Projet Metropolis

Metrologie POur l'Internet et les Services

Sous-Projet 1 : Rapport d'état de l'art

Table des matières

Chapitre 1

Introduction

Depuis quelques années, l'Internet est devenu l'un des enjeux majeurs pour les télécommunications. La décentralisation de l'administration, l'émergence de communautés d'utilisateurs avec des exigences différentes en termes de qualité, de sécurité, de fiabilité, ... et l'introduction de nouveaux services exigent à présent le recueil et l'interprétation d'informations sur le comportement et les performances du réseau.

La métrologie est une activité cruciale pour les opérateurs de réseaux afin de concevoir une ingénierie efficace pour leurs offres de services (incluant des aspects de qualité). Pour la recherche plus amont, la métrologie est un des points de passage obligés pour valider des modèles et en concevoir d'autres qui reflètent le plus fidèlement possible réalité. Les mesures effectuées dans un réseau sont ainsi des données très précieuses pour les études en milieu aussi bien industriel qu'universitaire.

Vu l'importance de la métrologie des réseaux IP, beaucoup de projets de recherche sur ce sujet sont en cours, en particulier conduits par des opérateurs Internet et des laboratoires d'étude et de recherche Internet aux Etats-Unis. Ceci c'est concrétisé durant les quelques années passées par l'émergence d'une littérature riche qui a été publiée aussi bien dans les conférences généralistes du domaine (SIGCOM, SIGMETRICS, Infocom, ITC, ICNP, *etc*) que dans des conférences spécifiques à ce domaine (IMW, PAM, *etc*).

Ce rapport n'a pas pour prétention de brosser un état de l'art exhaustif de ce domaine qui est caractérisé par une croissance très rapide du nombre de publications qui en relève. Notre objectif plus modeste est d'en faire un outil de travail et de référence pour la suite du projet.

structure du document

Ce rapport est constitué de 5 chapitres qui sont le fruit d'une collaboration entre les chercheurs des différents organismes qui constituent le projet Métropolis.

Les projets de mesure actuel peuvent être répartis en deux grandes classes : ceux fondés sur les mesures actives et ceux reposant sur les mesures passives décomposées elles-mêmes en mesures passives en ligne et hors ligne. Chacune de ces deux classes permet de mieux comprendre le compor-

tement à la fois du réseau (observation passive des taux de perte, des délais, etc.) et des applications (réaction en temps réel des applications aux pertes dans le réseau, du taux de transmission utile, etc.) et de mettre en lumière les interactions entre les applications et le réseau. Ces architecture de mesure pour ces deux classes sont décrites dans le premier chapitre.

Le second chapitre traite du vaste sujet de la caractérisation des flots dans un Internet. Ce chapitre qui est le résultat d'une large collaboration brosse un vaste état des lieux au sujet des classification des flots, qu'ils soient normaux ou pathologiques (comme les attaques). Ce chapitre fait aussi le point sur l'inférence des matrices de trafic qui sont les informations nécessaires à l'ingénierie de trafic.

Le troisiéme chapitre fait un état de l'art dans la compréhension de la dynamique complexe de TCP. Cette problématique a connu un intérêt particulier durant les quelques années aussi bien au niveau pratique, TCP étant à plus de 80% le protocole principalement utilisé dans l'Internet, qu'au niveau théorique et mathématique. La compréhension de la dynamique d'aggrégation de TCP est l'un des verroux principaux qui devrait permettre une meilleure compréhension de l'Internet.

Le quatrième chapitre traite de la modélisation empirique.

Le cinquième chapitre présente des outils statistique pour l'échantillonnage.

Chapitre 2

Infrastructures de métrologie

Philippe Owezarski (LAAS-CNRS), Francois-Xavier Andreu (GIP Renater),
Kavé Salamatian (LIP6), Christophe Chekroun (LIP6)

Vue l'importance de la métrologie des réseaux IP, beaucoup de projets de recherche sur ce sujet sont en cours, en particulier conduits par des opérateurs Internet et des laboratoires d'étude et de recherche Internet aux Etats-Unis. Ces projets peuvent être répartis en deux grandes classes : ceux fondés sur les mesures actives et ceux reposant sur les mesures passives décomposées elles-mêmes en mesures passives en ligne et hors ligne. Chacune de ces deux classes permet de mieux comprendre le comportement à la fois du réseau (observation passive des taux de perte, des délais, etc.) et des applications (réaction en temps réel des applications aux pertes dans le réseau, du taux de transmission utile, etc.) et de mettre en lumière les interactions entre les applications et le réseau. Les projets de métrologie ont pour l'instant traités en priorité les thèmes suivants :

- La mesure de la QoS (Délais, pertes, RTT, débits, ...) ;
- Inférer la structure / topologie interne des réseaux ;
- La tomographie du trafic ;
- Détecter les points de congestion ;
- Les matrices de trafic / tomographie et les liens avec les protocoles et tables de routage ;
- La sécurité des réseaux ;
- Le dimensionnement des réseaux ;
- L'évolution du trafic, notamment due à l'émergence de nouvelles applications (jeux, échanges point à point, e-commerce, ...) ;
- Représenter l'Internet ;
- Modélisation du trafic et prédiction d'évolution de ces modèles.

2.1 Métrologie active

Le principe des mesures actives consiste à générer du trafic dans le réseau à étudier et à observer les effets des composants et protocoles – réseaux et transport – sur le trafic : taux de perte, délai, RTT, etc. Cette première approche possède l'avantage de prendre un positionnement orienté utilisateur. Les mesures actives restent le seul moyen pour un utilisateur de mesurer les paramètres du service dont il pourra bénéficier. En revanche, l'inconvénient majeur de cette approche est la perturbation introduite par le trafic de mesure qui peut faire évoluer l'état du réseau et ainsi fausser la mesure. De nombreux travaux menés actuellement abordent ce problème en essayant de trouver les profils de trafics de mesures qui minimisent les effets du trafic supplémentaire sur l'état du réseau. C'est par exemple le travail en cours au sein du groupe IPPM[1] de l'IETF[2] [1] [2] [3] [4]. Les mesures actives simples restent tout de même monnaie courante dans l'Internet pour lequel de nombreux outils de test, validation et / ou mesure sont disponibles. Parmi eux, on peut citer les très célèbres *ping* et *traceroute*. *Ping* permet de vérifier qu'un chemin est valide entre deux stations et de mesurer certains paramètres comme le RTT[3] ou le taux de perte. *Traceroute* permet de voir apparaître l'ensemble des routeurs traversés par les paquets émis jusqu'à leur destination et donne une indication sur les temps de passage en chacun de ces nœuds.

L'un des projets les plus simples en théorie était le projet *Surveyor* [5] de la NSF[4] aux Etats-Unis qui reposait sur l'utilisation de *ping*, amélioré par la présence d'horloges GPS[5] sur les machines de mesure. L'objectif de ce projet était donc d'étudier les délais de bout en bout et les pertes dans l'Internet.

Plusieurs projets ont actuellement pour sujet les mesures actives.

- Le projet NIMI[6] (initié par Vern Paxson aux Etats-Unis) [6] a pour objectif le déploiement d'une infrastructure nationale (au niveau des Etats-Unis) de mesures actives. Cette infrastructure est flexible et permet le recueil de diverses mesures actives. Cette infrastructure a été utilisée durant les deux ou trois années passées pour plusieurs campagnes de mesures, dont la détermination d'une matrice de distance dans Internet. L'infrastructure NIMI s'est aussi étendue en Europe, notamment en Suisse.
- En Europe, le projet RIPE (Réseaux IP Européens) [7], tente de déployer une infrastructure semblable à l'infrastructure de NIMI en Europe. Par rapport à NIMI, RIPE fournit des services à des clients : RIPE se propose de réaliser toutes les études qui peuvent être demandées par des clients, en plus des services classiques d'accès à des statistiques globales d'utilisation des liens du réseau Européen de la recherche surveillés.
- Le projet MINC[7] [8] [9] est un client du projet NIMI. Il utilise la diffusion de sondes actives par le biais du multicast pour inférer sur la structure interne du réseau et les propriétés sur tous les liens d'interconnexion ainsi traversés. En allant plus loin, c'est le sujet de la tomographie qui est au centre de ce projet qui se focalise sur certains aspects dynamiques du trafic, comme les propriétés du routage, les pertes et les délais. Toutefois, comme le multicast n'est pas un service disponible partout, et comme il a été montré que le trafic dans l'Internet n'est pas forcément symétrique, l'intérêt du multicast dans cette tâche n'est concrètement pas évident.

[1] IPPM : IP Performance Metrics
[2] IETF : Internet Engineering Task Force
[3] RTT : Round Trip Time
[4] NSF : National Science Fondation
[5] GPS : General Positioning System
[6] NIMI : National Internet Measurement Infrastructure
[7] MINC : Multicast-based Inference of Network-internal Characteristics

Aussi, le projet UINC (Unicast INC) a vu le jour et tente de reproduire le travail de MINC en unicast.

– Le projet Netsizer [10] de Telcordia (ex Bellcore) a pour objectifs de mesurer la croissance de l'Internet, les points durs de congestion, les délais, etc. Pour cela, depuis une ensemble de stations situées chez Telcordia, un programme teste la présence sur le réseau de toutes les adresses IP existantes et met à jour suivant les résultats une carte de l'Internet. Un des gros problèmes de ce projet reste ainsi un problème de représentation.
– Le projet américain AMP[8] de NLANR[9] [11] [12] est l'un des plus récent à avoir démarré. De ce fait, peu de résultats sont aujourd'hui disponibles, mais l'objectif de ce projet est de faire de « l'active probing ».

Dans la suite nous étudions plus en détails deux de ces projets :

2.1.1 National Internet Measurement Infrastructure (NIMI)

introduction

NIMI (National Internet Mesurement Infrastructure) est une plate-forme de mesures basées sur les mesures actives. Cette plate-forme est constituée de nombreux serveurs de mesure dédiés appelés NIMI probe. Un de ces objectifs majeurs fut d'en faire une plate-forme pouvant supporter un grand nombre de serveurs de mesures (potentiellement +1000). L'achitecture de NIMI est heritée du premier démon de mesure développé par Vern Paxson le *Network Probe deamon* (NPD). Un des aspects important de cette architecture est la séparation des taches. Ainsi, les mesures, les demandes de mesures, l'analyse des résultats et la configuration sont effectués par des agents distincts.

L'architecture

En général les démons NIMI appartenant à un même domaine sont configurés par un même point de configuration (*Configuration Point of Contact* CPOC) qui est défini par l'administrateur du domaine. Cependant l'architecture de NIMI permet de déléguer cette configuration ainsi que celle des outils de mesures a d'autres CPOC. Les outils de mesure font partie d'un module externe qui sont utilisés directement ou via des scripts par un démon appelé NIMI probe. Ce même démon peut être divisé en deux parties, l'une appellée *nimid* est une interface de communication avec le monde extérieur et la seconde est un ordonnanceur qui se charge d'exécuter les mesures. Enfin le client de mesure (*Measurement Client* MC) est le seul élément de l'architecture NIMI que les utilisateurs doivent lancer pour effectuer des mesures.

Le CPOC

Le CPOC est l'élément de la structure NIMI qui doit gérer la configuration des différents démons NIMI. C'est le CPOC qui défini les politiques d'accés aux démons par le biais d'une liste d'accès (*Access Control Lists* ACL). Ces régles prennent la forme d'un tableau ou les colonnes représentent une requete et les lignes les agents qui ont des droit d'accés (voir Fig. 2.2). Le CPOC fournit au démarrage à chaque démons *nimid*) de sa sphere de contrôle les régles d'accès qu'il doit appliquer.

[8] AMP : Active Measurement Project
[9] NLANR : National Laboratory for Applied Network Research

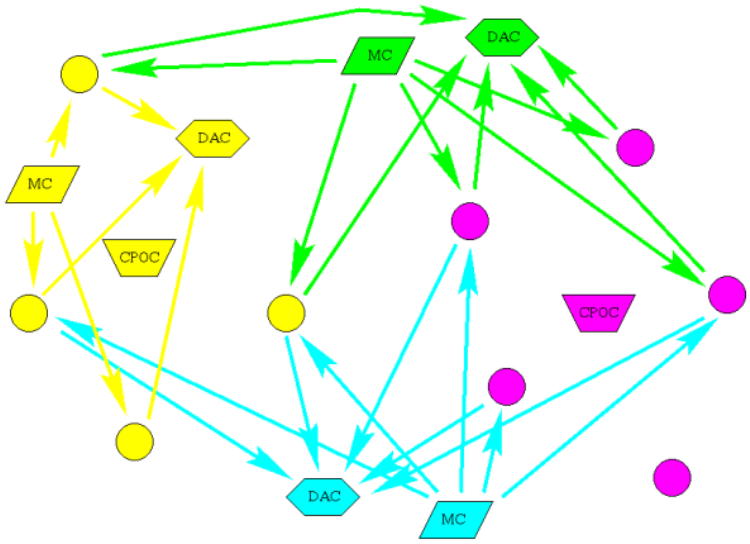

FIG. 2.1 – Communication entre les agents de la plate-forme NIMI

Cependant le CPOC peut déléguer un certain nombre de contrôle d'accès à d'autres CPOC. Ces régles peuvent éventuellement etre mise à jour a tout moment.

Le NIMI probe

Le *NIMI probe* joue un role fondamental dans l'exécution des mesures. En effet c'est lui qui traite les requêtes de mesures des clients et les exécute à la date fixée puis envoie les résultats au client d'analyse (*Data Analysis Client* DAC) qui doit les traiter. Le probe NIMI peut se diviser en deux démons bien distincts. Le nimid qui gére les accés grâce à la table ACLs et à qui les clients adressent leurs requêtes et le *scheduled* qui gére la mise en file d'attente des mesures, leur exécytion et l'empaquetage des résultats.

Le module de mesures

La plate forme NIMI n'a aucune connaissance des outils de mesures. elle ne sert qu'à organiser les mesures et le module de mesure se charge de les exécuter. En fait ce module se comporte comme un plug-in. Un module est composé de deux éléments, l'outil de mesure tel que ping ou traceroute et un script qui se charge d'exécuter la mesure avec les bons parametres et retourne les résultats sous une forme prédéterminée que la plate-forme NIMI sait traiter. A ce jour les modules de mesures existants sont :

- traceroute, mesure de trajet de bout en bout dans Internet
- mtrace, mesure d'arbre multicast
- treno, mesure de la bande passante

	ACL_ADD	ACL_DEL	TEST_ADD	TEST_DEL	DATA_GET	DATA_DEL	SAND_BOX	INHERITENCE
CPOC_KEY	T	T	T	T	T	T	F	F
MC_A_KEY	F	F	T	T	T	T	F	F
MC_B_KEY	T	T	T	T	T	T	mc_b	F
MC_B_KEY_2	F	F	TReno	T	F	F	mc_b2	MC_B_KEY
MC_C_KEY	T	T	< 512B	T	T	F	mc_c	F
MC_C_KEY_2	F	F	Zing	F	F	F	mc_c2	MC_C_KEY

FIG. 2.2 – tableau des autorisation possible

– cap/capd, mesure de la bande passante
– zing, mesure des delais et de pertes
– mflect,
– traffic/discardd, mesure de paramètre TCP
– ftp, mesure autour du protocole FTP

Pour créer un nouveau module, il faut créer le script qui correspond à cet outil et propager celui ci ainsi que l'outil a tous les démons NIMI. Cette propagation se fait aujourd'hui via une connection SSH.

Le MC et le DAC

Les utilisateurs utilisent le client de mesure (*Measurement Client* MC) pour commander des mesures. C'est un outil Unix qui peut etre lancé sur n'importe quelle station de travail se trouvant sur le meme réseau que le NIMI probe. Le MC communique directement avec le NIMI probe sans jamais passer par le CPOC. Quand un client fait une requête pour effectuer une mesure, il précise à quel DAC le NIMI probe doit renvoyer les résultats de mesure. Le DAC est le module qui est charger de stocker et analyser les données recoltées lors des mesures. le DAC peut faire partie du MC ou être un démon indépendant.

Conclusion

La plate-forme NIMI posséde de nombreux points positifs : c'est une plate forme trés flexible qui peut facilement évoluer. Ce que l'on peut regretter c'est qu'aucun outil de mesure unidirectionnel soit encore mis en place mais cette mise en place ne devrait pas poser trop de problémes.

2.1.2 RIPE Test Traffic Measurement

Présentation

Le RIPE NCC (Réseau IP Européen Network Coordination Center) est une organisation a but non lucratif fondée en 1992. Le but de cette association est de coordonner l'évolutions des services fournit par les opérateurs de réseaux IP. Aujourd'hui le RIPE NCC est composé de 2600 membres représentant 109 pays et de nombreux ISPs.
En 1997 les membres du RIPE NCC ont exprimés leur besoin de mesurer les performances sur les liens inter-opérateurs, ceci qui donna naissance au projet Test Traffic. Ce projet fut une étude de la faisabilité de la mesure unidirectionnelle des délais de transmission ainsi que des pertes entre deux sites. Quand il fut clair que ces mesures pouvait se faire a grande échelle, le projet Test Traffic fut rebaptisé Test Traffic Measurement service (TTM).
Le TTM effectue des mesures unidirectionnelles qui nécéssitent une synchronisation des machines effectuant les mesures.

La Synchronisation

Aujourd'hui plusieurs méthodes de synchronisation existent. NTP est une solution logicielle qui sychronise des machines grâce à un "serveur de temps" en échangeant des paquets et en évaluant les délais de transmission entre ce serveur et les machines. Un même "serveur de temps" peut étre utilisé par plusieurs machines mais il est quasi impossible d'obtenir une precision inférieure à 1ms avec cette méthode. Pour obtenir une plus grande précision on doit trouver accéder à une réference de temps universelle. Cela est possible en accédant au systéme GPS ou grace au CDMA. Ceci permet d'atteindre une précision de l'ordre de 100ns. Dans le projet TTM chaque machine effecuant des mesures est equipée d'un système GPS. Le recepteur GPS génére une pulsation par seconde sur le port serie de l'interface GPS, ces pulsations sont utilisées par le protocole NTP pour synchroniser l'horloge système de la machine. De ce fait toutes les machines sont parfaitement synchronisées.

Les mesures

Le RIPE TTM effectue des mesures unidirectionnelles. Des paquets sondes marqués par l'heure d'émission sont émis par une machine dédiée fonctionnant sous FreeBSD et sont envoyés à une autre machine. Le délai retour sera mesuré en inverse (Fig. 2.3. Les paquets "sondes" de 128 octets sont émis vers le port 60000 de la destination par le biais de UDP. Chaque Test Box du TTM posséde un démon qui planifie l'émission des paquets sonde. Chaque sonde contient la valeur de l'horloge système au moment de son émission. A la réception, la machine TTM de destination marque l'instant de réception et enregistre le paquet. La machine TTM de destination étant synchronisée par le meme mécanisme GPS que l'émetteur, la différence entre les deux marquages fait par l'émetteur et par le récepteur correspond au délai de transmission entre les deux machines. C'est le délai unidirectionnelle. L'ensemble des données ainsi recoltées par une machine TTM est transmis a un centre d'analyse.

En comparant la liste des paquets émis et la liste des paquet reçus on peut determiner le nombre de paquets perdus, qui ont été emis mais qui n'ont jamais été reçus. La transmission de ces données vers le centre d'analyse ne se fait que ponctuellement pour éviter un surcharge artificielle du réseau.

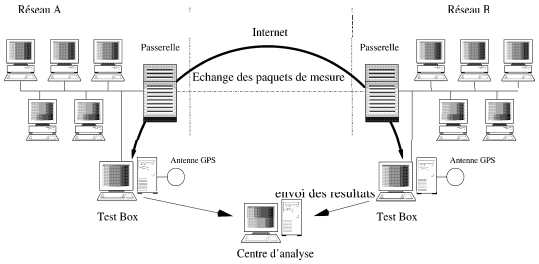

FIG. 2.3 – Systéme RIPE NCC Test Traffic Measurement

2.1.3 Le protocole OWDP de l'IPPM work group

introduction

Depuis 1997 le groupe de travail IPPM (IP Performance Metric) de l'IETF travail sur l'élaboration de métriques pour pouvoir chiffrer les performances des réseaux informatique. Le groupe IPPM a proposé, en février 2001, un dernier brouillon *draft* pour un protocole de mesure : l'OWDP (*One-way Delay Measurement Protocol*). Ce protocole est défini pour faire des mesures unidirectionnelles de délai à travers internet. l'IPPM espére le deploiement d'un maillage mondial de serveur libre OWDP qui pourrait permettre de remplacer l'utilisation de la mesure aller-retour de delai qu'utilisent les outils basés sur le protocole ICMP comme le ping.

Les principes

Le protocole OWDP est un protocole qui doit être très furtif, bien sécurisé et dont l'administration doit etre clairement séparée de l'activité de test. Il est furtif car les paquets "test" forment un flow UDP dont les ports d'origines et de destinations sont négotiés à chaque test et que la taille de ces memes paquets n'est pas fixé à l'avance par le protocole. Rien ne permet donc a priori d'identifier un paquet de test. De plus le protocole OWDP supporte un mode crypté qui rend les paquets illisibles et permet de détecter les erreurs de transmission au niveau du marquage temporel du paquet. La sécurité est renforcée par l'ajout d'un mode d'authentification pour les messages de controle et de test. Ceci permet d'éviter les accés non autorisés aux mesures et évite que des pirates générent de faux paquets test ou modifient le marquage temporel de vrai paquets test, ce qui fausserait les mesures. L'OWDP est constitué de deux protocoles bien indépendant : un protocole de controle l'OWDP-Control et un protocole de test l'OWDP-Test. L'OWDP-Control s'appuit sur une couche TCP et doit négocier, initier et controler les sessions de mesure puis récupérer leurs résultats. Il doit définir les ports UDP d'émission et de réception, la taille de la session de test, la taille des paquets de test et la répartition poissonienne des intervalles entre chaque paquet ainsi que de nombreux parametres defini dans la RFC 2330. Cependant l'OWDP-Control n'est pas indispensable au fonctionnement de l'OWDP-test. En effet la négociation de la session de test peut être effectuée par d'autres protocoles. Mais l'IPPM espére que l'OWDP-Control sera adopté pour l'administration des mesures unidirectionnelles.

L'OWDP-Test est un protocole qui s'appuit sur une couche UDP. Il gére le transfert des paquets de mesure qui transitent à travers le réseau ou l'internet. On fait une distinction entre les agents qui utilisent les protocoles OWDP (Fig. **??**). Ainsi on distingue,

- L'émetteur de session (Session-Sender) qui émet les paquets test.
- Le récepteur de session (Session-receiver), qui recoit les paquets test.
- Le Serveur qui gére plusieurs sessions de test et enregistre les résultats des sessions de test
- Le controleur (Control-Client) qui initie et fait la requête pour une session de test
- le récepteur (Retrieve-Client) qui demande les résultats d'une session de test.

.

Les liens non identifiés peuvent utiliser l'OWDP-control ou un protocole propriétaire. On peut aussi trouver plusieurs agents sur une meme machine, et a priori on utilisera le modèle suivant décrit dans la figure 2.5

Pour lancer ou administrer une session de mesure, le controleur ouvre une connection TCP avec le serveur fonctionnant sur la cible choisie, et sur un port prés defini par l'IANA, pour négocier les parametres de la session de mesure. Cette connection sera maintenu tout au long de la session de mesure, cependant les messages de controle ne sont envoyés qu'avant le début de la session de mesure et aprés sont achévement.

Le protocole OWDP est le premier protocole réellement orienté vers la mesure des performances du réseau. Le développement d'un tel protocole était nécessaire car jusqu'a présent c'était le protocole ICMP, a travers le ping, qui etait utilisé alors qu'il n'avait pas été conçus pour ce rôle.

2.2 Métrologie passive

Les projets de mesures passives sont apparus beaucoup plus récemment que les projets de mesures actives car ils nécessitent des systèmes de capture ou d'analyse du trafic en transit relativement avancés, et développés très récemment – même si quelques logiciels simples, mais aux capacités limitées, existent comme :

- TSTAT
- NTOP
- LIBCAP
- Tcpdump
- Tcptrace
- Etc.

Des équipements matériels plus performants sont en fait à la base de l'essor actuel en matière de métrologie passive, et ont notamment ouvert la voie à la métrologie passive microscopique (définie plus loin). Parmi ces équipements on peut citer :

- CISCO's Netflow [13] qui, à la base, est un mécanisme pour améliorer les performances des routeurs CISCO avec une gestion interne du routage par flux qui ne nécessitait de ne router que le premier paquet de chaque flux, les paquets suivants étant ensuite commutés vers le même port que le premier paquet du flux. L'aspect intéressant de Netflow en matière de métrologie se trouve dans le composant logiciel qui établit des statistiques sur le trafic qui transite par le routeur. Même si la commutation / routage Netflow a disparu des nouvelles générations de routeurs CISCO, le composant logiciels de mesures statistiques est resté.

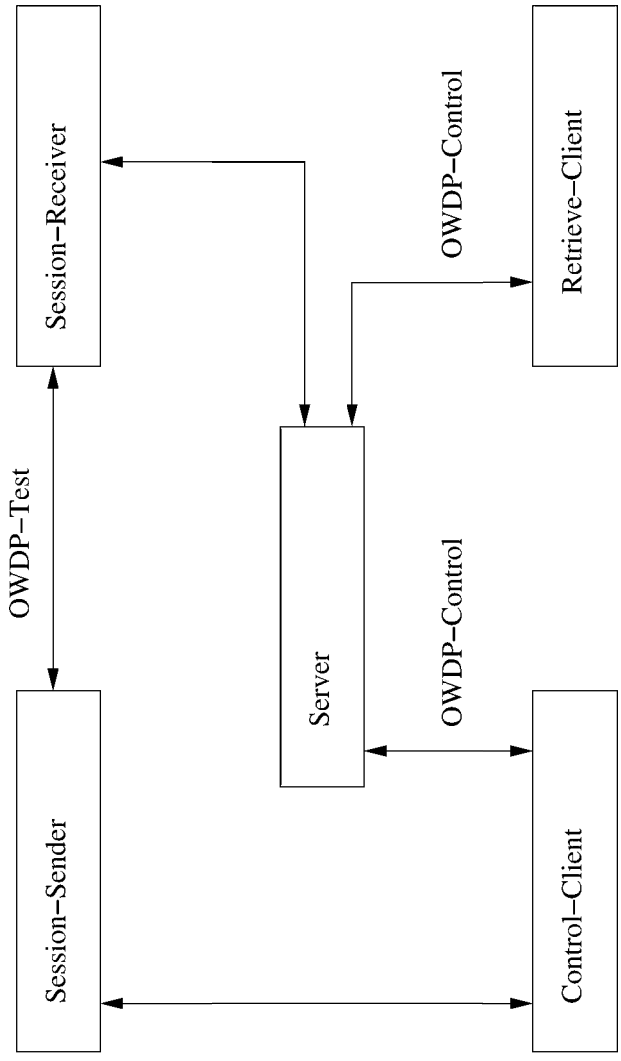

FIG. 2.4 – Architecture de l'OWDP-test

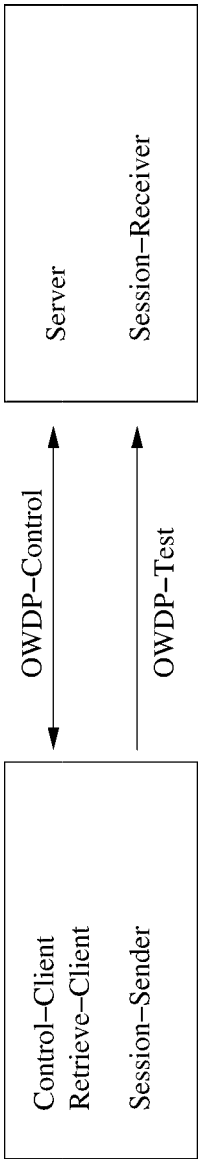

FIG. 2.5 – Architecture de l'OWDP-test avec plusieurs agents

– OCxMON [14] (essentiellement pour ATM) est une carte de capture des entêtes de chaque paquet pour analyse.

