verifying $\sum_{i=1}^{n} \alpha_i = 1$. The algorithm starts with arbitrary initial set of weights.

After the convergence of the algorithm, final weights according to the accuracy error level chosen by the user are listed in the estimated values section. Figure **A.3** depicts the window corresponding to EM algorithm.

A.2.2 Levenberg-Marquardt algorithm

We estimate the correlation structure of a time series using the Levenberg-Marquardt algorithm. This algorithm optimizes the parameters of a function model in order to give the best fit with input data. The correlation function model used for traffic modelling is based on the combination between three elementary correlation functions corresponding to Markov, SRD and LRD correlations defined as follows:

$$\rho(k) = \alpha_1 e^{-b_1 k} + \alpha_2 e^{-b_2 \sqrt{k}} + \alpha_3 (k+1)^{-b_3} \tag{A.2}$$

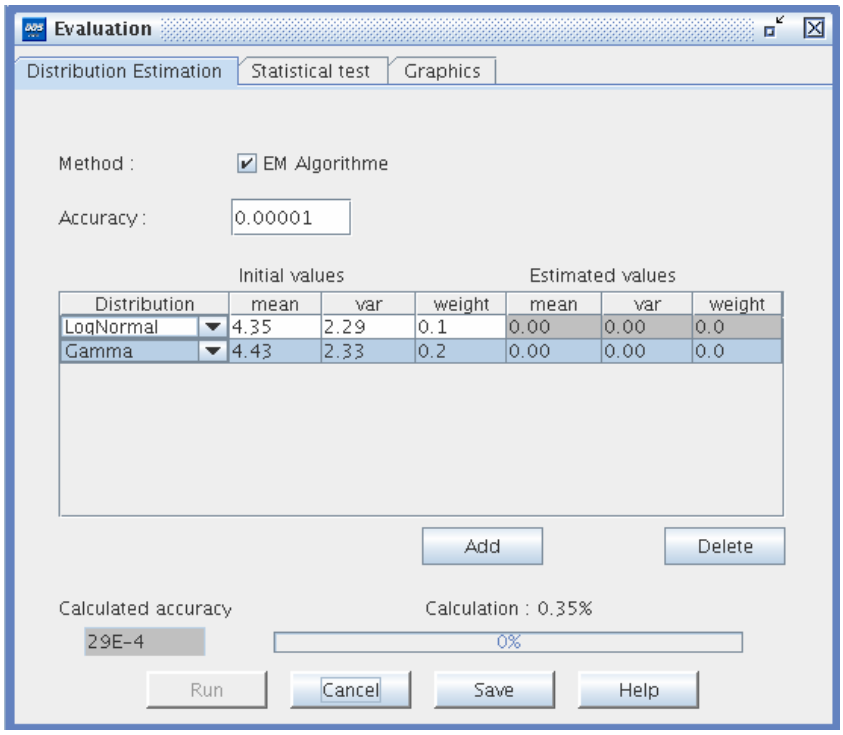


Figure A.3: The EM Algorithm Window

The Levenberg-Marquardt algorithm performs an optimization on $\{\alpha_i, i = 1 \cdots 3\}$, $\sum_{i=1}^{3} \alpha_i = 1$ and $\{b_i, i = 1 \cdots 3\}$ parameters of the autocorrelation function. Figure **A.4** depicts the window corresponding to the estimation of the autocorrelation function and Hurst parameter estimators.

Both the ACF and probability distribution of data slots are input parameters of the M/G/∞ process used for IP traffic modelling.

Besides, on the same window we have three estimators of the Hurst parameter. The value of Hurst parameter gives an indication about the LRD characteristic of a time series.

The Trace Analyzer module implements also moment estimators and empiricial PDF and CDF of a time series.

A.3 Source Modeller

The Trace Analyser module provides the user with a friendly tool to estimate traffic parameters directly on a traffic trace. The next step is to define the traffic generator corresponding to the traffic model. This is done in the Source Modeller module. Using this module user can describe precisely multimedia applications and usual traffic models.

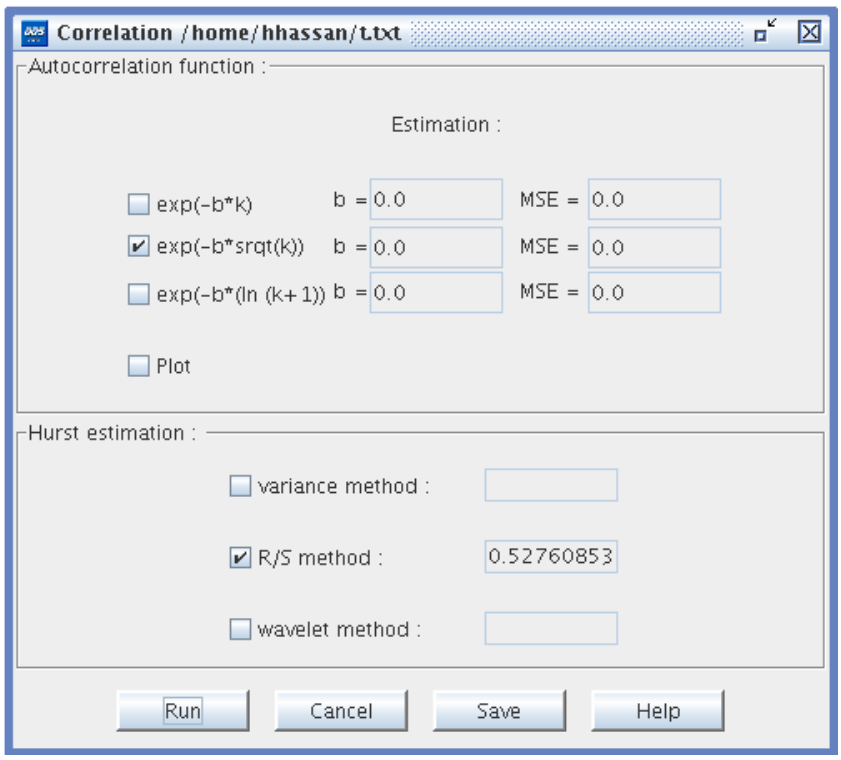


Figure A.4: Correlation Structure Window

A.3.1 Structure

The flexibilty we seek to achieve with the generic framework requires an efficient implementation structure. The three level model (see **Chapter 3**) allows the separation of the transport technology (or protocol) from the application description. Thus, whatever the client arrival process or the activity achieved by clients during sessions, packets can be transported either by UDP or TCP. From a traffic modelling point of view, this is very important as it allows the evaluation of application performance versus the transport protocol.

Multimedia application modelling concerns primarily the description of the exchanged data during the session. We use the activity level to define the application behaviour. In fact, the basic element of this level is the ON period. This period can be simple or complex with sub-ON periods. An active period ON is followed by an idle period OFF. It can also be followed by another active period ON (with different characteristics). We choose to use a traffic entity called Pattern, representing a set of periods. This Pattern can have a distribution of occurrence into time. So if a Pattern corresponds to a Web page, it can contain ON period for the page downloading time and OFF period for the reading time. Moreover, as one Web session may contain a random number of pages, we can set the occurrence of this Pattern to the desired page number distribution. In the case of FTP for example, we have a simple pattern with only one active period ON without any OFF periods.

