Application Level Performance in Wired and Wireless Environments

Aymen Hafsaoui

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2012-ENST-00

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Informatique et Réseaux »

présentée et soutenue publiquement par

Aymen HAFSAOUI

le 29 Septembre 2011

Peformances de Niveau Applicatif en Environnement Filaire et sans Fil

Directeur de thèse : **Guillaume URVOY-KELLER**
Co-encadrement de la thèse : **Denis COLLANGE**

Jury

M. Ernst BIERSACK, Professeur, Réseaux et Sécurité, EURECOM	Professeur
M. André-Luc BEYLOT, Professeur, IRT, ENSEEIHT	Professeur
M. Martin HEUSSE, Professeur, Drakkar ,ENSIMAG	Professeur
M. Guillaume URVOY-KELLER, Professeur, I3S CNRS/UNSA, Université de Nice	Professeur
M. Matti SIEKKINEN, Docteur, DCS, Université de Aalto	Directeur de Recherche
M. Taoufk EN-NAJJARY, Docteur, Orange Labs	Ingenieur de Recherche
M. Denis COLLANGE, Ingénieur, Orange Labs	Ingénieur de Recherche

THÈSE

TELECOM ParisTech
école de l'Institut Télécom - membre de ParisTech

THÈSE

Présentée pour obtenir le grade de docteur

de TELECOM ParisTech

Spécialité : Informatique et Réseaux

Aymen Hafsaoui

Peformances de Niveau Applicatif en Environnement Filaire et sans Fil

Thèse présentée le 29 Septembre 2011 devant le jury composé de :

Président	**Ernst Biersack, EURECOM, France**
Rapporteurs	**André-Luc Beylot, ENSEEIHT, France**
	Martin Heusse, ENSIMAG, France
Examinateurs	**Denis Collange, Orange Labs, France**
	Matti Siekkinen, Aalto University, Finland
	Taoufik En-Najjary, France Telecom, France
Directeur de thése	**Guillaume Urvoy-Keller, UNSA, France**

Abstract

The interest in traffic measurement and analysis has increased tremendously and provides us with new ways to understand, operate and improve network performance. The heterogeneity of the Internet is constantly increasing, with new access technologies, new client devices and with more and more services and applications. On the other hand, the interest of the research community to measure enterprise network performance has grown, due to a complexity that sometimes rivals Internet. These subjects, of crucial importance for service providers, network managers and companies have already received substantial attention in the research community. Despite these efforts, a number of issues still remain unsolved. This thesis is concerned with TCP traffic, which carries the large majority of the Internet's traffic. While analyzing the performance of TCP transfers, we focused on the connections that correspond to valid and complete transfers, from the TCP perspective. The present work consists of three parts dealing with various aspects of the challenging task of, revisiting TCP performance, performance study and anomaly detection.

In the first part, we revisit most important works and discuss problems faced when we studied TCP performance. We present an overview of the impact of the application, on the TCP transfers. We show that while losses can have a detrimental impact on short TCP transfers, the application significantly affects the transfer time of almost all short and even long flows. In this part we show that measurements from passively collected traces can be biased by specific technologies implemented in Cellular networks to boost performance and control users activity.

In the second part, we compare the performance of Cellular, FTTH and ADSL accesses with traces collected on access networks under the control of the same ISP. We shows that a study of classical performance parameters does not lead to a full understanding of client perceived throughput. Then, we propose and validate a method that drills down into the data transfer of each well-behaved connection. The Data time break-down approach automatically extracts the application, access, server and client behavior impacts from passively observed TCP transfers. It also groups together, with an appropriate clustering algorithm, the transfers that have experienced similar performance over different access technologies. We then characterize some salient aspects of analyzing enterprise traffic and we provide an overview of problems.

In the last part, we focus on the issue of profiling anomalous TCP connections that are defined as functionally correct TCP connections but with abnormal performance. Our method enables to pinpoint the root cause of the performance problem, which can be either losses or some idle times during data preparation or transfer. We apply this methodology to several traces corresponding to Internet and enterprise traffic. We demonstrate the existence of specific strategies to recover from losses on Cellular networks that seem more efficient than what is done currently in wired networks.

Résumé

L'intérêt pour l'analyse passive de traces a considérablement augmenté, nous offrant de nouvelles approches pour analyser et améliorer les performances réseaux. L'hétérogénéité d'Internet est en constante évolution : nouvelles technologies d'accès, des clients avec mobiles et toujours de plus en plus de services et d'applications. D'autre part, l'intérêt pour la mesure de performance des réseaux d'entreprises ne cesse de se développer. Ces sujets sont d'une importance cruciale pour les fournisseurs de services Internet, gestionnaires de réseaux et des entreprises, puisqu'ils ont déjà reçu une attention considérable de la part de la communauté de recherche. Malgré ces efforts, un certain nombre de questions reste ouvert. Dans cette thèse on traite le trafic TCP, qui représente la majorité des flux Internet. Lors de cette analyse, nous nous concentrons sur les connexions complètes, du point de vue TCP. Le présent travail se compose de trois parties traitant différent aspects sur les approches actuelles d'analyse de performances de TCP, l'étude des performances et la détection d'anomalies de niveau applicatif.

Dans la première partie, nous présentons les travaux les plus importants, les traces réseaux sur lesquelles nous nous sommes basé ainsi que les problèmes rencontrés lors de l'étude des performances de TCP au niveau applicatif. Nous présentons un premier aperçu de l'impact de l'application sur les transferts TCP. Nous démontrons que si les pertes peuvent avoir un impact négatif sur les petits transferts TCP, l'application affecte de manière significative le temps de transfert de la majorité des flux. Dans cette partie, nous démontrons que certaines mesures peuvent être biaisées par des technologies spécifiques mises en oeuvre dans les réseaux Cellulaires.

Dans la seconde partie, nous comparons sur des traces passives, les performances de clients Internet, d'un même operateur sur les trafics : Cellulaires, FTTH et ADSL. Nous montrons que l'étude des paramètres classiques d'analyse de performance ne permet pas d'expliquer totalement les performances perçues par les clients. Ensuite, nous validons une approche plus fine, permettant de décomposer chaque connexion TCP, bien formée, en intervalles de temps. Notre approche de décomposition de connexion TCP permet d'extraire automatiquement l'impact du comportement de l'application, l'accès, le serveur et le client. Nous regroupons, avec des algorithmes adéquats, les transferts avec des performances similaires sur les différents types d'accès. Puis, nous proposons une caractérisation de certains aspects de l'analyse de trafic dans un reseau d'entreprise.

Dans la dernière partie, nous nous concentrons sur la problématique de profilage d'anomalies sur les connexions TCP, définis comme correct mais avec des performances anormales. Notre méthode permet d'identifier la cause des problèmes de performance, qui peuvent être soit des pertes ou bien des temps perdus lors de la préparation des données ou du transfert. Nous appliquons cette approche pour le cas de plusieurs traces de trafic Internet et entreprise. Nous démontrons l'existence d'une adaptation spécifique pour la récupération sur les pertes sur le réseau Cellulaire qui semble plus efficace que sur les réseaux filaires.

Acknowledgements

First of all, I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family. It is with great pleasure and felicity that I would like to express thanks to all the people who have made this thesis possible.

I would like to thank my supervisor, Guillaume Urvoy-Keller, for offering me the opportunity to pursue my doctoral studies at EURECOM. You have continuously supported all my efforts, both in the research and in my personal projects, not to mention the invaluable support and countless contributions to this work. Thank you for everything, it was a truly great experience working with you !

I wish to thank the jury members and the rapporteurs, Ernst Biersack, Andre-Luc Beylot, Martin Heusse, Denis Collange, Matti Siekkinen, and Taoufik En-Najjary, as it is a great honor for me to have them evaluate my work.

I would like also to extend my thanks to all my colleagues and friends at EURECOM for the excellent and truly enjoyable ambiance. My warmest thanks extend to my dear friends, in France, back in Tunisia and in many other corners of the globe, for all the unforgettable moments I shared with them over the past years.

Last, but not least, I want to express my gratitude to my family for their unconditional love, support, and encouragement. I would like to thank my two sisters Chiraz and Douha and my brother Oussama. I would never thank enough my father Abdallah and my mother Samira for their love, trust, and support. Thank you for bringing so much sincere love and happiness to my life.

Table of Contents

List of Figures

List of Tables

Acronyms

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first occurs in the text. The English acronyms are also used for the French summary.

ACK	Acknowledgement
ADSL	Asymmetric Digital Subscriber Line
AL throughput	Application-Level throughput
ALPs	Application Limited Periods
APN	Access Point Name
ARPA	Advanced Research Projects Agency
BTP	Bulk Data Transfer Periods
CCDF	Complementary CDF
CDF	Cumulative Distribution Function
CPU	Central Processing Unit
CWND	Congestion Window
DAG	Data Acquisition and Generation
DBMS	Database Management System
DBS	Database Systems
DDOS	Distributed Denial of Service
Diffserv	Differentiated Services
DMZ	Demilitarized Zone
DSL	Digital Subscriber Line
EE Throughputs	Effective-Exchange Throughputs
EDGE	Enhanced Data Rates for GSM Evolution
EPMAP	End-Point Mapper
ERP	Enterprise Resource Planning
FR/R	Fast Retransmit/Recovery
FTTH	Fiber To The Home
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
GGSN	Gateway GPRS Support Node
HTTP	Hypertext Transfer Protocol
HSDPA	High-Speed Downlink Packet Access
ICMP	Internet Control Message Protocol
IM	Isolate and Merge
IMAP	Internet Message Access Protocol
IMAPS	IMAP Secure
ISP	Internet Service Provider

LAN	Local Area Network
LBNL	Lawrence Berkeley National Laboratory
LDAPS	Lightweight Directory Access Protocol (Over SSL)
LT	Limited Transmit
MSS	Maximum Segment Size
NFS	Network File System
NAT	Network Address Translation
NIC	Network Interface Card
OS	Operating System
PDL	Progressive Download
PEP	Performance Enhancing Proxy
PoP	Point-of-Presence
POP3	Post Office Protocol version 3
POPS	POP Secure
P2P	Peer to Peer
RCA	Root Cause Analysis Tool
RPC	Remote Procedure Call
RTT	Round-Trip Time
SCTP	Stream Control Transmission Protocol
SGSN	Serving GPRS Support Nodes
SIM	Subscriber Identity Module
SQL	Structured Query Language
SMB	Server Message Block
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SMTPS	SMTP Secure
SPAND	Shared Passive Network Performance Discovery
TCP/IP	Transmission Control Protocol/Internet Protocol
TELNET	TErminal NETwork
t-SNE	t-Distributed Stochastic Neighbour Embedding
TV	Television
UCLA	University of California, Los Angeles
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
VPN	Virtual Private Network
3G	Third Generation
3GPP	3rd Generation Partnership Project

Introduction

The Transmission Control Protocol/Internet Protocol (TCP/IP) is the dominant packet processing protocol in local and wide area networks. The TCP/IP protocols were initially developed as part of the research network. In the late 1960s the Advanced Research Projects Agency (ARPA, now called DARPA) of the U.S. Department of Defence began a partnership with U.S. universities and the corporate research community to design open, standard protocols and build multi-vendor networks. The Internet is a primary reason why TCP/IP is what it is today. In fact, the Internet and TCP/IP are so closely related in their history that it is difficult to discuss one without also talking about the other. TCP/IP has over the years continued to evolve to meet the needs of the Internet and also smaller, private networks that use the technology.

In the last 15 years, the interest in data collection, measurement and analysis of traffic has increased steadily. There has been an immense effort in recent years on various aspects of Internet measurements. Significant progress has been made on many fronts. Important aspects of the Internet's structure have been measured, and some general understanding of how the network is organized is starting to emerge, e.g, measuring the available bandwidth, capacity, application classification. However, there are still some missing pieces in the puzzle. In particular, there is a need for Internet Service Providers (ISPs) to measure the offered services and the performance of their end clients in order to overcome the identified network problems.

While wide-area Internet traffic has been heavily studied for many years, the characteristics of traffic inside enterprises remains almost wholly unexplored. Nearly all of the studies of enterprise traffic available in the literature are well over a decade old and focus on individual Local Area Networks (LANs) rather than whole sites. One likely reason why enterprise traffic has gone unstudied for so long is that it is technically difficult to measure. Unlike Internet traffic, which we can generally monitor by recording a single access link, an enterprise of significant size lacks a single choke-point for its internal traffic that would ease the measurement task.

From the beginning, the Internet was intended to provide a general infrastructure on which a wide range of applications could operate - Internet is less a traditional network and more like a programmable computer. In fact, the design of the Internet envisions two sorts of objectives at the same time. The ability to support a range of applications is critical, but so is the ability to operate over a range of new emergent network access technologies such as Cellular and high speed ones, different than a classical Digital Subscriber Line (DSL).

The most popular are Cellular access, on the one hand, and Fiber To The Home (FTTH) and Asymmetric Digital Subscriber Line (ADSL) on the other hand. Satellite connections are also available, but tend to only be practical in cases where no wired connection exists, ,e.g., to connect isolated Islands.

Nowadays Cellular networks offer the ability to connect to high-speed Internet, that gives full mobility to consumers. Several technologies are available, such as General Packet Radio Service

(GPRS), Third Generation (3G), Universal Mobile Telecommunications System (UMTS),etc.

ADSL and cable are arguably the most popular Internet accesses [1], and also in France. It is a broadband connection technology which enables computers to connect to the Internet using existing copper wired telephone networks. The main idea of DSL technology is that it works by splitting the existing telephone line signal into two parts : one for voice and the other for data.

In contrast to DSL, FTTH systems involve the installation of optical fiber from homes to a central point. The fiber technology is now becoming more accessible for residential Internet client. It promises speeds of up to 100M bit/sec, but costs considerably more than DSL or existing cable services.

With the emergence of those new technologies to access the Internet, we notice an emergence of new applications and services. Services previously designed for ADSL lines are now used in networks with low or high latency.

The need we address in this thesis is the ability to develop a global methodology that allows to study the performance of heterogeneous access technologies and to attribute performance problems perceived by the client separately to the specific characteristics of the access technology, behavior of the server, or behavior of the client. This is indeed a challenging task given that few seminal works [2, 3] had focused on this problem and developed TCP root cause analysis method, that allow users to determine from a passively captured packet trace the primary cause for the throughput limitation. The work by Matti Siekkinen et all constituted a starting point for our TCP performance analysis. While authors in [2] enumerate a number of causes that limit the throughput achieved, it was dedicated to TCP connections that carry at least 130 data packets. As short flows constitute the majority of flows, we decided to address the challenge of devising a generic methodology of profiling TCP connections, irrespectively of their size.

The task of profiling TCP connections is difficult with the variety of applications and the evolution of existing ones. Furthermore, with the new generation of access technologies such as Cellular and Fiber families, service providers have expressed their deep interest in being able to locate the main factors that limit throughput for Internet clients. It is important for service provider to demonstrate that poor client performance is not only due to the access link only, as the access link capacity is the usual suspect.

So far, despite extensive research in the domain, a number of aspects remain unsolved. In this thesis, through extensive evaluation, we uncover several formerly overlooked issues, as for instance, revisiting performances of short TCP transfers and presenting a definition in-line with their performance : connection unable to perform Fast Retransmit/Recovery (FR/R), after a packet loss detection.

To tackle the problem of root cause analysis, we adopt a divide and conquer approach, where we first focus on losses, which are arguably a major cause of performance problems for TCP. Next, we analyze the transfers or the parts of transfers that are unaffected by losses. We use a fine grained methodology, based on TCP data time transfers break-down to profile TCP transfers in general and discuss how anomalies can be uncovered by applying this technique.

For our work we collected several traces from different environments : traffic Internet from the network of an European ISP (Cellular, FTTH and ADSL), a wireless hotspot and research lab and a trace from Enterprise traffic. Those traces were collected during different periods of times. The subtlety of this diversity is to avoid the fact that obtained results be biased by locality or temporal aspects. We intend to propose a global performance analysis approach with a broad scope and application agnostic (no assumption is made concerning the application on top of TCP) or the traces considered.

Thesis Claims and Structure

We make the following claims in this thesis :

I. While losses can have a detrimental impact on short TCP transfers, the application significantly affects the transfer time of almost all short - and even long - flows in a variety of ways. Indeed, the application can induce large tear-down times and it can slow the rate of actual TCP transfers or affect the ability of TCP to recover using Fast Retransmit/Fast Recovery.

II. Several specific devices might affect classical performance metrics in Cellular networks, which should be taken into account when performing measurement studies.

III. Round-Trip Time (RTT) and packet loss alone are not enough to fully understand the observed differences or similarities of performance between the different access technologies,

IV. Our data time break down methodology for traffic analysis enables :
   - to present a general approach for traffic analysis based on passive measurements, available for multiple environments,
   - to perform a fine-grained profiling of the data time of transfers that sheds light on the interplay between service, access and usage, for the client and server side,
   - to attribute performance differences perceived by the client separately to the specific characteristics of the access technology, behavior of the server, and behavior of the client.

V. We aim at detecting and uncovering the reason behind ill-behaved TCP transfers, where a ill-behaved connection here is a functionally correct TCP connection – normal set-up/tear-down and actual data transfer – that experienced performance issues, e.g. losses or abnormally long waiting times at the server side.

VI. Our approach for detecting and uncovering the reason behind ill-behaved TCP transfers, is able to isolate various types of anomalies, some being related to the configuration of servers and some other being shared by several services.

Our thesis deals with different aspects of traffic measurements for the case of heterogeneous environments. In addition to the introduction and final conclusion, we divide the content of this thesis in three main parts.

In the first part, we introduce the world of Internet/Intranet measurements and revisit most of the important related works. We highlight problems faced when we revisited TCP performance. We introduce a new definition of short transfers. This part includes 3 chapters.

In Chapter 1, we briefly review the most important works that we focused on, and present a first overview of the problems tackled in the thesis. We then present InTraBase - the traffic analysis tool (used to manipulate all our traffic traces). We summarize the main characteristics of traces at the packet level used in this work to carry out our traffic analysis.

In Chapter 2, we highlight the interplay between TCP and the application on top. We discuss the definition commonly made of short TCP transfers. We observe that, while losses can have a detrimental impact on short TCP transfers, the application significantly affects the transfer time of almost all short - and even long - flows in a variety of way.

In Chapter 3, we study the performance of Cellular access networks and we bring to light phenomena introduced by Cellular core network equipments, which can bias measurements. As a second step we investigate the performance of Cellular networks, focusing on two key services : mail and webmail.

In the second part we compare the performance of Cellular, FTTH and ADSL accesses with traces collected on access networks under the control of the same ISP. We show that loss and access impacts are not only the main parameters that influence clients perceived performance. This part includes 5 chapters.

In Chapter 4, we report a classical approach to compare performance of different access technologies in order to conclude if clients fully benefit from their broadband access. We assess the stability of the traffic for the traces that we have to study. We briefly analyse usual suspects and parameters that can impact the results of different access technologies.

In Chapter 5, we propose a new analysis method that uncovers the impact of specific factors like the application and the interaction with user, and thus informs the comparison of heterogeneous access technologies.

In Chapter 6 we validate key elements of our analysis method, namely the data time breakdown approach and the clustering technique. This validation is achieved through simulations carried out using the Qualnet simulator.

In Chapter 7 we address the problem of comparing the performance perceived by end users when they use different technologies to access the Internet. We apply our data break-down and a clustering approaches to identify groups of connections experiencing similar performance over the different access technologies.

In Chapter 8, we revisit some salient aspects of enterprise traffic. Our goal is to provide an overview of the problem faced when performing measurements in such environments such as basic RTT estimation. We also present a fine-grained profiling of the most popular applications used in the network we measure.

The last part (III) of the thesis focuses on the issue of profiling anomalous TCP connections that are defined as functionally correct TCP connection but with abnormal performance.

In Chapter 9, we present a methodology to profile anomalous TCP connections, which leverages the approach proposed previously and applied to Internet and intranet traffic for profiling all the TCP traffic. We demonstrate the existence of specific strategies to recover from losses on Cellular network that seem more efficient than what is done currently in wired networks. When focusing on the transfers or parts of the transfers that are not affected by losses, we demonstrate that our approach is able to detect and classify different classes of anomalies, especially anomalies due to transient or persistent - provisioning - problems at the server side.

In Chapter 10, we apply a methodology similar to the one proposed in Chapter 9 to the case of characterizing TCP traffic anomalies for enterprise traffic.

Each part starts with a short introduction, contributions summary. Thesis conclusions and perspectives chapter concludes the thesis and gives our opinion on how this research could be extended in the future.

Part I

Challenges in Assessing TCP Performance

Overview of Part I

In Part I we revisit the most important related works, we highlight problems faced when we revisited TCP performance. Then we motivate our approach of data time break down.

In Chapter 1, we present the research efforts related to the different parts of the thesis and we provide a high level overview of the challenges we address in this work. We summarize the main characteristics of the traces captured at the packet level, used in this work to carry out our traffic analysis.

In Chapter 2, we highlight the interplay between TCP and the application on top. We discuss the definition commonly made of short TCP transfers - transfers that cannot rely on the Fast Retransmit/Fast Recovery (FR/R) mechanism - with the emergence of new mechanisms to improve the performance of small transfers, e.g. *limited transmit*. We present an overview of the impact of the application, on the TCP transfers. We show that while losses can have a detrimental impact on short TCP transfers, the application significantly affects the transfer time of almost all short - and even long - flows in a variety of way.

In Chapter 3 we highlight that measurements from passively collected traces can be biased by specific technologies implemented in Cellular networks to boost performance and control users activity. Also, we cast a first look to two key Internet services : mail and webmail in order to identify factors that lead to different perceived performance for the case of Cellular users.

In summary, this part essentially describes challenges of traffic analysis and presents guidelines to use in order to uncover the impact of specific factors like the application and the interaction with user, and thus informs the comparison of access technologies, presented in Part II.

Chapter 1

Overview of Challenges

1.1 Introduction

In this chapter, we present the research efforts related to the different parts of the thesis and we provide a high level overview of the challenges we address in this work. We then present an overview of InTraBase - the traffic analysis tool, used to manipulate the traces and to implement the algorithms we developed. We summarize the main characteristics of the traces captured at the packet level, used in this work to carry out our traffic analysis. Those traces were collected in heterogeneous wireless and wired environments, which highlight the wide scope of our study of traffic analysis performance.

1.2 Short TCP flows

TCP carries 95% of Internet traffic and constitutes 80% of the total number of flows in the Internet [4]. A large majority of TCP flows are short lived, also known as 'Mice'. This highlight the importance of understanding the behavior of short lived-flows. For example mice can contribute about to 6% of global traffic, but represent more than 97% of the total number of flows [5].

1.2.1 Definition of Short TCP flows

Several approaches and definitions have been proposed to present short TCP flows. Some proposals consider short flows as sessions which are smaller than a fixed threshold, e.g, 10 kbytes [6, 7, 8, 9] or 13.5 kbytes [4].

Alternatively, some works [10, 11] define a short connection as connections spending their lifetime in the slow start phase when the congestion window (cwnd) is increased exponentially.

A mouse [5] was defined as data transfer comprising a number of packets less than or equal to 20 packets ; a flow is terminated if no packets of the flow have been observed for a time period of 5 seconds.

Also, Cumulative Distribution Function (CDF) of Hypertext Transfer Protocol (HTTP) responses size with status 200 (class of status code that indicates that the client's request was successfully received) shows that other definitions can be found. For instance, three plausible values are specified [12] : 8 kbytes, 16 kbytes and 32 kbytes.

1.2.2 Short TCP Performance Analysis

Several techniques have previously been proposed for the prediction of the transfer time for a short TCP connection. In [12], two types of predictions are investigated : the estimation based on the initial RTT and the one based on the performance of recent transfers.

Authors introduced some modifications to increase the accuracy of predictions approaches adopting a trace based validation. Cardwell et al [9, 8] propose analytic models to fit TCP behaviour under realistic loss rates in the Internet.

Other proposals [4] include a recursive analytical model to predict the TCP performance of short lived flows in the presence of losses. Ebrahim et al [11] present a systematic study of scenarios where short-lived flows severely impact long-lived TCP flows. They demonstrate that in some cases, a reduction greater than 80% as compared to the throughput achieved in long-lived flows.

1.2.3 Harmful Scenarios for Short TCP Performance

TCP timeout values are based on round-trip time estimations from a flow's data-ack samples. When a connection is initiated, TCP uses a conservative timeout value due to lack of such samples from the connection. These timeout values are large [13] in practice (as large as three seconds). Loss of the connection establishment (SYN, SYN-ACK) can thus cause significantly increase in the latency for short flows.

Fast Recovery can be trigged only when the congestion window is larger or equal to four segments, which can happen only when the flow has at least seven segments [4].

Ayesta et al [6] present two losses scenarios that can be very harmful from the performance point of view and inevitably lead the sender to timeout :

- Congestion window < 1 + number of duplicate Acknowledgements (ACK).
- The remaining amount of data < Number of duplicate ACK * Maximum Segment Size (MSS), then short flows often do not have sufficient traffic to generate three duplicate ACKs.

In those scenarios, the sender will not receive three duplicate ACKs and will have to rely on a timeout to detect the loss. Balakrishnan et al. [14] report recovery from almost 50% of losses in web flows via timeout.

Cardwell et al [9] focus on circumstances under which delayed ACKs can cause relatively large delay for short transfers :

- When the sender sends an initial cwnd of one MSS : in this case the receiver waits in vain for a second segment, until finally its delayed ACK timer fires and sends an ACK.
- When the sender sends segments that are not full-sized segments before sending an ACK or when the sender sends small segments and the Nagle algorithm prevents it from sending further segments. And the receiver implementation waits for two full-sized segments before sending an ACK.

Finally, other reasons can slow down short flows transmission [15] :

- Packet dropping : Most routers deploying droptail queuing policy discard packets indistinguishably under congestion. Because of the poor loss recovery performance, even a small amount of packet drops can slow down short flows greatly. Furthermore, the retransmission of these dropped packets also consumes network resources (for example, bandwidth and buffer space) and makes things even worse.
- High queuing delay : Although routers can provide adequate buffer space to avoid packet dropping, short flows still suffer from the high queuing delay because they may be blocked

by long flows which send tens of packets within one congestion window.

Several proposals that attempt to solve one or more of the problems of short flows, have been proposed and can be mainly classified [16] in three categories : 1) reduce connection overheads [17, 18], 2) share network state information [19, 20, 21, 14, 19, 22] , and 3) improve performance during slow start [23, 24, 25, 6, 26].

1.3 Performance Analysis

1.3.1 The Challenge of Comparing Performance of Different Access Technologies

The domain of Internet measurements is rich with a number of different works. For our case we were especially interested by comparing the performance of heterogeneous wired and wireless networks. We enumerate the most important works from our viewpoint, i.e., we studied the performance of different accesses technologies.

1.3.1.1 Wired Networks

While residential broadband Internet access is popular in many parts of the world, only a few studies have examined the characteristics of such traffic. Users of residential broadband connections will often have different goals than those in other environments, and are not subject to the same sorts of strict acceptable use policies that may regulate their access at work or at school, such as prohibitions against accessing certain Web sites or employing certain applications. Optical technology plays a key role in new telecommunication networks. While this technology has been used for a long time in backbone networks, it progressively becomes available up to the end user through the deployment of FTTH access networks. The bit rates now available for end users reach very high values. Over the past few years, an unprecedented increase in Internet traffic has been observed worldwide, particularly in France due to high penetration rate of fiber-based broadband access.

In [27] the authors analyze passive traffic measurements from ADSL and FTTH commercial networks under the control of the same ISP (Orange). Packet-level traces are used to evaluate the impact of the new fiber access network on traffic characteristics. They demonstrate that only a minority of clients and flows really take advantage of the high capacity of FTTH access. The main reason is the predominance of Peer to Peer (p2p) protocols that do not exploit locality and high transmission capacities of other FTTH clients. The use of FTTH provides a slightly improved performance for the most commonly used peer-to-peer protocols. However, at the current deployment level, measurements show no increase in peer-to-peer traffic locality.

In [28] the authors report aggregated traffic measurements collected over 21 months from seven ISPs covering 42% of the Japanese backbone traffic. In this study, residential broadband traffic accounts for two thirds of the ISP backbone traffic and is increasing at 37% per year, which will force significant reevaluation of the pricing and cost structures of the ISP industry. Authors further investigate residential per-customer traffic in one of the ISPs by comparing DSL and fiber users, heavy-hitters and normal users, and geographic traffic matrices. The results reveal that a small segment of users dictate the overall behavior ; 4% of heavy-hitters account for 75% of the inbound volume, and the fiber users account for 86% of the inbound volume. About 63% of the total residential volume is user-to-user traffic.

The study of dominant applications exhibit poor locality. The distribution of heavy-hitters is heavy-tailed without a clear boundary between heavy-hitters and normal users, which suggests that users start playing with peer-to-peer applications, become heavy-hitters, and eventually shift from DSL to fiber.

Work in [29] constitutes an initial exploration of residential broadband Internet traffic, where authors present a broad range of dominant characteristics of residential traffic across a number of dimensions, including DSL session characteristics, network and transport-level features, prominent applications, and network path dynamics. Authors describe the network activity for more than 20,000 residential DSL customers in an urban area. Among the several observations presented the most important one is : HTTP traffic, not peer-to-peer, dominates. Overall, HTTP makes up nearly 60% of traffic by bytes while peer-to-peer contributes roughly 14%. DSL sessions run quite short in duration, with a median between 20 and 30 min. The short lifetime affects the rate of IP address reassignments, and find 50% of addresses are assigned at least twice in 24 h, and 1 to 5% of addresses more than 10 times, with significant implications for IP address aliasing. Delays experienced from the customer premise to the ISP's Internet gateway often exceed those over the wide-area path from the gateway to the remote peer (median local component of 46 ms , versus a median remote component of 17 ms). 802.11 wireless networking in customers' homes, and TCP settings on the residential systems, appear to limit the achievable throughput.

In [2], the authors pinpoint factors that limit ADSL performance through the analysis of a 24-hours packet trace containing TCP traffic of approximately 1300 residential ADSL clients.

The authors underscore a low utilization of upload and download capacity for most of the clients. To carry out the study (see Section 1.5.1), they rely on a TCP Root Cause Analysis Tool (RCA).

Application of RCA shows that in over 90% of the cases, the low utilization is mostly due to the p2p applications clients use, which limits the transmission rate and not due to network congestion. For instance, p2p applications typically impose upload rate limits to avoid uplink saturation that hurts download performance.

1.3.1.2 Wireless Networks

The past few years have seen a fast growth in Cellular data network technologies in terms of available services and coverage/usage extent ; smartphones and other advanced portable devices (e.g., iPad), and a wide variety of mobile telecommunication applications (such as mobile web, video conferencing, voice over IP, online social networking, online gaming, e-commerce, etc.). Technology advances in these areas, device, and application form a virtuous circle that further stimulates more technical innovation and drives popularity in the use of mobile applications even higher. 3G wireless communication has increasingly become an integral part of daily life. Rising together with the ever maturing technologies is users' expectation of the 3G service-users are looking beyond basic service availability and starting to demand higher service performance. Thus it is of interest to study the performance of new Cellular networks with available application like p2p, streaming and video conference, designed to wired networks. Several measurement works on 3G networks have been done to obtain a better understanding of 3G networks and to identify possible performance problems.

In [30], authors cast a first look on mobile hand-held device usage from a network perspective and what kind of services users are interested in when they are at home and have access to all

services. They base their study on anonymized packet level data representing more than 20,000 residential DSL customers (not mobile usage in Cellular networks), spanning a period of 11 month, for several 24 hours traces. The purpose was to observe the behavior of mobile users when they are connected via WiFi at home and compare their traffic patterns to the overall residential traffic characteristics.

The authors find that iPhones and iPods are by far the most commonly observed mobile users. This has an impact on the most popular mobile applications : Safari (Apple's browser), iTunes, and Weather. The largest fraction by volume of HTTP content is multimedia. Comparing HTTP object sizes of overall and mobile devices traffic, the authors find that mobiles HTTP objects are on average larger. The contribution of mobile devices to the overall traffic volume is still small, but rapidly growing, especially compared to the overall traffic growth.