Le principe des mesures passives est de regarder le trafic et d'étudier ses propriétés en un ou plusieurs points du réseau. L'avantage des mesures passives et qu'elles ne sont absolument pas intrusives et ne changent rien à l'état du réseau. De plus, elles permettent des analyses très avancées. En revanche, il est très difficile de déterminer le service qui pourra être offert à un client en fonction des informations obtenues en métrologie passive. Les systèmes de métrologie, actifs comme passifs, peuvent aussi se différencier en fonction du mode d'analyse des traces. Ainsi, le système peut faire une analyse en-ligne ou hors-ligne. Dans le cadre d'une analyse en-ligne, toute l'analyse doit être effectuée dans le laps de temps correspondant au passage du paquet dans la sonde de mesure. Une telle approche, temps-réel, permet de faire des analyses sur de très longues périodes et donc d'avoir des statistiques significatives. Par contre, la complexité maximale pour ces analyses reste très limitée à cause du faible temps de calcul autorisé. Une analyse hors-ligne par contre oblige la sonde à sauvegarder une trace du trafic pour analyse ultérieure. Une telle approche demande ainsi d'énormes ressources ce qui représente une limitation pour des traces de très longues durées. Par contre, une analyse hors-ligne permet des analyses extrêmement complètes et difficiles, permettant d'étudier des propriétés non triviales du trafic. De plus, comme les traces sont sauvegardées, il est possible de faire plusieurs analyses différentes sur les traces, et de corréler les résultats obtenus pour une meilleure compréhension des mécanismes complexes du réseau.

L'endroit idéal pour positionner des sondes de mesures passives est indéniablement dans les routeurs. CISCO a ainsi développé le module *Netflow* [13] dans ces routeurs, qui scrute le trafic en transit, et génère régulièrement des informations statistiques sur ce trafic. *Netflow* a ainsi été utilisé dans de nombreux projets. L'expérience montre toutefois que les performances de *Netflow* restent limitées (code écrit en Java et interprété), et que l'influence sur les performances du routeurs sont non négligeables.

Toutefois, le premier projet connu - le projet AT&T *Netscope* [15] - a débuté il y a environ 5 ans et repose sur ce système *Netflow* de CISCO. Ce projet de mesure passive en ligne a pour but d'étudier les relations entre le trafic transitant en chaque nœud du réseau et les tables de routage des routeurs. L'objectif final est d'utiliser ces résultats pour améliorer les politiques et décisions de routage, afin d'équilibrer au mieux la charge dans les différents liens du réseau, et ainsi améliorer la qualité de service perçue par chaque utilisateur. C'est de la tomographie afin de trouver ensuite des politiques d'ingénierie des trafics adéquates.

Vern Paxson et al. d'ACIRI ont également conduit un projet de mesure passive en ligne dont l'objectif était de proposer un modèle pour les arrivées de flux et de paquets sur les liens de l'Internet. Ce travail [16] achevé depuis 1995 a été et reste une référence dans le milieu de la recherche Internet. Cependant, aujourd'hui, avec l'apparition de nouvelles applications qui n'existaient pas à l'époque et avec les changements dans la façon d'utiliser l'Internet, ce travail doit être reconduit. Les résultats ne sont certainement plus valables aujourd'hui, et il n'existe aucune technique sûre d'extrapolation de ces résultats pour essayer de modéliser le trafic d'aujourd'hui.

De façon plus générale, le laboratoire CAIDA[10] [17] à San Diego, Californie, est spécialisé dans l'étude du trafic Internet et mène un projet dont l'objectif est d'étudier sur le long terme l'évolution du trafic, avec l'apparition des nouvelles applications comme les jeux, le commerce électronique, etc. D'autre part, ce projet étudie aussi les variations en fonction du moment de la journée, du jour de la semaine, de la période de l'année, etc. [18] [19]. Pour ce faire, le système de métrologie repose

[10] CAIDA : Cooperative Association for Internet Data Analysis

sur les modules OC3MON [14] et OC12MON qui permettent de traiter le trafic de liens IP/ATM dont les capacités respectent respectivement les normes OC3 (155 Mbps) et OC12 (622 Mbps). D'autre part, pour l'analyse statistique, CAIDA a développé la suite logicielle *CoralReef* [20] [21] qui est complémentaire des systèmes OCxMON. D'autres études sont en cours, notamment sur les problèmes de représentation de l'Internet [22], ou d'étude des délais. Pour plus d'information, le lecteur pourra consulter [17]. A noter que les systèmes OCxMON sont aussi utilisés par Worldcom et NLANR pour faire de la métrologie sur le réseau vBNS [23].

En France, le projet NetMet [24] a débuté il y a 2 ans environ et a été conçu et développé par et pour des administrateurs réseaux. Il repose sur la technologie CISCO Netflow et offre à ses utilisateurs deux approches passives de niveaux différents pour la métrologie. La première est une étude macroscopique du trafic comme le font la plupart des analyseurs et logiciels d'administration réseaux, notamment ceux basés sur l'utilisation des MIB SNMP. Le second niveau, plus fin, consiste en une trace de chaque flux qui a traversé le routeur. Il n'y a à l'heure actuelle pas d'outil pour analyser cette trace de flux (ils doivent être développés par les utilisateurs / administrateurs en fonction de leurs souhaits), mais cette trace est une mine d'information à la fois pour des activités d'administration et d'opération des réseaux, mais également pour la recherche en réseau. A noter également qu'une extension de NetMet appelée NetSec [25] propose des facilités pour l'étude des attaques générées à l'encontre d'un réseau. Il repose sur les mêmes principes métrologiques que NetMet (approches macroscopique du trafic, et orientée flux).

Enfin, SPRINT a démarré il y a presque 3 ans un des projets les plus ambitieux du moment basé sur des mesures actives hors ligne. Ainsi, Sprint enregistre des traces complètes de tous les entêtes de tous les paquets qui transitent en certains point de son réseau IP. Cette granularité microscopique permet d'approfondir les analyses que l'on peut faire dans la compréhension des interactions qui existent entre tous les flux, les mécanismes des routeurs, etc. A noter que le système IPMON de Sprint, décrit plus en détail dans la suite, repose sur la carte DAG [26] conçue par l'université de Waikato en Nouvelle Zélande et qui se charge d'extraire les entêtes des paquets, de les estampiller suivant une horloge GPS [27] et de les stocker sur un disque dur [28].

2.2.1 Le système de métrologie de Sprint

Le but du projet IPMON [29] de Sprint est de collecter des données sur son Backbone Internet afin de fournir des bases fiables à différents projets de recherche. En particulier, des études sur le comportement de TCP, l'évaluation des performances du réseau, et le développement d'outils d'analyse du réseau.

L'approche métrologique dans IPMON comporte trois phases :
- la première consiste à capturer des traces sur le réseau
- la seconde partie concerne l'analyse des traces (selon divers paramètres)
- la troisième consiste à déduire à partir des analyses des informations pertinentes sur le réseau et sur les services qu'il doit assurer ; cette étape permet aussi de comparer des modèles de trafic au cas réel.

Le système de capture consiste à insérer des *splitters* optiques sur les liens au niveau des POP. Ainsi, on collecte et on date (à l'aide d'une horloge GPS) toutes les entêtes TCP/IP (44 octets). L'utilisation d'une horloge globale permet, entre autre, de déterminer les délais de bout en bout, des temps de passage dans les routeurs, d'avoir une idée sur le temps passé dans les files d'attentes des routeurs, etc. Enfin, ces informations sont transférées sur la plate-forme d'analyse.

Les avantages d'un tel système sont nombreux : en particulier, on peut citer :
– la transparence et la non intrusivité pour le réseau
– les données collectées sont réalistes puisque provenant d'un réseau IP opérationnel
– capture de l'entête TCP/IP complète
– archivage des traces pour des exploitations futures
– analyse différée qui permet une exploitation très avancée
Mais un tel système a aussi des inconvénients :
– il requiert un développement sur un réseau opérationnel
– limitations technologiques (PC, espace disques)
– ne donne lieu qu'à des analyses différées
– 44 octets sont quelquefois insuffisants
Le système IPMON supporte des liens allant de OC3 (155 MBits/s) à OC48 (2480 MBits/s). Il est composé de PC Linux, d'une interface réseau POS/PCI (Packets Over Sonet) ainsi que de deux disques durs de taille importante (en effet une trace sur un lien OC3 de 24h, en supposant que la ligne est occupé à 40% au maximum, occupe 54 Go).

Les principaux résultats de ce projet seront détaillés dans les prochaines parties de ce rapport.

2.3 Description du réseau de Renater

Aujourd'hui plus de 640 sites ayant leur activité dans les domaines de la recherche, la technologie, et l'enseignement sont connectés au réseau RENATER (Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche). Connectés à très haut débit, il leur permet de communiquer entre eux et d'avoir un accès aux autres réseaux de recherche et à l'Internet. L'infrastructure actuelle du réseau est en phase de migration (RENATER–2bis – RENATER–3). Cette évolution aboutira à une nouvelle architecture à très haut débit, ainsi qu'à la transformation des services pilotes actuels (IPv6, Multicast) en services natifs. Ce chapitre présente l'architecture du réseau et les outils de métrologie mis en place par le GIP RENATER (Groupement d'Intérêt Public), maître d'ouvrage du réseau.

2.3.1 Présentation de l'architecture

Le réseau RENATER offre une couverture nationale, avec des points de présence dans chacune des régions et dans les départements et territoires d'Outre Mer. Tous ces points sont interconnectés par des liaisons à haut débit et le réseau dispose en plus de liaisons vers l'Amérique du Nord, l'Europe et l'Asie. Le réseau a également un accès sur le point d'échange SFINX (Service for French INternet eXchange) sur lequel sont raccordés une soixantaine d'ISP (Internet Service Provider).

L'infrastructure métropolitaine est accessible aux travers des réseaux de collecte (réseaux régionaux ou métropolitains) raccordés aux NR (Nœud RENATER ancien NRD Nœud Régional Distribué). La figure 2.6 représente la partie métropolitaine de RENATER–3, la figure 2.7 les accès vers les réseaux européens, américains, asiatiques et le point d'échange SFINX.

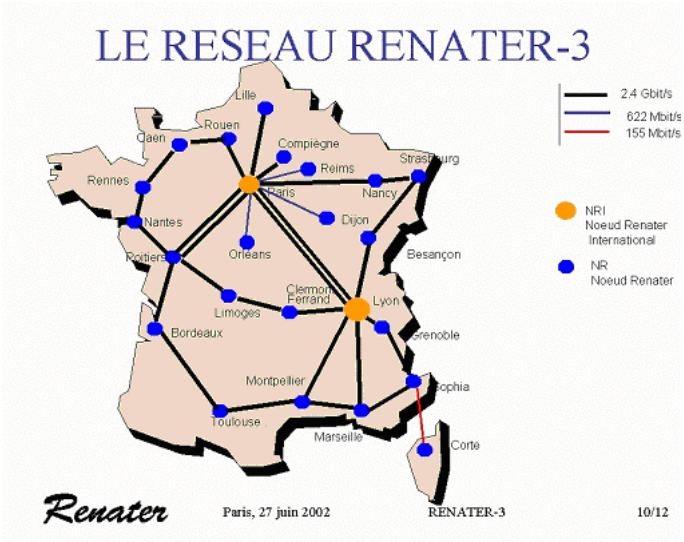

FIG. 2.6 – Le réseau Renater–3

2.3.2 Présentation des outils de métrologie utilisés sur Renater

La métrologie permet d'effectuer des mesures qualitatives et quantitatives de l'ensemble des trafics présents sur le réseau. Elle s'appuie sur des mesures réalisées au niveau de l'ensemble des matériels physiques (routeurs, commutateurs, ...). Les observations tiennent compte de l'hétérogénéité des protocoles au niveau des diverses couches de communication. La supervision s'effectue ainsi sur l'ensemble des NR et NRI (nœuds internationaux) de RENATER, sur les points de présence du SFINX et sur les liaisons internationales.

La métrologie est réalisée à l'aide de trois techniques complémentaires. La première consiste en l'interrogation de compteurs mis en place sur les équipements (les Managment Information Bases), par le protocole SNMP (Simple Network Managment Protocol). Ces "agents" permettent la surveillance (à distance et à intervalle régulier) des paramètres physiques et logiques des équipements (température, taux de charge des CPU, nombre de paquets transmis, débits des différents liens, nombre d'erreurs, ...). La deuxième technique consiste à mesurer les temps de réponses immédiats des équipements. Cette méthode délivre des informations sur l'état réactif de l'équipement (en marche, en panne, réponse trop lente) et ainsi sur la charge des liaisons (des délais trop long lors de l'interrogation peuvent impliquer une saturation). La disponibilité du réseau est ainsi visualisable au travers d'une interface web (figure 2.8).

La dernière méthode réside dans l'utilisation de la technologie « NetFlow » qui permet la mesure du trafic au niveau des flux circulants sur le réseau (figure 4). Elle permet d'observer certains problèmes de sécurité (attaques de type « Déni de Service »), de mesurer le trafic généré par un site en particulier (figure suivante), et de quantifier les protocoles des couches transports et applications par rapport à l'ensemble du trafic.

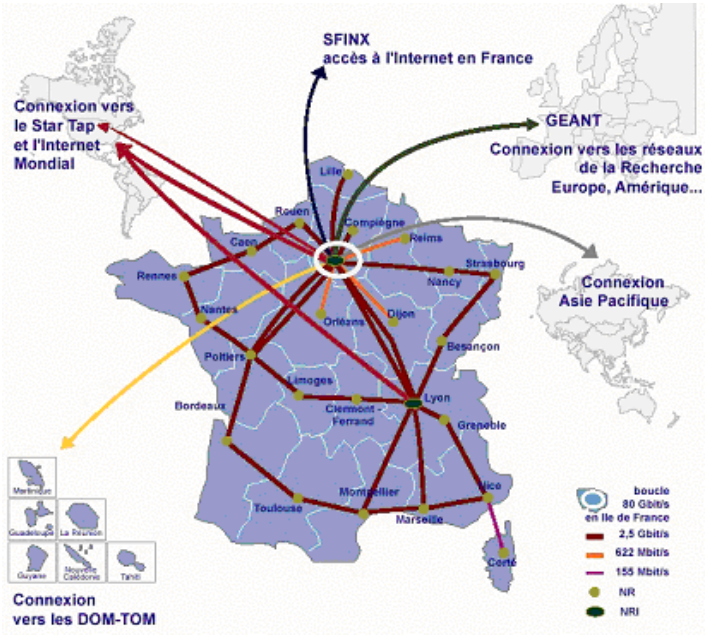

FIG. 2.7 – Interconnexion de Renater–3 avec l'étranger

Ces techniques fournissent en temps réel une vue globale du fonctionnement du réseau. L'enregistrement dans une base de données de toutes les informations recueillies sert de base pour définir les évolutions en termes de dimensionnement du réseau au regard de la croissance du trafic.

2.3.3 Conclusion

La métrologie est déjà présente dans RENATER mais ne répond pas encore à toutes les questions que l'on peut se poser : tarification des services, lien entre ressources consommées et perturbations induites sur les trafics parallèles, problèmes des « faux trafics » (attaques, . . .).

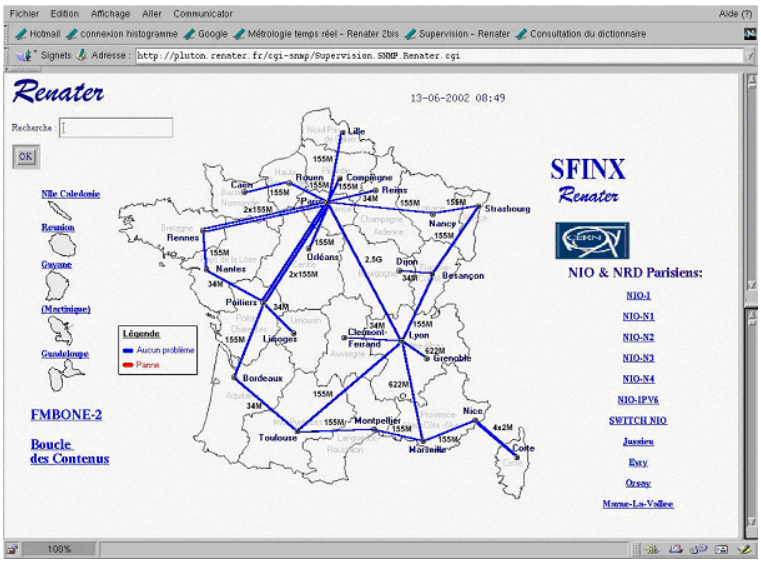

FIG. 2.8 – Interface de visualisation de l'état du réseau Renater

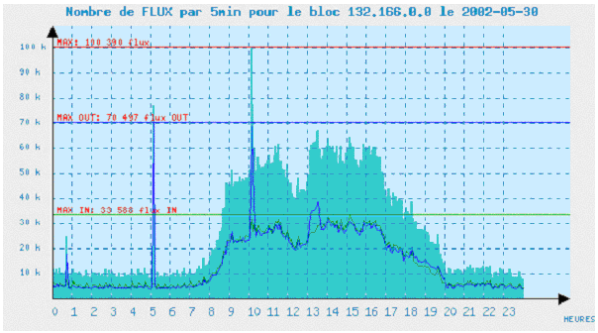

FIG. 2.9 – Mesure du trafic avec NetFlow

Bibliographie

[1] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, *"Framework for IP Performance Metrics"*, RFC 2330, May 1998.

[2] G. Almes, S. Kalidindi, M. Zekauskas, *"A One-way Delay Metric for IPPM"*, RFC 2679, September 1999.

[3] G. Almes, S. Kalidindi, M. Zekauskas, *"A One-way Packet Loss Metric for IPPM"*, RFC 2680, September 1999.

[4] G. Almes, S. Kalidindi, M. Zekauskas, *"A Round-trip Delay Metric for IPPM"*, RFC 2681, September 1999.

[5] S. Kalidindi, M.J. Zekauskas, *"Surveyor : An infrastructure for Internet performance measurements"*, Proceedings of INET'99, June 1999.

[6] V. Paxson, A. Adams, M. Mathis, *"Experiences with NIMI"*, PAM (Passive and Active Measurements) Workshop, 2000.

[7] *"RIPE NCC web site"*, http ://www.ripe.net.

[8] A. Adams, T. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S. B. Moon, V. Paxson, D. Towsley, *"The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavio"*, IEEE Communications, 38(5), May 2000.

[9] R. Caceres, N. Duffield, D. Towsley, J. Horowitz, *"Multicast-based Inference of Network-internal loss characteristics"*, IEEE Transactions on Information Theory, vol. 45, no. 7, November 1999.

[10] *"Netsizer web site"*, http ://www.netsizer.com.

[11] *"AMP web site"*, http ://watt.nlanr.net.

[12] T. McGregor, H.-W. Braun, J. Brown, *"The NLANR network analysis infrastructure"*, IEEE Communications, vol. 38, no. 5, May 2000.

[13] *"NetFlow Services Solutions Guide"*, http ://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/.

[14] J. Apsidorf, *"OC3MON : Flexible, affordable, high performance statistics collection"*, Proceedings of INET, June 1997.

[15] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, F. True, *"Deriving traffic demands for operational IP networks : Methodology and Experience"*, ACM SIGCOMM Conference, Stockholm, 2000.

[16] V. Paxson, and S. Floyd, *"Wide-Area Traffic : The Failure of Poisson Modeling"*, IEEE/ACM Transactions on Networking, Vol. 3 No. 3, June 1995.

[17] *"CAIDA web site"*, http ://www.caida.org.

[18] K.C. Claffy, G. Miller, K. Thompson, *"The nature of the beast : Recent traffic measurements from an Internet backbone"*, Proceedings of INET'98, Geneva, Switzerland, July 1998.

[19] S. McCreary, K. C. Claffy, *"Trends in wide area IP traffic patterns"*, in ITC Specialist Seminar, Monterey, California, May 2000.

[20] *"CoralReef website"*, http ://www.caida.org/tools/measurement/coralreef.

[21] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, K. Claffy, *"The Architecture of the CoralReef : An Internet Traffic Monitoring Software Suite"*, PAM'2001 (Passive and Active Measurements) workshop, Amsterdam, The Netherlands, April 2001.

[22] B. Huffaker, M. Fomenkov, D. Moore, K. Claffy, *"Macroscopic Analyses of the Infrastructure : Measurement and Visualization of Internet Connectivity and Performance"*, PAM'2001 (Passive and Active Measurements) workshop, Amsterdam, The Netherlands, April 2001.

[23] *"vBNS web site"*, http ://www.vbns.net.

[24] A. Simon, *"netMET - Network's METrology : Une solution de métrologie générale pour les réseaux régionaux, métropolitains et de campus"*, Journées réseaux (JRES'2001), Lyon, France, 10-14 décembre 2001.

[25] B. Martinet, J-F. Scariot, *«La métrologie, base pour la sécurité : NetSEC»*, Journées réseaux (JRES'2001), Lyon, France, 10-14 décembre 2001.

[26] *"Dag 4 SONET network interface"*, http ://dag.cs.waikato.ac.nz/dag/dag4arch.html.

[27] *"Dag synchronization and timestamping"*, http ://dag.cs.waikato.ac.nz/dag/docs/dagduck_v2.1.pdf.

[28] J. Cleary, S. Donnelly, I. Graham, A. McGregor, M. Pearson, *"Design principles for accurate passive measurement"*, PAM 2000, Hamilton, New Zealand, April 2000.

[29] C. Fraleigh, C. Diot, S. Moon, P. Owezarski, D. Papagiannaki, F. Tobagi, *"Experiences Monitoring Backbone IP Networks"*, PAM (Passive and Active Measurements) Workshop, Amsterdam, The Netherlands, April 23-24, 2001.

Chapitre 3

Caractérisation du trafic

Nabil Benameur, France Télécom R&D, Philippe Olivier, France Télécom R&D,
James Roberts, France Télécom R&D, Philippe Owezarski, LAAS-CNRS,
Kavé Salamatian, LIP6-UPMC

3.1 Introduction

Le trafic de type Internet, en plus de son exceptionnelle croissance, présente les propriétés suivantes : le protocole IP supporte de multiples applications, dont les diverses caractéristiques se mélangent au sein des ressources du réseau ; les paquets de gestion partagent le même réseau que les paquets de données utiles (il n'existe pas de réseau de signalisation séparé) ; enfin, les mécanismes de contrôle (tels TCP) interagissent fortement avec le trafic offert, produisant un trafic observable grandement modifié. Ces caractéristiques confèrent à la structure du trafic IP un caractère d'une grande complexité, dont une manifestation est par exemple le phénomène de dépendance à long terme ou d'auto-similarité.

Sa connaissance nécessite donc d'être recherchée (à défaut d'être maîtrisée) plus ou moins continuellement en fonction de son évolutivité. La nécessité d'effectuer des mesures et des analyses du trafic se justifie plus particulièrement par les besoins suivants :

- Evaluer la demande en trafic des utilisateurs, notamment dans le cadre de la fourniture à venir de classes de service différenciées (COS).
- Dimensionner les ressources du réseau : capacité de traitement des routeurs ; débit de transmission des liens ; taille des buffers aux interfaces. Adapter la gestion opérationnelle de ces ressources à l'évolutivité temporelle de la demande en trafic.
- Contrôler la QoS offerte par le réseau : taux de perte de paquets, délai et gigue de transfert des paquets de bout en bout pour les applications à contraintes temps-réel, débit utile de transport des flux de trafic de données.

– Tester l'adéquation des modèles de performance élaborés au moyen de calculs analytiques ou de simulations, tant au niveau de la validation des hypothèses considérées que de la pertinence des résultats.

Différents types de mesure du trafic peuvent être mises en œ uvre : mesures passives ou mesures actives (voir Chapitre 3). Dans le présent état de l'art, nous ne parlerons pas des outils, matériels et logiciels, permettant de mettre en œ uvre ces mesures passives et actives, un chapitre séparé étant dévolu à cette tâche. Nous nous attacherons ici à synthétiser les différents modèles de description du trafic qui ont été élaborés au moyen de l'analyse de campagnes de mesures. Celles-ci ne seront que peu évoquées en elles-mêmes. Il s'agira généralement d'observations réalisées à l'aide de sondes passives, plus spécialement adaptées à la fourniture de paramètres descriptifs du trafic. Les sondes actives fournissent quant à elles des informations plutôt liées à la performance du trafic et à la QoS réseau, sujets débordant quelque peu du cadre assigné à ce chapitre.

3.2 Diversité du trafic Internet

3.2.1 Caractéristiques générales du trafic IP

Un réseau de type Internet est dit multi-service : il a vocation à transporter un grand nombre de types de service possédant des caractéristiques de trafic différentes et éventuellement des contraintes de Qualité de Service (QoS) différenciées. Cependant, dans le souci d'une modélisation simplifiée autant que pour les besoins opérationnels de gestion du réseau, l'on recherche plutôt une classification grossière des différents types de trafic [74]. La plupart des auteurs s'accordent généralement pour distinguer deux grandes classes de trafic de télécommunications dans les réseaux à haut débit :

– Le trafic de type "**streaming**", dont la durée et le débit ont une réalité intrinsèque bien que variable éventuellement. Souvent associé à la notion de services "orientés connexion", son intégrité temporelle doit être préservée par le réseau. Le délai de transfert des données de même que sa variation, la gigue, doivent être contrôlables, tandis qu'un certain degré de perte de paquets peut être tolérable. Les flux de trafic streaming sont typiquement produits par les services téléphoniques et vidéo (vidéoconférence ou téléchargement "on-line" de séquences).
– Le trafic dit "**élastique**", ainsi nommé car son débit peut s'adapter à des contraintes extérieures (bande passante insuffisante par exemple) sans pour autant remettre en cause la viabilité du service. Cette classe de trafic est essentiellement engendrée par le transfert d'objets numériques par nature (par opposition au transfert en mode numérique d'informations analogiques à la source) tels que des pages Web (application HTTP), des messages électroniques (e-mail, application SMTP) ou des fichiers de données (application FTP). Le respect de leur intégrité sémantique est indispensable mais les contraintes de délai de transfert sont moins fortes. Cette intégrité sémantique est la plupart du temps assurée par le protocole de transport (TCP) et ne constitue donc pas un élément de performance sur lequel l'opérateur de réseau puisse agir ; en revanche, le maintien d'un certain débit effectif minimum de transfert des documents est un objectif de QoS.