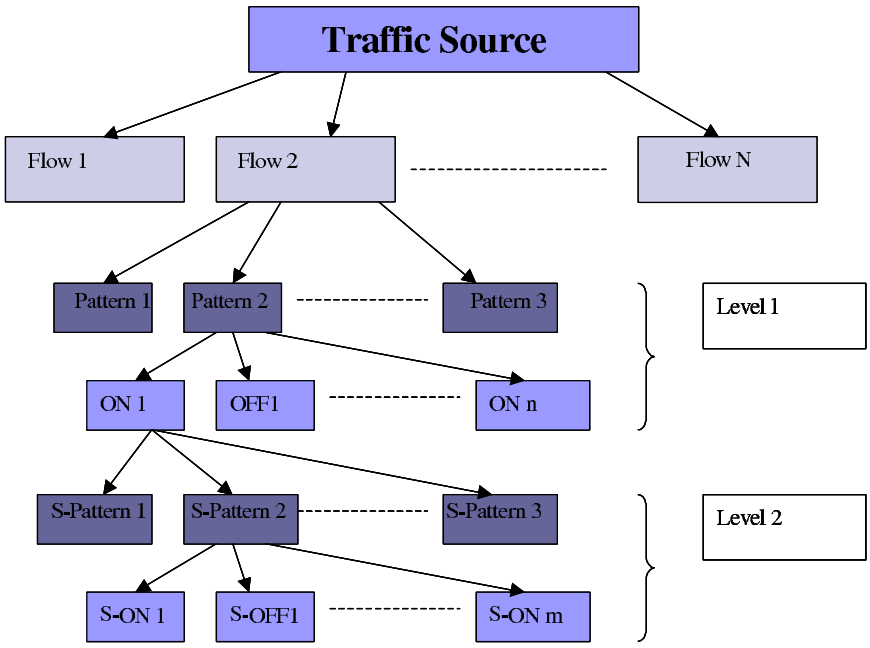


Figure A.5: Implementation Structure of the Generic Framework

The fact that one ON period can have sub-ON periods, which themselves can have sub periods, make the framework very generic. Thus, the activity level can be nested to N theoretically. Figure **A.5** depicts the implementation structure.

The framework seeks to represent a maximum of applications. We have one category of internet applications based on Streaming Control Transmission Protocol (SCTP), Real Time Streaming Protocol (RTSP), RTP and RTCP protocols, in which we need to represent two flows in parallel: one flow for data (using RTP for example) and another flow for control or signalling (using RTCP, for example). Some applications need also to describe two synchronized periods within two different flows. As an example of this behaviour we have applications using Session Initiation Protocol (SIP). To model these different cases, an application model may have several flows. These flows are composed of traffic patterns as seen before. We can choose to synchronize two patterns belonging to two different flows. As a consequence the patterns can be generated at the same time or one after the other.

Once we defined flows and patterns, it is important to define the transport protocol parameters. In UDP case, we can choose the distribution of packet inter-arrivals between the following laws: Constant, Exponential, Normal, Lognormal, Pareto, Weibull, Gamma, Inverse-Gaussian and Uniform. We can also define a mixed law between two basic laws. All laws are defined into truncated form also to be used for packet size definition. Besides basic distributions we have complex processes such as $M/G/\infty$, $MMPP-N$, FGN, ... that may be used to model aggregated traffic. We can use any of the previous laws to define any parameter of the application model. On the other hand, in TCP case, we can choose the TCP algorithm: Reno, New Reno and SACK. We can also define

general parameters of TCP: cwnd (initial and max), b, ssthresh, ...

A.3.2 Interface

On Figure **A.1** we can see the main window of Source Modeller. In this window we can define the source name, the target terminal bandwidth. This last value is used for the packet rate estimation of application models described with the tool (via the function Compute source rate). On the same window we define flows (or streams) structure of the source. Two options are available: first, stream-driven, used when the application activity is defined by its streams components, and second, session driven, used when application activity is defined by the user. In the later the stream activity will be stopped when the session end defined by the user is reached.

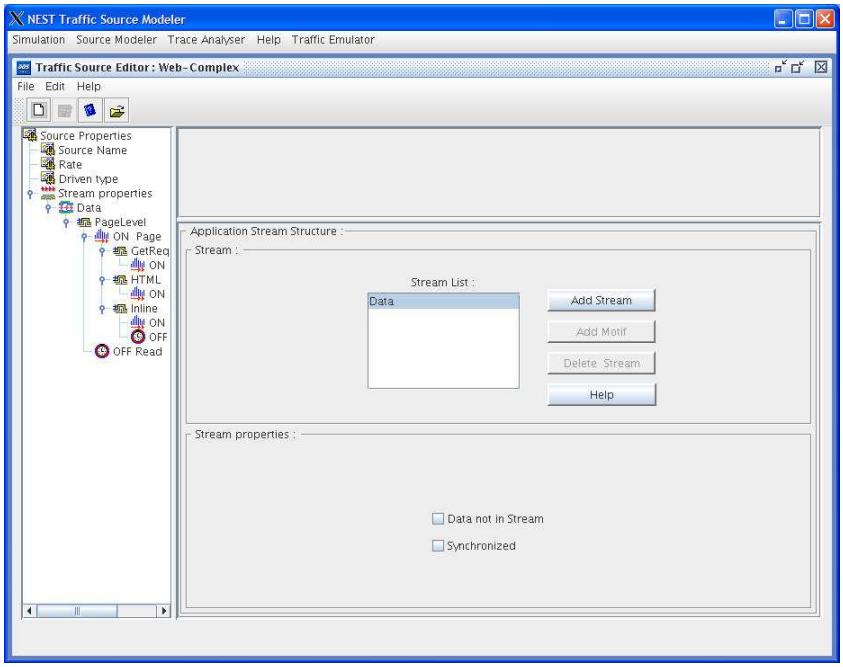


Figure A.6: Application Stream Structure

Then user can add flows (streams) to the application model according to its activity. This is done by the stream edition frame (see Figure **A.6**), in which each stream is assigned a name. Besides, user can choose to synchronize stream patterns (if it applies), and can chose whether to include or not the stream data in the application model. The last property is particularly useful to define virtual synchronization streams. Virtual synchronization streams are only used to define the functionality of the application without having packets generated really during the simulation.

Each stream is composed of patterns (Motifs) describing activity blocks. This is best illustrated by web application model in which, a pattern corresponds to the file downloading period followed by the reading time period. The repetition of the pattern across time models the page number distribution in this case (see Figure **A.7**). Patterns

are very useful to describe in details the succession of activity and idle periods in one application.

