Visualisation de trafic de réseau en temps réel

par

Meryem ELBAHAM

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE CONCENTRATION : RÉSEAUX DE
TÉLÉCOMMUNICATIONS
M. Sc. A.

MONTRÉAL, LE 15 JUIN 2017

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Mohamed Cheriet, directeur de mémoire
Département de génie de production automatisée à l'École de technologie supérieure

M. Alain April, président du jury
Département de génie logiciel et des technologies de l'information (TI) à l'École de technologie supérieure

M. Marco Pedersoli, membre du jury
Département de génie de production automatisée à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 17 MAI 2017

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens, tout d'abord, à remercier vivement le directeur de ma thèse, Monsieur Mohamed Cheriet, Professeur titulaire du département de génie production automatisé à l'École de Technologie Supérieure et directeur du laboratoire Synchromedia, pour ses conseils et ses directives, pour le temps qu'il a toujours consacré dans le but de satisfaire et répondre à mes questions ainsi que pour l'honneur qu'il m'a fait en acceptant de m'encadrer. Qu'il me soit permis de lui exprimer ma sincère et profonde reconnaissance.

Mes remerciements s'adressent, également, à Monsieur Kim Nguyen, Professeur assistant à l'École de Technologie Supérieure et associé de recherche au laboratoire Synchromedia, pour son encadrement, son soutien, ses conseils et ses efforts qui m'ont inspirés de mener à bien ces travaux de recherche.

Je remercie, aussi, les membres du jury, Messieurs Alain April et Marco Pedersoli professeurs à l'École de Technologie Supérieure, pour leur participation à l'évaluation de ce travail.

Je remercie affectueusement ma tendre mère Fatima pour ses sacrifices illimités, et à ma belle-mère Amina pour son soutien et ses encouragements. Je remercie, par la même occasion, mon oncle Mohamed pour son soutien continu et ses conseils précieux. Mes gratitudes s'adressent à ma petite famille, mon mari Rabie et ma fille Arwa, qui m'ont appuyés tout au long de cette recherche.

Finalement, je tiens à exprimer également mes remerciements à la famille du laboratoire Synchromedia, pour leur esprit d'équipe, leur appui, leurs conseils et d'avoir fait de cette durée de recherche une période de partage agréable.

À toutes les personnes qui ont contribuées de près ou de loin dans la réalisation de ce travail.
Merci à tous !

VISUALISATION DE TRAFIC DE RÉSEAU EN TEMPS RÉEL

Meryem ELBAHAM

RÉSUMÉ

Les avancées technologiques et l'évolution rapide d'Internet et des réseaux informatiques, y compris les réseaux d'entreprises, rendent nécessaire l'implémentation d'une stratégie de gestion de réseau afin de pouvoir optimiser l'utilisation des ressources, planifier le dimensionnement des infrastructures et opérer les systèmes de qualité de service et les mécanismes de sécurité. Ainsi, et pour atteindre cet objectif, il est indispensable de connaître le trafic véhiculant à travers le réseau. Cependant, une telle évolution a engendré une augmentation continue dans le volume de trafic et par conséquent la quantité de données à traiter.

L'utilisation de la métrologie et des outils traditionnels, tels que l'analyse les journaux de données, des tableaux ou encore des outils qui listent les paquets échangés entre les hôtes, ne répondent plus aux besoins actuels en termes d'exploration de données et ne permettent pas aux utilisateurs d'en tirer de l'information pertinente pour la prise de décision.

Dans ce mémoire, l'objectif est de concevoir et développer une plateforme de visualisation qui permet de résoudre la problématique d'exploration des données réseau ou trafics IP pour la surveillance des réseaux. Ceci nécessite de faire ressortir l'information disponible dans les données multidimensionnelles, et de grande taille de manière à rapporter à l'utilisateur l'état du réseau surveillé en temps réels. Pour y parvenir, il est proposé de concevoir un ensemble de méthodes pour le traitement, le filtrage, la classification et la visualisation de l'information. La visualisation nécessite un processus de traitement de données dont la finalité est la production de représentations graphiques de l'information utile dans des graphes simples et expressifs.

La plateforme de visualisation de trafic étudié dans ce document est conçue de manière à offrir trois niveaux complémentaires d'information. Le premier niveau est dédié à décrire le réseau de manière générale à travers l'analyse des volumes de trafic, les connexions établies et la distribution de la taille des paquets en temps réel. Le deuxième niveau permet d'analyser du trafic au niveau transport; il offre une carte de répartition de flux entre les différents hôtes (interne et externe) et une analyse de flux de point de vue des activités des ports dans l'objectif de détecter les flux malicieux. Un troisième niveau permet d'identifier et classifier les applications en temps réel en utilisant l'apprentissage machine. Des méthodes d'échantillonnage de données ont été développées afin de réduire le coût de traitement et assurer l'analyse en temps réel.

Un test de fonctionnement a été effectué afin de valider les différentes fonctionnalités d'analyse offertes par la plateforme proposé. De la même façon, un ensemble de mesures de performance ont été examinées et les résultats démontrent que la solution est plus performante en termes de taux de mémoire et de CPU utilisé en comparaison avec d'autres

applications de visualisation telles que TNV. En outre, la présente plateforme de visualisation permet de rapporter l'état du réseau en temps réel, ce qui permet une surveillance des systèmes de qualité de service et de sécurité en temps réel.

Mots clés: Apprentissage machine, classification du trafic, échantillonnage de trafic, visualisation du trafic.

REAL TIME NETWORK TRAFFIC VISUALIZATION

Meryem ELBAHAM

ABSTRACT

Technological advances and the rapid evolution of the Internet and computer networks, including enterprise networks, make it necessary to implement a network management strategy in order to optimize the use of resources, plan infrastructure sizing and Support quality-of-service systems and security mechanisms. To achieve this objective, it is essential to know the traffic conveying through the network. However, such evolution has led to an increase in the volume of traffic and consequently the amount of data to be explored.

Metrology and traditional tools such as the analysis of log files, tables or tools that list packets exchanged between machines do not meet new requirements on data mining and do not allow users to derive relevant information from a huge raw dataset for decision-making purposes.

In this thesis, we design and develop a visualization framework that explores network data or IP traffic to monitor various aspects of a network. This information used in multidimensional data in large quantity so as to report to the user the state of the network to monitor in real time. To achieve this, we have investigated a set of methodological approaches to visualize network data, which is a data processing process aiming at graphically present useful information in simple and expressive graphs.

The research framework proposed in this thesis provides three additional levels of information. The first level overviews the network through analytics of traffic volumes, the number of connections and the distribution of packet size in real time. The second analyzes traffic at the transport level mapping internal and external flows between the different machines, and analyzing port-based traffic to detect malicious flows. A third level allows identifying and classifying applications in real time using machine learning. Data sampling methods were used to reduce the processing cost ensure real-time analysis.

A functional test was carried out validating the features offered by the proposed framework. In the same way, a set of performance metrics were examined and the results show that our solution is more efficient in terms of memory rates and CPU used in comparison with a typical traffic visualization application called TNV. In addition, the proposed visualization framework makes it possible to report network states in real time, which enables quality of service and security.

Keywords: Machine learning, traffic classification, traffic sampling, and traffic visualization.

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ACP	Analyse en Composantes Principales
CoS	Class of service
CPU	Central Processing Unit
CFS	Correlation based Feature Selection
CSG	Commun Substring Graphs
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DPI	Deep Packet Inspection
FCS	Frame Check Sequence
FS	Feature Selection
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
IMAP	Internet Message Access Protocol
IP	Internet Protocol
MV	Machine View
NETMAT	Network Management, Analysis and Testing Environment
NFV	Network functions Virtualization
OSI	Open Systems Interconnection
OVS	Open vSwitch
P-SVM	Proximal Support Vector Machine
P2P	Peer-to-Peer

QoS	Quality of Service
RTA	Radial Traffic Analyzer
SDN	Software-Defined Networking
SDN	Software-Defined Networking
SFLOW	Sampled Flow
SMTP	Simple Mail Transfer Protocol
SMV	Small Machine View
SNMP	Simple Network Management Protocol
SPI	Stochastic Packet Inspection
SSH	Secure Shell
SSL	Secure Sockets Layer
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TDG	Traffic Dispersion Graph
UDP	User Datagram Protocol
VM	Virtual Machine
VoIP	Voice over IP

CHAPITRE 1

INTRODUCTION GÉNÉRALE

1.1 Contexte

Depuis leur apparition dans les années 60, les réseaux d'interconnexions ont connu une évolution rapide autant au niveau de la taille qu'au niveau de la complexité. De nos jours, l'Internet est devenu le média de communication universel et omniprésent pour la transmission de tous les types de données (Lime, 2015). Aujourd'hui, il doit offrir des services plus matures et adaptés aux exigences des applications et il transmet des données hétérogènes, en particulier avec la nouvelle tendance technologique qui se manifeste comme étant la convergence des technologies de l'information et de télécommunication. Cette convergence a donné naissance aux réseaux tout IP, notamment les réseaux de nouvelle génération qui transmettent à la fois la voix (haute priorité) et les données (basse priorité). De ce fait, l'amélioration des services d'Internet et la différentiation des applications sont indissociables pour la compréhension et de la caractérisation du trafic IP.

En parallèle, plusieurs types de réseaux ont vu le jour, entre autres, les réseaux d'entreprise et de centres de données. L'émergence de ces réseaux, que ce soit à grandes échelles comme les centres de données ou les réseaux moyens comme les réseaux d'entreprise engendre de nouveaux défis concernant la sécurité, l'optimisation d'utilisation des ressources, les performances réseau, la qualité de service, etc. Par conséquent, la surveillance et le contrôle des réseaux deviennent indispensables, surtout avec l'abondance et la diversité des données y transitant. Cependant, une méconnaissance du trafic occasionne sans doute une mauvaise gestion des réseaux (Géraldine, 2012). C'est dans ce cadre que les techniques de métrologie des réseaux ont été développées avec l'objectif de caractériser et analyser le trafic réseau.

Dans le domaine de la métrologie des réseaux, connue aussi sous le nom de la science de la mesure appliquée aux réseaux informatiques, est apparue dans les années 2000 (LARRIEU, 2005). Elle consiste en un ensemble de techniques visant à mesurer et à analyser le trafic, en

vue d'évaluer les performances et de permettre de comprendre les comportements des réseaux soumis à différentes applications. Elle joue un rôle important non seulement dans la modélisation du trafic, mais aussi dans la conception de stratégies de gestion des réseaux. C'est pour cette raison qu'elle suscite l'intérêt des chercheurs du domaine des réseaux de télécommunications, et elle est largement utilisée dans tous les travaux qui impliquent la mesure et l'analyse du trafic IP.

Avec l'évolution rapide des réseaux autant en termes de taille que de volume de trafic acheminé, l'utilisation traditionnelle des mesures, que ce soit passif ou actif, ne permet pas d'explorer facilement les données et de surveiller l'état des réseaux. Le couplage de la métrologie et des techniques de visualisation semblent être une solution prometteuse pour adresser ce problème. En adoptant cette approche, il pourrait être possible d'exploiter les capacités visuelles de l'être humain pour explorer les données volumineuses issues des réseaux. Les techniques de la visualisation de l'information permettent de surmonter les difficultés liées à la lecture, à l'analyse et à l'interprétation des données en grande quantité (sous forme de texte ou tableau de grande taille).

La visualisation de l'information est un processus ayant pour finalité la représentation graphique des données dans le but d'identifier les tendances et les corrélations qui ne peuvent pas facilement être perceptibles dans les données brutes ou textuelles. C'est la raison pour laquelle ces techniques sont devenues un outil d'exploration des données populaires.

1.2 Problématique

La problématique principale traitée dans cette recherche porte sur l'importance de la visualisation du trafic de réseaux, qui est une activité importante, en raison de la forte utilisation de l'Internet pas les entreprises et les enjeux important tels que la maitrise et la gestion des réseaux.

En effet, la complexité grandissante des réseaux due à leur évolution rapide, leur étendue et de la quantité croissante et la variété des types des données qu'ils acheminent provoquent de nouvelles problématiques de surveillance. Ainsi pour y parvenir efficacement, il est indispensable de connaître le comportement des réseaux vis-à-vis des trafics qu'ils transmettent, d'où la nécessité de les caractériser et de l'analyser. Les travaux de recherche portant sur la surveillance des réseaux continuent à faire face à plusieurs difficultés dues à la complexité d'exploration des données de trafic et, de ce fait, le besoin de résoudre des problèmes en la matière demeure parmi les préoccupations des chercheurs de ce secteur.

Étant donné qu'il y a une multitude de domaines de recherche qui étudient les limitations des services sur Internet, cette recherche se focalise sur la visualisation pour la surveillance des réseaux. La surveillance des réseaux représente la solution la plus générique pour maîtriser et gérer le mieux possible les infrastructures réseautiques.

Dans le même ordre d'idée, la collecte, le stockage et l'accès aux données sont devenus aujourd'hui courants, particulièrement celles qui sont générées par les réseaux informatiques, soient le trafic et les paramètres réseau, tandis que la capacité humaine à les gérer, les comprendre et les interpréter reste problématique. En effet, face à la quantité de données à analyser, qui sont plus abondantes qu'auparavant et qui continuent à croître, la prise de décision est un défi croissant. Aussi, il s'avère que les outils traditionnels, tels que les rapports et les tableaux sont inadaptés à répondre au défi lié aux données massives. Par conséquent, la capacité perspective visuelle humaine s'impose comme étant la solution adéquate pour envisager ce dilemme soit les données volumineuses et la capacité humaine de les gérer, d'où l'intérêt de la visualisation des données.

Quoique l'application des techniques de visualisation à l'exploration des données réseau ne soit pas un nouveau thème, la majorité des travaux effectués jusqu'à présent ne couvrent pas tous les aspects essentiels de la supervision des infrastructures réseau. Ils sont personnalisés et ne s'appliquent qu'à des situations bien particulières, telle que la sécurité (Guimarães, Freitas, Sadre, Tarouco, et Granville, 2015). Pour pallier cette lacune, cette recherche vise à

solutionner cette problématique et propose une solution générique qui vise à donner à l'utilisateur l'accès aux maximums d'informations concernant l'état du réseau d'une manière visuelle, simple et efficace.

Ainsi, et pour adresser cette problématique à l'aide des techniques de visualisation de la supervision des réseaux ; le problème global a été subdivisé en sous-problèmes:

Problématique 1 : Le saisi de données massives

L'Internet compte aujourd'hui un peu plus de 3 billions d'utilisateurs selon le rapport « Global Internet Report 2015 » publié par Internet Society (InternetSociety, 2015). De plus, selon une étude axée sur l'évolution du trafic IP effectuée par Cisco, comme le montre la Figure 1.1, le volume de trafic atteindra environ 168 Exaoctet/mois en 2019 soit à peu près le triple du trafic de 2014 qui a été d'environ 59.9 Exaoctet/mois, ce qui représente un taux de croissance annuel de 23 %. Ces quantités croissantes du trafic véhiculées sur les réseaux constituent un défi très important de cette recherche. Comment s'assurer d'une visualisation de trafic en temps réel?

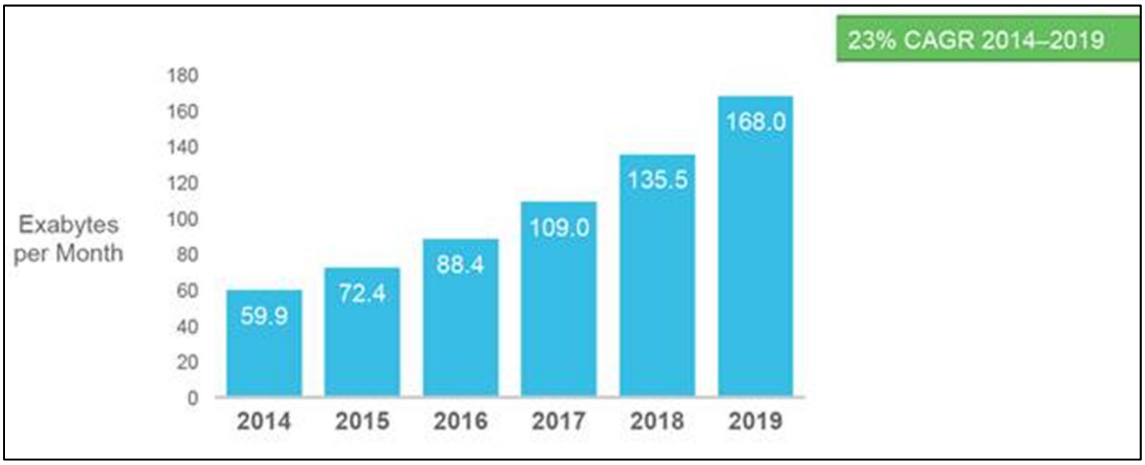

Figure 1.1 Évolution du volume de trafic Internet
Tirée de Cisco (2016b)

Problématique 2 : La visualisation de données multidimensionnelles

L'exploration des données multidimensionnelles devient complexe quand la dimension et le nombre des enregistrements augmentent. Les données de réseau, composées principalement de trafic IP sont massives et multidimensionnelles. Elles sont collectées sur les réseaux en grandes quantités et définies par plusieurs variables. Par exemple l'en-tête des paquets IP est définie par plusieurs champs tels que les adresses sources et de destinations, les ports, le protocole, et bien d'autre.

Aussi, et en plus de la perception humaine qui ne peut concevoir à la limite que trois dimensions s'ajoute la contrainte des représentations graphiques des données dans l'espace usuel qui ne permettent pas de simuler que trois dimensions et moins. Pour présenter plus que trois dimensions, il est nécessaire d'introduire d'autres éléments comme la couleur et les icônes. De ce fait, la visualisation des données multidimensionnelles est une problématique de cette recherche. Comment traiter et visualiser d'une informative les données multidimensionnelles sans surcharger les graphes?

Problématique 3 : La visualisation des données multidimensionnelles en temps réel

La surveillance en temps réel des réseaux permet l'évaluation immédiate de ses performances. C'est ainsi qu'un mécanisme de surveillance doit permettre à l'utilisateur de superviser le réseau en permanence et d'y appliquer des actions correctives le plus vite possible à la suite de la détection d'un problème. De ce fait, la visualisation en temps réel est une caractéristique recherchée et une problématique adressée par cette recherche.

La réalisation d'un tel objectif n'est pas une tâche triviale car il dépend, en premier lieu, des quantités massives de données à traiter, et de plus le processus de visualisation se décline en plusieurs étapes de traitement, ce qui entraîne une vraie appréhension de temps d'exécution. Donc, comment s'assurer d'un processus efficace de visualisation?

1.3 Les objectifs de la recherche

L'objectif principal de cette recherche est de concevoir et de développer une plateforme de visualisation du trafic, en vue de superviser et de contrôler les infrastructures de réseau d'une manière efficace. Le mécanisme proposé doit être en mesure d'offrir à l'utilisateur un certain niveau de visibilité du réseau qui va lui permettre de prendre des décisions adéquates au bon moment. Plus précisément, il doit couvrir plusieurs aspects, dont la caractérisation du trafic, une vision sur la qualité de service offerte et la consommation des ressources.

L'approche de la solution consiste à décrire le mieux possible le comportement du réseau par rapport au trafic par le biais des techniques de visualisation des données collectées sur un ou plusieurs nœuds réseaux avec l'objectif d'identifier les sources de perturbations, les flux gourmands en bande passante ou encore flux éléphants, la charge des liens, le taux d'utilisation des serveurs. Les représentations graphiques, facilitent l'analyse et l'interprétation de ces données. En particulier, à la prise des décisions de type déroutage du trafic, mise en place d'un système d'équilibrage de charge ou changement des priorités pour des fins de qualité de service, de sécurité, etc.

En vue de mener à bien cet objectif principal, la solution proposée doit satisfaire quelques exigences traduites en objectifs spécifiques de cette recherche:

Objectif 1 : La visualisation en temps réel

Le comportement des systèmes doit être connu à tout moment afin d'en assurer le bon fonctionnement et caractériser les états normaux en effectuant un suivi régulier et en prenant des actions préventives ou/et correctives afin d'éviter tout incident. Dans le cas d'un réseau informatique, cette analyse revient à surveiller et à contrôler un ensemble d'éléments tels que le trafic, le débit, les types d'applications transportées par le réseau en tout temps, ce qui n'est pas facile si l'on ne dispose pas d'outils appropriés.

Des travaux axés sur la modélisation du trafic ont démontré que ce dernier est devenu variable et instable à cause de sa dynamicité croissante (Olivier, 2012), (Khakpour et Liu, 2009). De ce fait, le réseau peut subir des perturbations et des surcharges imprévues qui peuvent occasionner la congestion et avoir d'autres conséquences très néfastes sur les niveaux de la sécurité et de la qualité de service offerte. En effet, le système de gestion du réseau ou l'administrateur réseau doit être conscient des changements des différents paramètres de surveillance en permanence pour qu'il puisse intégrer des modifications correctives au moment opportun. Pour cela, la solution doit être capable de détecter et de rapporter l'état du réseau en temps réels.

Objectif 2 : Restitution fidèle de l'information fournie par des graphes

Étant donné que les données brutes subissent des transformations et des traitements avant d'être représentées graphiquement dans le processus de visualisation de données, il est très important de garder un certain niveau d'informations dans les graphes permettant de faire des analyses et des interprétations d'une manière correcte.

Bien que la visualisation soit une solution à la fois prometteuse et nécessaire à la bonne compréhension des données, la restitution fidèle des informations contenues dans les données demeure une préoccupation d'une grande importance dans un tel processus. De ce fait, il faut prendre en considération cette exigence en produisant des représentations graphiques informatives toute en restituant les informations pertinentes contenues dans les données collectées.

Objectif 3 : La visualisation interactive

Les représentations graphiques des données, qu'elles soient statiques ou simples, permettent une compréhension facile et rapide. Toutefois, dans certaines situations ce type de graphe est inadapté et peut entraîner des risques de mauvaise compréhension à cause du manque et de l'insuffisance des informations complémentaires ou des détails d'un niveau plus élevé. Ainsi,

l'interactivité constitue une caractéristique importante dans les systèmes de visualisation. En effet, un système de visualisation doit maximiser le contenu informatif dans les graphes d'un côté et permettre à l'utilisateur d'explorer les données d'une manière interactive par le biais des éléments, tels que le zoom et les filtres afin de mettre en évidence des détails sur demande

Afin de tirer le maximum d'information de la représentation graphique, un certain degré d'interactivité est requis. C'est dans l'objectif de livrer à l'utilisateur final une solution satisfaisante, qui va lui donner l'accès aux différentes informations contenues dans les représentations graphiques et lui permettre par la suite une meilleure exploration des données, que nous prêtons une grande attention à la visualisation interactive.

Objectif 4 : La classification temps réel du trafic

La connaissance du trafic est un élément clé dans tous les systèmes de gestion de réseau et dans les mécanismes de qualité de service. C'est pourquoi il revient à identifier et à classifier le trafic pour pouvoir mettre en place ce type de fonctionnalité. Ce dernier besoin constitue une motivation pour se fixer un objectif de classification de trafic temps réel. le but est de pouvoir intégrer cette fonctionnalité dans la solution proposée en rapportant les informations au niveau d'applications issue de la classification dans des représentations graphiques en temps réel.

1.4 Plan du mémoire

Le mémoire de recherche est organisé en quatre chapitres suivis et une conclusion générale tel qu'illustré à la Figure.1.2.

Le premier chapitre est une introduction générale, qui met l'accent sur le contexte du thème de recherche, la problématique et les objectifs à atteindre.

Le deuxième chapitre présente une revue de littérature sur la visualisation du trafic en grande quantité en temps réel. Ce chapitre se divise en trois sections. La première est consacrée à la visualisation du trafic et met en évidence un ensemble de notions et de définitions relatives à ce sujet, les techniques de visualisation, la difficulté due au traitement des données multidimensionnelles en grande quantité et un résumé des systèmes de visualisation de trafic existants. La deuxième section porte sur les approches de la classification du trafic et sur l'état de l'art. La troisième section présente l'intérêt de l'utilisation des techniques d'échantillonnage pour réduire les données traitées et par conséquent contribuer à la conception d'une solution temps réel.

Le troisième chapitre se focalise sur la méthodologie suivie pour réaliser cette recherche. Il décrit les méthodes et les approches adoptées pour concevoir la plateforme de visualisation de trafic proposé dans ce mémoire.

Le quatrième chapitre porte sur l'analyse des performances du cadre du travail proposé. Ce chapitre est composé d'une première section décrivant de l'environnement de test, de la technologie, des méthodes et outils utilisés et une deuxième section dédiée aux résultats obtenus et leur interprétation. Enfin, une étude comparative a été effectuée pour valider la solution proposée.

La conclusion générale synthétise le travail réalisé dans ce mémoire et présente les recommandations ainsi que les perspectives éventuelles de cette recherche pour des travaux futurs.

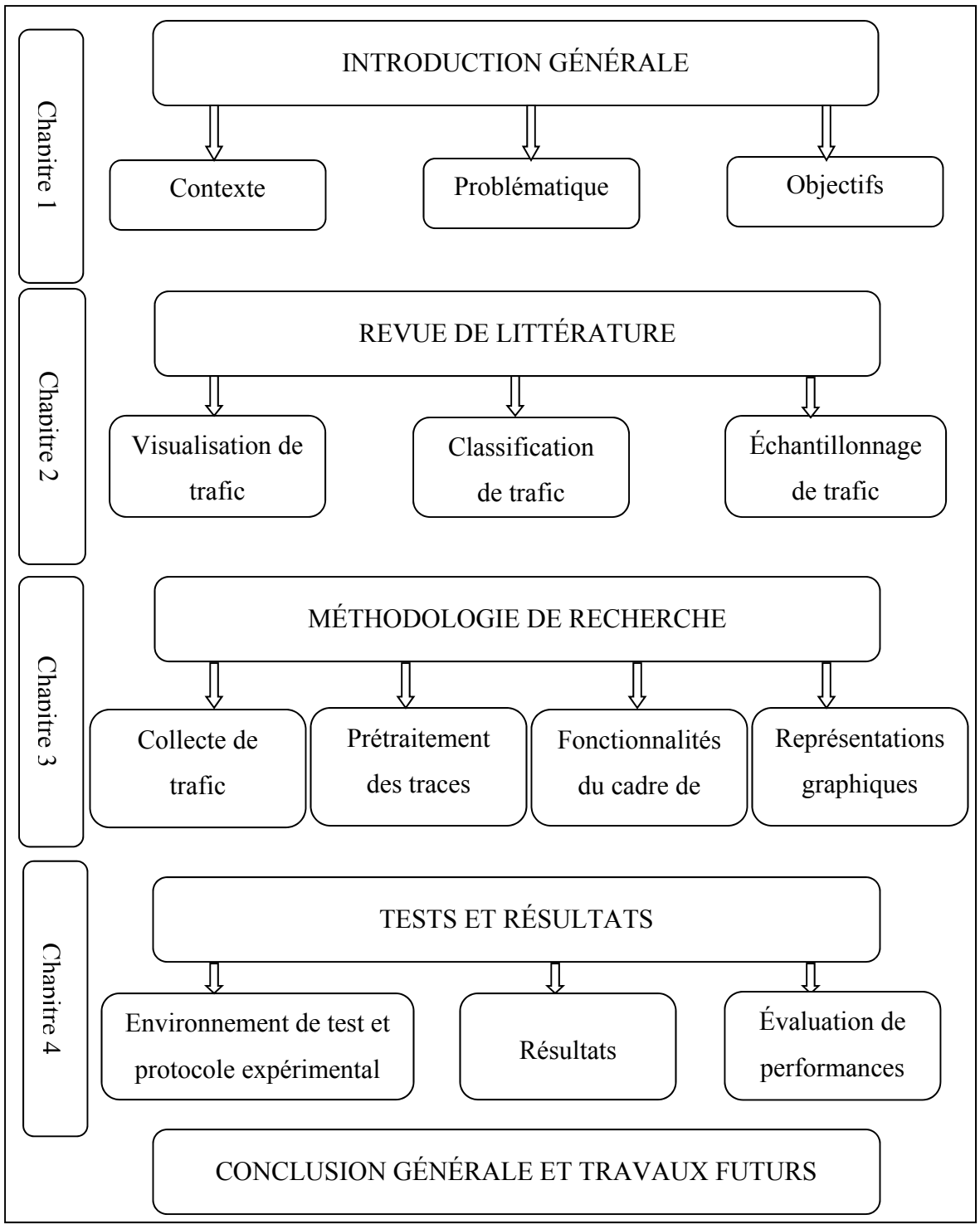

Figure 1.2 Organigramme de l'organisation du mémoire

CHAPITRE 2

REVUE DE LA LITTÉRATURE

2.1 Introduction

Ce chapitre présente une revue de la littérature portant sur des notions principales et les travaux se rapportant à la thématique de cette recherche; soit la visualisation du trafic réseau plus particulièrement le trafic IP. La première section présente un aperçu des notions et des définitions relatives à la visualisation et ses techniques d'une manière générale, et de la difficulté reliées au traitement de données multidimensionnelles en grande quantité ainsi que les méthodes possibles pour surmonter ces difficultés. Elle résume, également, les principaux travaux portant sur l'analyse et la découverte des patrons dans le trafic, avec l'objectif de supporter la gestion des réseaux et comprendre leurs comportements, par le biais de la visualisation. La deuxième section regroupe les travaux effectués dans le domaine de l'identification, la caractérisation et la classification du trafic et présente aussi un aperçu des notions de base relatives à ce domaine de recherche. Une analyse critique est effectuée dans le but de comparer les différentes solutions et d'en souligner les avantages et les limites éventuelles.

2.2 Visualisation du trafic

2.2.1 Définitions

La visualisation est largement déployée pour explorer et analyser les données complexes. Elle s'efforce d'exploiter les capacités du système de perception humain, très sophistiqué, et spécifiquement adaptées à repérer des modèles visuels pour interpréter une grande quantité de données. Elle ne se limite pas à l'affichage d'un graphe ou d'une image, mais elle peut être vue comme un processus ayant pour finalité la représentation graphique de données abstraites,avec le but d'identifier les tendances et les corrélations qui pourraient passer inaperçues en regardant seulement les données brutes ou textuelles.

L'utilisation des images, pour explorer les données scientifiques, remonte au 11^{e} siècle (Aigner, Miksch, Schumann, et Tominski, 2011). Dès lors, des travaux se sont succédés afin de répondre au besoin en termes de visualisation et produire des cartes et des images plus parlantes. Cependant, ce n'est que récemment que la visualisation est devenue une discipline de recherche indépendante. En effet, en 1987, une première définition de la visualisation a été introduite par McCormick (McCormick, A.DeFanti, et D.Brown, 1987) : « *Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights* ». Le but de cette nouvelle discipline est d'intégrer les capacités de la perception visuelle humaine avec la puissance grandissante du calcul des ordinateurs dans le but d'aider les utilisateurs dans l'analyse, la compréhension, l'interprétation et la communication des données.

2.2.2 Processus de visualisation

Pour formaliser le processus de la visualisation, plusieurs modèles ont été proposés. En 1990, Haber et McNabb (Haber et McNabb, 1990) ont introduit leur stratégie de visualisation appelée «pipeline» de visualisation qui compte trois blocs (Figure 2.1):

Filtrage : L'étape de filtrage prépare les données d'entrée brutes pour le traitement par l'intermédiaire du reste des étapes du pipeline. Elle ne s'intéresse pas uniquement à la sélection de données pertinentes, mais aussi à des opérations d'enrichissement de données, d'interpolation, de nettoyage des données, de regroupement, et de réduction de dimension.

Mappage : Cette étape permet de mapper les données préalablement préparées et filtrées à des variables visuelles. Cette opération s'avère de grande importance du fait qu'elle peut influencer l'efficacité et le raffinement des représentations graphiques. Si les variables visuelles sont mal choisies dans le sens où le résultat peut être difficile à analyser et vice versa.

Rendu : Le rendu permet de générer le graphe à partir des variables visuelles issues du bloc.

Mappage : C'est à ce niveau qu'on peut faire appel aux différentes techniques de visualisation proprement dite.

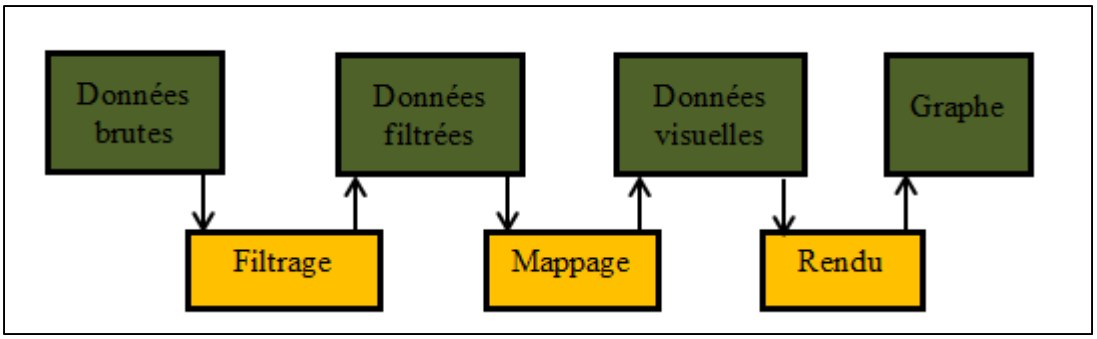

Figure 2.1 Pipeline de visualisation
Adaptée de Aigner *et al.* (2011)

En 2004, une version affinée du pipeline susmentionné a été proposée par Dos Santos et Brodlie (Dos Santos et Brodlie, 2004) afin de répondre aux exigences de la visualisation multidimensionnelles. Ils proposent l'ajout de l'étage de filtrage dans le pipeline d'origine en deux nouveaux blocs distincts, le premier connu sous le nom du bloc d'analyse ou «*analysis*» et l'autre sous le nom du bloc filtrage (Figure.2.2). Le bloc d'analyse permet d'effectuer un ensemble d'opérations, par exemple : la classification et l'interpolation. Le filtrage se focalise, dans ce nouveau modèle sur le filtrage et la sélection des informations pertinentes pour qu'elles soient représentées ou visualisées.