In [31] the authors present a study of the mobile data traffic characteristics by analyzing the data traffic trace from a commercial CDMA backbone network. Several characteristics of mobile traffic are presented. For instance, authors show an uneven in/outbound traffic utilization at the mobile, a low average packet size, a short session length, and a high retransmission ratio. Mobile traffic significantly differs from wired residential traffic. The authors report a retransmission rate of 80% and indicate that this observation is due unneeded retransmission. It generates a waste of network bandwidth and negatively influences the transparency of network usage billing. In addition, they correlate short session length with the temporary usage of user behavior in mobile data network. We note that this extremely high retransmission rates has not been reported in any other studies on wireless traffic, which sheds suspicion on this result.

In [32], the authors present results from a measurement campaign for GPRS, Enhanced Data Rates for GSM Evolution (EDGE), Cellular, and High-Speed Downlink Packet Access (HSDPA) radio access, to evaluate the performance of web transfers with and without caching. Results were compared with the ones of a standard ADSL line (down :1Mb/s ; up :256kb/s). Benchmarks reveal that there is a visible gain introduced by proxies within the technologies : HSDPA is often close to ADSL but does not outperform it ; In EDGE, the proxy achieves the strongest improvement, bringing it close to HSDPA performance.

In [33], the authors quantify the improvement provided by a 3G access compared to a 2G access in terms of delays and throughput. Authors measure performance metrics for ISP managed live Television (TV) and Progressive Download (PDL, e.g. YouTube or Deezer).

First they show that for wired access networks (ADSL and FTTH) the average number of servers accessed per subscriber is one order of magnitude lower on the mobile trace, due to the absence of P2P and different user behaviors. Authors show that the Web is still the most popular application for Cellular access. Then they quantify the performance gain from 2G to 3G and show that 3G allows users to experience both higher TCP throughputs and shorter delays. Focusing on the user experience when viewing multimedia content, they show how their behavior differs and how the radio access type influences their performances.

They observe that live TV streams only suffer from moderate packet losses thanks to conservative encoding bitrates chosen according to the radio access type, and note that the quality of live TV streams is high, explaining the popularity of this service. By focusing on the number of playback interruptions, they conclude that PDL streaming allows to efficiently deliver both video and audio content over a 3G access. 2G PDL video streams are often perturbed by interruptions.

In [34], the authors identify and study the most important factors that impact user perceived

performance of network applications on smartphones. They developed a systematic methodology for comparing this performance along several key dimensions such as carrier networks, device capabilities, and server configurations. To ensure a fair and representative comparison, authors conduct controlled experiments, informed by data collected through 3GTest ; a cross-platform measurement tool they designed, executed by more than 30,000 users from all over the world.

In this work, the authors study the 3G network and application performance of four major U.S. wireless carriers including AT & T, Sprint, Verizon, and T-Mobile. They choose popular devices including iPhone, Android G2 from HTC, and Windows Mobile phones from Palm, HTC, and Samsung for carrying out experiments. Results show that their performance varies significantly across network applications.

The four studied different carriers exhibit distinct network performance in terms of throughput, RTT, retransmission rate, and time-of-day effect [34]. TCP throughput, RTT, and retransmission rate vary widely even for a single carrier in measurements taken at different times and locations, e.g., downlink throughput ranges from 50 kbps to 4 Mbps for AT & T, with the median value of about 1 Mbps. The wireless delay in the 3G network dominates the whole network path delay, e.g., latency to the first hop is around 200 ms, which is close to the end-to-end Ping latency to landmark servers distributed across the U.S. Besides networks, devices heavily influence application performance. Given the same content and network condition, different devices exhibit vastly different Web page loading time, e.g., the page loading time of Samsung SCHi760 is consistently twice that of iPhone. Mobile devices can benefit from new content optimization techniques like the data URL scheme, e.g., page loading time for GPhone can improve by 20% in their experiments, despite its already good performance compared to other devices.

1.4 Enterprise Networks

We aim here to present an overview of research activities focusing on enterprise network issues. More specifically, the vast majority of works makes use of measurements collected in wired or wireless networks of enterprise that encompass campus and networks.

The majority of the studies rely on packet [35, 36] flow (Netflow) level traces [37]. This type of trace might be complemented with other sources such as SNMP [38] or syslog data [39]. In addition, information from the lower or higher layers might be requested. For the case of wired network, lower layer data might be topological information [40]. For the case of wireless network, it might be layer two [41], or physical layer data [42]. As for the upper layer, studies rely on operating systems logs [43] or specific application related performance metrics [44].

We list below some fairly general problems faced when analyzing the traffic of enterprise networks :

- The lack of representativity of each and every specific enterprise networks. This problem in fact also pops up while studying Internet traffic, e.g., traffic capture in residential networks in the US varies from traffic observed in residential networks in Europe due to some trends in the use of applications or new services.
- The complexity of establishing ground truth. Some applications are complex, not well document and uncommon to the practitioner of Internet traffic. A typical example is Window services and ERP applications.
- The structure of enterprise networks, where the complexity lies both in the network structure with its use of VLANs and the server side with the server consolidation and virtualization trend observed in typical Enterprise networks.

1.4.1 Measurement Process

A great deal of work has used measurements captured at an enterprise's access link, which allows characterization of network activity involving the external Internet, but does not shed any light on activity that stays confined within the enterprise. More recently, studies have drawn upon measurements made at an enterprise's core routers [35]

Alternatively, some studies have measured communication on the end-hosts themselves [45]. While this approach yields information about all of a host's traffic including communication that occurs outside the enterprise in the case of monitoring on a laptop-the measurements it lacks of a broader context of what is happening in the surrounding network (e.g., network load). A recent alternative approach presented in [46], was to capture traffic at different Ethernet switch ports.

In [46] the authors presented a number of techniques for calibrating packet traces captured at switches connecting end hosts in terms of : gain, loss, time and layout.

They rely on the following principles : (i) using one source of packets as unambiguous 'stakes in the ground' to hunt for thresholds and compare clocks, (ii) employing expected replication of broadcast packets to point to missing events from traces and aid in mapping networks, (iii) leveraging TCP semantics to identify measurement loss, particularly in terms of seemingly erroneous acknowledgments for data never observed in transmission, and (iv) leveraging multiple, simultaneous data collections to further illuminate unrecorded events and improve confidence in the timestamping process.

The authors reveal a predominance of phantoms in switch traces, they see many identical packets very closely separated in time. Authors define phantoms as identical copies of previous packets, observed less than 5 msec in the past.

The take away message here is that measurement of enterprise networks is a difficult task that has received little attention so far.

1.4.2 Preliminary Analysis

In this paragraph, we report on studies that do not rely on any advanced data mining technique but rather use techniques based on descriptive statistics to investigate performance of enterprise networks.

In [35], the authors presented a first work of its kind focusing on the traffic of large enterprise network, collected traces at LBNL (Lawrence Berkeley National Laboratory). Those (publicly available) traces amount for 100 accumulated hours of traffic, although given the large size and even more the complex structure of the LBNL network, they could not capture at a given time instant all the traffic flowing inside the network. They relied on Bro [1], an intrusion detection system that can do deep packet inspection, i.e., look for specific signatures within the packet payload of a trace, to identify the applications having generated traffic in the traces. In particular, they have extended Bro, or more precisely its signature base, to recognize Windows protocols and network file services protocols.

They first look at the overall traffic comparing internal and external traffic volumes. They also looked at the fan-in and fan-out of local peers, given that some local peers are servers accessible from the Internet. Next, they focus on specific applications, some being used in both worlds, e.g., HTTP or mail and some being intranet specific like Windows services and network file services.

The article is mostly descriptive, but they pinpointed some specific phenomena like the existence of failures to establish specific connections internally. They leveraged their knowledge of protocol semantics to check if failures are widespread among local peers (it turns out to be the

1. http ://www.bro-ids.org/publications.html

case) or not. However, they did not try to identify some specific root causes behind those observations. They looked also at the network load. Specifically, they identified traffic peaks at small time scales, indicating the possible existence of transient overload periods. They also addressed the load problem from the end hosts point of view by computing the amount of TCP retransmissions experienced by connections. They observed that TCP retransmission rates could reach values up to 1%, which is less than what is observed for Internet traffic, though still surprisingly large for an intranet.

In [47] the authors present an initial step towards assessing performance within enterprise networks. This work is somehow the sequel of [35] and uses the same data set. The authors base their analysis on a dataset consisting of switch-level packet traces taken at the LBNL over the course of four months. In this work the authors assess the prevalence of broken TCP transactions, applications used, throughput of TCP connections, and phenomena that influence performance, such as retransmissions, out-of-order delivery, and packet corruption. The study of prevalent application in the enterprise dataset shows that most applications are unbalanced in that they contribute a significant fraction of connections or bytes but not both.

The authors show that out-of-order packet delivery is much more rare, with 0.0035% of data packets, than observed for wide-area traffic, and likewise packet corruption and replication. Additionally, they find that 0.5% of TCP senders experience at least one retransmission. A wide range of transfer rates, with connections achieving throughputs between 3 and 12 times those seen for wide area TCP, was observed for internal traffic, which complies with intuition, due to the proximity of servers and clients and the data high rates of the internal network of typical companies.

1.5 How to Detect TCP Performance Anomalies ?

1.5.1 Internet Traffic

Traffic anomaly detection has received a lot of attention over recent years, but understanding the nature of these anomalies and identifying the flows involved is still a manual task, in most cases. Several traffic anomaly detection methods have been proposed. Note that this is different objective from the detection of traffic anomalies, where the focus is to detect threats against the network, e.g. Distributed Denial of Service (DDoS) [48, 49, 50, 51, 52].

Some of the techniques look at changes in traffic feature distributions [53] or apply methods involving the analysis of content or the behavior of each host or group of hosts [54].

More recently, with a new definition of TCP anomaly, Mellia et al [55] propose a heuristic technique to classify TCP anomalies, i.e., segments that have a sequence number different from the expected one, such as out-of-sequence and duplicate segments. In [56], the authors consider the use of the RTTs as a possible signal for detecting network anomalies.

On the other hand only few works have tried to address the problem of detecting traffic anomalies introduced by performance problems of distant server, upper layer application or service usage. The common view of a TCP transfer is that its transmission rate is limited by the network, i.e. by a link with a small capacity or a congested bottleneck link. In [3, 57] the authors defend the thesis that this view is too restrictive. Instead, the limitation causes may lie in different layers of the network stack, either in the end-points or in the middle of the TCP/IP data path.

In [3], Zhang et al pioneered research into the origins of TCP throughput limitation causes. They propose a taxonomy of rate limitations into (i) application, (ii) congestion, (iii) bandwidth,

(iv) sender/receiver window, (v) opportunity and (vi) transport limitations, and second applied it to various packet traces.

In this work the authors examined the rates of flows and the relationship between flow rates and other flow characteristics. They found that fast flows are responsible for most of the bytes transmitted in the Internet. They also observed a strong correlation between flow rate and size, suggesting an interaction between the bandwidth available to a user and what the user does with that bandwidth. They found that the dominant rate limiting factor appears to be congestion and receiver window limits.

More recently, some researchers [57] designed and implemented a set of algorithms, the root cause analysis toolkit, for doing root cause analysis of TCP throughput. They used a classification of TCP throughput limitations, greatly inspired by [3], and extend the scope of this initial work and discuss the difficulties of identifying TCP throughput limitation causes through examples.

The Isolate and Merge (IM) algorithm that partitions the packets of a given TCP connection into application limited periods (ALPs) and bulk data transfer periods (BTPs). A BTP is a period where the TCP sender never needs to wait for the application on top to provide data to transfer. On the other hand, when the TCP sender needs to wait for the application on top, we call that period an ALP. Once BTPs have been identified, the root cause analysis toolkit analyzes them for TCP and IP layer throughput limitations, i.e. inferring the root causes for the BTPs, which will be the focus of this paper. As a next step, Siekkinen et all describe a methodology to quantify TCP and IP level throughput limitations that is referred to as the root cause analysis toolkit. More specifically, they define a set of quantitative metrics, called limitation scores, that can be computed from the information contained in the packet headers collected at a single measurement point, and show how these scores can be used in a threshold-based classification scheme to derive a root cause for a given BTP.

The main limitation of the IM algorithm is that it processes only connections consisting of at least 130 data packets. This threshold is chosen since a TCP sender that starts in slow start needs to transmit approximately 130 data packets (assuming a MSS of 1460 bytes) in order to reach a congestion window size equal to 64 kbytes, which is a common size for the receiver advertised window. One of the starting points of this thesis work is the importance of also profiling the performance of short flows, and more generally of flows of any size.

1.5.2 Enterprise Traffic

Diagnosing problems in enterprise networks is challenging and complex. Modern networks have many components/services that interact in complex ways. Configuration changes in seemingly unrelated files, resource/components elsewhere in the network, and even 'just a software upgrades can ruin what worked perfectly yesterday. Thus, the development of tools to help operators diagnose faults has been the subject of much research and commercial activity [58, 59, 60, 61, 40].

The main difference between those tools and the ones described in the previous paragraph for Internet traffic is that in the context of enterprise networks, the sets of clients and servers are limited and relatively stable, which enables to feed the algorithms with much more information to infer the root of the performance problems in this type of network.

The systems for large enterprises, such as Sherlock [58], target only performance and reachability issues and diagnose at the granularity of machines. They essentially sacrifice detail in order to scale. Other systems, such as Pinpoint for online services [60] and SCORE for ISP networks [40], use extensive knowledge of the structure of their domains. Extending them to perform detailed diagnosis in enterprise networks would require embedding detailed knowledge of each

application dependency and failure mode. The range and complexity of applications inside modern enterprises can make this task difficult.

In [43], the authors of Sherlock adapt their technique to the case of small enterprise networks. It enables detailed diagnosis by harnessing the rich information exposed by modern operating systems and applications. A key challenge in their approach is to be application agnostic. They rely on a large number of heuristics to address many problems like the correlation that exists among the variables exposed by the operating system that they do not know a priori. The resulting solutions appear quite complex due to the many heuristics, even though the complexity of the problem requires such an empirical approach.

Troubleshooting of Enterprise Traffic, with a profiling of hosts approach is a topic that has received a significant attention. The purpose is to gain a logical understanding of the role played by hosts in a network [36, 62] or in complement, to combat malicious activities.

In [36], the authors tackle the problem of role classification of hosts within enterprise networks. Role classification consists in grouping hosts into related roles so as to obtain a logical view of the network in terms of who is using which resources.

In [62], the authors investigate the use of community of interest as a means to characterize data networks. Broadly speaking, a community of interest is defined as a set of communicating hosts. Authors investigate two possible definitions of COIs (popularity/frequency). The main objective of this work was to assess the stability of their two COI definition over the 11 week by varying various parameters. As a main conclusion they obtain that COI tends to be fairly stable and abrupt changes might thus be considered as abnormal behaviors.

In [63], the authors investigate the use of host profiling for enterprise network security. Their main contribution is a clustering algorithm that aims at grouping nodes with similar communication profiles over time. The behavior of a host on a given time interval is summarized through a small set of indicators related to the amount of bytes and packets per destination type and application.

In [64], the authors build upon host profiling to propose a new technique to mitigate propagation of malicious activities within an enterprise network. The starting point is to build a profile of the communications of a host based upon its communication pattern at the transport layer.

Graph techniques constitute an appealing solutions to uncover behavioral characteristics of network traffic. In [65, 37], Traffic Dispersion Graphs (TDGs) are introduced as a mean to visualize and analyze traffic from a specific network.

Several recent works have tackled the problem of mining network traffic in order to uncover temporal relations between flows [66]. In [66], the authors present eXpose, a tool that mines flow level traces to uncover communication patterns in the considered traces. In their approach, a flow trace is transformed into a matrix where rows correspond to time slices while columns correspond to flows where for each time slice, it is indicated whether the flow is present or not. To work around the problem, they propose to use the JMeasure, an entropy based metric.

1.6 Intrabase

Internet traffic analysis as a research area has experienced rapid growth over the last decade due to the increasing needs caused by the massive increase of traffic in the Internet together with new types of traffic generated by novel applications, such as peer-to-peer. Today, the state of the art in traffic analysis is handcrafted scripts and a large number of software tools specialized for a single task. The amount of data used in traffic analysis is typically very large.

For our case, we based our analysis on InTraBase, a traffic analysis tool [67] : a reliable and flexible tool, compared to existing ones [68]. It is a Database Management System (DBMS)-based approach for traffic analysis. It allows to manage collected dumps within the Database System (DBS). In other words, it processes the input data as little as possible prior to loading it into the database.

The data uploaded into the database is referred to as base data. Examples of base data are packet traces collected via tcpdump or a similar tool.

Once the base data is uploaded into the DBS, it is processed to derive new data that is also stored in the database. For instance, it demultiplexes the tcpdump packet traces into connections by assigning a connection identifier to each packet. All the processing is done within the DBS.

InTraBase is not designed, for instance, to monitor the health of a large ISP's network in real-time due to the immense amounts of data that would need to be treated constantly. It is rather an exploratory tool for fine-grained analysis of Internet traffic. It allows to make multiple iterations over the analysis process cycle, which is generally impossible with systems specialized into on-line analysis.

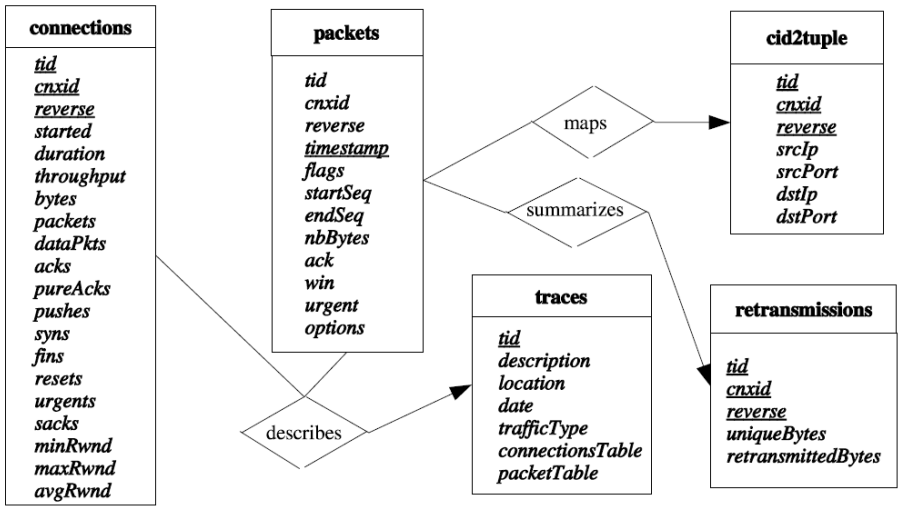


FIGURE 1.1 – Main Tables in InTraBase

The core tables used in InTraBase are described in Figure 1.1 [67]. The table **traces** contains annotations about all the packet traces that are uploaded in the database. The **packets** table holds all packets for a single trace. The two tables **connections** and **retransmissions** hold connection level summary data for all traces. The **cnxid** attribute identifies a single connection in a packet trace, **reverse** differentiates between the two directions of traffic within a connection, and **tid** identifies a single trace. **Cid2tuple** is a table to store a mapping between unique **cnxid** and **4-tuples** formed by source and destination IP addresses and TCP ports. The attributes of the packets table are directly from the standard output of tcpdump for TCP packets. The attributes of the other tables were chosen so that the connection level information roughly covers that given by tcptrace.

Processing a tcpdump packet trace with InTraBase includes five major steps [67] :

1. Copy packets into the **packets** table in the database ;
2. Build an index for the **packets** table based on the connection identifier ;
3. Create connection level statistics from the **packets** table into the **connections** table ;
4. Insert unique **4-tuple** to **cnxid** mapping data from packets table into the **cid2tuple** table ;
5. Count the amount of retransmitted bytes per connection from the packets table and insert the result into the retransmission table ;

Step 1, copying packets into the **packets** table is done as follows : tcpdump is used to read the packet trace file and the output is piped through a filter program to the database. The filter program's primary task is to make sure that each line of text, i.e. each packet, is well-structured before uploading it into the database. More specifically, each line of text representing a TCP packet contains all the attributes defined in the packet table. If an attribute is missing, the filter program adds a special character signifying that its value is null.

The remaining four processing steps are performed with Structured Query Language (SQL) queries. It would be logical to have the retransmission data created in step 5 in the same table with the other connection level statistics created in step 3, but the need to use separate SQL queries to create these two sets of data forces us to use separate tables. The table **packets** does not contain the **4-tuple** attributes and, in fact, the reason for performing the processing step 4 is that we can drop the **4-tuple** attributes data from the packets table, which saves disk space because we only store the **4-tuple** twice per connection (both directions) instead of once for each packet.

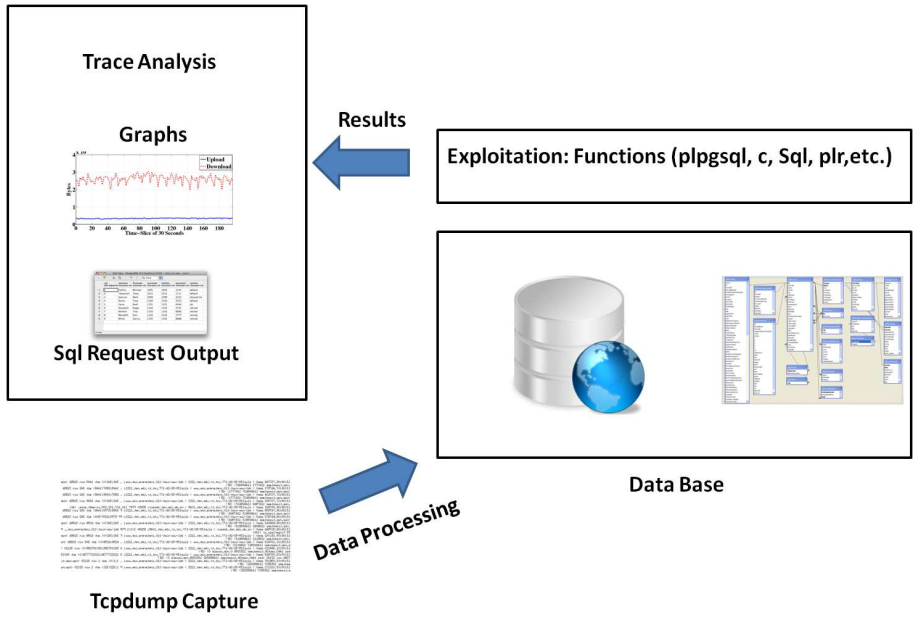


FIGURE 1.2 – Global Overview of InTrabase Processing

After the five processing steps the tables are populated with data from the packet trace and the user can either issue standard SQL queries or use a set of functions provided for more advanced querying on the uploaded data. Alternatively, the user may develop his own functions. User can not limit himself on connection level analysis but also drill down to per-packet analysis.

Intrabase offers the possibility to implement functions in procedural languages to perform operations that cannot be done with plain SQL queries. So to perform data analysis we developed

several functions to compute basic key performance indicators and complex ones to develop our fine grained TCP performance analysis presented next in this work. We present in Figure 1.2 the general schema of a tcpdump file processing within InTraBase. It is important to note that our main contribution in InTraBase is situated on the exploitation layer, since we developed several plpgsql functions to carry out our algorithms of TCP transfer time break-down and anomalies detection.

1.7 Overview of Datasets

We used, throughout the thesis, three different sets of traces. We provide an overview of these different sets in this section.

1.7.1 Heterogeneous Environments

Table C.1 summarizes the main characteristics of the packet level traces used in this work. These traces were collected from several different environments : the network of a DSL from an European ISP, a wireless hotspot in Portland and a research lab (Eurecom). Those traces are interesting because of their diversity in terms of access technology and also in terms of applications. For instance, p2p transfers are banned from the Eurecom network while it represents a large fraction of the bytes for the DSL trace. A wireless hotspot should differ from a DSL network in that users tend to focus more on interactive application in such environment and tend to refrain themselves from generating large transfers, e.g. applications updates or p2p transfers. As presented in Table C.1, these traces present several differences in their capture time and location, nature of traffic, as well as type of users selected. We detail in Section 2.2 the definition of well behaved TCP connections.

	Capture day	Duration connections	Nb connections	Well-behaved connections	Size in MB	Size in packets
ADSL	2005-05-31	1 min and 29 s	37790	5873	357.51	743683
Portland Hotspot	2007-09-14	2 h and 20 min	5051	3798	174.13	352569
Eurecom	2008-10-20	1 h and 1 min	32153	26837	1567.42	2867321

TABLE 1.1 – Heterogeneous Traces : Description

Table 1.2 reports some characteristics of the amount of flows, observed for mainly used destination and source ports. From presented statistics it is evident that applications using port 80 are more dominant, with for instance 42% for ADSL trace and 92% for Eurecom trace.

Our objective, with this first set of traces, was to apply our methodology of analysis of TCP connections, see Chapter 2, and obtain conclusions that were not bound to a specific location. Thus, while the ADSL trace is extremely short in terms of duration, it features enough connections to have a statistically sound analysis. However, the trace is not enough to characterize an ADSL trace. We also used longer traces obtained from Orange to carry more in depth analysis. We present them in the next paragraph.

	Total connexions	Port=80 Number of connections	Port !=80 Number of connections	Port=80 Mean data packets per connection	Port !=80 Mean data packets per connection
ADSL	5873	2496	3377	10.55	8.69
Portland Hotspot	3798	3504	294	18.27	121.95
Eurecom	26837	24855	1982	31.23	69.02

TABLE 1.2 – Port Distribution

1.7.2 Traces from Orange ISP

We study three packet level traces of end users traffic from a major French ISP involving different access technologies : ADSL, Cellular [2] and FTTH. ADSL and FTTH traces correspond to all the traffic of an ADSL and FTTH Point-of-Presence (PoP) respectively, while the Cellular trace is collected at a GGSN [3] level, which is the interface between the mobile network and the Internet. Table C.2 summarizes the main characteristics of each trace.

Note that measurements were performed at different time periods during the day to compare traffic stability and to get conclusions independent from a period of time or users behaviors.

As a consequence it is important to note the large variability and diversity of our considered data sets which is more accentuated with different users behaviors from one access network to an other, capture time and used services. For instance Cellular access should differ from FTTH and ADSL in terms of usage, because Cellular access users tend to a specific temporal usage as e-mail checking or web browsing ; We can expect further changes with the introduction of smart phones and the usage of 3G keys.

	Cellular	FTTH	ADSL
Date	2008-11-22	2008-09-30	2008-02-04
Starting Capture	13 :08 :27	18 :00 :01	14 :45 :02 :03
Duration	01 :39 :01	00 :37 :46	00 :59 :59
NB Connections	1772683	574295	594169
Well-behaved cnxs	1236253	353715	381297
Volume UP(GB)	11.2	51.3	4.4
Volume DOWN(GB)	50.6	74.9	16.4

TABLE 1.3 – Traces From a Major ISP : Description

In the present work, our focus is on applications on top of TCP, which carries the vast majority of bytes in our 3 traces, and close to 100% for the Cellular technology. We restrict our attention to the connections that correspond to presumably valid and complete transfers, that we term well-behaved connections. Well-behaved connections carry between 20 and 125 GB of traffic in our traces (see Table C.2).

2. Cellular corresponds to 2G and 3G/3G+ accesses as clients with 3G/3G+ subscriptions can be downgraded to 2G depending on the base station capability.

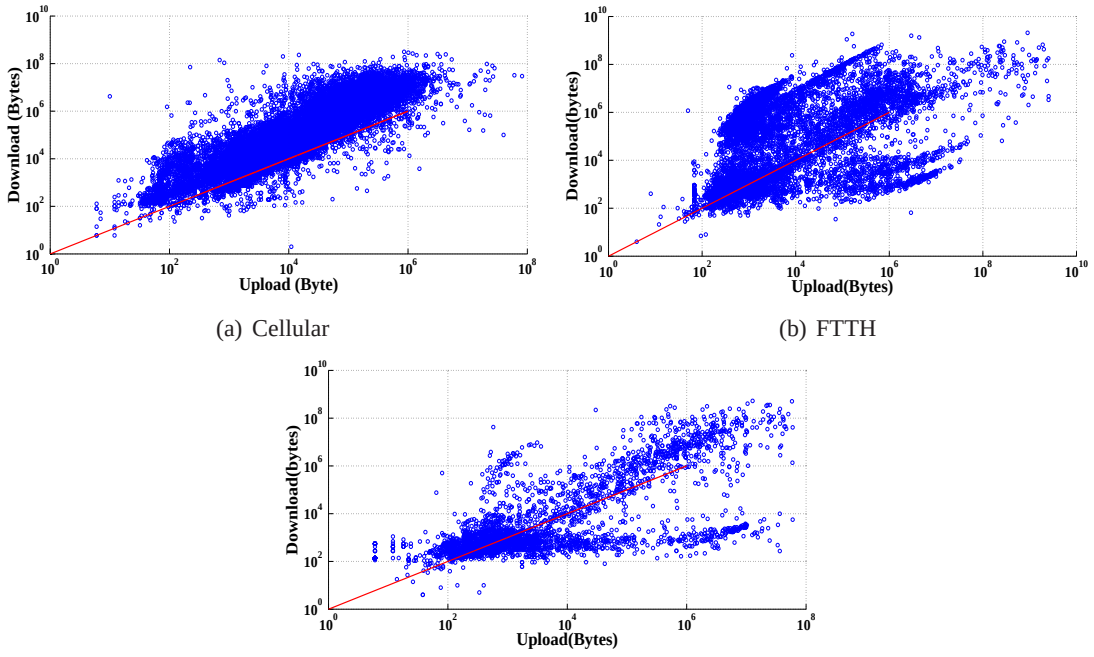
3. The Gateway GPRS Support Node (GGSN) is a main component of the GPRS network. The GGSN is responsible for the interworking between the GPRS network and external packet switched networks, like the Internet and X.25 networks.

1.7.2.1 Applications and Performance

Cellular		FTTH		ADSL	
TCP Port	Connection %	TCP Port	Connection %	TCP Port	Connection %
80	58.9	80	42.75	80	57.85
8080	17.12	443	2.38	443	4.4
443	7.7	25	2.52	135	4.38
110	2.4	6881	2.3	8080	3.98
143	2	30042	1.36	110	2.91
445	1.7	24350	1.07	445	2.29
993	1.63	51413	0.96	2000	1.41
5223	0.64	110	0.88	25	1.27
5001	0.62	26091	0.83	19898	1.03
995	0.4	4661	0.76	139	1
others	6.89	others	44.19	others	19.22

TABLE 1.4 – A First Classification

It is out of the scope of this work to precisely profile users' applications within the three traces we consider. We however performed a rough classification of traffic by identifying popular destination server's ports. Table 1.4 reveals that more than 84% of Cellular access connections targeted ports 80, 8080 and 443 unlike FTTH and ADSL with respectively 45% and 62%. Also, notice for ADSL and FTTH traces a large fraction of connections with non-trivial destination ports number, which symbolize p2p applications.



(a) Cellular

(b) FTTH

(c) ADSL

FIGURE 1.3 – Orange Client Traffic

We would need more sophisticated techniques to fully profile the applications active in our trace [69, 70]. However, those figures comply with intuitions : on Cellular access, a majority of traffic flows over HTTP (browsing, HTTP streaming, Webmail, etc) and on FTTH and DSL access,

HTTP tends to dominate again (at the expense of p2p transfers) with the rise of HTTP streaming, e.g., YouTube.

We next turn our attention to per client utilisation of available bandwidth. Figure 1.3 shows the scatter plot of the traffic volume uploaded and download per user for considered traces. We focus only on active client : i.e. clients having more than one data packet in each direction.

The main observation is that Cellular clients tend to download significantly more data than they upload. This is in contrast to wired networks usage profiles where one observes that a significant fraction of users upload large volumes of traffic because of p2p applications [69].

A fist explanation is that the usage of Cellular and wired networks are different. In fact for our case, the majority of Cellular connections are established using mobile phones, which mainly uses web browsing, streaming or video applications. Also, when users browse Internet pages, they tend in most cases to download data, unless they use an interactive application requiring uploading data than download like playing games or completing forms.