Another important feature that must be defined at the pattern level concerns the synchronization. In fact, in a synchronized stream, the synchronization is done by Patterns belonging to different streams. A synchronized pattern may be launched after or at the same time of the synchronizing pattern. Moreover, user may choose to run the synchronized pattern each time the synchronizing pattern is executed, each n times, or only n times. These different possibilities allow the user to define the synchronization activity as desired.

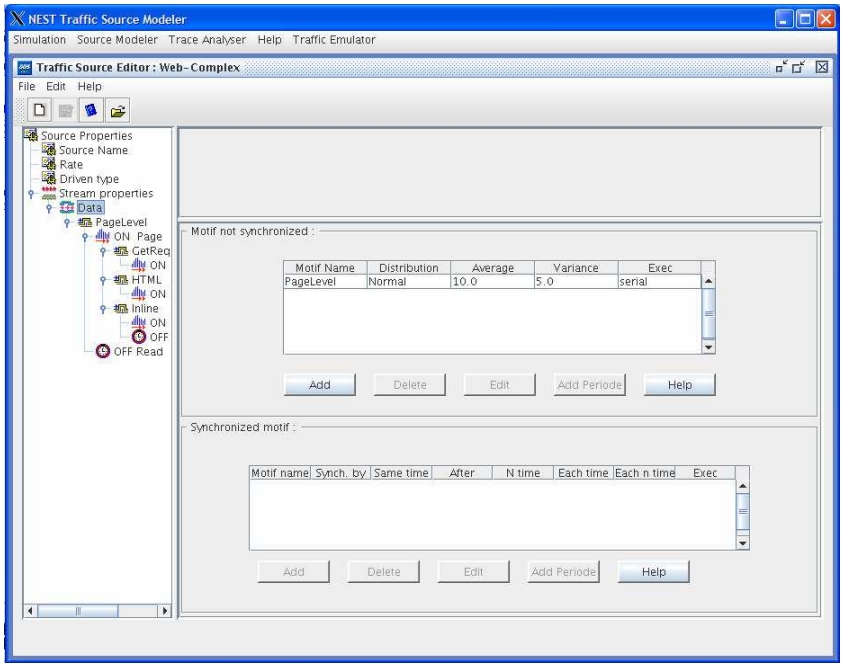


Figure A.7: Pattern Definition Window

Each pattern consists of ON and OFF periods. The succession of these periods as well as the type of these periods is defined in the Period Edition frame (see Figure **A.8**). Two types of periods may be defined (ON and OFF) while the order of definition implies the order of execution. In the case of an OFF period, user can only define the duration of this period as no packet generation activity will take place. On the other hand, ON periods are much more complicated because they are the basic brick in defining packet generation profile.

ON period can be organized in several sub-ON periods. In this case we call it a sub-Period driven. This allows us to repeat the same cycle of pattern definition described before. This feature illustrates the generic property of the framework as user can describe the activity in a nested detailed manner (Up to N theoretically). On the other hand, ON period may be simple one (Period driven). In this case user should define the ON period characteristics directly. Is it a period defined by size (e.g. a file transfer) or by duration (e.g. talkspurt)? Which size or duration distribution will be used? Using

which transport protocol? And in which direction the packets are generated (Source →Destination or Destination→Source)?

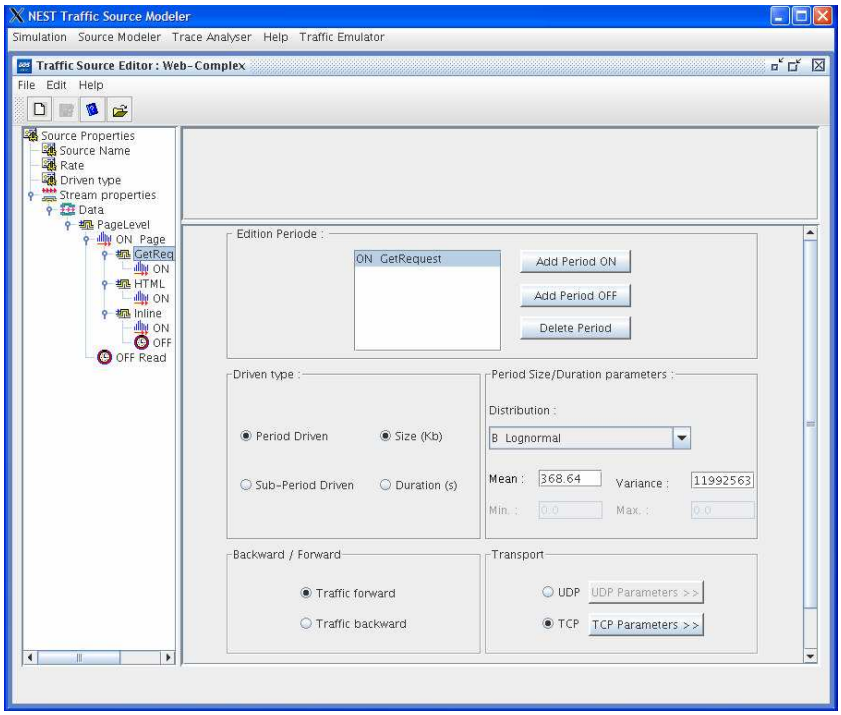


Figure A.8: Period Definition Frame

At this stage of the model description user starts to define the packet generation profile. For TCP based periods, one may select TCP algorithm parameters (see Figure **A.9**). In this case the packet inter-arrival and size distributions are completely controlled by TCP according to the choice of the algorithm parameters.

In the UDP case, one needs to specify the packet size distribution as well as packet inter-arrival distribution (see Figure **A.10**). Packet size distribution may be continuous (truncated) or discrete one. On the other hand, Packet inter-arrival distribution may be of any kind. Three cases are provided: basic distributions, mixed distributions and complex distributions. Mixed distributions are defined by the user previously according to the fitting results given by the EM algorithms (see Figure **A.11**), while complex distributions are complex processes such as M/G/∞, MMPP, FGN, ... As an example of predefined processes we show on Figure **A.12** the M/G/∞ process definition window. The tool offers the possibility to add new processes corresponding to new traffic models that may not be handled previously.