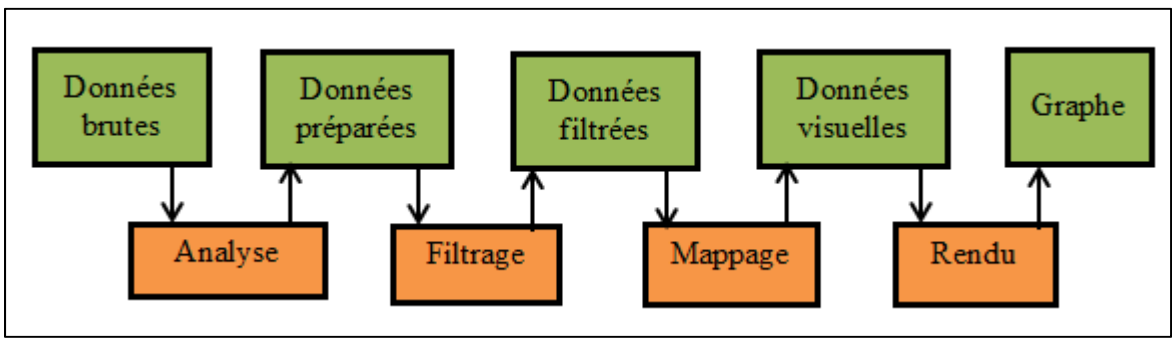

Figure 2.2 Pipeline de visualisation amélioré
Adaptée de Aigner *et al.* (2011)

Un autre modèle nomé «*data state reference model*» a été dérivé du pipeline de référence par Chi (Chi, 2000). Il s'appuie sur deux types d'opérateurs pour transformer progressivement les données brutes en image à travers quatre étapes. En effet, les opérateurs de transformation assurent les transformations des données d'un niveau d'abstraction à un autre tandis que les opérateurs ou «*stage operators*» traitent les données dans le même niveau d'abstraction comme l'illustre la Figure 2.3.

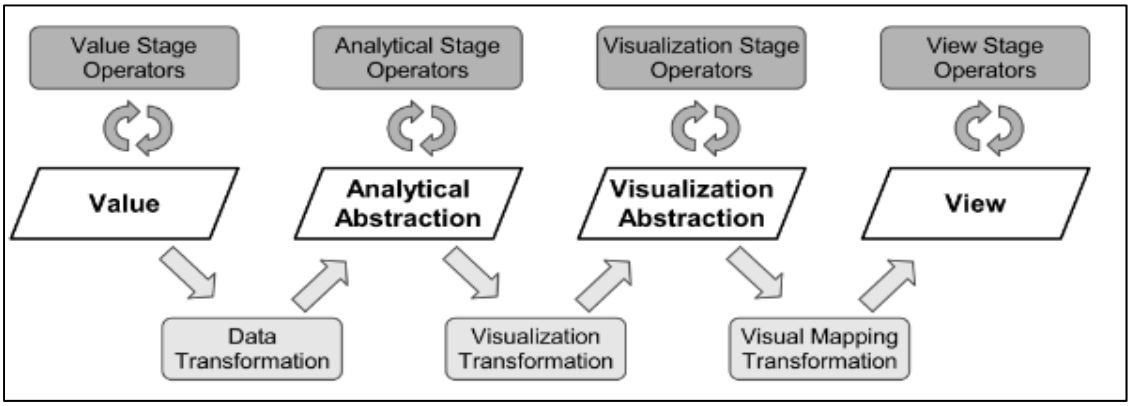

Figure 2.3 Modèle de référence de l'état des données
Tirée de Aigner *et al.* (2011)

Pratiquement, l'application de ces modèles et le choix des paramètres et des opérateurs dépendent principalement de l'objectif de la visualisation requise ainsi que des données d'entrées. La mise en œuvre de ce processus nécessite le contrôle et l'interaction de l'utilisateur. Ces derniers peuvent s'effectuer de différentes manières. En effet, l'interaction peut être simple et se limite à une visualisation dynamique où l'utilisateur est capable de zoomer, sélectionner ou encore demander plus d'information sur une partie d'un graphe. Elle peut aussi être générale; l'utilisateur peut alors contrôler l'ensemble du processus de visualisation et il peut agir à tous les niveaux ou blocs. Deux principaux modèles d'inteaction ont été proposés ; le premier en 1999 par Card et ses coauteurs (Card, Mackinlay, et Shneiderman, 1999) (Figure 2.4) et le deuxième en 2008 par Keim (Keim, Mansmann, Schneidewind, Thomas, et Ziegler, 2008) (D. Keim *et al.*, 2008) (voir Figure 2.5).

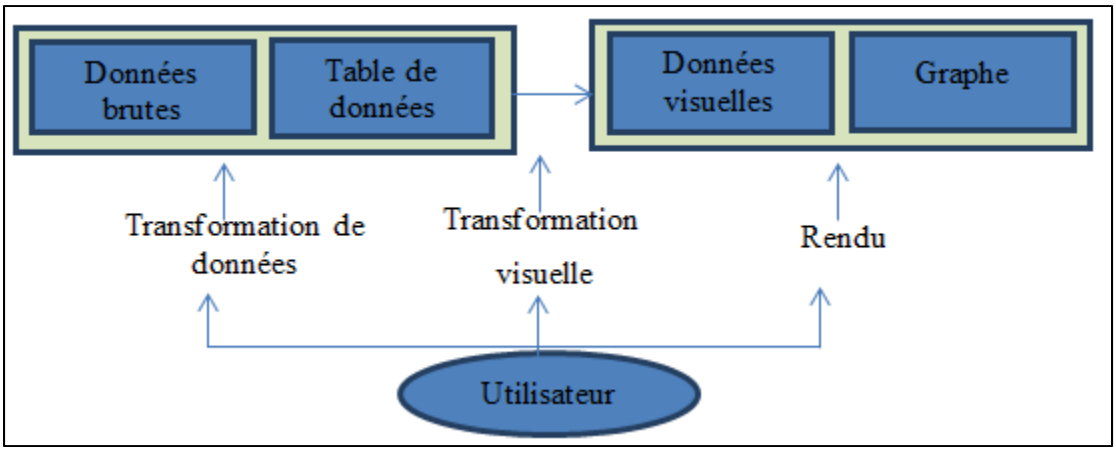

Figure 2.4 Modèle de référence de la visualisation d'information
Adaptée de Card *et al.* (1999)

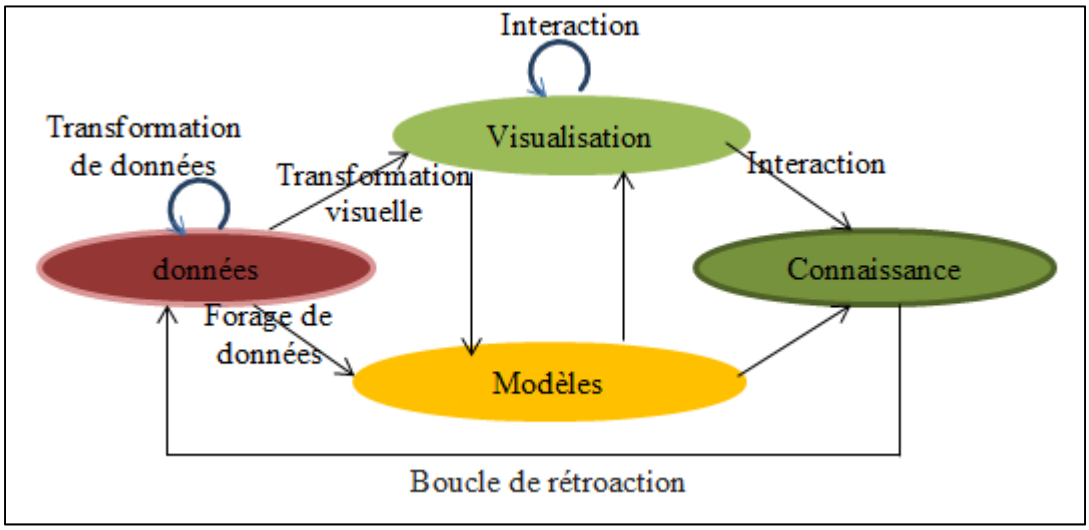

Figure 2.5 Processus analytique visuel
Adaptée de D. Keim *et al.* (2008)

2.2.3 Techniques de visualisation

Les progrès effectués par la théorie de la visualisation, en informatique graphique et ses algorithmes ont donné naissance à de nouvelles techniques de visualisation plus performantes capables de faire ressortir les tendances dans des données multivariées et les représenter corrélations entre les variables. Malgré ces avancées, la visualisation des données multidirectionnelles se heurte à plusieurs contraintes dont les plus importantes sont la

représentation tridimensionnelle de l'espace de représentation ainsi que l'utilisation efficace du système de perception visuel humain, qui ne peut concevoir facilement un espace de plus de trois dimensions. Pour ces raisons, la représentation graphique des données de quatre dimensions et plus nécessite l'introduction de métaphores, à savoir des couleurs, des formes, et bien d'autres.

Beaucoup de travaux ont mis l'accent sur l'intérêt et l'utilité des techniques de visualisation dans l'exploration des données, mais rares sont les auteurs qui se sont intéressés à les classer et en dresser un bilan. Néanmoins, une taxonomie de ces techniques représente une bonne méthode pour d'identifier ce qui se fait couramment et par conséquent un point de départ intéressant pour la mise au point d'un nouveau système de visualisation.

Dans la littérature, il existe deux travaux principaux de taxonomie concernant les techniques de visualisation : le travaux de (Chi, 2000) et de Keim (Keim, 1997). Dans ce qui suit, l'accent est mis sur la classification de Keim.

La première classification des techniques de visualisation des données multidimensionnelles a été proposée par Keim (Keim, 1997, 2002). Elle distingue six catégories de techniques de visualisation:

Techniques géométriques

Les techniques géométriques visent à trouver des transformations intéressantes des données multidimensionnelles (Keim, 2002). En effet, elles permettent de projeter les données multidimensionnelles dans un nouvel espace de représentation, généralement de deux dimensions. Elles sont utilisées pour traiter des jeux de données de grande taille, principalement pour détecter les données aberrantes et les corrélations entre les attributs, notamment avec l'introduction des techniques d'interactions. Une multitude de possibilités de projection dans les espaces de deux dimensions peuvent être imaginées, mais il est important que les nouvelles représentations doivent restituer fidèlement l'information

pertinente contenue dans les données explorées. En plus, des techniques issues du champ des statistiques exploratoires, typiquement la matrice de dispersion (*scatter matrix*), l'analyse en composantes principales et l'analyse factorielle, cette catégorie comprend d'autres techniques permettant de représenter des données multidimensionnelles entre autres les coordonnées parallèles (Inselberg, 2009).

Techniques iconiques

Les techniques iconiques se basent sur les formes géométriques et les icônes pour représenter les données multidimensionnelles dans un espace de deux dimensions. Elles mappent chaque observation à une forme géométrique (glyphe) dont les caractéristiques visuelles (les arrêts, les angles, etc.) varient en fonction des valeurs des attributs des données (Keim, 2002). Cette approche rend possible la représentation des données multidimensionnelles dans l'espace traditionnel. Bien que le nombre de dimensions qui peut être visualisé reste limité, ces techniques sont très utiles dans ce contexte. Quand les attributs des données sont relativement nombreux, par rapport aux dimensions de la représentation (deux dimensions de l'espace de représentation plus le nombre de caractéristiques visuelles du glyphe), la visualisation résultante présente des motifs visuels qui varient en fonction des caractéristiques des données et qui peuvent être détectés par la perception préattentive (Keim, 2002). Cette catégorie inclut plusieurs techniques, entre autres, *Chernof* (Glazar, Marunic, Percic, et Butkovic, 2016), *stick figure* (Peter J. Sackett, M. F. Al-Gaylani, Ashutosh Tiwari, et Williams, 2016), et bien d'autres.

Techniques orientées pixel

Les techniques orientées pixel ne permettent pas de visualiser seulement les données multidimensionnelles, mais aussi celles qui sont en grande en quantité. Elles consistent à représenter chaque valeur de données par un pixel coloré. Pour un jeu de données de dimension n*n, les pixels sont utilisés pour représenter une seule observation où les valeurs de chaque attribut sont arrangées dans une fenêtre séparée. Cette classe de technique se

décline en deux approches principales ; «*querry-dependant»* et «*querry-independant techniques»* (Keim, 1996).

Techniques hiérarchiques

Les techniques de visualisation hiérarchiques subdivisent l'espace de données en sous-espaces organisés d'une manière hiérarchique. Ces techniques sont utilisées pour représenter principalement des données qui renferment une structure hiérarchique. Cette catégorie compte plusieurs techniques entre autres *TreeMap* (Dundas, 2017), *Dimensional Stacking* (Aigner *et al.*, 2011), et bien d'autres.

Techniques basées sur graphes de réseaux (Network Graph based techniques)

Cette technique de visualisation est inspirée de la structure des réseaux au sens où un graphe consiste en un ensemble d'objets appelés nœuds et qui sont interconnectés par des liens appelés «*edges*» (McGraw_Hill, 2002). Ce type de visualisation permet de faire ressortir les groupes (cluster) et permet de découvrir les tendances des relations entre les différentes entités. Par exemple, elle est utilisée pour représenter le trafic Internet, typiquement celui des réseaux sociaux.

Techniques hybrides

Les techniques hybrides intègre plusieurs techniques dans une ou plusieurs fenêtres pour en produire une représentation graphique expressive.

2.2.4 Systèmes de visualisation

La littérature relative à ce sujet liste un nombre important de travaux proposant des systèmes de visualisation pour la gestion des réseaux. En effet, la visualisation de l'information a été utilisée dans ce contexte depuis les années 80 et 90, mais elle n'a pas évolué au même rythme

que celle des réseaux de communication (Gilbert et Kleinöoder, 1985). Cette évolution des technologies de réseaux a rendu nécessaire la surveillance et la gestion de l'infrastructure. Depuis, le nombre de publications portant sur la visualisation du trafic pour la gestion des réseaux a augmenté considérablement. Toutefois, une étude *(Guimarães et al., 2015)* démontre que 78,28 % des travaux effectués entre 2004 et 2013 traitent de problèmes de sécurité voir Figure 2.6.

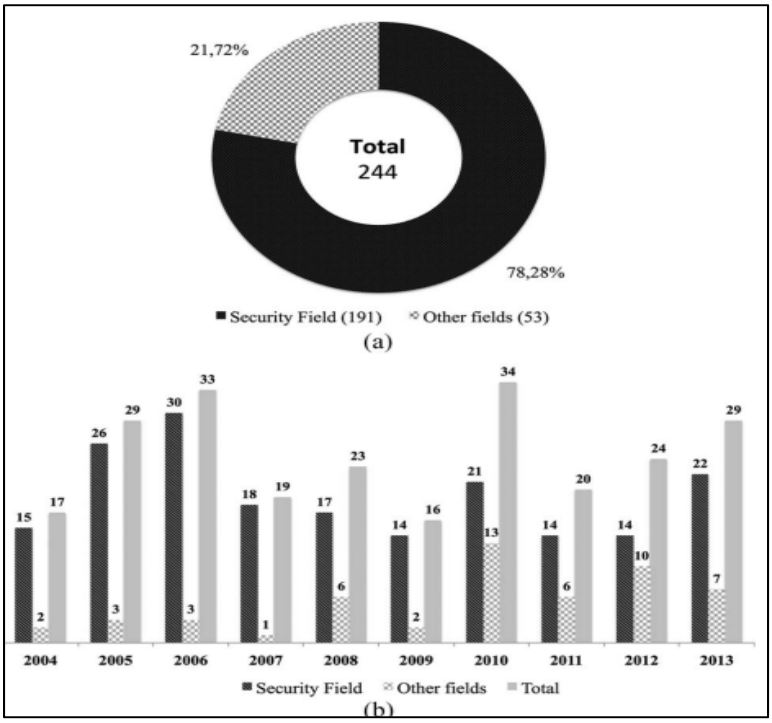

Figure 2.6 La portion d'articles publiés entre 2004 et 2013 en visualisation du trafic
Tirée de Guimarães *et al.* (2015)

Dans ce qui suit l'accent est mis sur un ensemble de travaux de recherche effectués dans le contexte de la visualisation du trafic pour la surveillance et le contrôle des réseaux. Les sections qui suivent présentent plusieurs exemples d'outils existants.

L'outil Visual (Ball, Fink, et North, 2004) est un système permettant de visualiser les communications et les flux entre un réseau local et un réseau externe afin de détecter rapidement le trafic malicieux, par l'analyse de l'intensité des activités des hôtes. Dans ce système, le réseau local est représenté par une matrice où chaque cellule représente un hôte

interne et les hôtes distants sont représentés par des carrés dont les tailles reflètent le niveau d'activités. La connexion entre un hôte interne et un autre distant est représentée par une simple ligne. Ce système offre des fonctionnalités de filtrage pour afficher les activités d'une machine particulière et ainsi éviter de surcharger les graphes, ce qui les rend difficile à analyser.

L'outil TNV (Goodall, Lutters, Rheingans, et Komlodi, 2005) a été conçu pour éviter la perte de la vue d'ensemble du réseau lorsque l'utilisateur analyse en détail le trafic malicieux au niveau des paquets. La composante principale de cet outil est une matrice visualisant les communications de réseau en fonction du temps. Le temps *(timestamp*) est représenté par l'axe des abscisses et les adresses IP sont listées tout au long de l'axe des ordonnées. Chaque colonne de la matrice représente un intervalle de temps et chaque ligne désigne un hôte. Les paquets visualisés dans une cellule (i, j) correspondent à ceux émis/reçus dans l'intervalle j de l'hôte i. Les flux entre deux hôtes, dans chaque période, sont représentés par des lignes (voir Figure 2.7). Une analyse de ports est développée également afin de détecter plus facilement une activité de balayage de ports. D'autres métaphores sont aussi utilisées, notamment la couleur pour mettre en évidence des informations supplémentaires telles que la densité des paquets et le type de protocole. Cet outil permet aussi de supporter la gestion de la sécurité du réseau. Néanmoins il n'est pas conçu pour une visualisation en temps réel car l'analyse des données ne s'effectue pas au cours de la capture du trafic, mais, en lot, à partir des traces préalablement collectées.

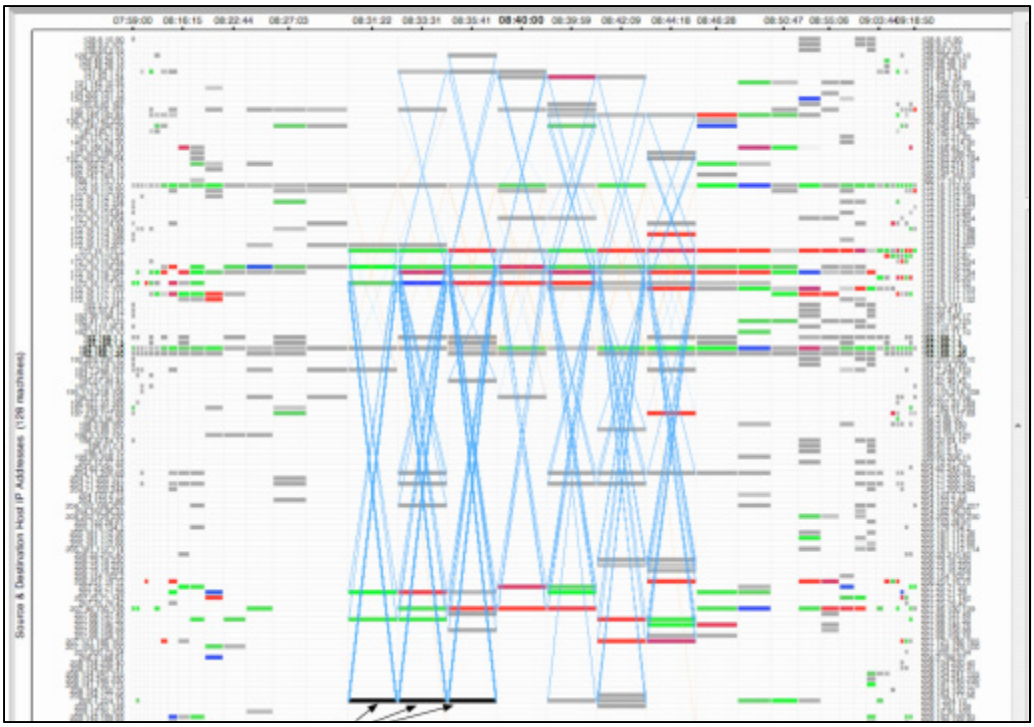

Figure 2.7 La vue principale de TNV
Tirée Goodall et *al*. (2005)

L'outil NVisionIP (Lakkaraju, Yurcik, et Lee, 2004) permet de visualiser le trafic dans un réseau de classe B (ref) pour des finalités de sécurité. Il offre la possibilité d'analyser l'état du réseau de trois manières différentes à travers son cadre principal de visualisation nommé vue *Galaxy* ou *Galaxy View*. Dans la première configuration de visualisation, les hôtes de tous les sous-réseaux sont représentés dans une grille ou une matrice. Les sous-réseaux sont listés au long de l'axe des abscisses tandis que les hôtes sont représentés sur l'axe des ordonnées. Chaque hôte est coloré en fonction de caractéristiques telles que le volume de trafic. La deuxième possibilité de visualisation consiste à regrouper les machines ayant les mêmes services dans des regroupements (Web, DNS, etc.). La troisième configuration visuelle permet de représenter les machines par des rectangles dont les tailles indiquent l'importance des caractéristiques d'intérêt. Ce cadre de visualisation est supporté par deux fenêtres de visualisation ; 1) SMV ; pour (*Small Machine View)* qui visualise les caractéristiques de plusieurs machines d'une région particulière de (*Galaxy view)*, 2) MV (pour *machine View*) qui permet de représenter les détails d'une machine spécifique.

L'outil VizFlowConnect (Yin, Yurcik, Treaster, Li, et Lakkaraju, 2004) et VizFlowConnect_IP (Yurcik, 2006) utilisent la technique de coordonnées parallèles afin de visualiser les communications entre les hôtes internes et externes. Il consiste en trois axes parallèles dont l'axe central correspond aux adresses IP des hôtes internes, le premier axe représente les hôtes externes source du trafic envoyé aux hôtes internes, le troisième axe correspond aux hôtes externes destination du trafic provenant du réseau interne. Bien que cette solution soit simple et permette d'obtenir une vue d'ensemble sur l'état des activités du réseau, en particulier la détection du trafic malicieux, elle ignore l'analyse du trafic interne qui peut être aussi malicieux.

L'outil RTA (Keim, Mansmann, Schneidewind, et Schreck, 2006) est un système de visualisation orientée hôte. Il vise à visualiser la distribution des paquets au niveau d'une machine particulière. Pour parvenir à cet objectif, RTA utilise une approche de visualisation radiale. Ainsi, dans la configuration par défaut, il utilise quatre cercles concentriques pour représenter les attributs d'un paquet (IP_src, IP_dst, Port_src, Port_dst). Le cercle interne représente les adresses IP sources, le deuxième correspond aux adresses IP destination, les deux derniers cercles correspondent respectivement aux ports sources et ports destination. Dépendamment du but de l'analyse, le nombre de cercles composant ce cadre peut être réduit à trois ou deux. Bien que; RTA constitue un système efficace qui permet de surveiller les activités des hôtes et de détecter le trafic malicieux, il se base sur le nombre de ports pour identifier le type d'application (HTTP – >80) ce qui peut entraîner de mauvaises conclusions, notamment avec l'émergence des applications non standards.

Abdullah et ses coll (Abdullah, Lee, Conti, et Copeland, 2005) ont proposé une solution de visualisation permettant d'analyser le réseau en termes d'activités des ports représentés dans un «*stacked graph*». Ils stipulent que cette analyse permet de détecter des problèmes de sécurité qui ne sont pas habituellement détectés par des outils de détection d'intrusion conventionnels tels que les attaques «*zero-day*».

D'autres travaux de visualisation ont mis l'accent sur des sujets particuliers tels que le routage (Au, Leckie, Parhar, et Wong, 2004; Ramachandran et Street, 2012), le flux large (Afaq, Rehman, et Song, 2015), l'étude de certain type de trafic tel que SNMP (Salvador et Granville, 2008; Schonwalder, Pras, Harvan, Schippers, et van de Meent, 2007) et la sécurité de réseau (les travaux décrits plus hauts). En plus de ces propositions, d'autres publications ont eu pour objectif de proposer des solutions plus génériques pour surveiller les réseaux tels que (Lee *et al.*, 2011; Zhang, Chen, et Hu, 2013).

2.3 Classification du trafic

Le but des fournisseurs de services d'Internet est d'optimiser l'utilisation de leurs ressources réseau et d'améliorer la qualité des services offerts aux clients, voire satisfaire voir dépasser leurs exigences. Ceci ne peut être réalisé qu'avec une meilleure connaissance du trafic et des activités sur les réseaux, notamment avec l'émergence de nouvelles applications et SDN (Sofware Difined Network). Pour ces raisons, la classification du trafic d'Internet suscite un intérêt particulier ces dernières années, de la part des chercheurs et des opérateurs de télécommunications. En effet, la classification du trafic constitue une activité cruciale dans toutes les activités d'ingénierie de trafic et de gestion de réseaux. Typiquement, les mécanismes de la gestion de la qualité de service sont une application directe de la classification du trafic. Pour répondre, adéquatement, aux exigences des différentes applications circulant sur le réseau, ces mécanismes doivent se référer à la classification du trafic afin de pouvoir assigner chaque flot à la classe de service (CoS) appropriée et lui attribuer ainsi une priorité adéquate. Les flots appartenant à une classe sont traités différemment par rapport aux autres selon les niveaux de qualité de services prédéfinis. De la même façon, les informations issues de la classification et de l'identification des tendances de trafic sont nécessaires pour mieux concevoir et dimensionner les réseaux. Avec le même niveau d'importance, la classification est une composante essentielle dans le système de sécurité, en particulier dans les mécanismes de détection d'intrusion, d'utilisation injustifiée des ressources réseau ou de trafic malicieux ainsi que les fonctions de sécurité conventionnelles telles que les pare-feu. Avec les avancées technologiques, l'émergence de

nouvelles applications et la popularisation des architectures telle que SDN (*Sofware Difined Network*) et NFV (*Netwok Function Virtuazliation*), font ressortir de nouveaux enjeux de performance des réseaux et de la qualité de service. Ils motivent ainsi la conception et le développement des solutions axées sur la classification et la connaissance du trafic pour y parvenir.

L'identification et la classification du trafic ont connu plusieurs stades d'évolution où différentes approches ont été envisagées voir Figure 2.8. Initialement, ces activités étaient relativement faciles et précises du fait que la majorité des applications utilisaient des numéros de port bien connus et enregistrés auprès de l'IANA (IANA). Toutefois, cette situation n'a pas duré longtemps avec l'apparition et la popularisation rapide des applications utilisant des ports dynamiques ou non standards. C'est ainsi que la classification est devenue un sujet de recherche de grande importance. Dès lors, des solutions basées respectivement sur l'inspection des paquets, l'approche statistique et l'approche comportementale ont fait les sujets de plusieurs travaux de recherche.

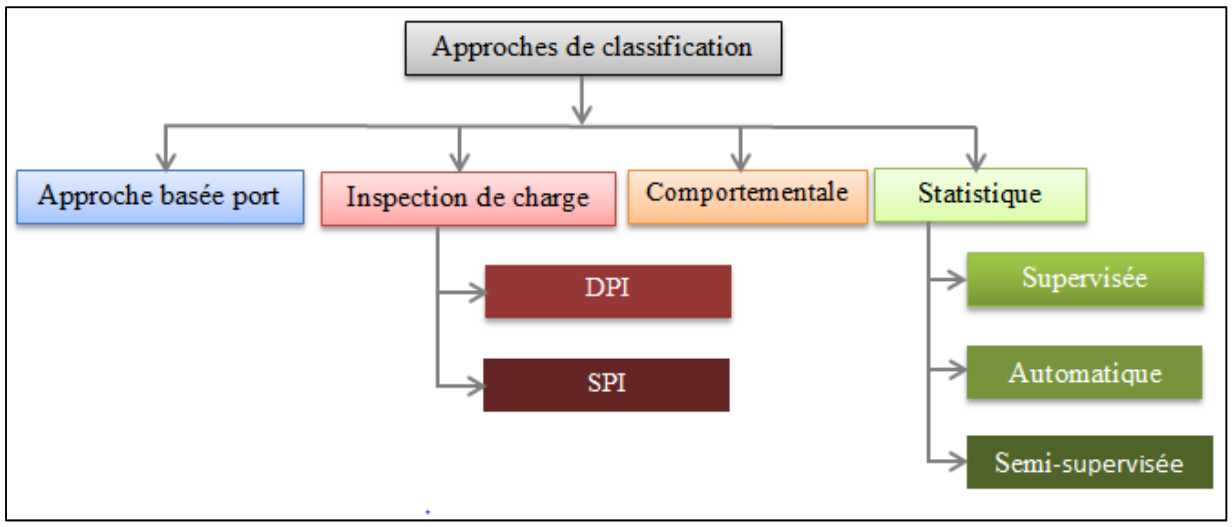

Figure 2.8 Approches de classification du trafic

2.3.1 Taxonomie des méthodes de classification de trafic

2.3.1.1 Classification basée sur les ports

Les premières solutions de classification de trafic se basent principalement sur les numéros de ports pour classifier les applications (Schneider, 1996). Celles-ci sont connues dans la littérature sous le nom de la classification basée sur les ports, en anglais (*port-based classification)*. Cette approche est particulièrement simple et rapide en comparaison avec d'autres approches ; elle classifie efficacement les applications standards vu qu'elles utilisent des ports assignés par l'IANA qui sont bien connues. Bien qu'elle présente des avantages, la classification basée sur les ports est devenue inefficace avec la présence des applications non standards telle que le trafic P2P (Peer to Peer). En fait, ces dernières peuvent contourner les systèmes de contrôle d'accès de plusieurs façons, par exemple en utilisant des ports non enregistrés ou via une allocation dynamique ou encore en utilisant les ports standards et enregistrés des autres applications.

2.3.1.2 Classification par l'inspection de charge

Pour pallier à la limite de la classification précédente, une approche alternative, dite la classification par l'inspection de charge des paquets, ou (*Payload based Classification*) a été envisagée. Cette approche consiste à examiner la charge utile de chaque paquet dans sa recherche d'un indice ou la signature de l'application. Elle se décline en deux branches ; i) Inspection profonde de paquets (DPI pour *Deep Packet Inspection*) et ii) inspection stochastique de paquets (SPI pour *Stocastic Packet Inspection*).

La première branche repose sur une inspection mécanique de la charge utile de chaque paquet lors de la recherche d'une expression ou d'un mot clé caractérisant une application donnée. Moore et Papagiannaki ont proposé un classifieur comportant deux étages combinant l'approche basée sur les ports et la classification par l'inspection de charge (Moore et Papagiannaki, 2005). Le premier étage examine le numéro de port, et lorsque le flux utilise un port non standard, le second étage se charge de chercher dans les premiers octets de la

charge utile du premier paquet une signature ou un protocole connu. Une inspection détaillée de toute la charge utile est nécessaire pour les flux qui restent non classifiés. Les résultats décrits dans cette publication démontrèrent que le premier étage permet de classifier correctement 69 % des octets en utilisant uniquement le numéro de port. Ce taux augmente pour atteindre 79 % en incluant les informations issues des premiers octets du premier paquet. La dernière étape, qui examine les flux non classifiés, permet d'obtenir un resultat élevé, à savoir de presque 100 %. L'Approche discutée dans (Sen, Spatscheck, et Wang, 2004) démontre que la classification du trafic P2P, par l'inspection de la charge, permet de réduire le taux des faux positifs et faux négatif jusqu'à 5 % du total des octets. Cette approche présente des avantages, notamment son taux de classification élevé. De plus, elle est susceptible d'être utilisée dans un processus de classification de trafic en ligne car la signature peut être déduite à partir des premiers paquets des flux. Pour ces raisons, cette approche est implémentée dans plusieurs solutions, telles que la détection d'intrusion (Paxson, 1999) et le pare-feu de Linux (L7-filter). Néanmoins, elle souffre de plusieurs problèmes, en particulier celui du trafic encrypté.

La deuxième famille de techniques reprend le même principe de base qui est l'inspection de la charge, mais d'une manière statistique de façon à chercher les propriétés distinctives de chaque application. Elle vise à combler certaines lacunes de la première famille de techniques. Ainsi, elle utilise des méthodes automatiques pour former des modèles distinctifs. Plusieurs techniques de reconnaissance de forme sont proposées. Par exemple, la solution décrite à l'article de (Ma, Levchenko, Kreibich, Savage, et Voelker, 2006) se base sur les méthode structurelles pour identifier les sous-chaînes de caractères communes entre les flux via le modèle CSG (*Commun Substring Graphs*). D'autres travaux ont appliqué des méthodes statistiques afin d'extraire la signature de la charge utile du trafic IP. Avec ce même objectif, les auteurs de (Finamore, Mellia, Meo, et Rossi, 2010) ont utilisé la valeur des premiers octets de la charge utile comme entrée pour leurs algorithmes d'apprentissage machine (c.-à-d. Naive Bayes, Adboost et Maximum Entropy) pour classifier les applications. Les auteurs de (Khakpour et Liu, 2009) ont développé un classifieur pour distinguer les types des contenus des paquets en utilisant l'entropie des premiers octets de la

charge et les techniques d'apprentissage machine. En effet, l'entropie constitue une caractéristique distinctive du fait qu'elle dépend de la nature du contenu du paquet ; les valeurs d'entropie les plus petites correspondent à un texte simple, les moyennes distinguent un contenu binaire et les plus grandes correspondent à un contenu chiffré.

Bien que les méthodes SPI (Stochastic Packet Inspection) permettent de distinguer la nature de plusieurs types de contenu de trafic, y compris le contenu chiffré, ce qui peut être utile pour prioriser un trafic par rapport aux autres (Khakpour et Liu, 2009). Ces méthodes héritent de plusieurs problèmes des méthodes DPI (Deep Packet Inspection) et elles sont incapables d'identifier le type du trafic du fait que certaines applications peuvent utiliser plusieurs types de contenu en même temps.