This result is also in line with the findings in [30] where the authors observe that the largest fraction in term of volume of HTTP over mobile devices is multimedia : watching video from Youtube, listening music from Itunes or downloading applications from Apple Store or Android Market, induce more download than upload traffic .

1.7.3 Enterprise Traffic

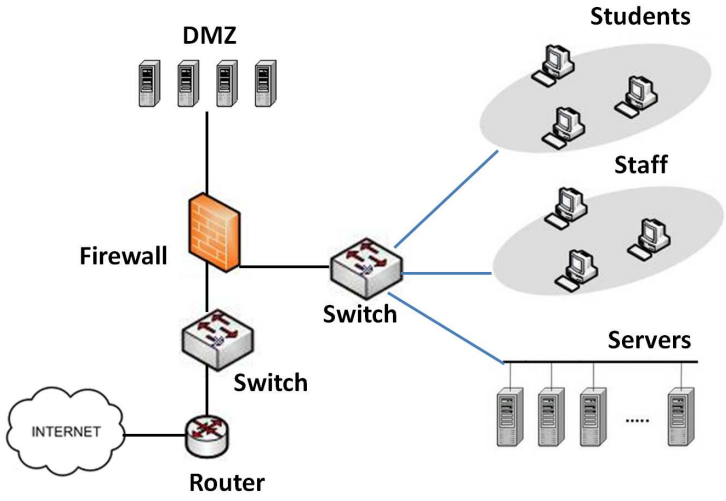


FIGURE 1.4 – Architecture of the Network

Our last set of traces consists of a single trace collected in a enterprise environment, thus consisting of a set of machines that might communicate either with internal servers or with machines on the Internet.

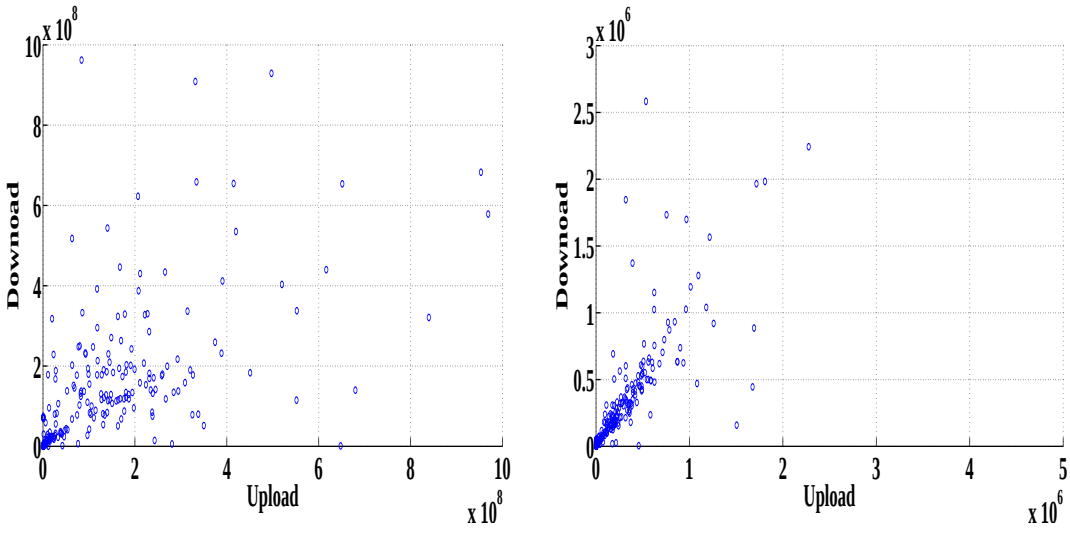
Figure C.1 presents a high level view of our network. This networking infrastructure, which consists of around 800 workstations equipped with a variety of operating systems. The network is organized into several Virtual Local Area Networks (VLANs) : servers, staff, DMZ, connected via a Cisco multilayer switch. We collected a trace of one day (in January 2010) long of all traffic flowing between the servers and the end users machines within the Eurecom network. We restrict our attention to TCP flows as they represent more than 97% of flows in each trace, and they carry over 99% of the bytes.

Table C.3 summarizes the main characteristics of the trace. The one day trace can be divided in several classes of traffic, according to the source and destination machines. As depicted in Table C.3 we can notice that client/server traffic dominates in terms of identified well behaved connections and exchanged data volumes. Interestingly exchanged data volumes are quite similar, except for the Demilitarized Zone (DMZ) traffic with more data volume for the upload.

	Server/DMZ	Client/Server	Server/Server
Well behaved connections	57348	128237	52333
Volume UP(GB)	8.581	127.061	76.290
Volume DOWN(GB)	6.651	114.054	76.365
Volume UP(data packets)	10798530	153704391	61114981
Volume DOWN(data packets)	9268532	145712454	61198436

TABLE 1.5 – Enterprise Trace : Description

1.7.3.1 Applications Break-Down



(a) Bytes (b) Data Packets

FIGURE 1.5 – Eurecom Client/Server Traffic

Figure 1.5 shows the scatter plot of the traffic volumes and data packets uploaded and download per user. We observe that enterprise clients tend to generate the same amount of data volume for upload and download. This significantly differs from already presented traces from Internet environments.

In the previous paragraph we have noticed that client/server traffic, e.g., HTTP ; represents the highest number of TCP connections and the larger volume of data exchanged. In Table 1.6 we report ten most popular targeted destination ports on the server side, in terms of number of connections, exchanged data volume and data packets.We observe the presence of new protocols as compared to legacy Internet traffic, which are typical of enterprise environements, e.g. Server Message Block (SMB).

As can be seen from Table 1.6, we notice that the Symantec Endpoint Protection Manager (SEPM) generates the highest number of TCP connections, but not the largest exchanged data volume. With clearly less TCP connections, Lightweight Directory Access Protocol Over SSL (LDAPS) and Network File System (NFS) generate more data volume.

Server Port	NB cnxs	UP MB	Down MB	Datapkts	Down Datapkts
Symantec SEP (8014)	114130	88.95	2,288.08	154155	1727323
SMB (445)	20679	17,448.04	30,031.54	54534755	63508706
LDAPS (636)	19186	72.94	401.09	411181	1888158
LDAP (389)	13886	42.23	186.62	85347	278375
Windows RPC (1025)	10524	21.066	9.30	56299	50626
epmap (135)	9208	3.69	3.39	29506	29576
Http (80)	8189	284.95	766.48	437805	849342
sunrpc (111)	8015	0.78	0.30	8042	8039
Https (443)	6604	36.95	31.21	103107	70526
nfs/shilp (2049)	4174	36,756.97	17,795.81	34536623	21141082

TABLE 1.6 – Eurecom Traffic Overview

1.8 Conclusion

In this Chapter we presented main research works in relation with our scope of TCP performance analysis for the case of Internet and enterprise traffics. We focused on short TCP transfers, since they represent the large amount of Internet connections. Next, we revisited relevant works in TCP performance analysis of (i) Internet wired and wireless accesses (ii) and enterprise traffic. We show that main works in TCP anomaly detection focus on security and attack aspects, and neglect the impact of new applications and remote server impacts in throughput limitation.

We then presented InTraBase - the traffic analysis tool, used to manipulate the traces and to implement the algorithms we developed.

Based on a DBMS approach it allows to manage collected dumps in order to processes the input tcpdump file as little as possible prior to loading it into the database. Finally, we summarized the main characteristics of the traces captured at the packet level, collected in heterogeneous wireless and wired environments, used to carry out our traffic analysis.

In the next Chapter, we revisit the performance of TCP transfers especially for short TCP transfers. Then we present an overview of our data time break-down methodology in order to focus on the impact of application on top of TCP.

Chapter 2

Revisiting the Performance of TCP Transfers

2.1 Introduction

In this Chapter, we highlight the interplay between TCP and the application on top. We first discuss the definition commonly made of short TCP transfers - transfers that cannot rely on the FR/R mechanism - with the emergence of new mechanisms to improve the performance of small transfers, e.g. *limited transmit*.

Our main contribution is to present an overview of the impact of the application, on the TCP transfers. We show that while losses can have a detrimental impact on short TCP transfers, the application significantly affects the transfer time of almost all short - and even long - flows in a variety of way. Indeed, the application can induce extremely large tear-down times and it can also slow the rate of actual TCP transfers.

In addition, the application can worsen the impact of losses by preventing TCP from sending large enough bursts of packets. We adopt an application agnostic approach, i.e., we do not make any assumption on the way the application is working, to develop a set of techniques that delineate the impact of the application from other causes that explain a given transfer duration, including the data transfer itself and the recovery time if any.

We illustrate our findings with the set of traces described in Section 1.7.1, which includes DSL, wireless hotspot and a research lab traffic.

2.2 Well-Behaved Connections

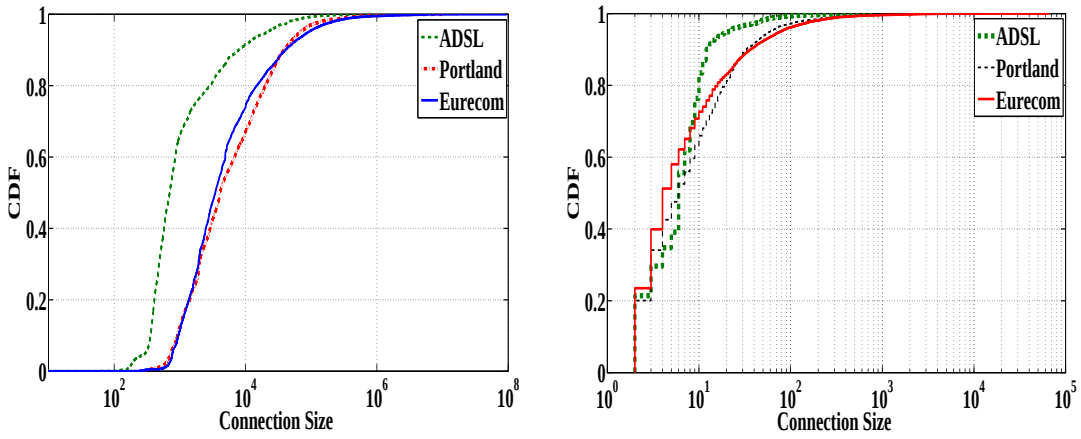
While analyzing the performance of TCP transfers, we focused on the connections that correspond to valid and complete transfers from the TCP perspective. Specifically, well-behaved TCP connections must fulfill the following conditions : (i) A complete three-way handshake ; (ii) At least one TCP data segment in each direction ; (iii) The connection must finish either with a FIN or RESET flag.

When applying the above heuristics for our traces, we are left with a total of over 35,000 TCP connections when summing over the three traces. The DSL trace is the one offering the smallest fraction of well-behaved connections, 5873 over 37,790, because of a large number of unidirectional transfers (SYN without a reply). The short duration of the trace also impacts this value as for a lot of cases, we do not observe the beginning or the end (or both) of the connection.

P2p applications tend to generate such abnormal connections (contacting a non available p2p server to download a content) as well as malicious activities.

Figure 2.1 depicts the cumulative distribution of well-behaved connection size using bytes and data packets for the 3 traces. We observe that the Eurecom and Portland traces offer a similar connection profile that significantly differs from the DSL trace. For instance, 65% of the DSL connections are less than 1 Kbytes and 25% are between 1 Kbytes and 1 Mbytes, unlike Portland and Eurecom traffic which offers larger values at similar connection percentiles. A reason behind this observation, again, is the small duration of the DSL trace. However, our focus is on short transfers, and from this perspective, the DSL trace offers valuable information. While Eurecom and Portland traces present different types of traffic (wired and wireless), they have roughly the same cumulative distribution of bytes. Secondly, considering the cumulative distribution of connection size in terms of data packets, we observe that the traces present the same shape until transfer size of 10 data packets. After this value, the DSL trace increases faster to reach 95% of connection for less than 20 data packets.

When focusing on the performance of TCP transfers, the number of data packets to be transfered is a key element to consider, as it impact the ability of TCP to recover using the Fast Retransmit/Recovery mechanism. We can already observe from Figure 2.1 that irrespectively of the trace, a significant portion of connections (between 53% and 65%) have less than 7 data packets.



(a) Size in Bytes (b) Size in Packets

FIGURE 2.1 – Trace Characteristics

2.3 Short Transfers : Definition

In this section we introduce a first definition of a short TCP connection, which is commonly used in the literature.

A short TCP connection is a well behaved connection unable to perform fast retransmit/recovery (FR/R), after a packet loss detection.

While simple, the above definition does not lead to a unique threshold value in terms of number of data packets for a short TCP transfer. Indeed, various TCP implementations and connection

characteristics can affect this definition : the initial congestion window, the use of delayed ACK, the number of duplicate acks that triggers a FR/R.

For instance, Windows Vista implements Limited Transmit, which means that only 2 duplicate ACKs are enough to trigger a fast retransmit. We estimated for the 3 traces, the number of segments observed in a duration equal to one RTT after the sending of the first data packet, and this for each direction - see Table 2.1. The obtained value provides a lower bound on the initial congestion window that the transport uses as the application may not provide TCP with enough data to send at the beginning of the transfer. This is especially true for the initiator side in the case of Web transfer where the GET request might fit in a single data packet. Overall, we observe that values of 1 and 2 MSS (and possibly higher values) seem to be common initial congestion windows. Initial congestion windows larger than 2 MSS (we observed values up to 12 MSS) might be due to specific optimizations of operating systems that cache TCP level variables of previous transfers for a few minutes [71].

Trace	Initiator			Remote party		
	1 pkt	2 pkts	$>$ 2 pkts	1 pkt	2 pkts	$>$ 2 pkts
DSL	99%	1%	0%	80%	18%	2%
Portland	82%	17%	1%	64%	24%	2%
Eurecom	90%	10%	0%	65%	24%	1%

TABLE 2.1 – Estimated Initial Congestion Window

Given the estimated initial congestion window of Table 2.1, we report in Table 2.2 the main scenarios we focus on to find the threshold in terms of number of data packets that triggers a FR/R. A short connection is thus, for each scenario, one with a number of packets strictly smaller than the threshold. Those scenarios cover, to the best of our knowledge, all the most commonly encountered cases.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Initial cwnd	1	1	2	2
Delayed ACK	no	yes	yes	yes
Duplicate ACK	3	3	3	2
Minimum connection size (data packets)	7	9	8	7

TABLE 2.2 – Minimum Connection Size to Perform Fast Retransmit/Recovery

Based on the results presented in Table 2.2, we observe that :

– Different scenarios lead to different thresholds, from 7 to 9 data packets ;
– A connection size with less than 7 data packets can not recover from packet loss using FR/R, whatever the exact scenario is ;
– When considering a given scenario and a connection whose size is one packet larger than the threshold, we observe that this connection is able to perform a FR/R for only a single packet in its last round. The loss of any other packet will lead to timeout. A connection is thus not always able to perform FR/R if it is larger than the threshold.

Based on the result obtained from this section, we adopt a first definition of a short TCP transfer, as a connection of size less than 7 data packets. This definition, while simple, relies on

the implicit hypothesis that the application on top of TCP does not impact the way TCP sends packets. As we will see in Section 2.5, this assumption can be too strong in practice, as even long TCP transfers can be divided into short bursts (due to the application on top) that prevent TCP from relying on FR/R in case of losses.

2.4 Transfer Time Break-Down

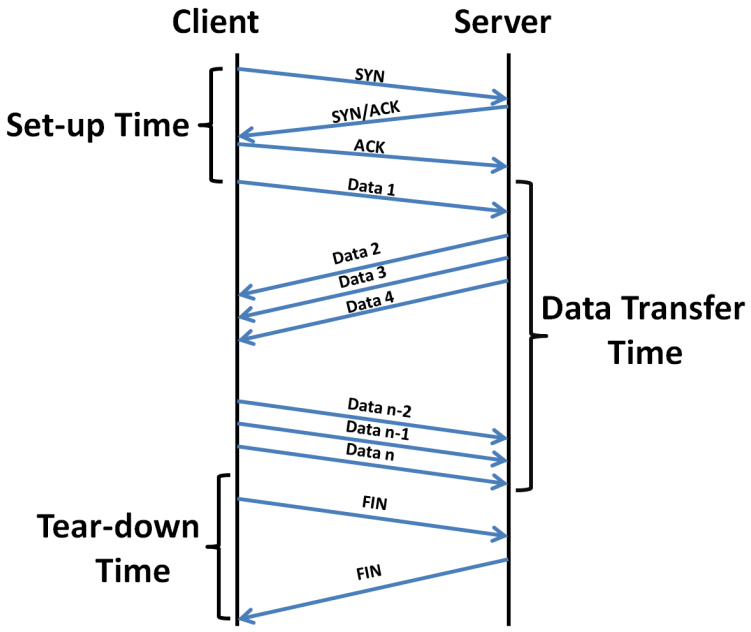


FIGURE 2.2 – Transfer Time Break-Down

To understand the factors that affect the performance of TCP transfers, we rely on the following decomposition in Figure 2.2 of each transfer into 3 different phases :

Set-up time : this is the time between the first control packet and the first data packet. Since we consider only transfers which have a complete three-way handshake, the first packet is a SYN packet while the last one is a pure ACK in general. The connection set-up time is highly correlated to the RTT of the connection. For the three traces we consider, we have a correlation coefficient of 70% for the DSL trace, 60% for the Portland trace, and 39% for the Eurecom trace.

Data transfer time : this is the time between the first and the last data packet observed in the connection. Note that it includes loss recovery durations, if any.

Tear-down time : this is the time between the last data packet and the last control packet of the connection. We impose, as explained in Section 2.2, that at least one FIN or one RESET be observed, but there can be multiple combinations of those flags at the end of the transfer. Unlike set-up, tear down is not only a function of the RTT of the connection, but also a function of the application on top of TCP. For instance, the default setting of an Apache Web server is to allow persistent connection but with a keep alive timer of 15 seconds, which means that if the user does not post a new GET request after 15 seconds, the connection is closed. A consequence of the rela-

tion between the tear-down time and the application is a weak correlation between tear-down times and RTT in our traces : 40% for the DSL trace (which is still quite high), 0.7% for the Portland trace, and -2% for the Eurecom trace.

Using the above decomposition, we analyze next, the impact of losses (Section 2.4.1) and of the application (Section 2.5) on the data transfer time.

2.4.1 Recovery and Tear-down

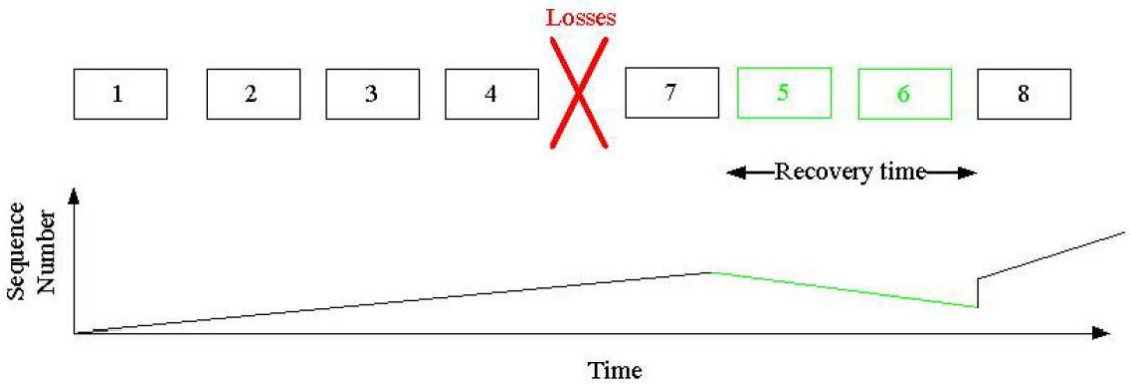
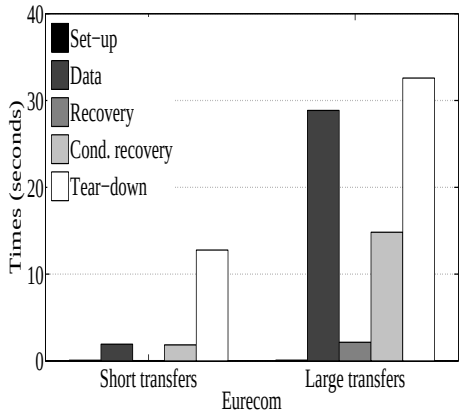


FIGURE 2.3 – Recovery Time

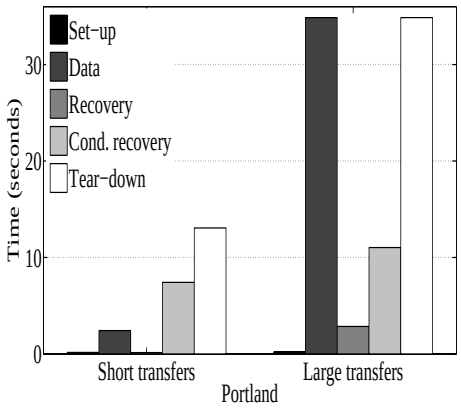
As explained above, the data transfer time possibly includes loss events. We estimate the time spent by TCP in recovering from losses using the *recovery time*. Specifically, for a given transfer, each time the sequence number in the stream of data packet decreases, we record the duration between this event and the observation of the first data packet whose sequence number is larger than the largest observed sequence number seen so far. For instance, we present in Figure 2.3 an example of a TCP connection suffering from data packet loss. Assuming that we associate a unique sequence number to each packet, if we observe the sequence 1,2,3,4,7,6,5,6,8, we will record the duration between packet 7 and packet 8. This duration is added to the *recovery time* of the transfer. To filter out reorderings that occur at the network layer, we discard each recovery time smaller than one RTT. Rewaskar et al. [72] developed algorithms to assess whether an observed loss event can be attributed to a time-out or a FR/R. We were not able to use this technique as it requires to perform a passive OS finger printing of the sender of the data. However, in our traces, most losses occurred in the data stream issued by the remote party and not the local clients. While p0f (`http://lcamtuf.coredump.cx/p0f.shtml`), which is recommended in [72], is effective when used on SYN packets, it fails when working on SYN/ACK packets, which limits the applicability of the techniques proposed in [72].

Figure 2.4 presents the break-down of the small and large TCP transfers for the three traces. We first observe from Figure 2.4 that while set-up durations are consistently small for all traces and transfer sizes, tear-down take very high values, between 2.5 and 27.5 seconds on average. The tear-down phase in itself often represents the majority of the connection time. Note however, that the tear-down time should have no impact on the performance perceived from the application on top as the data transfer is completed.

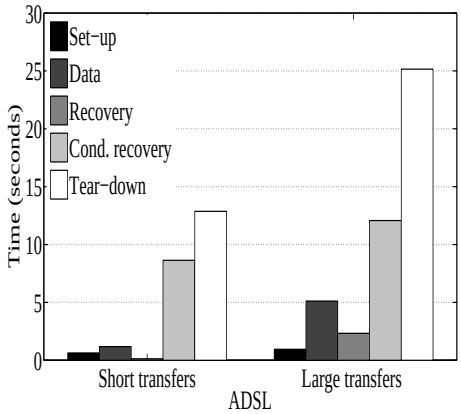
As for losses, we present two distinct values for the recovery time : the average conditional recovery time and the average recovery time. The latter is computed over all transfers of the category while the former is computed only for the transfers that experience at least one recovery



(a) Eurecom



(b) Portland



(c) ADSL

FIGURE 2.4 – Transfer Time Break-Down

event. Since only a small fraction of the transfers experience losses (9.4% for DSL trace, 13.2% for Portland and 6.8% for Eurecom), the average conditional recovery time is often much larger than the average transfer time. This impact is clearly more pronounced for small than for large flows, over the three traces, most probably because of the predominance of time-outs for short transfers.

Still, from a server point of view that is serving a large number of clients simultaneously, like a Web server, long tear down times can affect the service quality if a limit is set on the number of active clients. A side effect from those large tear-down values is when one estimates the throughput of transfers. If one divides the total number of data bytes by the total duration, one can greatly underestimate the actual throughputs perceived by the user and the application. Figure 2.5 depicts, for the case of the Eurecom trace, the throughput computed when considering the total connection time and the throughput computed when one considers only the set-up and data transfer times. We term the latter ”application-level” throughput (it is labelled AL in the graph), as this is the rate at which data are sent or received from the application perspective. Figure 2.5 shows a significant difference between both compared metrics for short and large transfers.

The main conclusion from the above study is that losses occur rarely, but have a highly detrimental effect. A second take-away is that the tear-down time should be removed when computing the throughput of a transfer as it can lead to a dramatic underestimation of the throughput percei-

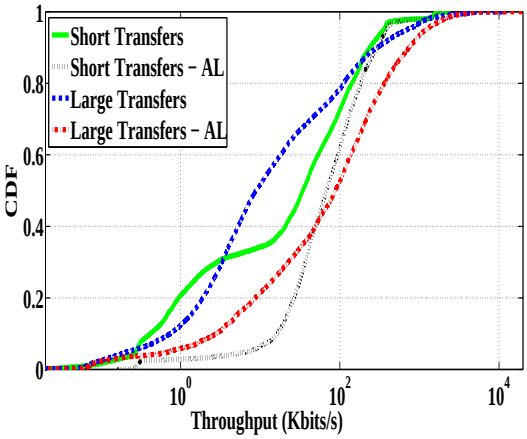


FIGURE 2.5 – Throughput and Application Level Throughput for Eurecom Trace

ved at the application level. For the case of the Eurecom trace, the median throughput of small (resp. large) transfers obtained when considering the tear down is 34 kbits/s (resp. 8.7), while it is 67 kbits/s (resp. 88) when tear-down is discounted.

2.5 Application Impact

In this section, we are interested in assessing the impact of the application on the transfer time of a TCP connection. There are many ways by which the application can influence the pace at which data flows in a network. First, the user might be involved in the transfer, as the case in a persistent HTTP connection, where the download of a new page is triggered by an HTTP Get message issued by the client browser. Second, the application might cap the rate at which information is sent to the TCP layer. This is typically what p2p applications do to limit the congestion on the uplink of the user. A third possibility is when the generation of data is done online. For instance, when querying Google for a specific keyword, several tens of machines are involved in this operation.

From the above discussion, we observe that the application may affect the transfer of data in many different ways. A first simple assessment that can be made to infer the impact of the application on a TCP transfer is to compute the fraction of packets with PUSH flags [73]. The PUSH flag is a way for the application to specify that it has no more bytes to send at the moment and the current segment can be sent. We plot in Figure 2.6 the ratio of PUSH flags as a function of the transfer size for the three traces. We observe that the impact of the application as captured by the PUSH flags decreases with increasing transfer size. For the short connections, the push flag ratio is extremely high, between 74% and 86%.

In the next sections, we assess in more details the way the application influences the transfer time. We show that the application tends to fragment the transfer in small flights of packets that prevent TCP from relying on FR/R in cases of losses.

2.6 Synchronism and Losses

For client/server applications, one often observes that even if the server is sending a large amount of bytes/packets, the actual exchange is fragmented : the server sends a few packets (hereafter called a train of packets), then waits for the client to post another request and then sends its

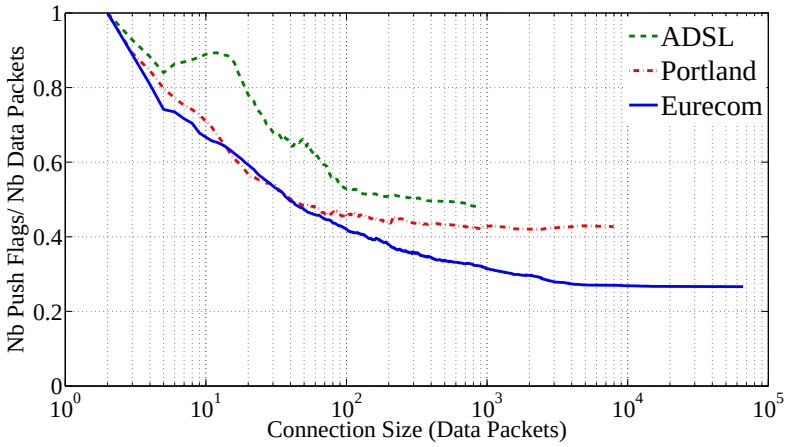


FIGURE 2.6 – Conditional Ratio of Push Flags

next answer. If such a behavior is predominant in TCP transfers, it can have a detrimental impact if ever the train size is too small as it might prevent TCP from performing FR/R in cases of losses.

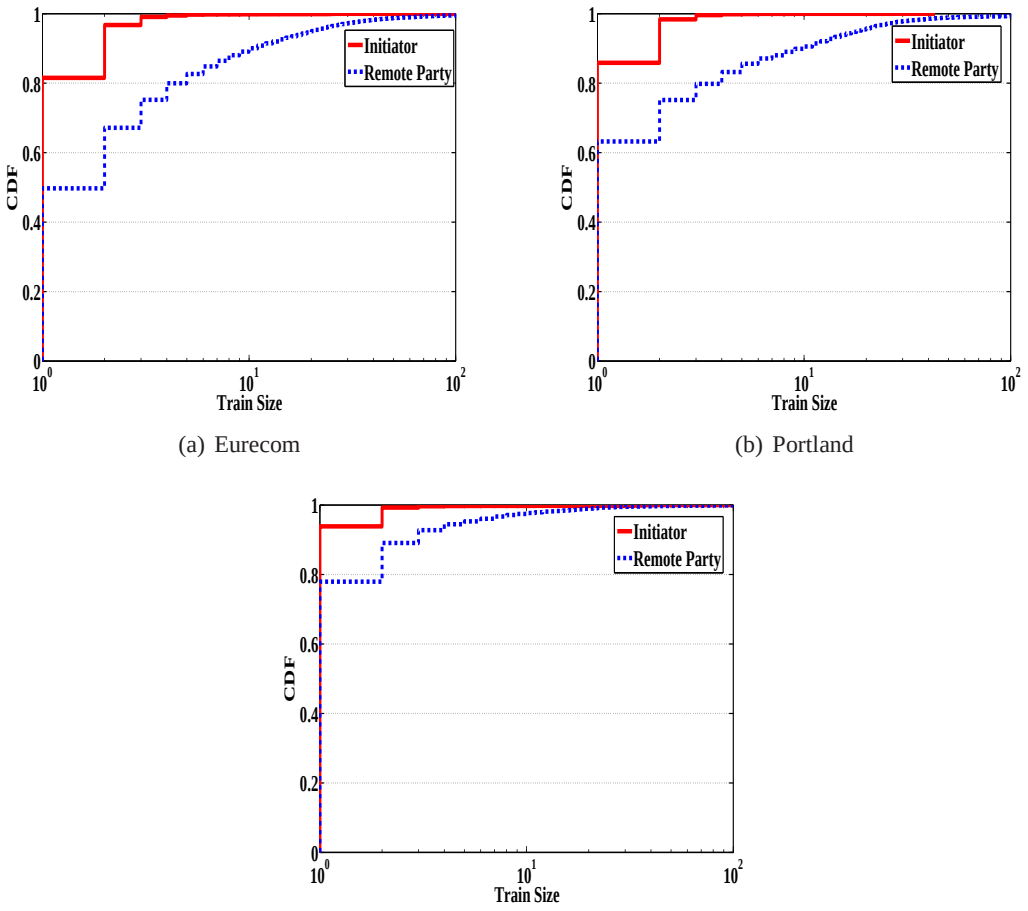
When we observe passively a connection, we see data flowing in both directions, i.e., each direction sends in turn a train of packets. This is not necessarily harmful if the two parties are not synchronized, i.e. if one party does not need to receive packets from the other party before sending its next train of packets. However, we observed that the two parties are apparently most of the time synchronised, i.e. that they have to wait for a signal from the other side before sending their next train of packets.

The question we raise is thus : are the two parties involved in a transfer synchronized or not ? Proving synchronism requires an a priori knowledge of the application semantics. We can however prove that the synchronism hypothesis cannot be rejected as follows : for a given transfer, each time we observe a transition from one side sending packets, say A, to the other side sending packets, say B, we observe if the first packet from B acknowledges the reception of the last packet from A. If this is not the case, then there is no synchronism, otherwise, synchronism can not be rejected. Applying this methodology to the three traces, we obtained that for each trace, the fraction of connections for which synchronism could not be rejected was extremely high : 88.6% for the ADSL trace, 94.4% for the Portland trace and 95.3% for the Eurecom trace.