Le trafic de type élastique est actuellement largement majoritaire sur les réseaux IP : on constate couramment [85] des proportions supérieures à 95% en volume (octets) et à 90% en nombre de paquets pour le trafic sous TCP, protocole sous lequel fonctionnent la plupart des applications men-

tionnées ci-dessus. Des proportions similaires ont été observées récemment sur le réseau de France Télécom (voir le paragraphe 3.2.2).

L'analyse des caractéristiques du trafic Internet s'effectue commodément en se plaçant à un niveau de représentation selon trois entités de trafic, correspondant à trois échelles de temps différentes et, quoique de manière assez grossière, à trois niveaux (couches) de la pile protocolaire des réseaux de données :

- Les "**paquets**" forment l'entité de trafic la plus fine que l'on considère dans les réseaux de données, le paquet étant l'unité élémentaire traitée par la couche réseau des protocoles. Les paquets sont *a priori* de longueur variable dans un réseau IP et leur processus d'apparition est très complexe, en raison notamment de la superposition de services de nature très diverse et de l'interaction des couches protocolaires (dispositifs de contrôle de flux et de retransmission sur perte de paquets, tels TCP [13]). Le trafic au niveau paquet possède la caractéristique unanimement reconnue d'auto-similarité, laquelle rend très ardue l'évaluation de ses performances à ce niveau. Les échelles de temps décrivant le processus des paquets sont la microseconde et la milliseconde, en fonction des ordres de grandeur du débit de transmission des liens.
- Les "**flots**" constituent une entité de trafic intermédiaire que l'on pense être la mieux adaptée pour effectuer les études d'ingénierie du trafic IP (voir 3.3.3). Ils correspondent à des transferts plus ou moins continus de séries de paquets associés à une même instance d'une application donnée. Les flots de type streaming sont associés à des communications audio/vidéo (téléphonie sur IP, vidéoconférence) ou encore à des téléchargements en temps réel de séquences vidéo. Les flots de trafic élastique sont créés par le transfert d'un fichier, d'un message, d'un objet (ou document) au sein d'une page HTML, etc. Un flot correspond donc plus ou moins à la couche transport de la pile protocolaire Internet ; mais pas complètement puisque cette notion n'est pas nécessairement équivalente à celle d'une connexion TCP, p. ex., comme on le verra par la suite. On peut estimer que les flots ont une durée s'étendant de quelques secondes à quelques minutes, voire quelques heures.
- Au plus haut niveau, on peut tenter de définir la notion de "**sessions**" dans le but de se rapprocher des périodes d'activité des utilisateurs (transposition de la notion d'appels considérée en téléphonie à commutation de circuits). Pour le trafic streaming, ce niveau ne se distingue guère de celui des flots, du moins temporellement, puisque ce dernier correspond déjà à des communications ou des appels. S'agissant du trafic élastique, les sessions peuvent être associées à des connexions Telnet, FTP, ou à des envois de messages électroniques. La notion de session est (encore) plus floue au sujet des connexions de type WWW ("World Wide Web") selon le protocole HTTP : on peut par exemple la définir comme étant la durée de transfert d'une page HTML dans son ensemble (comportant plusieurs objets à transférer) ou d'une suite de pages associées à une même consultation. Les sessions sont générées par la couche application des réseaux et l'ordre de grandeur de leur durée se situe entre quelques minutes et quelques heures.

3.2.2 Répartition par protocole

Sur l'Internet, au-dessus de la couche réseau IP, les protocoles de transport de loin les plus répandus sont UDP (User Defined Protocol) et TCP (Transport Control Protocol). Le trafic TCP est largement majoritaire, comme le montre le tableau 3.1 où l'on donne les fourchettes dans lesquelles s'insèrent la plupart des proportions de trafic UDP/TCP reportées dans la littérature. Il est intéressant de noter que ces valeurs sont relativement stables depuis quelques années (environ 5 ou 6 ans) [53].

Le protocole UDP ne transporte encore que très peu de trafic lié à des applications utilisateurs (les services de Voix sur IP ou de téléchargement audio/vidéo en temps-réel n'ont pas encore réussi leur percée). Comme on le verra plus loin, une grande partie des flux UDP est générée par l'application réseau DNS, ce qui explique leur poids en octets encore plus faible qu'en termes de paquets ou de flots.

D'autres protocoles de transport sont présents dans les observations, mais ne contribuent que pour une proportion négligeable au volume de trafic véhiculé, mesuré en nombre de paquets ou d'octets (en termes de flots, leur poids pourrait monter jusqu'à 1 ou 2% selon la définition utilisée pour identifier les flots) : ICMP (Internet Control Message Protocol), RSVP (Reservation Protocol), GRE (General Routing Encapsulation), SIPP-ESP (SIPP Encap Security Payload) et NHRP (NBMA Next Hop Resolution Protocol). On ne considère généralement pas ces protocoles lors des études de caractérisation de trafic, en particulier celles rapportées ici.

TAB. 3.1 – Proportions de trafic par protocole de transport

	TCP	UDP	Autres
% paquets	85 - 90	10 - 15	négligeable
% octets	94 - 98	2 - 6	négligeable

3.2.3 Répartition par application

Une étude très détaillée des caractéristiques et des profils de trafic [23, 85], basée sur des campagnes de mesure réalisées sur le réseau dorsal de MCI, fait référence comme source d'informations depuis plusieurs années. Deux liens, respectivement dédiés au trafic domestique (interne aux USA) et au trafic international (liaison USA - GB) sont observés durant une semaine. Des résultats très complets sont présentés sur les poids respectifs, exprimés en unité de flots, de paquets ou d'octets, des différentes applications ou des différents protocoles entrant dans la composition du trafic et sur leurs variations au cours du temps (les profils de trafic obtenus ne sont pas forcément similaires à ceux obtenus en France par exemple, en raison des décalages horaires à l'intérieur même des USA).

Le tableau 3.2 fournit les répartitions de trafic par application (reconnues par leur numéro de port) parmi les paquets sous protocole TCP. Y figurent les résultats de la campagne MCI citée ci-dessus (chiffres assez approximatifs) de même que ceux issus d'observations effectuées sur le réseau IP de France Télécom en deux points de collecte (POP) du trafic Internet. Pour ces dernières mesures, les deux sens de transmission, montant et descendant, sont distingués, étant donnée la structure hiérarchique très marquée du réseau.

Par rapport à la répartition par protocole (paragraphe précédent), la ventilation par application est beaucoup moins stable selon le lieu ou la période de mesure. On note cependant quelques faits saillants que l'on retrouve dans ce tableau et dans de nombreux autres résultats non reportés ici. En premier lieu, l'application dominante, du moins sur un réseau dorsal, est HTTP (protocole de navigation sur le World Wide Web), et ce dans une large proportion. Mais ceci est un phénomène relativement récent : l'explosion du trafic Web s'est produite au milieu des années 90 pour les USA (un peu plus tard dans d'autres pays) en liaison avec la "démocratisation" de l'accès à Internet. Auparavant, le trafic Internet était essentiellement dominé par les applications FTP, SMTP, NNTP et

TAB. 3.2 – Proportions de trafic par application sur TCP

		HTTP	SMTP	POP3	FTP	NNTP	Autres
MCI 1997	% paquets	75	6	-	3	< 1	15
Lien domestique	% octets	80	5	-	5	2	8
POP 1 FT 2000	% paquets	65	2	3	5	2	23
Sens montant	% octets	33	8	1	9	< 1	48
POP 1 FT 2000	% paquets	65	2	3	6	2	22
Sens descendant	% octets	64	0	2	9	3	22
POP 2 FT 2001	% paquets	79	3	3	-	-	15
Sens montant	% octets	65	17	1	-	-	17
POP 2 FT 2001	% paquets	72	2	4	-	-	22
Sens descendant	% octets	72	< 1	4	-	-	24

Telnet utilisées dans le monde professionnel [66]. Derrière HTTP, les applications les plus utilisées sont liées à la messagerie (SMTP pour le transport dans le réseau et POP3 pour la consultation des boîtes aux lettres sur les serveurs), au transfert de fichiers (FTP) ou aux services de News (NNTP, mais qui reste très concentré temporellement, en général la nuit). A noter toutefois qu'il y a deux ans (en 2000), l'application Napster, un précurseur pour l'échange de fichiers audio au format MP3 a éclaté, comme le montre la figure 3.1, au point de devenir, par la quantité de trafic générée, la seconde application dans l'Internet après le web.

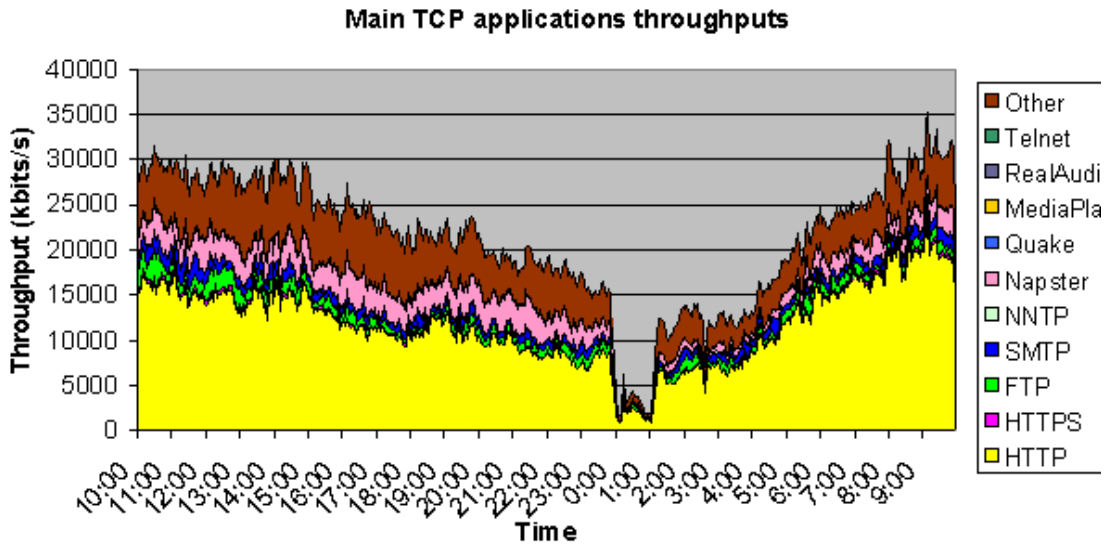

FIG. 3.1 – Exemple de répartition du trafic par application sur un lien Internet (2000)

Concernant le trafic UDP, la seule application généralement identifiée est DNS (traduction d'adresses IP), laquelle génère de nombreux flots de petite taille (1 paquet de faible taille, inférieure à 80 octets). Son poids est cependant faible, de l'ordre de 5 à 10% des paquets et de 2 à 8% des octets du

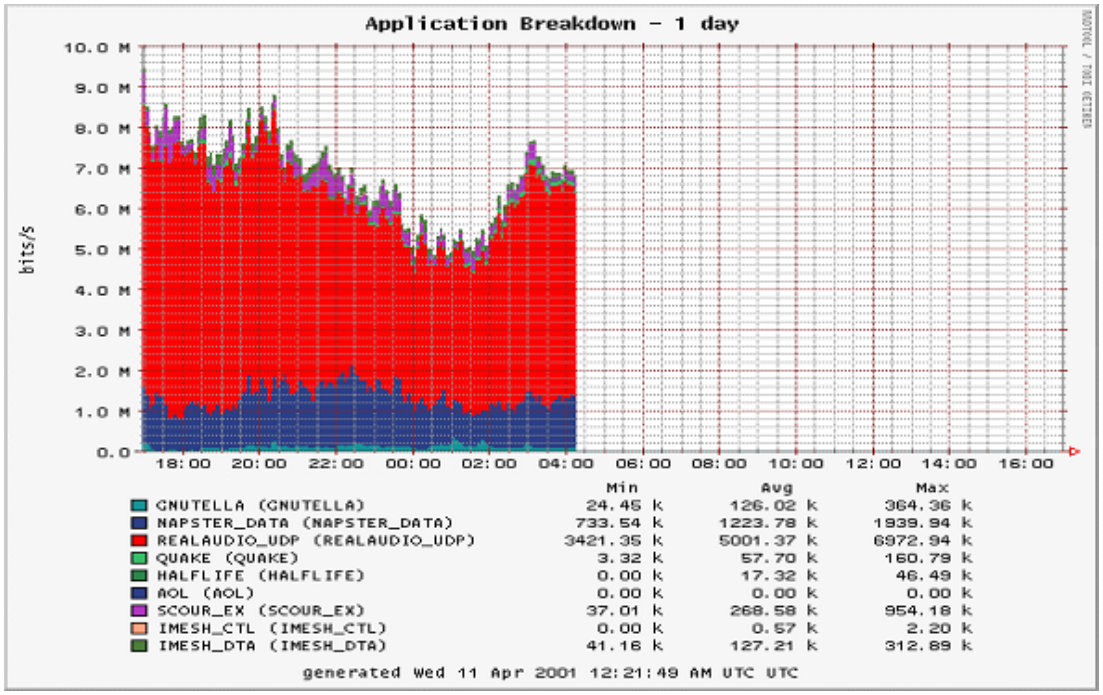

FIG. 3.2 – Applications émergentes en 2000

trafic UDP pour les observations France Télécom. Cependant, les observations MCI reportent des contributions sensiblement plus fortes : environ 30 et 20% du trafic UDP en paquets et en octets, (soit respectivement 3 et 1% du trafic total).

Très récemment (depuis fin 1999 et 2000), les applications orientées “streaming”, comme Mediaplayer de Microsoft, ont connu une croissance importante. Mediaplayer peut au choix de l’utilisateur fonctionner sur TCP, UDP et même RTP. Les mesures actuelles sur certains liens (Figure 3.2) montrent que dans sa version UDP, l’outil Mediaplayer a un certain succès.

Il faut signaler, car cela n’apparaît pas sur les résultats ici présentés, que les répartitions de trafic en termes d’applications sont globalement sensibles à la localisation géographique des points de mesure : ceci est en particulier vrai selon que l’on se place sur un réseau commercial d’opérateur ou sur un réseau local professionnel (de nombreuses publications portent sur des observations en entrée/sortie d’un LAN de campus universitaire ou de centre de R&D). Lorsque l’on se place sur un réseau d’accès, la répartition du trafic est aussi sensible au débit de connexion, comme le montre [92]. Mentionnons pour finir un phénomène très récent (2001), l’apparition d’une forte proportion (jusqu’à environ 30%) de trafic lié aux applications “Peer to Peer” (dont Napster était un précurseur), en particulier lorsque la mesure est effectuée sur un réseau d’accès haut débit (ADSL). La montée en puissance de ce type de trafic présente de plus la particularité de symétriser les flux de trafic par l’effet d’une augmentation sensible du trafic montant, de l’utilisateur vers le réseau.

3.3 Modélisation des processus

3.3.1 Introduction sur l'auto-similarité

Un trait important concernant la nature du trafic Internet, ou du trafic de données en général, est son aspect auto-similaire (ou possédant des dépendances à long terme, ou encore de caractère fractal, sans trop entrer dans les subtilités mathématiques qui différencient ses trois notions voisines). L'auto-similarité du trafic de données a été mise en évidence dans [50] et souvent confirmée depuis : la structure des variations d'amplitude du signal analysé (par exemple le nombre d'octets transférés par unité de temps) se reproduit de manière similaire quelle que soit la finesse temporelle avec laquelle il est représenté. Le comportement d'un trafic auto-similaire est à l'opposé de celui d'un trafic poissonnien, dont les variations d'amplitude sont filtrées au fur et à mesure que l'on augmente la taille de la fenêtre d'intégration.

Une monographie entière [65] récemment publiée est consacrée à la modélisation du phénomène d'auto-similarité du trafic dans les réseaux de données et à son impact sur l'évaluation des performances. Y figurent aussi bien des chapitres sur la description et la simulation des caractéristiques auto-similaires du trafic (streaming video, connections TCP, ...) et sur leur analyse au moyen de techniques sophistiquées telles que la transformée en ondelettes, que sur leur impact sur les modèles de performance basés sur les files d'attente et sur l'ingénierie des réseaux multi-service offrant de la Qualité de Service (dimensionnement, contrôle de congestion, etc.). Ce livre constitue une excellente et très complète source d'informations et de références sur ce thème ; nous donnons explicitement certains de ses chapitres dans la bibliographie figurant en fin de document. On pourra consulter aussi [96] pour une bibliographie exhaustive (à la date de parution) sur le sujet.

Les chercheurs d'AT&T ont publié de nombreux travaux sur l'auto-similarité du trafic [50, 97] ; mais les articles [29] et [30] sont particulièrement intéressants en ce sens que, outre la présentation de nombreux résultats mettant en avant ce phénomène, les auteurs insistent sur la nécessité de développer un modèle structurel d'analyse du trafic. Ce type d'approche permet de mieux cerner les aspects dynamiques (processus des connexions TCP au sein des sessions applicatives, processus paquets, ...) du trafic de données transporté par les réseaux longue distance (Wide Area Networks), et de fournir quelques interprétations "physiques" des phénomènes observés (relations entre auto-similarité et lois de probabilité à décroissance lente, par exemple). L'analyse des phénomènes liés aux échelles de temps est effectuée à l'aide d'une technique originale fondée sur la transformée en ondelettes [1] : les séries temporelles, telles que le nombre de paquets transmis par unité de temps p. ex., y sont analysées en termes d'un spectre d'énergie établi en fonction d'un facteur d'échelle temporelle.

A noter toutefois qu'un premier article [17], à l'origine de nombreux débats dans la communauté Internet et peut être à la base d'une nouvelle évolution dans l'approche choisie pour modéliser le trafic, défend une théorie opposée à celle de l'auto-similarité. En effet, l'étude présentée montre que plus la quantité de trafic augmente, plus le trafic devient régulier, avec des distributions d'arrivées exponentielles (et plus sous-exponentielles) et une dépendance qui diminue (évaluée à l'aide de méthodes entropiques). Le modèle de trafic ne serait ainsi plus auto-similaire à partir du moment où une certaine quantité importante de paquets et de flux s'entrelacent sur un lien haut débit.

3.3.2 Trafic au niveau paquets

Une caractéristique générale du trafic de paquets est de se présenter sous forme de rafales, ce qui est inhérent à ce mode de commutation (voir [75] et [86] et leurs nombreuses références). Il en résulte une extrême variabilité à toutes les échelles de temps des processus observés relatifs aux paquets : intervalle entre paquets successifs, nombre de paquets ou d'octets transférés par unité de temps, etc. Ceci se traduit mathématiquement par un comportement auto-similaire, voire multi-fractal selon les échelles de temps, des variables de trafic, un phénomène observé dans toutes les publications à ce sujet. Ce comportement se caractérise notamment par une décroissance lente, par exemple sous forme de loi puissance [36], de la fonction d'auto-corrélation du nombre de paquets, ou du nombre d'octets, transférés par unité de temps (typiquement 100 ms) : les processus auto-similaires sont des cas particuliers des processus à dépendance long terme (LRD).

Concernant le trafic de type streaming, les services liés à la vidéo sont appelés à un fort développement dans le futur : vidéoconférence, téléchargements en temps réel (Video on Demand). La caractéristique LRD a été identifiée dans de nombreuses publications concernant le transfert de séquences vidéo à débit variable (VBR selon la terminologie ATM) [10], probablement due à la variabilité des paramètres de transmission liés au codage des trames (MPEG, p. ex.), à la dynamique des images, etc. On pourra consulter [37] pour une revue complète sur les caractéristiques de ce trafic et sur leur impact sur les modèles de performance. Il y est en particulier noté que la LRD n'a pas forcément une influence déterminante, contrairement aux corrélations à court terme, sur le comportement des files d'attente des routeurs.

Dans le cas particulier de la téléphonie sur IP, on pourrait supposer que les paquets sont de longueur et d'intervalle inter-arrivées constantes au sein d'une même communication. Ces paramètres fixes sont le reflet, après encodage et compression à la source, du débit constant des communications téléphoniques. Cependant, ces hypothèses semblent quelque peu simplificatrices, d'une part à cause du phénomène de gigue de transfert si l'on se place sur un lien éloigné du routeur origine, d'autre part en raison de la diversité potentielle des paramètres de codage de la voix. Quoi qu'il en soit, et il en est de même pour le trafic vidéo, nous ne disposons pratiquement pas à l'heure actuelle d'observations concrètes de ce type de trafic sur réseau réel.

Concernant le trafic de type élastique, l'identification du processus des paquets tel qu'il est offert au réseau est particulièrement délicate. En effet, les dispositifs de correction d'erreur et de perte génèrent la retransmission de paquets supplémentaires et les mécanismes de contrôle de flux (TCP notamment) régulent les débits de transmission [13]. Les analyses de trafic doivent donc se contenter des données de trafic effectivement mesurées sur des liens, compte tenu de ces retransmissions et régulations.

Le caractère auto-similaire du trafic TCP a été largement étudié. En complément de ce qui a été dit au paragraphe précédent, notons les tentatives d'explication avancées : aux échelles de temps supérieures à un délai de transmission typique (RTT, "Round Trip Time", de l'ordre de 100 ms), le comportement auto-similaire (ou mono-fractal) serait dû à l'extrême variabilité de la taille des documents transférés (la loi de distribution est de type "heavy-tailed", telle la loi de Pareto), voir le paragraphe 3.3.3 sur la caractérisation des flots ; tandis que les caractéristiques multi-fractales aux échelles de temps inférieures seraient provoquées par les mécanismes de contrôle de congestion du protocole TCP [29, 30]. Effectuant une analyse des phénomènes d'échelle segmentée par composante de trafic (protocole de transport ou application), [55] montre que HTTP et FTP sont les

principales applications contribuant, de par leur extrême variabilité, aux propriétés de LRD détectées au niveau des paquets IP.

Tout ce qui précède concerne le processus d'arrivée des paquets. La loi de distribution des tailles de paquet, quant à elle, est difficilement modélisable statistiquement car elle possède de fortes composantes discrètes correspondant à certaines tailles caractéristiques (messages de gestion, accusés de réception, etc.) [85]. La distribution des tailles de paquet peut bien sûr varier selon le lieu et l'heure d'observation, mais elle est surtout sensible généralement au sens de transmission du trafic : la plupart des liens IP présentent une dissymétrie de répartition serveurs/usagers entre les deux nœuds qu'il relie (c'est le cas p. ex. de liens internationaux Europe - Etats-Unis, de liens de collecte du trafic grand public vers le cur de réseau, etc.). Le sens serveur vers usager a alors tendance à comporter des paquets de plus grande taille, liés au transfert des données demandées, par rapport au sens usager vers serveur, où le trafic est essentiellement composé de requêtes et d'informations protocolaires relatives à l'établissement des communications (accusés de réception, début et fin de connexion TCP, ...).

On observe ainsi trois modes déterministes principaux, qui se superposent bien entendu à une certaine distribution continue de poids assez variable. Les paquets de faible taille (40 octets ou guère plus), essentiellement composés des en-têtes TCP (ou UDP) et IP, sont produits par les protocoles et par de courtes requêtes (telles les commandes de Telnet générant des paquets de 1 caractère) ; ils constituent globalement presque la moitié des paquets observés, et jusqu'à 70 ou 80% des paquets dans le sens usager vers serveur. Les paquets de taille moyenne (typiquement 552 ou 576 octets) sont produits par certaines implémentations de TCP. Les paquets de grande taille (1500 octets) correspondent à la taille maximum (MTU) autorisée par la couche Ethernet et généralement reconnue par les protocoles de transport. Ces deux derniers modes comptent chacun pour environ 10 à 20% de la totalité des paquets.

3.3.3 Trafic au niveau flots

Notion de flot

En particulier pour les études liées à l'ingénierie de trafic, on s'intéresse tout particulièrement au niveau des flots [63]. Défini pour l'instant de manière assez floue comme une succession de paquets engendrés par une même instance d'une application donnée, et présentant donc une caractéristique de "cohérence temporelle", un flot de paquets correspond au transfert d'un message SMTP, d'un fichier au sein d'une session FTP, d'un objet (texte, image, son, ...) au sein d'une page HTML, éventuellement à une phase active d'une communication audio. Cette notion est donc très proche de la notion de documents dont la performance en termes de transfert forme le principal critère de QoS vue par les utilisateurs. Ceci constitue le principal argument conduisant à considérer le niveau flots pour proposer des modèles de dimensionnement, notamment s'agissant du trafic élastique.

Pour être plus précis, référons-nous en premier lieu à l'une des principales études menées de manière extensive sur cette notion [22] : on définit un **flot** comme un ensemble de paquets IP répondant à une même "**spécification de flots**" et se succédant les uns les autres à un intervalle de temps inférieur ou égal à un seuil donné que l'on nomme "**Time Out**" (**TO**). Cette notion de Time Out permet de garantir une certaine cohérence temporelle à la suite de paquets identifiée par une spécification de flots, c'est-à-dire que les flots de paquets ne présentent pas de trou trop important dans leur processus d'arrivées. La cohérence temporelle est essentielle à maintenir pour la modélisation

des performances du trafic sous TCP. Par ailleurs, l'introduction d'un TO sert aussi pour décider qu'un flot est terminé ou non ; c'est un élément indispensable si l'on envisage l'implémentation de traitements par flots dans les routeurs du réseau : le dépassement de TO sans nouvelle arrivée de paquet lié à une spécification de flot active permet de déclarer le flot terminé et de le supprimer de la table d'états du routeur. Dans ces conditions, l'estimation d'un TO optimal revêt une importance primordiale afin d'économiser les ressources en mémoire et en temps CPU des machines.

La spécification d'un TO est loin d'être triviale, elle peut être : (i) différenciée selon le type de protocole ou d'application ; (ii) adaptative en fonction du débit des flux [76] ; (iii) motivée par des préoccupations diverses (pour ne pas dire divergentes) telles que l'élaboration de modèles de performance du trafic, la mise en œuvre de disciplines de service par flot ou de politique de routage dans les nœ uds du réseau ; (iv) etc. Sans trop entrer ici dans les détails, mentionnons que la spécification de flots peut comprendre quatre dimensions [22] : la directionalité des flots (mono- ou bi-directionalité) ; la prise en compte d'une ou deux des extrémités (origine ou destination des paquets) ; la granularité des extrémités (de la plus fine, les applications, à la plus grossière, les sous-réseaux) ; enfin, le protocole de la couche transport.