A.4 Conclusion

In this appendix we presented an overview of the TSM modelling and simulation tool. The tool is built on the generic hierarchical model presented in **Chapter 3**. Basically,

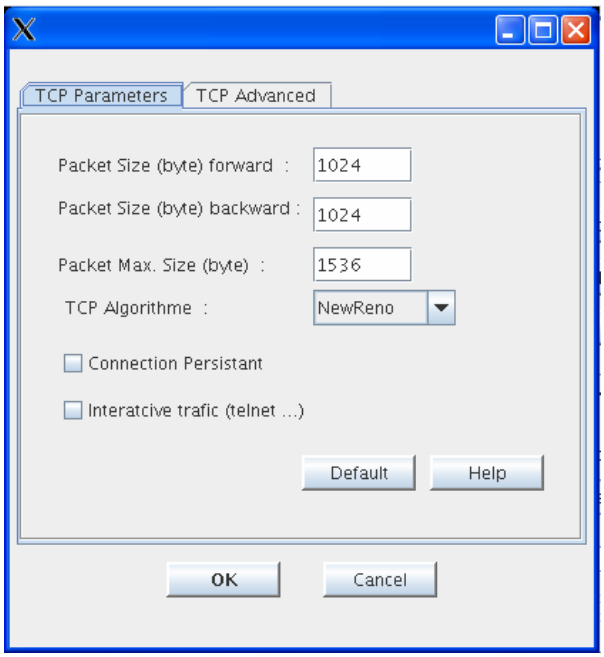


Figure A.9: TCP Parameters Window

the tool works in tandem with Distributed Hybrid Simulator (DHS) where traffic models described by the tool can be used in network simulation studies to evaluate the QoS requirements of multimedia applications. The main advantage of this implementation is that detailed multimedia applications described with the tool can be used with analytical models in DHS to perform fast Hybrid simulations.

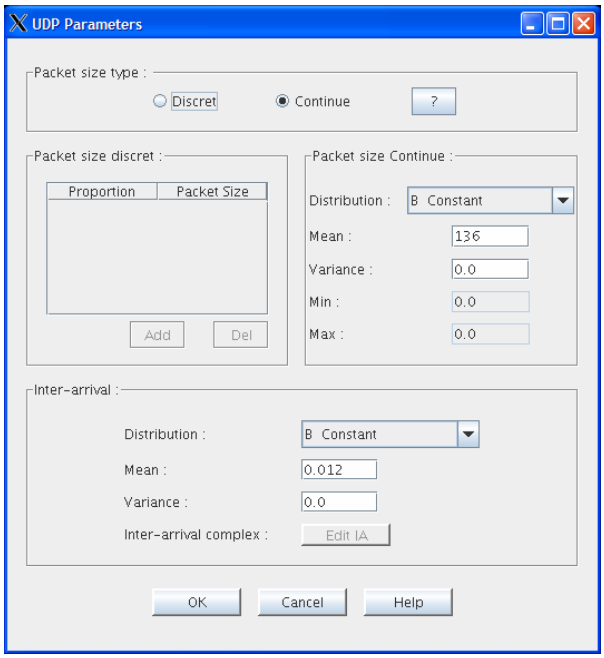


Figure A.10: UDP Packet Definition Window

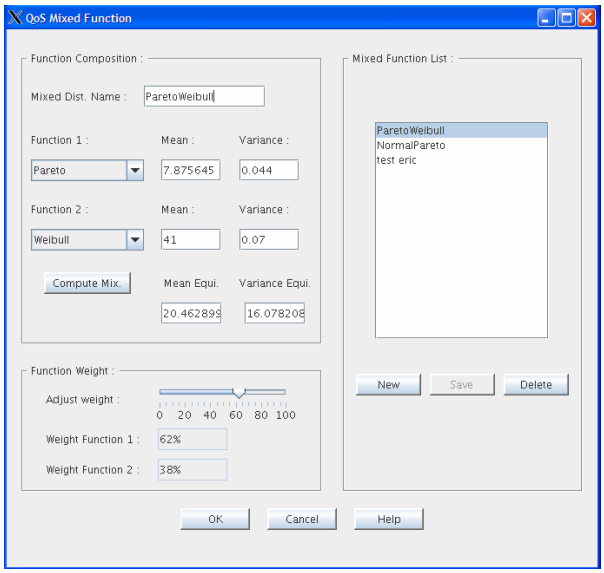


Figure A.11: Mixture Distribution Window

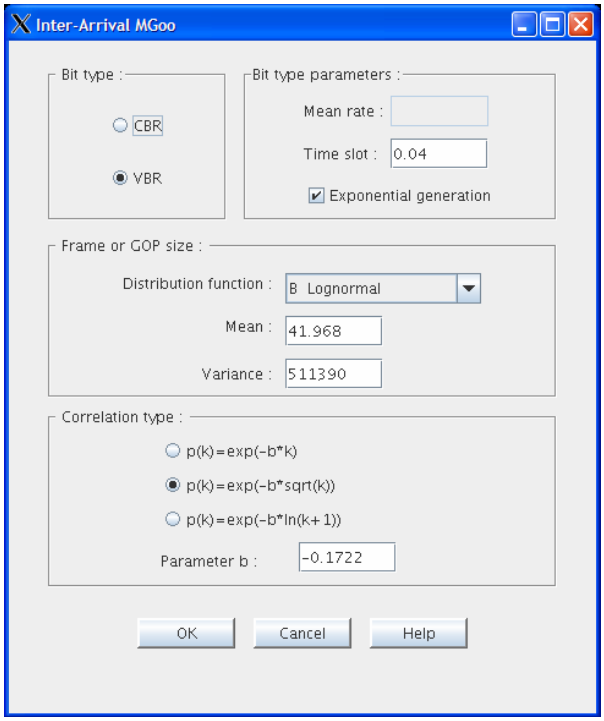


Figure A.12: M/G/∞ Definition Window

Appendix B

Video Traffic Models

B.1 Video Codecs

B.1.1 H261 Codec

H261 is an ITU video coding standard (ITU recommendation in 1990) specifically designed for transmission over ISDN lines for which data rates are multiples of 64 Kbps. The standard supports Common Intermediate Format (CIF) and Quarter Common Intermediate Format (QCIF) video frames at resolutions 352x288 and 176x144 respectively. The coding algorithm is a hybrid of inter-picture prediction, transform coding, and motion compensation. The inter-picture prediction removes temporal redundancy while transform coding removes the spatial redundancy. Motion vectors are used to help the codec compensate for motion. H261 is typically used for face-to-face videophone applications and for video conferencing.

B.1.2 H263 Codec

H263 is an ITU-T standard (ITU recommendation in 1998) designed for low bit rate communications. The coding algorithm of H263 is similar to that used by H261, however with some changes to improve performance and error recovery. H263 supports five resolutions: CIF, QCIF, SQCIF, 4CIF, and 16CIF. SQCIF is approximately half the resolution of QCIF. 4CIF and 16CIF are 4 and 16 times the resolution of CIF respectively. The support of 4CIF and 16CIF let the codec compete with other higher bit rate video coding standards such as the MPEG standards. The data rate associated with this codec is set to between 10 Kbps and 2 Mbps.

B.1.3 MJPEG Codec

Motion Joint Photographic Experts Group (MJPEG) is a video encoding scheme where each frame is separately compressed into a JPEG picture. MJPEG is best suited for broadcast resolution interlaced video (720x486 D1 NTSC or 720x576 PAL). Because it is

designed for interlaced video, MJPEG is not suitable for movies smaller than television resolution.