2.3.1.3 Approche comportementale

L'approche comportementale se focalise sur l'analyse du comportement des hôtes et de la distribution des connexions pour déduire le type de trafic avec le but d'identifier les applications actives sur un hôte donné. Les classifieurs de cette catégorie examinent les patrons générés par le trafic en observant un certain nombre de paramètres, tel que le nombre d'hôtes qui y sont connectés, le nombre de ports et les protocoles utilisés. L'idée derrière une telle approche est que différentes applications génèrent des patrons différents. Par exemple un serveur Web est interrogé par plusieurs clients par des sockets parallèles tandis que dans un réseau P2P (Peer to Peer) les hôtes sont interconnectées avec le même dégrée de popularité. L'outil BLINC (Karagiannis, Papagiannaki, et Faloutsos, 2005) est une solution qui permet de classifier le trafic en se référant à l'allure des communications au niveau d'un hôte. Les concepteurs de TDG (Iliofotou *et al.*, 2007) et (Iliofotou *et al.*, 2009) ont utilisé les répartitions graphiques des flux ainsi que des métriques de classification (dégrée de connexion, I/O, etc.) pour classifier le trafic.

2.3.1.4 Approche statistique

La classification du trafic par l'approche statistique s'appuie d'un côté sur les techniques d'apprentissage machine et d'un autre côté sur le fait que les différents types de trafic possèdent différentes caractéristiques ou métadonnées telles que la taille des paquets (petite ou grande), la taille des flux, le temps interarrivé des paquets, la durée des flux, etc. En effet, chaque flux est décrit par un vecteur de caractéristiques qui réfère à une observation ou une instance dans le jeu de données (Thuy TT Nguyen et Armitage, 2008).

2.3.2 Classification du trafic et méthodes d'apprentissage machine

Witten et Frank ont établi dans (Witten et Frank, 2005) une taxonomie des techniques de classification où ils ont défini quatre classes en fonction de la nature d'apprentissage :

- la classification supervisée,
- la classification automatique,
- les techniques d'association,
- et les techniques de prédiction.

Ces deux dernières techniques n'ont pas fait l'objet de discussions dans ce document.

L'apprentissage automatique est connu aussi dans la littérature sous le nom de la classification non supervisée, regroupement et *clustring* en anglais. Ce type de classification consiste à découvrir et organiser les données dans des groupes naturels, compacts et bien séparés et d'attribuer chaque nouvelle observation à une classe parmi les classes identifiées. Dans ce contexte, les exemples ou les données d'apprentissage ne sont pas étiquetés. C'est ainsi que la classification automatique s'appuie sur les propriétés de similarité/dissimilairité pour établir les groupes et de la même façon affecter une nouvelle observation à un groupe. Ceci revient à calculer les distances entre les exemples via des fonctions de distance dont la plus simple est la distance euclidienne. Pratiquement, les méthodes de classification automatique peuvent vérifier cette contrainte à l'aide du calcul d'inertie. En effet, l'inertie

d'un cluster mesure la concentration des points autour du centre du groupe et plus elle est faible, plus la dispersion des points autour du centre est petite. De la même manière, l'inertie inter-cluster mesure « l'éloignement » des centres des clusters entre eux et plus elle est grande, plus les clusters sont bien séparés. Il existe plusieurs méthodes de *clustring* entre autres ; les méthodes dynamiques en encore centre mobile (K-mans), les méthodes hiérarchiques et les méthodes probabilistes (Berkhin, 2006).

Alternativement, l'objectif de la classification supervisée est de déterminer, à partir d'un algorithme d'apprentissage, une fonction permettant d'associer chaque nouvelle observation ou instance à une classe parmi celles préalablement définies. Ainsi, et pour construire une telle fonction, l'algorithme d'apprentissage machine déployé nécessite la présence de données étiquetées. Parmi les algorithmes d'apprentissage supervisé on cite les arbres de décision, les machines à vecteurs support (SVM), et les classifieurs de bayes, etc. Dans cette recherche on s'intéresse aux classifieurs SVM, et aux algorithmes d'arbres de décision, en particulier le C4.5.

Le point commun de ces techniques réside dans le principe du schéma de classification. En effet, les méthodes d'entrainement d'un algorithme d'apprentissage machine consistent principalement en deux étapes : i) la phase d'apprentissage qui permet d'apprendre le modèle l'image (Figure 2.9-a) et ii) la phase de classification qui consiste à tester le modèle généré durant la première phase (Figure 2.9-b).

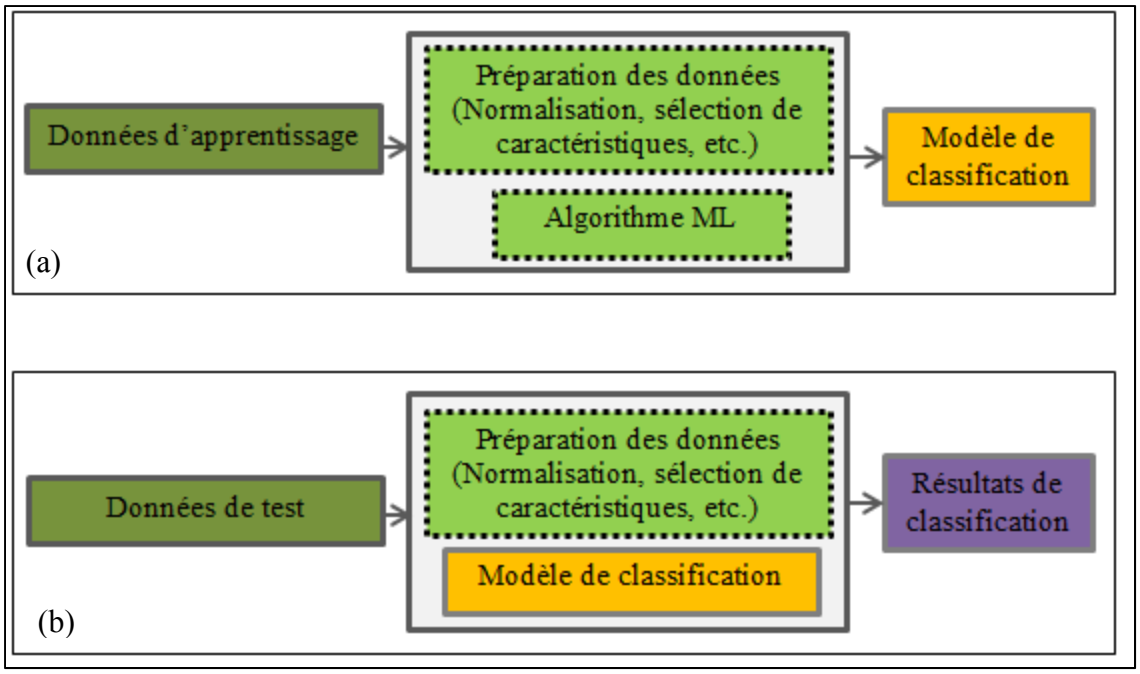

Figure 2.9 Processus de classification, phase d'apprentissage (a), phase de classification (b)

2.3.2.1 Arbre de décision

L'arbre de décision est l'une des méthodes les plus populaires dans l'apprentissage supervisé et dans la régression. Selon un sondage redû public sur site de kdnuggets (kdnuggets, 2017), les arbres de décision sont les plus utilisés dans l'exploration et l'analyse des données en 2015. En plus de leur nature compréhensible par l'être humain, ces méthodes non paramétriques ont plusieurs avantages tels que la robustesse au bruit, le coût d'exécution faible et son habilité à traiter des attributs redondants (Barros, de Carvalho, et Freitas, 2015).

D'après (Barros *et al.*, 2015) un arbre de décision est une structure hiérarchique qui permet de partitionner un ensemble d'individus en groupes homogènes et disjoints selon un ensemble d'attributs discriminants et de variables d'intérêt ou variables de sorties. Cette structure arborescente peut être vue comme un graph G (n, a) qui consiste en un ensemble fini non vide de nœuds et un ensemble d'arcs ou d'arrêt. Ce graphe doit satisfaire un certain nombre de propriétés, qui sont :

- le graphe doit être dirigé, c.-à-d les arrêts sont orientés par un pair de nœuds (v, w) qui indiquent respectivement les points d'origine et de fin;
- le graphe doit être acyclique;
- le graphe doit avoir un seul nœud racine sans arrêt entrant ou en d'autres termes qui n'a pas de parent;
- chaque nœud, à l'exception de la racine, doit avoir un seul arrêt entrant;
- le graphe doit avoir un chemin unique (suite de pair de nœuds) de la racine vers n'importe quel point dans l'arbre;
- dans un chemin du nœud v au nœud w, v = w, v est un ancêtre de w et w est un descendant de v. un nœud sans descendant propre (enfant) est appelé une feuille (ou un terminal). Tous les autres à l'exception de la racine sont appelés nœuds internes.

Dans un arbre, la racine et les nœuds internes correspondent aux tests dont les résultats sont représentés par les arcs. Chaque feuille, ou nœud terminal, représente une décision (une classe). Un nœud est dit terminal lorsque tous les éléments du sous-ensemble associé à ce nœud ou la majorité d'entre eux appartiennent à la même classe.

Un arbre de décision est caractérisé aussi par sa profondeur et sa largeur. Le premier indicateur correspond au nombre moyen des niveaux de l'arbre tandis que la largeur ou le degré désigne le nombre moyen de nœuds interne à chaque niveau. Ces deux indicateurs reflètent la complexité de l'arbre ; plus les valeurs sont élevées plus l'arbre est complexe. Ainsi, le but est de générer un arbre de décision aussi petit que possible. Pour y parvenir, l'idée intuitive est de chercher les attributs (tests) qui font progresser rapidement la classification des exemples d'entraînement. Ceci revient à mesurer la pertinence des variables et de choisir celle qui permet de mieux partitionner une proportion de données associée à une position déterminée de l'arbre. Les critères les plus populaires dans ce contexte sont le taux d'erreurs, le critère de Gini utilisé par la méthode CLART (Olshen et Stone, 1984; Timofeev, 2004) et l'entropie qui est l'élément clés C4.5 (Ruggieri, 2002). Dans ce qui suit, nous nous limitons à ce dernier algorithme. Ces deux principaux algorithmes reposent sur le protocole de construction générique Algorithme 2.1:

Algorithme 2.1 Génération d'un arbre de décision
Adapté de François Denis and Gilleron

Input *D* training dataset, *A* list of non-target attributes, *C* Class or target variable

Output T (tree)

Start

1 Initialize to empty tree;
2 Create a node *N*
3 ***If*** all samples of *D* belong to the same class *C* or a stopping criteria ***then***
4 Terminate (return *N* as leaf labeled as *C*)
5 ***endIf***
6 ***For*** all $a \epsilon A$ ***do***
7 Compute information criteria with respect to *a*
8 ***EndFor***
9 Select attribute a with the best information criteria
10 D_{av}=the subsets of *D* on *a* position (for each value *v* of *a*)
11 T= create node N for a in the root
12 ***For*** all value D_{av} ***do***
13 T_v = sub-tree generated by D_{av} and A-a
14 Attach T_v to T in node N with the corresponding branch
15 ***EndFor***
16 ***Return T***

Construction de l'arbre de décision par C4.5

L'algorithme d'apprentissage C4.5 (Ruggieri, 2002) est une amélioration du ID3 permettant de surmonter les limitations de ce dernier. Les deux algorithmes utilisent la même entropie pour partager les données. Mais contrairement à ID3, C4.5 est capable de traiter des valeurs non disponibles ou inconnues et les données continues. Pour construire l'arbre descendant, à chaque étape (nœud), C4.5 choisi la variable maximisant le gain d'information le plus élevé comme étant la variable discriminante. Ceci revient à calculer l'entropie ou la quantité

d'information contenue dans chaque nœud (position de l'arbre). Considérons un échantillon T ayant une partition (C_1, C_2… C_N) selon un ensemble de n classe (c_1, c_2…, c_n) et l'ensemble (T_1, T_2…. T_m) correspondant à la partition de T selon les modalités d'une variable V.

Le gain de l'information d'une variable V est défini par l'équation (2.1)

$$Gain(V,T) = info(T) - \sum_{i=1}^{n}\left((T_i / T) * info(Ti)\right) \tag{2.1}$$

Où

$$info(T) = -\sum((C / T) * log(C / T) \tag{2.2}$$

On note que T est le sous-ensemble de l'échantillon d'entrée (base d'apprentissage) qui correspond à un nœud donné de l'arbre ayant la position p, et T_i correspond aux éléments de T qui satisfassent le test de la $i^{ème}$ branche du test p.

L'utilisation du gain comme critère pose des problèmes dans certains cas où il favorise les attributs ayant un grand nombre de valeurs différentes même s'ils ne sont pas discriminants. Pour pallier à cet écart, C4.5 utilise un autre critère appelé *GainRatio*. Ce dernier n'est autre que le rapport calculé par la fonction (2.3).

$$GainRatio(V,T) = Gain(V,T) / Split(V,T) \tag{2.3}$$

Où

$$Split(V,T) = \sum_{i=1}^{m}(T_i / T) * log(T_i / T) \tag{2.4}$$

Dans certains cas, les algorithmes d'apprentissage ci-dessus peuvent générer des arbres de décision trop complexe. Dans ce contexte, l'arbre peut avoir des nœuds terminaux pures, mais ayant peu d'exemples pour pouvoir prédire les classes d'une manière correcte. C'est

pour cette raison et pour palier à ce problème de sur-apprentissage que l'élagage a été introduit. L'élagage permet de construire un arbre avec une taille optimale et d'éviter ainsi des feuilles non pertinentes. Il peut être effectué de deux manières, soit durant la construction de l'arbre par le prélavage, soit après l'obtention de l'arbre pur via une autre étape appelée post-élagage.

2.3.2.2 Forêts d'arbres décisionnels (RandomForest Classifier)

Les forêts d'arbres décisionnels font partie des techniques d'apprentissage machine supervisé qui ont été inventé par Breiman en 2001 (Breiman, 2001). Elles consistent à combiner la technique de «Bagging» (Breiman, 1996) et le concept de sous-espace aléatoire (Random subspace) (Ho, 1998). Cet algorithme effectue l'apprentissage des modelés sur de multiples sous-ensembles de données d'apprentissage et génère ainsi plusieurs arbres de décision ou une forêt d'arbre de décision (Biau, 2012).

Considérons un ensemble d'apprentissage $((x_1, y_1) (x_2, y_2)\ldots(x_n, y_n))$ et chaque observation x_i appartient à R^p La construction d'une forêt d'arbres de décision est effectuée en plusieurs étapes :

- La première étape consiste en échantillonnage pour créer les sous-ensembles d'entrainement en utilisant un tirage avec remise d'un échantillon de N observations. Il s'agit du processus «*bootstrap*» utilisé dans le «*Bagging*»;
- La deuxième étape correspond au choix k des variables parmi p pour chaque nœud de chaque arbre. La variable qui maximise le critère de CART est retenue.
- Sur chaque sous-ensemble, un arbre est entrainé à l'aide la procédure de construction d'un arbre de décision (Algorithme 2.1).
- Les N arbres de décision constituant une forêt d'arbres décisionnels sont utilisés afin de prédire la classe d'une nouvelle observation à l'aide d'un vote majoritaire.

2.3.2.3 Machine à vecteurs de support (SVM)

Les machines à vecteurs de support font partie des méthodes les plus rependues en l'apprentissage machine. Cette méthode a été introduite par Vapnik (Vapnik et Vapnik, 1998). Contrairement à d'autres algorithmes de classification qui reposent sur l'estimation de la densité de probabilité des données, les SVM se basent sur l'estimation d'une fonction de classification. Dans le cas de deux classes de données, cette fonction décrit typiquement un hyperplan. L'objectif d'un classifieur SVM est de trouver un hyperplan discriminatoire permettant de maximiser la distance entre les deux classes. Fondamentalement, les SVM sont des classifieurs linéaires binaires du fait qu'ils s'appuient sur l'existence d'une fonction séparatrice linéaire entre deux classes d'exemples dans un espace approprié. Ainsi, on distingue essentiellement deux types de classifieurs SVM binaires ; les classifieurs à marge dure et les classifieurs à marge souple.

A. Classifieur à marge dure

Dans le cas de deux classes de données, le classifieur cherche à déterminer un hyperplan qui va séparer le mieux les deux classes. Ceci revient à trouver une fonction discriminante optimale de forme linéaire. Ainsi, l'hyperplan est décrit par l'équation (2.5) :

$$H(x) = W^t x + b \tag{2.5}$$

Où W est un vecteur de dimension m (mêmes dimensions que x) et représente un terme.

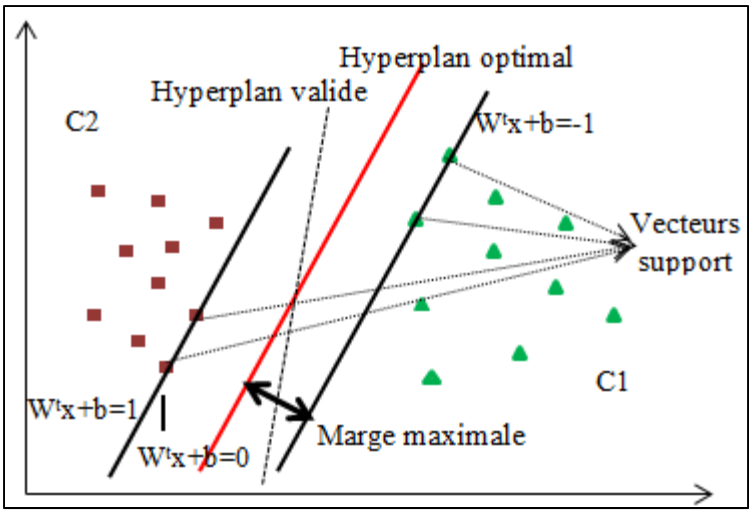

Figure 2.10 SVM binaire
Adaptée de (Abdelhamid, 2012)

La fonction de décision pour un élément x peut être écrite comme suit :

$$\begin{cases} x \in C1 \ \text{ si } H(x) \leq -1 \\ x \in C2 \ \text{ si } H(x) \geq 1 \end{cases} \tag{2.6}$$

Soit x un exemple de la base d'apprentissage X et id est la valeur de l'étiquette ∈ (1, -1) qui lui correspond et définissant ainsi sa classe d'apparence. La fonction de décision décrite dans (2.6) est équivalente à l'inégalité (2.7).

$$y_x(W^t x + b) \geq 1 \quad \forall\, x \in X \tag{2.7}$$

Il est évident qu'il existe une multitude d'hyperplans valides qui peuvent séparer les deux classes, mais uniquement un seul hyperplan qui maximise la marge d. Cette dernière définit la distance entre l'hyperplan et le point le plus proche ou vecteur support (Figure 2.10). La zone de généralisation du classifieur SVM correspond à la distance entre les deux hyperplans de marge (qui passent par les vecteurs supports de chacune des deux classes) définis par $W^t x+b=-1$ et $W^t x+b=1$. Cette distance qui n'est autre que le double de la marge est un indicateur de l'habilité du classifieur d'être généralisé. La recherche de l'hyperplan optimal

se base sur la maximisation de la zone de généralisation, ce qui revient à maximiser la marge d durant la phase d'apprentissage du classifieur. C'est ainsi que les classifieurs SVM sont appelé aussi les classifieurs à vaste marge.

La détermination d'hyperplan optimal consiste à trouver le vecteur W qui maximise la marge d, qui fait référence à la distance euclidienne minimale entre l'hyperplan et le point le plus proche (vecteur support). La distance entre un point et un hyperplan est exprimée par l'équation suivante:

$$d_x = \frac{\left|W^t x + b\right|}{\left\|W\right\|} \tag{2.8}$$

La marge est égale alors à :

$$d = \frac{1}{\left\|W\right\|} \tag{2.9}$$

Tout élément x de X doit remplir alors l'inégalité suivante :

$$\frac{y_x \times H(\mathrm{x})}{\left\|W\right\|} \geq d \tag{2.10}$$

Ce problème d'optimisation admet une infinité de solutions. En effet, si (W, b) est une solution de l'inéquation (2.10), tous les pairs (aW, ab) sont aussi des solutions qui ne diffèrent que par le facteur a. Toutefois, l'objectif est de trouver un seul hyperplan optimal. Pour ces raisons, il est nécessaire de restreindre le nombre de solutions à 1 en imposant une contrainte sur W. Ainsi l'hyperplan peut être calculé en résolvant l'équation quadratique suivante (2.11):

$$\begin{cases} \text{minimiser } \dfrac{\|W\|^2}{2} \\ \text{sous contraintes} \\ y_x(\mathrm{W}^t\mathrm{x}+\mathrm{b}) \geq 1, \forall x \in X \end{cases} \tag{2.11}$$

On note que la maximisation de la marge d ($\frac{1}{\|W\|}$) est équivalente à la minimisation de la norme $\|W\|$. La norme carrée du vecteur W est utilisée pour faciliter la résolution du problème.

B. Classifieur à Marge souple

Les SVM à marge dure sont utilisés idéalement lorsque la base d'apprentissage ne contient pas d'exemples mal-étiquetés, qui peuvent tomber dans la zone de généralisation et augmenter ainsi l'erreur de classification. Cependant, en réalité les données d'apprentissage sont souvent bruitées, ce qui peut rendre difficile la généralisation et dans certains cas la génération d'un classifieur à marge dure. Dans le cas des données non linéairement séparables, la génération d'un modèle qui colle exactement à la base d'apprentissage peut engendrer le phénomène du sur-apprentissage. C'est dans l'objectif de palier à ces limitations que les classifieurs à marge souple ont été introduits. Pour y parvenir et assurer l'existence d'un hyperplan séparateur, ces derniers tolèrent une certaine erreur de classification modélisée par un facteur de relaxation positif ε_x. Si $\varepsilon_x \prec 1$ x ne respecte pas la zone de généralisation, mais il est bien classé sinon l'exemple x est mal classé par l'hyperplan (voir Figure 2.11).

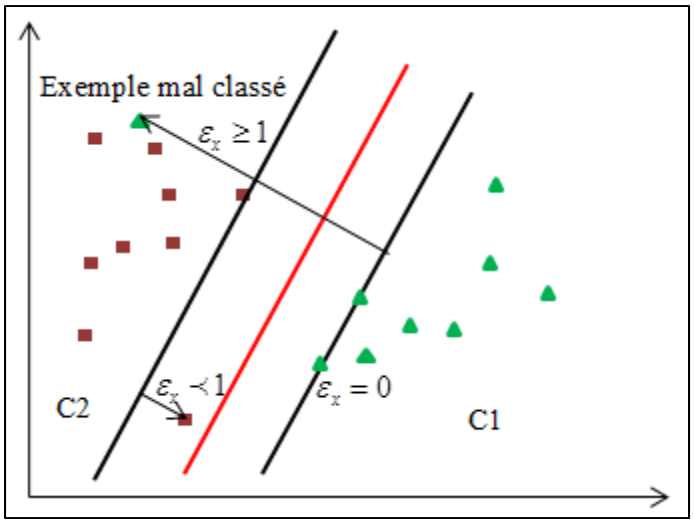

Figure 2.11 SVM à marge souple
Adaptée de (Abdelhamid, 2012)

Dans ce cas, l'hyperplan optimal doit, à la fois, maximiser la marge et minimiser la somme des erreurs de classification permise. Ainsi le problème d'optimisation devient (2.12):

$$\begin{cases} \text{minimiser } \dfrac{\|W\|^2}{2} + C\sum_x \varepsilon_x \\ \text{sous contraintes} \\ y_x(\mathrm{W}^t\mathrm{x}+\mathrm{b}) \geq 1-\varepsilon_x \quad \forall \mathrm{x} \in \mathrm{X} \\ \varepsilon_x \geq 0 \end{cases} \tag{2.12}$$

Les valeurs d'erreur ε_x pour les exemples mal classés peuvent être arbitrairement importantes. Ainsi; il convient de pénaliser les erreurs de classification par un coefficient C positif pour balancer les deux termes de la fonction objective.

Sans entrer dans les détails, en utilisant les multiplicateurs de Lagrange, ce problème primal (2.12) est équivalent au problème dual (2.13) :

$$\begin{cases} \text{maximiser } \varphi(\alpha)=\sum_x \alpha_i - \frac{1}{2}\sum_x \sum_x \alpha_{x_i}\alpha_{x_j} y_{x_i} y_{x_j} x^t{}_i x_j \\ \text{sous contraintes} \\ \sum_x \alpha_x y_x \\ 0 \le \alpha_x \le C \end{cases} \tag{2.13}$$

C. Noyaux

Bien que les classifieurs SVM à marge souple permettent de surmonter la limitation d'une machine à marge dure en admettant une certaine erreur de classification, ils ne peuvent pas être toujours généralisés. En effet, dans ce contexte le taux d'erreur accumulé devient important que la séparation linéaire devienne inefficace. L'utilisation d'une fonction alternative qui sépare le mieux les exemples d'apprentissage est désirable; cependant, la génération d'une telle fonction qui est par intuition non-linaire est extrêmement difficile. Ainsi, il convient de transformer les données non linéairement séparables dans un nouvel espace de dimension plus élevé appelé espace de caractéristiques où le principe de la séparation linéaire peut être appliqué. Cette transformation est réalisée par le biais d'une fonction de projection dont le produit scalaire de deux images désigne une nouvelle fonction appelée noyau K (x, y) (voir Figure 2.12). L'utilisation des noyaux permet d'élargir le champ d'application des SVM en ramenant un problème non linéaire vers un espace où il peut le devenir.

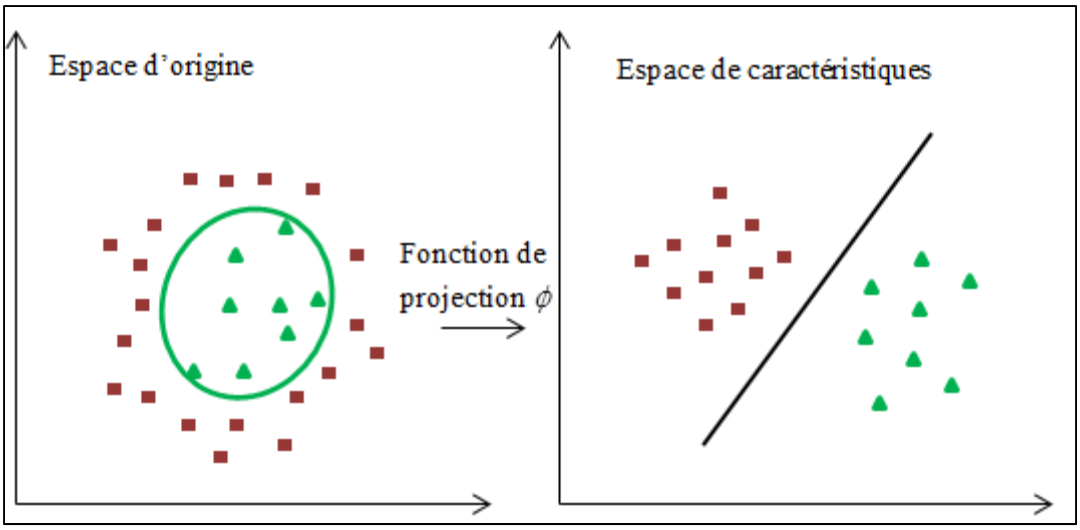

Figure 2.12 SVM avec noyaux
Adaptée de (Abdelhamid, 2012)

Le problème de séparation peut être résolu dans le nouvel espace en se basant sur le principe de base de SVM qui est la séparation linéaire, étant donné que les exemples sont projetés dans l'espace de caractéristiques où ils sont linéairement séparables. Le problème d'optimisation est donné par (2.14) :

$$\begin{cases} \text{maximiser } \varphi(\alpha)=\sum_x \alpha_i - \frac{1}{2}\sum_x \sum_x \alpha_{x_i}\alpha_{x_j} y_{x_i} y_{x_j} \langle \phi(x_i), \phi(x_j) \rangle \\ \text{sous contraintes} \\ \sum_x \alpha_x y_x \\ 0 \le \alpha_x \le C \end{cases} \tag{2.14}$$

Rappelant que la fonction noyau K (x_i, y_i) peut être vue comme le produit scalaire des images des deux exemples x_i et y_i par la fonction de projection ϕ. Il convient alors d'utiliser la fonction noyaux au lieu de la fonction de projection qui peut être inconnue.

D. Multiclasses

Les SVM sont fondamentalement utilisés pour séparer les exemples de deux classes. Cependant, en réalité ces classifieurs à vaste marge traitent des problèmes multiclasses. Dans ce contexte, l'objectif d'un classifieur SVM consiste à chercher plusieurs hyperplans plutôt qu'un seul pour pouvoir affecter à un exemple une classe parmi plusieurs. Pour ce faire, un problème multiclasses est décomposé en ensemble de plusieurs sous problèmes binaires. Ainsi l'hyperplan de chaque sous problème est déterminé par la séparation SVM binaire fondamentale. Il existe différentes méthodes de décomposition :

Une contre reste (*one against all*)

Cette méthode est proposée par Valnik (Vapnik et Vapnik, 1998). Elle consiste à déterminer l'hyperplan qui sépare les exemples d'une classe k du reste (exemples appartenant aux classes). On comprend que chaque classe k possède son propre hyperplan H_k (x) (classifieur k) qui la sépare des autres classes, ce qui consiste en n hyperplans pour un problème de discrimination à n classes. Intuitivement, l'affectation d'un exemple x à une classe k passe par le calcule de signe (H_k (x)) ou l'exemple appartient à une telle classe si signe (H_k (x))=1. Cependant, cette affectation immédiate peut soulever des problèmes en particulier quand plusieurs classes vérifient cette condition à savoir H_k (x)=1, ce qui engendre des zones d'ambiguïté ainsi que des exemples non classés. Pour y remédier, cette méthode se base sur le principe du *gagneur prend tous* ou "winner-takes-all" et la décision est prise en présentant l'exemple considéré aux n classifieurs et la classe retenue correspond à celle maximisant H_k(x) suivant la fonction (2.15) :

$$k = \arg\max_{1 \le k \le n} (\mathrm{H_k}(\mathrm{x})) \tag{2.15}$$

Une contre une (one against one)

Cette méthode développée par Khner (Knerr, Personnaz, et Dreyfus, 1990) consiste à trouver un classifieur pour chacune des combinaisons de deux classes. Ainsi, la méthode 1vs1 apprend n (N-1)/2 classifieurs binaire et la fonction de décision H_{kp} (x) définit l'hyperplan qui sépare la classe k de la classe p. Pour affecter un exemple x à une classe k, la méthode le présente à tous les classifieurs et procède par la suite à un vote pour déterminer la classe majoritaire via l'équation (2.16) :

$$k = \underset{1 \le k \le N}{\operatorname{argmax}} \left(\sum_{p=1}^{m} H_{kp}(\mathrm{x}) \right) \tag{2.16}$$

2.3.3 Travaux connexes

Après avoir présenté la classification du trafic et les notions de base de l'apprentissage machine, et en particulier, l'apprentissage supervisé (l'arbre de décision et SVM), cette section présente une synthèse des travaux antérieurs qui s'articule sur l'application de l'apprentissage machine pour la classification du trafic.

Les travaux s'articulant sur la classification du trafic en utilisant l'approche statistique sont assez nombreux. Parmi les premières solutions potentielles dans ce cadre, on trouve celle proposée par Moore et Zuev (Moore et Zuev, 2005). Ces derniers ont développé un classifieur de trafic Internet basé sur les techniques de Naive Bayes. Dans le même ordre d'idée, la classification du trafic, en temps réel, reçoit beaucoup d'intérêt et les chercheurs prêtent une attention particulière à ce sujet, vu son rôle fondamental dans tout système de contrôle et de performance de réseau. En appliquant les techniques de Bayes, les auteurs de (R. Gu, Wang, et Ji, 2010) proposent une solution de classification de trafic temps réel. D'une manière similaire, le classifieur proposé permet d'identifier les flux en observant les premiers n paquets. Contrairement à (Moore et Zuev, 2005) qui a utilisé 248 caractéristiques, ce classifieur n'en utilise que deux (c.-à-d. la taille des paquets et le temps interarrivé) pour classifier un nouveau flux. Cette approche peut effectivement améliorer le coût de la classification, mais cette solution ne prend en considération que les applications standards.

Les méthodes de machine à vecteurs de support (SVM) ont été largement utilisées dans la classification du trafic réseau. Une étude sur l'impact du choix des caractéristiques de performance d'un classifieur SVM ont été résumées dans (Yuan, Li, Guan, et Xu, 2010). Les auteurs de (Este, Gringoli, et Salgarelli, 2009) ont discuté d'un classifieur SVM en se référant uniquement à la taille et à la direction de chaque paquet. Le vecteur de caractéristique d'un flux est composé de n mesures pour n paquets formant le flux bidirectionnel et chaque mesure renferme les deux informations (taille et direction). Toutefois, les résultats exposés dans ce document démontrent que le classifieur proposé est incapable d'identifier le trafic P2P. Ce qui peut être expliqué par les limitations que présente cette classification dont la plus visible est celle liée à l'ordonnancement des paquets. Yu et ses coauteurs proposent dans (Yu, Lee, Im, Kim, et Park, 2010) un classifieur hiérarchique à trois niveaux, combinant un SVM binaire SVDD (*Support Vector Data Description*) qui est une approche de classification monoclasse et est étendue pour résoudre les problèmes multiclasses (*One Class Based Multiclass SVM*) (Tax et Duin, 1999). Le premier niveau se base sur un classifieur SVM binaire qui permet de séparer le trafic en deux classes (trafic P2P et non-P2P), le deuxième niveau permet de séparer le trafic P2P en plusieurs classes en utilisant SVDD et le troisième niveau est utilisé pour identifier des applications précises. Le déploiement d'un classifieur hiérarchique peut favoriser une identification du trafic P2P en temps réel. Cependant, la solution proposée ne peut pas satisfaire cette dernière propriété (c.-à-d. la classification en temps réel) parce qu'elle utilise la durée de flux qui ne se calcule qu'après que le flux soit terminé. L'article de (C. Gu, Zhang, et Huang, 2011) présente un classifieur du trafic appelé PSVM (*Proximal SVM*). Les auteurs argumentent que l'utilisation de PSVM simplifie le problème de classification par rapport aux méthodes SVM standards et améliore ainsi le coût de classification en termes de temps d'exécution. Par ailleurs, pour réaliser une classification rapide (c.-à-d. temps réel) le modèle n'utilise que les 10 premiers paquets pour retourner sa décision. Cependant, ce classifieur est sensible à la déviation des données à classifier des données d'apprentissage ce qui risque d'augmenter le taux d'erreur de classification. L'article de (D'Alessandro, Park, Romano, et Fetzer, 2015) décrit un classifieur SVM distribué destiné à classifier des quantités de trafic réseau à grande échelle. Cette solution utilise le langage *MapReduce* déployée sur une grappe d'ordinateurs de 20

nœuds. Principalement, le jeu de donnée est subdivisé en n jeu plus petit qui sont associés aux n nœuds de la grappe de calcul afin de former les modèles locaux (w,b) dans l'étape *Map*. Dans cette étape, le modèle global est généré par la combinaison des modèles locaux. Ce dernier modèle est sauvegardé sous forme d'un paramètre global qui est utilisé dans l'étape *Map* pour recommencer une nouvelle boucle dans un processus itératif jusqu'à l'obtention d'un modèle SVM optimal. Les résultats démontrent que cette solution permet de réduire le temps d'entraînement de 30 % par rapport à *CloudSVM*.