For the connections for which synchronism could not be rejected, we looked at the distribution of the size of the trains of packets sent. We distinguished between the initiator of the connection and the remote party, as we expect the latter to be some kind of server that usually sends larger amount of packets than the former that simply posts requests. As illustrated by Figure 2.7 :

- Trains size sent by the remote part are larger than those sent by the initiator, in line with our hypothesis that the remote party be a server ;
- More than 97% of initiator trains are less than 3 data packets, which leaves TCP unable to trigger any Fast Retransmit, even if Limited Transmit is used ;
- More than 75% of remote party trains are less than 3 data packets, which again leaves TCP unable to trigger the fast recovery/retransmit, even if limited transmit is used.

Taking a broader perspective, the fraction of connections that have a maximum train size of 3 packets is 85.2% for the DSL trace, 40.5% for the Portland trace and 54% for the Eurecom trace. Sizes of those connections remain quite in line with our definition of Section 2.3 We observe for our traces that over 97% of those connections have less than 20 packets.



(a) Eurecom (b) Portland

(c) ADSL

FIGURE 2.7 – Cumulative Distribution of Transmitted Bloc Size

2.7 Conclusion

We analyzed in this chapter the performance limitations of short and interactive TCP transfers, for heterogeneous traffic traces. Short transfers sending less than seven packets are not able to apply Fast Retransmit. Thus, they are really sensitive to loss events in the network. These short transfers represent the majority of transfers. We have also observed very long tear-down delays, between the last data packet of the connection and the last control packet. This tear-down delay does not influence the user perception, but it may affect the measurement of response times of short transfers in network management functions.

The sensitivity to loss concerns also many long transfers as many of them are a sequence of alternate exchanges and the vast majority of these bursts are less than 3 packets. Such a feature has a direct influence on the ability of TCP to recover from a loss using Fast Retransmit.

In the next Chapter, we highlight that measurements from passively collected traces can be biased by specific technologies implemented in Cellular networks to boost performance and control users activity. Also, we cast a first look to two key Internet services : mail and webmail in order to identify factors that lead to different perceived performance for the case of Cellular users.

Chapter 3

Profiling Cellular Applications

3.1 Introduction

In the previous chapter, we studied TCP performance in general without paying attention to the specific features of a given access technology. In contrast in this chapter, we focus on a specific technology, 3G access, to highlight some of the difficulties encountered when profiling its traffic.

In this Chapter, we present observations from a passive packet level trace with more than 1.7M TCP connections, collected at the access network of a major European ISP. Our study includes different classes of access : 3G, EDGE and 2.5G connections. A given user can be observed using any of these technologies as Cellular contracts work in a best effort manner : the client is granted a 3G access whenever it is available at the base station to which it is connected ; or downgraded to former technologies, EDGE or 2.5G, if 3G is not available. We further observe a diversity related to users devices, e.g., mobile phones and Universal Serial Bus (USB) pluggable 3G modems.

We study the performance of Cellular access network and we bring to light phenomena introduced by Cellular core [1] network equipments, which can bias measurements. As a second step we investigate the performance of Cellular networks, focusing on two key services : mail and webmail.

Mail and webmail are a key applications from the end user point of view and while most of work has focused on trendy applications, e.g., p2p, streaming or social networks, mail has received little attention.

3.2 Impact of Core Network Equipments

In this section, we highlight that in modern Cellular networks, estimating latency turns out to be a complex task. Indeed, we demonstrate that latency can be under estimated due to the use of new mechanisms or services, like proxies for content adaptation or applications acceleration. We investigate how these mechanisms impact our measurements and the performance perceived by end users.

3.2.1 RTT Estimation

The round trip time corresponds to the spent time between a sender transmitting a segment and the reception of its corresponding acknowledgement. This interval includes propagation, queuing, and other delays at routers and end hosts [74].

1. Core relates here to the wired part of the ISP network that enables access of 2G/3G clients to the Internet.

Several approaches have been proposed to accurately estimate the RTT from a single measurement point [75, 76, 77, 3]. To estimate RTT, we adopted two techniques. The first method is based on the observation of the TCP 3-way handshake [76] : one first computes the time interval between the SYN and the SYN-ACK segment, and adds to the latter the time interval between the SYN-ACK and its corresponding ACK. It is important to note that we take losses into account in our analysis. The second method is similar but applied to TCP data and acknowledgement segments transfered in each direction [2]. One then takes the minimum over all samples as an estimate of the RTT.

Due to the location of the probe within the network of the ISP (see Section 1.7.2), we are able to distinguish between a local and a remote RTT. The local RTT is measured within the access network, including the wireless link, of the ISP, while the remote RTT factors both the latency over the path from inside the network ISP to the first peering link and then to the remote server.

3.2.1.1 Impact of Active Devices

While analyzing modern Cellular networks, we face a double difficulty : (i) the access technology can vary (from 2G to 3G) from one user to the other and over times and (ii) the capabilities of the device itself varies from one device to the other, which sometimes prevents the user from accessing all types of Internet applications. In the network we analyze, devices with limited display capability are serviced by a specific device. Redirection to this specific device is achieved at the mobile client using Access Point Name (APN). It can be seen as the equivalent of a dial-up phone number of an ISP. For convenience, we term those connections APN transfers below.

An Access Point Name is a specific network to which a mobile can be connected. It corresponds to the name of an external network that is accessible from a terminal [78]. In practice, the Subscriber Identity Module (SIM) card of the end user terminal is configured with the IP address of the APN of the Service Provider. It provides routing information for Serving GPRS Support Nodes (SGSN) and GGSN. In the 3rd Generation Partnership Project (3GPP), content billing of GPRS activity is generated based on APN accounting features.

In our Cellular trace, we have more than 17% of APN transfers, which can be identified as targeting a specific private IP address and port 8080.

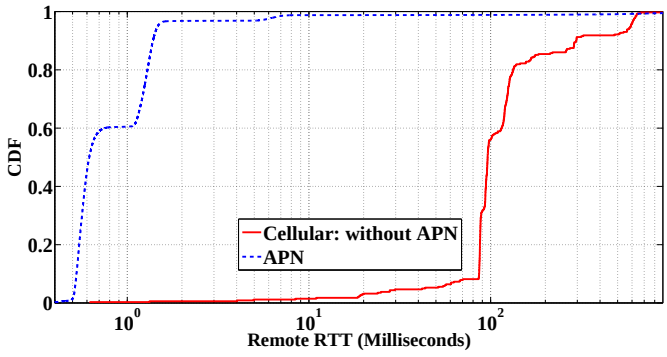


FIGURE 3.1 – Remote RTT of APN Transfers

We compared the RTT of APN and non APN transfers. For the latter ones, we restrict our attention to connections targeting port 8080, to have a somewhat comparable basis (though it is still quite arbitrary). In Figure 3.1, we compare remote RTTs – the two RTT estimation methods gave similar results – for the two types of transfers. We notice a difference of about 100 ms between

2. Keep in mind that we focus on well-behaved transfers for which there is at least one data packet in each direction.

the two types of traffic, which is explained by the split mode used at the device, which adapts the content for those limited capacity devices.

We now restrict our attention to non APN transfers. These transfers are characterized in our trace by a straightforward manner : they have a public remote IP address. Still, the devices that generate this type of traffic do not necessarily communicate directly with the remote server. The ISP is using a set of devices for user authentication (Radius), Network Address Translation (NAT) (as we find also in wired networks) and a proxy (a specificity of Cellular networks) whose main objective is to boost performance of the initial phases of TCP transfers. This proxy intercepts the first SYN of new connections and responds on behalf of the remote server, with a SYN-ACK, while in parallel, the initial SYN is forwarded to the remote server. The proxy later applies various tricks to (try to) improve the performance of TCP transfers. The way the proxy works at connection establishment leads to a significant discrepancy between the two methods we use to compute the RTT, which often reaches 100 ms, as can be seen from Figure 3.2.

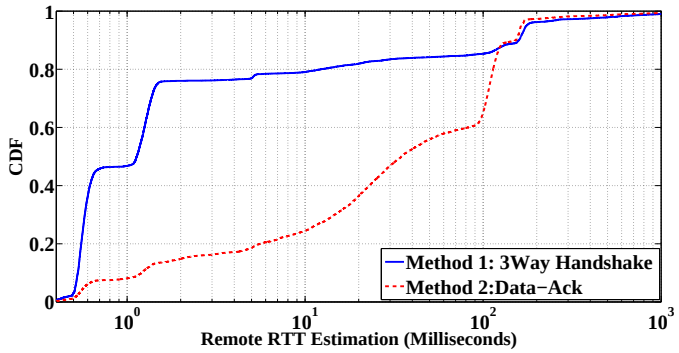


FIGURE 3.2 – Proxy Impact for Latency Estimation

The key message from this section is that several specific devices might affect classical performance metrics in Cellular networks, which should be taken into account when performing measurement studies. In the rest of analysis of Cellular traffic, we focus only on non APN traffic and our estimation for latency will be based on the DATA-ACK method only.

3.3 Mail and Webmail : Characteristics and Usage

The E-mail service is often overlooked in traffic analysis studies, even though it represents a key service for end users that use it on a daily basis. Traditional works in traffic measurement, usually, study the performance of p2p applications, streaming and more recently the impact of social networks [33, 79, 80]. The shortcoming of such studies is that they neglect mail and webmail impact, even though they represent one of the most popular Internet application [81, 82] and millions of Internet users use them several times per day for professional or personal usage.

In this section, we detail how mail and webmail traffic is extracted from the trace and evaluate the popularity of mail and webmail usage. We further extract pieces of information related to the popularity of the different service providers and end user devices for the case of webmail, taking advantage of the fact that the HTTP protocol exposes some key information.

Therefore, during our next analysis we will distinguish between webmail and mail performances in order to evaluate the main features that characterize each service and in next level to respond to the question : Why users prefer one service at the expense of the other ?

3.3.1 Service Identification

Internet traffic classification is an area that attracted a lot of attention recently [83, 84, 85]. In most cases, mail traffic is classified based on the legacy mail protocols : IMAP, POP3 and SMTP. Concerning webmail, the following identification techniques have been proposed :

1) Map the destination IP address with a list of URLs of popular webmail providers [86] 2) Combine the previous method of URL matching with keyword matching (based on unique keywords that appear in the packets payload that can identify webmail traffic) [81] 3) Use statistical methods [82].

In this paragraph, we adopted the second approach to detect webmail traffic : we first extract HTTP requests with webmail key words, then we identified the connections corresponding to these requests. To identify mail traffic for the upload and download, we use TCP port numbers and remote address resolution.

3.3.2 Usage and Popularity

Using the detection method presented in the previous paragraph, we extracted mail and webmail traffic from our trace. It turns out in our trace that mail and webmail represent about 5% of all flows and 17% of overall traffic volume.

Tables 3.1 and 3.2 summarize characteristics of mail and webmail connections, including number of connections, volumes uploaded and downloaded in terms of total amount of bytes at the IP layer and in terms of data packets, number of servers, and number of clients.

Concerning mail (see Table 3.1), we observe that Post Office Protocol version 3 (POP3) and POP Secure (POPS) dominate downloads while Simple Mail Transfer Protocol (SMTP) and SMTP Secure (SMTPS), obviously, dominate uploads. Internet Message Access Protocol (IMAP) and IMAP Secure are the most popular service in terms of number of established TCP connections, followed by POP3/POPS and finally SMTP/SMTPS. The smaller number of mails uploaded as compared to mails downloaded is likely to be due to the limited capabilities of devices (most of them are smart phones and not PC with USB pluggable 3G modems as we will see soon) as compared to legacy wired access with desktops and laptops, which feature convenient displays and also store data that can be used as attachments, as opposed to smart phones in general [3].

	SMTP/SMTPS	POP3/POPS	IMAP/IMAPS
Nb cnxs	7330	51202	64493
Upload (MB)	116.1	5.7	59.8
Download (MB)	1.2	1741.6	853.8
Upload (Data Pkts)	78631	261828	731616
Download (Data Pkts)	10130	1523961	1270844
Nb Servers	360	1578	883

TABLE 3.1 – Mail Traffic Characteristics

Table 3.2 shows general information about webmail traffic in our trace. We observe similar results as for mail : users tend to download more than they upload. We hypothesize again that it is a result of the limited capacities of devices in general. Our point of view was that this trend is strongly correlated with mobile devices limitation, because until now, not all existing devices offer

3. Our experience with wired traces of DSL and FTTH accesses shows that email traffic is also asymmetric in wired accesses, e.g., because of mailing lists and wanted or unwanted advertisements ; but the extent of asymmetry is far smaller than in Cellular networks.

the possibilities to send mails with one or more attached documents. Hence, Cellular user, using their mobile devices, tend to write short mails without attached files.

	Cellular
Nb Cnxs	16275
Upload (MB)	1364.4
Download (MB)	7169.8
Upload (Data Pkts)	1712270
Download (Data Pkts)	2022705
Nb Servers	528

TABLE 3.2 – Webmail Traffic Characteristics

Comparing mail and webmail volume statistics, we observe that webmail is much more popular than classical mail. Concerning connection sizes, a number of webmail connections are smaller in size as compared to mail transfers.

At this stage a natural question is : Why Cellular users tend to use more webmail than classical mail ? We can envisage several options :

1) Cellular users prefer webmail at the expense of legacy mail 2) Webmail services offer in general better performance than mail 3) Cellular devices are more adapted to webmail usage.

Option 1 stems from two intuitions. First, the natural intuition that the complexity of configuring a POP/IMAP client as compared to using a Webmail access is a barrier for a lot of users. Second, the intuition that users prefer to have a mail account from a mail service provider that is not their network provider in order to keep the account even if the network provider changes. Though mail service providers offer in general POP/IMAP interfaces, Web based interfaces, i.e., webmail, is by far the most popular way to reach those services.

We gathered also statistics on webmail servers, client devices and their OSs, and clients browsers, taking advantage of the presence of many key information in the HTTP fields. Figure 3.3 reports the percentage of transfers per webmail service providers (for the most popular ones). We observe a dominance of Hotmail, Gmail and Yahoo. Only after we find webmail services offered by network providers like Orange, Tele2 and Alice. These results show that webmail service providers that propose free mail boxes are much more popular than the corresponding services offered by network providers. The latter means that hypothesis 1) mentioned above plays a role in the higher popularity of webmail at the expense of traditional mail.

Let us now focus on devices and their Operating Systems (OSes). Figure 3.4 shows that among the currently popular devices and OSs, we find iPhone at the first position followed by Microsoft OSs (Vista, XP and CE). MacOS, Linux and other mobile devices remain marginal in our data set. The above result was obtained for clients using webmail and not for clients using mail, as we have no access to similar information in the latter case. We can however conjecture that the trends (OS shares) for these other clients be similar. More generally, the above observation is in line with current market trends that shows that, at least in France, the Iphone is the dominating smart phone at the moment. The small fraction of OSes of laptops suggests that devices connected with USB pluggable 3G modems are still marginal in the Cellular network we study.

Operating systems and Web browser can impact network performance through several parameters and especially the number of connections established to the Web server they connect to. To assess if there was significant difference in the strategy used by the different OS/device in our data set, we report in Table 3.3 the mean numbers of connections per webmail session. A session consists of all the connections between a specific pair of client and server IP addresses in our trace. The results in Table 3.3 suggest a similar behavior for the dominant OSes/devices we observe in

OS/Device	NB Cnxs	NB Sessions	NB Cnxs/Session
Iphone	9780	2562	3.81
Windows Vista	1283	304	4.22
Windows XP	1170	376	3.11
Windows CE	290	140	2.07
Macintoch	169	55	3.07
Symbian	138	50	2.76
Linux	22	12	1.83
Others	326	126	2.58

TABLE 3.3 – Webmail Connections and Sessions

our trace. Main observations here was that Microsoft Vista users were characterized by highest number of connections per session and Iphone devices generate more connections compared to Windows CE and Symbian operating systems.

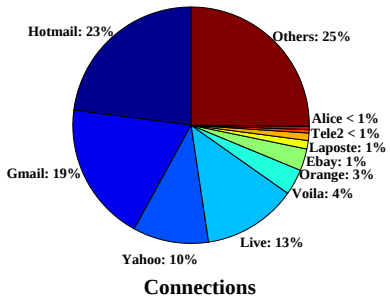


FIGURE 3.3 – Webmail Service Provider

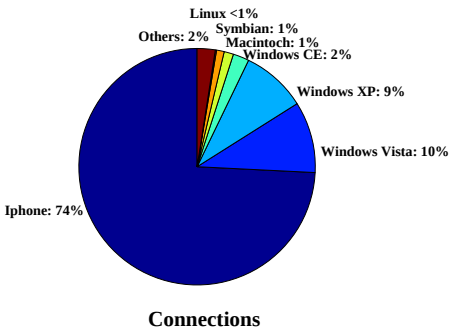


FIGURE 3.4 – OS and Devices for Webmail Traffic

3.3.3 Application Level Throughput

Throughput is an important metric for a lot of applications. Common practice is to use throughput for applications generating bulk transfers, while response time is used for interactive applications. Mail traffic in general appears to be a mixed application, generating interactive and bulk

transfers. Bulk transfers are generated by large mails (with attachments), while interactive transfers are due to mailbox checking and the sending/reception of small mails. In this section, we use the throughput to compare the performance of mail and webmail. Our purpose here is to show that the access technology influences the throughput but is not the only factor. Congestion, transport layer details or the application on top (e.g., rate limiters in p2p applications) can also impact the observed throughput.

We have shown in Chapter 2 that a straightforward estimation of throughputs where the amount of bytes transfered at the TCP layer is divided by the total duration between the first packet (first SYN) and last packet of the connection (e.g., FIN) provides a biased view of the throughput perceived at the user side. The tear down of a connection, that we define as the time between reception of the last data packet and the last control packet can be extremely high due to numerous reasons : the application, the server implementation or the operating system. We thus introduced in 2 the notion of Application-Level (AL) throughput where the amount of bytes transfered at the TCP layer is divided by the total duration between the first packet (first SYN) and last *data* packet of the transfer.

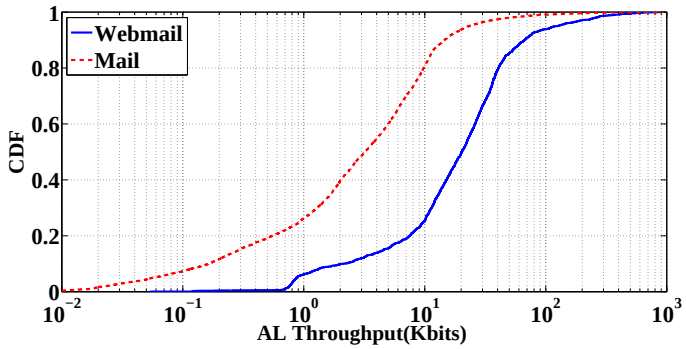


FIGURE 3.5 – Application Level Throughput

In Figure 3.5, we report the AL throughput for mail and webmail connections. A first striking observation is that webmail offers significantly higher throughputs than mail. More than 75% of webmail connections achieve a throughput higher than 10 kb/s, unlike mail where the equivalent portion is only 20%. Several factors can explain this discrepancy. In the following section, we explore in more details mail and webmail traffic characteristics, in order to find which parameters degrade mail performance. We focus on volumes of data exchanged, application impact, and time spent to recover from losses.

3.4 Detailed Performance Comparison

3.4.1 Connections Size

Figure 3.6 depicts the cumulative distribution of well-behaved (see Section 2.2) mail and webmail connection size in bytes. It appears that mail transfers are clearly smaller than webmail transfers. This observation is in line with the results in Tables 3.1 and 3.2 where we noticed the smaller number of webmail connections but the larger amount of data exchanged. We believe that two factors explain this observation : (1) webmail applications not only convey data related to the mailbox of the user but also data related to the HTTP frame of the Web page in which the content of the mailbox is displayed, (2) web(mail) applications use persistent connections unlike legacy mail protocols (POP, SMTP - but not IMAP), which results in longer transfers. A smaller amount

of data to transfer leads inevitably to a smaller throughput with TCP on average, which is a first explanation behind the observation of mail achieving smaller throughputs than webmail. However, different connection size is not the only factor that explains the lower throughput of mail as compared to webmail.

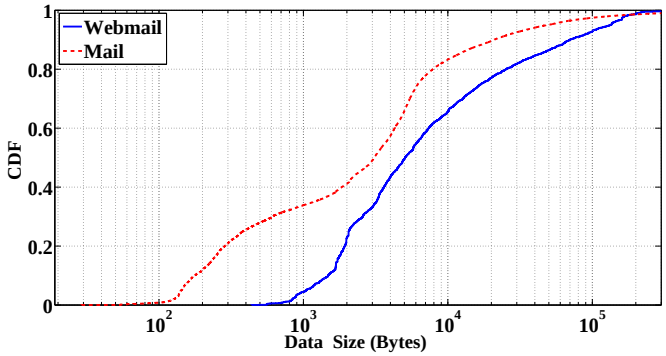
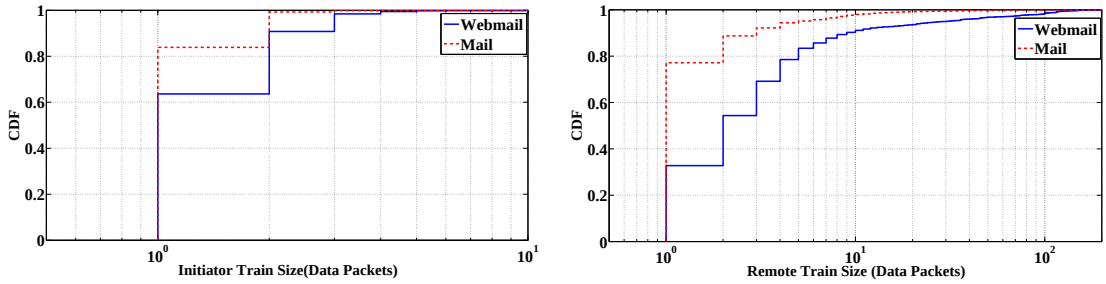


FIGURE 3.6 – Connections Size

3.4.2 Impact of Application on Top



(a) Initiator Side (b) Remote Side

FIGURE 3.7 – Exchanged Trains Size

For client/server applications, one generally observes that even if the server is sending a large amount of bytes/packets, the actual data exchange is fragmented : the server sends a few packets (hereafter called train), then waits for the client to post another request and then sends its next answer 2. If such a behavior is predominant, it can have a detrimental impact to TCP if the train size is too small, as it prevents TCP from performing FR/R in the case of losses.

We evaluate here the distribution of train sizes for mail and webmail transfers. For the connections for which synchronism could not be rejected, we looked at the distribution of the size of the trains of packets sent. We distinguish between the initiator of the connection, which is in our case the Cellular client and the remote party, which is the mail or webmail server.

Figure 3.7 reports the distribution of train sizes for webmail and mail transfers. We observe that :

- Trains sent by servers (remote party) are larger than those sent by the initiator (local client) ;
- Webmail trains are larger than the ones of mail traffic, for both initiator and remote party. In fact, more than 38% of webmail initiator trains are larger than 2 data packets, unlike mail where it is only 16%.

– More than 99% of initiator mail and webmail trains are smaller than 3 data packets, which leaves TCP unable to trigger any Fast Retransmit, even if Limited Transmit is used [87]. This might lead to performance issues during mail uploads.
– More than 92% of remote party trains are also smaller than 3 data packets, compared to only 70% for webmail. This again leaves TCP unable to trigger a fast recovery/retransmit, even if Limited Transmit is used in a lot of case. Mail is more affected than webmail though.

A conclusion of the above analysis is that both mail and webmail throughputs are affected by the behavior of the application on top of TCP with a potentially more detrimental effect for mail than for webmail transfers. Smaller train sizes tend to slow down TCP, as it prevents the protocol from opening its congestion window, but can also lead to longer recovery time during loss events. We turn our attention to this specific issue in the next paragraph.

3.4.3 Losses

To assess the impact of TCP loss retransmission times on the performance of mail and webmail, when we observed throughput estimation in Figure 3.5 , we developed an algorithm to detect retransmitted data packets, which happen between the capture point and the server or between the capture point and the client. This algorithm [4] is similar to the one developed in [75].

If ever the loss happens after the observation point, we observed the initial packet and its retransmission. In this case, the retransmission time is simply the duration between those two epochs [5]. When the packet is lost before the probe, we infer the epoch at which it should have been observed, based on the sequence numbers of packets. We try to separate real retransmission from network out of sequence events by eliminating durations smaller than the RTT of the connection.

Note that computations of all those durations are performed at the sender side, as time series are shifted according to our RTT estimate. For our trace it is easier to detect losses for the second case, because in case of packet loss the retransmitted data packet is seen twice. But, when the loss happened between the capture point and the distant server, we are only able to detect an out of sequence packets.

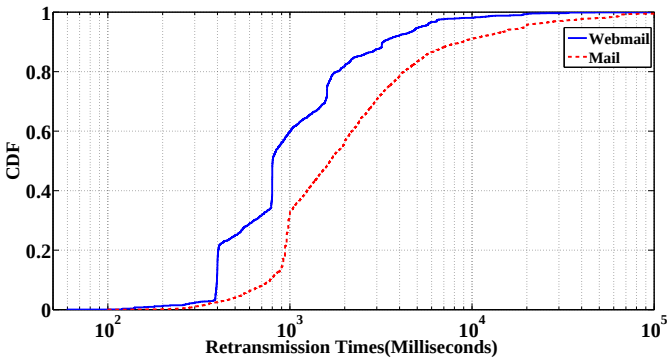


FIGURE 3.8 – Retransmission Times per Loss Event

We define the retransmission time as the time elapsed between two duplicate TCP data packets or the moment where we observe [6] a decrease of the TCP sequence number and the first time where it is reaches a value larger than the largest sequence number observed so far. Once losses are

4. The used loss' detection algorithm is available on http ://intrabase.eurecom.fr/tmp/papers.html. People are invited to check the correctness of our algorithm to detect losses
5. Those epochs are computed at the sender side by shifting the time series according to our RTT estimate.
6. at the sender side – time series are shifted according to our RTT estimate.

identified with (i) data packet retransmission and (ii) out of sequence data packet, we compute for each TCP connection retransmission time for each loss event. We do not distinguish between out of sequence data packets and retransmitted data packets. We will only use the term "retransmission".

Figure 3.8 plots the cumulative distribution of retransmission time per each loss event, for mail and webmail traffic. As expected from our study of train times, mail traffic experiences larger recovery times than webmail traffic.

We can further notice two thresholds of common retransmission times at 400 ms and 1 seconds for webmail and mail respectively. This in in-line with work in [56, 88] of RTO estimation for Cellular networks, where authors show that RTO bound has been shortened in modern widely spread TCP implementations for Cellular networks.

In summary, several factors contribute to the degradation of mail performance as compared to webmail. Some of these factors are driven by clients usage while others are more fundamentally related to the way those different mail implementations work and their interplay with the transport layer.

3.5 Conclusion

In this Chapter we have reported some observations about the Internet traffic of a Cellular network with users connected via handsets or USB pluggable 3G modems. The predominance of Iphone however suggests that the first category of users actually dominates over the second one, for the ISP we consider. We have highlighted that measurements from passively collected traces can be biased by specific technologies implemented in Cellular networks to boost performance and control users activity. RTT, which is a key metric, is especially affected by those network appliances.

We cast a first look to mail and webmail traffic in Cellular networks. We found that mail seems to be less popular than webmail as the majority of mail data is transfered using webmail.

A first explanation to this difference in usage is the high popularity of free webmail service providers like Google, Yahoo !, Hotmail, etc. We indeed observed that those providers are much more used than the webmail services offered by the network providers. This is presumably because users want an email account that is independent of their network provider, in case they switch to another network provider.

We further observed that webmail performance outperforms the one of mail. We demonstrated that several factors lead mail to offer smaller throughputs than webmail, especially, the size of the transfers, the application semantics which leads to smaller data exchange phases, which slows down TCP in general and prevents fast retransmit if losses are detected.

In the next Chapter, we presented a first look of a methodology in order to compare performance of different accesses technologies. We focus on usual suspects indicators that can influence client perceived performance.

Conclusion of Part I

In Part I of this thesis, we first revisited main research works related to our objectives of TCP performance analysis for the case of Internet and enterprise traffics. Our main observation was that focusing on the impact of new applications, client behavior and server impacts, has been often overlooked in the literature. To perform our traffic analysis, we used a traces collected from heterogeneous wireless and wired environments, which highlight the wide scope of our study of traffic analysis performance. Our traffic analysis study is based on a DBMS approach, which allows to manage collected dumps into a database.

We presented an overview of the impact of the application on top of TCP. With our connection time break-down, we showed that while losses can have a detrimental impact on short TCP transfers, the application significantly affects the transfer time of almost all short and long flows in a variety of way, e.g. tear down and short exchanged train size of data.

Our study of a Cellular trace with users connected via handsets or USB pluggable 3G modems highlighted that measurements, e.g. RTT estimation, from passively collected traces can be biased by specific technologies implemented in Cellular networks to boost performance and control users activity. To study the application impact we performed a first study of mail and webmail applications. Mainly we observed that webmail performance outperforms the one of mail. We demonstrated that several factors lead mail to offer smaller throughputs than webmail, especially, the size of the transfers and the application semantics.

In Part II, we introduce an analysis method that uncovers the impact of each layer that contributes to the overall data transfer time, namely the application, the transport layer, and the end-to-end path. The analysis method that we use consists in two steps. First, the transfer time of each TCP connection is broken down into several factors that we can attribute to different causes. Second, we use a clustering approach to uncover the major trends within the different data sets under study.

Part II

A New Approach to Performance Analysis of TCP Transfers

Overview of Part II

In the previous part, we have underscored both the crucial impact of the applications on top of TCP and also, for the case of the Cellular technology, the impact of some technological choices on the metrics typically used to assess performance levels. In part II of the thesis, we present a method that drills down into the data transfer of each well-behaved connection, which is the main contributions of this thesis. The approach developed is exemplified with the set of traces collected on the Cellular/FTTH and ADSL backbones of Orange.

In Chapter 4, before turning our attention only on the data transfer phase, we explore several factors that are classically used to assess the performance of TCP connections, namely RTT and losses. The crucial impact of those parameters are formally known since the derivation of the well-known TCP throughput formula [89]. We discuss the derivation of those parameters for the case of our traces. At the end of this chapter, we illustrate shortly the fact that RTT and losses are not enough to characterize TCP connection in the wild, justifying our efforts in Chapter 5 of drilling down into the data transfer phase.

In Chapter 5 we propose a new analysis method that uncovers the impact of specific factors like the application and the interaction with user, and thus informs the comparison of heterogeneous access technologies. The analysis method that we use consists of two steps. In the first step, the transfer time of each TCP connection is broken down into several factors that we can attribute to different causes, e.g., the application or the end-to-end path. In a second step, we use a clustering approach to uncover the major trends within the different data sets under study.

In Chapter 6, we address the problem of comparing the performance perceived by end users when they use different technologies to access the Internet. Users primarily interact with the network through the networking applications they use. We tackle the comparison task by focusing on several Internet key services such as Google search and mail. This is because we focus on user perceived performance and users do not care about raw performance metrics. They care about the performance of the applications they use. We then apply our data time break-down approach, based on a fine-grained profiling of the data time of transfers that sheds light on the interplay between service, access and usage, for the client and server side. We use clustering approaches to identify groups of connections experiencing similar performance over the different access technologies.

In Chapter 8 we present characteristics of some salient aspects of enterprise traffic. Our goal is to provide an overview of the problem faced when performing measurements in such environments such as basic RTT estimation. We also present a fine-grained profiling of the most popular applications used in the network we measure.