B.1.4 MPEG1 Codec

MPEG1 is the first of a family of motion video and audio compression standards (Standard ISO/IEC 1991). It provides DCT (Discrete Cosine Transform) lossy compression with rather high quality. It is intended for broadcast quality applications. Bit rates targeted for the MPEG1 video standard are between 1-1.5 Mbps.

B.1.5 MPEG2 Codec

MPEG2 is the second in a family of motion video and audio compression standards (Standard ISO/IEC 1994). The result of a natural evolution from MPEG1, it provides DCT (Discrete Cosine Transform) lossy compression ranging from low to rather high quality although not as good as MPEG1. It is intended for broadcast quality applications. MPEG2 has largely supplanted MPEG1 and is used for coding multimedia images from CD-ROM, DVD, broadcasting, pay TV and high quality video conferencing. Bit rates targeted for the MPEG2 video standard are between 4-10 Mbps.

B.1.6 MPEG4 Codec

In contrast to MPEG1 and MPEG2, the MPEG4 ISO standard (International standard in 1999) is object-oriented. MPEG4 objects are part of a scene, which can be manipulated independently. MPEG4 is based on the segmentation of audiovisual scenes into AVOs or "audio/visual objects" which can be multiplexed for transmission over heterogeneous networks. MPEG4 achieves higher compression ratios than MPEG2 and has better coding tools. Bit rates targeted for the MPEG4 video standard are between 5-64 Kbps for mobile or Public Switched Telephone Network (PSTN) video applications and up to 4 Mbps for TV/film applications.

B.2 Video Traffic Models Library

We estimated traffic parameters corresponding to M/G/∞ process for many video traffic traces available on the Web [TNG00]. The corresponding traffic generators are defined using the TSM tool and are available as predefined traffic generators to use in network simulation studies within TSM. The following table lists the parameters of traffic generators per video traffic trace and per codec type.

Video Source	**Codec**	**Time Slot**	**Correlation**	**Size Distribution** Byte
MJEPG (Beauty and Beast)	MJEPG VBR	Frame 1/25s	$e^{-0.03\sqrt{k}}$	**Gamma** Mean=12550 Var=1.255e+07
MPEG1 (Dino)	MPEG1 VBR	GOP 12/25s	$e^{-0.35\sqrt{k}}$	**LogNormal** Mean=141350 Var=2.1544e+09
MPEG2 (Wizard of Oz)	MPEG2 VBR	Frame 1/25s	$e^{-0.055\sqrt{k}}$	**LogNormal** Mean=21015 Var=1.1443e+08
MPEG4_Film_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.154\sqrt{k}}$	**Gamma+LogNormal** Mean=9265 Var=2.9235e+07
MPEG4_Film_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.1597\sqrt{k}}$	**Gamma+LogNormal** Mean=14860 Var=9.151e+07
MPEG4_Film_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.1722\sqrt{k}}$	**Gamma+LogNormal** Mean=41968 Var=5.1139e+08
MPEG4_Cartoon_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.3733\sqrt{k}}$	**Gamma+LogNormal** Mean=13826 Var=3.3858e+07
MPEG4_Cartoon_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.341\sqrt{k}}$	**Gamma+LogNormal** Mean=25385 Var=9.5858e+07
MPEG4_Cartoon_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.3095\sqrt{k}}$	**Gamma+LogNormal** Mean=78154 Var=4.2495e+08
MPEG4_Sport_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.106k}$	**Gamma+LogNormal** Mean=11258 Var=2.8204e+07
MPEG4_Sport_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.108k}$	**Gamma+LogNormal** Mean=18835 Var=9.179e+07
MPEG4_Sport_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.104k}$	**Gamma+LogNormal** Mean=49958 Var=5.7125e+08
MPEG4_ParkingCam_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.0038k}$	**Normal+Normal** Mean=7983 Var=9.2239e+05

MPEG4_ParkingCam_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.0032k}$	**Normal+Normal** Mean=14186 Var=4.8892e+06
MPEG4_ParkingCam_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.003k}$	**Normal+Normal** Mean=47295 Var=3.7749e+07
MPEG4_News_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.0778k}$	**Gamma+LogNormal** Mean=43674 Var=5.5207e+08
MPEG4_Talk_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.024k}$	**Gamma+LogNormal** Mean=32792 Var=1.053e+08
MPEG4_OfficeCam_L (L: Low Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.2537k}$	**Normal+Normal** Mean=5406.5 Var=2.1443e+05
MPEG4_OfficeCam_M (M: Medium Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.2176k}$	**Normal+Normal** Mean=6658.8 Var=7.159e+05
MPEG4_OfficeCam_H (H: High Quality)	MPEG 4 VBR	GOP 12/25s	$e^{-0.2256k}$	**Normal+Normal** Mean=23738 Var=5.0029e+06
H263_Film_16 (16kbps)	H263 CBR	Frame 1/25s	$(k+1)^{-0.9196}$	**Gamma+LogNormal** Mean=428.0631 Var=6.0393e+04
H263_Film_64 (64kbps)	H263 CBR	Frame 1/25s	$(k+1)^{-0.519}$	**Normal+Normal** Mean=1124.4 Var=1.5475e+05
H263_Film_256 (256kbps)	H263 CBR	Frame 1/25s	$(k+1)^{-0.6343}$	**Normal+Normal** Mean=4536.9 Var=2.4194e+06
H263_Cartoon_16 (16kbps)	H263 CBR	Frame 1/25s	$e^{-1.2036k}$	**Gamma+LogNormal** Mean=560.0774 Var=1.9425e+05
H263_Cartoon_64 (64kbps)	H263 CBR	Frame 1/25s	$e^{-7.535k}$	**Normal+Normal** Mean=1081.1 Var=1.5879e+05
H263_Cartoon_256 (256kbps)	H263 CBR	Frame 1/25s	$e^{-1.1288k}$	**Normal+Normal** Mean=4246.3 Var=2.3814e+06