Il existe aussi une portion de travaux qui s'appuient sur des approches hybrides, dans le but d'alléger le processus de classification et d'améliorer la performance. Dans cette catégorie, on le trouve, entre autres, des solutions combinant la classification basée sur les ports et l'approche statistique. Avec l'objectif de pallier aux limitations de chacune des approches de classification ; classification basée sur les ports, classification par inspection de charge et l'approche statistique, les auteurs de (Awad *et al.*, 2014) ont proposé un système de classification hybride appelée SSPC (*Signature Statistical and Port Classifier*). Ce dernier utilise ces approches sous forme de trois classifieurs indépendants et chacun possède sa propre décision. Pour attribuer un flux à une classe, le classifieur définit cinq poids de décision (priorité de décision). La priorité la plus élevée est associée à la décision dégagée par l'inspection de charge (la classe d'apparence d'un flux est celle prédite par inspection de charge). Le flux est attribué à la classe inconnue avec la deuxième priorité si les trois classifieurs n'arrivent pas à prendre une décision. Si la classification basée sur les ports et la classification statistique arrivent à la même décision, le flux est attribué à la classe correspondante à la troisième priorité. Lorsque ces deux derniers classifieurs ont différentes sorties, la décision prise en considération est celle du classifieur statistique. Finalement, la décision est basée sur le résultat de la classification basée sur les ports, si seulement, et contrairement aux deux autres classifieurs, cette dernière retourne une étiquette. Cette stratégie de classification a un impact positif sur la performance de la classification, cependant elle peut entraîner un coût de classification plus élevé comparativement à une approche hybride par étage ou simplement avec un classifieur individuel.

Par ailleurs, d'autres travaux de recherche ont eu pour objectif de donner un aperçu général des performances des différents algorithmes d'apprentissage machine, en effectuant des études comparatives. L'étude (Alshammari et Zincir-Heywood, 2010) révèle que l'algorithme C4.5 permet de classifier correctement le trafic VoIP avec un taux de 99 %. Le travail (Singh, Agrawal, et Sohi, 2013) comparant 5 algorithmes d'apprentissage machine démontre que le classifieur Bayes net offre une meilleure performance en termes de précision de classification et de temps d'apprentissage. De la même façon, une troisième étude (Gowsalya et Amali, 2014) démontre que les SVM permettent d'obtenir une performance de classification élevée. On peut déduire que la performance des algorithmes dépend, entre autres, de la nature des données ainsi que de leur pertinence. Le document (Ibrahim, Al Zuobi, Al-Namari, MohamedAli, et Abdalla, 2016) a jeté la lumière sur les éléments pouvant avoir un impact sur la classification du trafic réseau. En effet, les données peuvent influencer la précision de la classification si les données d'apprentissage et de test sont collectées de deux réseaux différents ou sur des périodes très espacées vu qu'il y a des caractéristiques statistiques qui changent de propriété (distribution) dans le temps ou d'un réseau à un autre tel que le temps d'interarrivé et la taille des paquets.

2.4 Échantillonnage de trafic

L'analyse du trafic et la supervision des réseaux sont deux tâches indispensables dans le processus de la gestion des réseaux et elles constituent une activité incontournable. Cette analyse se base essentiellement sur la collecte et l'extraction des informations contenues dans les paquets acheminés sur le réseau. Toutefois, le volume du trafic véhiculé sur les réseaux modernes devient de plus en plus important; ce qui rend le coût de la classification et de la visualisation du trafic de plus en plus important.

Il y a quelques années, les réseaux n'offraient que des débits relativement faibles (autour des 100MB tel que FDDI), ce qui pourrait justifier l'utilisation des sondes pour collecter le trafic. En effet, ces sondes copient tout le trafic traversant le nœud réseau pour le traiter par la suite. Cette approche est relativement facile à mettre en place et assure une précision des mesures

élevées, puisque ces dernières sont faites à partir de l'intégralité de trafic sur un réseau à faible vitesse.

Avec l'évolution d'Internet et l'apparition de nouvelles technologies qui permettent d'utiliser des réseaux de haut débit, de nouveaux enjeux sont apparus, notamment la gestion de données massives. Avec ces changements, l'analyse du trafic en utilisant l'approche classique n'est plus possible à cause des quantités des données grandissantes acheminées sur les réseaux haut débit, ce qui peut occasionner des perturbations comme la sursaturation des équipements. En effet, la collecte de tout le trafic n'est pas seulement coûteuse durant le processus de traitement, mais il peut consommer énormément de ressources réseautiques, particulièrement, les ressources mémoires sur les nœuds de réseau où s'effectue la collecte du trafic. Les équipements peuvent être surchargés avec cette tâche supplémentaire d'une part, et la bande passante peut être significativement affectée d'autre part, si les données collectées devraient être envoyées vers un serveur de sauvegarde distant.

2.4.1 Techniques d'échantillonnage de trafic

Plusieurs techniques d'échantillonnage ont été abordées par la littérature. En particulier; l'échantillonnage systématique, l'échantillonnage aléatoire et l'échantillonnage aléatoire adaptatif.

2.4.1.1 Échantillonnage systématique

L'échantillonnage systématique suit une règle simple, au sens où les échantillons sont sélectionnés d'une manière périodique selon une fonction déterministe. En effet, chaque $k^{ème}$ élément est sélectionné à partir d'un point de départ choisi au hasard entre 1 et k. Considérons une population constituée de N éléments numérotés de 1 jusqu'à N; la constante k est appelé le pas d'échantillonnage et elle est égale au nombre entier le plus proche à N/n. L'échantillonnage systématique de paquets peut être fait de deux façons à savoir : 1) à base nombre de paquets, dans ce cas l'échantillon; est formé par la sélection de chaque $k^{éme}$ paquet 2) à base du temps, dans ce cas la sélection d'un paquet est effectuée chaque période de

temps. Il est noté dans (Kimberly C Claffy, Polyzos, et Braun, 1993) que la première approche est plus précise que la deuxième en terme de l'estimation des paramètres du trafic. L'échantillonnage systématique est plus facile à effectuer et il peut assurer un bon niveau de précision car il peut assurer une bonne répartition des éléments dans l'ensemble de l'échantillon. Par contre il risque de biaiser les données à cause de la périodicité.

2.4.1.2 Échantillonnage Aléatoire

Échantillonnage aléatoire simple

Cette technique consiste à choisir des individus de la population de telle sorte que tous les éléments ont la même chance de figurer dans l'échantillon. De ce fait cette technique est appelé parfois n-out-of-N. La population est divisée en un ensemble de groupe de N élément. Pour chaque groupe n élément sont sélectionnés aléatoirement.

La méthode d'échantillonnage 1-out-of-N est utilisée par outils de collecte de paquets ou de flux sFlow décrits dans (InMon, 2004) et *Random Sampled NetFlow* (Cisco, 2005).

Échantillonnage probabiliste

Cette technique choisis les échantillons en fonction d'une probabilité de sélection prédéfinie. Pour l'échantillonnage probabiliste uniforme, chaque paquet est sélectionné indépendamment avec une probabilité P fixe. Dans le cas où la probabilité p dépend de l'entrée (par exemple le contenu des paquets), l'échantillonnage est dit échantillonnage probabiliste non uniforme. Cette approche non uniforme peut être utilisée pour pondérer les probabilités d'échantillonnage afin de stimuler la chance des paquets qui sont rares mais sont jugés importants à être sélectionnés.

Échantillonnage stratifié

L'échantillonnage stratifié consiste à diviser la population en groupes homogènes (appelés strates), qui sont mutuellement exclusif, puis à sélectionner à partir de chaque strate des échantillons indépendants. N'importe quelle des méthodes d'échantillonnage mentionnées précédemment peut être utilisée pour sélectionner l'échantillon à l'intérieur de chaque strate. La méthode d'échantillonnage peut varier d'une strate à une autre. L'échantillonnage aléatoire simple est utilisée pour former l'échantillon à l'intérieur de chaque strate. Cet échantillonnage est appelé échantillonnage aléatoire simple stratifié. La raison principale visant à diviser la population en strates est de rendre la stratégie d'échantillonnage plus efficace. Le gain de cette technique peut être illustré à travers cet exemple; soit un jeu de données (trafic) divisé selon la taille des paquets où les paquets similaires sont regroupés dans la même strate. Pour estimer la taille moyenne des paquets pour chaque strate, il suffit de prendre des échantillons relativement petits des strates et d'en calculer la moyenne de la taille des paquets de la totalité de la population. En contrepartie si un échantillonnage aléatoire simple est appliqué sur la population entière sans effectuer de stratification, il faudrait un échantillon plus grand afin d'obtenir une estimation du même degré de précision pour la taille moyenne de paquets sur le réseau.

2.4.1.3 Échantillonnage aléatoire adaptatif

L'idée principale de l'échantillonnage aléatoire adaptatif est de combler les lacunes des techniques échantillonnages décrites précédemment. En effet, la plupart de ces techniques sont utilisées, plus au moins, d'une façon statique comme l'échantillonnage systématique où les éléments de l'échantillon sont sélectionnés périodiquement selon un pas d'échantillonnage prédéfini. Ceci peut occasionner un sous-échantillonnage qui rend la détection de certain comportement du réseau plus difficile voire impossible, ou un sur-échantillonnage qui peut générer une grande quantité de données, qui risque d'être perturbé à cause de saturation des équipements, notamment au moment des transmissions du trafic en rafales. Le taux

d'échantillonnage élevé peut aussi influencer le trafic de manière générale au sens où le trafic peut être considéré anormal.

Bien que toutes ces techniques d'échantillonnage sont destinées à réduire le nombre données collectées et traitées toute en garantissant un certain niveau de performance dans le processus d'analyse et d'estimation des caractéristiques du trafic, ces techniques ne prennent pas en compte les contraintes liées à la capacité des équipements et le compromis nécessaire entre le taux d'échantillonnage et le taux d'erreur d'estimation des caractéristiques du trafic requis. Ceci rend l'introduction des techniques d'échantillonnage adaptatif nécessaire. L'échantillonnage aléatoire adaptatif repose sur des méthodes heuristiques ainsi que sur les techniques de prédiction des caractéristiques du trafic futur pour ajuster le taux d'échantillonnage. Des solutions d'échantillonnage adaptatif du trafic sont proposées dans la littérature. Ces solutions reposent sur les variations des ressources réseau pour adapter le taux d'échantillonnage tel que le taux d'utilisation de CPU (Central Processing Unit) et le temps d'inter-arrive des paquets (Drobisz et Christensen, 1998) et la prédiction de la charge du trafic dans le prochain intervalle de temps (Choi, Park, et Zhang, 2003).

Les méthodes et les standards d'échantillonnage et de collecte de trafic s'appuient principalement sur l'une des techniques décrite plus haut tels les standard populaire sFlow et Netflow

2.4.2 Standard sFlow

Le Sflow, l'acronyme de (*Sampled Flow*), est définit par Inchon (InMon) comme étant « un standard pour la surveillance des réseaux à haute vitesse». Cette technologie s'installe dans les équipements de réseau et offre une visibilité complète sur l'activité du réseau, en permettant sa gestion et son contrôle. Il est conçu pour être implémenté dans la majorité des dispositifs réseau et pour fournir des statistiques d'une manière périodique de sorte que tout le trafic d'un réseau peut être caractérisé.

Le standard sFlow est basé sur l'échantillonnage aléatoire de paquets. En particulier 1-out-of-N où le commutateur capture un paquet parmi n paquets successifs par interface et l'envoie vers le collecteur. Ce taux d'échantillonnage est configurable en choisissant le nombre N, si aucune configuration n'est spécifiée, le protocole utilise les taux d'échantillonnage selon la vitesse de l'interface.

Le système sFlow est composé essentiellement de deux composantes dont la première est l'agent sFlow qui est incorporé dans les commutateurs ou les routeurs sujets de surveillance et la deuxième est le collecteur sFlow (Figure 2.13).

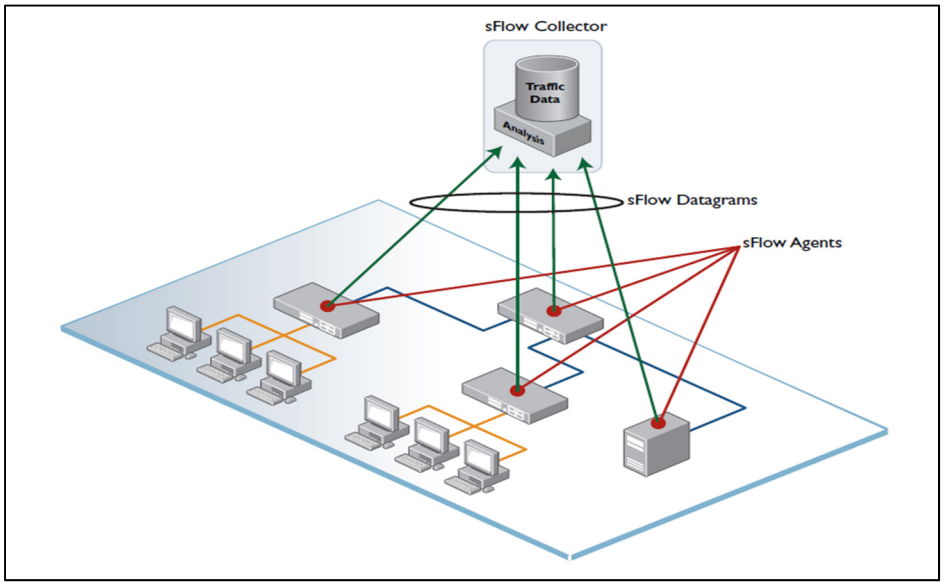

Figure 2.13 Architecture de sFlow
Tirée de Telesis (2013)

Agent sFlow

L'agent sFlow est incorporé dans un commutateur ou un routeur. Il s'occupe de la capture du trafic par le biais d'échantillonnage selon la vitesse du lien si aucun taux n'est précis. Pratiquement, l'échantillonnage de paquet se fait à base de la méthode 1-out-of-N et il est possible de personnaliser le taux de prélèvement sur chaque port.

Les données collectées, à savoir les paquets échantillonnés et les statistiques de chaque port sont encapsulées dans le datagramme sFlow. Les paquets sont échantillonnés en fonction du taux d'échantillonnage, et les statiques de chaque interface sont prélevées dans des périodes maximums dites *polling Interval* de 20 s par défaut. Les agents sFlow envoient les datagrammes sFlow au collecteur en fonction du taux d'échantillonnage et la taille maximale de datagramme qui est de 1400 octets par défaut.

Le datagramme est composé de trois champs :

- L'en-tête contient des informations ordinaires permettant d'acheminer le datagramme vers le collecteur où on trouve les adresses IP sources (agent sFlow), adresse destination (Collecteur) et la version du protocole sFlow en plus d'autres informations;
- *Packets Samples* contient essentiellement les en-têtes des paquets échantillonnés durant le *polling Interval*. Il peut contenir aussi une partie de la charge des paquets;
- *Interface counter* permet d'envoyer les statistiques relatives à chaque interface. Il contient l'indice de l'interface, son statut et le nombre d'octets entrants et sortants, etc.

Collecteur sFlow

Le collecteur sFlow a pour objectif de stocker les statistiques reçues d'un ou plusieurs agents sFlow. Il peut présenter une analyse du trafic à l'administrateur du réseau s'il est doté d'une fonction analyseur. La majorité de ces collecteurs peuvent utiliser SNMP pour configurer les agents et pour résoudre les numéros des index des interfaces de manière à pouvoir présenter les informations relatives aux noms des interfaces physiques associées.

2.4.3 Netfow

NetFlow (System) est une architecture de surveillance des réseaux développé par Cisco afin de collecter des informations sur les flux de réseau générés par les routeurs et les commutateurs compatibles avec NetFlow. Il définit un format d'exportation de ces informations nommé *NetFlow services export format* (format d'exportation des services

NetFlow) ("Netflow", 2016). Tel qu'illustré à la Figure 2.14, une architecture Netflow intégrale consiste en trois composantes essentielles dont la première est nommé *Netflow Exporter*. Cette dernière permet de former des flux IP en se basant sur l'observation des paquets transitant à travers le routeur. Le *NetFlow Exporter* maintient une cache de flux utilisée pour sauvegarder les statistiques de trafic basée sur les flux. Les statistiques de chaque flux actif sont maintenues dans la cache et sont mis à jour lorsque les paquets de chaque flux sont commutés. Les flux expirés sont exportés, périodiquement, par le biais du protocole (UDP) et SCTP (Stream Control Transmission Protocol datagrammes), vers un collecteur appelé *NetFlow Collector*. Ce dernier reçoit et sauvegarde les enregistrements des flux envoyé par le *Netflow Exporter* après avoir les encapsuler dans un datagramme UDP (*datagrame Netflow*) (Cisco, 2016a).

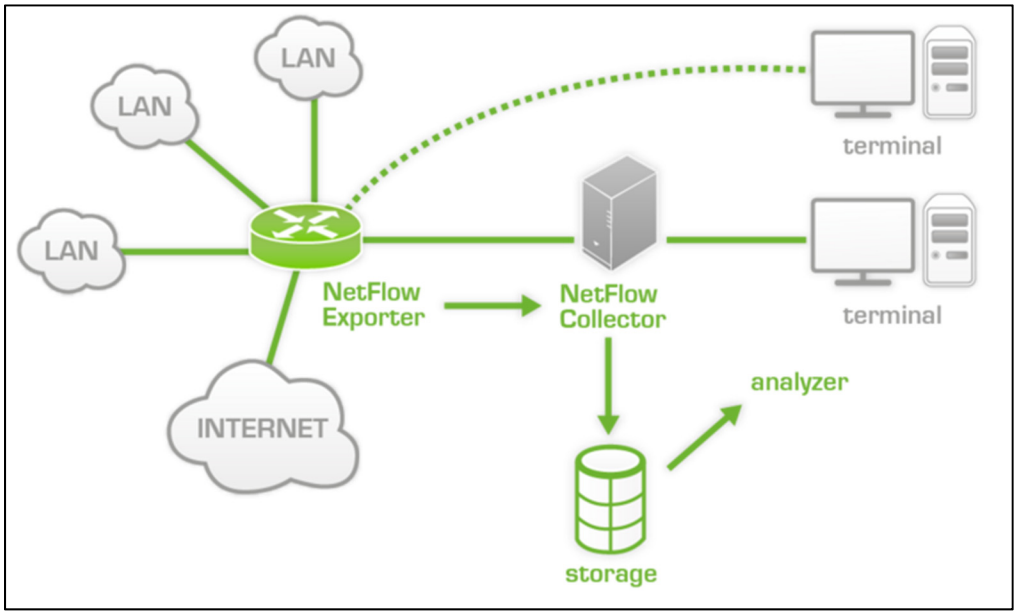

Figure 2.14 Architecture de Netflow
Tirée de Tecnológicas (2016)

Netflow et l'échantillonnage

Initialement le protocole *Netflow* est conçu pour traiter tous les paquets qui traversent les routeurs surtout les versions 1, 5 et 8. Cette approche de création des flux par le biais d'un traitement de tous les paquets permet de fournir des statistiques plus précises. Par contre

quand les réseaux deviennent plus larges et génèrent plus de données, l'implémentation de cette approche peut occasionner des perturbations du fonctionnement du réseau à cause de l'utilisation importante de ressources pour collecter le trafic. Pour remédier aux limitations des premières versions de Netflow, Cisco a conçu une autre version qui permet de prendre en considération la surcharge du réseau. C'est le Netflow v9. Dans cette version il est possible de configurer une méthode d'échantillonnage pour sélectionner certains paquets parmi l'ensemble des paquets commutés, à partir des échantillons capturés. Netflow peut créer la table des flux. En effet, Cisco a introduit plusieurs techniques d'échantillonnage dont celle du sFlow soit l'échantillonnage aléatoire simple plus précisément 1-out-of-N. Cette dernière est implémentée dans *Random Sampled Netflow* (Cisco, 2005) et Netflow Lite (Router-Switch, 2015).

Comparaison de sFlow et Netflow

Les fonctions permises par chacun de ces deux protocoles sont résumées dans le tableau 2.1. Ce dernier démontre que sFlow supporte presque toutes les fonctions comparativement à avec Netflow. Netflow v9 (Cisco, 2016a) permet de mettre en place *l'Interface Counters* (capture des informations des ports ou des interfaces des routeurs). Mais en général sFlow donne plus de détails car il se base sur la capture des paquets, contrairement à Netflow qui se base sur les flux. Ce dernier peut être plus efficace si on envisage de faire une analyse de flux car l'étape de création des flux est effectuée par le protocole. Contrairement à Netflow, qui se focalise plus sur la couche 3, sFlow permet de collecter les statistiques de toutes les couches du modèle ISO. Avec le protocole Netflow il est ainsi possible d'analyser en plus du trafic IP normal, le trafic MPLS et IPv6.

Tableau 2.1 Comparaison sFlow et Netflow
Adapté de Brocade (2009) et So-In (2009)

	Netflow	**sFlow**
Capture de paquets	non	partiel
Compteurs d'interface	non	oui

	Netflow	**SFlow**
Échantillonnage	Partiel	oui
Protocoles:		
en-têtes de paquets	non	oui
Ethernet/802.3	non	oui
IP/ICMP/UDP/TCP	oui	oui
Layer2:		
Interface d'entrée / sortie	oui	oui
Priorité entrée / sortie	non	oui
Input/Output VLAN	non	oui
Layer3:		
Source sous-réseau/préfixe	oui	Oui
Destination sous-réseau/préfixe	oui	Oui
Prochain saut	oui	Oui
Collecte de données en temps réel	partiel	Oui
Configuration sans SNMP	oui	Oui
Configuration via SNMP	non	Oui
Taux d'échantillonnage par interface	non	Oui
Faible coût	non	Oui
Évolutif	non	Oui

2.5 Conclusion

Ce chapitre a fait la synthèse de la visualisation des données volumineuse et multidimensionnelle, et en particulier celles issue des réseaux informatiques ou encore de trafic réseau. Dans le contexte de cette recherche, la revue de littérature s'est focalisée sur la visualisation d'information pour surveiller les infrastructures réseautique par l'exploration des données de trafic.

De plus, une revue de la littérature concernant la classification du trafic a aussi été présentée ainsi que son rôle très important dans la gestion des systèmes de qualité de service. Ceci permet, d'une manière générale, de mieux connaitre le trafic véhiculé sur le réseau. Le chapitre suivant présente la méthodologie de recherche suivie afin de résoudre cette problématique

CHAPITRE 3

MÉTHODOLOGIE DE RECHERCHE

3.1 Introduction

Ce chapitre est consacré à la présentation et à la discussion des solutions proposées pour les quatre problématiques décrites à la section 1.3. Ces solutions permettent de mettre à disposition tous les éléments nécessaires pour comprendre le comportement d'un réseau donné et avoir une visibilité des activités sur le réseau tout en prenant en compte les contraintes de temps réel.

La suite du chapitre présente des méthodes pour résoudre les différentes problématiques et atteindre les objectifs de la recherche.

3.2 Description générale de la plateforme de visualisation de trafic

3.2.1 Modules de la plateforme de visualisation de trafic

Les différentes composantes de sont décrites à la Figure 3.1. Le cadre de travail comporte quatre modules : le module de collecte de trafic; le module de préparation de données; le module d'analyse de données et le module de représentation graphique.

- le module de collecte de trafic, qui instaure une méthode d'échantillonnage sFlow, permet de réduire la quantité de données à capturer puis de les transférer vers une base de données pour qu'elles soient traitées par les autres modules;
- le module de préparation de données effectue le prétraitement et la préparation du trafic brut avant d'entamer le processus d'analyse. Ce module fait la conversion des formats de données à l'entrée en autre format compatible pour faciliter leur lecture et leur traitement par le reste des composantes, par exemple le décodage des datagrammes sFlow qui

enveloppe les informations de plusieurs échantillons compressés dans un seul datagramme, contient aussi la partie prétraitement de classification du trafic. Ainsi, cette opération est basée sur les flux, elle consiste en la construction des caractéristiques de flux et de la sélection des attributs. Cette dernière est importante pour la classification car, parmi les variables décrivant les flux ou les observations de manière générale, seulement certaines d'entre elles contiennent des informations pertinentes et distinctives;

- le module d'analyse de données permet d'examiner, de traiter des données, et d'en faire ressortir des informations utiles selon les niveaux d'analyses illustrés à la Figure 3.2. Cette composante est déclinée en trois parties : i) la surveillance réseaux en particulier les caractéristiques générales du réseau (c.-à-d. le volume de trafic, connexion établies, et la distribution de la taille de paquets), ii) l'analyse des communications au niveau de transport du modèle OSI et TCP/IP, et plus particulièrement la tendance du trafic de point de vue des flux, iii) la classification de trafic au niveau applications;

- le module de représentation graphique a pour objectif d'assister l'administrateur réseau en présentant des graphes simples et riches représentant les résultats du module d'analyse de données. Cette dernière étape du processus de visualisation fait appel à un processus de transformation géométrique de données (*data rendering*) ainsi que des techniques de visualisation. La transformation géométrique doit refléter la nature des données en prenant en considération le type d'information à communiquer via des graphes. Par exemple, une relation peut être représentée par une ligne reliant deux ou plusieurs nœuds qui sont représentés par d'autres formes géométriques par exemple un cercle ou un point pour optimiser l'utilisation d'espace. En outre, le choix d'une technique de visualisation est un élément critique de cette recherche du fait qu'elle est responsable de faire ressortir des tendances, des patrons, des corrélations ou d'autre relations dominantes dans ces données.

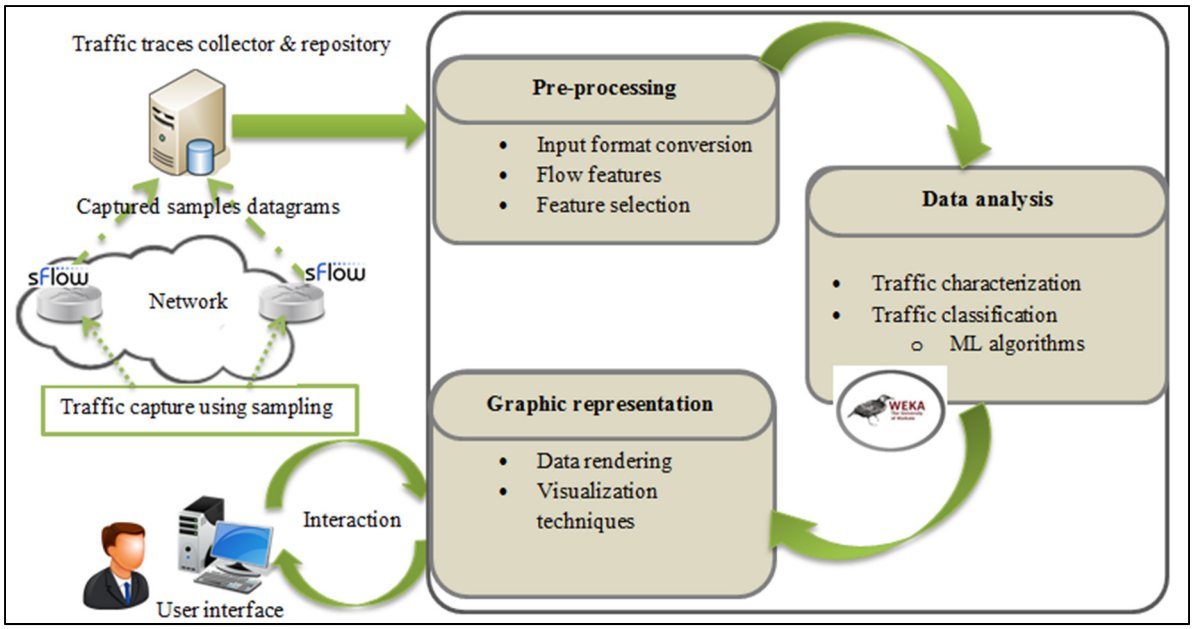

Figure 3.1 La plateforme de visualisation de trafic

3.2.2 Niveaux d'analyse

La plateforme de visualisation de trafic proposée offre différentes fonctionnalités pouvant décrire le mieux possible la tendance du trafic dans un réseau. Il offre, comme avancée, trois niveaux de caractérisation de trafic dans le but de comprendre et surveiller le comportement des réseaux et chaque niveau correspond à l'analyse de trafic à une échelle déterminée tel qu'il est présenté à la Figure 3.2. La proposition comporte trois niveaux d'analyse de trafic : les métriques générales, les caractéristiques de la couche transport et les caractéristiques de la couche application.

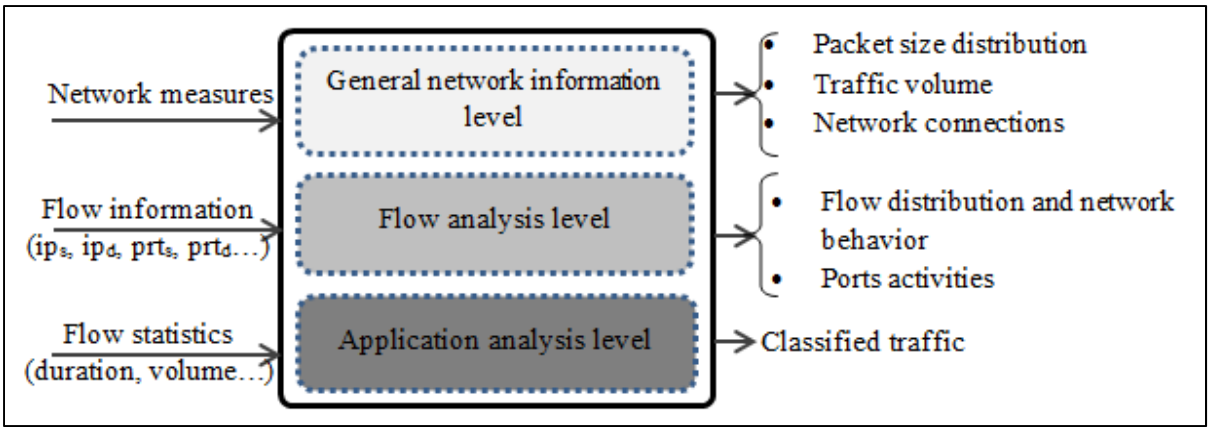

Figure 3.2 Les niveaux d'analyse

3.2.2.1 Métriques générales

Parmi les mesures rapportées par la plateforme de visualisation on trouve le débit des liens, la distribution de la taille de paquets ainsi que les connexions actives sur le réseau.

Débit

Le débit soit une information centrale. Il est important et incontournable pour la supervision des réseaux, car il donne une idée du volume de trafic véhiculé sur le réseau et de ses variations dans le temps. En prenant en considération que la charge des réseaux suit le cycle des activités humaines et manifeste des patrons bien particuliers sur une échelle hebdomadaire et annuelle, cette caractéristique peut être un excellent indicateur de base de la santé du réseau. Celui-ci peut être mesuré en termes de paquet ou d'octets accumulés dans un intervalle de temps fixe. Aussi, une étude du phénomène éléphant et souris a montré que 1 % des flux (flux larges) sont responsables de plus de 90 % de volume de trafic, tandis que les petits flux (c.-à-d. souris) contribuent moins dans la charge de trafic même ils sont plus abondants (99 % des flux génèrent 4 % du volume total du trafic) (Rivillo, Hernández, et Phillips, 2005). Cette mesure peut être exploitée pour anticiper l'arrivée imminente d'un gros flux en analysant simultanément le débit des liens en termes de paquets et d'octet.

Ainsi, cette mesure permet d'avoir une idée concernant la nature des pics ou des rafales qui peuvent se produire sur les liens. Ces phénomènes constituent des menaces probables pour la stabilité du réseau en se référant par exemple, à leur durée et à l'heure de la journée, ou ils surviennent aux différentes périodes de l'année, ou encore s'il y a d'autres évènements qui peuvent augmenter le taux de connectivité de la population à Internet.

Distribution de la taille de paquets

La distribution de la taille de paquets est représentée comme étant une métrique générale. Elle renferme plusieurs informations qui peuvent révéler, entre autres, la façon dont des internautes utilisent le réseau ou encore la nature des applications qui l'utilisent. chacune type d'application possède une empreinte correspondante à la distribution de la taille des paquets (Hernandez et Serrano, 2015).

Historique des connexions réseau

Les caractéristiques générales incluent l'historique des connexions réseau ainsi que leurs statistiques. Notamment le nombre de paquets et le volume de trafic. Ces connexions désignent principalement tous les flux échangés entre deux hôtes. Ainsi, elles peuvent être vu comme la séquence de paquets ayant les même adresse source et adresse destination.

3.2.2.2 Caractéristiques de la couche transport

Au niveau du transport, l'analyse du trafic est basée essentiellement sur les flux échangés entre les hôtes, que ce soit, en interne ou externe, c'est à dire entre les hôtes dans un même réseau ou entre les machines locales et celles dans des réseaux externes. Plusieurs caractéristiques peuvent être étudiées à ce niveau, telles que la répartition graphique des flux et la détection du trafic malicieux.

Répartition des flux

Dans ce contexte, un flux est défini comme étant l'ensemble de paquets ayant les mêmes 5-tuples (adresse source, adresse destination, port source, port destination, protocole). La durée et la taille d'un flux correspondent, respectivement, à la période entre l'arrivée du premier paquet et l'arrivée du dernier paquet et au nombre d'octets accumulés durant cette période. L'exploration de ces caractéristiques peut expliquer l'état du réseau pendant une période de temps et peut être un indicateur du type de trafic acheminé sur le réseau. Par exemple, l'existence des flux éléphants peut être déduite en analysant, et selon la définition d'un flux éléphant adopté, la durée de chaque flux et sa contribution dans le volume du trafic.