Dans la suite, on associera implicitement la notion de flot à celle de **micro-flot**, en considérant que la spécification de flots est déterminée par la donnée du **5-uple {*adresse IP source, adresse IP destination, port source, port destination, protocole de transport*}**. (A noter que les numéros de port identifiant les applications n'existent que dans le cas de transport sous les protocoles TCP et UDP.) Sur le lien considéré et pour un intervalle d'observation donné, on appellera "**flux**" tout ensemble de paquets conformes à un même 5-uple de spécification de flots. Un flux est donc éventuellement constitué d'un ou plusieurs (micro-) flots successifs séparés d'au moins un intervalle de temps égal à TO.

Insistons, pour terminer, sur le fait que les flots ainsi définis ne correspondent pas forcément à des connexions dans le cas de flux TCP. Certains auteurs préconisent en effet d'utiliser les informations SYN/FIN de début et de fin de connexion, présentes dans les en-têtes TCP, pour identifier les flots [28]. Non seulement ces informations TCP peuvent ne pas être toujours présentes en un point donné du réseau (pour cause de routes différentes suivies par les paquets, p. ex.), ce qui perturbe l'identification des connexions, mais de plus, de telles connexions TCP sont susceptibles de comporter plusieurs flots au sens que nous venons de définir : une session de connexion à distance sous Telnet, ou bien une session de navigation Web sous protocole HTTP 1.1 à connexions persistentes [43], peut éventuellement comporter le transfert de plusieurs fichiers successifs bien délimités dans le temps.

Caractérisation des flots

De nombreuses études ont été publiées récemment sur la modélisation du trafic au niveau des flots, motivées par les nombreuses perspectives d'application offertes par la considération de cette entité de trafic. En premier lieu, comme souligné plus haut, l'analyse des performances du trafic et l'inférence qui en découle de méthodes de dimensionnement des ressources s'effectuent plus aisément, et de manière plus adéquate, à ce niveau. Par ailleurs, les différents schémas d'architectures de routage, de "traffic engineering" (MPLS, routage orienté QoS, etc.), voire de fourniture de services différenciés (IntServ, DiffServ), qui ont été proposés ces dernières années de manière quelque peu foisonnante dans la mouvance de l'IETF, prennent en compte également la notion de flots, éventuellement selon des niveaux d'agrégation très variables.

Une comparaison détaillée est malaisée à effectuer entre les différents résultats produits, en raison des définitions diverses accordées à la notion de flots : micro-flots selon la définition ci-dessus, connexions TCP, documents Web, etc.). Cependant, quelques points communs importants peuvent être dégagés dans l'ensemble : la non conformité du processus des arrivées de flot à un processus de Poisson, et le comportement général de décroissance lente des distributions statistiques de longueur de flot. A noter que la plupart des publications sur ce sujet ne traitent que des caractéristiques de flots TCP, à l'exception de [63] où l'on montre que les modèles de représentation des processus de flots sont similaires pour UDP et TCP, notamment pour ce qui concerne le processus d'arrivée. Cependant, ce dernier résultat est à tempérer par le fait que très peu de véritables flux de trafic temps-réel circulent sur les réseaux IP, du moins jusqu'à très récemment.

Tous les résultats statistiques produits dans la littérature mettent en évidence des lois de distribution à décroissance lente ("heavy-tailed") dès que l'on s'intéresse à un paramètre lié à la taille, au volume, à la durée, ..., d'un objet à transférer sur un réseau de données (voir p. ex. [14, 21, 26, 51, 60, 63, 67] pour le trafic IP). Ce phénomène de décroissance lente signifie que la probabilité d'obtenir des très grandes valeurs de la variable aléatoire est asymptotiquement beaucoup moins faible que pour une loi de type exponentiel. Les lois de probabilité couramment utilisées pour modéliser un tel comportement sont la loi de Pareto et la loi log-normale. Si l'on entend littéralement par "décroissance lente" une décroissance équivalente (selon une définition mathématique que l'on ne précisera pas ici) à une fonction puissance, la loi log-normale n'est pas une loi à décroissance lente [70], elle est seulement à décroissance sous-exponentielle. De même, la loi de Weibull est souvent assimilée à une loi à décroissance lente alors qu'il s'agit d'une généralisation (de même que la loi Gamma) de la loi exponentielle ; elle peut bien sûr être à décroissance plus lente que l'exponentielle en fonction des valeurs prises par l'un de ses paramètres.

La loi de Pareto est l'archétype de la loi à décroissance lente puisque sa distribution (cumulative complémentaire) est proportionnelle à $x^{-\alpha}$ avec $\alpha > 0$. Pour $1 < \alpha \leq 2$, la valeur moyenne est finie, mais la variance ne l'est pas. Pour $0 < \alpha \leq 1$, la valeur moyenne elle-même devient infinie. Ce comportement est significatif d'une très grande variabilité de la variable aléatoire considérée : dans le cas de la taille des documents des sessions Web, de très nombreux documents de taille modeste coexistent avec des documents très volumineux, en nombre plus faible mais non négligeable. C'est cette extrême variabilité de la taille des documents qui semble être à l'origine des phénomènes de dépendance à long terme observés aux grandes échelles de temps (typiquement supérieures à 1 s), du moins à propos du trafic de type Web ou de transfert de fichiers [24, 64].

Dans la plupart des cas, seule la queue de distribution de la loi des longueurs de flot est identifiée comme étant à décroissance lente ; la loi de Pareto ne fournit en général pas un bon modèle de représentation de l'ensemble de la distribution, le corps de la distribution étant par exemple bien modélisé par une loi log-normale [60, 63]. Circonstance qui pourrait inciter à proposer de modéliser l'ensemble de la distribution par des mélanges de loi (log-normal/Pareto [43] ou mélange de plusieurs lois log-normales [14]), quoi que l'on puisse douter de l'intérêt pratique de disposer de tels modèles de représentation, certainement peu utilisables analytiquement et peu économes en nombre de paramètres, pour mener des études ultérieures sur la performance du trafic. A l'opposé de ce caractère généralement multi-modal des distributions, on observe dans [63] un ajustement remarquablement bon (du moins visuellement, mais néanmoins selon deux modes de représentation, en échelle linéaire et en échelle logarithmique) de l'ensemble de la distribution par une loi de Pareto dès lors que l'on traite des longueurs de flots TCP mesurées en nombre de paquets, et ce à partir de diverses campagnes d'observation, aussi bien sur un réseau dorsal que sur un réseau d'accès haut

débit ADSL. Dans tous les cas où il est estimé, le paramètre de puissance α de la loi de Pareto se situe entre 1 et 2, voire proche de 1, signe d'extrême variabilité comme souligné plus haut.

Les travaux publiés sont un peu moins nombreux concernant la caractérisation du processus d'arrivée des flots, malgré le fait que la plupart des modèles de performance au niveau flot supposent qu'il s'agit d'un processus de Poisson et demandent donc à être validés. Dans le cadre d'un modèle hiérarchique sessions/flots [15] où les flots correspondent à des demandes de transfert successives de documents (pages Web) ou de fractions de documents en parallèle ou en série (fichiers ou objets d'une page Web) au sein d'une même session d'un utilisateur, il est logique de s'attendre à des inter-dépendances dans le processus temporel d'apparition des flots, lequel devra alors s'écarter sensiblement du modèle Poissonnien. C'est ce que l'on vérifie expérimentalement, à l'exception de l'étude commentée au paragraphe suivant. La distribution statistique des intervalles inter-arrivée de flots est remarquablement bien représentée (y compris au vu de tests numériques d'ajustement) par une loi de Weibull [28, 63] ou par une loi Gamma (toutes deux des extensions à deux paramètres de la distribution exponentielle, mais la dernière étant moins usitée dans ce domaine). Dans la seconde de ces deux études, les flots sont des micro-flots TCP ou UDP, tandis que dans la première, ce sont des connexions TCP (identifiées par les informations de début et de fin fournies par ce protocole) ; on en déduit une certaine robustesse des résultats obtenus. Pour compléter l'étude au second ordre, des estimations de la fonction d'auto-corrélation de l'intervalle inter-arrivée mettent en évidence une corrélation persistante dans le temps (sur une ou quelques dizaines de seconde), quoi que de niveau assez faible (de l'ordre de 0,1). Poussant plus loin le caractère non Markovien du processus d'arrivée des flots, [28] met en évidence une composante de LRD concernant le nombre d'arrivées de connexions TCP par unité de temps, sur toutes les échelles de temps au delà de 1 s. Les auteurs notent par ailleurs le fait intéressant que ce phénomène est apparu graduellement au cours de ces dernières années, au fur et à mesure de l'importance croissante du trafic Web (protocole HTTP).

Se différenciant des résultats précédents, l'étude [60] montre que la loi log-normale fournit (de même que pour la statistique des longueurs) la meilleure représentation de la distribution des intervalles inter-arrivées de documents si l'on traite l'ensemble des données recueillies sur plusieurs jours consécutifs (typiquement une semaine). En se limitant à des période chargées de 2 heures où les composantes non stationnaires (profils journaliers déterministes) sont moins présentes, la meilleure représentation est alors fournie par la loi exponentielle. Ce résultat est surprenant, d'autant qu'il repose sur une analyse descriptive très complète des statistiques du trafic Web (jeux de données sur plusieurs semaines, en provenance de 4 sites différents). Il peut néanmoins s'expliquer par différents facteurs : en premier lieu, les flots sont en fait des requêtes adressées vers des sites Web ; ensuite, parmi les lois testées, seule la loi exponentielle n'est pas à caractère de décroissance sous-exponentielle.

En tout état de cause, il semble que les modèles de performance du trafic IP doivent tenir compte du caractère non Poissonnien sensiblement marqué du processus d'arrivée des flots, soit en intégrant explicitement, de manière exogène, les caractéristiques de représentation du premier ordre, voire du second, soit en développant une approche de plus haut niveau prenant en compte la structuration hiérarchique des flots en sessions comme le propose [15].

3.3.4 Trafic au niveau sessions

Le processus des demandes de communication, qu'elles soient de type streaming ou de type élastique, a toutes les raisons de pouvoir être considéré comme Poissonnien (dans la mesure où

l'on peut admettre que la population source est de taille quasi-infinie). C'est l'un des principaux invariants communément reconnus en modélisation du trafic Internet [32]. En effet, au niveau des sessions utilisateurs, le processus d'arrivée résulte de la superposition d'un nombre élevé de demandes élémentaires indépendantes entre elles (à opposer à la dépendance inter-flot au sein d'une même session).

Des campagnes d'observation remontant au début de la précédente décennie, époque où le trafic Internet était essentiellement dominé par les applications FTP, Telnet ou SMTP, ont montré que les arrivées de session obéissaient correctement à un processus de Poisson [70], bien que ces diverses applications aient des modes de fonctionnement très différents (au niveau des connexions TCP générées par exemple). Dans le but de caractériser les sessions Web, des observations indirectes ont été effectuées [29] par la collecte d'informations relatives aux appels par modem à destination d'un ISP : instants d'arrivée, taille et durée des appels. Bien que ces demandes de connexion ne contiennent pas uniquement des sessions Web, ces données ont été utilisées pour tenter de caractériser les processus liés à ces dernières. L'analyse, uniquement qualitative, montre que le processus d'arrivées est cohérent avec un processus poissonnien, du moins avec un processus de renouvellement (processus pour lequel les intervalles inter-arrivées sont indépendamment et identiquement distribués). En fait, peu de travaux ont eu pour objet de valider rigoureusement le caractère poissonnien du processus d'arrivée des sessions, probablement en raison du caractère naturel, presque évident, de l'hypothèse, mais aussi certainement à cause de la difficulté d'identifier des sessions générées par les utilisateurs à partir de traces réelles. Citons seulement l'article récent [62] qui obtient sur des données collectées aux Bell Labs (donc sur un réseau local) un excellent comportement, aux premier et second ordre, du modèle de représentation des arrivées de session par un processus de Poisson.

La loi des durées (ou des longueurs), quant à elle, possède des caractéristiques de distribution à décroissance lente, quoiqu'elle soit difficile à identifier comme indiqué plus haut. Que ce soit dans [70] ou dans [29], donc avec ou sans présence prépondérante des sessions Web, la loi des durées de session (ou de taille exprimée en koctets) possède une queue de distribution caractéristique d'une loi de Pareto de moyenne finie, mais de variance infinie, significative d'une grande variabilité. D'un point de vue qualitatif, les paramètres quantitatifs des modèles de représentation étant bien entendu différents, les caractéristiques statistiques des durées de session sont dans l'ensemble similaires à celles des longueurs de flot.

3.4 Représentation du trafic

Un système de représentation du trafic similaire à celui utilisé pour le trafic téléphonique n'existe pas pour le trafic IP. La principale raison en est l'exceptionnelle croissance de ce trafic, qui se prête plus à une gestion dynamique et réactive du réseau qu'à une planification raisonnée de ses ressources et de leur dimensionnement, laquelle suppose, sinon une situation d'équilibre, du moins une croissance modérée de la demande en trafic. On peut cependant essayer de transposer au trafic IP les notions de représentation du trafic, au moins pour les courtes échelles de temps, afin, d'une part, de mieux comprendre la variabilité du trafic, et, d'autre part, de tenter de rationaliser sa gestion à court terme.

Au niveau de la normalisation internationale, les principes régissant la mesure du trafic sont regroupés dans la recommandation E.500 de l'UIT-T [88] *a priori* pour tout type de trafic, aussi bien le trafic des réseaux à commutation de circuits que celui des réseaux de paquets, à bande étroite

tel le réseau de signalisation par canal sémaphore, ou à large bande tels que les réseaux multi-service IP (bien que non mentionné explicitement) ou ATM. Cependant, ce document ne donne que peu d'informations numériques, notamment au sujet de la période de référence pour estimer un trafic de pointe journalier. Ces principes de mesure doivent être ultérieurement traduits de manière quantitative pour chaque type de trafic ou (et) chaque partie de réseau. Seule la recommandation E.492 [87] indique (très succinctement) que la période d'intégration pour calculer une intensité de trafic du RNIS - Large Bande devrait être d'une durée comprise entre 5 et 15 mn, comme préconisé pour le trafic de signalisation CCITT No 7.

3.4.1 Définitions liées au trafic de pointe

En premier lieu, définissons la **durée d'intégration** du trafic (ou temps ou intervalle d'intégration) comme la durée recommandée sur laquelle on peut estimer de manière fiable une intensité moyenne de trafic (volume total de trafic divisé par la durée de l'intervalle considéré, d'où le terme "intégration"). Par estimation **fiable**, on entend estimation **précise** (intervalle de confiance réduit) et estimation **significative** (valeur moyenne stable, ou trafic stationnaire). Nous reviendrons ci-dessous sur ces deux notions.

L'UIT [88] propose deux méthodes d'extraction d'une valeur de pointe du trafic journalier :

- La **période de pointe journalière** (DPP, "Daily Peak Period"), définie comme la période d'intégration d'intensité de trafic maximale détectée au cours d'une observation continue sur la journée par périodes d'intégration consécutives. La valeur de pointe que l'on conserve du trafic journalier est bien entendu l'intensité moyenne de trafic correspondant à cette DPP. C'est la méthode recommandée préférentiellement par l'UIT-T dans le cas général.
- L'**intervalle fixe de mesure journalière** (FDMI, "Fixed Daily Measurement Interval"), qui suit le même principe que DPP à la différence, notable, que la période d'observation du trafic se limite à un intervalle de temps fixe et prédéterminé, dont on sait préalablement qu'il contient les périodes chargées, composé de plusieurs périodes d'intégration successives. On peut se contenter de ce genre d'observations non continues lorsque les profils de trafic journaliers sont suffisamment prévisibles (à vérifier périodiquement par des mesures complémentaires).

3.4.2 Intensité de trafic et stationnarité

La mesure de l'intensité moyenne de trafic, sur une période d'intégration, doit se faire au sein d'un intervalle d'observation où le trafic est stationnaire [16]. Si l'on se réfère aux conclusions de [99], on peut compter sur un ordre de grandeur de plusieurs minutes durant lequel les propriétés du trafic Internet restent approximativement stables. Dans les situations où le temps moyen de service est très petit devant la durée d'intégration du trafic, il suffit que le processus d'arrivée des unités de trafic soit stationnaire. L'intensité de trafic peut alors être estimée par la mesure du taux d'arrivée moyen λ des unités, du temps moyen de service h et du nombre moyen n de ressources requises pour le traitement d'une unité, *via* la formule de Little : $A = \lambda nh$ (le nombre de ressources est souvent égal à 1).

Dans le cas des réseaux de données à haut débit, le temps de transfert d'une unité étant proportionnel à sa longueur, on exprimera couramment, par abus de langage, l'intensité de trafic en termes de débit d'information (multiple de bits/s, nb. de paquets/s). De plus, le temps de transfert d'un pa-

quet ou d'une cellule est de l'ordre de la microseconde ou de la milliseconde. Il n'y a donc pas lieu de distinguer instant d'arrivée et instant de départ des unités, notamment pour les problèmes liés à la stationnarité, tant que l'on ne considère que des durées d'intégration supérieures à la seconde, typiquement, ce qui constitue déjà une borne inférieure plus que raisonnable.

3.4.3 Durée d'intégration et précision de la mesure

La durée d'intégration pour estimer une intensité de trafic doit ensuite être suffisamment longue pour garantir une convergence correcte de la mesure vers l'intensité "réelle". La précision de l'estimation augmente en effet avec la taille des échantillons, sous réserve de stationnarité. La détermination d'une borne inférieure de la durée d'intégration, assurant une précision suffisante, dépend du modèle de trafic sous-jacent. La recommandation E.500 [88] rappelle l'exemple classique d'un trafic de Poisson pour lequel cette borne inférieure s'exprime simplement, quelle que soit l'intensité du trafic, en termes d'un nombre minimal d'échantillons à collecter. Celui-ci doit être d'environ $(1,96/\alpha)^2$, où α est une spécification en valeur relative de l'intervalle de confiance $[A - \alpha A; A + \alpha A]$, et pour un niveau de confiance de 95%. Ce résultat suppose en particulier que l'on a approximé par une loi Gaussienne la loi de distribution du volume de trafic observé. Pour un coefficient α de 0,1 (c'est-à-dire une précision assez médiocre), la durée d'intégration doit comporter au moins 384 arrivées d'unités (appels p. ex. puisqu'il s'agit d'un trafic poissonnien).

Pour analyser la précision de mesure de l'intensité du trafic, il est nécessaire d'utiliser un modèle de représentation de la variabilité du trafic. Le trafic IP est connu pour présenter un caractère d'extrême variabilité, associé au phénomène d'auto-similarité ou plus généralement de dépendance à long terme (voir 3.3.1 ci-dessus). La conséquence en est que la variance du trafic agrégé sur un certain intervalle de temps décroît moins vite, en fonction de la durée de cet intervalle, que celle d'un processus de Poisson tel que le trafic téléphonique par exemple. Sur une plage d'intervalles de temps où le trafic peut être considéré comme mono-fractal, ce qui a été généralement observé aux échelles de temps supérieures à 1 s (échelles qui nous importent ici) [24, 30, 50, 97], la variance décroît sous forme d'une fonction puissance dont l'exposant est simplement relié au paramètre de Hurst H. La connaissance de H, pour un certain type de trafic et sur un lien donné, permet alors d'estimer de manière empirique la précision que l'on obtiendra sur l'intensité moyenne de trafic calculée avec un pas d'intégration donné. On pourra admettre par exemple que la précision relative sur l'intensité estimée est donnée par le coefficient de variation, rapport de l'écart-type sur la moyenne. Par ailleurs, le paramètre de Hurst n'ayant évidemment pas une valeur universelle, sa détermination devra être effectuée par l'analyse de campagnes de mesure menées au préalable.

3.4.4 Valeurs de référence du trafic

L'UIT-T [88] définit les notions de charge normale et de charge élevée de trafic sur une période de 1 mois. Ce moyen terme paraît être un bon compromis pour s'affranchir de variations trop fortes, dues à la croissance du trafic et aux fluctuations saisonnières, et autoriser un minimum de robustesse aux traitements statistiques sur les valeurs de pointe journalières. En revanche, la correspondance entre ces niveaux de charge et le degré de QoS produit par les conditions d'exploitation du réseau n'est pas établie de manière très convaincante. Ce qui suit peut se décliner sur plusieurs plages horaires au cours de la journée, définies généralement en fonction de politiques tarifaires différenciées et éventuellement associées à des objectifs variés de Qualité de Service.

La **charge normale** de trafic est recommandée comme étant la $4^{ème}$ plus forte parmi les valeurs de pointe journalières du mois ; la **charge élevée** de trafic comme la $2^{ème}$ plus forte. De l'ensemble des jours retenus préalablement à cette détermination des niveaux de charge, on pourra exclure raisonnablement les jours exceptionnels tels que le Jour de l'An, la Fête des mères, etc., pour lesquels des mesures spéciales sont généralement prises. Par contre, il ne sera pas nécessaire de différencier les jours ouvrables des jours de week-end (ou autres jours non travaillés) dans la mesure où, si l'un de ces groupes de jours présente un trafic journalier statistiquement plus faible, il s'éliminera de lui même de la procédure de sélection des jours de charge normale et élevée.

Pour servir de base au dimensionnement des ressources d'un réseau, l'idéal serait de se fonder sur un comportement statistique des valeurs de charge de référence (normale ou élevée) de manière à déterminer des objectifs de dimensionnement en termes de probabilité de satisfaction d'indicateurs de Qualité d'Ecoulement du Trafic (GOS, "Grade Of Service"). C'est une manière plus économique de concevoir le dimensionnement que de viser avec certitude les niveaux de performance souhaités. Cependant, cette méthode se heurte à des difficultés statistiques dues au faible nombre d'échantillons et à l'inhomogénéité des profils de trafic mensuels (fluctuations saisonnières).

L'UIT-T se reporte donc de manière concrète sur la notion de valeur représentative annuelle. Il est proposé de définir cette dernière comme la $2^{ème}$ plus forte ou la plus forte des valeurs de référence mensuelles (charge normale ou élevée), respectivement selon que le trafic est homogène ou non d'un mois sur l'autre. La valeur de trafic finalement retenue pour le dimensionnement sera obtenue à partir de la valeur représentative annuelle par application d'un facteur de croissance (donné par les méthodes de prévision du trafic) et d'une marge de sécurité palliant les incertitudes de la prévision.

Soulignons pour terminer que l'attention doit être portée, dans un réseau multiservice, sur la nécessité d'effectuer des estimations de charge par type de service, dans la mesure où ceux-ci peuvent présenter des périodes critiques ou des caractéristiques de croissance différentes.

3.5 Inférence des matrices de trafic IP

La matrice de trafic exprime la demande de trafic entre toutes les paires origine-destination du réseau. La répartition du trafic sur les liens du réseau est ensuite determinée par les protocoles de routage. Dans un réseau IP, le chemin qu'emprunte une demande n'est pas dynamique. Il est donc très important de bien définir le routage pour être sûr que la capacité du réseau est suffisante pour acheminer la demande de trafic. L'efficacité de ces opérations et des autres méthodes d'ingénierie de trafic dépend fortement de la connaissance et de la précision sur la matrice de trafic.

Le problème est qu'il est difficile d'obtenir des matrices de trafic pour des réseaux IP et particulièrement lorsque ceux ci sont très maillés. Une première difficulté provient de l'absence de correspondance entre adresses IP et localisation géographique. Pour obtenir la route d'un paquet IP, il est nécessaire de remonter aux protocoles de routage internes et externes du réseau à l'instant de la mesure. Il apparaît également qu'il est difficile d'avoir des mesures de trafic directes et complètes ne serait ce que pour des raisons de complexité, de performance et de coût. De plus, la matrice de trafic peut changer fréquemment dans un réseau IP à cause de plusieurs facteurs : croissance rapide du trafic Internet, changement des sources ou puits de trafic ou même changements fréquents de routage inter-domaines (BGP, "Border Gateway Protocol"). Il peut donc être utile de développer des méthodes d'inférence à partir d'informations incomplètes : charges de trafic sur les liens du réseau et estimations initiales de la matrice. Il est intéressant de noter que des problèmes similaires se

posent dans le cadre des réseaux de transport routier, domaine duquel nous avons tiré de nombreuses références bibliographiques.

L'objectif de cette section est d'identifier les difficultés rencontrées dans l'inférence des matrices de trafic IP et de présenter les méthodes adoptées par certains opérateurs ainsi que les techniques d'inférence qui permettent l'obtention d'une matrice de trafic à partir de données incomplètes. Nous verrons que les matrices de trafic obtenues ne sont jamais très précises. Ceci est dû aux difficultés de mesure et aux instabilités du trafic IP : variabilité des protocoles de routage, changements des sources de trafic. Cette variabilité qui ne peut pas toujours être controlée par les opérateurs doit être prise en compte en développant des méthodes d'ingénierie de trafic plus robustes.

3.5.1 Caractéristiques du trafic

Des résultats récents sur la modélisation du trafic IP [15] ont montré qu'il suffit, du moins pour le trafic élastique, de connaître le trafic offert moyen. Une matrice de trafic peut alors être définie en utilisant les valeurs représentatives de trafic définies dans le paragraphe 3.4. La matrice donne alors la valeur représenative de la demande de trafic point à point dans le réseau, i.e. le trafic entre toutes les paires origine-destination du réseau. Cependant, la spécificité du routage IP fait qu'il peut être intéressant d'avoir plus de détails au sens où le trafic IP n'est pas forcèment défini par un seul point d'entrée et un seul point de sortie. La multiplication des points d'interconnexion entre opérateurs fait qu'une demande peut avoir plusieurs points d'entrée ou points de sortie possibles. La modification des tables de routage internes du réseau ou de celles du réseau voisin peuvent alors complétement changer la matrice de trafic. Le trafic est donc mieux caractérisé par des demandes point à multi-point ou point à point. Cet aspect peut avoit un impact non négligeable sur les réseaux de transit mais est difficile à mesurer.

Des variations fréquentes (à l'echelle d'une semaine ou d'un mois) peuvent donc survenir à cause de changements des points d'interconnexion entre opérateurs IP, des tables de routage ou plus simplement de la localisation des sources de trafic (nouveaux serveurs, événements saisonniers). La figure 3.3 montre un exemple de variation sur un lien de cœ ur de réseau sur une période de 1 mois. Cette variabilité de la matrice de trafic ainsi que la difficulté des mesures (comme nous le verrons dans le paragraphe 3.5.2) montre la nécessité d'avoir des méthodes d'estimation simples et rapides.

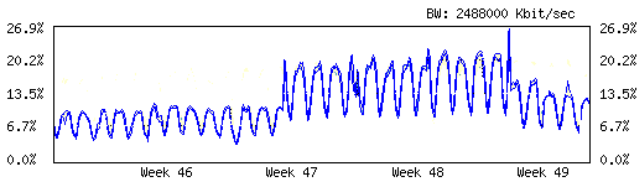

FIG. 3.3 – Profil de trafic mensuel sur un lien

3.5.2 Mesures directes

Une méthode d'estimation directe de la matrice de trafic consiste à observer les paquets à leur entrée dans le réseau et à déduire le nœ ud de sortie à partir de l'adresse IP destination. Cette opération n'est pas immédiate puisqu'il n'y a aucune correspondance entre l'adresse IP et la localisation géographique. Il est donc nécessaire d'examiner les protocoles de routage internes et externes pour déduire le point de sortie de ce paquet. Ceci nécessite donc deux opérations :

- Mesure du trafic pendant la période chargée ;
- Analyse des destinations du trafic mesuré aux points d'entrée à partir des tables de routage utilisées dans le réseau à l'instant de mesure.