H263_News_16 (16kbps)	H263 CBR	Frame 1/25s	$0.379e^{-0.3794k} + 0.621(k+1)^{-3.03}$	**Gamma+LogNormal** Mean=437.5441 Var=7.212e+04
H263_News_64 (64kbps)	H263 CBR	Frame 1/25s	$e^{-0.2302\sqrt{k}}$	**Normal+Normal** Mean=1228.5 Var=1.8851e+05
H263_News_256 (256kbps)	H263 CBR	Frame 1/25s	$0.535e^{-0.0205k} + 0.465(k+1)^{-3}$	**Normal+Normal** Mean=4875.1 Var=2.903e+06
H263_ParkingCam_16 (16kbps)	H263 CBR	Frame 1/25s	$e^{-13.52k}$	**Normal+Normal** Mean=414.2058 Var=2.0203e+03
H263_ParkingCam_64 (64kbps)	H263 CBR	Frame 1/25s	$e^{-15.505\sqrt{k}}$	**Normal+Normal** Mean=1585.4 Var=7.415e+03
H263_ParkingCam_256 (256kbps)	H263 CBR	Frame 1/25s	$e^{-12.4736\sqrt{k}}$	**Normal+Normal** Mean=6318.1 Var=3.7874e+04
H263_OfficeCam_16 (16kbps)	H263 CBR	Frame 1/25s	e^{-15k}	**Gamma+LogNormal** Mean=444.9297 Var=1.0494e+05
H263_OfficeCam_64 (64kbps)	H263 CBR	Frame 1/25s	$0.896e^{-1.092\sqrt{k}} + 0.104(k+1)^{-0.221}$	**Normal+Normal** Mean=1565.7 Var=4.1747e+04
H263_OfficeCam_256 (256kbps)	H263 CBR	Frame 1/25s	$(k+1)^{-1.0095k}$	**Gamma+LogNormal** Mean=6192.6 Var=4.2687e+05
H263_Sport_16 (16kbps)	H263 CBR	Frame 1/25s	$e^{-0.3568\sqrt{k}}$	**Gamma+LogNormal** Mean=653.3933 Var=1.5125e+05
H263_Sport_64 (64kbps)	H263 CBR	Frame 1/25s	$e^{-0.4663\sqrt{k}}$	**Normal+Normal** Mean=902.674 Var=1.0692e+05
H263_Sport_256 (256kbps)	H263 CBR	Frame 1/25s	$e^{-0.4476\sqrt{k}}$	**Normal+Normal** Mean=3392.3 Var=1.5872e+06

Bibliography

[AAB00] Eitan Altman, Konstantin Avrachenkov, and Chadi Barakat. A stochastic model of TCP/IP with stationary random. In *SIGCOMM*, pages 231–242, 2000.

[AM95] A. Adas and A. Mukherjee. On resource management and qos guarantees for long range dependent traffic. In *INFOCOM 1995*, volume 2, pages 779–787, 2-6 April 1995.

[Aye05] U. Ayesta. *Stochastic Scheduling and its Application to TCP/IP Networks*. PhD thesis, INRIA, 2005.

[BA00] Chadi Barakat and Eitan Altman. A markovian model for TCP analysis in a differentiated services network. In *QofIS*, pages 55–67, 2000.

[BBC+98] S. Blake, D. Black, M. Carlso, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. *IETF, RFC2475*, 1998.

[BEF+02] P. Belloni, T. Eskedal, M. Ficaccio, C. Hatch, D. Milham, S. Hughes, D. Mccarthy, O. Osterbo, and G. Vallazza. Umts quality of service survey, January 2002.

[Ber94] J. Beran. *Statistics for long-memory processes*. Chapman & Hall, New York, 1994.

[BLFF96] T. Berners-Lee, R. Fielding, and H. Frustuck. Hypertext transfer protocol: Http 1.0. *IETF RFC1945*, 1996.

[Bra68] P.T. Brady. A model for generating on-off speech patterns in two-way conversations. *Bell System Technology Journal*, 48, 1968.

[Bra89] R. Braden. Requirements for internet hosts - communication layers. *IETF, RFC1122*, 1989.

[BVE99] Christian Bettstetter, Hans-Jörg Vögel, and Jörg Eberspächer. Gsm phase 2+ general packet radio service gprs: Architecture, protocols, and air interface. *IEEE Communication Surveys*, 2(3), 1999.

[BW97] G. Brasche and B. Walke. Concepts services and protocols of the new gsm phase 2+ general racket radio service. *IEEE Communication Magazine*, 1997.

[CAI03] CAIDA. Oc48 traffic traces. http://www.caida.org/data/passive/index.xml, 2003.

[CL66] D. R. Cox and P. A. W. Lewis. *Statistical Analysis of Series of Events*. Monographs on Applied Probability & Statistics. Methuen young books, 1966.

[CL99] H.K. Choi and J. Limb. Behavioural of web traffic. In *International Conference of Networking Protocol*, 1999.

[Cro82] David H. Crocker. Standard for arpa internet text messages. *IETF, RFC822*, 1982.

[CW95] C. Courcoubetis and R. Weber. Effective bandwidths for stationary sources. *Probability in Engineering and Informational Sciences*, 9(2):285–294, 1995.

[CW96] C. Courcoubetis and R. Weber. Buffer overflow asymptotics for a switch handling many traffic sources. *Journal of Applied Probability*, 33:886–903, 1996.

[CWKS97] B.P. Crow, I. Widjaja, L.G. Kim, and P.T. Sakai. Ieee 802.11 wireless local area networks. *Communications Magazine, IEEE*, 35(9):116–126, Sept. 1997.

[Den95] Shuang Deng. Traffic characteristics of packet voice. In *ICC 1995*, volume 3, pages 1369–1374, 18-22 June 1995.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Royal Statistics Society*, 39(1):1–38, 1977.

[dVW95] G. de Veciana and J. Walrand. Effective bandwidths: Call admission, traffic policing and filtering in atm networks. *Queueing Systems*, 20:37–59, 1995.

[EM78] P. E.Gill and Murray. Algorithms for the solution of the nonlinear least-squares problem. *SIAM Journal on Numerical Analysis*, 15(5):977–992, 1978.

[EM93] Anwar I. Elwalid and Debasis Mitra. Effective bandwidth of general markovian traffic sources and admission control of high speed networks. *IEEE/ACM Trans. Netw.*, 1(3):329–343, 1993.

[ESP94] A. Erramilli, R. P. Singh, and P. Pruthi. Chaotic maps as models of packet traffic. In *ITC 1994*, pages 329–338, 1994.

[EV03] C. Estan and G. Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM transactions on Computer Systems*, 2003.

[FGM+97] R. Fielding, J. Getty, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext transfer protocol: Http 1.1. *IETF RFC2068*, 1997.

[Fla92] P. Flandrin. Wavelet analysis and synthesis of fractional brownian motion. *IEEE Transactions on Information Theory*, 38(2):910–917, March 1992.

[GAN91] R. Geurin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, 1991.

[GGB+01] JM. Garcia, D. Gauchard, O. Brun, P. Bacquet, J. Sexton, and E. Lawless. Modélisation différentielle du trafic et simulation hybride distribuée. *Réseaux et Systèmes Répartis. Calculateurs Parallèles. Performances des réseaux et systèmes*, 13(6):635–664, 2001.

[Gil05] J. Gil. Modelling tcp/ip with a discrete time markov chain. In *Het-Net 2005*, 2005.

[GSM95] Gsm 03.02, network architecture, 1995.

[GSM98] Gsm 02.60, gprs, service description, stage 1, 1998.

[GW94] Mark Garrett and Walter Willinger. Analysis, modeling and generation of self-similar vbr video traffic. In *SIGCOMM 1994*, pages 269–280, 1994.

[HG05] H. Hassan and J. Garcia. Fractal analysis of tcp flows. In *4th Workshop on the Internet, Telecommunications and Signal Processing (WITSP 2005), Noosa Heads (Australia), 19-21 December*, page 5 pages, 2005.