Analyse de ports

Dans le même ordre d'idée, une autre analyse de flux TCP peut être effectuée, mais cette fois en se basant sur la répartition des ports. Cette dernière est motivée par le fait que le balayage ou le «*scan»* de ports est une technique populaire de trafic malicieux. Cette technique peut être effectuée selon trois méthodes ; de façon horizontale appelée aussi balayage de réseau où le pirate envoie des requêtes au même port sur toutes les machines de réseau. Un balayage vertical balaye tous les ports sur une seule machine. Une numérisation collaborative combine les deux premières stratégies de balayage pour découvrir la/les machine(s) vulnérable(s). Chacune d'entre elles est caractérisée par une allure de distribution du trafic dans l'espace composée des adresses sources et destinations et ports sources et destinations où le trafic dans le premier type a tendance à être dense autour de la machine de balayage (plusieurs machines pirates), tandis qu'il (trafic) se concentre dans le second cas autour du port cible (Houerbi, 2009) et ce qui s'explique par le fait que l'attaquant interroge toutes les machines sur le même numéro de port, dans la troisième stratégie le trafic se concentre sur les adresses malveillantes ainsi que sur les adresses des machines cibles.

Comme il sera présenté un peu plus loin dans ce chapitre, la représentation graphique de la distribution du trafic dans l'espace des quatre coordonnées (adresse source, adresse destination, port source, port destination) peut faire ressortir un trafic malicieux visant à chercher des machines vulnérables par balayage de port. Ceci permet également de marquer

les adresses sources de menaces de sécurité. Par ailleurs, une contribution de cette recherche vise à proposer une nouvelle représentation de graphe permettant de mettre en clair ce genre de trafic (scan) qu'il soit interne ou externe.

3.2.2.3 Caractéristiques de la couche application

Le troisième niveau d'analyse de trafic s'intéresse à la caractérisation du trafic au niveau d'application, plus particulièrement à la répartition du trafic par application. Ceci vise à identifier et à classifier le trafic dans plusieurs classes. La classification des applications est un élément clé dans l'ingénierie de trafic et incontournable dans les mécanismes de qualité de service. Par exemple, pour l'allocation des priorités à différentes classes (classe de services) ainsi que leurs répartitions dans les files d'attente. En outre, la connaissance des applications est indispensable pour la saine gestion des réseaux, et pour la compréhension de leurs comportements. Elle est aussi nécessaire pour les maintenir. La caractérisation et la classification de trafic permettent à l'administrateur d'appliquer des règles sur les trafics qui peuvent occasionner la congestion ou qui consomment plus de bandes passantes sans être prioritaires. Avec l'arrivée de la technologie SDN, des règles qui gèrent l'acheminement et les priorités de trafic peuvent être incorporés dans les contrôleurs plus facilement. Par exemple, les flux de trafic P2P peuvent être transmis seulement si les ressources sont disponibles et ils sont bloqués autrement avec l'objectif de satisfaire les contraintes des applications prioritaire telle que la voix. Ainsi, la classification du trafic doit s'effectuer en temps réel afin d'agir le plus tôt possible sur les performances des réseaux et d'apporter des actions correctives rapidement.

Pour réaliser la plateforme de visualisation décrite plus haut, nous avons suivi la méthodologie ci-dessous qui englobe l'ensemble des méthodes et des approches sélectionnées ainsi que les solutions proposées. Notant que le trafic utilisé dans cette recherche est collecté sur les nœuds réseau d'une manière passive.

3.3 Modèle d'échantillonnage adaptif

Bien que sFlow présente un ensemble d'avantages et un niveau de flexibilité élevé en donnant l'accès aux informations des différentes couches de réseau, à l'exception de la couche physique, et en plus d'être un standard «open source» supporté par la majorité des vendeurs d'équipements réseaux favorisant la virtualisation et l'implémentation pratique de sa politique d'échantillonnage, il risque d'être une limitation pesante du fait qu'il se base essentiellement sur les vitesses théoriques des interfaces pour préciser le taux d'échantillonnage, qui se fasse lui-même de manière constante.. L'algorithme d'échantillonnage de sFlow repose essentiellement sur le taux d'erreur d'échantillonnage tel qu'illustré à Figure 3.3, ce qui dénote une grande évolutivité, du fait que la performance d'échantillonnage dépend du nombre d'échantillons sélectionnés. Par exemple pour un taux d'erreur égal à 5%, le nombre d'échantillons doit être égal ou supérieur à 1500 paquets au détriment de la vitesse du réseau.

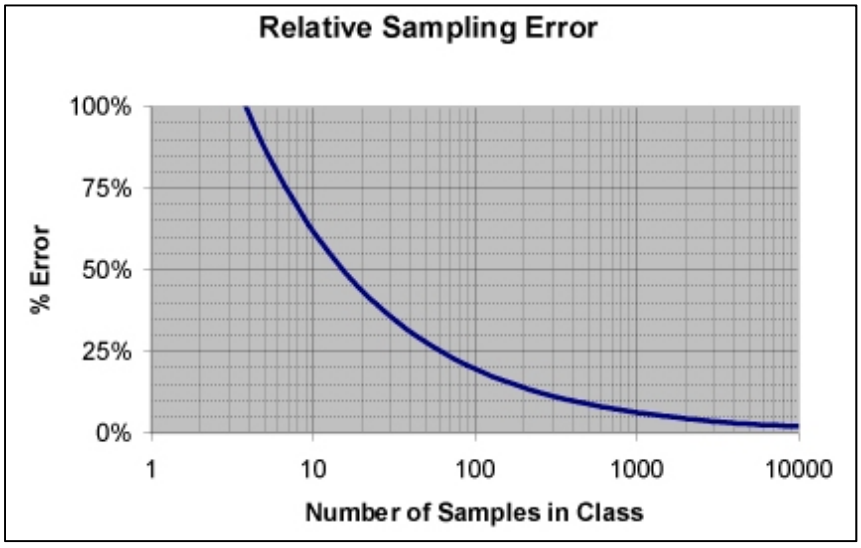

Figure 3.3 Erreur relative d'échantillonnage
Tirée de InMon (2004)

Toutefois, cet écart réside dans la phase de l'implémentation de cet algorithme où les interfaces sont censées acheminer le maximum de données et le taux d'échantillonnage est déterminé d'une manière statique. Ceci permet d'atteindre le seuil du nombre de paquets

échantillonnés assurant un taux d'erreur acceptable. En pratique, la quantité des données transitant sur un réseau fluctue constamment ce qui rend l'affectation statique de taux d'échantillonnage inefficace.

Afin d'améliorer cet aspect, une méthode dynamique pour adapter le taux d'échantillonnage en fonction des débits au lieu des vitesses a été proposée. Ainsi, au lieu de se référer aux vitesses des interfaces, le taux d'échantillonnage est déterminé en se basant sur les débits effectifs afin de réduire le taux d'erreur d'échantillonnage. De cette façon, le taux d'échantillonnage associé à une interface changera de valeur en fonction des fluctuations de quantité de données échangées sur un lien donné. Par exemple, si un lien de réseau est surveillé pendant un intervalle de temps, indépendamment de la capacité maximale de l'interface, et si l'on considère un taux d'erreur d'échantillonnage inférieur ou égal à 5%, cela se traduit à un nombre de paquets échantillonnés supérieur ou égale à n (n=1500). Le taux d'échantillonnage satisfaisant cette condition est calculé par la formule (3.1) :

$$T_{et} = \begin{cases} \dfrac{n}{D_t \times d} \times 100 & \text{si } D_t \times d \geq n \\ 100 & \text{sinon} \end{cases} \qquad (3.1)$$

Où T_{et} est le taux d'échantillonnage, D_T le débit à l'instant t et d représente la durée de supervision de trafic.

Contrairement à la méthode statistique qui peut occasionner des erreurs élevées dans certains cas, la méthode dynamique d'association des taux d'échantillonnage aux interfaces permet de s'adapter aux variations réseautiques et d'améliorer par conséquent la performance d'échantillonnage.

3.4 Classification des données massives en temps réel

3.4.1 Approche de classification

La classification du trafic suscite l'intérêt des chercheurs exerçant dans le domaine de l'ingénierie du trafic. Naturellement, une bonne gestion de réseaux et des ressources informatiques nécessite une connaissance de plus en plus détaillée et précise du trafic. De ce fait, la différentiation du trafic continue à être élément essentiel et indispensable pour l'analyse et l'évolution des réseaux actuels. La classification en temps réel, avec un taux de précision élevé, est un souci actuel et qui intéresse les chercheurs d'ingénierie de trafic. L'état de l'art, a précisé que les approches basées sur les ports standards et l'inspection de la charge se heurtent à plusieurs obstacles qui sont beaucoup plus répandus dans les réseaux actuels qu'au début d'Internet. Par exemple, concernant les applications utilisant des ports dynamiques et le chiffrement. Pour ces raisons, les approches basées sur l'apprentissage machine sont populaires. Elles exploitent des informations statistiques pour entraîner des modèles et classifier par la suite les données. Ceci a pour but d'apprendre le comportement de chaque type d'application pour générer des modèles prédictifs au lieu d'analyser l'information extraite d'un en-tête de paquet comme les numéros de ports ou de la charge comme la signature de l'application.

Cette recherche propose une approche reposant sur les algorithmes d'apprentissage automatique et les caractéristiques des sous-flux qui constituent un flux père afin d'identifier et classifier ce dernier le plus tôt possible. Cette approche a été publiée par Kgunen dans la phase d'apprentissage d'un classifieur (Thuy T Nguyen et Armitage, 2006). Cette recherche vise à généraliser cette approche afin d'effectuer la classification temps réel du trafic en utilisant uniquement le premier sous-flux détecté lors de la phase de classification.

La classification élaborée dans cette recherche repose sur les caractéristiques des sous-flux autant dans la phase d'apprentissage que dans la phase de classification. Tel l'illustre à la Figure 3.4, la génération du modèle est effectuée à partir des sous-flux comme données d'entraînement. Dans la phase de classification l'échantillonnage adaptif a été utilisé pour

sélectionner seulement le premier sous-flux détecté pour classifier le flux entier. Les avantages de cette méthode sont :

La classification en temps réel : même si l'entraînement du modèle est effectué hors ligne, la classification est fortement adaptée aux processus temps réel, du fait qu'il nécessite seulement le premier échantillon ou sous-flux pour prédire la classe du flux père.

Le début de flux : l'utilisation des sous-flux permet de surmonter la limitation relative au début des flux. Typiquement, un classifieur ne peut pas assumer qu'il va recevoir les premiers paquets de chaque flux. Ainsi, cette problématique peut causer des problèmes de classification. En revanche, le calcul des caractéristiques des sous-flux peut commencer à n'importe quel point durant la période active du flux.

L'échantillonnage : la considération des sous-flux favorise un type d'échantillonnage qui peut se faire en ligne directement à partir des interfaces réseau ou bien à partir des fichiers de sauvegarde du trafic. Les flux sont subdivisés en flux plus petits et seulement le premier échantillon pour la prédiction des classes est utilisé.

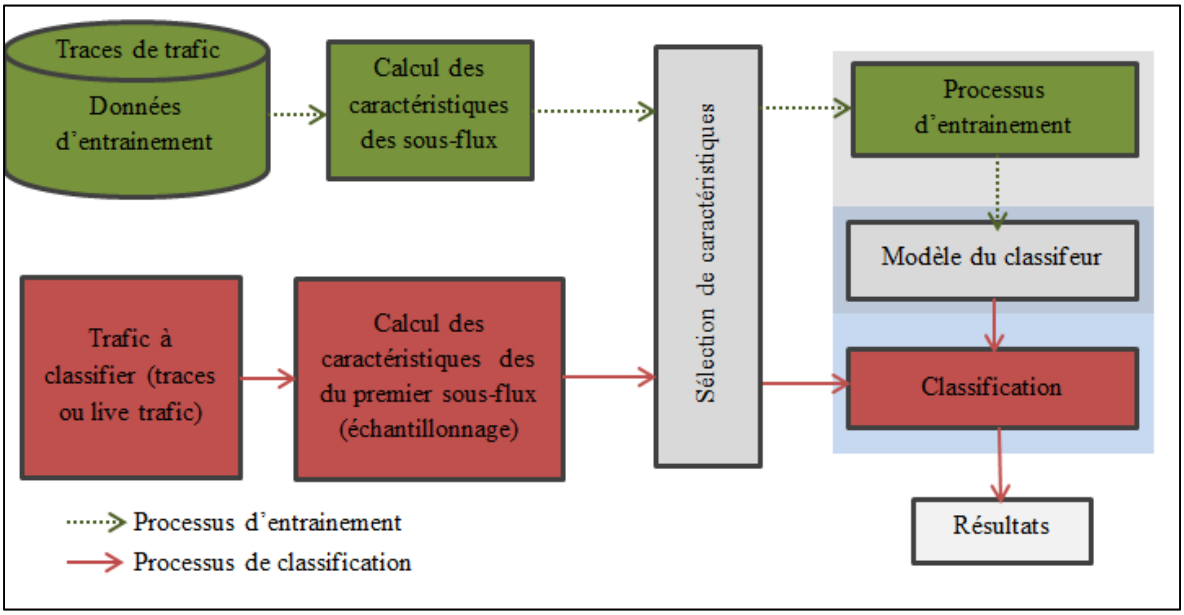

Figure 3.4 Schémas de classification

La Figure 3.4 synthétise l'utilisation des sous-flux et de l'échantillonnage afin de classifier le trafic en temps réel. La première étape consiste à construire le modèle en déterminant un seuil de séparation, pour la prise de décision, en utilisant des vecteurs de caractéristiques issues des sous-flux. Des données d'entraînement sont utilisées et la fonction définissant le classifieur ou de façon générale algorithme d'apprentissage machine est découverte. Notant que les sous-flux d'un même flux père peuvent avoir des caractéristiques différentes, et dans ce travail, deux catégories d'algorithmes, particulièrement les arbres de décision et les SVM ont été expérimentés, dans l'optique de sélectionner celui ayant les meilleurs résultats pour ce type d'apprentissage. La deuxième phase permet de tester les modèles entrainés. Pour ce faire, seulement le premier sous-flux est pris en considération pour associer le flux père à une classe k. Cet échantillonnage permettra une classification temps réel du trafic. En plus de la génération du modèle de classification, le processus d'apprentissage fait appel à deux étapes principales de préparation des données. Dans ce contexte, il s'agit de génération et de sélection des caractéristiques des flux.

3.4.2 Génération des caractéristiques de flux

Cette approche est motivée par la nature des flux réseau. Un flux est défini comme une série de paquets ayant des caractéristiques communes (spécification du flux) transférées dans un intervalle de temps inférieur ou égal à un seuil appelé *Timeout* d'après (Kimberly C. Claffy, Braun, et Polyzos, 1995). Cette définition permet de délimiter les flux et d'éviter de garder ceux qui sont inactifs longtemps dans la table de cache des routeurs. Un flux est dit inactif quand il n'y a pas de nouvelle arrivée de paquet correspondant à son entrée existante avec un état actif dans la table de cache pendant une durée appelée durée de repos ou (*idle time*), et il est déclaré terminé et supprimé de la table de cache si cette durée dépasse le seuil (*Timeout*). Une multitude de spécifications ou de caractéristiques de flux peut être définie, selon son utilisation, en se basant sur les quatre dimensions citées dans (Kimberly C. Claffy *et al.*, 1995). Un exemple de flux traditionnel et unidirectionnel peut défini par 5-tuples (adresse source, adresse destination, port source, port destination, protocole).

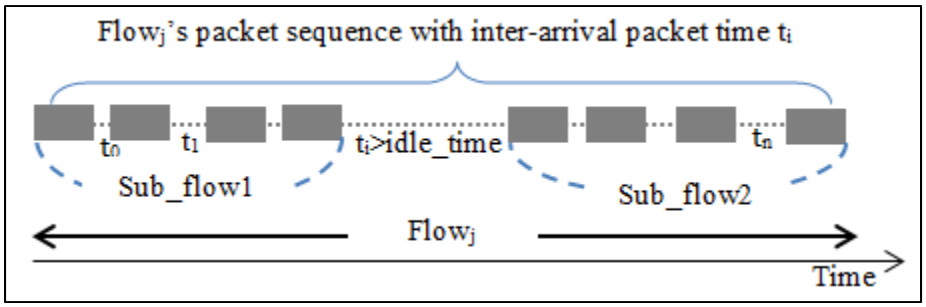

Figure 3.5 Décompositions de flux en sous-flux

Cette flexibilité de la définition des flux et les paramètres de construction dont le temps de repos (*idle-time)* et le seuil maximum «*Timeout»* sont des points favorisant une classification de trafic en temps réel. Ces paramètres permettent de construire les entités (sous-flux) de cette approche de classification utilisant l'étude de (Thuy T Nguyen et Armitage, 2006) effectuée sur l'entraînement des classifieurs par les sous-flux. Le principe tel que le démontre la Figure 3.5 repose sur le temps de repos et le «*Timeout»* pour décomposer le flux entier en flux plus petits nommés sous-flux. Le «*timeout»* sert à délimiter le flux entier où ce dernier est maintenu dans la table tant que sa durée de repos ne dépasse pas le seuil. Quant au temps de repos, d'après (Williams, Zander, et Armitage, 2006) il correspond à la durée entre deux paquets successifs et il est utilisé dans ce contexte pour décomposer le flux ; un flux est actif quand il est en dehors du temps de repos, autrement il est inactif. Pour décomposer le flux en sous-flux nous utilisons un deuxième seuil du temps de repos appelé «*idlethreshold*» ou les paquets arrivant dans la même période active constituent un sous-flux, autrement dit, si les temps d'interarrivés d'une série de paquets sont inférieurs au seuil (*idlethreshold*) alors ces paquets appartiennent au même sous-flux, sinon un nouveau sous-flux est instancié suivant l'Algorithme 3.1.

Algorithme 3.1 Algorithme de décomposition de flux

```
Input Idle_Time=1, Timeout=60
Output Sub-flow= []
start
1       n=0
2       t0=t(p(j,0))
        #p(j,n), le nième paquet du flux j
3       Wihle (t(p(j, n+1)-t0)≤ Timeout) do
4       Δt=t(p(j, n+1))-t(p(j, n))
5       if  Δt≤ Idle_Time then
6           if Sub-flow is empty then
7               instanciate subflow(j,len(Sub-flow))
8               add  p(j,0) to subflow(j,len(Sub-flow))
9               add subflow(j,len(Sub-flow)) to Sub-flow
10          else
11              update subflow(j,len(Sub-flow)-1) # add p(j,n+1) to subflow(j,len(Sub-
                flow)-1)
12      else
13          instanciate subflow(j,len(Sub-flow))
14          add p(j,n+1) to subflow(j,len(Sub-flow))
15      endif
16      n=n+1
17      endwihle
19      return Sub-flow
```

Parmi les éléments clés de classification basée sur l'apprentissage automatique, nous retrouvons la préparation ou les prétraitements des données sont nécessaires avant qu'elles puissent être utilisées pour entraîner un classifieur. Dans ce contexte, la classification des applications nécessite des prétraitements des données de traces de trafic brut enregistrées

sous forme de paquets dans des fichiers «pcap», entre autres. La conversion de ces derniers formats vers des formats que les algorithmes d'apprentissage machines implémentés peuvent traiter en général des fichiers de types arff et dans certain cas csv ou TXT est nécessaire. Ce processus de prétraitement est effectué par deux composantes, la première vise le calcul des caractéristiques des flux et la deuxième s'occupe de la sélection des caractéristiques.

L'approche basée sur les flux possède une étape indispensable de processus est le calcul des caractéristiques des flux à partir des traces de trafic. Cette étape enregistre, sous forme de paquets ou bien directement du trafic capturé, sur une interface réseau. Selon la définition du flux de données précisés plus haut, des flux bidirectionnels comportant 44 caractéristiques ont été adoptés dont les caractéristiques ordinaires comme les adresses sources et destinations, les ports et les protocoles et d'autres caractéristiques, entre autres, la distribution de volume de trafic dans les deux directions (*forward*) et (*bachward*), l'interarrivée des paquets, la durée des flux, etc. (voir détails à l'ANNEXE I).

Pratiquement, l'étape précédente se fait systématiquement par une méthode ou un outil de calcul de caractéristiques (Netmate). Ces dernières sont toutes calculées malgré que quelques-unes ne soient pas discriminantes où elles peuvent même avoir un effet négatif sur la différentiation des applications, en plus une dimension de plus augmente certainement le temps d'exécution pour plusieurs algorithmes. C'est pour cette raison que la méthode de sélection de caractéristiques a été envisagée pour réduire la dimension de l'échantillon. Plusieurs méthodes de sélection d'attributs sont développées dans la littérature, tel résumé à la Figure 3.6.

L'analyse des composantes principales est une méthode d'extraction de caractéristiques permettant de réduire le nombre de variables décrivant les observations. Rappelons que cette technique permet de projeter les données d'origine dans un autre espace formé par les composantes qui sont des combinaisons linéaires non corrélées des variables d'origine. Dans notre cas, elles sont les 44 caractéristiques définissant un flux. L'importance de cette transformation réside dans les poids des composantes dépendamment de l'information

qu'elles renferment. En effet, l'ordonnancement des composantes principales est effectué en fonction d'ordre décroissant des variances de façon à ce que la première composante principale ait la plus grande valeur, la deuxième ait la plus grande valeur suivante et ainsi de suite. Ceci facilite le choix du nombre des composantes principales composant le nouvel espace en fonction du taux d'information à préserver.

3.4.3 Sélection de caractéristiques

La sélection d'attributs basée sur la corrélation (CFS) est utilisée dans l'apprentissage machine pour chercher les caractéristiques importantes pour un système de classification de données. Elle regroupe principalement deux types d'algorithmes dont (*weighting algorithm*e) et (*subset search algorithme).* Dans ce mémoire l'évaluation est restreinte à ce dernier avec deux méthodes de recherche, à savoir *Greedy* et Génétique via l'implémentation *Weka CFsubseteEval* (*Correlaion-based subset feature selection*).(Witten, Frank, Hall, et Pal, 2016)

En plus des méthodes de la catégorie filtre qui sont largement utilisées, des méthodes *warpper* ont été aussi évaluées pour chacun des algorithmes d'apprentissage automatique implémentés (C4.5 et SVM) via l'implémentation *Weka ClassifiersubsetEval* (Witten et al., 2016). Cette dernière évalue l'importance des attributs en prenant en considération le classifieur utilisé.

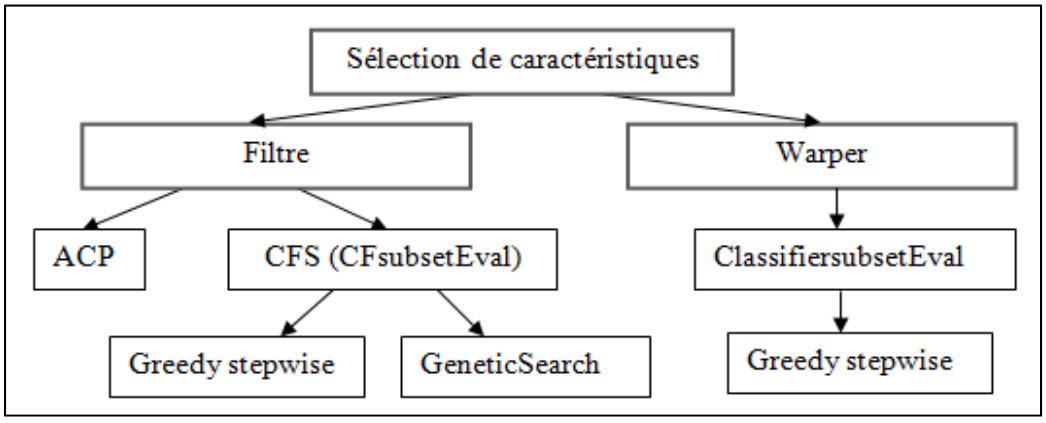

Figure 3.6 Schémas d'évaluation des méthodes de sélection des caractéristiques

Aussi, d'autres informations peuvent être tirées des représentations graphiques des communications de réseau. En effet, les graphes ont pour objet de mettre en évidence la tendance du trafic et de visualiser le maximum d'information possible, entre autres, ils permettent de déduire le type de trafic échangé à travers le réseau. De ce fait, ils supportent la classification du trafic et de suivent son évolution dans le temps. Dans la section suivante nous allons discuter de sur la visualisation des données multidimensionnelles et des techniques de visualisation adéquates pour ce type de données.

3.5 La visualisation des données multidimensionnelles

Sur le plan technique, la visualisation des données se heurte à plusieurs difficultés, dont les plus marquantes sont l'effectif et la dimension de plus en plus élevés des données. Dans le contexte de cette recherche, les données sont multidimensionnelles. De manière générale, une donnée multidimensionnelle est définie comme étant une donnée ayant au moins quatre dimensions. Or, la projection de cette donnée dans des espaces inadéquats ; dans des espaces de haute dimension ou encore dans des espaces réduits, peut être inadaptée pour la compréhension humaine et peut conduire à une dissimulation des informations pertinentes que renferment les jeux de données. C'est pourquoi, et afin de parvenir à restituer toutes les informations pertinentes disponibles dans les données dans des représentations graphiques riches, simples et faciles à analyser, il est primordial de choisir minutieusement les techniques de visualisation propres pour mettre en évidence les informations souhaitées dans les représentations graphiques en fonction des finalités du système de visualisation.

Pour réaliser une visualisation à la fois simple et riche, dans le sens où l'utilisateur peut analyser facilement les graphes tout–en restituant toutes les informations pertinentes, un ensemble de techniques de visualisation en fonction des aspects de supervision des réseaux couverts par le système sont proposées. Le système offre différents niveaux d'analyse complémentaires afin d'assurer une bonne compréhension du réseau ainsi qu'une analyse souple. On distingue trois niveaux illustrés dans la Figure 3.2:

- le premier niveau vise à décrire le trafic d'une manière générale dans le temps en visualisant, entre autres, son volume en termes d'octets et de paquets, la distribution de la taille des paquets, ainsi que la distribution du temps d'interarrivé des paquets. Bien que ces informations soient de base, leurs représentations graphiques dynamiques et simultanées permettent de comprendre la tendance du trafic. Par exemple, en connaissant les caractéristiques des flux éléphants et souris, l'utilisateur peut déduire l'existence possible de ce type de flux en analysant les graphes des volumes de trafic;
- le deuxième dit niveau flux, a pour objectif de traiter et de représenter les données de la couche transport du modèle OSI, notamment les flux échangés sur les réseaux, afin de faire ressortir les informations utiles pour analyser l'évolution du trafic. À ce niveau, deux aspects ont été abordés. Le premier décrit la répartition des flux pour chaque nœud du réseau dans un graphe où l'on peut reconnaître facilement le degré des activités de chaque machine. Ce type de représentation des transmissions réseau permet aussi d'identifier les points d'étranglement ou de surcharge et peut être également utilisé pour prédire les types d'applications qui envoient du trafic entre les machines en analysant l'allure du graphe. L'autre aspect se base sur les mêmes données, soit les informations des flux pour offrir une analyse de ports plutôt que les flux en soient. Ce deuxième aspect est orienté sécurité, étant donné que la majorité des attaques commencent par une écoute réseaux pour déterminer les entrées possibles et fragiles présentant des risques de sécurité, une surveillance des évents réseaux basée sur les ports s'avère être fondamental, importante et indispensable dans un mécanisme de sécurité. En outre, d'autres informations relatives aux flux jugées utiles pour une bonne compréhension du trafic ont fait l'objet de ce niveau. Les métriques telles que la durée de charges et de protocoles des flux sont des caractéristiques pouvant être utilisées pour identifier le trafic, vu que chaque classe d'applications est connue par profil plus ou moins déterminé. Par exemple, un flux large peut être défini comme étant un flux avec une charge et une durée supérieure à certains seuils. Ainsi, les représentations visuelles de ces caractéristiques permettent d'avoir une vue sur la tendance du trafic et d'en tirer des conclusions;
- le niveau application s'intéresse à l'identification et la classification du trafic. Comme il est développé précédemment, un ensemble de méthodes et d'algorithmes sont

implémenté dans le but de favoriser une classification de trafic temps réel, les résultats de cette opération sont par la suite visualisés dans un graphe pour suivre l'évolution du trafic dans le réseau en termes des applications ou des classes d'application et de garder la vue sur le pourcentage de chacune d'entre elles.

La combinaison de ces niveaux d'analyse permet d'interpréter, de comprendre et d'expliquer l'état du réseau et de déduire les causes qui sont derrière son comportement à partir des représentations graphiques simples. La mise au point de ces objectifs passe par le traitement et la visualisation des données de trafic. Ainsi, afin de satisfaire cette contrainte et de visualiser des données multidimensionnelles dans un espace de deux dimensions, plusieurs techniques de visualisation sont proposées pour chaque niveau. En plus du choix des techniques, nous avons utilisé également les métaphores telles que les formes, la taille des icônes et la couleur pour enrichir davantage les graphes. Les techniques de visualisation implémentées sont:

Graphes de séries temporelles (Time series graphes): Ce type de graphe est répandu pour représenter et évaluer les données dans le temps. Il affiche les observations ou les valeurs d'une variable sur l'axe des ordonnées en fonction d'une granularité ou période fixe, par exemple chaque minute. Un «time series graph» peut être adapté pour visualiser les variations de plusieurs variables grâce à l'introduction des lignes multiples ou chaque ligne est associée à une variable. Cette technique est utile pour cette recherche afin de représenter les volumes du trafic ; total, sortant et entrant sur le même graphe pour faciliter la comparaison et l'évaluation de leurs changements au fil du temps.

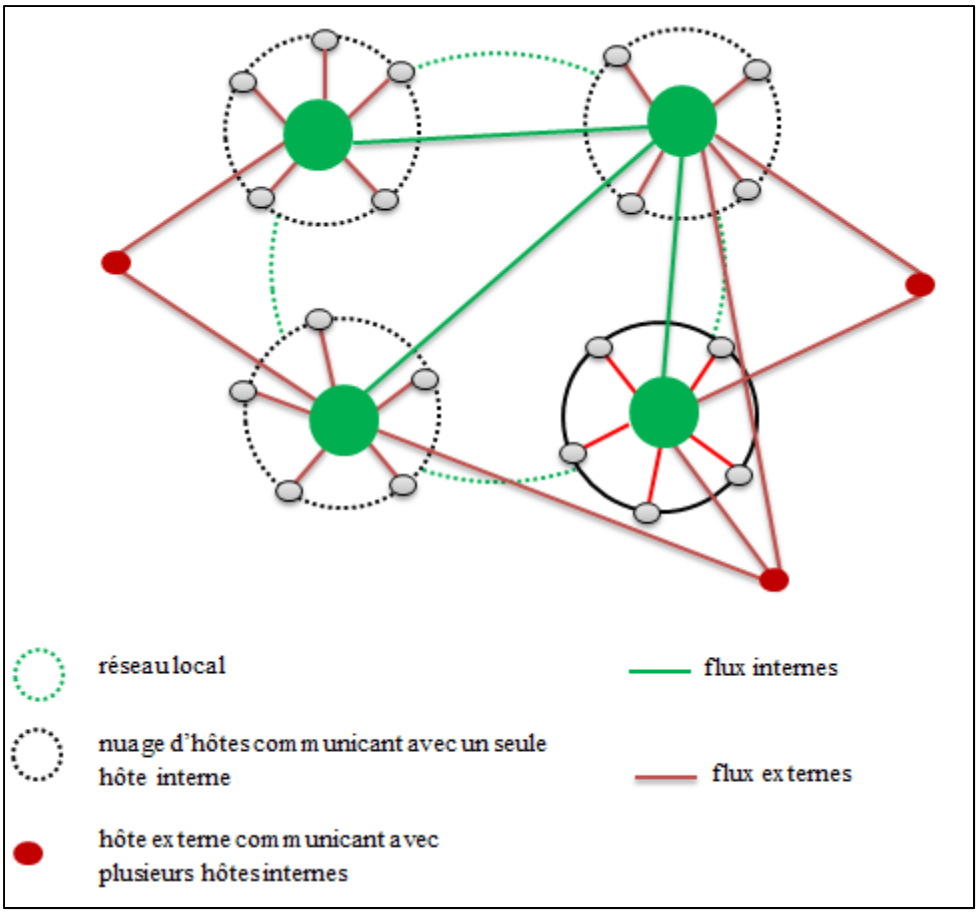

Figure 3.7 Représentations des flux

Graphe de réseau (Network graph): Cette technique de visualisation permet de modéliser les relations et les interactions entre entités dans différents domaines y compris les réseaux informatiques où elle est utilisée afin de représenter les communications réseaux, par exemple pour étudier les comportements des abonnés dans les réseaux sociaux. Pour cette recherche, elle est utilisée afin de visualiser la répartition des flux dans un réseau donné tel qu'illustré à la Figure 3.7. En effet, les flux sont représentés par des lignes qui connectent des entités sous forme de cercles représentant les machines. Les hôtes sont séparés dans deux groupes dits internes et externes. Pour distinguer ces derniers, la taille des métaphores est mise en jeu où de grands cercles correspondent aux hôtes internes tandis que les petits cercles correspondent aux hôtes du groupe externe. Des détails relatifs aux flux peuvent être affichés au besoin pour ne pas encombrer le graphe et rendre sa lecture difficile. Ces détails sont : la

direction, la durée, la taille et le protocole des flux. Ces caractéristiques aident à bien analyser et évaluer les patrons existant dans les données et identifier le trafic en analysant, par exemple, le degré d'activités d'un hôte et la direction des flux. Une telle représentation permet de restituer le maximum d'information dans les représentations graphiques.

Coordonnées parallèles: La visualisation des données par les coordonnées parallèles permet de représenter un point de n dimensions dans un espace de deux dimensions. Chaque dimension est traduite en un axe vertical et chaque point du jeu de données est représenté par un polygone dont les sommets se positionnent sur les axes parallèles suivant les valeurs d'attributs du point. Cette technique est utilisée pour visualiser les activités des ports dans le réseau. Ainsi, la structure de la disposition des axes dans l'espace permet de détecter facilement les corrélations entre les attributs. Cette analyse permet de détecter le trafic malicieux dans les premières étapes de l'attaque.