Chapter 4

A First Look on Key Performance Parameters

4.1 Introduction

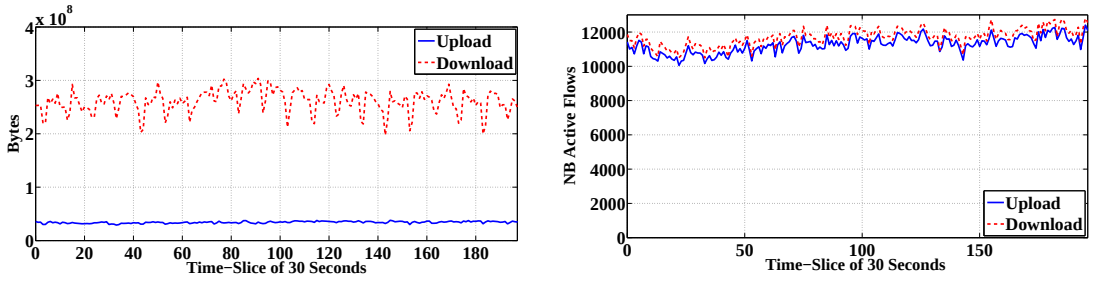
The study of TCP behavior, specifically its performance in terms of delay, losses and throughput, has been studied since its emergence for specific environments and users. However, comparing and understanding key parameters that influence perceived performance from different access technologies such as Cellular, FTTH and ADSL traffics becomes difficult when it is interacting with the application layer above and the network layer below. Here after we report a classical approach to compare performance of different access technologies in order to conclude if clients fully benefit from their broadband access. In this Chapter, we first assess the stability of the traffic for the trace that we have to study. Then we briefly analyse usual suspects that can impact the results of different accesses. We also provide a systematic study of the tear-down phase of the well-behaved connections that highlight the diversity of scenarios observable in practice. Finally, we illustrate shortly the fact that RTT and losses are not enough to characterize a TCP connection in the wild, justifying our efforts in chapter 2 of drilling down into the data transfer phase.

4.2 Traffic Stability

4.2.1 Data Volume

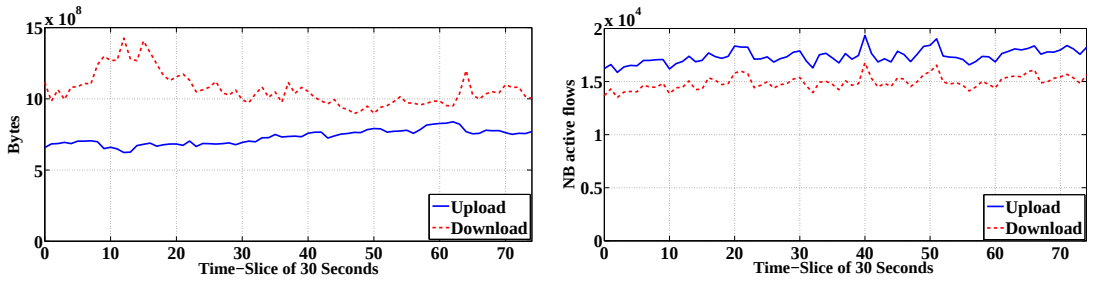
In this part, we assess the stability of the traffic for the second datasets, which we introduced in 1 and consists of traces captured under different environments : Cellular, FTTH and ADSL. For this purpose, we observe the time series of traffic volume and the number of active flows. The objective is to assess if several regimes exist in our data, which would require to analyze the performance within each corresponding time interval. As we will see, it is apparently not the case with our trace. This justifies our approach in this Chapter, where we will look at marginal distributions of different metrics where all samples of the trace are grouped together to form those distributions.

Figures 4.1, 4.2 and 4.3 shows the evolution of traffic volume and the number of active flows for the upload and the download directions. To obtain those figures, we broke up our trace into short time windows of 30 seconds and we compute the number of active flows and the exchanged data volume for each direction in each window. Note that a flow is considered active for a given time slice if it transmits at least one data packet during the slice.



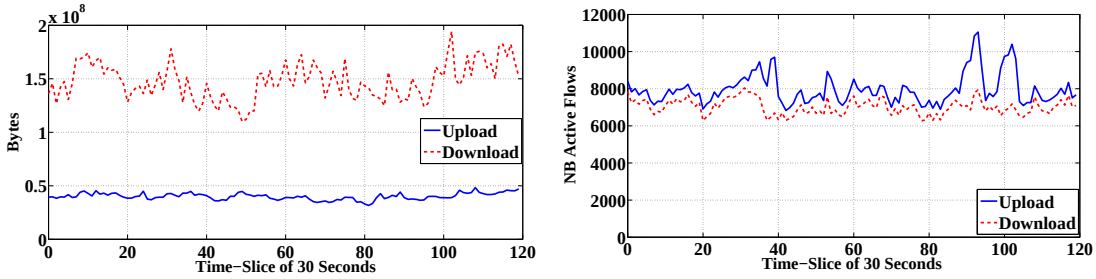
(a) Bytes (b) Number of Active Flows

FIGURE 4.1 – Upload and Download Data Volume Evolution : Cellular



(a) Bytes (b) Number of Active Flows

FIGURE 4.2 – Upload and Download Data Volume Evolution : FTTH



(a) Bytes (b) Number of Active Flows

FIGURE 4.3 – Upload and Download Data Volume Evolution : ADSL

Figures 4.1(a), 4.2(a) and 4.3(a) show that traffic is qualitatively more bursty in the download than the upload direction. This is presumably because bursts are shaped by the limited uplink capacity implemented by the operator. Another immediate observation is the difference of uplink and downlink capacity between observed accesses. FTTH traffic is characterized by higher values of exchanged data in terms of bytes and active flows.

Concerning active flows, Figures 4.1(b), 4.2(b) and 4.3(b) demonstrate that they do not vary drastically over time, further reinforcing the idea that traffic is stable over the time span of our trace.

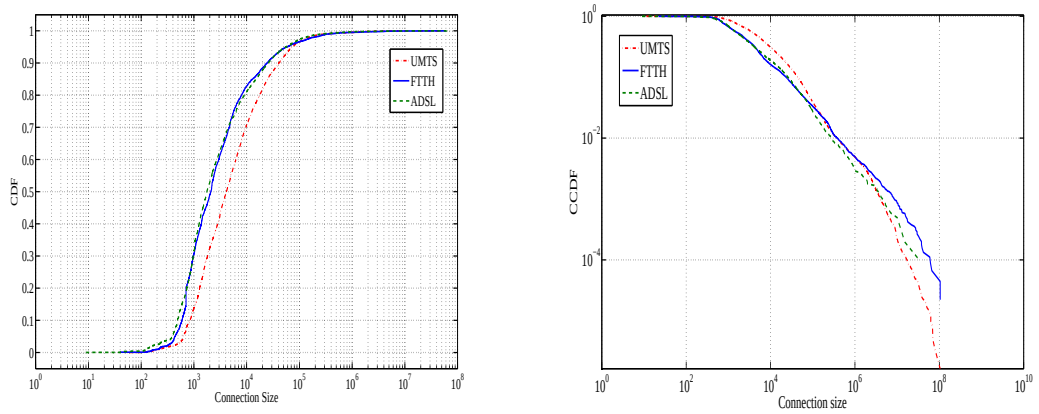
The study of the evolution of exchanged data volume and number of active flows did not reveal any abnormal phenomenon or anomalies in our traces. Hence, further analysis will be based on all identified well behaved connections.

4.3 Usual Suspects

4.3.1 Exchanged Data Volume

Figure 4.4 depicts the CDF and Complementary CDF (CCDF) of connection size in terms of bytes, for Cellular, ADSL and FTTH traces. Only well behaved connections are considered. We observe that FTTH and ADSL traces offer a similar connection profiles that significantly differs from the radio access. For instance 30% of ADSL and FTTH traces are less than 1kbytes and 55% are between 1kbytes and 10 kbytes, unlike Cellular which offers larger values at similar connection percentiles.

The inspection of CCDF, shows that the probability to obtain transfers with 1 Megabyte is very low (under 0.01). It reveals that while the majority of Cellular connections did not target p2p ports Cellular users are able to perform as large connections as wired accesses.



(a) CDF (b) CCDF

FIGURE 4.4 – Connection Size (bytes)

In fact, several explanations can be found for this observation. For instance, the usage of persistent HTTP connections (more than 84% of Cellular traffic target HTTP(s) ports). Also, the usage of new applications or services in new devices like the download of applications from 'Apple Store' or 'Android Market' and the increase of streaming applications (Youtube, etc) explain the higher values of connections size for Cellular access compared to FTTH and ADSL.

The main conclusion from this paragraph is that, nowadays, Cellular users tend to use their handsets to perform a new usage different from the simple call or Short Message Service (SMS) send, in line with the increase of display and Central Processing Unit (CPU) capacities of smartphones This means that Cellular access is not used for a limited period or nomadic usage but for a current uses.

4.3.2 Access

We observed that both RTT estimation methods with SYN-/SYN-ACK or DATA-ACK lead to a same estimate of the round trip time for ADSL and FTTH traces, while we observe differences for Cellular access because of a Performance Enhancing Proxy (PEP) and APN, as presented in Section 3.2.

We thus rely on the DATA-ACK method to estimate RTTs over considered traces. Figure 4.5 depicts the resulting RTT estimations for the three traces. It clearly highlights the impact of the

access technology on the RTT. FTTH access offer low RTT in general – less than 110 ms for more than 60% of connections. This finding is in line with the characteristics generally advertised for FTTH access technology. On other hand, RTTs on the Cellular technology are notably longer than under ADSL and FTTH, in line with intuition.

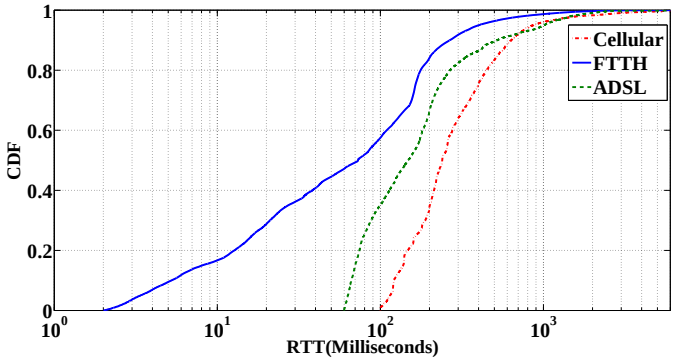
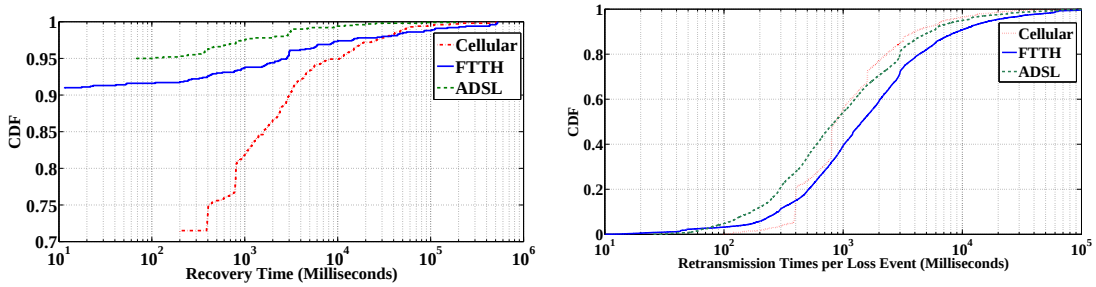


FIGURE 4.5 – RTT Estimation

4.3.3 Data Packet Retransmission

To assess the impact of TCP losses on the performance of considered access, we based our study on approach presented in 3.4.3.



(a) Per Connection (b) Per Loss Event

FIGURE 4.6 – Retransmission Time

We do not distinguish between out of sequence data packets and retransmitted data packets. We will only use the term "retransmission". We try to separate real retransmissions from network out of sequence events by eliminating durations smaller than the RTT of the connection. Once losses are identified with (i) data packet retransmission and (ii) out of sequence data packet, we compute total retransmission time for each TCP connection.

Figure 4.6 depicts the cumulative distribution of retransmission time per connection, for considered accesses. The main observation is that retransmission ratio is higher for Cellular with more than 28.6% and only less than 9% for ADSL and FTTH accesses. It also demonstrates that loss ratio decreases with high bandwidth. An intuitive explanation of such observation may lay in the difference of reliability between Cellular and wired accesses.

From previous works, we noticed that authors presented several factors that influence loss ratio for Cellular access. In fact, in [90] authors recommend to use a loss detection algorithm, which uses dumps of each peer of the connection (this algorithm is not adapted for our case because our

measurements have been collected at a GGSN level) to avoid spurious Retransmission Timeouts in TCP. In addition, authors report in [56] that spurious retransmission ratio in Cellular networks is higher for Google traffics than other ones, due to short implemented Timeouts in Google servers.

4.4 How Applications Free TCP Connections ?

4.4.1 FIN vs RST flags

Closing TCP connection is an operation which means that the closing side has no more data to send. The notion of closing a full-duplex connection is subject to ambiguous interpretation [73], since it may not be obvious how to decide to the receiving side of the connection.

The final flag is the FIN flag, standing for the word finished. This flag is used to tear down the virtual connections created using an established connection.

It is important to note that when a host sends a FIN flag to close a connection, it may continue to receive data until the remote host has also closed the connection, although this occurs only under certain circumstances.

In other hand, upon reception of RST segment, the receiving side will immediately abort the connection. This statement has more implications than just meaning that each side will not be able to receive or send any more data to/from this connection.

Once the connection is liberated by both sides, the buffers associated on each end for the connection are released. Previous work on TCP performance and the application layer did not cover or focus on the step of tearing down a connection.

In 2.4.1 we defined the tear-down as the time between receiving the last data packets and the last FIN or RST control packets. We have shown that the tear-down phase in itself often represents the majority of the connection time. Hence, large tear-down times can alter throughput estimation by largely underestimating actual throughput, i.e., the throughput perceived by the application.

Note however, that the tear-down time should have no impact on the performance perceived by the client, as the data transfer is completed. Client and server are more interested by data exchange time than time to free TCP connection. In contrast large tear-down times can be penalizing for the server in terms of allocated resources, memory and processes. In fact, servers are in general configured to have a limited number of connections that are allowed to connect from clients.

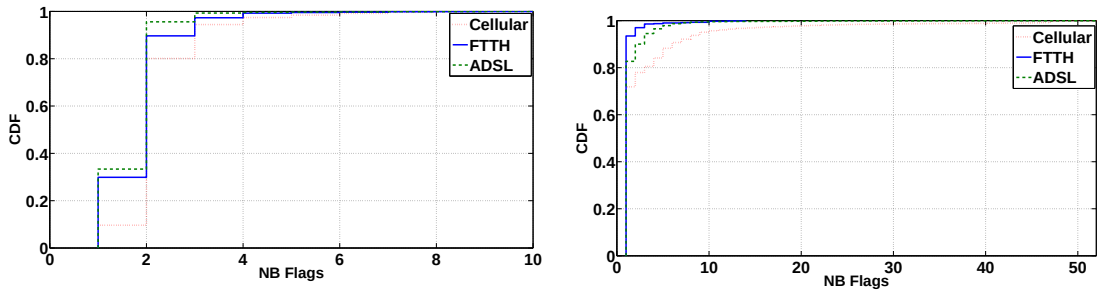
	FIN		RST	
Traces	%	AL TH(Kbts)	%	AL TH(Kbts)
Cellular	92.42	28.74	7.57	9.73
FTTH	92.96	77.25	7.04	57.58
ADSL	89.91	71.62	10.08	66.96

TABLE 4.1 – Tear-down Flags

For instance, the default value for an Apache server is 300 connections. The objetive is to avoid exhaustion of memory resource at the server side as might for instance occur during a SYN DoS attack.

We present in Table 4.1 the percentage of connections and Application Level Throughput, for each access, that perform tear-down with FIN or RST flags. In all traces, results show that ended TCP connections with FIN flag are characterized by higher throughput than ones finished with RST flag. It suggests that connections ended with RST flag offer worst performance and can be characterized as connections that are not finished correctly and probably having an anomaly.

Also, Table 4.1 demonstrates that more than 89% of TCP connections were finished with FIN flags and less than 10% with RST flags. This further enforces the hypothesis of an anomaly that triggered the sending of RST flags.



(a) FIN (b) RST

FIGURE 4.7 – FIN/RST Flags Distribution

Figure 4.7 shows distributions of exchanged FIN and RST flags. We find that connections that finish correctly exchange a median values of 2 FIN data packets : one per side, in line with [73]. Median value of RST control packets. Also, more than 80% of connections that finish correctly, have less than 2 FIN control packets.

More than 72% of up to the remaining connections were finished with a median of one RST flag. It is important to note that we noticed several high values of exchanged RST flags which could reach up to 100 for Cellular and 46 for ADSL accesses.

A worth to investigate hypothesis that connections with large number of tear-down control packets probably correspond to an applicative anomalies.

	FIN		RST	
Trace	Init (%)	Rem (%)	Init (%)	Rem (%)
Cellular	85.96	14.03	49	52
FTTH	48.71	51.28	72.72	27.27
ADSL	44.11	55.88	83.09	16.9

TABLE 4.2 – Tear Down Side Initiation - Percentages

Table 4.2 depicts the percentage of clients and servers that finish TCP connections with FIN or RST flags. The main observation here is the difference between connections depending on the underlying access technology. However, for FTTH and ADSL we observe in Table 4.2 that between 51% and 55% of TCP connections are closed by the server, with FIN flags, which means that approximatively we find the same ratio of TCP connections closed by each side. While only 14% of Cellular connections are closed by the server with FIN flags.

A basic explanation for this observation is that Cellular users are limited by their devices, in the sense that after performing a browsing (which is the main user activity as described in paragraph 1.7.2.1 user closes immediately the Internet browser after the end of browsing in order to switch to other activity or to lock their phone. While, for FTTH and ADSL, client keeps the browser opened in the background and can do other activities at the same time.

However, the study of RST flag shows that FTTH and ADSL users are more frequently closed by the client with more than 72% of connections. For Cellular trace, result are more balanced with 49% of connections are closed by the client while 52 are closed by the server.

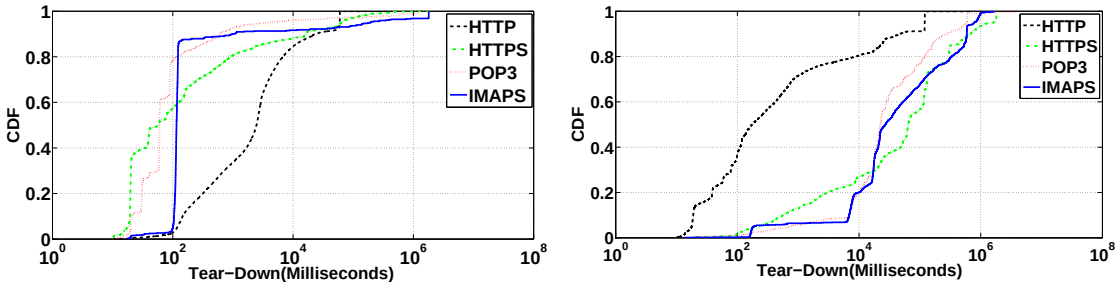
	FIN		RST	
Trace	Init (ms)	Rem (ms)	Init (ms)	Rem (ms)
Cellular	2000	6000	154.79	36000
FTTH	55.54	8528	39148	40854
ADSL	80.46	516.08	702.31	25444

TABLE 4.3 – Tear Down Side Initiation - Median Times

Table 4.3 shows the median of tear down values found in each case of figures. We distinguish cases where the initiator or the remote side of the connection sends FIN or RST flags. Our main observation is that tear down times are higher when the remote side initiate tear down step. This suggests that remote server close the current opened connections once the client is idle for a period higher than the maximum time-out.

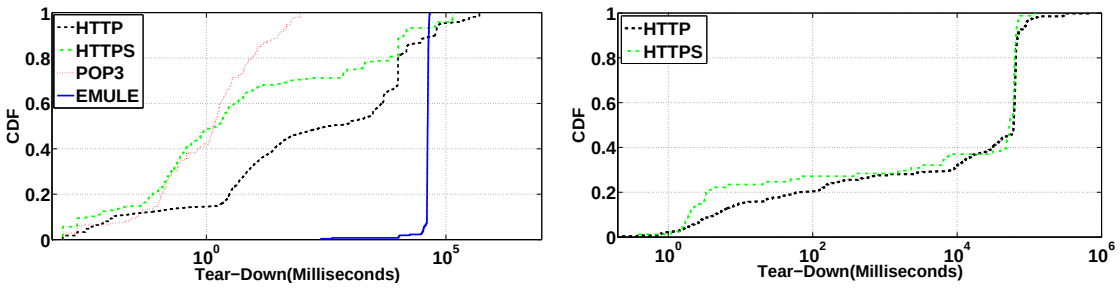
4.4.2 Diversity of Thresholds

Based on the previous analysis and in order to further investigate these results, we categorize tear-downs in several classes based on FIN and RST flags. Tear-down times are defined as the time between the last data packet and the first FIN/RST control packet. The idea here was to check if different behaviors exist when connection is liberated, depending on the application on top of TCP. We expect to observe specific time-out values when a TCP connection is finished with FIN or RST for each considered service or application.



(a) FIN (b) RST

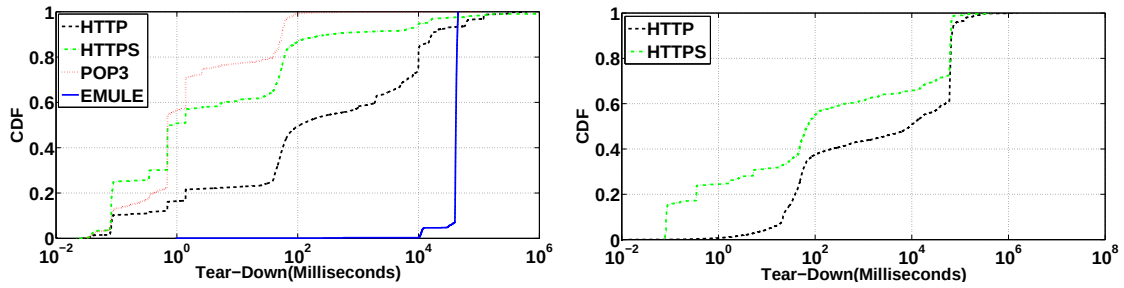
FIGURE 4.8 – Cellular : Heterogeneous Tear-down Times



(a) FIN (b) RST

FIGURE 4.9 – FTTH : Heterogeneous Tear-down Times

In Figures 4.8, 4.9 and 4.10 we present tear down times for different TCP services and for our



(a) FIN (b) RST

FIGURE 4.10 – ADSL : Heterogeneous Tear-down Times

Cellular, FTTH and ADSL traces. For each access we distinguish between connections finished with RST or FIN flags. Main observations are :

- 87% of IMAPS Cellular connections, finished with FIN flag show a tear-down time-out of 200 ms
- Few ADSL and FTTH connections using POP3 and Emule finish TCP connection with RST flags, the majority of connections are closed using FIN flags
- Large variability of tear-down times within observed application, finished with FIN or RST flags
- FTTH and ADSL accesses show similar tear-down values for Emule connections finished with FIN flag
- Even for the same service, we observe different distributions for tear-down times for connections finished with RST or FIN flags.

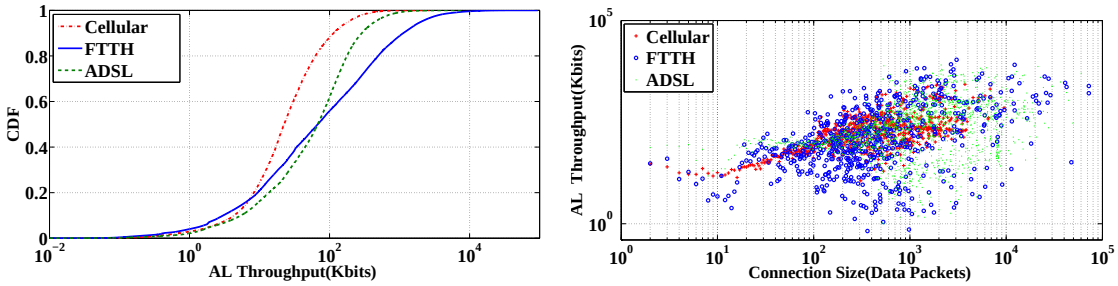
From this study, we see that connection tear-down times depend on several factors. (1) The flag : tear-down values are higher for connections finishing with RST flags than with FIN flags. (2) and the used application used on top of TCP.

Our study was mostly descriptive. A primary reason for it is that, to the best of our knowledge, no research work so far as paid attention to the tear down-phase. We have only scratched the surface of the problem. More work needs to be done. We have not done much in the context of this thesis as our focus is on the client performance and somehow, the tear down phase does not impact the client. The latter is true if the connection finishes correctly from the TCP viewpoint, i.e. with a FIN and not a RST flag. This further justifies our choices of focusing on well behaved connections.

4.5 Performance Comparison Challenge

Our purpose here is to show that the access technology influences the throughput, but it is not the only factor. Congestion, transport layer details or the application on top (e.g., rate limiters in p2p applications) can also impact the observed throughput. We base our estimation of throughput on the definition presented in Section 2.4.1 where throughput corresponds to the amount of bytes transferred at the TCP layer, divided by the total duration between the first packet (first SYN) and last *data* packet of the transfer. Formally, this is what we called the application layer throughput.

In Figure 4.11(a), we report CDF of AL throughput for our traces. A first striking observation is that FTTH and ADSL accesses offer significantly higher throughputs than Cellular. As we presented previously in Section 4.3.2, we can confirm that this observation is a consequence of the RTT available for each used access. On the other hand, we can notice that AL throughput for



(a) CDF (b) Median

FIGURE 4.11 – Application Level Throughput

ADSL and FTTH often similar(up to the 50th percentile), in contrast with what end users expect. A first explanation of this fact, was the RTT distribution for ADSL and FTTH.

In order to avoid the mixed results of AL throughput for short and large connections, we plot in Figure 4.11(b) median values of AL throughput per connection size in terms of data packets. It shows that higher values of AL throughout were obtained with FTTH connections. But in other hand, it confirms results observed in Figure 4.11(a) : throughput for FTTH, ADSL and Cellular are not as different as one can expect, when we focus only on RTT, loss and connection size.

To compare performance of different Internet accesses technologies, we started with a classical approaches based on the study of the two key factors that influence the throughput of TCP transfers (see the TCP throughput formula [89]), namely loss rate and RTT. It suggests that the performance over FTTH should significantly outperform the one of ADSL, which should in turn outperform the one of Cellular. But, it turns out that reality is slightly more complex as can be seen from Figure 4.11(a). Indeed, while the Cellular technology offers significantly small AL Throughput, in line with RTT and loss factors, FTTH and ADSL have much closer performance than RTT and loss were suggesting.

Next in our work, we present a new method to uncover the impact of the application and to better explain the differences or lack of differences between the access technologies. By application, we mean the way applications work, and also the way the user interacts with the application, as the latter directly impacts the way that data is delivered to the transport layer. In addition of the user behavior, which is a function of the access technology. For instance, large file downloads might be rare on Cellular technology unlike wired technologies.

4.6 Conclusion

In this section, we have applied a somewhat classical methodology in order to compare performance of different accesses technologies. We first reported the stability of the traffic within the trace that we study : Cellular, FTTH and ADSL. The objective was to assess if several regimes exist in our data, which would then require to analyze performance within each corresponding time interval. We focused then on key parameters that can influence client perceived performance. Specially emphasized the importance of the time to recover from losses and to free TCP connections.

From the study of tear down process, we uncovered a high diversity in terms of observed timeout and times to liberate TCP connection. We demonstrate that connection tear-down time depends on several factors. Especially, the used flags : SYN or RST, the used application on top of TCP and the access technologies (user behavior depends on the used device)

We conclude that the observed values of loss recovery times and RTT are not sufficient to explain the observed performances, specially, FTTH and ADSL have much closer performance than RTT and loss were suggesting.

In the next Chapter, we propose a new analysis method that uncovers the impact of specific factors like the application and the interaction with user, and thus betters informs the comparison of access technologies.

Chapter 5

Methodology : The Interplay Between Application, Behaviors and Usage

5.1 Introduction

Our study of the key factors that influence the throughput of TCP transfers, namely, connection size, loss rate and RTT. RTT suggested that FTTH should significantly outperform the one of ADSL, which should in turn outperform Cellular. It turns out that reality is slightly more complex as seen in Section 4.5. Indeed, while the Cellular technology offers significantly longer response time, in line with RTT and loss factors, FTTH and ADSL have much closer performance than RTT and loss were suggesting.

In this Chapter, we propose a new analysis method that uncovers the impact of specific factors like the application on top of TCP and the interaction with the user, in order to inform the comparison of different access technologies.

The analysis method that we use consists in two steps. In the first step, the transfer time of each TCP connection is broken down into several factors that we can attribute to different causes, e.g., the application or the end-to-end path. In a second step, we use a clustering approach to uncover the major trends within the different data sets under study.

5.2 Methodology

In this paragraph, we introduce a methodology that extends what has been introduced in Section 2.4. The objective is to reveal the impact of each layer that contributes to the overall data transfer time, namely the application, the transport, and the end-to-end path (network layer and layers below) between the client and the server.

We perform a break down of the duration of the data transfer phase of a TCP connection, which we term *data time*, i.e., excluding the connection establishment and tear down phases.

The starting point is that the vast majority of transfers consist of dialogues between the two sides of a connection, where each party talks in turn. This means that application instances rarely talk simultaneously on the same TCP connection [91]. We call the sentences of these dialogues *trains*.

For instance, as explained in Section 2.6 we observe that even if the server is sending a large amount of bytes/packets, the actual data exchange is fragmented : the server sends a few data packets (one train), then waits for the client to post another request and then sends its next answer, i.e. the next set of packets (another train).

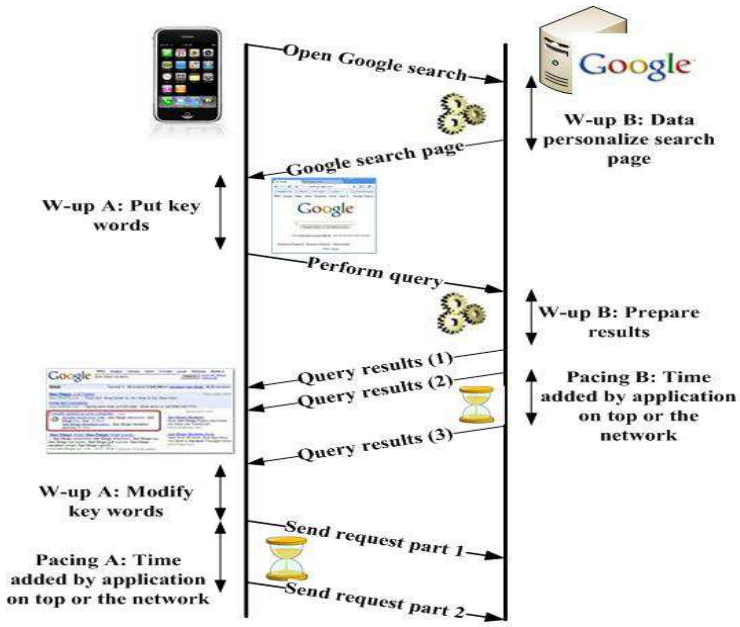


FIGURE 5.1 – Data Time Break-Down

We term A and B the two parties involved in the transfer (A is the initiator of the transfer) and we break down the data transfer into three components : Warm-up time, Theoretical time and Pacing time. Figure 5.1 illustrates this break down in the case of a Google search where A is a client of the ISP and B is a Google server.

A Warm-up corresponds to the time taken by A or B before answering to the other party. It includes durations such as thinking time at the user side or data preparation at the server side. For our use case, a Warm-up of A corresponds to the time spent by the client to type a query and to browse through the results before issuing the next query (if any) or clicking on a link, whereas a Warm-up of B corresponds to the time spent by the Google server to prepare the appropriate answer to the request.