[HG06] H. Hassan and J. Garcia. Aggregate modelling for tcp sessions. In *2nd ACM Workshop on Wireless Multimedia Networking and Performance Modelling (WMuNeP 2006), Torremolinos, Málaga, Spain , 2-6 October*, page 6 pages, 2006.

[HGB06a] H. Hassan, J. Garcia, and C. Bockstal. Aggregated traffic models for voip applications. In *International Conference on Digital Telecommunications (ICDT 2006), Cap Esterel (France), 29-31 August*, page 6 pages, 2006.

[HGB06b] H. Hassan, J. Garcia, and C. Bockstal. Differential modeling and its application to tcp/ip. In *European Simulation and Modelling Conference (ESM 2006), Toulouse, France, 23-25 October*, page 6 pages, 2006.

[HGB06c] H. Hassan, J. Garcia, and C. Bockstal. Modeling internet traffic: performance limits. In *International Conference on Internet Surveillance and Protection (ICISP 2006), Cap Esterel (France), 27-29 August*, page 6 pages, 2006.

[HGB06d] H. Hassan, J. Garcia, and O. Brun. Internet traffic modelling. In *2nd IEEE International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA 2006), Damas (Syria), 24-28 April*, page 6 pages, 2006.

[HGB06e] H. Hassan, J. Garcia, and O. Brun. Session scheduling in sip based network. In *IEEE 3rd Consumer Communications and Networking Conference(CCNC 2006), Las Vegas (US), 8-10 January*, volume 2, pages 1263– 1267, 2006.

[HGB06f] Hassan HASSAN, Jean-Marie GARCIA, and Olivier BRUN. Bandwidth allocation and session scheduling using sip. *Journal of Communications*, 1(5):17–27, August 2006.

[HGCB06] H. Hassan, J. Garcia, F. Camps, and C. Bockstal. A generic framework for modelling multimedia applications. In *13th International Conference on Telecommunications 2006 (ICT 2006), Funchal, Portugal, 9-12 May*, page 4 pages, 2006.

[HL86] H. Heffes and D. Lucantoni. A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on Selected Areas in Communications*, 4(6):856–868, Sep 1986.

[Hur51] H. E. Hurst. Long-term storage capacity of reservoirs. *Transactions of American Socety of Civil Engineering*, 116:770–799, 1951.

[Çin75] E. Çinlar. *Introduction to Stochastic Process*. Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1975.

[Jac88] Van. Jacobson. Congestion avoidance and control. In *SIGCOMM 1988: Symposium proceedings on Communications architectures and protocols*, pages 314–329, New York, NY, USA, 1988. ACM Press.

[KC03] D. Kleban and H. Clearwater. Quelling queue storms. In *HPDC 2003*, 2003.

[Kel91] F. P. Kelly. Effective bandwidths at multi-class queues. *Queuing Systems*, 9:5–16, 1991.

[Kel96] F. P. Kelly. *Stochastic Networks: Theory and Applications*, chapter Notes on effective bandwidths, pages 141–168. Oxford University Press, 1996.

[Kle75] L. Kleinrock. *Queueing Systems, Volume 1: Theory*. John Wiley & Sons, 1975.

[KLL01] A. Klemm, C. Linderman, and M. Lohmann. Traffic modelling and characterization for umts networks. In *IEEE Globecom 2001*, 2001.

[KM98] M.M. Krunz and A.M. Makowski. Modeling video traffic using m/g/oo input processes: a compromise between markovian and lrd models. *IEEE Journal on Selected Areas in Communications*, 16(5):733–748, June 1998.

[Lev44] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly Journal of Applied Mathematics*, 2:164–168, 1944.

[LEWW95] W-C. Lau, A. Erramilli, J. Wang, and W. Willinger. Self-similar traffic generation: The random midpoint displacement algorithm and its properties. In *ICC 1995*, 1995.

[LGK03] R. Ludwig, A. Gurtov, and F. Khafizov. Tcp over second (2.5g) and third (3g) generation wireless networks. *IETF, RFC3481*, 2003.

[LNSH98] T. Le-Ngoc and S. Shah-Heydari. Mmpp modelling for aggregated atm traffic. In *IEEE Canadian Conference on Electrical and Computer Engineering*, pages 129–132, 1998.

[LTWD94] W. Leland, M. Taqqu, W.Willinger, and D.Wilson. On the self-similar nature of ethernet traffic. *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.

[Man04] E. Mannie. Generalized multi-protocol label switching (gmpls) architecture. *IETF, RFC3945*, 2004.

[Mar63] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441, 1963.

[MAS+88] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J.D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Transactions on Communications*, 36(7):834–844, July 1988.

[MAW06] MAWI. Internet traffic traces. http://tracer.csl.sony.co.jp/mawi/, 2006.

[MGT99] V. Misra, W. Gong, and D. Towsley. Stochastic differential equation modeling and analysis of tcp-windowsize behavior. In *PERFORMANCE99, Istanbul, Turkey, 1999.*, 1999.

[MH00] A. Mena and J. Heidemann. An empirical study of real audio traffic. In *INFOCOM 2000*, volume 1, pages 101–110, 26-30 March 2000.

[MP04] E. Mannie and D. Papadimitriou. Generalized multi-protocol label switching (gmpls) extensions for synchronous optical network (sonet) and synchronous digital hierarchy (sdh) control. *IETF, RFC3946*, 2004.

[MS92] B. Melamed and B. Sengupta. Tes modelling of video traffic. *IEICE Trans on Communications*, E75-B(12):1292–1300, 1992.

[MW69] B. B. Mandelbrot and J. R. Wallis. Robustness of the rescaled range r/s in the measurement of non-cyclic long run statistical dependence. *Water Resources Research*, 5:967–988, 1969.

[NKT91] R. Nagarajan, J. F. Kurose, and D. Towsley. Approximation techniques for computing packet loss in finite-buffered voice multiplexers. *IEEE Journal on Selected Areas in Communications*, 1991.

[NS201] The network simulator ns-2. http://www.isi.edu/nsnam/ns/, October 2001.

[Pax97] V. Paxson. Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic. *Computer Communication Review*, 27:5–18, 1997.

[PF95] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modelling. *IEEE/ACM Transactions on Networking*, 3:226–244, June 1995.

[PGP04] C. Psounis, A. Ghosh, and B. Prabhakar. Sift: a simple algorithm for identifying large flows. In *Stochastic Networks Conference*, 2004.

[PKC96] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *BU-CS-96-016*, 1996.

[PMKN04] M. Poikselka, G. Mayer, H. Khartabil, and A. Niemi. *The IMS IP Multimedia Concepts and Services in the Mobile Domain*. WILEY, 2004.

[Pos81] J. Postel. Transmission control protocol. *IETF, RFC793*, September 1981.

[RLB+05] B. Rong, J. Lebeau, M. Bennani, M. Kadoch, and A. Elhakeem. Traffic aggregation based sip over mpls network architecture. In *AINA 2005*, 2005.