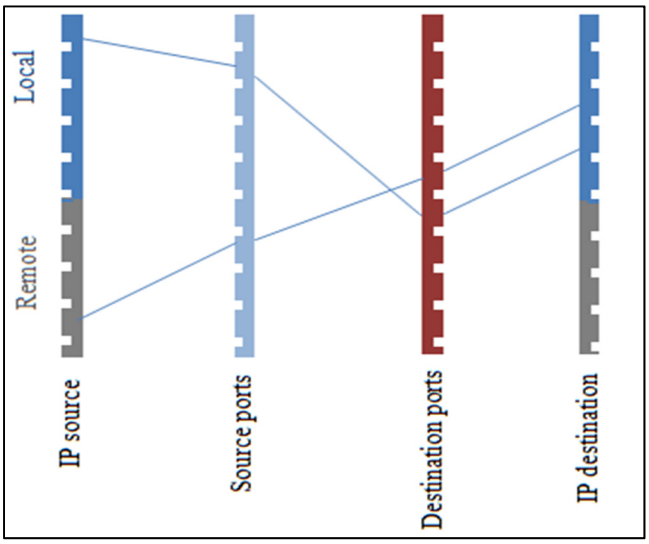

Figure 3.8 Design du graphe d'analyse de ports

Les travaux de recherche antérieurs ont effectué des représentations de la visualisation pour des fins de sécurité des réseaux, et ont proposé des solutions pour surveiller les évènements pouvant menacer le système de sécurités, entre autres, la détection d'intrusion et l'analyse de port comme un processus fondamentale. Cependant, ils ont mis l'accent sur les communications externe, c.-à-d. le trafic échangé entre le réseau interne et un réseau externe

ou l'Internet de manière large sans prendre en considération les menaces parvenant des hôtes internes. Pour pallier à cette limitation, la méthode des coordonnées parallèles est proposée pour pouvoir représenter les deux types de communications soit internes ou externes. Ce nouveau design, illustré à la Figure 3.8, permet de prendre en considération les activités internes et de les rapporter dans le graphe à l'aide des axes doubles représentant deux attributs plutôt qu'un seul, comme il est le cas dans la technique des coordonnées parallèles de base. Quatre attributs sont tenus en compte dans cette analyse ; on retrouve les adresses IP source et destination et les ports sources et les destinations des flux qui sont représentés par quatre axes verticaux ; le premier et le dernier sont destinés aux adresses sources et destinations respectivement, les ports sources sont listés tout au long du deuxième axe et les ports destination sont listés sur le troisième axe. Les axes représentant les adresses source et destination où chacun d'entre eux range les adresses selon deux groupes soient les adresses (sources/destination) locales et adresses (sources/destination) distantes, les éléments du premier groupe sont positionnés dans la partie supérieure de l'axe quant aux éléments du deuxième groupe, ils sont listés tout au long de sa partie inférieure.

D'autres types de graphes ont été implémentés pour faire sortir le maximum d'informations et de rapporter l'état du réseau de façon visuel. Le graphe «*Pie chart*» a aussi été utilisé pour représenter les résultats de classification élaborée lors de l'étape d'analyse. L'objectif de cette représentation graphique est de supporter et d'aider l'utilisateur à comprendre, le mieux possible, le comportement du trafic en termes de type d'application par des graphes au lieu d'analyser des fichiers contenant les classes prédites des flux. «Pie Chart» se compose de deux niveaux : le premier affiche le pourcentage des différentes classes de trafic et le deuxième niveau donne le pourcentage de contribution de chaque application dans la classe à laquelle il appartient.

3.6 La visualisation de trafic en temps réel

La conception des systèmes ou des processus temps réel se heurte à l'abondance des données ainsi que le temps d'exécution et de traitement qui devient de plus en plus importants,

notamment lorsque le nombre de dimensions devient important. Plusieurs approches ont été proposées pour manipuler les données massives en temps réel. La majorité démontre l'utilité des processus parallèles où les parties du système sont exécutés en parallèle, mais d'une manière synchronisée afin d'assurer un échange efficace entre les différents processus. Le parallélisme peut se faire par plusieurs façons, dont les plus répandues dans le domaine de programmation sont : i) le «multithreading» qui est une coopération de plusieurs processus chacun à une tache et ii) le «*multiprocessing»* ou processus parallèles qui se base sur l'utilisation de plusieurs unités centrales dans le même ordinateur ou plusieurs ordinateurs en parallèles.

Dans l'optique de réaliser un cadre de travail permettant de rapporter l'état du réseau en temps réel, un certain nombre d'actions favorisant la réduction du temps de traitement ont été mises au point. Ainsi, trois méthodes ont été combinées, pour parvenir à une visualisation de trafic temps réel, dont la réduction de données, le parallélisme et un traitement immédiat des entrées sans procéder à un stockage préalable. Nous rappelons que la visualisation est définie comme un processus incluant le prétraitement de données, l'analyse de données la transformation en objets géométriques et finalement l'affichage. Ceci signifie que ce n'est pas seulement le temps d'affichage des graphes qui affecte la visualisation, mais les temps que fait à chaque étape du processus et surtout celui de l'analyse. Le cadre de travail peut être amélioré en termes de temps d'exécution en utilisant:

- l'échantillonnage, il est utilisé pour plusieurs raisons, entre autres, l'abondance des trafics ce qui a une influence négative sur le processus de collecte en occasionnant un travail supplémentaire au niveau des nœuds réseau ainsi qu'une consommation de la bande passante lors de la transmission des traces vers les bases de données. En plus, un traitement aveugle de tout le trafic a des effets néfastes sur les processus de temps réels. La nature des systèmes de visualisation en général, impose une série d'opérations dans le but de représenter un graphique d'information pertinente contenue dans les données brutes, ce qui va générer des effets cumulatifs de temps. L'échantillonnage et la réduction de dimension via les méthodes de sélection de caractéristiques permettent alors de réduire la quantité des données et par conséquent le temps de leur traitement;

- le traitement immédiat des données collectées après leur capture sans procéder à un stockage préalable. La phase de sauvegarde des traces de trafic dans des bases de données ou dans des fichiers implique une relecture par le processus de traitement, ce qui peut éventuellement causer des délais importants pour un système temps réel. Ainsi, une lecture instantanée des traces permet d'améliorer le temps de visualisation;
- le multithreading, il est utilisé afin de réduire davantage le temps de visualisation de trafic. Les différentes phases se sont implémentées en tant que taches parallèles (multithread en Java).

3.7 Conclusion

Ce chapitre a décrit les différents modules qui composent la solution de la visualisation de trafic pour la surveillance des réseaux haut débit proposée dans ce travail tout en mettant l'accent les différentes méthodes et approches adopté pour la mettre au point. Pour que le présent cadre de travail satisfasse une certaines propriétés dont principalement l'analyse en temps réel, la simplicité et la richesse des représentations graphiques nous a conduits à utiliser:

- une méthode d'échantillonnage adapté pour réduire la quantité des données a traité, qui se base sur sFlow;
- la classification du trafic en temps réel par le biais d'une approche statistique basée sur les caractéristiques des sous-flux au lieu des flux entiers;
- des techniques de visualisation pour les données multidimensionnelle permettant de mettre en clair les patrons existant dans les données brutes toute en assurant une analyse facile des représentations graphiques. Le chapitre suivant décrit les expérimentations et les résultats de la proposition.

CHAPITRE 4

EXPÉRIMENTATION ET RÉSULTATS

4.1 Introduction

Le présent chapitre présente le protocole expérimental pour évaluer la performance du cadre de travail développé dans cette recherche ainsi que les résultats obtenus des différents cas d'utilisations mis en place en fonction des niveaux d'analyse du trafic présenté au Chapitre 3. La première partie décrit l'environnement de test et les outils utilisés pour mettre au point un banc d'essai. La deuxième partie se concentre sur l'analyse et l'évaluation des fonctionnalités et des performances du cadre de travail mis en place à travers un certain nombre de scénarios. Une étude comparative avec d'autres solutions de la visualisation de trafic a aussi été réalisée.

4.2 Protocole d'expérimentation et banc d'essai

Pour analyser et évaluer les fonctionnalités offertes par le cadre de travail décrit dans ce document, nous avons conçu des scénarios de tests tout en prenant en considération les différents niveaux d'analyse du trafic décrits dans le Chapitre 3.

Pour les deux premiers niveaux d'analyses l'information générale du réseau et l'analyse des flux sont effectuées en deux modes : en ligne et hors ligne. Cela permet aussi bien d'effectuer une analyse du réseau en temps réel qu'une analyse des traces de trafic préalablement collectée. Le troisième niveau qui consiste à la classification du trafic au niveau d'application s'appuie principalement sur des traces de trafic public bien qu'il soit conçu pour une classification quasi temps réel. Les détails de ce processus seront abordés plus loin dans ce chapitre.

4.2.1 Environnements de tests

Les fonctionnalités du cadre du travail destinées aux deux premiers niveaux d'analyse ; les informations générales du réseau et l'analyse des flux sont testées sur un réseau simulé à l'aide de l'émulateur Mininet (Mininet, 2016). Le réseau local est composé principalement de quatre machines et un communicateur virtuel connecté au réseau Internet comme réseau externe. Le trafic véhiculé sur ce réseau est généré de deux façons :

- le trafic interne échangé entre les machines locales et générées par Iperf (Iperf) et Nmap (Nmap). Ce dernier outil est un outil de balayage de port libre et permet aux administrateurs de réseau d'effectuer un audit de sécurité en analysant les résultats qu'il retourne. En effet, il est conçu pour repérer entre autres les ports ouverts sur une machine distante;
- le trafic externe contient un trafic réel (FTP, Web, pin) échangé entre les machines locales et des machines externes.

Pour la collecte du trafic, au lieu de collecter tous les paquets qui passent par chaque interface du nœud réseau en question, seulement des paquets échantillons sont collectés dans l'objectif de réduire l'utilisation des ressources du réseau Cette situation rend possible la mise en échelle de la solution proposée et son adaptation aux réseaux haut débit .Pour cet effet le standard d'échantillonnage sFlow a été comme expliqué dans le Chapitre 3. Tel que présenté à la Figure 4.1, le processus commence par la collecte du trafic au niveau du commutateur qui est envoyé par la suite sous forme de datagrammes sFlow au collecteur *sFlow-RT*. Ce dernier dispose d'un système de collecte qui se met à jour à chaque fois où il y a un nouveau datagramme qui arrive. Par contre, il ne sauvegarde pas toutes les informations reçues, mais il expose uniquement les derniers datagrammes reçus, ce qui permet de réduire en termes de ressources de stockage. La dernière partie du cadre de travail consiste en une application qui requête le collecteur périodiquement pour récupérer l'information brute qui est l'objet d'un traitement spécifique en fonction de l'objectif de l'analyse.

Le volume du trafic véhiculé essentiellement en interne atteint 1Gbits (le volume maximal permet sur un réseau Mininet si possible).

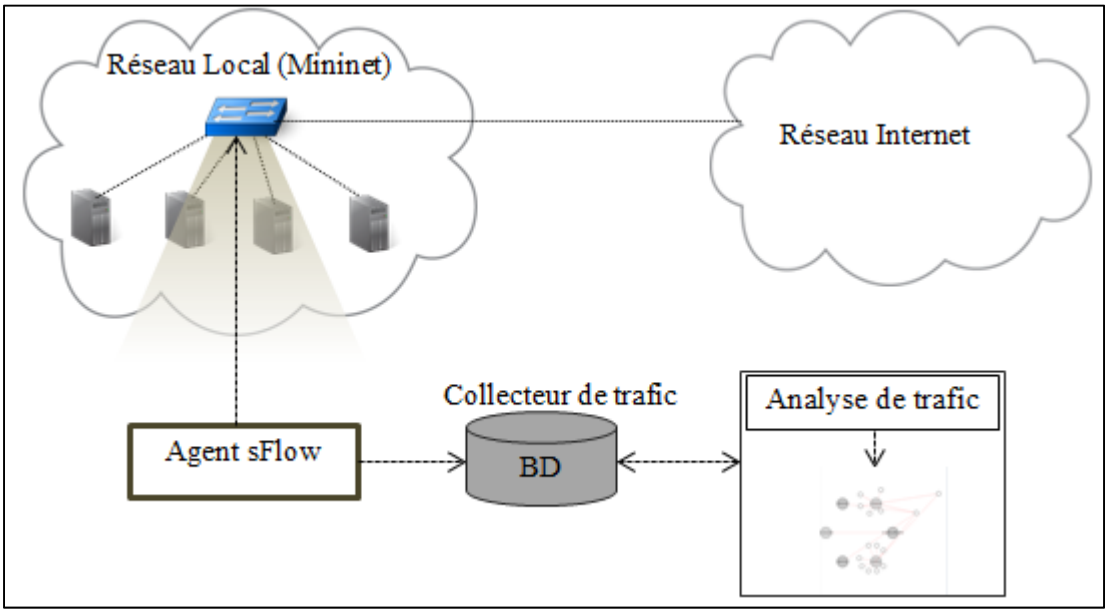

Figure 4.1 Schéma du banc d'essai

4.2.2 Scénarios de tests

4.2.2.1 Scénario 1 : Test de l'analyse de base des réseaux

Ce premier scénario consiste à tester le premier niveau d'analyse offert par la plateforme de visualisation de trafic proposée. Il s'intéresse à faire sortir les tendances générales du trafic l'état du réseau. En effet, la fonctionnalité développée dans ce volet permet de traiter les données de bas niveau, en particulier, le volume de trafic en termes de nombre d'octets et de paquets, la distribution de la taille des paquets et des informations de chaque connexion. Ces derniers sont représentés au cours du temps dans des graphes dits «*time series graphs*». Ces données (voir tableau 4.1) présentent principalement les statistiques d'une interface réseau particulière et elles sont extraites du troisième champ du datagramme sFlow «*Interface Counter*».

Tableau 4.1 Statistiques d'interface réseau

Compteur d'interface	Statistique
If index	L'indice de l'interface
Paramètres physiques de l'interface	La vitesse, le mode et le statut de l'interface
Compteur d'entrée	Nombre d'octets entrant, nombre de paquets entrant.
Compteur de sortie	Nombre d'octets sortant, nombre de paquets sortant.
Statistique Ethernet	Erreur FCS, collision.

4.2.2.2 Scénario 2 : Test de l'analyse de flux

L'analyse de flux traite les données issues de la couche transport du modèle OSI en particulier les flux. Dans ce contexte le flux est défini comme une suite de paquets ayant en commun le 5-tuple (ip_scr, ip_dst, port_src, port_dest, protocole), un flux unidirectionnel. Cette analyse de flux est subdivisée en deux branches d'analyse :

- l'analyse de la répartition des communications réseau internes et externes représentée dans un graphe réseau (Network graph). Ce dernier permet de présenter la tendance du trafic et sa répartition et mettre à disposition un ensemble de fonctionnalité de filtrage et de sélection pour bien comprendre et interpréter l'information exposée. Comme nous allons le voir dans les résultats, cette analyse peut aider à déduire le type de trafic véhiculant sur le réseau;

- l'analyse orientée numéros de ports qui est destinée à détecter une attaque ou d'un trafic malicieux dans les premières étapes en particulier le balayage de port ou de réseau. De la même façon que la première analyse, celle-ci s'intéresse aux flux unidirectionnels du protocole TCP. Les résultats ont été représentés dans un graphe de coordonnée parallèle à quatre axes (voir le Chapitre 3).

4.2.2.3 Scénario 3 : Test de l'analyse d'application

Le test du troisième niveau d'analyse met l'accent sur l'identification et la classification du trafic IP par le biais de deux algorithmes d'apprentissage supervisé, à savoir l'algorithme C4.5 et les SVM. Le mécanisme d'échantillonnage adaptatif proposé dans cette recherche a été testé étant donné qu'il est difficile de l'incorporer dans le standard sFlow utilisé sur les nœuds réseaux. Les modèles de classification ainsi que les tests sont effectués en utilisant des traces de trafic public (University, 2009). Les classes considérées dans ce travail et qui sont résumées au Tableau 4.2 dépendent des données d'apprentissage présentées au (Tableau 4.3).

Tableau 4.2 Classes d'application

Class	Application
P2P	Bittorrent, edonkey
Web	HTTP, https
Mail	imap, pop3, smtp, ssl
Skype	Skype (TCP), skype (UDP)
autre	autre applications (DNS, DHCP…)

Tableau 4.3 Composantes de traces de trafic

	Flux (%)	**Octets (%)**
Web	61.2	12.5
Courriel	5.7	0.2
Bittorrent	9.3	15.9
Edonkey	18.4	70.2
Skype	5.2	1.0
Autre	0.2	0.2
Totale	79000	27 GB

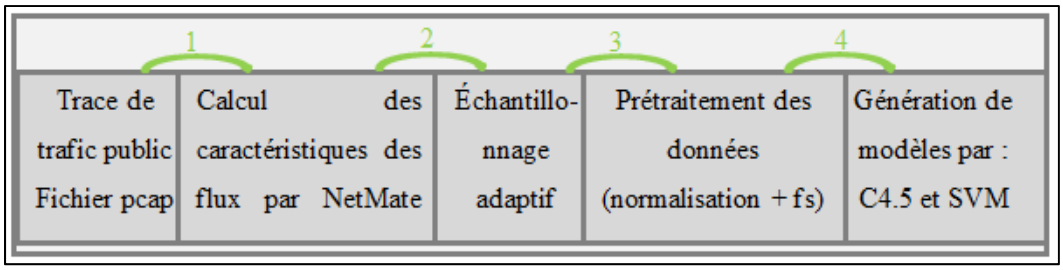

Figure 4.2 Étapes de la classification de trafic

La Figure 4.2 illustre les différentes étapes suivies pour mettre au point les fonctionnalités d'analyse d'application :

- la première consiste à calculer les caractéristiques des sous-flux à partir des traces paquets sauvegardés en format «pcap». Pour ce faire, l'outil normé NetMate (Arndt, 2015) et a été utilisé. Ce dernier génère un ensemble de caractéristiques (voir ANNEXE I) en fonction de sa configuration, en particulier le type de flux, le temps de repos et le timeout (voir Tableau 4.4);
- la deuxième étape consiste à réduire la taille de l'échantillon en appliquant la méthode d'échantillonnage adaptif (voir l'équation 3.1);
- la troisième étape permet de préparer les données pour l'apprentissage machine. Le prétraitement permet de normaliser les données afin d'éviter la dispersion des valeurs des attributs et permet également de réduire les données d'entrainement par le processus de sélection de caractéristiques. Tel qu'il a été décrit au Chapitre 3, trois techniques sont évaluées, en particulier, ACP, sélection pour SVM et Greedy;
- La dernière étape consiste à appliquer les algorithmes d'apprentissage pour générer le modèle de classification. Trois algorithmes sont évalués : C4.5 et RandomForest comme des arbres de décision et les SVM linéaires et non linéaires (RBF). Le SVM linéaire a été testé avec différentes valeurs du paramètre de généralisation (C). SVM non linéaire a été testé avec différentes valeurs du paramètre du noyau Gamma (g). Le temps d'apprentissage des différents modèles de classification a été évalué dans l'objectif de choisir celui qui a le meilleur compromis entre le taux classification correcte et le temps d'apprentissage. Dans phase de test, la méthode de la validation croisée «cross validation» avec k=10 a été adopté pour évaluer les performances de ces modèles.

Nous rappelons que les mesures d'évaluation des performances des classifieurs adoptées dans cette recherche sont:

- le rappel de la classe k est défini comme le rapport entre le nombre d'éléments correctement attribués à la classe k et le nombre d'élément appartenant à la même classe et il est exprimé par (4.1);

$$Rappel_{classek} = \frac{VP}{VP + FN} \tag{4.1}$$

- le taux de précision de la classe k représente le rapport entre les éléments qui sont correctement attribués à la classe k et le nombre totale des éléments attribués à la même classe et il est exprimé par (4.2).

$$\Pr écision_{classek} = \frac{VP}{VP + FP} \tag{4.2}$$

Avec

VP (vrai positif) représente le nombre d'éléments correctement attribué à la classe k.

FP (faux positif) correspond au nombre d'éléments incorrectement attribués à la classe k.

FN (vrai positif) représente le nombre d'éléments appartenant à la classe k incorrectement attribués à une autre classe.

Tableau 4.4 Paramètres de génération des sous-flux

Paramètre	**Valeur**
Type de flux	bidirectionnel
Idle time	1 s
Timeout	60 s

4.2.3 Outils et bibliothèques

Le présent cadre de travail a été développé avec le langage Java (le choix a été effectué pour s'aligner et faciliter l'intégration de la solution dans le projet général WAN-Hypervisor

supporté par Ciena). Pour la mise au point des différents modules qui le constituent, nous avons utilisé un ensemble d'outils et de bibliothèques Java présentés au tableau 4.5.

Tableau 4.5 Outils de mis en place du cadre de travail

Méthode/bibliothèque	**Définition et utilité**
Outils et bibliothèques communes pour tous les modules	
JFreeChart (JFreeChart)	C'est une bibliothèque java open source permettant de créer une large variété de graphes et des diagrammes.
Piccolo2D (Piccolo2D)	C'est une boîte à outils qui permet de programmer des applications graphiques personnalisées tout en offrant une interaction avec l'utilisateur, grâce à ses différents évènements, zoom arrière pour avoir un aperçu global et un avant pour obtenir les informations détaillées et sa structure hiérarchique des objets permettant au développeur d'application d'orienter, regrouper et de manipuler des objets de façons significatives.
Module de classification	
NetMate (Dupay, Sengupta, Wolfson, et Yemini, 1991) et (Arndt, 2015)	NetMate est un outil permettant de convertir les fichiers de capture du trafic réseau sous forme de séquences de paquets en statistiques de flux.
Weka (Weka)	Weka possède une collection d'algorithmes d'apprentissage pour les tâches d'exploration de données. Les algorithmes peuvent être appliqués directement à un ensemble de données ou appelés à partir d'un code Java en utilisant la librairie Weka. Il contient des outils pour le prétraitement, la régression, la classification et le *clustring* de données.

4.3 Résultats

Cette section résume les résultats des tests effectués pour chacune des fonctionnalités implémentées dans le cadre de travail.

4.3.1 Fonctionnalités du premier niveau

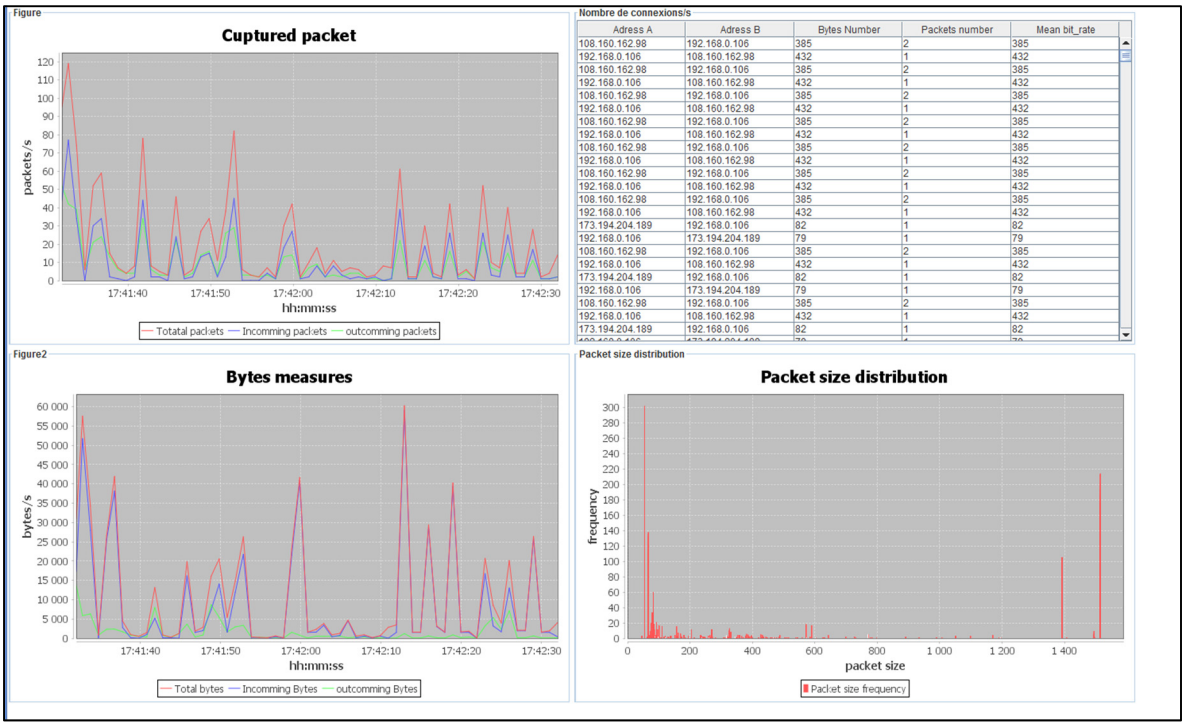

Figure 4.3 Métriques générales du réseau (a) volume de trafic (paquets), (b) volume de trafic en octets, (c) historiques des connexions, (d) distribution de paquets

La Figure 4.3 illustre les mesures et les statistiques crées au premier niveau, dont le volume de trafic sortant et entrant en termes de nombre de paquets et d'octets, la distribution de la taille de paquets ainsi que des informations relatives aux différentes connexions établies (le nombre de paquets, le nombre d'octets et le nombre d'octets moyen). Le graphe en haut à gauche présente le nombre de paquets entrant et sortant ainsi le nombre de paquets échangés à travers une interface réseau particulière contre le temps. De la même façon, le graphe en bas à gauche représente le nombre total d'octets entrant et sortant sur la même interface. La table en haut à droite résume toutes les connexions établies et leurs statistiques (le volume de

donnée échangé) et le dernier graphe illustre la distribution de la taille des paquets, comme on peut le voir les paquets de taille 1500 octets sont fortement présent sur le réseau puisqu'on utilise Ethernet. Bien que la simplicité de ces statistiques, elles offrent une visibilité de base assez importante du réseau. Une connexion est définie à ce niveau comme étant l'ensemble de paquets ayant la même source et destination indépendamment des autres paramètres (ports, protocole, etc.).

4.3.2 Fonctionnalités au niveau transport

La Figure 4.4 montre la vue principale de la répartition de flux internes et externes. Cette vue est dotée d'un ensemble de filtres permettant d'explorer le graphe ou l'utilisateur se voit présenté les flux en fonction du filtre ou des filtres actif(s).

- zone1 (Flow graph) : cette zone permet d'afficher les différentes machines actives sur le réseau et les flux échangés entre elles, elle permet aussi aux utilisateurs d'interagir avec le graphe grâce aux évènements offerts par Piccolo2D, ce qui met en évidence les informations relatives à chaque flux sélectionné (IP source, IP destination, Port source, port destination, protocole, nombre de paquets et la charge) et de colorer les flux d'une machine par un simple clic. Elle permet aussi un zoom arrière pour voir la totalité du graphe et un zoom avant de se focaliser sur une partie du graphe. Ce graphe différencie entre le réseau local et le réseau distant (Internet). Ainsi, de cette manière l'utilisateur peut diagnostiquer le réseau plus rapidement dans le sens où il peut voir les chevauchements et les machines extérieures qui communiquent le plus avec les machines internes;

- zone2 (filtres) : cette zone donne à l'utilisateur l'accès à un ensemble de filtres qui favorisent la compréhension du graph en mettant en évidence les flux ayant les caractéristiques choisies par le biais de ces filtres basés sur les adresses IP, le Type de protocole et la taille des flux;

- zone3 (flow details): elle sert à afficher tous les flux dans le réseau (adresse IP source, Port source, Adresse IP destination, Port destination, Port destination et le protocole);

- zone4 (Information on demand): cette zone permet d'afficher les informations relatives à un flux représenté par une simple ligne (ligne sur la zone1) dont l'adresse source et destination (direction du flux), le port source et destination, le protocole et la taille du flux.

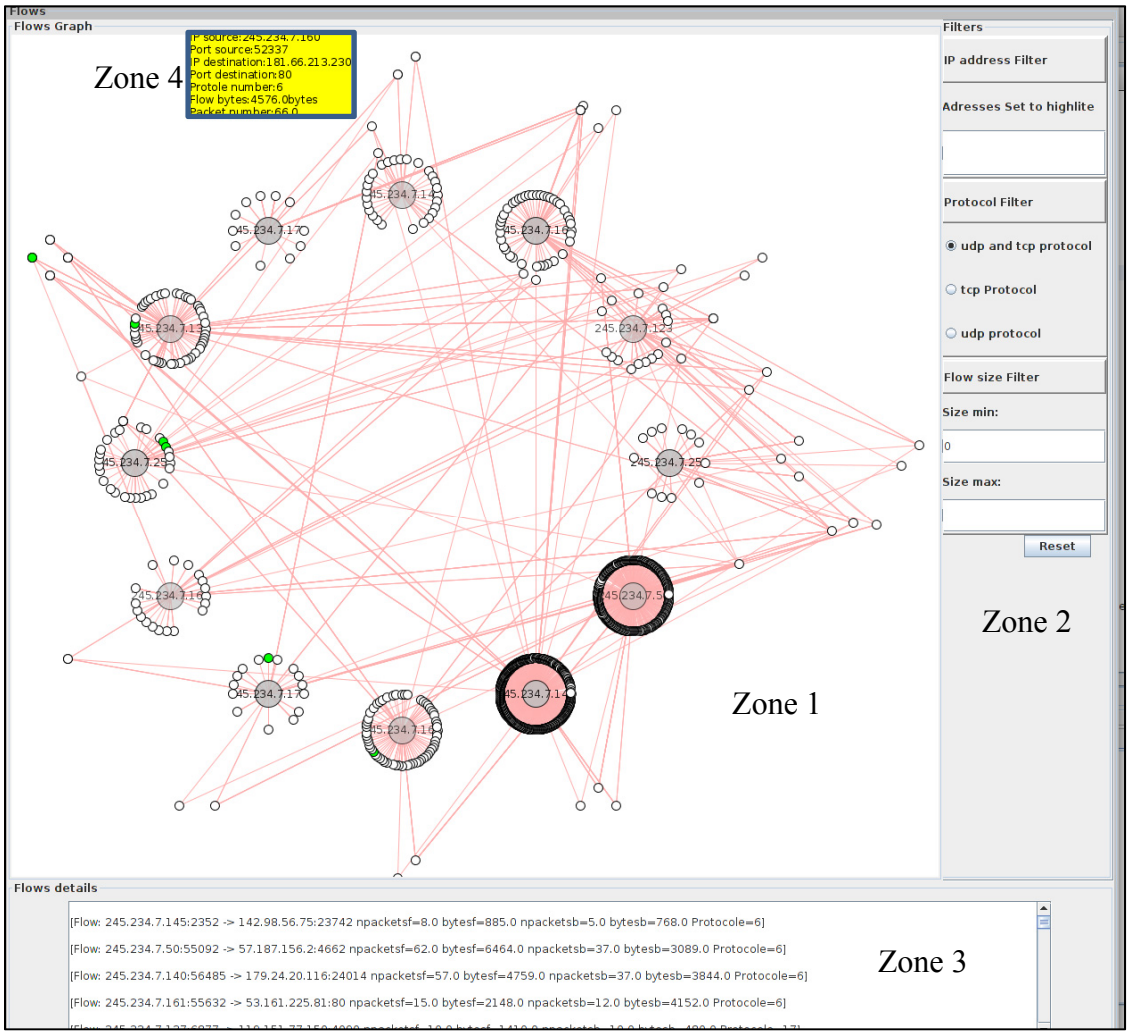

Figure 4.4 Vue principale de la répartition des flux

L'utilisateur peut se servir des différents filtres disponibles pour interagir avec le graphe. Pratiquement, il peut mettre en évidence les détails d'un flux particulier étant donné que le graphe montre seulement qu'il a une communication ou plusieurs entre deux hôtes sans surchargé le graphe par des détails.

Les filtres implémentés et les autres capabilités telles le «zoom» et la sélection offre la possibilité de mieux analyser les graphes et éviter les difficultés conjuguées aux graphes surchargés. Dans le graphe principal (Figure 4.4), la zone 1 offre une vue générale de la répartition des flux ce qui rapporte l'état du réseau en termes de nombre de flux échangé et leur type (externe ou interne) ainsi que l'état des machines locales (les serveurs surchargés, ou éventuels cibles du trafic malicieux). Pour cette raison, les capabilités de navigation sont utilisées pour une analyse plus élaborée et précisée et l'utilisateur peut se servir de ces différents filtres disponibles pour interagir avec le graphe. Généralement, il peut choisir d'explorer le graphe suivant un ou plusieurs filtres. Sur le même graphe, on comprend que la machine sélectionnée échange un nombre assez important de flux avec des machines externes, et analysant les caractéristiques de ces différents flux on comprend qu'il s'agit d'un serveur Web.

La Figure 4.5 illustre des exemples d'utilisation des filtres ou les flux UDP (l'image (a)) et TCP (l'image (b)), mis en évidence avec différentes couleurs. De la même façon, l'utilisateur peut visualiser clairement plusieurs adresses et aussi les tailles des flux. Ce dernier filtre peut être utilisé pour pouvoir anticiper l'existence des flux gourmands en bande passante.