Cette approche de mesure directe a été utilisée par Feldmann *et al.* [31] qui mesurent le trafic entrant et sortant sur les liens d'interconnexion du réseau en utilisant l'outil Netflow [61]. Netflow permet d'effectuer des mesures de trafic par flot sur les interfaces des routeurs. Le trafic ainsi collecté est exporté toutes les 15 minutes par datagrammes UDP vers un serveur central et est ensuite analysé en utilisant les tables de routage collectées sur les routeurs du réseau. L'utilisation de Netflow n'est cependant pas idéale : la quantité de données à exporter vers le collecteur représente une partie non négligeable de trafic : 95% des données ont ainsi été perdues à certains moments dans les expériences effectuées dans [31] ! De plus, Netflow peut affecter la performance des unités centrales de calcul des routeurs. Néanmoins, en utilisant cette méthode, les auteurs ont pu reconstituer les demandes de trafic point à multi-point ayant leur origine et leur destination à l'extérieur du réseau.

Une autre méthode possible est celle utilisée par Sridharan *et al.* dans une étude sur les stratégies de routage pour le backbone de Sprint [83]. Cette méthode n'est pas présentée comme une procédure opérationnelle pour la mesure des matrices de trafic mais mérite d'être citée. Les auteurs ont effectué des mesures détaillées sur le trafic entrant et sortant d'un nœ ud du réseau. Des équipements de mesure ont permis de capturer tous les détails des paquets IP observés sur ces interfaces. Un traitement a été ensuite appliqué sur les données obtenues pour reconstituer les flots de trafic à différents niveaux d'agrégation (différents masques sur les adresses IP). L'examen des tables de routage BGP a permis ensuite de déduire la quantité exacte de trafic entre ce nœ ud et les autres nœ uds du réseau, ce qui constitue donc une ligne et une colonne de la matrice de trafic. Pour obtenir le reste de la matrice, ils appliquent une méthode d'extrapolation qui tient compte de paramètres tels que le nombre de points de peering, le nombre de clients entreprises et le nombre de clients résidentiels connectés aux autres nœ uds du réseau. L'obtention d'une matrice exacte nécessiterait la répétition des mêmes mesures sur tous les nœ uds du réseau. Cette méthode apparaît donc comme trop complexe pour une utilisation opérationnelle régulière mais permet d'identifier des particularités intéressantes du trafic comme la grande variabilité des tailles des demandes point à point, résultat également observé dans [31].

Les méthodes précédentes requièrent donc la connaissance des tables de routage à l'instant des mesures et un traitement des données fonction de ces tables. Cet aspect peut constituer une contrainte supplémentaire dans la constitution des matrices de trafic. Duffield et Grossglauser [27] proposent une méthode qui n'utilise pas cette information. Cette approche repose sur l'échantillonnage des paquets et l'observation des trajectoires des paquets échantillonnés dans le réseau. L'échantillonnage permet de sélectionner une faible proportion de paquets à observer à l'aide d'une fonction de hash appliquée à l'entête TCP/IP du paquet. La même fonction de hash déterministe est utilisée dans tous les nœ uds du réseau, un paquet est donc observé sur tous les routeurs qu'il traverse. Les mesures sont ensuite envoyées à un collecteur qui reconstitue les demandes de trafic point à point ainsi que les

trajectoires empruntées dans le réseau. Le nombre de paquets échantillonnés est un compromis entre précision des mesures et quantité d'information à traiter et à véhiculer dans le réseau. Cette approche apparaît donc comme très intéressante mais nécessite néanmoins une implémentation spécifique sur les routeurs. La création d'un groupe de travail sur la question est en cours de discussion à l'IETF à l'initiative d'AT&T.

Il apparaît ainsi, de ce qui precède, que la constitution des matrices de trafic IP par des méthodes directes peut être complexe, des mesures exhaustives sur le réseau étant difficiles à réaliser.

3.5.3 Inférence à partir d'informations incomplètes

Puisque des mesures directes et complètes de la matrice de trafic sont difficiles à obtenir, on se propose de la déduire d'informations incomplètes. Dans un réseau IP, on dispose d'autres sources d'information : il est en effet courant de mesurer les charges de trafic sur les interfaces des routeurs *via* des outils comme SNMP (Simple Network Management Protocol). Ces dernières informations peuvent être utilisées pour l'inférence des matrices de trafic. Des études récentes sur les matrices de trafic IP se sont inspirées de méthodes d'inférence de matrice de trafic à partir de la charge sur les liens, méthodes développées dans le domaine des réseaux routiers. L'objectif est de trouver la "meilleure" matrice de trafic reproduisant les charges observées sur les liens. Diverses approches d'inférence à partir de la charge des liens existent. Nous détaillons dans la suite de ce paragraphe certaines de ces techniques qui peuvent être séparées en deux familles : approches déterministes et approches statistiques.

Inférence déterministe

L'hypothèse est que l'on connaît parfaitement le trafic sur chaque lien du réseau, le schéma de routage dans le réseau, et l'on cherche à estimer la matrice de trafic. Si L est le nombre de liens et R le nombre de routes, on pose y_l le volume de trafic sur le lien l, pour $1 \leq l \leq L$, et x_r le volume de trafic sur la route r, pour $1 \leq r \leq R$. Ici, la matrice de trafic est donc représentée sous la forme d'un vecteur $X = \{x_r\}$ et les volumes sur les liens sont définis par le vecteur $Y = \{y_l\}$.

Le routage dans le réseau est donné par la matrice $A = [a_{lr}]$ où a_{lr} représente la proportion de trafic de la route r qui utilise le lien l. Dans un réseau avec routage statique et non dynamique, les coefficients a_{lr} sont constants, ne dépendant pas de la charge de trafic, et sont compris entre 0 et 1. Ceci est le cas des réseaux IP courants où le routage est en général fixé avec possibilité de partage de charge des liens sur des chemins de même coût.

Chaque route r contribue avec le volume x_r sur les liens qui composent les chemins de la route r, ce qui se traduit par l'équation suivante :

$$\sum_{r=1}^{R} a_{lr} x_r = y_l, \quad 1 \leq l \leq L. \tag{3.1}$$

ou, de manière équivalente :

$$Y = AX \tag{3.2}$$

Puisque $L \ll R$, il existe un grand nombre de solutions $\{x_r\}$ pour un ensemble de charges $\{y_l\}$ donné. Pour déterminer la solution la plus probable, il est courant d'utiliser des informations

supplémentaires exprimées par une matrice (ou vecteur) de trafic initiale $\{\widetilde{x}_r\}$. Cette matrice peut être une matrice de trafic antérieure que l'on souhaite corriger ou une matrice obtenue en utilisant un modèle de trafic tenant compte de la population des utilisateurs. L'objectif est alors de minimiser une certaine distance entre la matrice recherchée $\{x_r\}$ et la matrice initiale $\{\widetilde{x}_r\}$, avec $\{x_r\}$ satisfaisant les contraintes exprimées par l'équation (3.1).

Ce problème a été traité en détail dans le contexte des réseaux de transport routier, cf. [7, 12, 90]. Les méthodes utilisées dépendent de la distance ou métrique qui est choisie. Une approche courante est de préserver l'information contenue dans la matrice $\{\widetilde{x}_r\}$ en maximisant l'entropie [90]. On cherche ainsi à obtenir la matrice $\{x_r\}$ qui ajoute le minimum d'information au contenu de la matrice de trafic initiale. Ceci conduit à considérer la distance suivante : $\sum(x_r(\log x_r/\widetilde{x}_r - 1)$. Une autre méthode possible est la minimisation de la distance euclidienne : $\sum(x_r - \widetilde{x}_r)^2$. L'objectif de cette distance est de trouver la solution du problème la plus proche de la matrice initiale.

Nous pouvons citer également une technique connue dans le cas des réseaux téléphoniques, qui est la méthode de Kruithof [47]. Dans cette approche, on suppose que l'on connaît le trafic total entrant et sortant à chaque nœ ud du réseau, de même qu'une estimation de la matrice initiale $\{\widetilde{x}_r\}$. Les éléments de la matrice $\{\widetilde{x}_r\}$ sont alors successivement corrigés, d'abord pour l'ensemble des trafics sortants (somme des colonnes de la matrice de trafic), et ensuite pour l'ensemble des trafics entrants (somme des lignes de la matrice). Ce procédé est répété de manière itérative jusqu'à convergence. Cette méthode n'est pas très adaptée aux réseaux IP, plus maillés que les réseaux téléphoniques commutés, où les chemins sont typiquement plus longs et les estimations initiales plus difficiles à obtenir.

Evaluation

Pour illustrer l'utilité de telles méthodes d'inférence déterministe, nous montrons un exemple d'application sur un réseau de 12 nœ uds et 19 liens où le problème à résoudre est le suivant :

$$\text{Min}_{1\leq r\leq R} \sum(x_r - \widetilde{x}_r)^2 \tag{3.3}$$

sachant que :

$$\sum_{r=1}^{R} a_{lr}x_r = y_l, \quad 1 \leq l \leq L, \quad x_r \geq 0. \tag{3.4}$$

On commence donc par définir une matrice de trafic sur le réseau en tirant aléatoirement les coefficients entre 0 et 200 Mbits/s pour toutes les routes origine-destination possibles. Après routage de la matrice, les volumes de trafic sur les liens sont obtenus. Ensuite, pour représenter la matrice de trafic initiale $\{\widetilde{x}_r\}$ (estimation de départ utilisée par la méthode d'inférence), nous appliquons une perturbation aléatoire aux coefficients de la matrice réelle (matrice précédente) en multipliant chaque coefficient de $\{\widetilde{x}_r\}$ par un facteur aléatoire compris entre $-a$ et a. Le paramètre a mesure l'amplitude de la perturbation.

Il faut alors obtenir les coefficients de la matrice $\{x_r\}$ vérifiant l'équation (3.4) et minimisant la distance donnée par (3.3).

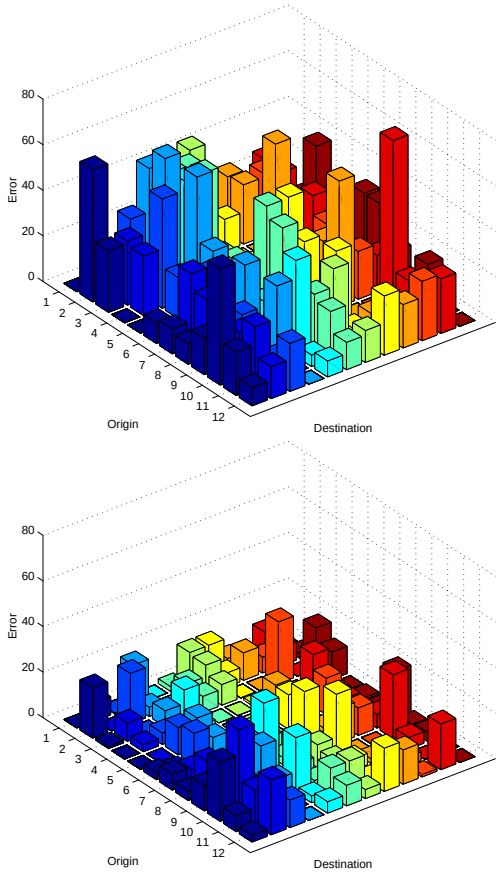

FIG. 3.4 – Ecart entre matrice estimée et matrice réelle ; résultats avant et après correction.

Des résultats sur un exemple particulier sont montrés sur la figure 3.4. Le graphique de gauche montre les écarts entre la matrice réelle et la matrice de départ (écarts dus à la perturbation aléatoire). Le graphique de droite indique les écarts obtenus après minimisation de la distance. Ces résultats montrent une réduction importante des erreurs lorsqu'on utilise l'information apportée par les volumes de trafic sur les liens. D'autres résultats du même genre sont obtenus lorsqu'on fait varier l'amplitude et le type de perturbation.

Inférence statistique

Des études récentes sur l'inférence de matrices de trafic IP utilisent des méthodes d'inférence statistique développées dans le cadre des problèmes de réseaux de transport routier, cf. [82, 84, 98]. Les données disponibles peuvent être un ensemble de mesures successives de volumes de trafic sur les liens. Cela peut provenir de données collectées par SNMP sur les interfaces des routeurs par période de 5 minutes.

En partant d'une hypothèse sur la forme des distributions des tailles de demande en trafic, l'objectif est de déterminer les valeurs optimales des paramètres des lois de probabilité étant donné l'ensemble des mesures disponibles.

L'approche la plus répandue est celle basée sur le Maximum de Vraisemblance [82]. La matrice de trafic initiale (information *a priori*) ainsi que les volumes de trafic sur les liens sont considérés comme étant des observations résultant de la matrice de trafic réelle. La méthode consiste alors à maximiser la vraisemblance d'observer $\widetilde{X} = \{\widetilde{x}_r\}$ et $Y = \{y_r\}$ conditionnellement à $X = \{x_r\}$. En supposant que $\{\widetilde{x}_r\}$ et $\{y_r\}$ sont indépendants, la vraisemblance s'exprime de la manière suivante :

$$L(\widetilde{X}, Y|X) = L(\widetilde{X}|X).L(Y|X) \tag{3.5}$$

Dans [82], l'expression exacte de la vraisemblance est obtenue dans le cas où $\{\widetilde{x}_r\}$ suit une distribution multinomiale (quand la taille de l'échantillon est petite) ou une distribution de Poisson (quand la taille de l'échantillon est plus importante). Pour les volumes sur les liens, on suppose aussi une distribution de Poisson. L'estimation de la matrice de trafic se fait alors en maximisant la vraisemblance obtenue sur toutes les demandes de trafic possibles $\{x_r\}$ satisfaisant les contraintes exprimées dans (3.1).

Une autre approche se fonde sur les moindres carrés généralisés où l'on suppose que $\widetilde{X}$ est le résultat d'une mesure de X affectée d'un terme d'erreur ϵ de moyenne nulle et de variance finie [20, 8]. Le problème revient alors à la minimisation d'une distance entre $\widetilde{X}$ et X qui tient compte de la matrice de variance-covariance du terme d'erreur.

Dans une toute autre approche, Vardi suppose que toutes les demandes de trafic $X = \{x_r\}$ suivent une distribution de Poisson ($X \sim Poisson(\lambda)$), puis estime les paramètres de cette loi par la méthode du Maximum de Vraisemblance [91]. L'objectif est d'estimer le vecteur de paramètres $\lambda = \{\lambda_r\}$ en se basant sur les observations des volumes de trafic sur les liens. En supposant une série de K observations des charges sur les liens notées $Y^k = \{y_r^k\}$, le Maximum de Vraisemblance conduit à l'équation suivante :

$$\lambda = \frac{1}{K}\sum_{k=1}^{K} E_\lambda[X^k|Y^k = AX^k] \tag{3.6}$$

Les méthodes EM (Expectation-Maximisation) peuvent alors être utilisées pour calculer λ en appliquant l'itération suivante :

$$\lambda^{n+1} = \frac{1}{K}\sum_{k=1}^{K} E[X^k|Y^k, \lambda^n] \tag{3.7}$$

Cette solution présente néanmoins quelques inconvénients : elle nécessite de trouver toutes les solutions en nombre entier de l'équation $AX = Y$ pour le calcul de l'espérance exprimée dans l'équation 3.7. Pour lever cette complexité, Vardi propose d'utiliser une approximation basée sur le théorème de la Limite Centrale, appliquée aux volumes de trafic sur les liens :

$$\bar{Y} = \frac{1}{K}\sum_{k=1}^{K} Y^k \rightarrow N(A\lambda, \frac{1}{K}A\Lambda A'), \quad \text{avec } \Lambda \equiv diag(\lambda) \tag{3.8}$$

Cette approximation permet de simplifier le problème, l'estimation du vecteur de paramètres se fait alors en maximisant la vraisemblance suivante :

$$l(\lambda) = -log|A\Lambda A'| - K(\bar{Y} - A\lambda)'(A\Lambda A')^{-1}(\bar{Y} - A\lambda) \tag{3.9}$$

Cette approche a été reprise par Cao *et al.*. Ils supposent que les demandes de trafic $\{x_r\}$ suivent une loi normale, tenant compte du fait que les volumes sur les liens sont en général plus variables que Poisson [18]. Des séries de mesure dans le temps sur les volumes de trafic des liens du réseau sont utilisées pour l'estimation. La méthode exposée dans [18] est complexe et n'est donc pas très adaptée à des réseaux de grande taille. Les auteurs proposent une méthode de décomposition qui permet le calcul sur des réseaux plus grands [19].

Nous citons enfin une méthode différente qui est l'approche bayesienne proposée par Tebaldi et West [84]. Ces auteurs identifient des problèmes de surestimation de la demande sur les routes de faible charge avec les méthodes d'inférence statistique classiques. Ils montrent alors qu'un choix d'une information *a priori* donnée par $p(\lambda)$ (déduite d'une estimation initiale de la matrice comme dans le paragraphe 3.5.3) améliore significativement la précision de l'estimation lorsque $X = \{x_r\}$ suit une distribution de Poisson ($X \sim Poisson(\lambda)$) :

$$p(X, \lambda) = p(\lambda)\prod_{i=1}^{R} \lambda_i^{x_i} exp(-\lambda_i)/x_i! \tag{3.10}$$

La méthode consiste à calculer la probabilité d'observer $X = \{x_r\}$ conditionnellement à $Y = \{y_r\}$. Cette distribution est alors liée aux deux distributions *a posteriori* $p(\lambda|X, Y)$ et $p(X|\lambda, Y)$ à travers la relation suivante :

$$p(X|Y) = p(X|\lambda, Y)p(\lambda)/p(\lambda|X, Y) \tag{3.11}$$

Le calcul de ces deux distributions utilise des méthodes de simulation itératives, du type MCMC, qui permettent de générer des échantillons de λ et X.

Il semble que les méthodes statistiques fournissent des estimations plus fiables de la matrice de trafic par rapport aux approches déterministes, puisqu'elles utilisent des données plus complètes. Cependant, elles sont aussi plus complexes à mettre en œ uvre et il est possible que les méthodes déterministes soient suffisantes pour la plupart des besoins de planification.

Estimation initiale de la matrice

Les méthodes d'inférence reposent sur une estimation initiale de la matrice de trafic $\{\widetilde{x}_r\}$. Cette estimation peut être obtenue de plusieurs manières : matrices antérieures ou mesures par échan-

tillonnage comme cité au paragraphe 3.5.2 pour les méthodes de [83] et [27]. Une autre approche est l'utilisation d'un modèle gravitaire pour la répartition du trafic dans le réseau [45, 94]. Le modèle gravitaire classique relie la demande de trafic sur une route r, d'origine le nœ ud i et de destination le nœ ud j, aux populations M_i et M_j et à la distance d_{ij} entre i et j, selon la relation :

$$\widetilde{x}_r = \alpha \frac{M_i M_j}{d_{ij}^{\beta}} \tag{3.12}$$

avec un choix adéquat des paramètres α et β.

Cette formulation peut être adaptée au trafic téléphonique [45] ou routier [94] mais ne semble pas valide dans le cas des réseaux IP où la distance a un impact plus faible sur le trafic. On peut néanmoins adapter ce modèle en utilisant d'autres facteurs, du type trafic total entrant et sortant (à l'instar de la méthode de Kruithof), ainsi que des facteurs impliquant des aspects tels que la localisation de points de peering, de serveurs, de clients entreprise ou le nombre de points d'accès grand public.

[54] propose une variation de la méthode gravitationnelle fondé sur des modèles de choix. Dans cette approche le trafic émis par un Point de Présence (POP) i vers un POP j est décrit comme une proportion α_{ij} du trafic émis par le POP i (O_i).

$$X_{ij} = O_i \alpha_{ij} \tag{3.13}$$

Le facteur de distribution α_{ij} est ensuite ajusté par le biais d'un modèle de choix [**?**]. L'idée sous-jacente est que la proportion de trafic allant de i vers j est supposée comme résultant d'un choix effectué au niveau du POP i sur la base de critère objectifs.

3.5.4 Comparaison des méthodes d'inférence

Une comparaison des différentes approches d'estimation de la matrice de trafic est présentée dans l'article [54]. Cette recherche compare analytiquement et empiriquement trois classes principales de techniques parmi celles qui ont été proposées à ce jour pour estimer la matrice de trafic d'un réseau en fonction d'informations incomplètes, telles que le trafic sur les liens : la **tomographie de réseau** [18, 91], l'**inférence bayésienne** [84], et la **programmation linéaire** [33].

Ces trois techniques diffèrent principalement par les aspects suivants : la programmation linéaire est une méthode déterministe alors que la tomographie du réseau et l'inférence bayésienne sont des méthodes statistiques. Les techniques statistiques peuvent exploiter des hypothèses raisonnables sur les propriétés statistiques de la matrice de trafic à estimer. Ces hypothèses peuvent inclure la distribution de probabilité (Poisson, loi Normale) et les paramètres de celle-ci, de même que la relation entre la moyenne et la variance des éléments de la matrice de trafic [18]. Rappelons en outre que l'approche Bayésienne proposée dans [84] suppose une distribution de Poisson pour les éléments de la matrice de trafic, tandis que [18] se fonde sur une hypothèse Gaussienne. Les deux approches divergent aussi sur la méthode d'estimation choisie : la tomographie du réseau applique une méthode de Maximum de Vraisemblance, tandis que l'approche Bayésienne est fondée sur l'analyse Bayésienne et sur des techniques de MCMC.

Le problème a été formalisé dans les paragraphes précédents : le vecteur des volumes de trafic par route $\mathbf{X}^{(k)}$, où k désigne l'une des observations disponibles, est solution d'un système d'équations linéaires sous-déterminé $AX^{(k)} = Y^{(k)}$ où A désigne la matrice de routage et $\mathbf{Y}^{(k)}$ le vecteur des volumes de trafic observés par lien. L'objectif est ici d'estimer le vecteur des débits moyens $\Lambda = (\lambda_1, ..., \lambda_R)$, où λ_r est le débit moyen de la paire Origine/Destination (OD) r. La comparaison de ces trois approches a été effectuée sur la base d'une matrice de trafic synthétique définie sur une partie de la topologie du réseau de l'opérateur SPRINT. Cette topologie contient 14 nœuds. Le problème d'estimation de matrice de trafic dans ce cas se réduit à résoudre un système avec 182 variables et 40 équations.

Afin de comparer les trois méthodes actuelle : programmation lineaire, Bayésienne et EM, nous avons généré quatre scénario de trafic.

- Constant : Dans ce scénario tous les éléments de la matrices de trafic sont constants.
- Uniforme : Les éléments de la matrice de trafic sont choisis suivant une distribution uniforme
- Poisson : Les éléments de la matrice de trafic sont choisis suivant une distribution de Poisson.
- Gaussien :Les éléments de la matrice de trafic sont choisis suivant une distribution de gaussienne.
- Bimodale :Les éléments de la matrice de trafic sont choisis suivant un mélange de deux gaussiennes.

On peut raisonnablement s'attendre à ce que la méthode bayésienne atteignent une meilleure estimation pour le scénario Poissonien, alors que la méthode EM dévrait avoir de meilleurs résultats pour l'hypothèse gaussienne. Le scénario bimodal a été ajouté car des observations empiriques le rende plausible.

La programmation linéaire a été appliquée à la minimisation de deux fonctions de coût : la première étant simplement la somme des erreurs d'estimation sur les paires OD et la seconde correspondant à $\sum_r c_r x_r$, où c_r est égal au nombre de sauts sur le chemin de la paire OD r [33]. Des contraintes de conservation de flux et de positivité des x_r ont aussi été appliquées.

La méthode Bayésienne nécessite la connaissance d'une distribution de trafic *a priori*. Celui ci est généré and ajoutant une erreur distribué selon une gaussienne de moyenne $\mu = 50$ et de variance $\sigma = 20$ à chaque élément de la matrice de traffic initiale.

La méthode EM nécessite aussi une valeur initiale. Cette valeur est aussi choisit suivant la même approche que celle décrite pour la méthode Bayésienne.

Le tableaux 3.3 effectuent la comparaison des l'erreur relative moyenne et maximal observé en appliquant chacune des trois méthodes d'estimation (Programmation linéaire, Bayésienne et EM) aux quatres scénarios de matrice de trafic.

La comparaison des quatres méthodes montre que l'approche de programmation linéaire ne donne pas de bons résultats, car l'erreur moyenne ainsi que l'erreur maximale sont très élévées. Ceci était prévisible, car la performance de cette approche est totalement liée à la fonction de coût et nous n'avons aucun moyen de définir celle ci. L'estimation bayésienne et EM atteignent de meilleurs résultats. Comme attendu la méthode bayésienne atteint ces meilleures performances pour le scénario poissonien, tandis que la méthode EM donne de meilleur résultat pour le scénario gaussien (et accessoirement pour le scénario constant qui est le scénario gaussien avec une variance nulle). D'autre part les performances de ces deux approches ne différent pas entre les scénarios constant, uniforme, poissonnien et gaussien. Ceci est intéressant car en pratique la distribution des éléments de la matrice

TAB. 3.3 – Résultats pour une matrice de trafic *a priori* égale à la matrice de trafic réelle augmentée d'un terme d'erreur Gaussien de moyenne $\mu = 50$ et d'écart-type $\sigma = 20$

	LPR		Bayésienne		EM	
Dist	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**
`Constant`	1.70	12.0	0.41	2.13	0.22	1.00
`Uniforme`	2.08	24.0	0.43	5.55	0.24	1.01
`Poisson`	1.73	12.8	0.37	2.83	0.23	1.28
`Gaussienne`	1.74	12.4	0.41	5.26	0.24	1.11
`Bimodale`	2.01	19.0	0.63	4.97	0.39	1.50

de trafic n'est pas *a priori* connue, et l'insensibilité de ces méthodes par rapport à la distribution les rend robuste. Néanmoins toutes les trois méthodes ont des problèmes avec le cas bimodal. Ceci est due au fait que la distribution bimodale n'est pas bien caractérisée par des moments d'ordre 2.

D'autre part la comparaison des méthodes EM et bayésienne montre de meilleurs résultats pour la première. Il semble que la qualité de l'estimation bayésienne est fortement conditionnée à la qualité de l'*a priori*. En conclusion, il apparait que les méthodes statistiques présentent de meilleurs approches au problème de l'inférence des matrices de trafic. Jusqu'ici la méthode EM semble présenter de meilleurs résultats.