Theoretical time is the duration that an ideal TCP transfer would take to transfer an amount of packets from A to B (or from B to A) equal to the total amount of packets exchanged during the complete transfer. Theoretical time can be seen as the total transfer time of this ideal TCP connection that would have all the data available right at the beginning of the transfer. For this ideal transfer, we further assume that the capacity of the path is not limited and an RTT equal to RTT_{A-B} (or RTT_{B-A}). We depict in Table 5.1 an example of Theoretical time computation.

Round	Current Window	Data Sent
1	2	2
2	3	5
3	3	8
4	5	13
5	6	19
6	8	23

TABLE 5.1 – Example : Theoretical Time Computation(Data packets=23, Cwnd initial= 2, ACK for 2 data packets)

Theoretical times can be seen, for example, as the durations that FTP transfers would take to complete when neglecting the dialogue between A and B, i.e., the application impact. For our case, let us assume, that we have 4 data packets sent by our client (connects to Google server + Performs a query + request part 1 + request part 2). Then our client Theoretical data time will correspond to the time spent by a TCP model [92] to send 4 data packets with a certain initial congestion window (already estimated from the initial congestion window).

Once Warm-up and Theoretical times have been substracted from the total transfer time, some additional time may remain. We term that remaining time Pacing time. Theoretical time can be attributed to characteristics of the path, Warm-up time to applications and/or user, and finally Pacing is due to the access link and the application on top of TCP. Indeed, as we assume in the computation of Theoretical time that A and B have infinite access bandwidth, we in fact assume that we can pack as many MSS size packets within an RTT as needed, which is not necessarily true due to a limited access bandwidth. In this case, the extra time will be factored in the Pacing time. Similarly, if the application or some middle-boxes are throttling the transmission rate, this will also be included in the Pacing time. A contextual interpretation that accounts for the access and application characteristics is thus needed to uncover the cause behind an observed Pacing time. The above breakdown of the total transfer time is computed for each side A and B separately.

Note that to obtain accurate estimations of those durations that are related to the sender or receiver side, we have to shift in time the time-series of packets received at the probe. Specifically, we assume that a packet received from A at probe P was sent $\frac{RTT_{P-A}}{2}$ in the past and will be received $\frac{RTT_{P-B}}{2}$ in the future, where RTT_{P-A} (resp. RTT_{P-B}) is the RTT between P and A (resp. B). While doing this operation, we assume that the RTT of the transfer stays constant.

The above breakdown strategy results in a complete partition of the total transfer time.

5.3 How to Present Results ?

5.3.1 Crude Representation

After performing data time break-down, each well-behaved connection is transformed into a point in a 6-dimensional space (Pacing, Theoretical and train time of the client and the server).

We report in Figure 5.2 a representation of the data time break down for Cellular, FTTH and ADSL traces. Figures 5.2 shows the breakdown per direction and per access technology with, for each case, the median of each component in relative (left y axis - relative to total data time) and absolute values (right y axis - in seconds). Medians enable to eliminate potential outliers.

The first observation from FTTH and ADSL data time break down is that they feature similar duration in terms of percentage (and approximatively the same in terms of seconds), in line with the similarity between the two traffics in terms of usage and basic characteristics (connections size, destination port and traffic volume distribution,etc..)

From Figure 5.2, we observe that 45.4% of data time of the Cellular transfers is spent on Warm-up time at the client side, against only 30.6% for all Theoretical data transfer time (client and server). We notice that Pacing is more important on the server side (13.1%) than the client one. It suggests that Cellular users were more affected by server policies/performance and remote network side impact, than throughput limitation on the client side. FTTH and ADSL have the same Warm-up B (a median value of 50 ms). This suggests that servers do not distinguish between ADSL and FTTH servers, which corresponds to intuition as we expect to observe similar usage of users in both environments. Warm-up B values is more important for the Cellular access. We note the same observation for warm-up A.

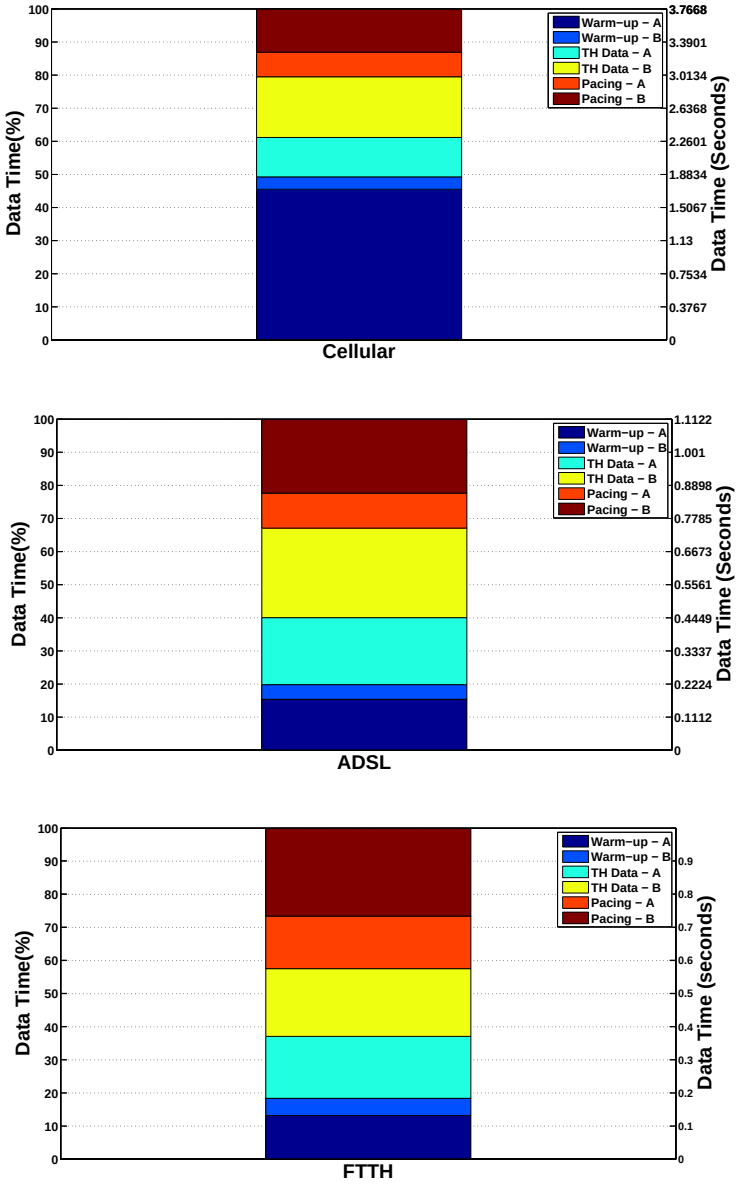


FIGURE 5.2 – Data Time Break Down

The main conclusions, at least at this stage from this data time break down, is that more than 55% of data time is spent during data preparation on the server side (Warm-up B), client interaction/thinking (Warm-up A) and Pacing times. It is clear from the comparison of Theoretical times that performances are better for fast accesses, but their impact is not as important as Warm-up or Pacing times.

5.3.2 Clustering Approach

Here-after we use clustering approaches to obtain a global picture of the relation between the service, the access technology and the usage.

After performing data time break-down, each well-behaved connection is transformed into a

point in a 6-dimensional space (Pacing, Theoretical and train time of the client and the server). To mine this data, we use a clustering techniques to group connections with similar characteristics.

We use an unsupervised clustering approach, namely the Kmeans algorithm. A key issue when using Kmeans is the choice of the initial centroids and the number of clusters targeted. To assess the number of clusters we use a test and trial approach were we start with an initially large number of clusters and then reduce this number as long as insignificant (i.e., too small) clusters remain. Concerning the choice of the centroids, we perform one hundred trials and take the best result (i.e., the one that minimizes the sum over all clusters of the distances between each point and its centroid). Note that we use the Matlab implementation of Kmeans [93].

To assess the number of used clusters, we rely on a visual dimensionality reduction technique, t-Distributed Stochastic Neighbour Embedding (t-SNE) [94]. t-SNE projects multi-dimensional data on a plane while preserving the inner neighbouring characteristics of data.

A straightforward application of the 6-dimensional points obtained from the 2-step approach presented above, bears a difficulty. Indeed, the per dimension values tend to be highly dependent on the connexion size. For instance, the warm-up A value is the sum of all warm-up periods over the whole duration of the transfer (for the A to B direction). Theoretical and Pacing time depend on the total number of packets to send. Then it is important when presenting results to keep a look on connection size, because it is more probable that large connection size have largest Warm-up and Pacing times (but also as we can notice in our further analysis that this assumption is not always true due to several parameters to be detailed next in this work).

Finally, to present results, we use boxplots [1] to obtain compact representations of the values corresponding to each dimension. On the top of each cluster we put the median size of grouped connections, cluster ID and for each trace the percentage of connections. This percentage is computed as the number of connections in a cluster over the total number of connection for a trace, i.e., an access technology. It is important to note that when performing clustering, we use the same number of connection form each trace.

For each clustering case, we use the same number of samples per access technology to prevent any bias in the clustering. Note that connections were chosen randomly among the ones in each traces.

5.4 Conclusion

We presented in this Chapter our methodology to reveal the impact of each layer that contributes to the overall data transfer time, namely the application, the transport, and the end-to-end path. We focused on the duration of the data transfer phase of a TCP connection, which we term *data time*, i.e., excluding the connection establishment and tear down phases.

After performing data time break-down, each well-behaved connection is transformed into a point in a 6-dimensional space (Pacing, Theoretical and train time of the client and the server). A warm-up corresponds to the time taken by each side involved in a TCP transfer before answering to the other party. It includes durations such as thinking time at the user side or data preparation at the server side. Theoretical times can be seen, as the time needed by an optimal TCP algorithm to transfer an amount of data computed for each studied connection. Once warm-up and Theoretical times have been substracted from the total transfer time, some additional time may remain. We term that remaining time Pacing time.

To mine this data, we discussed different approaches to present data time break-down results.

1. boxplots are compact representations of distributions : the central line is the median and the upper and lower of the box the 25th and 75th quantiles. Extreme values - far from the waist of the distribution - are reported as crosses.

First we plot for Cellular, FTTH and ADSL trace median values obtained from data break-down values in relative and absolute values. It allows to have a global picture of the impact of access technologies through Theoretical times, client and server behaviors with Warm-up times and finally Pacing, which merges application and access limitation.

To go farther in our analysis we proposed clustering techniques to group connections with similar characteristics. Through this clustering technique, we plane to tackle the issue of comparing networking applications over different access technologies. It automatically extracts the impact of Warm-up, Pacing and Theoretical times from passively observed TCP transfers and group together, with an appropriate clustering algorithm, the transfers that have experienced similar performance over the three access technologies.

In the next Chapter, we validate key elements of our analysis method of data time break down time and clustering. This validation is achieved through simulations carried out using mixed scenarios with one or more applications. Then, we test the ability of the proposed clustering methods to lead to clusters that can be easily related to the expected behavior of the service under study.

Chapter 6

Validation of Data Time Break-down and Clustering Techniques

6.1 Introduction

The main objective of this chapter is to validate key elements of our analysis method, namely the data time breakdown approach and the clustering technique, introduced in Chapter 5. This validation is achieved through simulations carried out using the Qualnet simulator [1].

As the data time breakdown (see Chapter 5) relies on the estimation of the RTT, we also validate our RTT estimation technique, introduced in Chapter 3.

We create a variety of scenarios, which correspond to different link latencies or client think times (time spent at the client side to interact with the application and to perform queries) using several TCP application models available in Qualnet : FTP, TELNET and HTTP. In particular, we show that our clustering approach naturally groups clients with similar profiles at the application (e.g. similar Warm-up or Pacing) or network layers (e.g. similar RTT).

Those controlled experiments also allow to illustrate the different throughput definitions that we introduced : the Application Layer (AL) and the Effective-Exchange (EE) throughputs.

6.2 Network Setup

We designed several scenarios to reflect different Internet user behaviors like varying thinking times (time needed at the client side before performing a request to the server) or network conditions like rate limitation or large RTT. Simulations were carried out under a Fedora Linux (kernel 2.6.22) environment, using the QualNet 4.5.1 simulator.

As shown in Figure 6.1, the reference topology is a wired network comprising two sites : a local site, which consists exclusively of client machines with wired access to the network and a remote site with application servers and also wired and wireless clients. On the two sites, access points, wired clients and servers are inter-connected using a switch directly connected to a global router, in order to ensure inter-site connectivity.

We vary parameters like latency, access link capacity and other configuration parameters depending on the simulation scenario.

1. QualNet is a commercial simulator. It is based on GloMoSim developed at the University of California, Los Angeles (UCLA). GloMoSim uses the Parallel Simulation Enviroment for Complex Systems (PARSEC) for basic operations. QualNet has a graphical user interface for creating the model and its specification [95].

In all scenarios, tcpdump traces are collected at the server side. It is an advantage of Qualnet to generate tcpdump traces. However, the latter can be captured at a device that feature a TCP layer, i.e. a client or a server only.

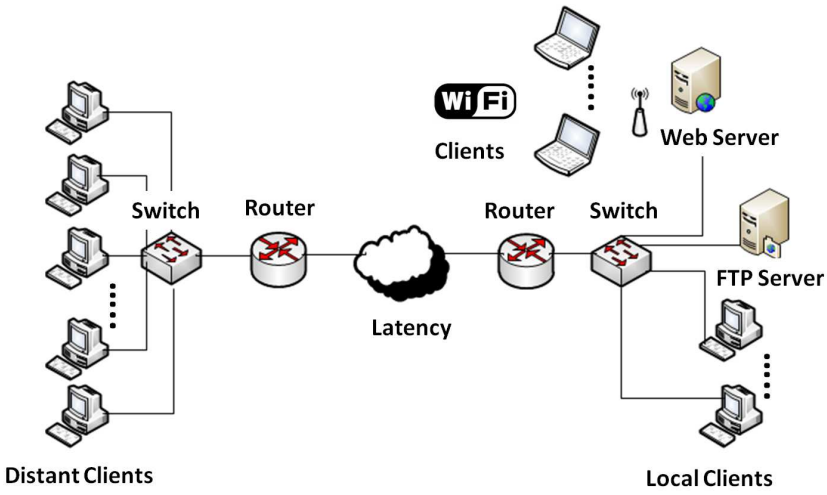
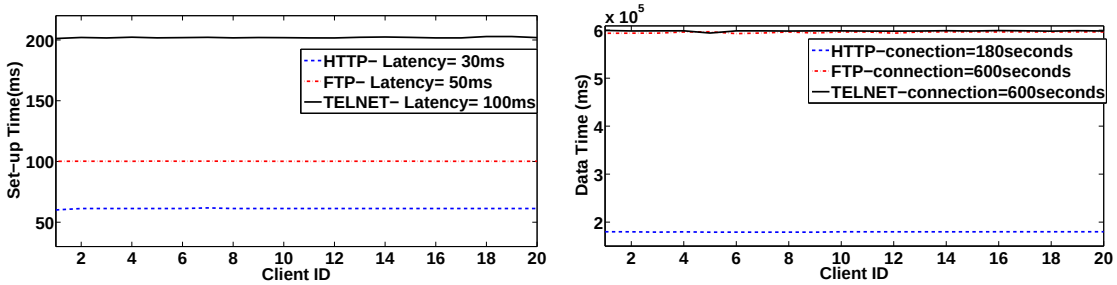


FIGURE 6.1 – Used Simulation Network

6.3 Macroscopic Connection Time Break-down : Set-up and Data Time

Our objective here is to assess the ability of our techniques to assess the set-up and data times of the transfers. These are relatively easy tasks, which do not represent a major challenge for our tools, but it also enabled us to validate the way the simulator is working.

In this scenario, we use File Transfer Protocol (FTP), HTTP and TErminal NETwork (TELNET) servers and wired clients, which post requests to those servers resulting in different connection sizes. We tune link latencies so that 20 clients on the local site observe respectively 30, 50 and 100 milliseconds when accessing HTTP, FTP and TELNET servers. The duration of HTTP session (consisting of a single transfer) was set to 180 seconds, while the durations of FTP and TELNET connections were set to 600 seconds.



(a) Set-up (b) Data-Transfer

FIGURE 6.2 – Connection Time Break Down

Figure 6.2 shows the connection break-down results for each of the 20 clients. From the left plot, we notice that set-up time (time between the first SYN packet and first data packet) is strongly

correlated with RTT for all observed protocols. It means that FTP, HTTP and TELNET clients tend to start data transmission just after the TCP connection establishment, in line with what is observed in the wild.

The right plot shows effective data transmission time, i.e., time between the first and the last transmitted data packet. We notice, for all clients and observed protocols, that due to short set-up values, the connection time corresponds approximatively to data transfer time. To finish with connection time break-down, we note that we omit to report tear-down here, which is defined as the time between the last data packet and the last control (in general FIN or RST) control packet, due to the way tear down phases are implemented in Qualnet. For instance, with QualNet doing an experiment with HTTP connection of 180 seconds means that a client establishes connection and sends data until the end of 180 seconds without closing the connection at the end.

In the next section, we start zooming into the data time of the transfer.

6.4 Microscopic Connection Time Break-down : Think and Data Preparation Time

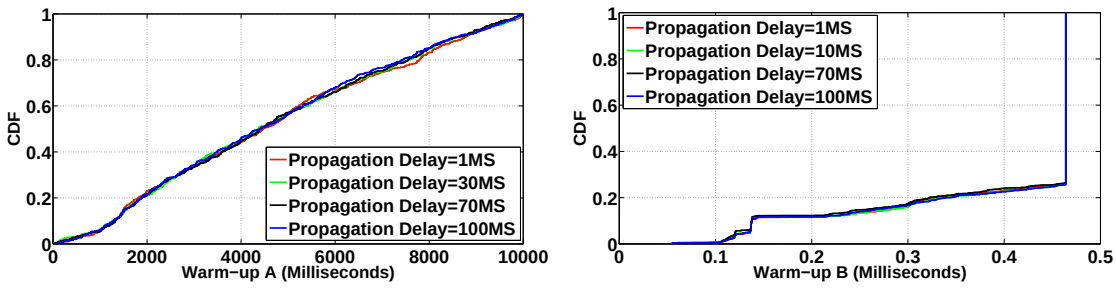
We address the problem of validating our data-time breakdown approach using simulation results and also measurements collected in the wild. While Qualnet allows to mimic specific users and servers behaviors, some details concerning the underlying distributions used to implement those behaviors are not externally visible. For instance, while one can specify different maximum thinking time, the distribution of thinking time is not specified in the Qualnet manual. In such a context, we use an indirect approach to validate our approach by varying some parameters, e.g. the link latency, while leaving other constant, e.g., the maximum thinking time, and check that our estimation of the constant parameters is unaffected by the variation of the other parameters.

We further present, in a second part of this section, some results obtained in the wild where we observe that some parameters, e.g. the server behavior, is unaffected by the type of clients (we consider clients using different access technologies : Cellular, ADSL and FTTH) that make the request. While indirect, we believe that this approach greatly increases the confidence one can have in our analysis tools.

6.4.1 Simulation results

We consider the topology of Figure 6.1 with 20 clients (on the local site) targeting a single Web server (on the remote site).

We first vary the link latency in the following range of values : 1, 10, 70 and 100 milliseconds. The maximum thinking time is kept fixed at 10 seconds. Duration of each simulation was 600 seconds.



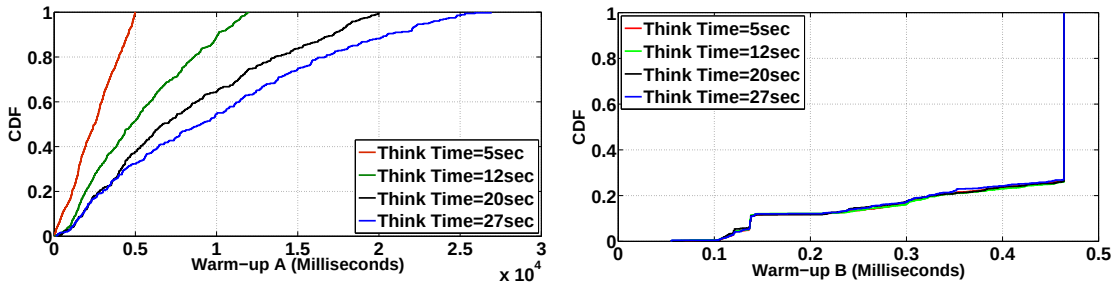
(a) Client (b) Server

FIGURE 6.3 – Different Link Delay

Figure 6.3, shows results of estimated think time at the client side (Warm-up A) and processing time at the server side (Warm-up B) for the different link delay values. Values for Warm-up A and B were computed for each exchanged train of data, see Section 5.2.

Results in Figure 6.3(a) irrespectively show the link delays, the distributions of computed think time on the client side remains the same (it looks approximatively as a uniform distribution between 0 and the maximum thinking time, which is 10 s here). This constitutes a strong argument for our methodology for computing Warm-up A values.

To get a clearer picture of what is happening on the server side, we computed (using our data time break-down methodology) the processing time on the server side, for each transmitted train of data. These results are plotted in Figure 6.3(b). Again, we observe that the estimated distribution remains the same for all link latencies, which complies with the idea that server thinking time is a property of the application and not of the link characteristics.



(a) client (b) Server

FIGURE 6.4 – Different Think Time

In a second stage, we used the same simulation setup but we now set the link delay to 10 milliseconds and we varied the think time of web clients. Recall that a web session lasts 600s and consists of a single long connection, where clients post requests and the web server sends the corresponding objects. Thinking and processing times are shown in Figure 6.4. These new experiments are consistent with our earlier results. Figure 6.4(a) shows that for clients with different think time, we obtain different distributions of computed Warm-up A, each following approximatively a uniform distribution. As for processing time, we observe in Figure 6.4(b) same distributions of Warm-up B irrespectively of the client behavior. The short estimated processing times (less than a millisecond) on the server side further underscore the accuracy of our approach for estimating Warm-up times.

6.4.2 Real-life Traces

In the above section, we observed that absolute values of Warm-up B should not be correlated neither with user think time nor with link latency. This is in line with intuition and with what should be observed for real traffic : if we assume an homogeneous implementation of a service and similar load conditions at the server side, Warm-up at the server side should have a similar distribution.

We present measurements obtained from the study of traces collected by Orange for different heterogeneous environments : ADSL, Cellular and FTTH. Our focus is on the study of POP3 traffic for Orange clients and Orange's mail servers.

We report in Figure 6.5, CDFs of each Warm-up at server side (time to prepare the answer for the client) for each access technology. It shows that, despite the diversity in access technology, using our data time break-down methodology, we are able to retrieve very similar distributions of

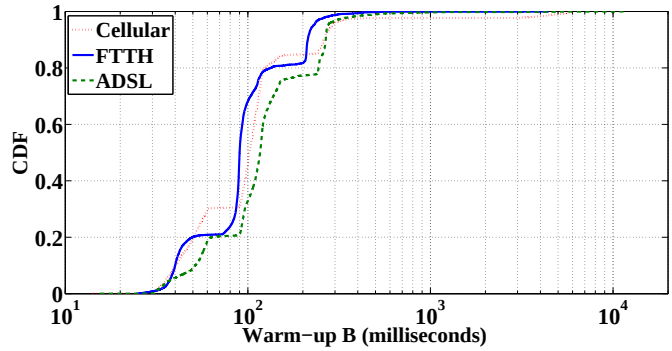
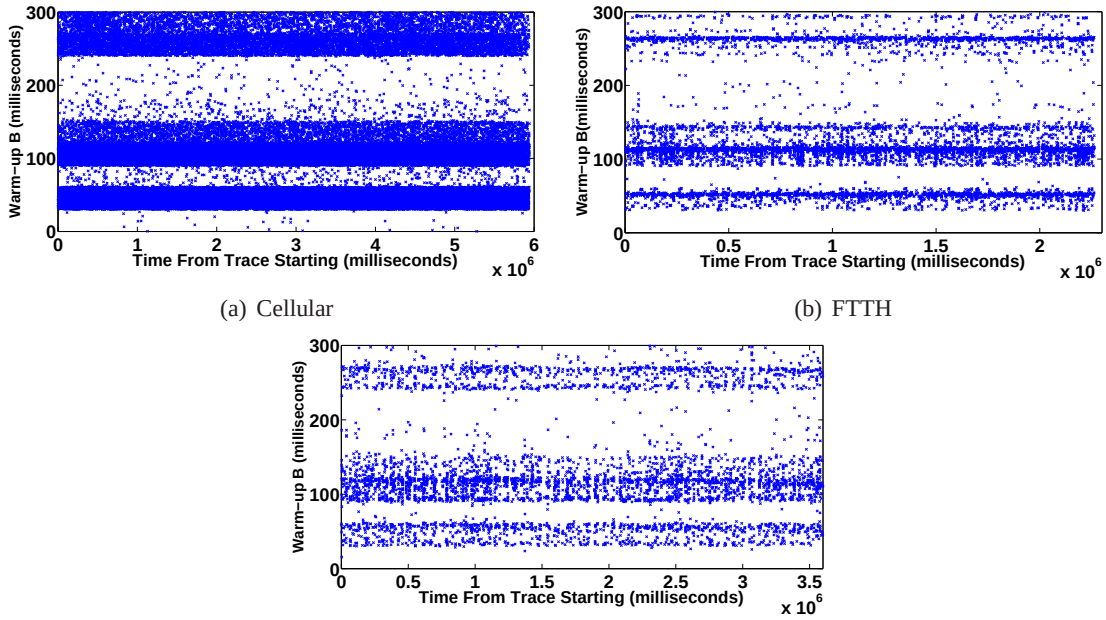


FIGURE 6.5 – Server's Warm-up Time : Orange POP Service

data preparation for each technology. Note that the traces that we focus on, were not captured at the same time period and thus, the load conditions might explain the little differences observed in the CDFs.



(a) Cellular

(b) FTTH

(c) ADSL

FIGURE 6.6 – Warm-up Time Series : POP3 Orange

To better understand the root cause of the peaks in the distributions in Figure 6.5, we inspected the time series of Warm-up B values. Figure 6.6 depicts the time series of warm-up for each access technology. A key observation is that the presence of peaks in Figure 6.5. They do not seem to be time dependent (because of load variations) but rather application dependent, as we believe it represents the service times of different type of interactions between the client and the server, e.g., authentication, (empty) mail box checking, etc.

To sum up, we have presented in this section different results obtained through simulations or real-life examples that validate, even if indirectly, our data-time breakdown methodology. We next turn our attention to our clustering approach.

6.5 Clustering Validation

6.5.1 Single Application Scenario

In this section and in the remaining of this chapter, we present validation results obtained using simulation and the network topology of Figure 6.1. In this section, we consider exclusively Web traffic. Twenty Web clients on the local site are interacting with a Web server on the remote site.

We implemented different scenarios corresponding to different client behaviors, different network conditions, or different transport layer parameters. These scenarios are detailed in Table 6.1. Each scenario is executed sequentially in order to avoid server or network overload. For a given scenario, several connections from the different clients are simultaneously active but the global load remains moderate and we observed no impact in terms of losses or time-outs at the TCP layer.

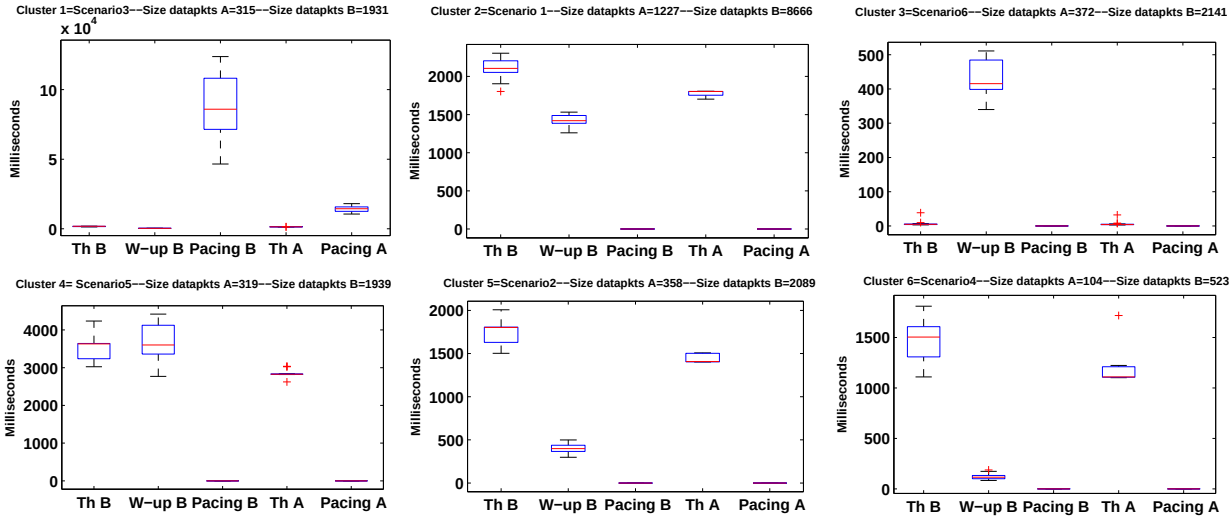
Once each scenario has been executed, we group all the traces into a unique trace consisting of all the observed connections. We next apply our data time break down technique : each well-behaved HTTP connection is transformed into a point in a 6-dimensional space (Pacing, Theoretical and Warm-up time of the client and the server). We then apply our clustering technique on the aggregate trace. To avoid biases introduced by think time at the client side, we omit to use Warm-up A values in the clustering. Indeed, thinking time at the client side represent large values as compared to the other dimensions and tend to dominate in the clustering phase.

Our clustering technique is unsupervised. We use the K-means algorithm. A key issue when using K-means is the choice of the initial centroids and the number of clusters targeted. To address the problem of the choice of the initial centroids, we run the K-means algorithm several times (100 times, which is considered as a good practice) with different initial centroids and pick the best results in terms of distance between clusters. To guide the choice of the number of centroids, we consider two options : either we use a test and trial approach were we start with an initially number of clusters and then reduce this number as long as insignificant clusters remains ; or we rely on a dimensionality reduction technique, t-SNE, that projects multi-dimensional data on a two dimensional plane. Figure 6.7(b) shows the application of t-SNE on the global trace (we explain the colors later) and suggest that 3 to 6 clusters are present in our aggregate trace.

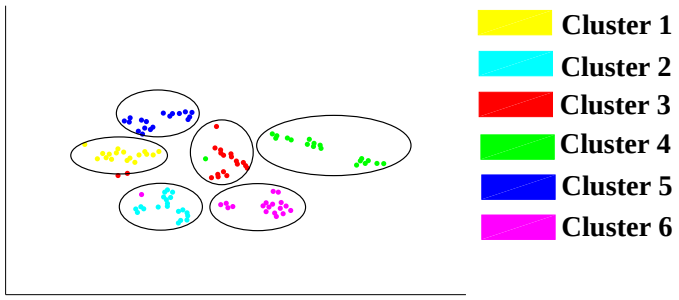
	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Application	HTTP	HTTP	HTTP	HTTP	HTTP	HTTP
Connection Time (sec)	600	600	600	180	600	600
Bandwidth (Mbps)	10	10	10	10	1	10
Link Delay (ms)	50	50	10	10	100	wifi(802.11b)
MSS	1460	1460	1460	1460	1460	1460
Sender Buffer Size	64500	64500	16000	64500	64500	64500
Receiver Buffer Size	64500	64500	16000	64500	64500	64500
Think Time (sec)	2	10	10	10	10	10

TABLE 6.1 – Scenarios Configuration : Different Delays

We focus hereafter on the clustering with 6 clusters, equal to the number of scenarios we have. Figure 6.7(a) depicts the characteristics of the 6 clusters obtained with K-means. We use boxplots to obtain compact representations of the values corresponding to each dimension. We indicate, on top of each cluster, to which scenario the connections in the cluster correspond to, and the median size of the number of data packets from and to the clients. We first observe by inspecting those labels that each identified cluster corresponds to a unique scenario. Cluster 1 corresponds to scenario 3 and groups connections characterized by large Pacing B due to sender and receiver buffer size limitation. Cluster 3 aggregates connections from scenario 6, with short Pacing and Theoretical times. Those are penalized by large processing time at the server side. Cluster 4 groups



(a) K-means



(b) t-SNE

FIGURE 6.7 – HTTP scenarios : Data Clustering

connections with large Theoretical and Warm-up B times, which are the connections in scenario 5 that correspond to slow access links (1Mb/s) with high latency (100ms). Cluster 6 groups the shortest connections corresponding to scenario 5.