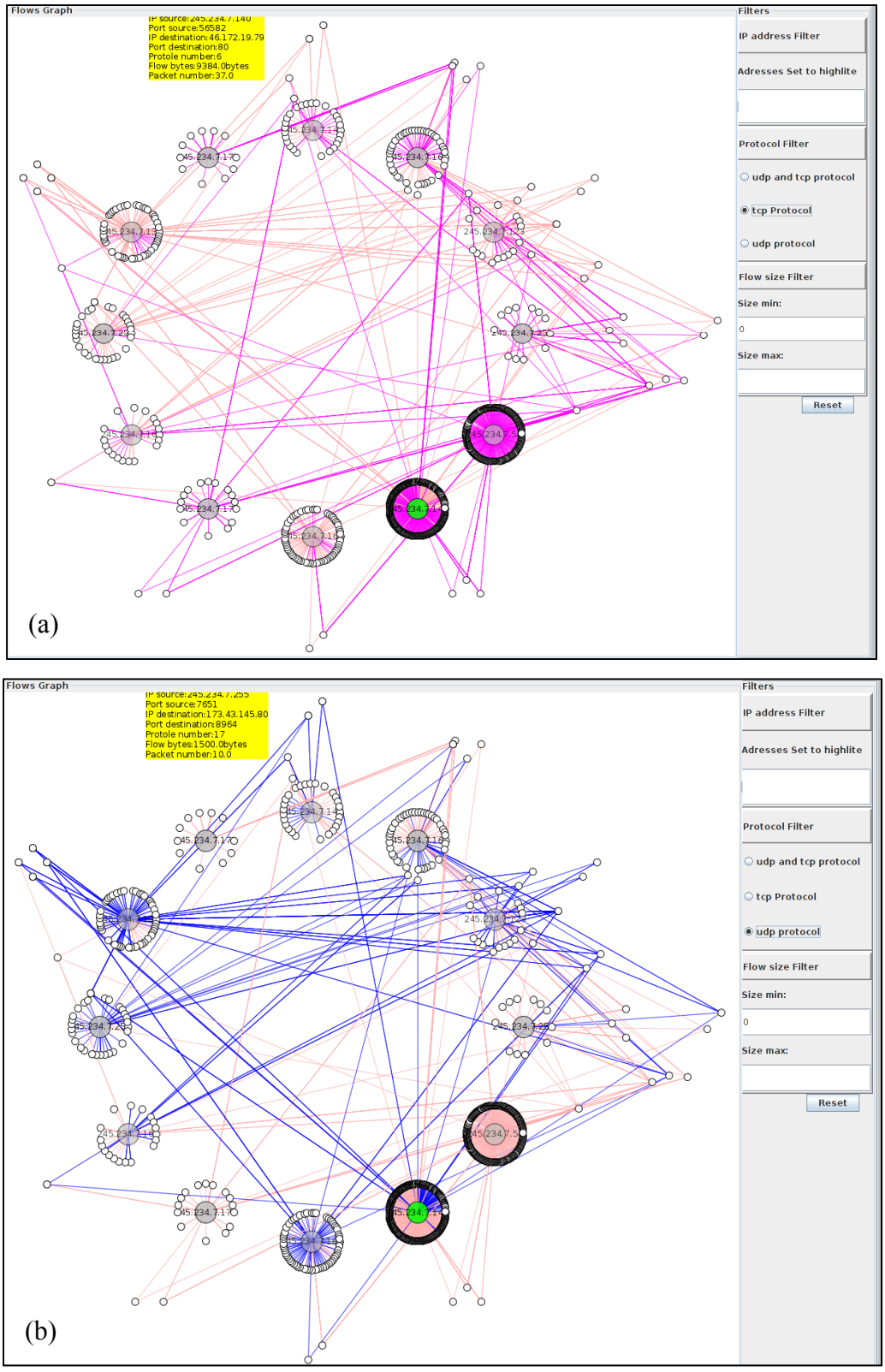

Figure 4.5 Filtrage des flux UDP (a) et filtrage des flux TCP (b)

Pour assister davantage l'utilisateur, une analyse de flux est conçue pour surveiller et détecter les flux malicieux principalement le balayage de port et de réseau. Cette partie s'intéresse aux flux TCP et renvois une représentation graphique de ces flux à la fois internes et externes. La Figure 4.6 montre une attaque ou un balayage de ports réalisé par (Nmap) qui vient d'une machine interne. La technique de visualisation (quatre axes) permet de repérer et reconnaître facilement le trafic anormal. Comme l'illustre la partie (a) de la figure, la machine dont l'adresse IP est «10.0.0.254» effectue un balayage composé des ports du réseau interne, ce qui peut être visible sur le graph à partir du nombre de requêtes et leurs durées envoyées de cette machine « attaquante » vers les machines cibles et la manière dont elles sont reparties par rapport aux numéros de port ou plusieurs ports sur toutes les machines interrogés pour repérer ceux qui sont ouverts. Pour prévenir des graphes chargés et difficiles à explorer, un sous-graphe est prévu pour afficher les activités réseau d'une machine particulière ou la machine susceptible d'être la source du trafic malicieux. Un exemple de sous-graphe est illustré dans la partie (b) de la Figure 4.6 qui visualise les requêtes TCP envoyées par la machine «10.0.2.254» vers les autres machines du même réseau.

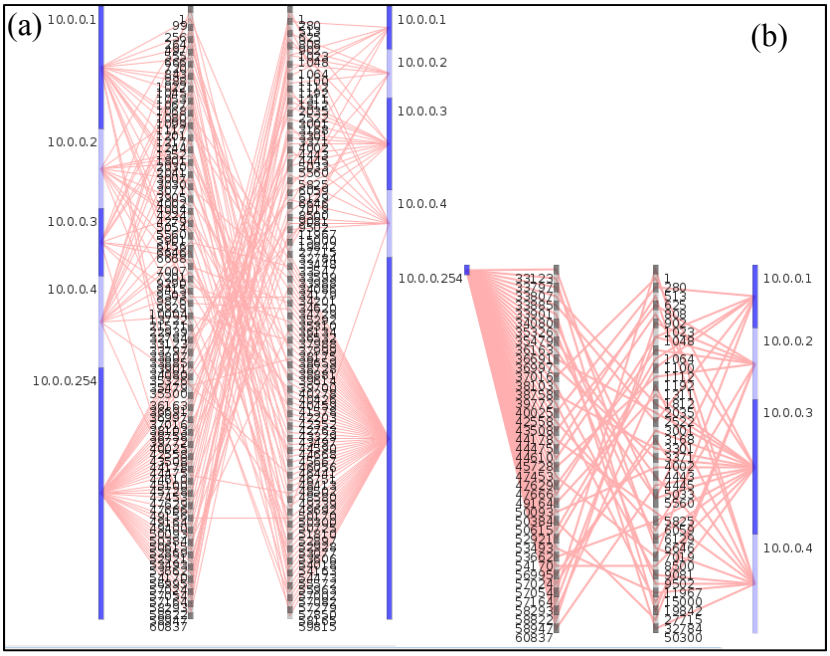

Figure 4.6 Détection de balayage de port (a), sous-graphes des activités de la machine source du flux malicieux (10.0.0 254) (b)

4.3.3 Fonctionnalités au niveau application

Tel que présenté plus haut, la méthode d'échantillonnage adaptif proposée dans cette recherche est testé à ce niveau d'analyse du moment qu'il était très difficile de l'incorporer dans le code de sFlow et la tester sur un commutateur. En effet, elle est utilisée pour échantillonner la trace du trafic avant la génération des caractéristiques des flux. L'utilisation de NetMate avec les spécifications présentées au Tableau 4.4 a généré un fichier de 330000 sous-flux de 44 caractéristiques ce qui était assez coûteux dans la phase de la génération des modèles de classification en termes de temps d'apprentissage pour les classifieurs SVM. La Figure 4.7 montre le taux d'échantillonnage qui est inversement proportionnel au débit à l'instant (t), ce qui permet d'assurer la collecte de la quantité nécessaire de trafic pour l'analyse et la surveillance réseau. Notons que cette méthode permet d'adapter le taux d'échantillonnage en fonction du débit, ce dernier est calculé à l'aide de l'outil Captcp (Captcp) à partir d'un fichier de trafic en format «pcap».

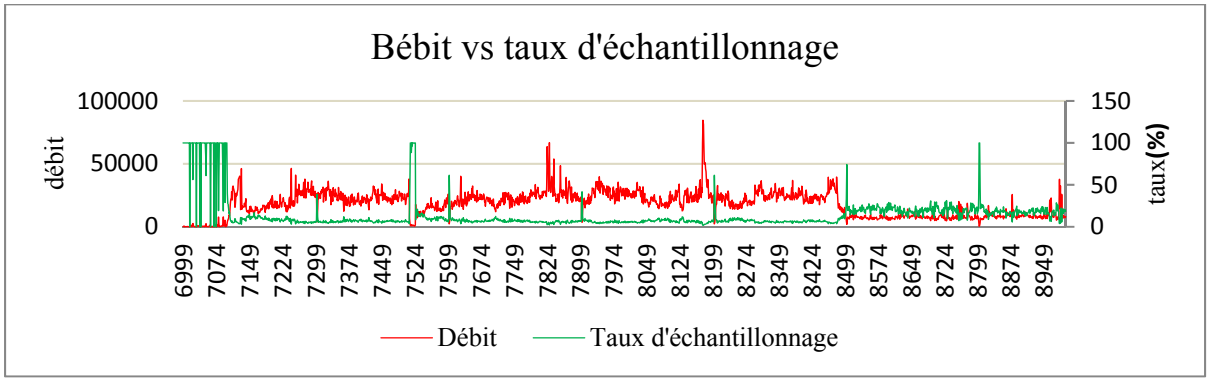

Figure 4.7 Taux d'échantillonnage en fonction du débit

Classification par l'arbre de décision

Le tableau 4.6 illustre le taux de classification et le temps d'apprentissage de chacun des algorithmes d'arbre de décision (C4.5 et RandomForest). Pour les traces de trafic utilisées dans cette recherche cette technique de classification supervisée permet d'achever des taux de classification correcte avec un coût relativement faible en termes de temps

d'apprentissage. Les deux algorithmes obtiennent des taux de classification presque égales; 99,79% pour RandomForest et 99,67% par C4.5. Par contre ce dernier est considérablement plus rapide: il ne nécessite que le 1/3 (5 min) du temps d'apprentissage (17 min) du deuxième pour générer le modèle de classification.

Tableau 4.6 Taux de classification achevé par C4.5 et RandomForest

Algorithmes	**Taux de classification (%)**	**Temps d'apprentissage (min)**
C4.5	99,67	5
RandomForest	99,79	17

Les Figures 4.8 et 4.9 illustrent les taux de rappel et de précision par application obtenus par chacun des deux algorithmes. Bien que les taux sont assez semblables, le C4.5 est plus performant comparativement au RandomForest pour toute les classes. En contrepartie il est moins précis. D'une manière individuelle, chacun de ces algorithmes permet d'atteindre des taux de rappel et de précision élevés. Le RandomForest achève un taux de rappel minimal de 87,5% pour «FTP» et le C4.5 atteint un taux de précision minimal de 89,7% pour «smtp».

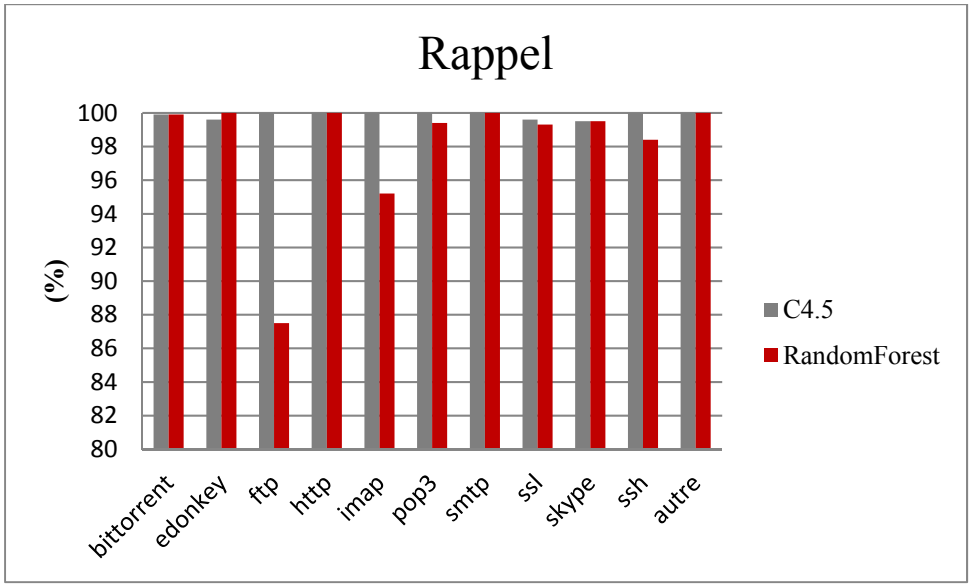

Figure 4.8 Comparaison de taux de rappel C4.5 et RandomForest

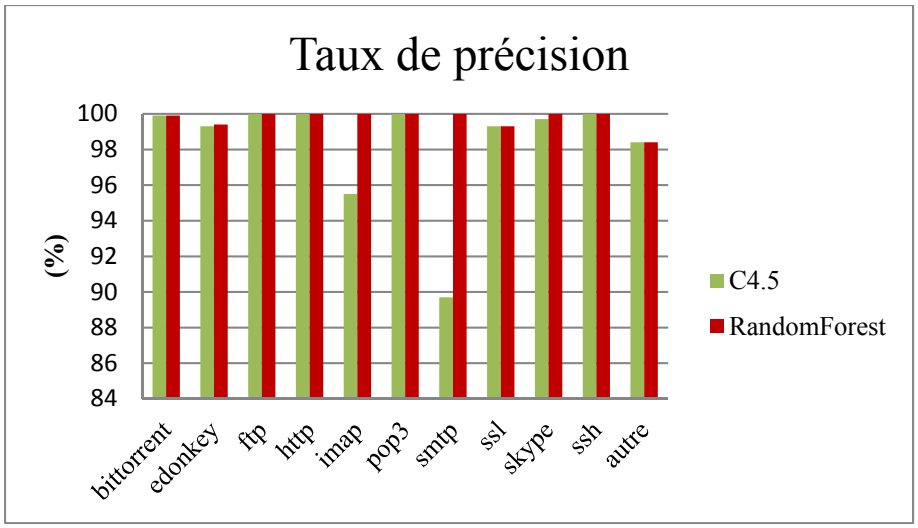

Figure 4.9 Comparaison de taux de précision C4.5 et RandomForest

Les tableaux 4.6 et 4.7 décrivent que l'utilisation d'un «*warpper*» pour la sélection des caractéristiques permet d'améliorer autant le taux de classification que le temps d'apprentissage pour les deux algorithmes même en comparaison avec l'utilisation de tous les attributs. La méthode de sélection des caractéristiques par un filtre (Greedy) a permis de réduire considérablement le temps d'apprentissage des deux algorithmes (1/5 du temps original pour C4.5 et 1/3 pour randomForest) tout obtenant un taux de classification supérieur à 99%. Toutefois, la méthode d'extraction de caractéristiques (ACP à 95%) a permis de réduire le temps d'apprentissage pour C4.5 mais elle a tendance à augmenter celui de RandomForest. Selon ces résultats il est possible de déduire que l'utilisation d'un «warpper» permet de sélectionner les attributs appropriés pour la classification et d'éviter la dégradation qui peut être engendrée par des attributs causant du bruit.

Tableau 4.7 Taux de classification de C4.5 et RandomForest en fonction de la méthode SF

Algorithme / **Méthode SF**	**C4.5**		**RandomForest**	
	TC(%)	**TA (s)**	**TC(%)**	**TA (s)**
Greedy (filtre)	99,20	60	99,40	300
Greedy (warpper C4.5)	99,88	25	99,94	176
ACP	98,64	180	99,31	1200

Classification par SVM (linaire et RBF)

Pour les classifieurs SVM (linaire et non linaire avec le noyau rbf) testés dans cette partie, le tableau 4.8 décrit que le classifieur non linaire est plus adapté (99,25%) aux données de trafic utilisées lors de cette recherche en comparaison avec le classifieur linaire qui atteint un taux de classification d'environ 93%. Le taux de classification d'un SVM non linéaire dépend fortement des paramètres C et Gamma. Dans ce contexte le classifieur avec C=10000 et gamma=1 atteint un taux de classification de 99,25%. Cependant ces classifieurs nécessitent des ressources machines et un temps d'apprentissage important qui peuvent augmenter leur coût d'utilisation.

Le classifieur offrant le meilleure taux de classification, à savoir classifieur rbf (gamma=1 et C=10000) a été entrainé de nouveau en utilisant des données réduites. Le tableau 4.9 précise que l'utilisation des techniques de réduction de dimension ne permet pas de réduire significativement le temps d'apprentissage dans ce cas.

Tableau 4.8 Grille d'évaluation de SVM

Classifieur / **C**	**Linéaire**	**RBF**		
		Gamma=0.0	**Gamma=0.1**	**Gamma=1**
	Taux de classification (%)	**Taux de classification (%)**	**Taux de classification (%)**	**Taux de classification (%)**
1	92,36	91,71	94,00	96,28
10	92,52	94,12	95,20	97,63
100	92,84	95,13	96,00	98,78
1000	92,87	96,18	97,25	99,00
10000	94	96,97	98,36	99,25

Tableau 4.9 Taux de classification de SVM en fonction de la méthode SF

Méthode SF	**Taux de classification(%)**	**Temps d'apprentissage(h)**
Greedy	95.74	3
ACP	95.40	10
SelectionSVM	87.55	6

Comparaison de SVM et les arbres de décision

Les résultats présentés dans les tableaux 4.6, 4.7, 4.8 et 4.9 démontrent que les différents algorithmes testés convergent vers des taux de classification élevés. C4.5 permet d'achever un taux de classification de 99,88%. De la même façon RandonForest et SVM atteignent respectivement 99,94 % et 99,25 %. Cependant le classifieur est couteux en comparaison avec les deux autres algorithmes et nécessite des ressources importante ce qui explique son taux d'apprentissage élevé. Un classifieur basé sur les arbres de décision peut être entrainés dans un temps assez court, par exemple C4.5 ne nécessite que 25s comme temps d'apprentissage (Tableau 4.7).

La Figure 4.10 présente la comparaison des taux de précision des trois classifieurs. Elle décrit que C4.5 est plus précis pour toutes les applications en comparaison avec RandomForest et SVM qui sont moins précis particulièrement pour les données de FTP. Pour le taux de rappel et contrairement à SVM C4.5 et RandomForest obtiennent des taux assez élevé pour toutes les applications (Figure 4.11). Pour ces raisons, le classifieur basé sur C4.5 a été choisi pour être intégré dans le cadre de travail.

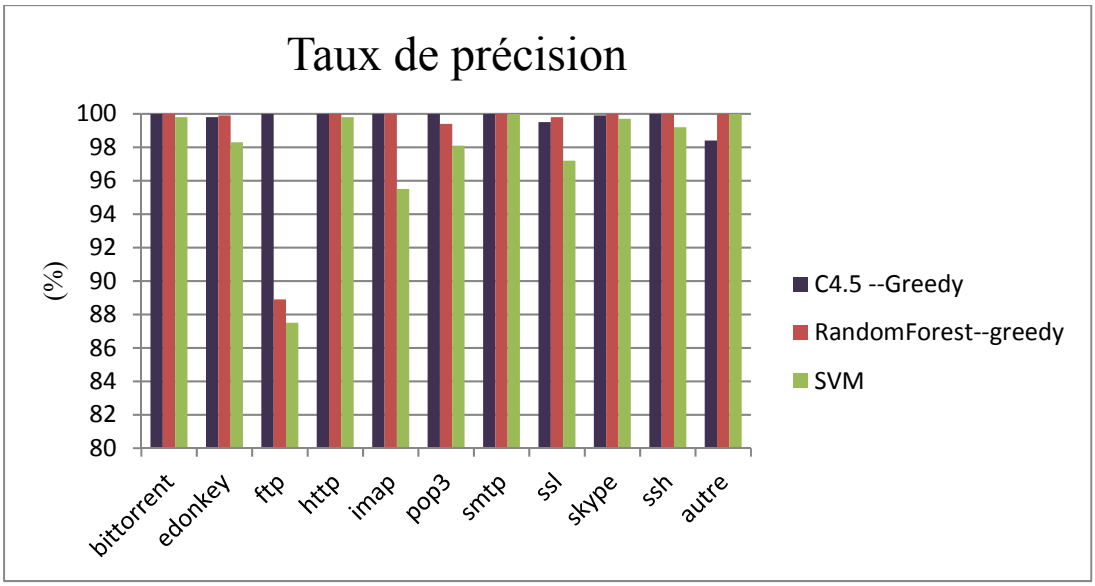

Figure 4.10 Comparaison des taux de précision de C4.5, RandomForest et SVM

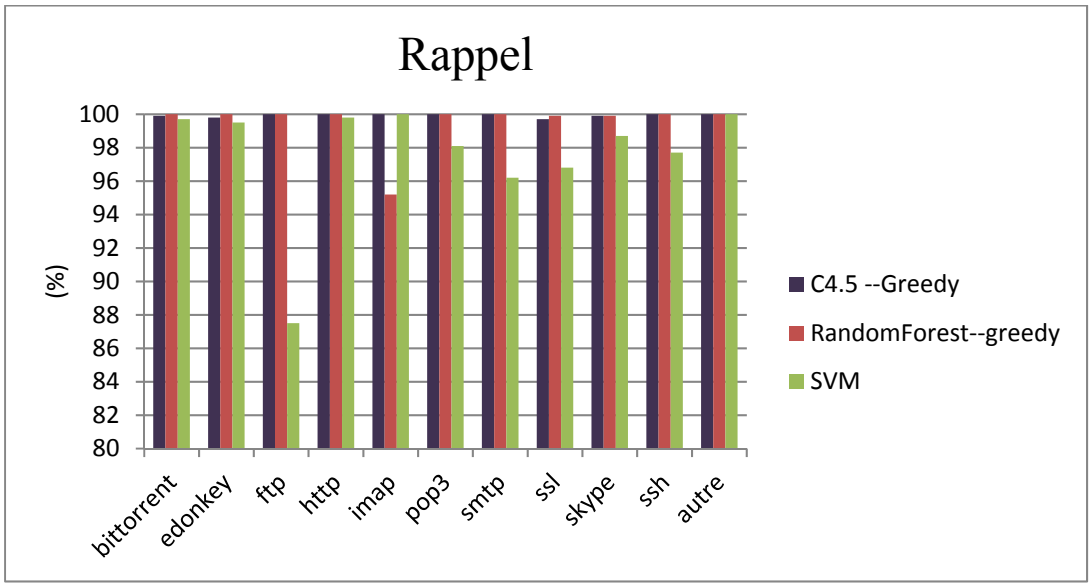

Figure 4.11 Comparaison des taux de rappel de C4.5, RandomForest et SVM

Intégration dans le cadre de travail

Pour faciliter l'analyse du trafic au niveau application les résultats de la classification sont rapportés dans un graphe. Ce dernier est un *Pie Chart* à deux niveaux permettant à

l'utilisateur de distinguer les applications appartenant à une classe particulière. Le premier niveau affiche les portions des classes (P2P, Web, Mail, Skype et Autre). Le deuxième niveau permet de visualiser les portions des applications de chaque classe (Figure 4.12). Le graphe est mené de capabilités de navigation qui permettent de visualiser le pourcentage de chaque application dans chaque classe.

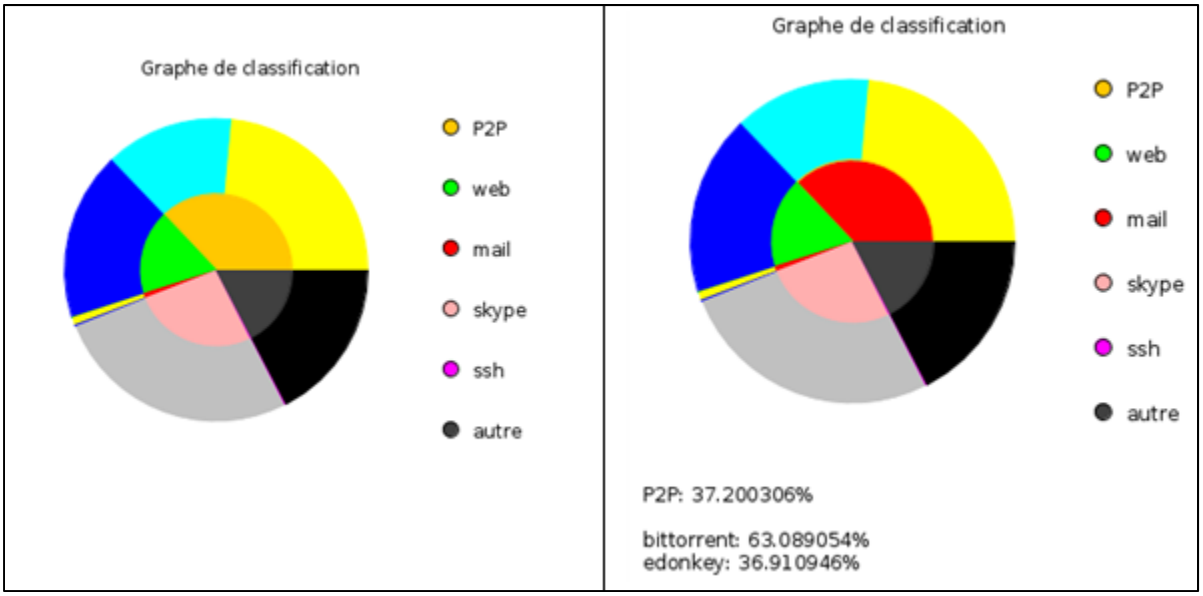

Figure 4.12 Visualisation de résultats de classification de trafic

4.4 Analyse de performances et évaluation

4.4.1 Analyse de CPU et mémoire

L'analyse de performance du cadre de travail fait référence à son comportement d'utilisation des ressources en particulier le CPU et la quantité de mémoire. Les métriques de performance avec une autre application de visualisation de trafic open source (TNV) sont comparées. Cette dernière application utilise sur la bibliothèque Java Jpcap (Jpcap) pour collecter tout le trafic circulant sur une interface réseau particulière. Ces données sont ensuite sauvegardées sous le format brut (paquet) dans une base de données MySQL. Après avoir capturé le trafic, une application permet d'interroger la base de données pour les récupérer et les visualiser.

Pour réaliser la comparaison de performance quelques expérimentations de TNV ont été reproduites en utilisant son code source. Pour ce faire, tshark (Tshark) a été utilisé afin de collecter le trafic qui est traité par les autres classes du code TNV. La lecture et le décodage des paquets sont effectués par Jpcap. Concurremment l'application est utilisée tel que décrit précédemment. La collecte des échantillons du trafic est effectuée à l'aide sFlow et sFlow-RT. Pour évaluer les ressources allouées à chacune des applications, nous avons utilisé la fonction de surveillance des applications intégrales de Netbeans (NetBeans), et la commande top (Top) pour surveiller les taux de consommation du CPU et de la mémoire.

Les Figures 4.13 et 4.14 présentent, respectivement les taux de consommation de la mémoire et de CPU des deux applications. L'utilisation d'échantillonnage peut permettre de réduire le taux de mémoire consommé d'environ 2/3 en comparaison avec l'application qui traite tout le trafic (Figure 4.13). D'une manière similaire, le taux d'utilisation du CPU de la première approche (avec échantillonnage) est assez faible et ne présente qu'à peu près 14 % du taux d'utilisation. Dans le second cas (sans échantillonnage) il peut atteindre 15 % (Figure 4.14). L'utilisation des ressources (pour la mémoire et le CPU) varie significativement entre ces deux cas, où le comportement du premier est stable contrairement au deuxième qui connaît des variations des utilisations assez importantes au cours du temps. Ceci peut être expliqué par la quantité des données traitées par chaque application. Étant donné que ces deux tests sont effectués simultanément et sur le même réseau, l'application ne manipule que 1 % de la totalité du trafic. La Figure 4.15 illustre l'utilisation des ressources par différentes méthodes. En effet, la méthode permettant de décoder les paquets consomme plus de CPU du fait qu'elle traite plus de données dans le cas sans échantillonnage par rapport au cas avec échantillonnage.

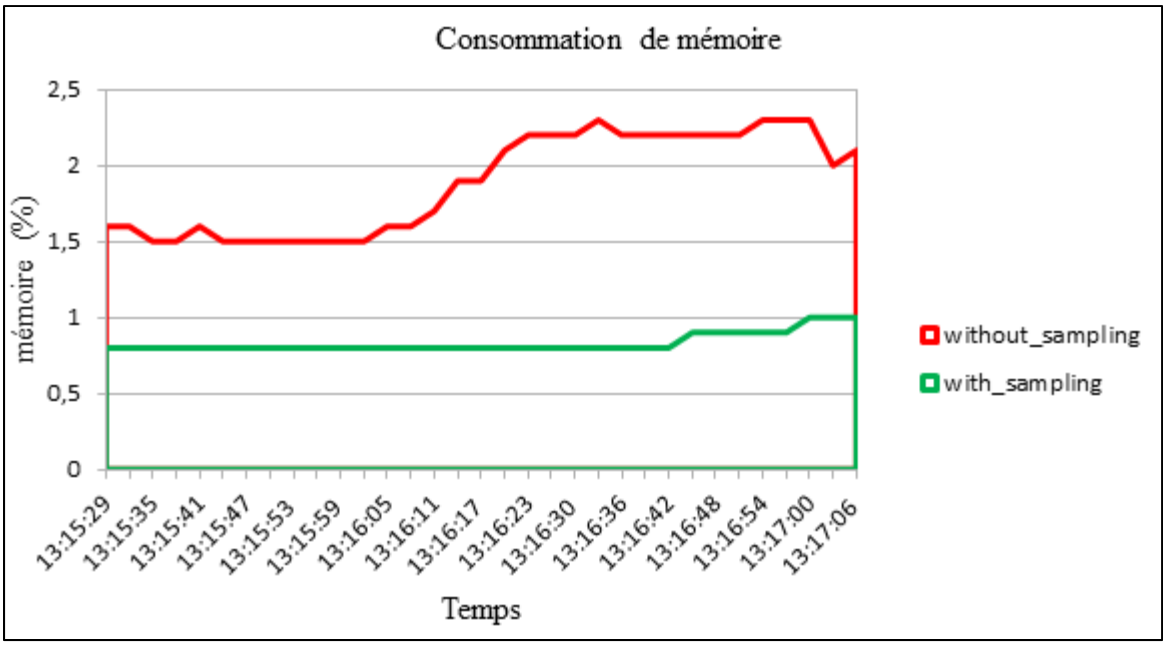

Figure 4.13 Comparaisons des taux d'utilisation de mémoire

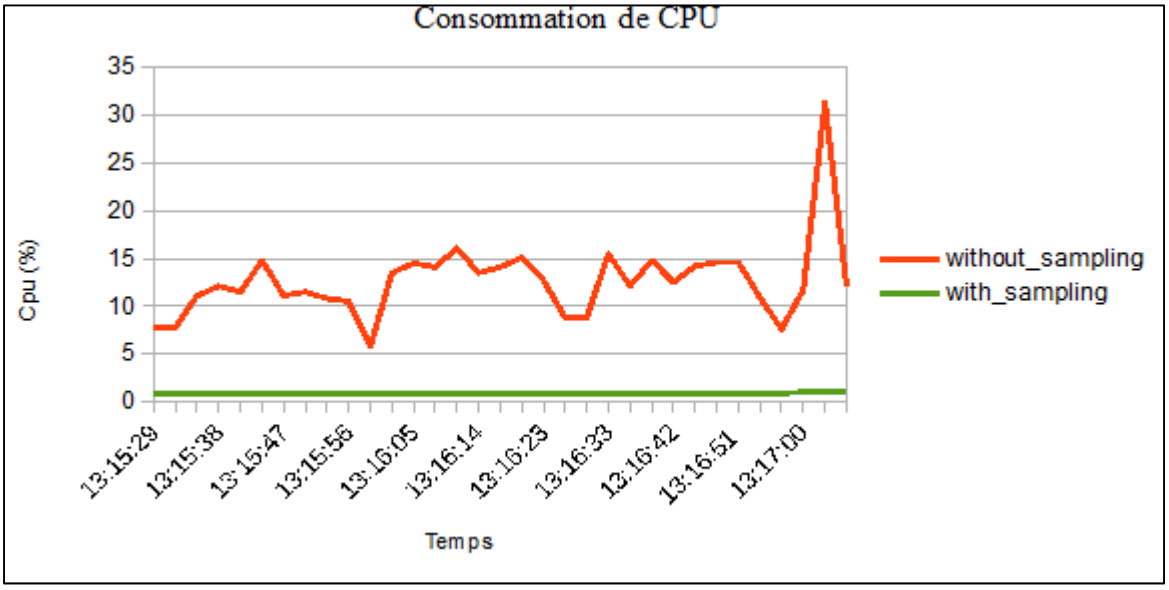

Figure 4.14 Comparaisons des taux d'utilisation de CPU

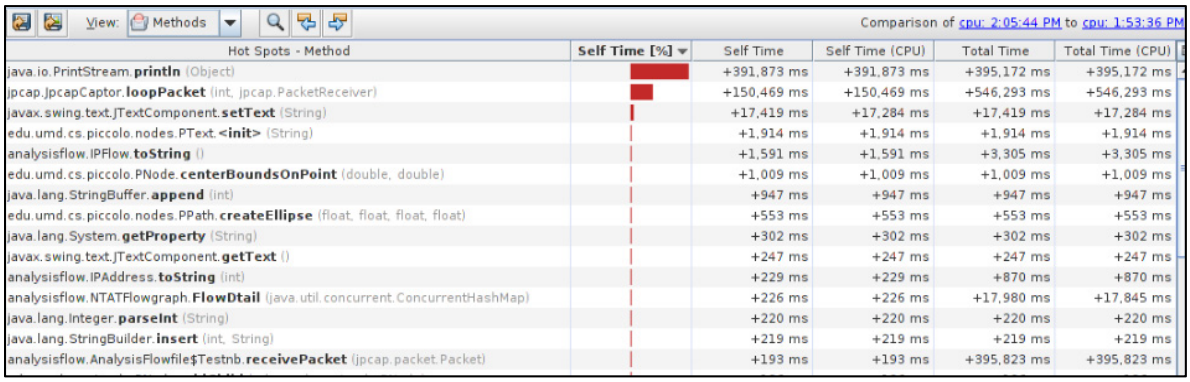

Figure 4.15 Utilisations de ressource par les méthodes des deux applications (avec échantillonnage et sans échantillonnage)

4.4.2 Comparaison de fonctionnalités

Cette section présente une comparaison de fonctionnalités et de la capabilité offerte par le présente plateforme de visualisation par rapport à d'autres systèmes de visualisation de trafic, particulièrement : RTA, TNV et NetMark dont le bilan est résumé au le tableau 4.10.