[Ros95] O. Rose. *Statistical Properties of MPEG Video traffic and their Impact on Traffic Modelling in ATM Systems*. PhD thesis, University of Wurzburg, 1995.

[RSC+02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol. *IETF, RFC3261*, 2002.

[RVC01] E. C. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. *IETF, RFC3031*, 2001.

[SFB01] P. Stuckmann, H. Finck, and T. Bahls. A wap traffic and its appliance for the performance analysis of wap overs gprs. In *3Gwireless 2001*, 2001.

[SG95] A. Simonian and J. Guibert. Large deviations approximations for fluid queues fed by a large number of on-off sources. *IEEE Journal on Selected Areas in Communications*, 13(7):1017–1027, 1995.

[SKV01] B. Sikdar, S. Kalyanaramana, and K.S Vastola. An integrated model for the latency and steady-state throughput of tcp connections. *Performance Evaluation*, 46(2-3):139–154, 2001.

[SLTG00] D. Staehle, K. Leibnitz, and P. Tran-Gian. Source traffic wireless applications. Technical report, University of Würzburg, 2000.

[SSFJ98] H. Schulzrinne, S.Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications. *IETF, RFC1889*, 1998.

[Ste97] W. Stevens. Tcp/ip slow-start, congestion avoidance, fast retransmit and fast recovery algorithms. *IETF, RFC2001*, 1997.

[STP+04] S. Stoev, M. Taqqu, C. Park, G. Michailidis, and J. S. Marron. Lass: a tool for the local analysis of self-similarity in internet traffic. *Computational Statistics and Data Analysis*, 2004.

[SV02] S. Salsano and S. Veltri. Qos control by means of cops to support sip-based applications. *IEEE Network*, 16(2):27–33, 2002.

[SW86] K. Sriram and W. Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, 4(6):833–846, Sep 1986.

[TDL98] T. Taqqu, M. Devetsikiotis, and I. Lambadaris. Efficient fractional gaussian noise generation using the spatial renewal process. In *ICC 1998*, pages 1455–1460, 1998.

[Thi02] E. Thibault. Analyses de traces de trafic réel. Technical report, INRIA, Nov 2002.

[TNG00] Germany Telecommunication Networks Group, Technical University of Berlin. Video traffic traces. http://www.tkn.tu-berlin.de/research/trace/trace.html, 2000.

[Tou98] F. Toutain. Ordonnancement de paquets équitable par les disciplines gps et dgps. *Annales des Télécommunications*, 53(9-10), 1998.

[TWS97] S. Taqqu, W. Willinger, and R. Sherman. Proof of a fundamental result in self-similar traffic modelling. *ACM SIGCOMM Computer Communication Review*, 27(2):5–23, Sep 1997.

[Vil00] C. Jalpa Villanueva. *Modélisation et Optimisation du Web*. PhD thesis, INRIA Sophia Antipolis, 2000.

[Whi83] W. Whitt. The queuing network analyzer. *Bell System Technical Journal*, pages 2779–2813, 1983.

[Whi93] W. Whitt. Tail probabilities with statistical multiplexing and effective bandwidths in multi-class queues. *Telecommunication Systems 2*, pages 71–107, 1993.

[WHM01] Gang Wu, P.J.M. Havinga, and M. Mizuno. Wireless internet over heterogeneous wireless networks. In *Global Telecommunications Conference, 2001. GLOBECOM 2001. IEEE*, volume 3, pages 1759–1765, 25-29 Nov. 2001.

[WPT98] W. Willinger, V. Paxson, and S. Taqqu. *A Practical Guide To Heavy Tails: Statistical Techniques and Applications*, chapter Self-Similarity And Heavy Tails: Structural Modelling of Network Traffic, pages 27–53. Birkhauser, Boston, 1998.

[ZG03] C. Zhang and C.G Guy. Te-sip server design for a sip-over-mpls based network. In *ICCT 2003*, volume 2, pages 1758–1761, 2003.

List of Figures

List of Tables

Résumé:

Ces travaux concernent la modélisation et l'analyse de performances du trafic et des applications multimédia dans les réseaux hétérogènes. Le trafic IP agrégé et les applications audio, vidéo et données sont étudiés. Cette étude nous conduit à proposer un modèle générique et hiérarchique pour la représentation des sources de trafics multimédia qui permet de décrire les applications multimédia d'une façon simple, précise et générique. Le modèle générique est implémenté et constitue le noyau d'un outil de modélisation et simulation des sources de trafics. Une caractérisation du trafic IP issu d'applications multimédia est conduite en utilisant les modèles développés avec cet outil. Particulièrement, la problématique de la modélisation des sources de trafics agrégées est adressée, et des modèles agrégés simples sont déduites pour la superposition des sources de trafics audio, vidéo et données. Le trafic agrégé de type TCP présente des propriétés statistiques variables en fonction du taux de pertes de paquets sur le réseau à cause du contrôle en boucle fermée imposé par TCP. Un nouveau modèle analytique du protocole TCP basé sur la théorie différentielle du trafic est ensuite proposé. Ce modèle permet une représentation fiable du trafic TCP tout en étant très performant sur les réseaux à grande échelle. Finalement, une extension de l'architecture du protocole SIP est présentée afin de permettre une gestion de la qualité d service au niveau session. Les mécanismes proposés reposent sur l'ordonnancement stochastique des sessions et l'allocation de la bande passante par des approches d'évaluation de bande passante équivalente. Cette dernière technique rend possible l'utilisation des formules d'Erlang dans les réseaux à commutation de paquets.

Mots clés : *Modélisation, Analyse de performances, Trafic, Multimédia, Audio, Vidéo, Données, TCP/IP, SIP, Bande passante équivalente.*

Abstract:

This thesis contains research results concerning traffic and multimedia application modelling and performance analysis in heterogeneous networks. Aggregate IP traffic as well as audio, video and data applications are studied, and a generic hierarchical model to describe multimedia traffic sources is derived. This model helps representing multimedia applications in simple, precise and generic way. The generic model is implemented and constitutes the core of a powerful modelling and simulation tool for traffic sources. We characterize IP traffic generated by multimedia applications using models developed with this tool. In particular, we handle the issue of modelling aggregated traffic sources, and provide simple aggregated models used to replace the superposition of audio, video and data traffic sources. Aggregate TCP traffic presents dynamic statistical properties in function of packet loss rate observed in the network. This is because of the closed-loop control used by TCP. Thus, a new analytical model of TCP based on differential traffic theory is proposed. This model offers a reliable representation of TCP traffic while achieving good performances in large scale networks. Finally, we extend the architecture of SIP protocol in order to provide an application level management of Quality of Service. The suggested mechanisms rely on stochastic scheduling of sessions and bandwidth reservation using equivalent bandwidth evaluation techniques. This last mechanism makes it possible to use Erlang formulas in packet switching networks.

Keywords*: Modelling, Performance Analysis, Traffic, Multimedia, Audio, Video, Data, TCP/IP, SIP, Equivalent Bandwidth.*