Figure 6.7(b), that present the projection obtained by t-SNE, further demonstrates that t-SNE and K-means are in are in good agreement as the data samples in the figure are indexed using their cluster identifier obtained from K-means.

The above results were obtained for a number of clusters that corresponds to the exact number of scenarios. We also tested what happens if we reduce the number of clusters in K-means (remember that t-SNE suggested that it should not be larger than 6).

If we decrease the number of cluster to 4, we also obtain satisfying results : in this case, clusters 1, 3 and 4 remain unchanged and clusters 2,5 and 6 will be grouped together. This is because clusters 2, 5 and 6 have similar absolute times along each dimension of the data time break-down procedure.

6.5.2 Heterogeneous Scenario

In this section, we continue the validation of our clustering technique that we started in the previous section. We consider again the topology in Figure 6.1. We created classes of users corresponding to different applications. We employed a sensitivity analysis, which exposes the capacity of our methodology in finding significant categories of traffic and delineates performance problem from an original mixed traffic.

Table 6.2 summarizes the key characteristics of the 3 classes (FTP, TELNET and HTTP) of users we use. HTTP and TELNET traffic is bi-directional in Qualnet while FTP traffic is unidirectional. For all classes of clients, the duration of each connection is set to 600 seconds, the access bandwidth to 10 Mbps and the link delay to 30 ms. For each application, we designed two cases. An optimal case where the parameters of client/connection (MSS and client/server buffer size) allow to reach good performance and a non-optimal case where we limit the MSS and client/server buffer size.

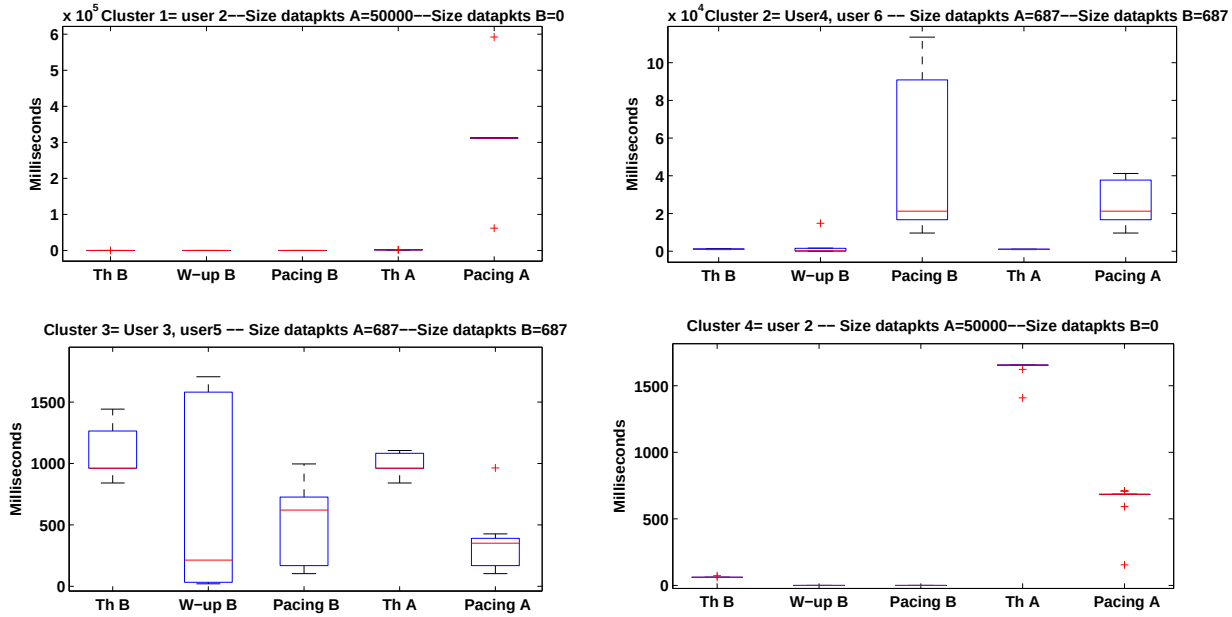
	users 1	users 2	users 3	users 4	users 5	users 6
Application	FTP	FTP	TELNET	TELNET	HTTP	HTTP
Connection Time (sec)	600	600	600	600	600	600
Bandwidth (Mbps)	10	10	10	10	10	10
Link Delay (ms)	30	30	30	30	30	30
MSS	1460	1460	1460	65	1460	1460
Sender Buffer Size	64500	64500	64500	64500	64500	16000
Receiver Buffer Size	64500	16000	64500	64500	64500	16000
Think Time (sec)	-	-	-	-	2	2

TABLE 6.2 – User Classes

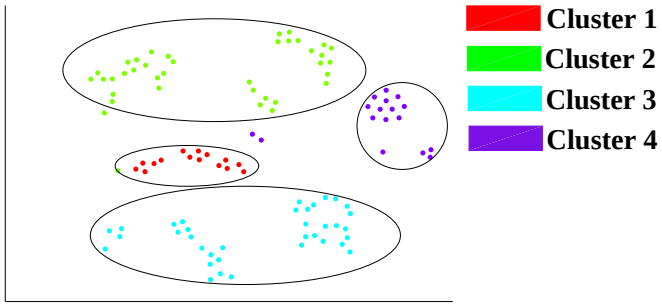
Figure 6.8 presents the clustering results using K-means and the projection obtained via t-SNE. The first observation here is that the clusters obtained with K-means are in good agreement with the projection obtained by t-SNE as indicated in Figure 6.8(b), where data samples are indexed using their cluster identifier in K-means.

Before moving to the interpretation of the individual cluster, we observe that two of them gather the interactive traffic while the two other ones gather the bulk transfers. Indeed, Figure 6.8(a) shows that cluster 2 and 3 correspond exclusively to HTTP and TELNET connections while clusters 1 and 4 correspond to FTP traffic with median transfer size of 50000 data packets. Cluster 1 corresponds to FTP connections characterized by large Pacing A value due to the limited receiver buffer size for users of class 1. Cluster 2 groups TELNET and HTTP connections with large Pacing A and B values : in fact users of class 4 were penalized by very short MSS size and users of class 6 by limited client and server buffer size. Let us now consider clusters 3 and 4. Those clusters correspond to shorter data time break-down values, while they present the same amount of data as in clusters 1 and 2 respectively. Cluster 3 shows that Web and TELNET connections are feature high Warm-up B when MSS as their sender/receiver buffer size are optimal. It means that when connection is optimally tuned, the impact of processing times is more important than Pacing and theoretical times. Finally cluster 4 corresponds to FTP transfers with low Pacing A as those users have large sender/receiver buffer size.

Overall, we observe our clustering method, when applied to traffic profiles with different connections parameters, lead to clusters that can be easily related to the expected behavior of the service under study (clusters 3 and 4) while some others will relate to anomalous behavior because of non optimal setting of some parameters (clusters 1 and 2).



(a) K-means



(b) t-SNE

FIGURE 6.8 – Heterogeneous Traffic : Data Clustering

6.6 Comparison with RCA Technique

In [2], the authors develop a methodology similar to ours. They identify four types of throughput limitations for connections : (i) unshared bottleneck limitation that corresponds to the case where a single connection uses the full capacity of the bottleneck link, (ii) shared bottleneck limitation, which occurs when several connections share a bottleneck link. (iii) receiver window limitation, if ever the receiver window is too small as compared to the bandwidth-delay product of the path, which prevents the sender to achieve a higher throughput and finally (iv) sender buffer limitation, if the sender buffer is too small (rare case in practice).

In our simulations in Sections 6.5.2 and 6.5.1 we reproduced different scenarios of performance limitation. To degrade client throughput, we modified link capacity/latency, MSS, sender/receiver buffer size and think time for HTTP client.

We observed that, for HTTP transfers, connections with small receiver buffer size feature an increase of Pacing A and large value of Pacing B and globally fewer data than connections with buffer size of 64500 bytes. Small sender buffer size leads also to an increase of Pacing A and large Pacing B. This means that our technique can not distinguish these two cases – receiver or sender window limitations. We note that in QualNet, for the case of HTTP transfers, if we focus on the number of exchanged data, the server sends more data packets than the client, which explains the large values of Pacing B. In contrast, when we perform experiments with FTP traffic from the client to the server (upload), results show that small receiver and sender buffer size lead to large Pacing A.

Hence, we conclude that (i) if more data is transferred from the client to the server, small sender/receiver buffer size lead to large Pacing A (ii) if more data is transferred from the server to the client, small sender/receiver buffer size lead to large Pacing B.

Simulations of unshared bottleneck limitation show that for this case, TCP connections present large Pacing values. Finally, with shared bottleneck limitation, as defined in [2], connections experience a high loss ratio and are penalized by large retransmission time. In our simulations we excluded the recovery time from the data transfer time to avoid biases when we compute Theoretical, Pacing and Warm-up times. However, we account for this metric in our global connection time break-down, and we are thus also able to infer this limitation.

6.7 Throughput Computation Methods

This last section does not present any validation result but rather illustrates the different metrics we have proposed to measure the throughput using simulation, which offers a controlled environment.

We use the same architecture as presented in Figure 6.1. We simulate two scenarios of Web clients with different think time values, respectively 2 and 10 seconds. Table 6.3 presents the main configuration parameters for each scenario. We use 20 clients interacting concurrently with a single Web server. Consequently, the client perceived performance can depend on the load at the Web server and also the bottleneck link load. The purpose of this experiment was to compare the performance of users with different think times and the same amount of exchanged bytes. Since users in scenario 2 have 10 seconds of think time limit, against only 2 seconds for scenario 1, Table 6.3 shows that we used longer connections for scenario 2.

	Scenario 1	Scenario 2
Application	HTTP	HTTP
Connection Time (sec)	600	2500
Bandwidth (Mbps)	10	10
Link Delay (ms)	30	30
MSS	1460	1460
Client Data Packets (mean)	1343	1477
Server Data Packets (mean)	9955	10849
Think Time (sec)	2	10

TABLE 6.3 – Scenarios Configuration : Different Think Times

Figure 6.9 presents performance results for each scenario. For all configurations, we report two metrics related to the throughput of the transfers : (i) AL throughput, which corresponds to the amount of bytes transferred at the TCP layer, divided by the total duration between the first packet (first SYN) and last data packet of the transfer ; (ii) the EE throughput that corresponds to the amount of bytes transferred at the TCP layer, divided by the total duration between the first packet (first SYN) and last data packet of the transfer minus the cumulated think time at the client.

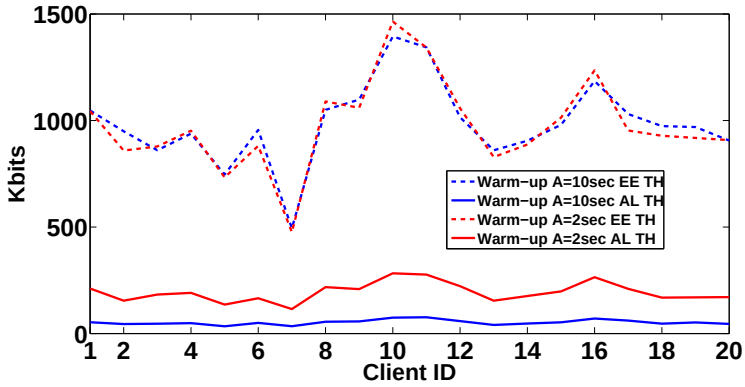


FIGURE 6.9 – Throughput Estimation

A comparison of AL throughput in Figure 6.9 shows that user throughput in scenario 1 clearly outperforms the throughput in scenario 2, which was to be expected from its definition as the AL throughput is sensitive to the end user behavior. On the other hand, the EE throughput conveys the message that the performance achieved during the actual transfer times is essentially the same in both scenarios.

6.8 Conclusion

We validated the ability of our approach of data break-down to study the interplay between TCP connection factors such as : the applications on top of TCP, clients behavior, the application usage, and the access technology. We designed several scenarios to reflect different Internet user behaviors with the Qualnet Simulator. We based our analysis on a topology comprising two sites : a local site, with wired client machines and a remote site with application servers and also wired and wireless clients. We then vary network parameters like latency, access link capacity and client/application server configuration parameters depending on the simulation scenario.

First, we validated the process of computing thinking time at the server side (what we called previously in 5.2 Warm-up B) for each transmitted train of data. Through different results from simulations or real-life examples, we show that the estimated distribution of thinking time at the server side remains the same for all link latencies and access technologies, which complies with the idea that for a considered service, server thinking time is a property of the application and not of the link characteristics.

Then, we test the ability of used clustering methods to lead to clusters that can be easily related to the expected behavior of the service under study. To do that, we implemented different scenarios corresponding to different client behaviors, different network conditions or different transport layer parameters. Results show that clusters obtained with K-means are in good agreement with the projections obtained by t-SNE.

Finally, we compare results of our data time-break down methodology with the RCA Technique. We discuss how limitations identified in RCA are handled with our methodology. We propose different metrics : the Application Layer (AA) and the Effective-Exchange (EE) throughputs, to measure the throughput and to avoid biases introduced by the time introduced by the client to interact with the application/service.

In the next chapter, we apply our methodology to real traffic traces. In particular, we focus on the Google Web search service, being accessed by different access technologies.

Chapter 7

A Fine Grained Analysis of TCP Performance

7.1 Introduction

Telecommunication operators offer several technologies to their clients for accessing the Internet. We have observed an increase in the offering of Cellular and FTTH accesses, which now compete with the older ADSL and cable modem technologies. However, until now it is unclear what are the exact implications of the significantly different properties of these access technologies on the quality of service observed by clients.

In this chapter, we address the problem of comparing the performance perceived by end users when they use different technologies to access the Internet. Users primarily interact with the network through networking applications they use. We tackle the comparison task by focusing on several Internet key services such as Google search and mail. This is because we focus on user perceived performance and users do care about raw performance metrics. They care about the performance of the applications they use. Similarly to what we did for the overall traffic in Chapter 4 we first demonstrate when focusing on Google search traffic that RTT and packet loss alone are not enough to fully understand the observed differences or similarities of performance between the different access technologies. We then apply our data break-down approach, detailed in Chapter 5, based on a fine-grained profiling of the data time of transfers that sheds light on the interplay between service, access and usage, for the client and server side. We use clustering approaches, introduced in Chapter 5, to identify groups of connections experiencing similar performance.

7.2 The Case of Google Search Traffic

7.2.1 Problem Statement

	Cellular	FTTH	ADSL
Well-behaved Cnxs	29874	1183	6022
Data Packets Up	107201	2436	18168
Data Packets Down	495374	7699	139129
Volume Up(MB)	74.472	1.66	11.39
Volume Down(MB)	507.747	8	165.79

TABLE 7.1 – Google Search Traffic

To identify traffic generated by the usage of Google search engine, we adopted the following

approach : we first extract HTTP requests containing the string www.google.com/fr in their HTTP header.

We paid attention to excluding requests to or from other services offered by Google like gmail, Google map, etc. This first step provides a set of pairs of IP addresses and source/destination ports identifying the local client and the remote Google Web server. We then flagged all connections between those pairs of IP addresses and source/destination ports, as Google Web search traffic. To identify Google search traffic for the upstream and downstream directions, we use TCP port numbers and remote address resolution (Nslookup). Table 7.1 summarizes the amount of Google search traffic we identified in our traces.

7.2.1.1 Connection Size

Figure 7.1 depicts the cumulative distribution of well-behaved Google search connection size in bytes. It appears that data transfer sizes are very similar for the three access technologies. This observation constitutes a good starting point since the performance of TCP depends on the actual transfer size. RTTs and losses also heavily influence TCP performance, as the various TCP throughput formulas indicate [89, 96]. Also, the available bandwidth plays a role. With respect to these metrics, we expect the performance of a service to be significantly influenced by the access technology since available bandwidth, RTTs [1] and losses are considerably different over ADSL, FTTH and Cellular. However, as we demonstrated in Chapter 4 and in the remaining of this section, those metrics alone fail at fully explaining the relative performance observed in our traces.

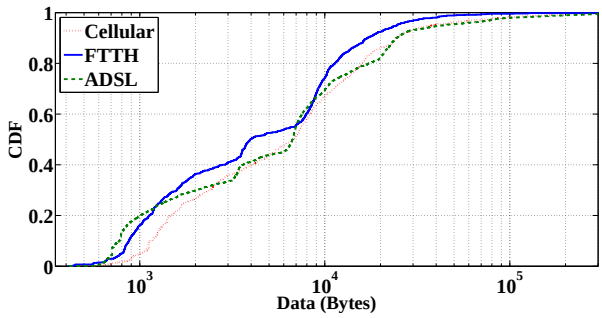


FIGURE 7.1 – Connection Size

7.2.1.2 Latency

Figure 7.2 depicts the resulting RTT estimations for the 3 traces, connecting only to the Google Web search service. It clearly highlights the impact of the access technology on the RTT. FTTH access offer very low RTT in general – less than 50 ms for more than 96% of connections. This finding is in line with the characteristics generally advertised for FTTH access technology. In contrast, RTTs on the Cellular technology are notably longer than under ADSL and FTTH.

7.2.1.3 Packet Loss

Figure 7.3 depicts the cumulative distribution of retransmission time per connection for each trace. Retransmissions are clearly more frequent for the Cellular access with more than 35% of

1. As noted in several studies on ADSL [29]. See also Appendix A where we contrast what we call local and remote RTT (see Figure A.1) and Cellular networks [56], the access technology often contributes a significant part of the overall Round Trip Time.

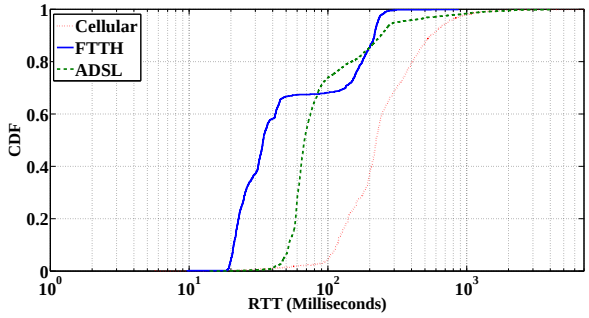


FIGURE 7.2 – RTT Estimation

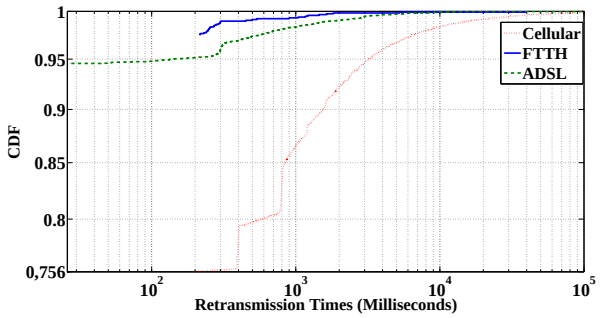


FIGURE 7.3 – Retransmission Time per Loss Event

transfers experiencing losses compared to less than 6% for ADSL and FTTH accesses. From previous works, we noticed that several factors explain the higher loss ratio for Cellular access. Note again that the metric we consider here is not the loss rate but the fraction of connections that experience losses. The overall loss rates are small, in the order of a few percent at most on all access technologies here. In fact, in [90] authors recommend to use a loss detection algorithm, which uses dumps of each peer of the connection (this algorithm is not adapted for our case because our measurements have been collected at a GGSN level) to avoid spurious Retransmission Timeouts in TCP. In addition, authors report in [56] that spurious retransmission ratio in Cellular networks is higher for Google servers than others. For Google servers, authors show short retransmission timeouts.

Most of the transfers are very short in terms of number of packets and we know that for such transfers, packet loss has a detrimental impact to the performance presented in Chapter 2. Thus, the performance of these transfers are dominated by the packet loss. It is important to note that using our data time break-down approach, we analyze all connections, including the ones that experience losses by first removing recovery times from their total duration.

7.2.1.4 Application Level Performance

Our study of the two key factors that influence the throughput of TCP transfers, namely loss rate and RTT, suggest that, since Google Web search transfers have a similar profile on the 3 access technologies, the performance of this service over FTTH should significantly outperform the one of ADSL, which should in turn outperform the one of Cellular. It turns out that reality is slightly more complex as can be seen from Figure 7.4 where we report the distribution of transfer times. Throughput analysis is qualitatively similar, but we prefer to report transfer times since Web search is an interactive service. Indeed, while the Cellular technology offers significantly longer

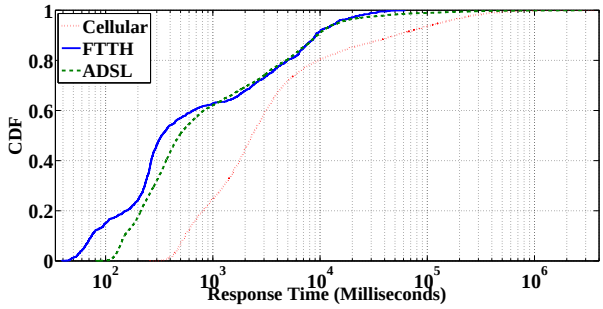


FIGURE 7.4 – Google Transfer Time

response time, in line with RTT and loss factors, FTTH and ADSL have much closer performance than RTT and loss were suggesting.

In the next section, we apply our data time break-down approach to uncover the impact of specific factors like the application and the interaction with user, and thus informs the comparison of access technology, for Google search traffic.

7.2.2 Break-down Results

The analysis method that we use consists in two steps as described in Chapter 5. In the first step, the transfer time of each TCP connection is broken down into several factors that we can attribute to different causes, e.g., the application or the end-to-end path.

At the end of step 1, each well-behaved Google search connection is transformed into a point in a 6-dimensional space (pacing, theoretical and train time of the client and the server). To mine this data, we use in a second step, a clustering approach to uncover the major trends within the different data sets under study.

Application of t-SNE to our 6-dimensional data leads to the right plot seen in Figure C.17(a). This figure indicates that a natural clustering exists within our data. In addition, a reasonable value for the number of clusters lies between 5 and 10. Last but not least the right plot of Figure C.17(a) suggests that some clusters are dominated by a specific access technology while some others are mixed. We picked a value of 6 for the number of clusters in Kmeans.

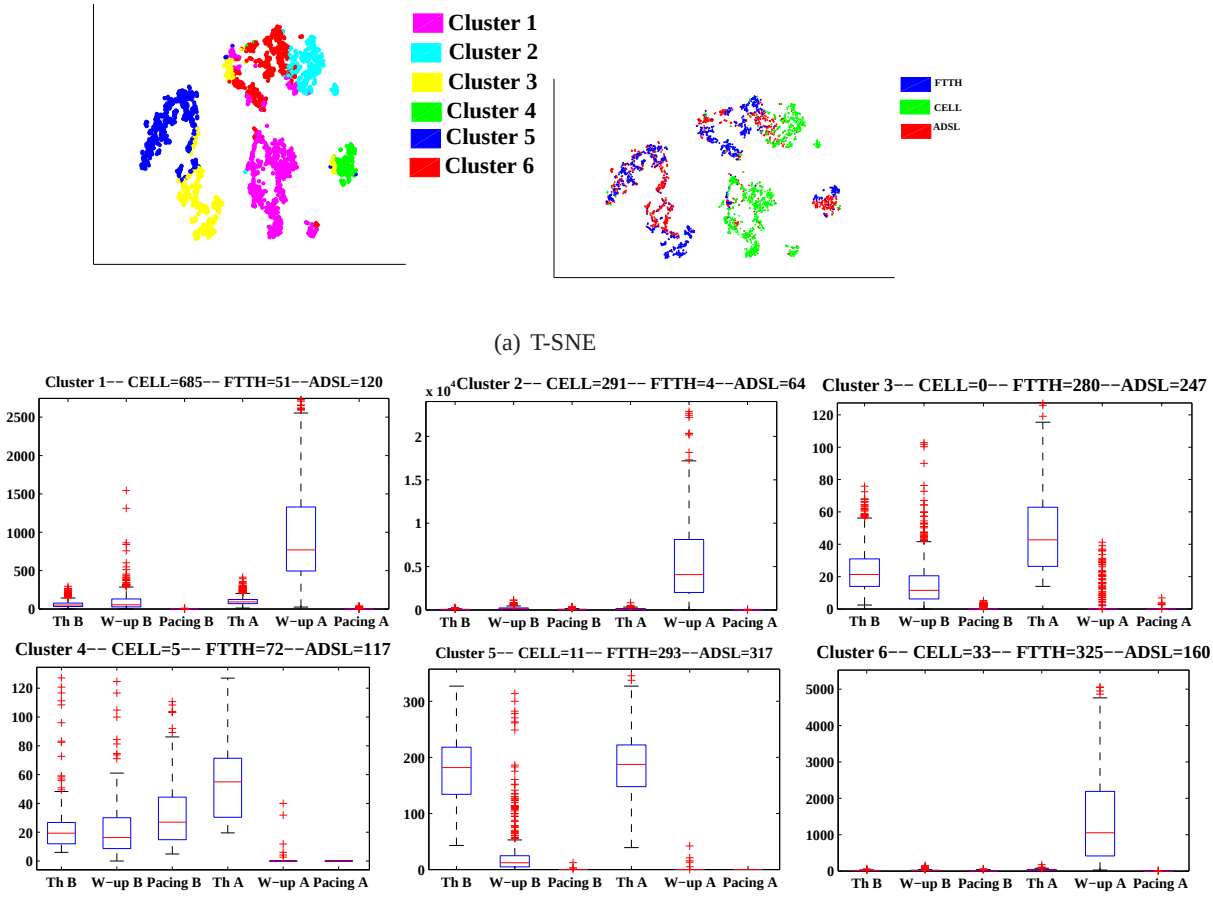
Figure C.17(b) depicts the 6 clusters obtained by application of Kmeans. We use boxplots [2] to obtain compact representations of the values corresponding to each dimension. We indicate, on top of each cluster, the number of samples in the cluster for each access technology. We use the same number of samples per access technology to prevent any bias in the clustering, which limits us to 1000 samples, due to the short duration of the FTTH trace. The ADSL and Cellular samples were chosen randomly among the ones in the respective traces. In Figure 7.6(b) we plot the size of the transfers of each cluster and their application layer throughput [3].

We first observe that the clusters obtained with Kmeans are in good agreement with the projection obtained by t-SNE as indicated in the left plot of Figure C.17(a), where data samples are indexed using their cluster id in Kmeans.

Before delving into the interpretation of the individual clusters, we observe that three of them carry the majority of the bytes. Indeed, Figure 7.6(a) indicates that clusters 1 and 2 and 6 represent

2. Boxplots are compact representations of distributions : the central line is the median and the upper and lower of the box the 25th and 75th quantiles. Extreme values -far from the waist of the distribution - are reported as crosses.

3. We compute the throughput by excluding the tear down time, which is the time between the last data packet and the last packet of the TCP connection

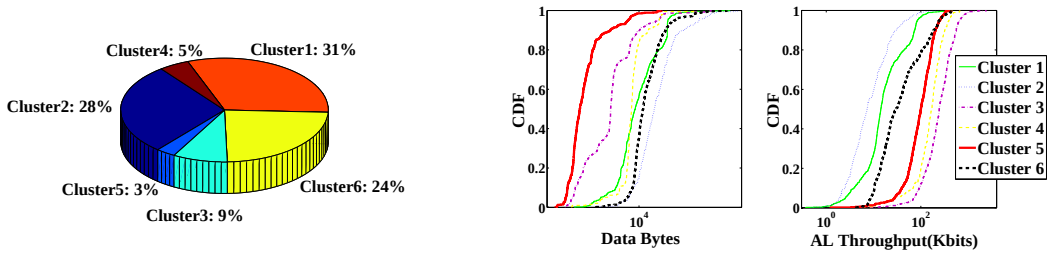


(a) T-SNE

(b) K-means

FIGURE 7.5 – Google Search Engine Clusters

83% of the bytes. Let us first focus on these dominant clusters.



(a) Bytes Volume

(b) Clusters Parameters

FIGURE 7.6 – Google Search Engine Parameters

Clusters 1, 2 and 6 are characterized by large warm-up A values, i.e., long waiting time at the client side in between two consecutive requests. The warm-up A values are in the order of a few seconds, which are compatible with human actions. This behavior is in line with the typical use of search engines where the user first submits a query then analyzes the results before refining further her query or clicking on one of the links of the result page. Thus, the primary factor that influences

observed throughputs in Google search traffic is the user behavior (how the client interact with the application). In fact, identified values in clusters 1, 2 and 6 of Warm-up A are in line with results in [97] of the time between query submission and first click, where authors identified different users trends.

We can further observe that clusters 1 and 2 mostly consist of Cellular connections while cluster 6 consists mostly of FTTH transfers. This means that the clustering algorithm first based its decision on the Warm-up A value ; then, this is the access technology that impacts the clustering. As ADSL offers intermediate characteristics as compared to FTTH and Cellular, ADSL transfers with large Warm-up A values are scattered on the three clusters.

Let us now consider clusters 3, 4 and 5. Those clusters, while carrying a tiny fraction of traffic, feature several noticeable characteristics. First, we see almost no Cellular connections in those clusters. Second, they total two thirds of the ADSL and FTTH connections, even though they are smaller than the ones in clusters 1, 2 and 6 – see Figure 7.6(b). Third, those clusters, in contrast to clusters 1, 2 and 6 have negligible Warm-up A values. From a technical viewpoint, Kmeans separates them based on the RTT as cluster 5 exhibits larger ThA and ThB values and also based on Pacing B values. A deeper analysis of these clusters revealed that they correspond to very short connections with an exchange of 2 HTTP frames. In fact, cluster 3 corresponds to cases when a client opens the Google Web search page in his/her Internet browser without performing any search request, then after a time-out of 10 seconds, the Google server closes the connection. On the other hand, cluster 4 and 5 correspond to GET requests and HTTP OK responses with an effective search, the main difference between cluster 4 and 5 being RTT and connection size.

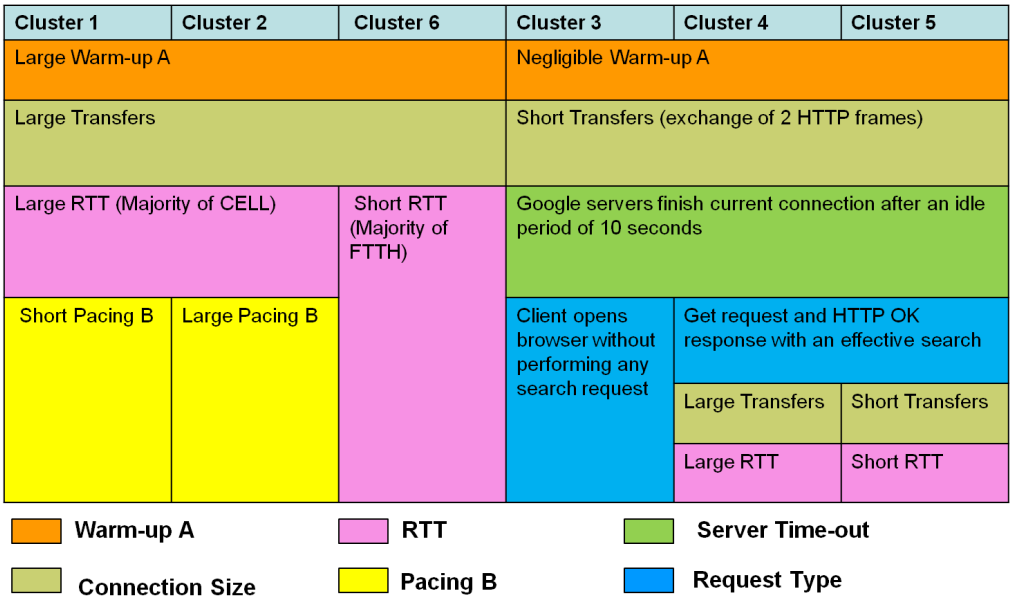


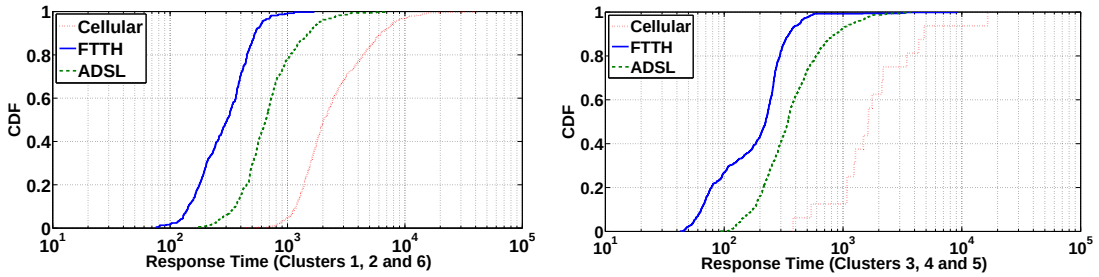
FIGURE 7.7 – Overview of Google Clusters

Our clustering results thus comply with intuition : in a Cellular environments, there is no -at the moment and in our trace - default opening of pages like Google search unlike for computers where this is often the default case. In the latter case, a time out occurs after a long idle period when the server decides to close the HTTP connection. This leads to clusters 3, 4 and 5 that we observe on ADSL and FTTH only. Cellular environments are optimized differently and it is only when the user issues a query that the Google server is accessed. Our clustering technique enables to pinpoint those different usages by a precise profiling of what is happening at the client and

server side.