Tableau 4.10 Comparaison des fonctionnalités de la présente solution, TNV et NetMark

Solution / **Capabilité**	Notre solution	RTA	TNV	NetMark
Sécurité	Partiel	Oui	Oui	Partiel
Flux externe	Oui	Oui	Oui	Oui
Flux interne	Oui	Non	Non	-- —
réseaux	Oui	Non	Oui	Oui
machine	Oui	Oui	Oui	Non
classification	Oui	Non	Non	Non
Offline	Oui	Oui	Oui	Oui
Online	Oui	Non	Non	Non
zoom	Oui	Non	Oui	-- —

Solution / Capabilité	Notre solution	RTA	TNV	NetMark
sélection	Oui	Oui	Oui	-- —
Détail	Oui	Oui	Oui	-- —
Filtrage	Oui	-- —	Oui	-- —

4.5 Conclusion

ce chapitre a présenté l'environnement de test déployé pour évaluer la plateforme de visualisation de trafic proposée dans cette recherche. Les données de deux premiers modules sont collectées directement par le standard sFlow du réseau surveillé. Pour le troisième module, nous avons utilisé la trace de trafic public pour générer et tester les modèles de classification. Une étude de performance a été effectuée à l'aide d'une application qui traite la totalité du trafic et celle proposée dans cette recherche qui utilise l'échantillonnage pour en réduire le coût.

L'analyse des résultats des cas d'utilisation démontre que la solution de visualisation proposée de trafic permet de surveiller le réseau en temps réel sur plusieurs niveaux de représentations et ce d'une manière complémentaire. Elle permet de présenter l'état du réseau et le type de trafic dans des graphes simple à analyser, à interpréter et en tirer des conclusions constructives et qui l'aident à apporter des actions correctives et aussi préventives en termes de QoS et de sécurité.

De la même façon, l'analyse des performances, en particulier le taux de CPU et de la mémoire nécessaire pour le traitement des données et la génération des représentations graphique, démontre que la consommation des ressources de cette proposition est assez faible et stable par rapport à l'application type de comparaison qui est assez élevé et fluctue rapidement après le lancement de l'application.

CONCLUSION

De nos jours, la surveillance et la gestion efficace des réseaux font face à plusieurs obstacles dus d'un côté, à l'évolution rapide des réseaux informatiques qui est accompagnée d'une croissance importante et continue des quantités de trafic véhiculé, et d'un l'autre côté de l'émergence et à la popularisation de nombreuses applications dites non standards telles que les applications P2P qui causent une forte concurrence dans la réservation des ressources réseaux pour les applications de haute priorité telles que la VoIP ce qui peut détériorer la qualité de service offert.

Par ailleurs, la convergence des réseaux de télécommunications classiques, dédiés à la téléphonie et les réseaux de transfert de données engendrent plusieurs défis et de nouvelles problématiques de gestion de réseau et de priorités afin de pouvoir satisfaire les utilisateurs finaux. Dans le cadre de ces défis, les outils traditionnels d'analyse de réseau ne permettent pas de répondre à un tel besoin et il est nécessaire d'étudier de nouvelles approches de surveillance de réseau dont les plus prometteuses incluent la visualisation de l'information.

Lors de cette recherche, les problèmes liés à la visualisation du trafic, en présence de grande quantité de données en temps réels ainsi que leurs défis de la visualisation des données multidimensionnelles ont été étudiés. Pour y parvenir, nous avons recherché un ensemble de techniques de visualisation adaptées accompagnées de méthodes de réduction de la quantité de données. La solution proposée comprend plusieurs modules.

Après avoir survolé et analysé les travaux effectués dans les domaines de recherche liés au sujet de cette recherche, en particulier : la visualisation de l'information, la classification et l'échantillonnage de trafic sont présentés et synthétisés au le chapitre 2. La méthodologie de recherche suivie afin de résoudre les problèmes liés au présent sujet de recherche et a été décrite au le chapitre 4. La contribution principale est une plateforme de visualisation de

trafic qui comporte quatre modules; le module de collecte de trafic, le module de préparation des donnés, le module d'analyse et le module des représentations graphiques.

La validation des fonctionnalités de la proposition a été effectuée en utilisant un réseau émulé et connecté à l'Internet. De plus, les performances, particulièrement le CPU et la mémoire ont été évaluées en comparaison avec une application existante de visualisation de trafic (TNV). Les résultats démontrent que la solution proposée permet une visualisation de trafic à un coût plus bas, en termes de consommation de ressources et du temps de traitement.

Dans le même ordre d'idée, l'étude comparative des fonctionnalités révèle que la solution proposé satisfait la problématique d'analyse en temps réel et prend en considération le trafic interne dans le processus d'analyse pour permettre une surveillance générale, et qui pourrait aider à prévenir les attaques internes qui sont indétectables lors de l'utilisation de la majorité des systèmes de visualisation actuellement disponibles.

Au niveau de l'analyse d'application, l'utilisation des sous-flux pour la classification du trafic permet une approche de temps réel et d'identifier les flux le plus rapidement possible (c.-à-d. dès la détection du premier sous flux) avec un taux de classification qui peut atteindre plus que 99%. Les classifieurs expérimentés dans cette recherche à savoir le classifieurs C4.5, le classifieur RandomForest et le classifieur SVM obtiennent respectivement un taux de classification de 99,88%, 99,94% et 99,25%. Néanmoins, la différence entre ces classifieurs réside dans leurs coûts d'exécution, plus précisément relié au temps d'apprentissage qui est sensiblement important pour les classifieurs SVM malgré l'utilisation des techniques de réduction de données. Contrairement au classifieur SVM, les algorithmes des arbres de décision permettent d'atteindre des résultats prometteurs tout en étant rapides et moins coûteux. Une analyse des taux de précision et de rappel démontre que l'algorithme C4.5 est plus stable pour toutes les classes de trafic. Conséquemment, le classifieur basé sur l'algorithme C4.5 a été choisi pour être intégré dans la présente plateforme de visualisation pour l'analyse du trafic au niveau application.

Travaux futurs

Les résultats expérimentaux de la plateforme de visualisation, et ses performances sont prometteuses. Cependant, l'analyse du trafic à large échelle et la surveillance des réseaux de haute vitesse sont complexes, notamment avec l'évolution continue des réseaux et des technologies ainsi que le besoin qu'elle crée à plusieurs niveaux, dont la classification du trafic, le contrôle d'accès, la gestion de la qualité de services (QoS), la sécurité, et bien d'autres aspects. Avce l'objectif d'améliorer la solution décrite ci-dessus nous proposons un ensemble de pistes de travaux futurs éventuels. Ces derniers sont résumés comme suit :

- l'échantillonnage de trafic permet de réduire la quantité de données de trafic collectées et analysées. Toutefois il est important de mener des investigations de l'impact de l'échantillonnage sur la précision de l'analyse de trafic;
- bien que l'utilisation des sous-flux donne des résultats encourageants, il serait intéressant d'étudier l'intérêt et l'efficacité de cette méthode dans le cas des flux instables, c'est à dire qui changent de caractéristiques au cours du temps;
- L'intégration d'autres fonctionnalités d'analyse, dans la plateforme de visualisation, dans le but d'assurer une visibilité complète d'états et des activités réseau. L'Extension de la proposition afin d'analyser des types de trafic autre que le trafic IP. Finalement il serait opportun d'introduire une analyse permettant d'identifier les flux larges et les flux souris.

Plusieurs parties de cette recherche ont fait l'objet d'une publication de l'article : M. Elbaham, K. K. Nguyen and M. Cheriet, "A traffic visualization framework for monitoring large-scale inter-datacenter network," *2016 12th International Conference on Network and* Service Management (CNSM), Montreal, QC, 2016, pp. 277-281.

ANNEXE I

CARACTÉRISTIQUES DE FLUX

Tableau-A I-1 Caractéristiques de flux

Attribut	**type**	**description**
srcip	Chaîne de caractère	L'adresse source du flux
srcport	numérique	Le port source du flux
dstip	Chaîne de caractère	L'adresse destination du flux
dstport	numérique	Le port destination du flux
proto	numérique	Protocole (TCP=6, UDP=17)
total_fpackets	numérique	Le nombre total de paquets du flux dans la direction « forward »
total_fvolume	numérique	Le nombre total d'octets du flux dans la direction « forward »
total_bpackets	numérique	Le nombre total de paquets du flux dans la direction « backward »
total_bvolume	numérique	Le nombre total d'octets du flux dans la direction « backward »
min_fpktl	numérique	La taille du plus petit paquet envoyé dans la direction « forward » (en octets)
mean_fpktl).	numérique	La taille moyenne des paquets envoyés dans la direction forward (en octets)
max_fpktl.	numérique	La taille du plus grand paquet envoyé dans la direction « forward » (en octets)
std_fpktl	numérique	L'écart type par rapport à la moyenne des paquets envoyés dans la direction « forward » (en octets)
min_bpktl	numérique	La taille du plus petit paquet envoyé dans le sens inverse « backward » (en octets).
mean_bpktl	numérique	La taille moyenne des paquets envoyés dans le sens inverse « backward » (en octets).
max_bpktl	numérique	La taille du plus grand paquet dans le sens inverse « backward » (en octets).

std_bpktl	numérique	L'écart-type par rapport à la moyenne des paquets envoyés dans le sens inverse « backward » (en octets).
min_fiat	numérique	Le temps minimum entre deux paquets envoyés dans la direction « forward » (en microsecondes).
mean_fiat	numérique	Le temps moyen entre deux paquets envoyés dans la direction « forward » (en microsecondes).
max_fiat	numérique	La durée maximale entre deux paquets envoyés dans la direction « forward » (en microsecondes).
std_fiat	numérique	L'écart type par rapport au temps moyen entre deux paquets envoyés dans le sens direct « forward » (en microsecondes).
min_biat	numérique	La durée minimale entre deux paquets envoyés en s ens inverse « backward » (en microsecondes).
mean_biat	numérique	Le temps moyen entre deux paquets envoyés en sen s inverse « backward » (en microsecondes).
max_biat	numérique	La durée maximale entre deux paquets envoyés en s ens inverse « backward » (en microsecondes).
std_biat	numérique	L'écart-type par rapport au temps moyen entre deux paquets envoyés dans le sens inverse « backward » (en microsecondes).
duration	numérique	La durée du flux (en microsecondes)
min_active	numérique	Le temps minimum pendant lequel le flux était actif avant de passer au repos (en microsecondes).
mean_active	numérique	Le temps moyen pendant lequel le flux était actif av ant s'être inactif (en microsecondes).
max_active	numérique	Le temps maximum pendant lequel le flux était actif avant d'être inactif (en microsecondes).
std_active	numérique	L'écart-type par rapport au temps moyen pendant le quel le flux était actif avant de passer au repos (en microsecondes).
min_idle	numérique	Le temps minimum pendant lequel un flux était inactif avant de devenir actif (en microsecondes).
mean_idle	numérique	Le temps moyen qu'un flux était inactif avant de devenir actif (en microsecondes).

max_idle	numérique	Le temps maximum pendant lequel un flux était inactif avant de devenir actif (en microsecondes).
std_idle	numérique	L'écart type par rapport au temps moyen pendant lequel un flux était inactif avant de devenir actif (en microsecondes).
sflow_fpackets	numérique	Le nombre moyen de paquets dans un sous-flux dans le sens direct « forward »
sflow_fbytes	numérique	Le nombre moyen d'octets dans un sous-flux dans le sens direct « forward »
sflow_bpackets	numérique	Le nombre moyen de paquets dans un sous-flux dans le sens inverse « backward »
sflow_bbytes	numérique	Le nombre moyen d'octets dans un sous-flux dans le sens inverse « backward »
fpsh_cnt	numérique	Le nombre de fois où l'indicateur PSH a été placé da ns des paquets se déplaçant dans la direction directe (0 pour UDP).
bpsh_cnt	numérique	Le nombre de fois où l'indicateur PSH a été placé da ns des paquets se déplaçant dans la direction inverse (0 pour UDP).
furg_cnt	numérique	Nombre de fois où le drapeau URG a été placé dans des paquets se déplaçant dans la direction directe (0 pour UDP).
burg_cnt	numérique	Nombre de fois où le drapeau URG a été placé dans des paquets se déplaçant dans la direction « backward » (0 pour UDP).
total_fhlen	numérique	Les octets totaux utilisés pour les en-têtes dans la dir ection directe « forward ».
total_bhlen	numérique	Les octets totaux utilisés pour les en-têtes dans la dir ection inverse « backward ».

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Abdelhamid, D. (2012). « *Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données* ». (thése de doctorat), Mohamed Khider, Biskra. En ligne. < http://abdelhamid-djeffal.net/web_documents/thesedjeffal.pdf >. Consulté le 27 mai 2017.

Abdullah, K., Lee, C., Conti, G., et Copeland, J. A. (2005). « *Visualizing network data for intrusion detection* ». In the Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop.

Afaq, M., Rehman, S. U., et Song, W.-C. (2015). « A Framework for Classification and Visualization of Elephant Flows in SDN-Based Networks ». *Procedia Computer Science, 65*, 672-681.

Aigner, W., Miksch, S., Schumann, H., et Tominski, C. (2011). «*Visualization of time-oriented data*». Springer Science & Business Media.

Alshammari, R., et Zincir-Heywood, A. N. (2010). « *An investigation on the identification of VoIP traffic: Case study on Gtalk and Skype* ». In the 2010 International Conference on Network and Service Management.

Arndt, D. (2015). « netmate-flowcalc ». En ligne. < https://code.google.com/archive/p/ netmate-flowcalc/ >. Consulté le 11 janvier 2017.

Au, S. C., Leckie, C., Parhar, A., et Wong, G. (2004). « Efficient visualization of large routing topologies ». *International Journal of Network Management, 14*(2), 105-118.

Awad, H., Ibrahim, H., Sulaiman, M. N., Izzeldin, I. M., Mohamed Saad, M., et Haitham, A. J. (2014). « Real-time Traffic Classification Algorithm Based on Hybrid of Signature Statistical and Port to Identify Internet Applications ».

Ball, R., Fink, G. A., et North, C. (2004). « *Home-centric visualization of network traffic for security administration* ». In the Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security.

Barros, R. C., de Carvalho, A. C., et Freitas, A. A. (2015). «*Automatic design of decision-tree induction algorithms*». Springer.

Berkhin, P. (2006). « A survey of clustering data mining techniques » *Grouping multidimensional data* (pp. 25-71). Springer.

Biau, G. (2012). « Analysis of a random forests model ». *Journal of Machine Learning Research, 13*(Apr), 1063-1095.

Breiman, L. (1996). « Bagging predictors ». *Machine learning, 24*(2), 123-140.

Breiman, L. (2001). « Random forests ». *Machine learning, 45*(1), 5-32.

Brocade. (2009). « *Brocade sFlow for Network Traffic Monitoring* ». White Paper.

Captcp. « Captcp Analyser ». En ligne. < http://research.protocollabs.com/captcp/ >. Consulté le 17 janvier 2017.

Card, S. K., Mackinlay, J. D., et Shneiderman, B. (1999). «*Readings in information visualization: using vision to think*». Morgan Kaufmann.

Chi, E. H. (2000). « *A taxonomy of visualization techniques using the data state reference model* ». In the Information Visualization, 2000. InfoVis 2000. IEEE Symposium on.

Choi, B.-Y., Park, J., et Zhang, Z.-L. (2003). « *Adaptive random sampling for traffic load measurement* ». In the Communications, 2003. ICC'03. IEEE International Conference on.

Cisco. (2005). « Random Sampled NetFlow ». En ligne. < http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/nfstatsa.html#wp1078265 >. Consulté le 31 janvier 2017.

Cisco. (2016a). « Cisco IOS NetFlow Version 9 Flow-Record Format ». En ligne. < http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html >. Consulté le 31 janvier 2017.

Cisco. (2016b). « The Zettabyte Era- trends and analysis- Cisco ». En ligne. < http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html >. Consulté le 31 janvier 2017.

Claffy, K. C., Braun, H.-W., et Polyzos, G. C. (1995). « A parameterizable methodology for Internet traffic flow profiling ». *IEEE Journal on selected areas in communications, 13*(8), 1481-1494.

Claffy, K. C., Polyzos, G. C., et Braun, H.-W. (1993). « *Application of sampling methodologies to network traffic characterization* ». In the ACM SIGCOMM Computer Communication Review.

D'Alessandro, V., Park, B., Romano, L., et Fetzer, C. (2015). « *Scalable network traffic classification using distributed support vector machines* ». In the 2015 IEEE 8th International Conference on Cloud Computing.

Dos Santos, S., et Brodlie, K. (2004). « Gaining understanding of multivariate and multidimensional data through visualization ». *Computers & Graphics, 28*(3), 311-325. En ligne < http://ac.els-cdn.com/S0097849304000251/1-s2.0-S0097849304000251-main.pdf?_tid=70df7518-f3b9-11e6-9832-00000aab0f02&acdnat=1487189027_e0f704b09f204d1027c5d4059d44787e >.

Drobisz, J., et Christensen, K. J. (1998). « *Adaptive sampling methods to determine network traffic statistics including the hurst parameter* ». In the Local Computer Networks, 1998. LCN'98. Proceedings., 23rd Annual Conference on.

Dundas. (2017). « Using TreeMap ». En ligne. < http://www.dundas.com/support/learning/documentation/data-visualizations/using-a-treemap >. Consulté le 5 avril 2017.

Dupay, A., Sengupta, S., Wolfson, O., et Yemini, Y. (1991). « NETMATE: A network management environment ». *IEEE Network, 5*(2), 35-40.

Este, A., Gringoli, F., et Salgarelli, L. (2009). « Support vector machines for TCP traffic classification ». *Computer networks, 53*(14), 2476-2490. En ligne < http://ac.els-cdn.com/S1389128609001649/1-s2.0-S1389128609001649-main.pdf?_tid=59c99dc4-b654-11e6-8993-00000aacb360&acdnat=1480438588_473952f3fd7cb7d01c66fe02c4698144 >.

Finamore, A., Mellia, M., Meo, M., et Rossi, D. (2010). « Kiss: Stochastic packet inspection classifier for udp traffic ». *IEEE/ACM Transactions on Networking, 18*(5), 1505-1515.

François Denis, et Gilleron, R. « Apprentissage automatique : les arbres de décision ». En ligne. < http://www.grappa.univ-lille3.fr/polys/apprentissage/sortie004.html >. Consulté le 11 janvier 2017.

Géraldine, T. (2012). « Métrologie des réseaux- Mesure de la qualité de service ». En ligne. < http://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/administration-de-reseaux-applications-et-mise-en-oeuvre-42481210/metrologie-des-reseaux-te7605/fonctions-de-la-mesure-te7605v2niv10001.html >. Consulté le 27 mai 2017.

Gilbert, R. S., et Kleinöoder, W. B. (1985). «*CNMGRAF—graphic presentation services for network management*» (Vol. 15). ACM.

Glazar, V., Marunic, G., Percic, M., et Butkovic, Z. (2016). « Application of glyph-based techniques for multivariate engineering visualization ». *Engineering Optimization, 48*(1), 39-52.

Goodall, J. R., Lutters, W. G., Rheingans, P., et Komlodi, A. (2005). « *Preserving the big picture: Visual network traffic analysis with tnv* ». In the IEEE Workshop on Visualization for Computer Security, 2005.(VizSEC 05).

Gowsalya, R., et Amali, S. M. J. (2014). « SVM Based Network Traffic Classification Using Correlation Information ». *Networking and Communication Engineering, 6*(5), 188-192.

Gu, C., Zhang, S., et Huang, H. (2011). « Online internet traffic classification based on proximal SVM ». *Journal of Computational Information Systems, 7*(6), 2078-2086.

Gu, R., Wang, H., et Ji, Y. (2010). « *Early traffic identification using Bayesian networks* ». In the 2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content.

Guimarães, V. T., Freitas, C. M. D. S., Sadre, R., Tarouco, L. M. R., et Granville, L. Z. (2015). « A Survey on Information Visualization for Network and Service Management ». *IEEE Communications Surveys & Tutorials, 18*(1), 285-323.

Haber, R. B., et McNabb, D. A. (1990). « Visualization idioms: A conceptual model for scientific visualization systems ». *Visualization in scientific computing, 74*, 93.

Hernandez, J. A., et Serrano, P. (2015). « Probabilistic models for computer networks: Tools and solved problems ». En ligne. < https://books.google.ca/books?id=kdzQBQAAQBAJ&pg=PA14&lpg=PA14&dq=packet+size+and+application+type&source=bl&ots=23UWl0y8Wa&sig=2R4TmDizrQPzd6G_DKD4gBCJ4IU&hl=fr&sa=X&ved=0ahUKEwjpwKLa05bOAhWDej4KHVPyCYsQ6AEIKzAC#v=onepage&q=packet%20size%20and%20application%20type&f=true >. Consulté le 31 septembre 2016.

Ho, T. K. (1998). « The random subspace method for constructing decision forests ». *IEEE transactions on pattern analysis and machine intelligence, 20*(8), 832-844.

Houerbi, K. R. (2009). « *Mesures et Caractérisation du Trafic dans le Réseau National Universitaire (RNU)* ». (Thése de Doctorat), Ecole Nationale des Sciences de l'Informatique, université de manouba, Tunis, Tunisie. En ligne. < https://tel.archives-ouvertes.fr/tel-00656376/document >. Consulté le 24 septembre 2016.

IANA. « List of Enregistred port numbers ». En ligne. < http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml >. Consulté le 13 mars 2016.

Ibrahim, H. A. H., Al Zuobi, O. R. A., Al-Namari, M. A., MohamedAli, G., et Abdalla, A. A. A. (2016). « *Internet traffic classification using machine learning approach: Datasets validation issues* ». In the 2016 Conference of Basic Sciences and Engineering Studies (SGCAC).

Iliofotou, M., Kim, H.-c., Faloutsos, M., Mitzenmacher, M., Pappu, P., et Varghese, G. (2009). « *Graph-based p2p traffic classification at the internet backbone* ». In the INFOCOM Workshops 2009, IEEE.

Iliofotou, M., Pappu, P., Faloutsos, M., Mitzenmacher, M., Singh, S., et Varghese, G. (2007). « *Network monitoring using traffic dispersion graphs (tdgs)* ». In the Proceedings of the 7th ACM SIGCOMM conference on Internet measurement.

InMon. « sFlow Standard ». En ligne. < http://www.inmon.com/technology/ >. Consulté le 29 novembre 2016.

InMon. (2004). « sFlow Accuracy and Billing ».

Inselberg, A. (2009). «*Parallel coordinates*». Springer.

InternetSociety. (2015). « Global Internet Report ».

Iperf. « Iperf ». En ligne. < https://iperf.fr/ >. Consulté le 22 juillet 2016.

JFreeChart. « JFreeChart ». En ligne. < http://www.jfree.org/jfreechart/ >. Consulté le 20 mars 2015.

Jpcap. « Jpcap ». En ligne. < http://jpcap.sourceforge.net/ >. Consulté le 23 decembre 2014.

Karagiannis, T., Papagiannaki, K., et Faloutsos, M. (2005). « *BLINC: multilevel traffic classification in the dark* ». In the ACM SIGCOMM Computer Communication Review.

kdnuggets. (2017). « Top 10 Data Mining Algorithms ». En ligne. < http:// www.kdnuggets.com/2015/05/top-10-data-mining-algorithms-explained.html >. Consulté le 27 mai 2017.

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., et Melançon, G. (2008). « Visual analytics: Definition, process, and challenges » *Information visualization* (pp. 154-175). Springer.

Keim, D. A. (1996). « Pixel-oriented database visualizations ». *ACM Sigmod Record, 25*(4), 35-39.

Keim, D. A. (1997). «*Visual techniques for exploring databases*». Bibliothek der Universität Konstanz.

Keim, D. A. (2002). « Information visualization and visual data mining ». *IEEE transactions on Visualization and Computer Graphics, 8*(1), 1-8.

Keim, D. A., Mansmann, F., Schneidewind, J., et Schreck, T. (2006). « *Monitoring network traffic with radial traffic analyzer* ». In the 2006 IEEE Symposium on Visual Analytics Science and Technology.

Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., et Ziegler, H. (2008). « Visual analytics: Scope and challenges » *Visual data mining* (pp. 76-90). Springer.

Khakpour, A. R., et Liu, A. X. (2009). « *High-speed flow nature identification* ». In the Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on.

Knerr, S., Personnaz, L., et Dreyfus, G. (1990). « Single-layer learning revisited: a stepwise procedure for building and training a neural network » *Neurocomputing* (pp. 41-50). Springer.

L7-filter. « Application Layer Packet Classifier for Linux ». En ligne. < http://l7-filter.clearos.com/#application_layer_packet_classifier_for_linux >. Consulté le 12 novembre 2016.

Lakkaraju, K., Yurcik, W., et Lee, A. J. (2004). « *NVisionIP: netflow visualizations of system state for security situational awareness* ». In the Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security.

LARRIEU, N. (2005). « *Controle de congestion et gestion du trafic à partir demesures pour l'optimisation de la QdS dans l'internet* ». (Thése de Doctorat), INSA de Toulouse, France. En ligne. < https://tel.archives-ouvertes.fr/tel-00009745/document >. Consulté le 12 janvier 2017.

Lee, S., Kim, H., Barman, D., Lee, S., Kim, C.-k., Kwon, T.*, et al.* (2011). « Netramark: a network traffic classification benchmark ». *ACM SIGCOMM Computer Communication Review, 41*(1), 22-30.

Lime. (2015). « Évolution d'Internet ». En ligne. < http://www.limeblogue.ca/numerique/evolution-internet/ >. Consulté le 24 janvier 2017.

Ma, J., Levchenko, K., Kreibich, C., Savage, S., et Voelker, G. M. (2006). « *Unexpected means of protocol inference* ». In the Proceedings of the 6th ACM SIGCOMM conference on Internet measurement.

McCormick, B. H., A.DeFanti, T., et D.Brown, M. (1987). Visualization in Scientific Computing*, 21*.

McGraw_Hill. (2002). « Network graphs ». En ligne. < http://www.mhhe.com/math/ltbmath/bennett_nelson/conceptual/netgraphs/graphs.htm >. Consulté le 29 novembre 2016.

Mininet. (2016). « Mininet ». En ligne. < http://mininet.org/ >. Consulté le 31 mai 2016.

Moore, A. W., et Papagiannaki, K. (2005). « *Toward the accurate identification of network applications* ». In the International Workshop on Passive and Active Network Measurement.

Moore, A. W., et Zuev, D. (2005). « *Internet traffic classification using bayesian analysis techniques* ». In the ACM SIGMETRICS Performance Evaluation Review.

NetBeans. « Netbeans ». En ligne. < https://netbeans.org/downloads/ >. Consulté le 17 janvier 2017.

Netflow. (2016). Dans W*ikipédia*. En ligne < https://en.wikipedia.org/wiki/NetFlow >. Consulté le 12 janvier 2017.

Nguyen, T. T., et Armitage, G. (2006). « *Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world ip networks* ». In the Proceedings. 2006 31st IEEE Conference on Local Computer Networks.

Nguyen, T. T., et Armitage, G. (2008). « A survey of techniques for internet traffic classification using machine learning ». *IEEE Communications Surveys & Tutorials, 10*(4), 56-76.

Nmap. « Nmap Network Scanning ». En ligne. < https://nmap.org/man/fr/man-port-scanning-techniques.html >. Consulté le 12 juin 2016.

Olivier, B. (2012). « *Analyse et optimisation de performance des r´eseaux de communication* ». (Thèse de doctorat), Université Toulouse, France. En ligne. < https://tel.archives-ouvertes.fr/tel-00728757/document >. Consulté le 27 mai 2017.

Olshen, L., et Stone, C. J. (1984). « Classification and regression trees ». *Wadsworth International Group, 93*(99), 101.

Paxson, V. (1999). « Bro: a system for detecting network intruders in real-time ». *Computer networks, 31*(23), 2435-2463. En ligne < http://ac.els-cdn.com/S1389128699001127/1-s2.0-S1389128699001127-main.pdf?_tid=52d278ca-b653-11e6-8416-00000aacb361&acdnat=1480438147_791d43c37f1265e492d373826265ad28 >.

Peter J. Sackett, M. F. Al-Gaylani, Ashutosh Tiwari, et Williams, D. (2016). « A review of data visualization: Opportunities in manufacturing sequence management ». *International Journal of Computer Integrated Manufacturing*(5 avril 2017), 689-704 En ligne < https://www.researchgate.net/publication/220381967_A_review_of_data_visualization_Opportunities_in_manufacturing_sequence_management >.

Piccolo2D. « Piccolo2D java ». En ligne. < http://piccolo2d.org/download.html >. Consulté le 25 février 2015.

Ramachandran, V., et Street, D. (2012). « *Pathsift: a library for separating the effects of topology, policy, and protocols on IP routing* ». In the Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques.

Rivillo, J., Hernández, J.-A., et Phillips, I. W. (2005). « On the efficient detection of elephant flows in aggregated network traffic ». *Research School of Informatics, Loughborough University, Tech. Rep*.

Router-Switch. (2015). « Cisco Catalyst 4948E NetFlow-lite/NFLite in Detail ». En ligne. < http://blog.router-switch.com/2015/05/cisco-catalyst-4948e-netflow-litenflite-in-detail/ >. Consulté le 31 janvier 2017.

Ruggieri, S. (2002). « Efficient C4. 5 [classification algorithm] ». *IEEE transactions on knowledge and data engineering, 14*(2), 438-444.

Salvador, E. M., et Granville, L. Z. (2008). « *Using visualization techniques for SNMP traffic analyses* ». In the Computers and Communications, 2008. ISCC 2008. IEEE Symposium on.

Schneider, P. (1996). « TCP/IP traffic Classification Based on port numbers ». *Division Of Applied Sciences, Cambridge, MA, 2138*.

Schonwalder, J., Pras, A., Harvan, M., Schippers, J., et van de Meent, R. (2007). « *SNMP traffic analysis: Approaches, tools, and first results* ». In the 2007 10th IFIP/IEEE International Symposium on Integrated Network Management.

Sen, S., Spatscheck, O., et Wang, D. (2004). « *Accurate, scalable in-network identification of p2p traffic using application signatures* ». In the Proceedings of the 13th international conference on World Wide Web.

Singh, K., Agrawal, S., et Sohi, B. (2013). « A Near Real-time IP Traffic Classification Using Machine Learning ». *International Journal of Intelligent Systems and Applications, 5*(3), 83.

So-In, C. (2009). « A survey of network traffic monitoring and analysis tools ». *Cse 576m computer system analysis project, Washington University in St. Louis*.

System, C. « Cisco IOS Netflow ». En ligne. < http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html >. Consulté le 12 janvier 2016.

Tax, D. M., et Duin, R. P. (1999). « Support vector domain description ». *Pattern recognition letters, 20*(11), 1191-1199. En ligne < http://ac.els-cdn.com/S0167865599000872/1-s2.0-S0167865599000872-main.pdf?_tid=8180e204-b736-11e6-ad15-00000aacb362&acdnat=1480535721_2661ca7a2eab40d291391a8b14a45655 >.

Tecnológicas, A. S. (2016). « Netflow ». En ligne. < http://wiki.pandorafms.com/index.php?title=Pandora:Documentation_en:Netflow >. Consulté le 03 avril 2017.

Telesis, A. (2013). « How to sFlow in a network ».

Timofeev, R. (2004). « *Classification and regression trees (CART) theory and applications* ». (Master Thése de Master), Humboldt University, Berlin, Berlin. En ligne. < http://s3.amazonaws.com/academia.edu.documents/38106508/timofeev.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1490646499&Signature=%2FPcNUS4Gxxzdl%2FzC0NkUBlZvk%2Fs%3D&response-content-disposition=inline%3B%20filename%3DClassification_and_Regression_Trees_CART.pdf >. Consulté le 11 janvier 2017.

Top. « Comande line Top ». En ligne. < http://www.computerhope.com/unix/top.htm >. Consulté le 20 juin 2016.

Tshark. « Tshark ». En ligne. < https://www.wireshark.org/docs/wsug_html_chunked/AppToolstshark.html >. Consulté le 20 juin 2016.

University, B. (2009). « UNIBS trafic traces ». En ligne. < http://netweb.ing.unibs.it/~ntw/tools/traces/ >. Consulté le 15 janvier 2016.

Vapnik, V. N., et Vapnik, V. (1998). «*Statistical learning theory*» (Vol. 1). Wiley New York.

Weka. « Weka ». En ligne. < http://www.cs.waikato.ac.nz/ml/weka/ >. Consulté le 26 janvier 2016.

Williams, N., Zander, S., et Armitage, G. (2006). « Evaluating machine learning methods for online game traffic identification ». *Centre for Advanced Internet Architectures,* http://caia. *swin. edu. au/reports C, 60410*.

Witten, I. H., et Frank, E. (2005). «*Data Mining: Practical machine learning tools and techniques*». Morgan Kaufmann.

Witten, I. H., Frank, E., Hall, M. A., et Pal, C. J. (2016). «*Data Mining: Practical machine learning tools and techniques*». Morgan Kaufmann.

Yin, X., Yurcik, W., Treaster, M., Li, Y., et Lakkaraju, K. (2004). « *VisFlowConnect: netflow visualizations of link relationships for security situational awareness* ». In the Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security.

Yu, J., Lee, H., Im, Y., Kim, M.-S., et Park, D. (2010). « Real-time Classification of Internet Application Traffic using a Hierarchical Multi-class SVM ». *TIIS, 4*(5), 859-876.

Yuan, R., Li, Z., Guan, X., et Xu, L. (2010). « An SVM-based machine learning method for accurate internet traffic classification ». *Information Systems Frontiers, 12*(2), 149-156. En ligne < http://download.springer.com/static/pdf/349/art%253A10.1007%252Fs10796-008-9131-2.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Farticle%2F10.1007%2Fs10796-008-9131-2&token2=exp=1480537428~acl=%2Fstatic%2Fpdf%2F349%2Fart%25253A10.1007%25252Fs10796-008-9131-2.pdf%3ForiginUrl%3Dhttp%253A%252F%252Flink.springer.com%252Farticle%252F10.1007%252Fs10796-008-9131-2*~hmac=2636a48feb2b7f8e3100b78b8244e6a40444873929d64ce980023a3e77efe8ccc >.

Yurcik, W. (2006). « *VisFlowConnect-IP: a link-based visualization of Netflows for security monitoring* ». In the 18th Annual FIRST Conference on Computer Security Incident Handling.

Zhang, H., Chen, X., et Hu, H. (2013). « *HintVis: The Hierarchical Visualization of Network Traffic Data* ». In the Virtual Reality and Visualization (ICVRV), 2013 International Conference on.