3.6 Typologie des attaques

3.6.1 Introduction

Le développement des applications en réseau, des infrastructures de communication et le besoin de les interconnecter, ainsi que la démocratisation de cette nouvelle infrastructure qu'est l'Internet, devenue (ou en passe de devenir) le système de communication de données le plus universel, ont du même coup introduit un besoin très fort en matière de sécurité. En effet, ce n'est pas parce que des organismes ont décidé de se raccorder à l'Internet pour communiquer plus facilement avec leurs clients, futurs clients, partenaires, pour enrichir leurs modes de communication, pour faciliter leurs coopérations avec des partenaires distincts, etc., qu'ils ont nécessairement décidé de mutualiser l'ensemble de leurs ressources et connaissances. Ceci nécessite donc de mettre en place des mécanismes de sécurité pour contrôler notamment les accès extérieurs à leurs ressources, et ceci est vrai pour leur(s) éventuel(s) partenaire(s), mais aussi et surtout par rapport aux autres utilisateurs du réseau d'interconnexion. En effet, le fait de se connecter à un réseau d'interconnexion, pour bénéficier des facilités de communication, rend en même temps vulnérable à des attaques de pirates informatiques. Ce point concerne l'aspect le plus connu de la sécurité qui consiste à préserver la confidentialité de ses données et de lutter ainsi contre des pratiques d'espionnage industriel rendues plus aisées avec l'arrivée du réseau Internet très (trop) peu sécurisé aujourd'hui. Toutefois, les nouveaux usages et les nouvelles applications sur un réseau peu sécurisé, notamment ceux qui sont en relation avec le milieu de la finance (comme les services électroniques des banques) ou de l'économie (avec, par exemple, tout ce qui est en rapport avec le commerce électronique) ont attiré la convoitise de gens mal intentionnés qui trouvent là un secteur privilégié pour effectuer des extorsions et autres malversations financières. En outre, l'Internet peut aussi être un moyen de porter préjudice à une personne

ou à une entreprise, par exemple au moyen de techniques de déni de service, en empêchant cette entreprise d'offrir son service en le rendant indisponible de l'extérieur et contre la volonté de celui ou ceux qui le fournissent (vendent). Il est à noter également, et c'est certainement un des points les plus importants avec l'arrivée de l'Internet, que le fait d'attaquer un site ou une machine, de s'introduire sur un serveur pour récupérer les informations qui y sont stockées (si possible confidentielles), de trouver une faille dans un système de sécurité, etc., est devenu un jeu pour beaucoup d'informaticiens (des pirates ou "hackers" en herbe), lesquels voient dans ces pratiques un moyen de se mesurer les uns aux autres, de montrer l'étendue de leurs connaissances. Même si, à la base, les intentions de ces informaticiens ne sont pas mauvaises, les conséquences peuvent en être désastreuses. Il est également apparu que les pirates, qui ont besoin de plus en plus de capacité de traitement pour casser des systèmes de sécurité de plus en plus performants, cherchent en fait à détourner des machines peu sécurisées un peu partout dans le monde pour attaquer celles qui sont leurs cibles, ce qui a de plus l'avantage de détourner les soupçons vers des informaticiens honnêtes. Récemment même, de telles pratiques sont favorisées par l'explosion des nouvelles applications "Peer-to-Peer", telles que Napster, Kazaa, Morpheus, etc., qui créent pour leurs utilisateurs de nouveaux dangers. En effet, en permettant à n'importe quel autre utilisateur de venir lire ou copier des fichiers dans une zone partagée du disque, les utilisateurs de ces applications s'exposent aux attaques des pirates qui ont alors un espace de choix pour attaquer le système ciblé, ou pour détourner ce système et lui en faire attaquer d'autres.

Le problème de la sécurité des réseaux informatiques est extrêmement important de nos jours et stratégique pour tout le monde. Les solutions de sécurité, quant à elles, sont discutables sur de nombreux points qui vont de leur efficacité à leur adaptabilité.

Dans ce rapport, l'accent ne porte pas sur un état de l'art des techniques d'attaque qui peuvent être perpétrées en utilisant un réseau de communication. Ce n'est pas l'objectif de ce travail qui est fait par ailleurs [4]. L'objectif du projet METROPOLIS consiste à étudier et analyser les attaques qui sont perpétrées sur Internet, d'analyser leurs effets sur le trafic, de voir comment des virus (par exemple) se propagent, à quelle vitesse, etc.

Ce type d'étude n'a pour l'instant quasiment jamais été entrepris (à part, très récemment, l'étude de la propagation du vers Code-Red [57] par exemple). En effet, en matière de sécurité, les efforts avaient jusqu'à présent pour but de proposer des solutions de sécurisation, généralement regroupées sous le terme de "firewall" ou "pare-feux". Ces solutions, même si elles n'offraient pas des garanties de sécurité à 100 % contre tous les types de tentative d'intrusion (ce qui est totalement impensable en matière de sécurité informatique), offraient tout de même des niveaux de sécurisation satisfaisants par rapport aux types d'attaque qui existaient alors. Aujourd'hui, on assiste à une nette amélioration des attaques, qui utilisent des techniques distribuées pour être moins facilement détectables par les outils existants de type firewall. Des techniques collaboratives sont également utilisées afin de passer outre les systèmes de sécurité mis en place. Certains nouveaux types d'attaque mettent même en œ uvre des stratégies évoluées et adaptatives afin de tromper les firewalls actuels. Face à ces nouvelles attaques - distribuées, collaboratives et/ou stratégiques - les firewalls actuels sont dépassés. Il faudrait en fait qu'ils soient capables d'auto-apprendre, notamment les nouvelles stratégies d'attaque. La métrologie doit donc permettre d'aider à analyser ces nouveaux types d'attaque, de les caractériser, les analyser et les modéliser afin, dans un second temps, de montrer les principes qui devront régir le fonctionnement des futurs systèmes de sécurité.

Comme nous l'avons brièvement mentionné plus haut, la métrologie des attaques n'a pour l'instant quasiment jamais été mise en œuvre. L'état de l'art dans ce domaine est donc extrêmement court.

Toutefois, il existe une base commune entre les deux domaines des firewalls et de la métrologie des attaques. Ce point commun concerne la détection des intrusions : les IDS (ou “systèmes de détection d’intrusion”), qui servent à détecter les attaques et reposent parfois sur des techniques d’analyse du trafic, et qui pourront être utilisées dans METROPOLIS pour étudier les attaques éventuellement présentes dans les traces capturées. L’objectif de cette section est donc de faire un bilan des techniques de détection d’intrusion pouvant être déployées au moyen d’approches métrologiques. Pour cela, le paragraphe 3.6.2 présente une classification des différents types d’attaque et le paragraphe 3.6.3 présente les techniques de détection d’intrusion présentes dans les solutions de sécurisation actuelles et qui, reposant sur des approches similaires à la métrologie réseau, pourront être exploitées dans METROPOLIS. A noter que l’état de l’art relatif aux techniques des IDS n’est pas complet, tant il existe de systèmes différents ; en particulier, de nombreux produits sont à caractère commercial et leurs techniques de détection d’intrusion ne sont pas révélées. Toutefois, nous nous sommes efforcés de donner des exemples de toutes les familles d’IDS existantes.

3.6.2 Les principales attaques et leur impact sur la QoS réseau

Les attaques qui peuvent être perpétrées à partir d’un réseau sont très nombreuses, et leur nombre augmente régulièrement avec l’apparition de nouveaux logiciels ou de nouvelles techniques. On peut toutefois les segmenter en deux types d’attaque principaux :

- Les attaques qui sont liées à des bugs informatiques ou protocolaires (erreurs de programmation ou mauvaise utilisation) et qui génèrent des trafics qui peuvent par exemple s’apparenter à des attaques de déni de service ;
- Les tentatives d’intrusion volontaire qui cherchent réellement à stopper un service (et qui se manifestent par des anomalies sur le trafic), à s’introduire dans un système, ou toute autre forme d’action qui cherche à créer un préjudice à quelqu’un.

Il faut noter toutefois que les attaques ne viennent pas forcément de l’extérieur. Certaines sont aussi le fruit d’utilisateurs internes qui essaient d’outrepasser leurs droits.

Dans ce qui suit, on présente les attaques les plus classiques, classées en différentes familles, ainsi que les nouveaux types d’attaques telles que les attaques distribuées, collaboratives, et leurs nouvelles stratégies. Cette partie n’est en aucun cas complète car il est de toute manière impossible de connaître à un instant t tous les types d’attaque existant, tant ce domaine évolue rapidement, mais aussi, et surtout, en raison de son côté secret par nature.

Les attaques de ” flooding ”

Les attaques de “flooding” visent à surcharger les ressources d’un système afin que celui-ci ne puisse plus assurer le service qu’il a à rendre. Dans la suite, nous allons illustrer ce type d’attaque par l’exemple des attaques de “Syn-flooding” TCP, lesquelles ont été décrites dans [78, 59, 73] etc.

Le principe des attaques de Syn-flooding est de générer envers un serveur ou un groupe de serveurs un grand nombre de requêtes d’ouverture de connexion qui ne seront ensuite pas utilisées pour transporter des données, mais seulement allouées pour bloquer les ressources des routeurs ou des serveurs du réseau et ainsi dégrader la qualité de leur service, voire même saturer ces équipements qui ne pourraient plus alors accepter de nouvelle requête - d’où le nom de “déni de service”. Ce type d’attaque exploite une faille du protocole TCP au niveau de l’ouverture de connexion (“three-way

handshake"). En effet, de par son implémentation réelle, le récepteur comporte forcément un nombre limité d'entrées pour les requêtes de connexions. De plus, le protocole fixe un délai d'attente, habituellement de 75 s, pour la réception de la confirmation d'ouverture de connexion après réception de la demande d'ouverture. Ainsi, en générant suffisamment de requêtes d'ouverture de connexion, toutes les entrées du système récepteur peuvent être saturées, d'où le rejet de toute nouvelle demande de connexion. Les recommandations actuelles pour lutter contre le Syn-flooding consistent à augmenter la taille de la table des requêtes de connexion et à diminuer le délai acceptable entre demande et confirmation d'ouverture de connexion [59]. Toutefois, l'augmentation de taille de la table des connexions pendantes fait baisser les performances du système (recherche plus lente dans une table plus grande), et la diminution du délai risque de provoquer le refus d'ouverture de nouvelles connexions pourtant légitimes, venant de réseaux peu performants à longs délais.

Les figures 3.5, 3.6 et 3.7 - pouvant être exploitées pour mettre en œ uvre un système de sécurisation basé sur une comptabilisation des connexions pendantes comme dans [81] (mais faite au niveau du système opératoire) ou [59] (au niveau du réseau) - montrent comment une telle attaque se manifeste lorsque l'on observe le trafic sur un lien avec un outil de métrologie de type DAG (cf. Chapitre 3). Le lien supervisé est un lien d'accès à 155 Mbits/s chargé au tiers de sa capacité en heures pleines. Il faut noter que les outils actuels de supervision de réseau (par exemple les outils d'administration de réseaux basés sur SNMP) ne permettent pas de détecter de telles attaques. En effet, ces outils ne mesurent que la quantité de trafic envoyée sur un lien, comme sur la figure 3.5. En observant cette figure, rien ne permet de déceler que le réseau est en train de subir une attaque. En revanche, c'est en observant les figures 3.6 et 3.7, qui mesurent respectivement le nombre de paquets transportés par le réseau et le nombre de flux actifs, que l'on peut voir apparaître deux pics qui ne se traduisent pas par une augmentation en terme de trafic. Une analyse plus appuyée montre que les pics sont dus à une augmentation dramatique du nombre de paquets de synchronisation de TCP (Syn). Vu le nombre de flux ouverts pendant les périodes d'attaque, le serveur visé est saturé et ne peut plus ouvrir de nouvelles connexions. De même, les connexions déjà ouvertes voient la qualité de service réseau qui leur était offerte se dégrader de façon importante, du fait du plus grand nombre de paquets que doivent traiter les routeurs.

Ce type d'attaque reste pourtant facile à détecter avec les firewalls installés en bordure du réseau, mais en cœ ur de réseau et avec les outils actuels basés sur SNMP, peu réactifs, il aura fallu près d'une heure pour détecter l'attaque et la parer. Pendant une heure, le service offert par l'opérateur s'est donc trouvé dégradé, en tous cas sans rapport avec le contrat de service (SLA) signé avec ses clients. De telles attaques doivent donc être détectées au plus tôt au niveau du réseau.

Ce qui précède a détaillé spécifiquement le cas des attaques de Syn-flooding. Le principe reste le même pour tous les types de protocole, par exemple ICMP, UDP, DNS, etc., souvent en utilisant des options inadaptées. De plus, ces attaques peuvent cibler différents types de démon pour faire stopper un service.

Les intrusions

Les intrusions sont des attaques bien connues des administrateurs de réseaux d'ordinateurs et/ou de serveurs. [4] en donne une liste assez conséquente qui peut servir d'état de l'art, même si celui-ci évolue très rapidement. Un des principes courants de ces attaques consiste pour un pirate à exploiter une faille d'un système opératoire et s'en servir pour consulter, voire voler des fichiers et documents, les modifier, utiliser cette machine pour attaquer d'autres systèmes ou installer des chevaux de Troie

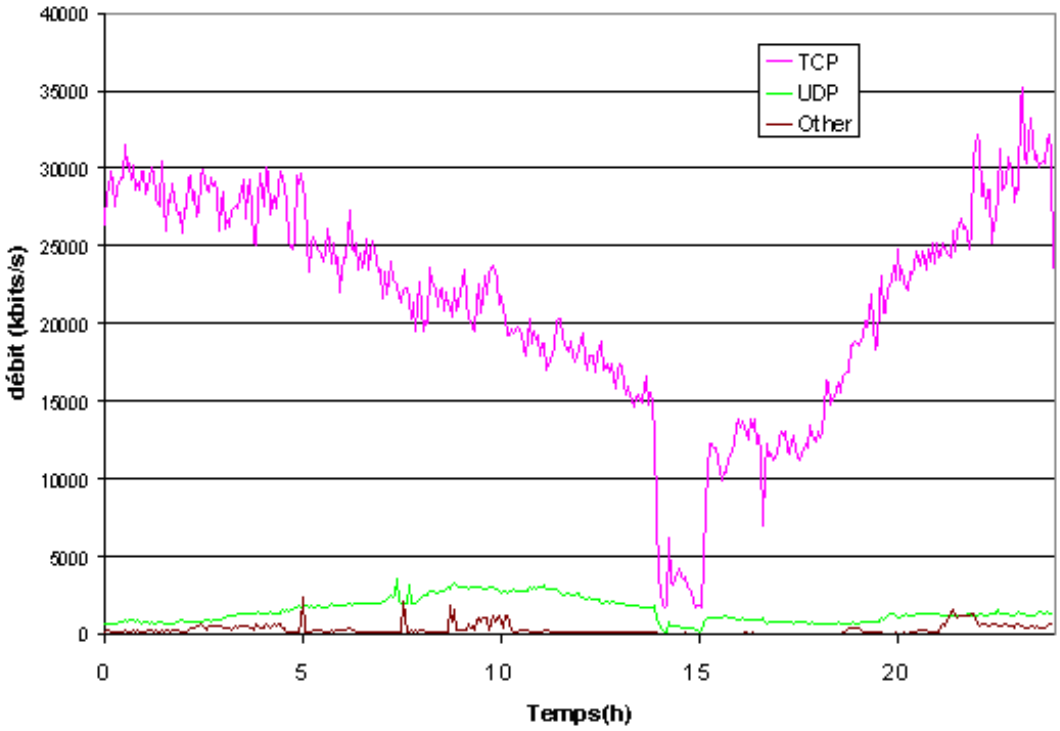

FIG. 3.5 – Trafic instantané sur un lien d'accès à 155 Mbits/s

permettant ensuite d'entrer à volonté dans ce système, etc. L'exemple suivant présente une de ces tentatives d'intrusion (une tentative d'attaque massive) sur les ports connus des chevaux de Troie. Elle a été révélée en analysant les traces (logs) de connexions sur un routeur :

210.225.32.82 (io.indexo.co.jp)

Scanning port(s) tcp (10008), tcp (1008), tcp (11753), tcp (12345),

tcp (12754), tcp (15104), tcp (1524), tcp (22252), tcp (2400), tcp (3879),

tcp (39168), tcp (5300), tcp (6635), tcp (6723), tcp (8282), tcp (9112), tcp (9705)

Beginning at : Jun 3 2001 23 :33 :27

Ending at : Jun 3 2001 23 :39 :54

On these address in our network : 140.93.13.10

Dans le cas présent, cette attaque visait un serveur de fichiers et d'applications - ce sont là les types d'attaque les plus fréquents. Cependant, il est apparu récemment que ce type d'attaque vise aussi les routeurs. En effet, le moyen le plus efficace pour provoquer des dénis de service est certainement de perturber, dégrader, ou même stopper l'activité des routeurs du réseau. Ce type d'attaque

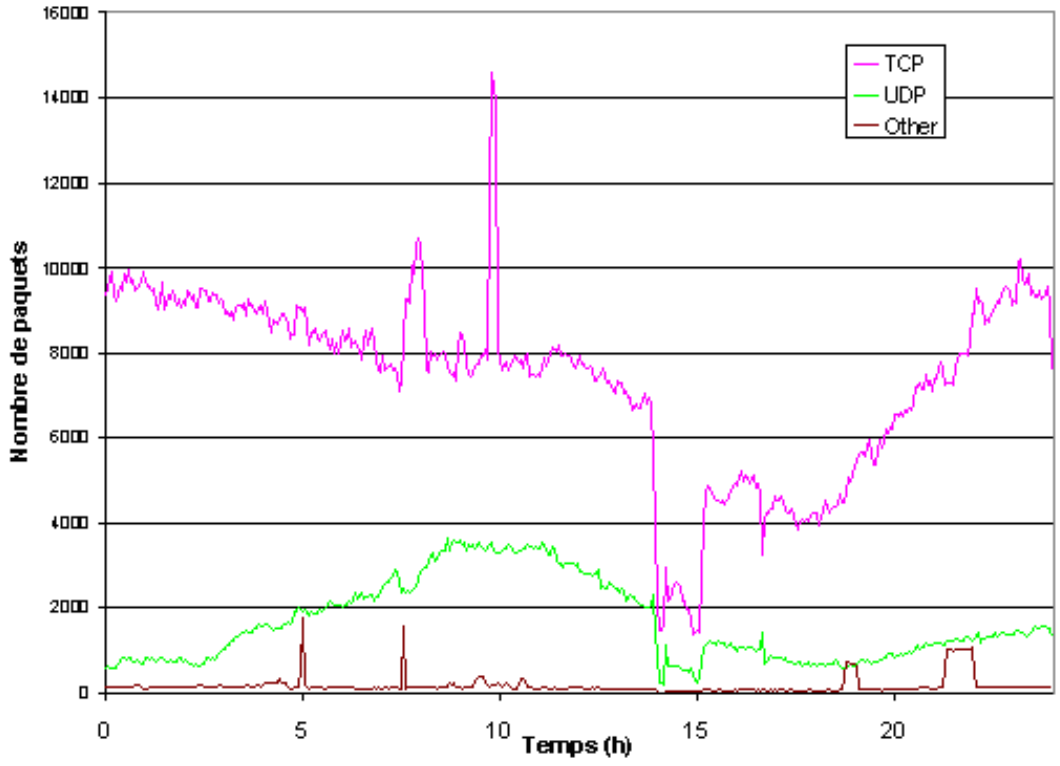

FIG. 3.6 – Nombre de paquets transitant sur un lien d'accès à 155 Mbits/s

est évidemment très pénalisant pour les opérateurs de réseaux, lesquels doivent dorénavant se pencher sur ces problèmes, ce qui n'était pas forcément le cas lorsque les attaques visaient seulement les équipements d'extrémité et leurs contenus.

Les attaques évoluées

Comme nous l'avons mentionné dans l'introduction, pour prendre le dessus sur les équipements de sécurité de plus en plus répandus et de plus en plus performants, les pirates imaginent des attaques de plus en plus évoluées, comme les attaques distribuées, ayant des stratégies qui s'adaptent aux systèmes de défense en présence ou qui impliquent plusieurs machines ou plusieurs pirates de manière collaborative.

Les attaques de Déni de Service distribuées (ou DDoS) - Comme l'ont montré [4, 49] ou [69] (par exemple), les attaques de déni de service sont de plus en plus souvent distribuées, c-à-d perpétrées à partir de plusieurs machines situées en des endroits très différents afin d'être moins facilement

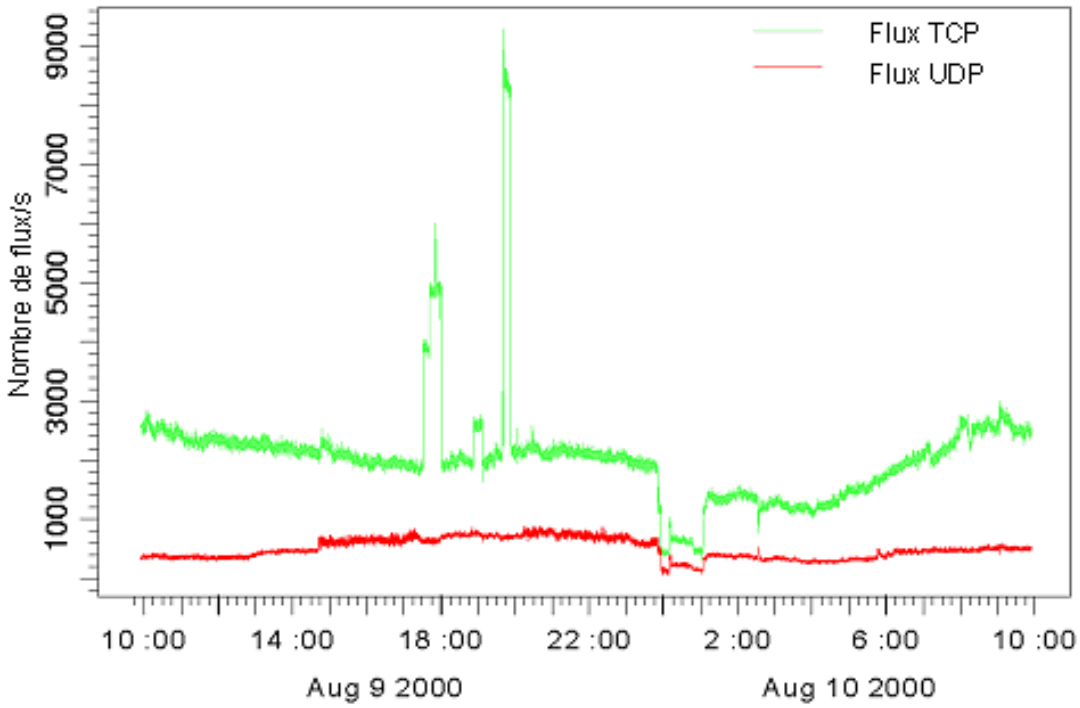

FIG. 3.7 – Nombre de flux actifs sur un lien d'accès à 155 Mbits/s

détectables. En effet, l'augmentation du nombre de certains éléments caractéristiques comptés par les systèmes de sécurité (comme les paquets Syn) est ainsi plus légère et peut mystifier un système de sécurité tel que celui décrit dans [59] pour les attaques de Syn-flooding.

De même, un système qui analyserait le profil des adresses sources, en les classant par niveau de confiance [59, 78], pourrait être floué car il lui serait difficile de déterminer rapidement si une adresse est une adresse correspondant à un internaute honnête ou à un pirate.

Les attaques à stratégies évoluées - Des attaques menées avec des stratégies évoluées commencent à apparaître. De telles attaques sont évidemment difficiles à détecter par les systèmes de sécurité actuels. [39] en relate certains exemples. En guise d'illustration, les figures 3.8 et 3.9 présentent un exemple de stratégie d'attaque permettant de déguiser son adresse source ("IP address spoofing").

Ainsi, si I (Intrus) veut attaquer V (Victime) en déguisant son adresse source, il peut appliquer la stratégie suivante (figure 3.8) :

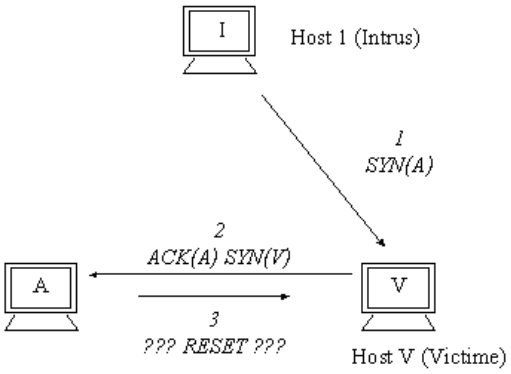

FIG. 3.8 – Attaque classique avec déguisement d'identité

1. En premier lieu, I envoie une requête d'ouverture de connexion à V en prétendant être A
2. En réponse à la requête venant apparemment de A, V envoie un acquittement et un accord de connexion à A
3. Le problème pour I est d'éviter que A reçoive cette confirmation/acquittement. En effet, s'il reçoit ce paquet, n'ayant pas demandé d'ouverture de connexion, il va émettre vers V un paquet RST (Reset) pour fermer cette connexion

En procédant ainsi, I ne pourra tromper V, notamment grâce à la coopération de A qui va fermer au plus vite la connexion. Pour que cette attaque soit efficace, il faut changer de stratégie et appliquer plutôt celle représentée sur la figure 3.9 :

1. Pour éviter que A ne réponde à V et fasse avorter la tentative de connexion de I vers V, I va alors faire précéder son attaque de V par une attaque de Syn-flooding vers A (en se faisant passer pour X), pour éviter que A puisse répondre à V et terminer ainsi la connexion
2. I envoie ensuite une requête d'ouverture de connexion à V en prétendant être A
3. En réponse à la requête venant apparemment de A, V envoie un acquittement et un accord de connexion à A
4. A ne pouvant pas répondre à cause de l'attaque de déni de service, I envoie alors à V un acquittement déguisé de A confirmant l'ouverture de connexion
5. I peut alors scanner les ports de V à la recherche par exemple d'un cheval de Troie

Les attaques collaboratives - Enfin, [39] évoque également des attaques collaboratives où plusieurs machines collaborent pour attaquer un système. Ces attaques n'ont pas pour l'instant été mises en évidence dans les logs des firewalls, mais certains logiciels d'attaque commencent à apparaître sur Internet. Ces logiciels profitent des toutes dernières améliorations techniques en matière d'attaques (virus notamment) et, grâce à leurs connaissances sur les techniques de sécurisation, mettent

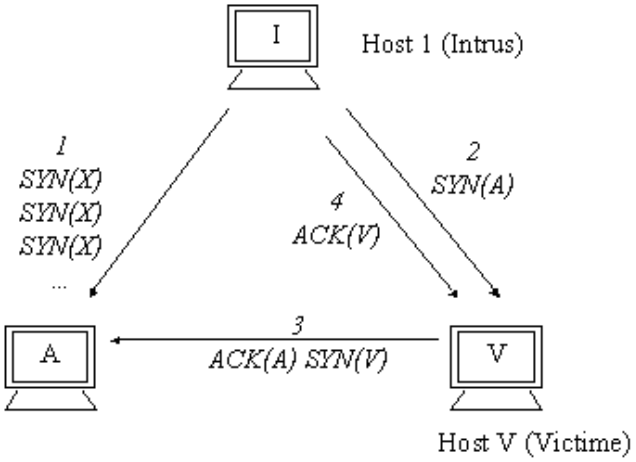

FIG. 3.9 – “Flooding” de A permettant le déguisement d’adresse source

en place des stratégies coopératives avancées entre plusieurs sites pour tromper les firewalls. Pour contrer ces attaques collaboratives, [39] recommande de mettre en relation tous les IDS du monde pour analyser et mettre en évidence toutes ces nouvelles stratégies d’attaque en corrélant leurs logs ou traces.

3.6.3 Les grands principes des IDS actuels

Les IDS actuels se différencient sur de nombreux points :

- La provenance des sources d’information : système ou réseau
- Le type d’IDS : actif ou passif
- Le mode de détection des intrusions : comportemental (ou à partir de signatures) ou basé sur un profil statistique
- Le type d’apprentissage : initialisé au début ou construit sur les observations faites pendant le fonctionnement de l’IDS (auto-apprentissage)
- Le fonctionnement individuel ou en collaboration avec d’autres IDS

Pour plus de détails, [6] propose un état de l’art des différents IDS issus du monde de la recherche, tandis que [48] en fait de même pour les IDS commerciaux. A noter cependant que les informations sur les IDS commerciaux restent très secrètes et tout ce qui est relatif à ces systèmes provient des rares informations données par leurs concepteurs.

Sources de l'IDS

Le premier élément qui permet de différencier les IDS est relatif aux sources d'information qui leur permettent de détecter les attaques ou anomalies. Ainsi, de très nombreux systèmes (généralement les plus anciens) reposent sur l'analyse de traces système comme les traces des appels système réalisés. C'est le cas par exemple de [81] qui compte le nombre de demandes de connexions TCP passant dans le système opératoire de la machine (OS), afin de détecter ainsi les attaques de Syn-flooding. C'est également le cas pour [52] qui compte le nombre d'appels système émis par les utilisateurs, et qui identifie des utilisateurs suspects, voire les déconnecte, dès que les proportions traditionnelles pour chaque type d'appel (le profil type) ne sont pas respectées. [93] va plus loin en introduisant également la notion d'ordre : en effet, certaines combinaisons ordonnées d'appels système sont connues comme étant dangereuses. Si de tels ordres sont observés, l'auteur de ces séquences d'appels est automatiquement déconnecté et interdit de connexion au système. ASAX (Architecture and rule-based language for universal audit trail analysis) [34] analyse également les appels système UNIX et les compare à des profils typiques que le système crée à partir de l'expérience acquise au cours de son fonctionnement.

Toutefois, certains IDS récents observent et analysent des traces d'évènements ou de paquets transitant sur le réseau ; c'est le type de système qui nous intéresse plus particulièrement dans le cadre de METROPOLIS. La suite de ce paragraphe va donc se focaliser sur les IDS de niveau réseau (même si parfois, pour illustrer certains types de détection d'intrusion non encore exploités par des IDS de niveau réseau, nous citerons en exemple des outils fonctionnant au niveau host). A noter cependant que les IDS de niveau réseau se différencient également par la couche du modèle OSI qu'ils considèrent : certains regardent les traces au niveau 3, 4, 5 ou 7. Bro [69], par exemple, travaille au niveau 7 (application). A noter également que si les produits issus de la recherche reposent essentiellement sur des analyses de niveau host, les IDS commerciaux fonctionnent le plus souvent sur l'analyse des paquets du réseau et parfois de leur contenu utile ("payload"). C'est le cas pour Net Ranger, Skate Out, SessionWall-3, SecureNet PRO, Intouch INSA, T-Sight, ID-Trak, SecureCom ou Network Flight Recorder. D'autres comme RealSecure, Intruder Alert ou CyberCop combinent des sources réseau et host.

Type d'IDS

Les IDS se divisent aussi en deux classes en fonction de leur réaction à la détection d'intrusion. Ainsi, les systèmes passifs se contentent de signaler l'intrusion détectée à l'organe responsable de la politique de sécurité. D'autres, les systèmes actifs, mettent en place des parades à ces intrusions. En guise d'exemple, revenons sur les parades pouvant être mises en place en cas de détection d'attaques de type Syn-flooding. [59] et [78], lorsqu'ils détectent une telle attaque, génèrent automatiquement un paquet RST vers l'émetteur du paquet SYN pour fermer au plus vite la connexion, et ainsi éviter de saturer la table des connexions pendantes. [73], par contre, met en place un mécanisme de rejet aléatoire des paquets SYN lorsque ceux-ci arrivent avec un taux supérieur à la normale.

Mode de détection

Les IDS se différencient aussi par le mode de détection des intrusions qu'ils mettent en œ uvre. Deux grandes familles existent :

– Les IDS basés sur la détection d'anomalies par rapport à des comportements protocolaires (aussi appelées anomalies par rapport à des signatures)
– Les IDS basés sur la détection d'anomalies par rapport à des profils statistiques considérés comme normaux

Signature - [78], par exemple, détecte les attaques de type Syn-flooding en comparant la trace des paquets reçus par rapport à la machine à états de TCP, et notamment par rapport aux timers mis en place par TCP. Si la machine à états n'est pas respectée, le système écarte les paquets de la connexion considérée. Bro [69] agit de même au niveau applicatif : si les protocoles pour des applications telles que Finger, Telnet, FTP, etc., ne sont pas respectés, des anomalies sont alors décelées. NSM (Network Security Monitor) [35] [58] agit sur le même principe que Bro. USTAT (de niveau host !) [40, 41] détecte aussi les intrusions par rapport à des schémas état/transition qui ne correspondent pas aux protocoles considérés. A noter que ce système qui analyse une trace réseau se base sur un modèle comportemental de TCP au niveau UNIX. IDIOT [46] (de niveau host !) est un peu similaire a USTAT mais remplace la machine état/transition par un réseau de Petri coloré, détectant ensuite les intrusions par des scénarios d'événements qui ne respectent pas les comportements possibles et acceptés par le réseau de Petri. NIDES [2] (de niveau host !) implémente aussi un mécanisme de détection d'intrusion basé sur un système expert fonctionnant à partir de signatures.

Les IDS commerciaux fonctionnent pour la plupart sur ce principe, tels RealSecure, Intruder Alert, NetRanger, CyberCop, SecureNet PRO, ID-Trak, SecureCom ou Network Flight Recorder.

Statistiques (anomalies) - NIDES, que nous venons de voir dans le paragraphe précédent, implémente également des techniques de détection d'intrusion basées sur des profils considérés comme corrects statistiquement. IDES [42], le prédécesseur de NIDES, fonctionnait également sur ce principe. IDES ainsi que NIDES (tous deux fonctionnant au niveau host !) possèdent donc un module qui crée des profils statistiques, pour un utilisateur ou un groupe d'utilisateurs, selon plusieurs granularités temporelles (cette notion d'échelles temporelles étant essentielle pour pouvoir détecter à la fois des attaques brèves et des attaques longues). Si un comportement anormal est détecté, une alarme est alors générée. Wisdom & Sense [89] (de niveau host !) fonctionne sur le même principe en créant des statistiques par utilisateur de leur comportement, pour déterminer un comportement propre qui leur est normal et détecter ensuite les intrusions en cas de comportement ne respectant plus ce modèle. NADIR [38] (de niveau host !) fonctionne de même, mais avec des statistiques recueillies sur une semaine. Parmi les IDS commerciaux, à notre connaissance, seul T-Sight fonctionne avec une approche statistique, qui est dans ce cas là purement visuelle, c-à-d qui met en évidence de façon graphique les traces d'un intrus, comme sur les figures3.6 et 3.7 par exemple.

[56], qui utilise un système de métrologie réseau au sens de METROPOLIS, a proposé des observations qui pourraient être utilisées dans le cadre d'un IDS. Ainsi, des statistiques sur la distribution géographique des adresses sources ou sur la distribution des numéros de port pourraient être intéressantes, notamment pour déceler des adresses déguisées qui ne respecteront certainement pas la distribution des adresses généralement reçues, ou pour déceler des scans de chevaux de Troie, par exemple, qui s'installent sur des numéros de port élevés (au delà des ports 1 à 1024 utilisés par les services standards) et vont en conséquence modifier la distribution nominale sur une courte échelle de temps. A noter également que [56] identifie un certain nombre de lacunes concernant les IDS précédents, de type statistique, essentiellement basés sur des analyses du trafic, car ils ne tiennent pas

compte de certaines caractéristiques du trafic Internet, telles que la dépendance à long terme (LRD), qui rendent les statistiques malaisées à évaluer et le choix de la granularité d'étude primordial.

Mixtes - EMERALD [71, 72] met en place à la fois un mécanisme de détection par signature (basé sur un système expert d'étude des comportements des protocoles) combiné à un système de détection d'anomalies à partir de statistiques descriptives non temporelles.

D'autres systèmes reprenant les mêmes approches existent, notamment dans les IDS commerciaux, comme Skate Out qui utilise des techniques d'Intelligence Artificielle pour détecter les comportements anormaux, et Intouch INSA qui effectue des comparaisons par rapport à des profils génériques statiques. Leur description n'apporterait pas d'information supplémentaire par rapport aux familles de stratégies qui sont appliquées dans les IDS. Pour plus d'informations, le lecteur pourra se reporter à [6]. A noter toutefois que la plupart des IDS existants reposent sur des approches orientées host. Les IDS orientés réseau sont moins nombreux et généralement basés sur un mode de détection par signature (vérification des comportements protocolaires). Les approches de détection par anomalies statistiques sont moins utilisées car moins performantes aujourd'hui, notamment en raison des caractéristiques difficiles du trafic Internet telles que la LRD par exemple [56].

Il est important de signaler également qu'une des problématiques importantes des IDS consiste à ne pas générer de fausses alertes, lesquelles sont un des travers récurrents des IDS. Ce phénomène s'explique par le fait que la population des événements normaux est de taille très supérieure à celle de la population des attaques, comme on peut le démontrer aisément à l'aide de la formule de Bayes [5].

Mode d'apprentissage

Un élément important qui différencie les IDS concerne leur mode d'apprentissage. Deux approches existent : la première consiste à donner au système toutes les connaissances dont il a besoin au moment de son installation. C'est l'approche qualifiée de "programmée", qui est donc statique. C'est l'approche retenue par exemple dans un système comme Bro, EMERALD ou Skate Out. Intouch INSA ou T-Sight ont de plus la particularité de permettre à l'administrateur de modifier directement les règles de détection d'intrusion.

L'autre approche consiste pour le système à apprendre lui même les profils ou signatures qui devront être respectés par le trafic. Pour cela, l'IDS soit établira des profils statistiques en fonction des paquets qu'il voit passer (comme IDES, NIDES, Wisdom & Sense, NADIR, etc.), soit créera sa base de connaissance à partir de règles (MIDAS [79]) (tous ces derniers IDS fonctionnant au niveau host). A notre connaissance, seul ID-Trak, au niveau réseau, est capable d'ajouter dynamiquement des signatures d'attaque à sa Base de Données d'intrusions.

A noter que le mode d'apprentissage programmé est le mode toujours utilisé par les IDS de niveau réseau. Cette famille d'IDS repose sur des analyses de comportement protocolaire, et non pas sur la détection d'anomalies (qui devrait enrichir en temps-réel ses connaissances sur le trafic et s'adapter en permanence à des caractéristiques ponctuelles de ce trafic).

Collaboration entre IDS

La dernière tendance sur les IDS récents consiste à faire coopérer les différents IDS positionnés en différents points du réseau, afin de s'adapter aux attaques qui sont de plus en plus distribuées. En ce sens, les nouvelles tendances suivent les recommandations de [39]. Des exemples de tels systèmes sont CMS [95] (de niveau host !), JiNAo [44] (de niveau host !) ou EMERALD.

3.6.4 Conclusion

La métrologie réseau forme la base principale pour concevoir des IDS [6], que ce soit avec des approches d'analyse protocolaire, de comptage des différents éléments caractéristiques de certaines attaques ou de comparaison par rapport à des profils normaux de trafic. Toutefois, le travail que l'on se propose de faire dans METROPOLIS - à savoir analyser, caractériser et modéliser les attaques présentes dans le trafic - n'a que très peu été abordé pour l'instant. A notre connaissance, seul CAIDA a abordé cet aspect avec une étude des caractéristiques des attaques (quel protocole est utilisé, quelle est la durée des attaques, quel est statistiquement le risque d'attaque en fonction du nom de domaine correspondant à l'adresse source d'un paquet, etc. ?) [56]. De même, CAIDA a également proposé une analyse métrologique de la propagation du vers Code-Red dans l'Internet [57]. A un niveau moindre, [69] a également réalisé avec Bro quelques statistiques sur les attaques, basées sur l'utilisation non conforme de certains protocoles applicatifs comme Telnet ou FTP.

Il faut noter également que, jusqu'à présent, les IDS basés sur l'étude du trafic n'effectuent leur analyse qu'en bordure du réseau (entre le LAN et le WAN). La métrologie permettra de voir quels sont les effets de ces attaques au niveau des routeurs de cœur de réseau ou à l'accès, donc selon un point de vue opérateur ou ISP. Nous pensons d'ailleurs, même si ce thème n'est pas dans le cadre de METROPOLIS, que l'outil de métrologie peut permettre d'obtenir de meilleures analyses des attaques que celles faites à l'extrême bordure du réseau Internet, voire même contribuer à la sécurisation de l'Internet et de ses utilisateurs en permettant de détecter les attaques dans le cœ ur du réseau (et donc bien avant le firewall protégeant un réseau local). De même, avec un réseau complètement supervisé par des équipements de métrologie (comme les équipements DAG sélectionnés dans le cadre de METROPOLIS), on pourrait envisager de corréler les traces de trafic en de nombreux points du réseau afin de détecter les attaques distribuées, mettre en évidence les stratégies distribuées et collaboratives des attaques les plus récentes, et ainsi pouvoir les bloquer au plus tôt (en imaginant un système de métrologie qui pourrait influer sur la politique de sécurité mise en place au niveau des routeurs et des équipements de sécurité de cœ ur et de bordure de réseau).

A noter toutefois que la menée de telles études dans le cadre du projet METROPOLIS reste subordonnée à la mise en place en nombre suffisant d'équipements de métrologie au cœ ur des réseaux, qui permettrait d'effectuer des analyses sur les attaques distribuées et/ou collaboratives ou sur la propagation des vers et des virus, ou encore sur la mise en évidence de nouvelles stratégies d'attaque, ce qui représenterait le cadre d'étude le plus prospectif, le plus ambitieux, et surtout celui qui présente le plus d'intérêts scientifiques et commerciaux pour les années à venir [39]. Une autre limitation qui pourrait être rencontrée dans METROPOLIS est due au nombre croissant de connexions qui sont cryptées et ne pourront donc sans doute pas être analysées du point de vue des intrusions au niveau TCP/IP [48].

Bibliographie

[1] P. Abry and D. Veitch, *"Wavelet analysis of long-range dependent traffic"*, IEEE Trans. Information Theory, Vol. 44, pp. 2-15, 1998.

[2] D. Anderson, T. Frivold, A. Valdes, *"Next generation intrusion-detection expert system (NIDES)"*, Technical report SRI-CSL-95-07, Computer Science Laboratory SRI International, May 1995.

[3] M. F. Arlitt and C. L. Williamson, *"Web server workload characterization : The search for invariants"*, Proc. of ACM SIGMETRICS '96, pp. 126-137, May 1996.

[4] S. Aubert, *"Les dénis de service réseau"*, Journées Réseaux (JRES'2001), Lyon, France, 10-14 Déc. 2001.

[5] S. Axelsson, *"On a difficulty of intrusion detection"*, 2nd International Workshop on Recent Advances in Intrusion Detection (RAID'99), Purdue University, West Lafayette, Indiana, USA, Sept. 7-9, 1999.

[6] S. Axelson, *"Intrusion detection systems : a survey and taxonomy"*, Technical report No 99-15, Department of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.

[7] M. G. H. Bell *"The Estimation of an Origin Destination Matrix from Traffic Counts"* Transportation Science 17, pp 198-217, 1983.

[8] M. G. H. Bell *"The Estimation of Origin Destination Matrices by Constrained Generalised Least Squares"* Transportation Research 25B, pp 13-22, 1991.

[9] S. Ben Fredj, T. Bonald, A. Proutière, G. Regnié, J. Roberts, *"Statistical bandwidth sharing : a study of congestion at flow level"*, Proceedings of ACM Sigcomm 2001, Computer Communication Review, Vol 31, No 4, October 2001.

[10] J. Beran, R. Sherman, M. S. Taqqu and W. Willinger, *"Long-range dependence in Variable-Bit-Rate video traffic"*, IEEE Trans. Comm., Vol. 43, No 2/3/4, pp. 1566-1579, 1995.

[11] S. Bhattacharyya, C. Diot, J. Jetcheva and N. Taft, *"POP-level and access-link-level traffic dynamics in a Tier-1 POP"*, Proc. of ACM SIGCOMM Internet Measurement Workshop, Nov. 2001.

[12] M. Bierlaire, Ph. L. Toint, *"MEUSE : An Origin Destination Matrix Estimator That Exploits Structure"*, Transportation Science 31(4), pp 363-371, 1994.

[13] U. Black, *"TCP/IP and related protocols"*, McGraw-Hill, 1992.

[14] V. A. Bolotin, Y. Levy and D. Liu, *"Characterizing data connection and messages by mixtures of distributions on logarithmic scale"*, Proc. of ITC '16, pp. 887-894, P. Key and D. Smith (Editors), Elsevier, June 1999.

[15] T. Bonald, A. Proutière, G. Régnié and J. Roberts, *"Insensitivity results in statistical bandwidth sharing"*, Proc. of ITC '17, Moreira de Souza, Fonseca and de Souza e Silva (eds.), Elsevier, Dec. 2001.

[16] J. Cao, W. S. Cleveland, D. Lin and D. X. Sun, *"On the nonstationarity of Internet traffic"*, Proc. of ACM SIGMETRICS '01, June 2001.

[17] J. Cao, W. S. Cleveland, D. Lin and D.X. Sun, *"Internet traffic tends to Poisson and independent as the load increases"*, Bell Labs report, 2001, available at http ://cm.bell-labs.com/cm/ms/departments/sia/InternetTraffic/webpapers.html.

[18] J. Cao, D. Davis, S. Vander Wiel, B. Yu, *"Time-Varying Network Tomography : Router Link Data"*, Journal of the American Statistical Association, Vol 95, No 452, pp 1063-1075, 2000.

[19] J. Cao, S. Vander Wiel, B. Yu, Z. Zhu, *"A scalable Method for Estimating network traffic matrices"*, Bell Labs technical report, 2000, available at http ://cm.bell-labs.com/~cao.

[20] E. Cascetta, *"Estimation of Trip Matrices from Traffic Counts and Survey Data, a Generalized Least Squares Estimator"*, Transportation Research 18B, pp 288-299, 1984.

[21] J. Charzinski, *"HTTP/TCP connection and flow characteristics"*, Performance Evaluation, Vol. 42, pp. 149-162, 2000.

[22] K. Claffy, H-W Braun and G. Polyzos, *"A parametrizable methodology for Internet traffic flow profiling"*, IEEE JSAC, Vol. 13, No 8, Oct. 1995.

[23] K. Claffy, G. Miller and K. Thompson, *"The nature of the beast : recent traffic measurements from an Internet backbone"*, Proc. of INET '98, 1998 (http ://www.isoc.org/isoc/conferences/inet/98/proceedings/6g/6g_3.htm).

[24] M. Crovella and A. Bestavros, *"Self-similarity in World Wide Web traffic : Evidence and possible causes"*, IEEE/ACM Trans. on Networking, Vol. 5, No 6, pp. 835-846, Dec. 1997.

[25] A. B. Downey, *"The structural cause of the file size distributions"*, Proc. of IEEE MASCOTS '01, 2001.

[26] A. B. Downey, *"Evidence for long-tailed distributions in the Internet"*, Proc. of ACM SIGCOMM Internet Measurement Workshop, Nov. 2001.

[27] N. G. Duffield, M. Grossglauser, *"Trajectory Sampling for Direct Traffic Observation"*, Proceedings of ACM Sigcomm 2000, Computer Communication Review, Vol 30, No 4, 2000.

[28] A. Feldmann, *"Characteristics of TCP connection arrivals"*, In 'Self-similar network traffic and performance evaluation', edited by K. Park and W. Willinger, J. Wiley & Sons, 2000.

[29] A. Feldmann, A. C. Gilbert and W. Willinger, *"Data networks as cascades : Explaining the multifractal nature of Internet WAN traffic"*, Proc. of ACM SIGCOMM '98, Aug. 1998.

[30] A. Feldmann, A. C. Gilbert, W. Willinger and T. G. Kurtz, *"The changing nature of network traffic : Scaling phenomena"*, Computer Communication Review, Vol. 28, No 2, April 1998.

[31] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, F. True, *"Deriving Traffic Demand for Operational IP Networks : Methodology and Experience"*, Proceedings of ACM Sigcomm 2000, Computer Communication Review, Vol 30, No 4, 2000.

[32] S. Floyd and V. Paxson, *"Difficulties in simulating the Internet"*, IEEE/ACM Trans. on Networking, Vol. 9, No 4, pp. 392-403, Aug. 2001.

[33] O. Goldschmidt, *"ISP Backbone Traffic Inference Methods to Support Traffic Engineering"*, Internet Statistics and Metrics Analysis (ISMA) Workshop, San Diego, CA, USA, Dec. 2000.

[34] J. Habra, B. Le Charlier, A. Mounji, I. Mathieu, *"ASAX : Software architecture and rule-based language for universal audit trail analysis"*, ESORICS'92, Toulouse, France Nov. 23-25, 1992.

[35] T. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, D. Wolber, *"A network security monitor"*, IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA, USA, 1990.

[36] D. Heyman, *"Some issues in performance modeling of data teletraffic"*, Performance Evaluation, Vol. 34, pp. 227-247, 1998.

[37] D. P. Heyman and T. V. Lakshman, *"Long Range Dependence and queueing effects for VBR video"*, In 'Self-similar network traffic and performance evaluation', edited by K. Park and W. Willinger, J. Wiley & Sons, 2000.

[38] J. Hochberg, K. Jackson, C. Stallings, J. F. McClary, D. Dubois, J. Ford, *"NADIR : an automated system for detecting network intrusion and misuse"*, Computers & Security, Vol. 12, No 3, 1993.

[39] M. Y. Huang, R. J. Jasper, T. Wicks, *"A large scale distributed intrusion detection framework based on attack strategy analysis"*, Computer Networks, Vol. 31, No 23-24, 1999.

[40] K. Ilgun, *"USTAT : a real-time intrusion detection system for Unix"*, IEEE Symposium on Security and Privacy, Oackland, CA, USA, May 24-26, 1993.

[41] K. Ilgun, R. A. Kemmerer, P. A. Porras, *"State transition analysis : a rule-based intrusion detection approach"*, IEEE Trans. on Software Engineering, Vol. 21, No 3, March 1995.

[42] H. S. Javitz, A. Valdes, *"The SRI IDES statistical anomaly detector"*, ESORICS 1991.

[43] A. K. Jena, A. Popescu and P. Pruthi, *"Modeling and analysis of HTTP traffic"*, Proc. of ITC Spec. Seminar on IP Traffic Modeling, Measurement and Management, Sept. 2000.

[44] Y. F. Jou, F. Gong, C. Sargor, S. F. Wu, C. W. Rance, *"Architecture design of a scalable intrusion detection system for the emerging network structure"*, Technical report CDRL A005, Depart. of Computer Science, North Carolina State University, Releigh, NC, USA, April 1997.

[45] J.P. Kowalski, B. Warfield, *"Modelling Traffic Demand Between Nodes in a Telecommunication Network"*, ATNAC'95, 1995.

[46] S. Kumar, E. H. Spafford, *"A pattern matching model for misuse intrusion detection"*, 17th National Computer Security Conference, Baltimore, MD, USA, 1994.

[47] R.S. Krupp, *"Properties of Kruithof's projection method"*, The Bell System Technical Journal, 58, pp517-538, 1979.

[48] H. Kvarnström, *"A survey of commercial tools for intrusion detection"*, Technical report No 99-8, Chalmers University of Technology, Depart. of Computer Engineering, Sweden, 1999.

[49] F. Lau, S. H. Rubin, M. H. Smith, L. Trajkovic, *"Distributed denial of service attacks"*, IEEE Int. Conference on Systems, Man and Cybernetics, Nashville, TN, USA, Oct. 2000.

[50] W. Leland, M. Taqqu, W. Willinger and D. Wilson, *"On the self-similar nature of Ethernet traffic (extended version)"*, IEEE/ ACM Trans. on Networking, Vol. 2, No 1, pp. 1-15, 1994.

[51] B. Mah, *"An empirical model of HTTP network traffic"*, Proc. of INFOCOM '97, pp. 592-600, Apr. 1997.

[52] J. Marin, D. Ragsdale, J. Surdu, *"A hybrid approach to the profile creation and intrusion detection"*, DARPA Information Survivability Conference and Exposition, Anaheim, CA, June 12-14, 2001.

[53] S. McCreary and K. C. Claffy, *"Trends in Wide Area IP traffic patterns : A view from Ames Internet Exchange"*, Proc. of ITC Spec. Seminar on IP Traffic Modeling, Measurement and Management, Sept. 2000 (see also CAIDA Technical Report http ://www.caida.org/outreach/papers/2000/AIX0005/).

[54] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya and C. Diot, *"Traffic Matrix Estimation : Existing Techniques Compared and New Directions"*, Proc. of ACM SIGCOMM '02, Pittsburgh, PA, USA, Aug. 2002.

[55] S. Molnar and T. D. Dang, *"Scaling analysis of IP traffic components"*, Proc. of ITC Spec. Seminar on IP Traffic Modeling, Measurement and Management, Sept. 2000.

[56] D. Moore, G. M. Voelker, S. Savage, *"Inferring Internet Denial-of-Service activity"*, Usenix Security Symposium, 2001.

[57] D. Moore, C. Shannon, K. Claffy, *"Code-Red : a case study on the spread and victims of an Internet worm"*, SIGCOMM Internet Measurement Workshop (IMW'2002), Marseille, France, Nov. 2002.

[58] B. Mukherjee, L. T. Heberlein, K. N. Levitt, *"Network intrusion detection"*, IEEE Network, Vol. 8, No 3, May/June 1994.

[59] P. Mutaf, *"Defending against a Denial-of-Service attack on TCP"*, 2nd Int. Workshop on Recent Advances in Intrusion Detection (RAID'99), West Lafayette, Indiana, USA, Sept. 1999.

[60] M. Nabe, M. Murata and H. Miyahara, *"Analysis and modeling of World Wide Web traffic for capacity dimensioning of Internet access lines"*, Performance Evaluation, Vol. 34, pp. 249-271, 1998.

[61] *"Netflow Aggregation"*, Cisco IOS release 12.0(5)T.

[62] C. J. Nuzman, I. Saniee, W. Sweldens and A. Weiss, *"A compound model for TCP connection arrivals"*, Proc. of ITC Spec. Seminar on IP Traffic Modeling, Measurement and Management, Sept. 2000.

[63] P. Olivier and N. Benameur, *"Flow level IP traffic characterization"*, Proc. of ITC '17, Moreira de Souza, Fonseca and de Souza e Silva (eds.), Elsevier, Dec. 2001.

[64] K. Park, G. Kim and M. Crovella, *"On the relationship between file sizes, transport protocols, and self-similar network traffic"*, Proc. of IEEE ICNP, 1996.

[65] K. Park and W. Willinger, *"Self-similar network traffic : an overview"*, In 'Self-similar network traffic and performance evaluation', edited by K. Park and W. Willinger, J. Wiley & Sons, 2000.

[66] V. Paxson, *"Growth trends in wide-area TCP connections"*, IEEE Network, Vol. 8, No 4, pp. 8-17, July/Aug. 1994.

[67] V. Paxson, *"Empirically derived analytical models of wide-area TCP connections"*, IEEE/ACM Trans. on Networking, Vol. 2, No 4, Aug. 1994.

[68] V. Paxson, *"End-to-end Internet packet dynamics"*, IEEE/ACM Trans. on Networking, Vol. 7, No 3, pp. 277-292, June 1999.

[69] V. Paxson, *"Bro : a system for detecting network intruders in real-time"*, Computer Networks, Vol. 31, No 23-24, 1999.

[70] V. Paxson and S. Floyd, *"Wide area traffic : The failure of Poisson modeling"*, IEEE/ACM Trans. on Networking, Vol. 3, pp. 226-244, 1995.

[71] P. A. Porras, P. G. Neumann, *"EMERALD : Event monitoring enabling response to anomalous live disturbances"*, 20th Nat. Information Systems Security Conference, Baltimore, Maryland, USA, Oct. 7-10, 1997.

[72] P. A. Porras, A. Valdes, *"Live traffic analysis of TCP/IP gateways"*, ISOC Symp. on Network and Distributed Systems Security, San Diego, CA, March 11-13, 1998.

[73] L. Ricciulli, P. Lincoln, P. Kakkar, *"TCP SYN flooding defense"*, Communication Networks and Distributed Modeling and Simulation, 1998.

[74] J. Roberts, *"Engineering for Quality of Service"*, In 'Self-similar network traffic and performance evaluation', edited by K. Park and W. Willinger, J. Wiley & Sons, 2000.

[75] J. Roberts, U. Mocci and J. Virtamo (editors), *"Broadband Network Teletraffic (Final report of COST 242)"*, LNCS 1155, Springer Verlag, 1996.

[76] B. Ryu, D. Cheney and H-W Braun, *"Internet flow characterization - Adaptive Timeout and statistical modeling"*, Proc. of Passive and Active Measurement Workshop, Apr. 2001.

[77] S. Sarvotham, R. Riedi and R. Baraniuk, *"Connection-level analysis and modeling of network traffic"*, Proc. of ACM SIGCOMM Internet Measurement Workshop, Nov. 2001.

[78] C. L. Schuba, I. V. Krsul, M. G. Kuhn, *"Analysis of a denial of service attack on TCP"*, IEEE Symp. on Security and Privacy, Oackland, CA, USA, 1997.

[79] M. M. Sebring, E. Shellhouse, M. E. Hanna, R. A. Whitehurst, *"Expert systems for intrusion detection : a case study"*, 11th Nat. Computer Security Conference, Baltimore, Maryland, USA, Oct. 17-20, 1988.

[80] F. D. Smith, F. Hernandez-Campos, K. Jeffay and D. Ott, *"What TCP/IP protocol headers can tell us about the Web"*, Proc. of ACM SIGMETRICS '01, June 2001 (extended version at http ://www.cs.unc.edu/Research/dirt).

[81] O. Spatscheck, *"Defending against Denial of Service Attacks in Scout"*, 3rd OSDI Symposium, Feb. 1999.

[82] H. Spiess, *"A maximum-likelihood model for estimating origin-destination matrices"*, Transportation Research 28B, pp 395-412, 1987.

[83] A. Sridharan, S Bhattacharyya, C. Diot, R. Guerin, J. Jetcheva, N. Taft, *"On the Impact of Aggregation on the Performance of Traffic Aware Routing"*, Proceedings of ITC 17, Elsevier, pp 111-123, 2001.

[84] C. Tebaldi, M. West, *"Bayesian Inference on Network Traffic Using Link Count Data "*, Journal of the American Statistical Association, June 1998.

[85] K. Thompson, G. Miller and M. Wilder, *"Wide-area internet traffic patterns and characteristics"*, IEEE Network, Vol. 11, No 6, Nov./Dec. 1997.

[86] Ph. Tran-Gia and N. Vicari (editors), *"Impacts of new services on the architecture and performance of broadband networks (Final report of COST 257)"*, Chapter on 'Traffic measurement and data analysis', 2000.

[87] UIT-T, Recommandation E.492, *"Période de référence du trafic"*, Fév. 1996.

[88] UIT-T, Recommandation E.500, *"Principes de mesure d'intensité du trafic"*, Nov. 1998.

[89] H. S. Vaccaro, G. E. Liepins, *"Detection of anomalous computer session activity"*, IEEE Symp. on Security and Privacy , Oackland, CA, USA, May 1989.

[90] H.G Van Zuylen, L.G Willumsen, *"The most likely trip matrix estimated from traffic counts"*, Transportation Research 143, pp 281-293, 1980.

[91] Y. Vardi, *"Network Tomography : estimating source-destination traffic intensities from link data"*, Journal of the American Statistical Association, Vol 91, n433, Theory and Methods, March 1996.

[92] N. Vicari and S. Köhler, *"Measuring Internet user traffic behavior dependent on access speed"*, Proc. of ITC Spec. Seminar on IP Traffic Modeling, Measurement and Management, Sept. 2000.

[93] D. Wagner, D. Dean, *"Intrusion detection via static analysis"*, IEEE Symp. on Security and Privacy, Oackland, CA, USA, May 2001.

[94] M. West, *"Statistical Inference for Gravity Models in Ttransportation Flow Forecasting"*, Discussion Paper 94-40, IS- DS, Duke University, December 1994.

[95] G. White, V. Pooch, *"Cooperative security managers : distributed intrusion detection systems"*, Computer & Security, Vol. 15, No 5, 1996.

[96] W. Willinger, M. S. Taqqu and A. Erramilli, *"A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks"*, In F. P. Kelly, S. Zachary and I. Ziedins, eds., 'Stochastic networks : theory and applications', Clarendon Press, Oxford, UK, 1996.

[97] W. Willinger, M. S. Taqqu, R. Sherman and D. V. Wilson, *"Self-similarity through high-variability : statistical analysis of Ethernet LAN traffic at the source level"*, IEEE/ACM Trans. on Networking, Vol. 5, pp. 71-86, 1997.

[98] L.G Willumsen, *"Estimating time dependent trip matrices from traffic counts"*, Proceeding of the ninth international symposium on traffic theory, 1984.

[99] Y. Zhang, N. Duffield, V. Paxson and S. Shenker, *"On the constancy of Internet path properties"*, Proc. of ACM SIGCOMM Internet Measurement Workshop, Nov. 2001.

Chapitre 4

État de l'art sur la modélisation probabiliste du trafic TCP

Philippe Robert INRIA, Christine Fricker, INRIA, Fabrice Guillemin, France Télécom R&D

4.1 Introduction

Dans cette présentation le terme "réseau" désigne un réseau IP.

L'infrastructure

Schématiquement, le réseau se décrit comme des Sources-Destinations reliées entre elles par des *routeurs*. De façon macroscopique, c'est l'interconnexion de plusieurs types de réseaux : des sites "clients" (terminaux individuels, réseaux locaux d'entreprise, réseaux métropolitains ou plaques régionales ADSL, etc.) interconnectés par des réseaux d'opérateurs, qui peuvent eux-mêmes être supportés par des réseaux dorsaux (backbone).

Les algorithmes

1. TCP* Ce protocole de niveau 3 s'est imposé de façon quasi-universelle dans ces réseaux pour la transmission de données. Il régule (actuellement) une grande partie du trafic de données sur l'Internet et est amené à évoluer vers la transmission de flux temps réel (TCP friendly protocols). Ses principes peuvent être décrits de manière simplifiée de la façon suivante : pour transmettre, une source envoie W paquets sans s'occuper de la réponse de la destination. La destination envoie un accusé de réception pour chaque paquet reçu. Si l'un de ces paquets est perdu la variable W, appelée

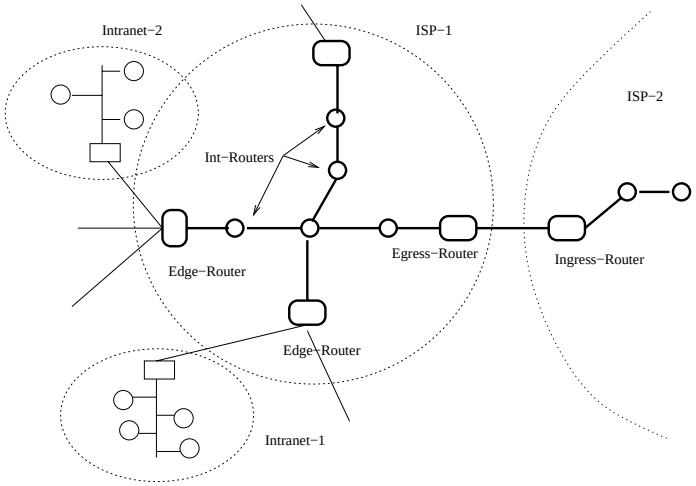

FIG. 4.1 – Une architecture de réseau

taille de la fenêtre de congestion, est divisée par deux, sinon elle est augmentée d'une unité quand tous les paquets de la fenêtre courante ont été transmis avec succès. Les pertes de paquets ont lieu si les mémoires tampons à la sortie des routeurs sont saturées ou s'il y a défaillance du réseau (ex : coupure d'un lien).

L'idée de base est très simple, la source n'augmente ses envois de paquets que de façon très progressive dans le réseau. En se rappelant que les connexions à une instant donné sont en compétition pour l'accès aux ressources de transmission, une augmentation trop rapide de la taille de la fenêtre de chacune des connexions engendrera plus rapidement des phénomènes de débordement des mémoires tampons, et par conséquent impliquera un plus grand nombre de connexions ayant des paquets perdus. (Jacobson [28]).

À l'inverse, si une perte est détectée, les envois sont réduits de manière drastique en divisant par deux la taille de la fenêtre de congestion. S'il y a une perte, cela peut être dû au fait que la mémoire tampon d'un routeur est saturée par plusieurs connexions TCP accédant au même port de sortie de ce routeur. Réduire la fenêtre de congestion de chacune de ces connexions de quelques unités ne décongestionnera pas significativement ce goulet d'étranglement. Diviser par deux la taille de la fenêtre de congestion de chacune des connexions revient à "refroidir" brutalement l'activité du réseau dans ce secteur.

Le protocole TCP est un mécanisme de contrôle de l'émission des paquets qui circulent dans le réseau. Le contrôle est fait au niveau de la source qui réduit ses envois suivant les informations de la destination qui lui parviennent ou non. Ce mécanisme a tendance à auto-réguler les envois de messages à travers le réseau. Cependant le protocole TCP pose de nombreux problèmes d'ingénierie dans le réseau. Par exemple, on peut citer

– L'intégration des flux temps réel et du trafic contrôlé par TCP (dit trafic élastique). Tous les envois de paquets qui circulent dans le réseau ne sont pas forcément contrôlés par le protocole

TCP. C'est le cas par exemple de certains flux vidéo ou audio envoyés sans contrôle de transmission (protocole UDP). Pour ces flux, il est possible de tolérer un certain taux de perte (et donc ne pas utiliser TCP) sans détérioration de la qualité de réception ou, quitte à envoyer un peu plus de paquets et avec une algorithmique adéquate, il est possible de corriger certaines pertes en utilisant l'information redondante contenue dans le flux.

– Problème de vérification de conformité d'une source au protocole TCP. Le protocole TCP lui-même suppose que toutes les sources se plient à ses mécanismes de contrôle. En cas de congestion, une source "agressive", voulant profiter des mécanismes de contrôle de perte de TCP, peut ne pas réduire la taille de sa fenêtre de congestion et continuer à envoyer ses paquets au même débit. Dans ce cas le réseau est détourné au profit des sources de ce type, la propriété d'équité qui doit prévaloir sur les nœuds du réseau n'est plus satisfaite.

Enfin le seul contrôle réel des flux TCP est effectué par les sources. Le mécanisme RED (Floyd et Jacobson [13]) a été introduit pour contrôler la diffusion des paquets au niveau des routeurs.

2. RED Les paquets qui transitent par un routeur sont stockés dans une mémoire tampon dans l'attente de leur envoi à la prochaine étape de leur route. Quand le nombre moyen L (estimé) de paquets en transit dans le routeur est en-dessous d'un seuil $\min_q$ tous les paquets sont acceptés et au-dessus du seuil $\max_q$ ils sont systématiquement refusés. Entre ces deux valeurs, les paquets sont acceptés avec une probabilité qui dépend de L.

Les avantages principaux de ce type d'algorithme sont de plusieurs ordres :

1. *Petits délais*. L'algorithme privilégie les tailles de mémoire tampon en-dessous de $\max_q$, et par conséquent cela garantit aux paquets qui sont acheminés une file d'attente petite et donc un temps de transmission faible par le routeur si $\max_q$ n'est pas trop grand. Il est facile de concevoir que le réseau puisse avoir des taux de perte très faibles si les tailles de mémoire tampon sont assez grandes. Dans ce cas si le trafic est soutenu, les mémoires tampons des routeurs seront alors largement remplies et si les pertes sont petites, les délais seront néanmoins très grands. Nombre de paquets TCP dépasseront alors leur timeout et seront donc considérés comme perdus par leur source.
2. *Possibilité de contrôle des flots "agressifs"*. Quand la taille de la mémoire tampon est au-dessus de la valeur $\min_q$, chaque paquet a une probabilité non nulle d'être rejeté. Si une source "agressive" envoie ses paquets par le routeur, elle expérimentera d'autant plus de pertes qu'elle enverra de paquets, au contraire d'une source "raisonnable" qui modulera ses envois en fonction du trafic.

Pour l'instant l'algorithme RED n'a pas le degré d'universalité du protocole TCP. Il n'est pas installé sur la majorité des routeurs. L'alternative, l'algorithme Tail Drop, qui ne rejette les paquets que lorsque la mémoire tampon est pleine, est encore majoritaire. L'utilité de RED est même contestée (voir par exemple Brandauer *et al.* [9] et May *et al.* [20]) en raison des complications algorithmiques engendrées au niveau des routeurs : pour être pleinement efficace (i.e. meilleur que l'algorithme fainéant Tail Drop) les seuils $\max_q$ et $\min_p$ doivent être optimisés. Les valeurs optimales sont malheureusement trop sensibles aux conditions de trafic qui fluctuent beaucoup dans le réseau ; elles sont difficiles à réactualiser en raison du coût algorithmique.

4.2 Les modèles probabilistes

Dans la description sommaire qui vient d'être faite d'un réseau, des composantes aléatoires interviennent principalement dans deux domaines :

4.2.1 La caractérisation du trafic

La représentation du trafic Internet a fait l'objet de nombreuses mesures et analyses (voir Allman et Paxson [2], Paxson [27] et Paxson et Floyd [27] par exemple). Il s'agit de modéliser le processus d'arrivée des paquets aux différents points du réseau : routeurs sur la frontière, nœuds internes ou encore les routeurs de sortie ("Egress-routers"). Plusieurs types de représentation sont envisageables.

1. Détermination a priori d'une enveloppe de trafic qui conduit à définir des cas pires, "worst case". Des valeurs maximales pour le débit crête et la longueur de rafale par exemple sont fixées, il s'agit de déterminer les algorithmes qui assurent un niveau de qualité de service donné avec des trafics complètement arbitraires respectant les contraintes de débit. Ce type de problématique conduit généralement à des algorithmes qui sous-utilisent le réseau, voir par exemple l'article d'El Walid *et al.* [12] pour le contrôle d'admission.
2. Caractérisation statistique. Les trafics sont définis par leurs statistiques. Ces modèles présentent l'avantage de représenter correctement un grand nombre de types de trafic, de plus des résultats qualitatifs précis peuvent être obtenus (ou du moins, envisagés) à partir des modèles mathématiques associés.
   (a) Modélisation de niveau paquet (microscopique) : trafics fractals. Si cette description est très populaire depuis les travaux de Leland *et al.* [18], elle n'a pas donné lieu, pour l'instant, à de réels développements algorithmiques spécifiques. Il est très difficile d'obtenir des résultats qualitatifs (sur les taux de débordements des mémoires tampons par exemple) avec ce type de modèle. Voir dans ce domaine les travaux de Norros sur le mouvement brownien fractionnaire [22, 23]. Actuellement, les travaux dans ce domaine se concentrent principalement sur la définition et l'estimation des paramètres caractérisant ces trafics.
   (b) Modélisation de niveau flot (macroscopique) : trafics dont les rafales ont lieu de façon aléatoire et la taille d'une rafale a une queue de distribution à décroissance lente (i.e. beaucoup de longues rafales et très grande variabilité de la taille de celles-ci).
   (c) Modélisation des processus de perte. Si les trafics poissonniens ne peuvent pas représenter globalement le trafic *au niveau des paquets* sur Internet (voir Paxson et Floyd [26]), ils peuvent cependant décrire convenablement certaines arrivées de rafales de paquets ou encore les processus de débordement des mémoires tampons des routeurs (Paxson [27]).

4.2.2 Les algorithmes probabilistes

Le mécanisme de sélection de RED utilise des variables aléatoires propres pour déterminer les paquets qui peuvent potentiellement être rejetés. Les algorithmes probabilistes sont amenés à jouer un rôle de plus en plus important dans la gestion des réseaux. L'introduction d'éléments aléatoires exogènes présente, en général, l'avantage de lisser les comportements des systèmes (cf. Motwani et Raghavan [21]). De plus, ces algorithmes permettent dans certains cas d'assurer les propriétés

d'équité qui sont très importantes dans les réseaux : voir le procédé de sélection de l'algorithme RED par exemple.

4.3 Les études des modèles probabilistes de TCP

La plupart des études du protocole TCP concernent le cas d'une très longue connexion en supposant que le taux de perte α est très faible. Il s'agit de déterminer que est le débit d'une telle connexion. Le résultat classique dans ce domaine est que la taille de la fenêtre de congestion est de l'ordre de $1/\sqrt{\alpha}$ quand α devient petit. Si (W_n) est la suite des tailles de fenêtres de congestion successives, divers modèles ont été utilisés pour représenter cette suite

- *les mesures et les simulations*
  Madhavi et Floyd [19] et Floyd [14] ont mené de telles mesures au début des années 1990.
- *Les modèles auto-régressifs à priori.*
  On suppose qu'il existe des suites i.i.d. (A_n), (B_n) telles que

$$W_{n+1} = A_n W_n + B_n,$$

  cette dépendance linéaire permet de déterminer la plupart des caractéristiques de la connexion à l'équilibre. C'est l'hypothèse faite dans les travaux d'Altman *et al.* [3, 4] et Baccelli et Hong [6] . Baccelli et Hong [7]
- *Les approximations par chaîne de Markov finie.*
  Cela concerne entre autres les travaux de Padhye *et al.* [25], Vojnović *et al.* [30] et Vojnović et Le Boudec [29]. La formule du débit de Padhye *et al.*est souvent reprise dans la littérature.
- *Les modèles fluides.*
  La variable W est supposée à valeurs continues et l'évolution de cette variable gouvernée par une équation différentielle déterministe perturbée par un processus de Poisson. Le résultat principal est celui Ott *et al.* [24], voir aussi Altman *et al.* [5] ainsi que Rubino *et al.*.
- *Les approximations de champs moyen.*
  Il s'agit ici de considérer qu'un routeur est traversé par un très grand nombre de connexions et de supposer que deux connexions prises séparément sont quasiment indépendantes. Cette approximation permet d'écrire une équation différentielle déterministe de l'évolution d'une connexion donnée. C'est l'optique d'Adjih *et al.* [1] Baccelli *et al.*et Makowski.
- *Modèles de perte de paquets*. Pour ces modèles seul le processus de perte est la composante aléatoire d'une connexion TCP. Dumas *et al.* [11] considère le comportement de la connexion quand les pertes de paquets sont faibles et indépendantes. Guillemin *et al.* [15, 16] étudie la connexion TCP avec le processus de perte mis en évidence par les mesures de Paxson (pertes corrélées). Voir aussi Bolot [8] et Yajnik *et al.* [31]. De façon remarquable, les constantes des les mesures de Floyd, du modèle de Ott et de l'approximation de champs moyen d'Adjih *et al.*sont obtenues par des théorèmes limites rigoureux de cette façon. Voir aussi Altman *et al.* [4].

D'autres aspects d'une connexion TCP ont été envisagés par Brown [10] (quand les temps d'aller et retour des différentes connexions ne sont pas identiques) et par Kelly [17].

Bibliographie

[1] Cédric Adjih, Philippe Jacquet, and Nikita Vvedenskaya, *Performance evaluation of a single queue under multi-user TCP/IP connections*, Tech. Report RR-4141, INRIA, March 2001.

[2] Mark Allman and Vern Paxson, *On estimating end-to-end network path properties*, SIGCOMM, 1999, pp. 263–274.

[3] Eitan Altman, Konstantin Avrachenko, and Chadi Barakat, *A stochastic model of TCP/IP with stationary random losses*, ACM-SIGCOMM'00 (Stockholm), no. 4, 2000, pp. 231–242.

[4] Eitan Altman, Konstantin Avrachenko, Chadi Barakat, and R udesindo Nuñez-Queija, *TCP modeling in presence of nonlinear window growth*, ITC'17 (Salvador da Bahia), 2001.

[5] Eitan Altman, Kostya Avrachenkov, Chadi Barakat, and Rudesindo Nunez Queija, *State-dependent M/G/1 type queueing analysis for congestion control in data networks*, Proceedings of IEEE INFOCOM (Anchorage, Alaska), April 2001 2001.

[6] François Baccelli and Dohy Hong, *TCP is max-plus linear*, ACM-SIGCOMM'00 (Stockholm), no. 4, 2000, pp. 219–230.

[7] ———, *AIMD, fairness and fractal scaling of TCP traffic*, Tech. Report 4155, INRIA, Domaine de Voluceau, Rocquencourt B.P.105, 78153 Le Chesnay Cedex, April 2001.

[8] Jean Bolot, *End-to-end packet delay and loss behavior in the internet*, ACM Sigcomm '93 (San Fransisco, CA) (ACM, ed.), September 1993, pp. 289–298.

[9] C. Brandauer, T. Ziegler, G. Iannaccone, C. Diot, S. Fdida, and M. May, *Comparison of tail drop and active queue management performance for bulk-data and web-like internet traffic*, 6th IEEE Symposium on Computers and Communications, 2001.

[10] Patrick Brown, *Resource sharing of tcp connections with different round trip times*, Proc. IEEE Infocom (Tel-Aviv, Israel), March 2000.

[11] Vincent Dumas, Fabrice Guillemin, and Philippe Robert, *A Markovian analysis of Additive-Increase Multiplicative-Decrease (AIMD) algorithms*, Advances in Applied Probability **34** (2002), no. 1, –.

[12] A. Elwalid, D. Mitra, and R.H. Wentworth, *A new approach for allocating buffers and bandwidth to heterogeneous regulated traffic in an ATM node*, IEEE Journal on Selected Areas in Communications **13** (1995), no. 6, 1115–1127.

[13] S. Floyd and J. van Jacobson, *Random early detection gateways for congestion avoidance*, IEEE/ACM Transactions on Networking **1** (1993), no. 4, 397–413.

[14] Sally Floyd, *Connections with multiple congested gateways in packet-switch ed networks part 1 : One way traffic*, Computer Communication Review **21** (1991), no. 5, 30–47.

[15] Fabrice Guillemin, Philippe Robert, and Bert Zwart, *Performance of TCP in the presence of correlated packet loss*, 15th ITC Specialist Seminar on Internet Traffic Engineering and Traffic Management (Wurzburg), July 2002.

[16] _____, *AIMD algorithms and exponential functionals*, Annals of Applied Probability (2003), To appear.

[17] F.P. Kelly, *Effective bandwidths at multi-type queues*, Queueing Systems, Theory and Applications **10** (1991), no. 1-2, 5–15.

[18] W.E. Leland, W. Willinger, M.S. Taqqu, and D.V. Wilson, *On the self-similar nature of ethernet traffic*, Computer Communication Review **23** (1993), no. 4, 189–193.

[19] J. Madhavi and S. Floyd, *TCP-friendly unicast rate-based flow control*, End2end-interest mailing list, January 1997.

[20] M. May, J. Bolot, C. Diot, and B. Lyles, *Reasons not to deploy red*, Proc. of 7th. International Workshop on Quality of Service (London), June 1999, pp. 260–262.

[21] Rajeev Motwani and Prabhakar Raghavan, *Randomized algorithms*, Cambridge University Press, Cambridge, 1995.

[22] I. Norros, *A storage model with self similar input*, Queueing Systems, Theory and Applications **16** (1994), 387–396.

[23] _____, *On the use of fractional brownian motion in the theory of connectionless networks*, IEEE Journal on Selected Areas in Communications **13** (1995), 953–962.

[24] Teunis J. Ott, J.H.B. Kemperman, and Matt Mathis, *The stationary behavior of ideal TCP congestion avoidance*, Unpublished manuscript, August 1996.

[25] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, *Modeling TCP throughput : A simple model and its empirical validation*, IEEE/ACM Transactions on Networking **8** (2000), 133–145.

[26] Vern Paxson, *End-to-end internet packet dynamics*, IEEE/ACM Transactions on Networking **7** (1999), no. 3, 277–292.

[27] Vern Paxson and Sally Floyd, *Why we don't know how to simulate the internet*, Winter Simulation Conference, 1997, pp. 1037–1044.

[28] J. van Jacobson, *Congestion avoidance and control*, SIGCOMM '88 Symposium : Communications Architectures and Protocols, ACM, August 1988.

[29] M. Vojnović and J.-Y. Le Boudec, *Some observations on equation-based rate control*, Proc. ITC'17 (Salvador de Bahia, Brazil), 2001.

[30] M. Vojnović, J.-Y. Le Boudec, and C. Boutremans, *Global fairness of additive-increase and mutliplicative-decrese with heteregeneous round trip times*, IEEE Infocom (Tel Aviv, Israel), March 2000.

[31] M. Yajnik, J. Kurose, and D. Towsley, *Packet loss correlation in the MBone multicast network experimental measurements and Markov chain models*, Tech. Report UM-CS-1995-115, University of Massachusetts, Amherst, 1995.

Chapitre 5

Outils théorique pour l’échantillonage

Table des figures

Liste des tableaux