Finally, to wrap-up results of data time break-down methodology, we present in Figure 7.7 the main characteristics and features that differentiate each cluster. As we can see, user behavior is the main discriminant factor between clusters 1,2 and 6 on the one hand and on the other hand clusters 3,4 and 5 followed respectively by the usage and access impact.

To consolidate our finding we report in Figure 7.8(a) response times for clusters 1, 2 and 6 without Warm-up A. Also, we report in Figure 7.8(b) response times for clusters 3, 4 and 5. The two figures match with each other and feature a similar shape as RTT distributions.



(a) Clusters 1, 2 and 6 (b) Clusters 3, 4 and 5

FIGURE 7.8 – Computation of Response Time Without Warm-up A

More generally, we expect that our method, when applied to profile other services, will lead to some clusters that can be easily related to the behavior of the service under study while some others will relate anomalous or unsual behaviors that might require further investigation. For the case of Google search engine, we do not believe clusters 3,4 and 5 being anomalies that affects the quality of experience of users since the large number of connections in those clusters would prevent the problem from flying below the radar. We found only very few cases where the server's impact to the performance was dominating and directly impacting the quality of experience of the end user. Observing many such cases would have indicated issues, e.g., with service implementation or provisioning.

Clustering results show that Warm-up A influences response time estimation and represents the most discriminant clustering parameter. In order to avoid the bias introduced by user behavior, we limit next our break down study to Warm up B, Pacing A/B and Theoretical A/B. As for previous results, we obtain 6 clusters and we observe that clusters obtained with Kmeans are in good agreement with the projection obtained by t-SNE.

We summarize in Figure 7.9 characteristics of each identified cluster, when performing clustering without taking into account Warm-up A.

A comparison of generated data volume per cluster, depicted in Figure 7.10(a) shows that cluster 5 contains the highest volume of data, while clusters 1, 2, 3 and 4 have between 12 % and 19 % of data volume, cluster 6 contains only 5% of generated Google Web search traffic.

Figure 7.9 shows that like for clusters presented previously (with Warm-up A) we have two classes of cluster with short and large transfers. Then large transfers are more penalized by Pacing B since they are more affected by access and application on top. It is important to note that we identified 3 categories of clusters based on their Warm-up B values. In fact from Figure 7.10(b) we observed 3 profiles of Warm-up B distributions, corresponding to : (i) short transfers (ii) Mobile devices with windows CE and iPhone and finally (iii) client with windows machines. This suggests that Google Web search Engine adapts content for mobile device. This hypothesis will be studied with more details in next Section 7.3.

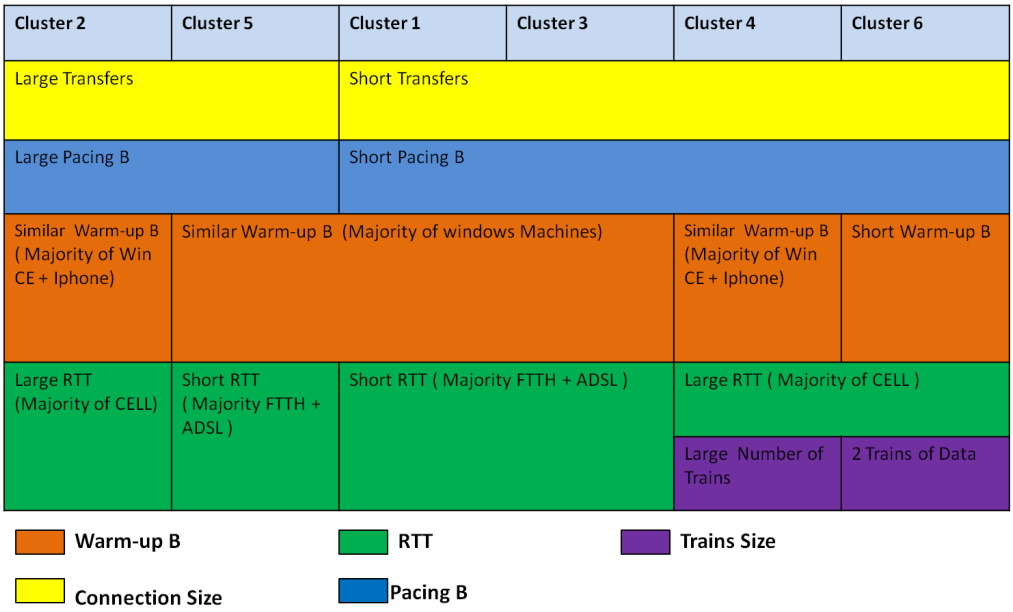
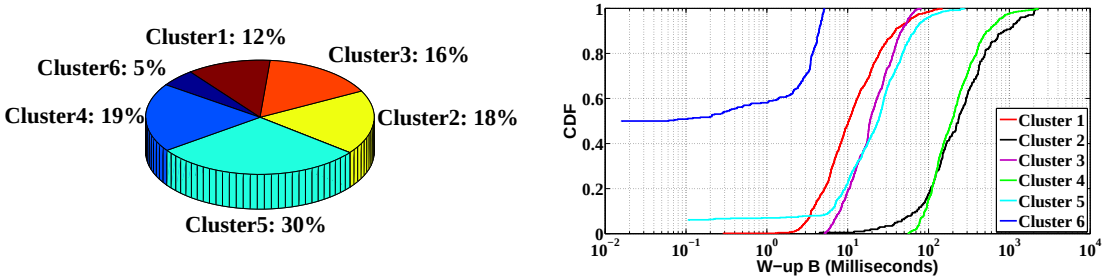


FIGURE 7.9 – Overview of Google Clusters Without User Behavior Impact



(a) Data Volume per Cluster

(b) Server Data Preparation

FIGURE 7.10 – Google Clusters Parameters (without Warm-up A)

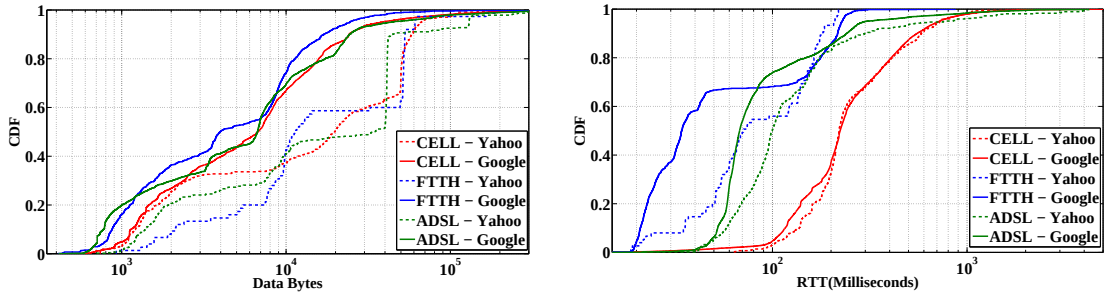
A comparison of clustering results with and without taking into account Warm-up A shows that while we obtain the same number of clusters, we obtain differences in cluster characteristics. Results show that user behavior plays an important role and corresponds to the discriminant parameters for first results in Figure 7.7. In the other hand, by factoring out client behavior impact, we noticed from Figure 7.7 that server policy, with an adaptation of data preparation time becomes the discriminant parameter in obtained clusters.

7.3 Contrasting Web Search Engines

The main idea in this section is to contrast Google results with others Web search services. For the case of our traces, we observed that the second dominant Web Search engine is Yahoo, with few connections. This low number of samples somehow limits the applicability of our clustering approach as used in the Google case. We restrict our attention to the following questions : (i) Do the two services offer similar traffic profile ? (ii) Are services provisioned in a similar manner ? The architecture of Google and Yahoo data-centers are obviously different but they must both obey

the constraint that the client must receive its answer to a query in a maximum amount of time that is in the order of a few hundreds of milliseconds [98]. We investigate the service provisioning by analyzing the Warm-up B values offered by the two services.

7.3.1 Traffic Profiles



(a) Connection Size (b) RTTs

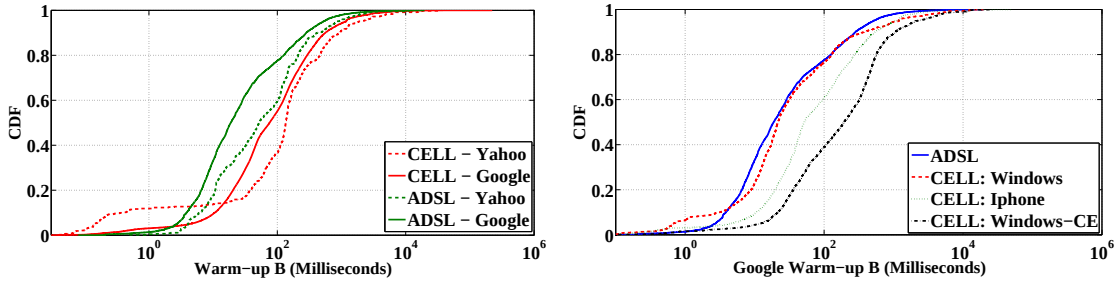
FIGURE 7.11 – Yahoo vs. Google Web Search Services

Figure 7.11(a) shows CDFs of data connections size for Cellular, FTTH and ADSL traces for both Google and Yahoo. We observe for our traces that Yahoo Web search connections are larger than Google ones. An intuitive explanation behind this observation is that Yahoo search pages contain, on average, more photos and banners than ordinary Google pages.

Figure 7.11(b) plots cdfs of RTTs. We can observe that RTT values on each access technology are similar for the two services, which suggests that the servers are located in France and that it is the latency of the first hop that dominates.

We do not present clustering results for Yahoo due to the small number of samples we have. However, a preliminary inspection of those results revealed the existence of clusters due to long Warm-up A values, i.e., long waiting times at the client side – in line with our observations with the Google Web search service. In the next section, we focus on the waiting time at the server side.

7.3.2 Data Preparation Time at the Server Side



(a) Yahoo vs. Google (b) Google

FIGURE 7.12 – Warm-up B

Figure 7.12(a) presents the cdf of warm-up B [4] values for both Yahoo and Google for the ADSL and Cellular technology (we do not have enough samples on FTTH for Yahoo to present

4. We have one total warm-up B value per connection, which is the total observed warm-up B for each train.

them). We observe an interesting result : for both Yahoo and Google, the time to craft the answer is longer for Cellular than for the ADSL technology. It suggests that both services adapt the content to Cellular clients. A simple way to detect that the remote client is behind a wired or wireless access is to check its Web browser-User Agent as reported in the HTTP header. This is apparently what Google does as Figure 7.12(b) reveals (again, due to a low number of samples on Yahoo, we are not able to report a similar breakdown). Indeed, Cellular clients featuring a laptop/desktop Windows operating system (Vista/XP/2000) experience similar warm-up B as ADSL clients while clients using iPhones or a Windows-CE operating system experience way higher warm-up B. As the latter category (esp. iPhones : more than 66% of Google connections) dominates in our dataset they explain the overall Cellular plot of Figure 7.12(a). Note that further investigations would be required to fully validate our hypothesis of content adaptation. We could think of alternative explanations like a different load on the servers at the capture time or some specific proxy in the network of the ISP. In [99] authors show that market leaders in mobile data services (T-Mobile and Vodafone) intercept the response from the Web server and secretly infiltrate into Web page's JavaScript code and forward responses to the corresponding client. However, it is a merit of our approach to pinpoint those differences and attribute them to some specific components like the servers here.

7.4 Conclusion

In this chapter, we tackled the issue of comparing networking applications over different access technologies : FTTH, ADSL and Cellular. We focused on the specific case of search services. First, we showed that packet loss, latency, and the way clients interact with their mobile phones all have an impact on the performance metrics of the three technologies.

Second, we applied our technique of data time break-down that presented in Chapter 5. It automatically extracts the impact of each of these factors from passively observed TCP transfers and then group together, with an appropriate clustering algorithm, the transfers that have experienced similar performance over the three access technologies.

Application of this technique to the Google Web search service demonstrated that it provides easily interpretable results. It enables us for instance to pinpoint the impact of usage or of raw characteristics of the access technology. We demonstrate that user behavior dominates clusters with large volume of data packets and connections. This explains the similar behavior of FTTH and ADSL as response time is dominated by Warm-up A. Clustering results without taking into account Warm-up A suggest that clusters depend mainly on connection size and access impact. We observe for identified clusters different data preparation times at the server side, depending on the terminal used by the end user. Especially, Cellular connections from mobile devices like iPhone and Windows CE have larger data preparation time at Google's servers than on Windows devices.

To provide evidence for these observations, we further compared Yahoo and Google Web search traffic and provided evidences that they are likely to adapt content to the terminal capability for Cellular clients which impacts the performance observed. Indeed, Cellular clients featuring a laptop/desktop Windows operating system (Vista/XP/2000) experience similar warm-up B as ADSL clients while clients using iPhones or a Windows-CE operating system experience way higher warm-up B.

In the next chapter, we characterize a number of the most salient aspects of enterprise traffic. The idea is to present an overview of some problems faced when performing measurements such as basic RTT estimation, and then investigate performance of main used application in the considered enterprise network using our break-down and clustering approaches.

Chapter 8

A First Characterisation of an Enterprise Traffic

8.1 Introduction

Enterprise networks have a complexity that sometimes rival the one of the larger Internet. The characteristics of traffic inside enterprises remain almost wholly unexplored. Nearly all of the studies of enterprise traffic available in the literature are well over a decade old and dedicated to individual Local Area Networks (LANs) rather than whole sites.

In this chapter we present a broad overview of traffic traces collected from a medium-sized site with heterogeneous characteristics in terms of client access link (wired, wireless and VPN accesses) and client usage (student, staff and nomad). The packet traces span one day of capture, over which we observed a total of 345 clients and 56 internal servers .

The main idea here is to characterize a number of the most salient aspects of enterprise traffic. Our goal is to provide an overview of some problems faced when performing measurements such as basic RTT estimation, and then present a fine-grained profiling of popular applications.

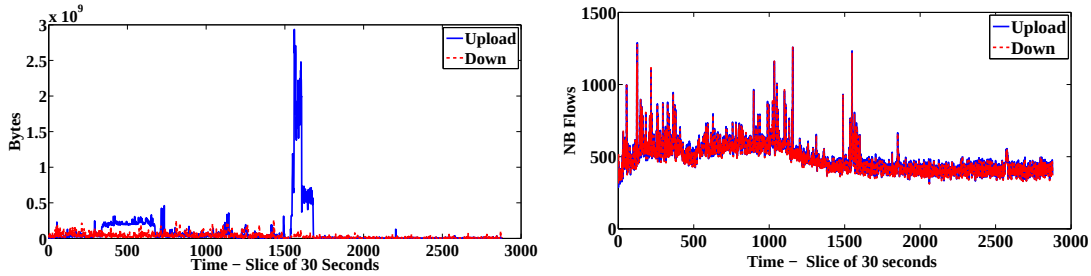
8.2 Overall characteristics

In this section we first examine some basic characteristics of TCP connections in our dataset. Secondly, we describe problems faced when we inferred RTT measurement.

For our measurements, we observe different classes of traffic inside Eurecom traffic. It involves DMZ, sever to server and client to server.

8.2.1 Backup Traffic Impact

We start with the study of the time series of traffic volume, to check if several regimes exist in our data. To distinguish upload and download flows, we consider the client as the initiator of TCP connection and the server as the remote part. The initiator corresponds mostly to a regular end user machine, but it can also be a server requesting service from another server as we will see soon. Based on this assumption, we display in Figure 8.1 the evolution of traffic volume and number of active flows for upload and download directions. From Figure 8.1(a) we observe that upload traffic volume is characterized by two high peaks. After investigation we concluded that (i)the first peak corresponds to the usage by the client of their homes directory to store data (ii)the second one which happens at night, precisely at around 10 pm corresponds to backup traffic.

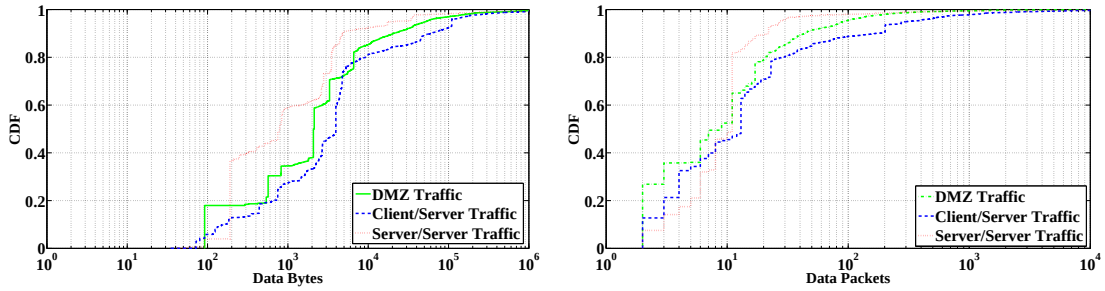


(a) Data Volume (b) NB Active Flows

FIGURE 8.1 – Data Volume and Nb Flows Stability

On the other hand, Figure 8.1(b) shows that the number of active flows vary during the capture time for two periods of times. The first period is between 7 :35am and 6 :45am : which is the time where Eurecom employees are in their office. The second period, as we have identified in Figure 8.1(b) on data volume, corresponds to flows generated by backup traffic.

8.2.2 Connection Characteristics



(a) Bytes (b) Data Packets

FIGURE 8.2 – Connection Size

Figure 8.2 shows the distribution of the total number of bytes, in both directions, transferred across each connection in our dataset, for each class of traffic and well-behaved TCP connections.

It appears that client/server traffic is larger than other transfers. In other hand, we shows that server to server traffic is characterized by shorter TCP connections.

A comparison between median transfers size, for connections between clients and servers, and median transfers size for internet traffic presented in Figure 4.4, shows that client/server traffic inside enterprise is larger than Internet ones. A first explanation of this observation is the difference of used applications and services. In these environements as one finds in enterprise networks some applications like network file systems applications (NFS, SMB) which involve massive data transfers.

8.2.3 Throughput for Enterprise Traffic

The estimation of throughput of enterprise traffic is important for a broad class of applications. We focus in this paragraph on results obtained with AL throughput estimation method, introduced previously in Section 2.4.1, in order to avoid TCP tear-down impact. We first classify TCP throughput estimation into different categories, depending on the class of traffic.

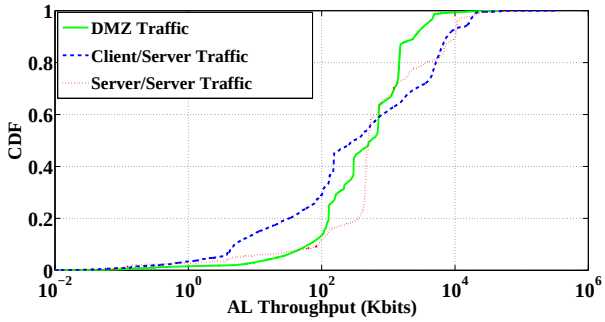
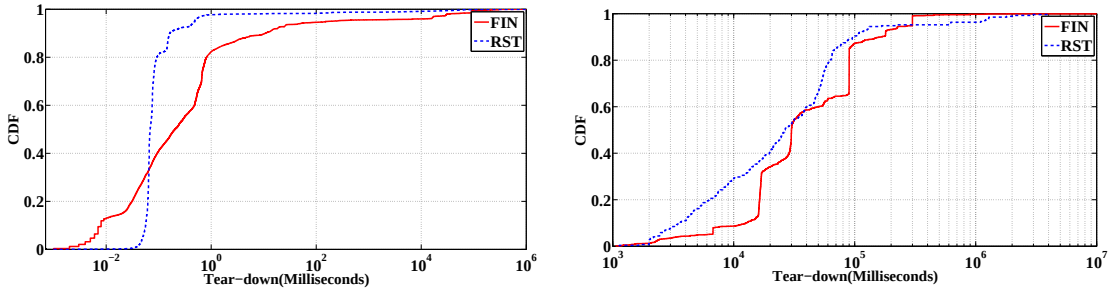


FIGURE 8.3 – Application Level Throughput

Figure 8.3 shows cumulative distributions of Application Level throughput for Eurecom Internal traffics. We observe that identified classes of traffic present approximatively similar AL throughput with a little advantage for server to server traffic. This is in line with the usage inside Eurecom and also in enterprise traffic in general where one generally caps end hosts capacity to 100 Mb/s (even though they might have 1 Gb/s) while servers are set up with 1 Gb/s access. Still, the rates observed are modest as compared to those capacities, highlighting again the impact of the application in actual data exchanges.

8.2.4 Tear-down Analysis



(a) All Tear-Down

(b) Tear-down More than 1 second

FIGURE 8.4 – Tear-Down Times

Before examining different facets of TCP connection closing methods within an enterprise, we recall that we define the tear-down as the time between the last sent data packet and last control packet. The last control packet corresponds to a packet with FIN or RST flag.

Flag	Tear-Down	Tear-Down >1 seconds
FIN (%)	62.36	85.71
RST (%)	37.63	14.28

TABLE 8.1 – Tear-Down Flags

We examine in Table 8.1 the percentage of TCP connections finished with FIN or RST flags. In our dataset, we observe that 62% of all TCP connections are finished with FIN flag. Then it is important to note that this percentage increases to reach 85% for connections with tear-down values more than one second.

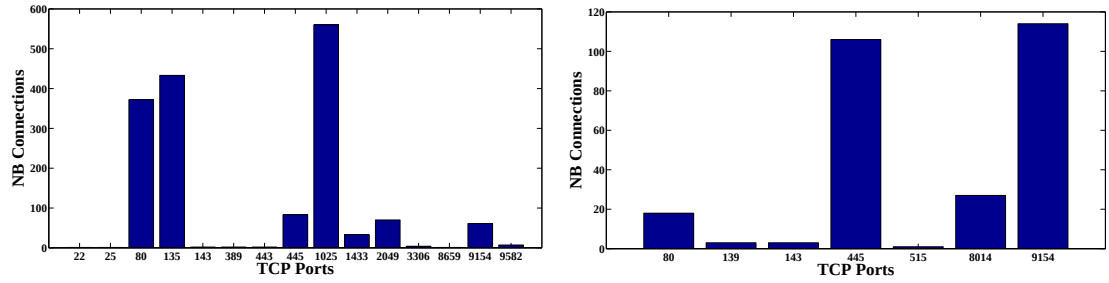
Figure 8.4(a) shows cumulative distributions of time needed to perform tear-down. We distinguish between two exiting TCP connection tear-down methods : with FIN or RST flags. From this figure we can notice that more than 33% of tear-down times using FIN flags are larger than using RST ones. At this stage it is a bit early to present a final conclusion about the comparison of tear-down methods.

To go farther in our analysis of TCP tear-down methods, we focus in Figure 8.4(b) only on connections with large tear-down : more than 1 second. Figure 8.4(b) shows that a large fraction of connections finished with FIN flag tend to have large tear-down than ones finished with RST flag.

Additionaly, Figure 8.4(b) shows for connections finished with FIN flag several peaks, e.g, at 10 or 100 seconds. We conjecture that they could be due to internal timers in servers/services set-ups.

The next step was to identify prevalent applications and services with large tear-down values. To do that, we report in Figure 8.5 statistics about targeted ports and the corresponding number of TCP connections. Figure 8.5 shows for connections finished with FIN flags, that the most targeted ports are ports HTTP, End-Point Mapper (EPMAP), SMB and 1035 (used for windows Remote Procedure Call (RPC)). In contrast, for connections with RST flags we find as target ports : HTTP, SMB (445), 8014 (used by SYMANTEC) and 9154.

It thus appears that different liberation methods correspond to different services or different regimes (normal, abnormal) within an applications.



(a) FIN Flag (b) RST Flag

FIGURE 8.5 – Connection With Large Tear-Down : Destination Ports

We assessed in this paragraph characteristics of connection liberation methods with FIN or RST flags. We observed that most connections are closed with FIN flags, the 'usual way' to close correctly a current TCP connection. While we are focusing on an environment with short latency we observed high values of tear-down times, especially for FIN flag. The study of high FIN tear-down times reveals several peaks that presumably correspond to time-outs at the application layer.

We did not continue this study further as we expect that in an enterprise network, actual tear-down times are loosely if at all related with clients performance.

8.3 RTT Estimation in Enterprise Network

One likely reason why enterprise traffic has gone unstudied for so long is that it is technically difficult to measure. Enterprise networks, unlike a site's Internet traffic, which we can generally record by monitoring a single access link, the capture process is more difficult within an enterprise with different sub-networks.

In the case of Eurecom network, the small size of the network, presented in Chapter 1 as considerably simplified the capture process for us. The small time scales values appears to be challenging in a typical enterprise networks. However the problem of RTT estimation is raised.

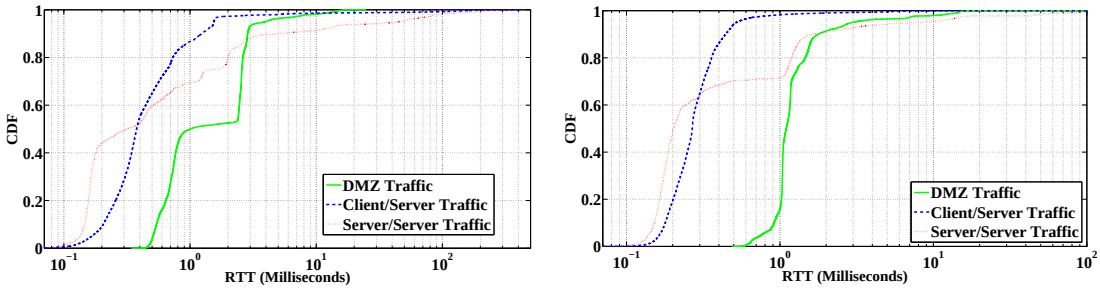
The RTT measurements are composed of several delays, e. g., transmission delay or propagation delay, but also queuing delay at various network elements and end hosts. Network measurements usually include all of these delays. However, for end-to-end measurements, most of tools and researchers assume that local processing and queuing delays are negligible and interpret their results without considering local delays.

In this paragraph, we compare different RTTs estimations method in order to identify a method that allows to minimise biases introduced within measurement process. On the other hand it is important to note that our setting – a mirror port connected to a collection machine that uses tcpdump – is de facto weak as compared to hardware methods that rely, e.g., on Data Acquisition and Generation (DAG) cards which are Network Interface Card (NIC) dedicated to capturing traffic, where time stamping is handled directly at the NIC level with a precision higher than any software (tcpdump, windump) method.

To estimate RTT, we adopted two techniques already presented for Internet traffic in Section 3.2.1. The first method is based on the observation of the TCP 3-way handshake. The second method is similar but applied to TCP data and acknowledgement segments transferred in each direction [1].

Figure 8.6 shows RTT estimations for each class of traffic. From theses figures we can observe :

– RTT using DATA-ACK estimation method are larger than ones with three way handshake,
– Majority of RTT are very short,
– RTT values inside Eurecom network can reach 100 ms. In fact, we have noticed that these cases correspond to several clients connected to Eurecom network using Virtual Private Network (VPN) connections. Clients connected via VPN access were characterized by large latency due to their localisation outside the enterprise building,
– Large RTT were computed for traffic between DMZ zone and servers.



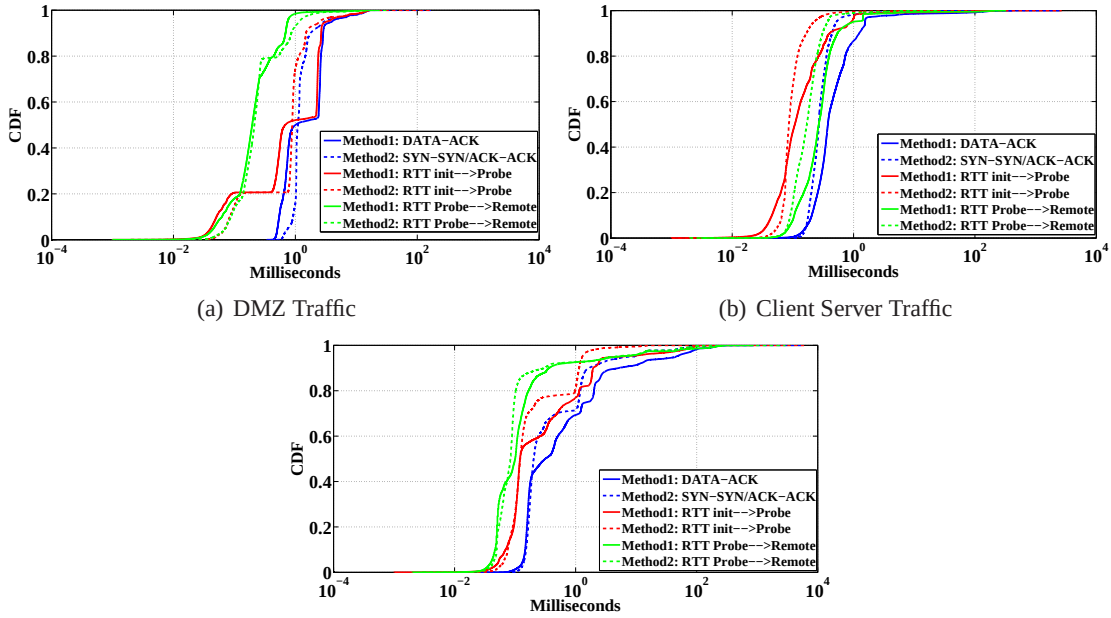
(a) Data-ACK (b) SYN-SYN/ACK-ACK

FIGURE 8.6 – RTT Estimations

In Figure 8.7 we compare RTT estimation methods, for each RTT side. We divide RTT estimation in two values. For each RTT estimation approaches, we compute : first, the time elapsed between connection initiator and the probe, second, the time elapsed between the probe and the distant side.

A first observation is that RTT estimation using DATA-ACK method provide larger RTT estimates for all observed traffics. A possible explanation for the introduced delay by DATA-ACK

1. For our case we focus only on well-behaved transfers with a minimum of one data packet exchanged in each direction.



(a) DMZ Traffic

(b) Client Server Traffic

(c) Server - Server Traffic

FIGURE 8.7 – RTT : Detailed Comparison

method, was the small number of data packets of these connections. In fact, delays added by delayed ACKs timer (maximum 500 ms [73]) can increase the time to sent data packet and to receive the corresponding ACK. Indeed, we have observed that difference between RTT estimation methods :$RTT_{\text{DATA-ACK}} - RTT_{\text{SYN-SYN/ACK-ACK}}$ was less than 400 ms that could confirm the hypothesis of delayed ACKs.

Further, to investigate differences between RTT estimation methods, next in our experiments we restrict our analysis on client server traffic.

8.3.1 Short Connection Impact

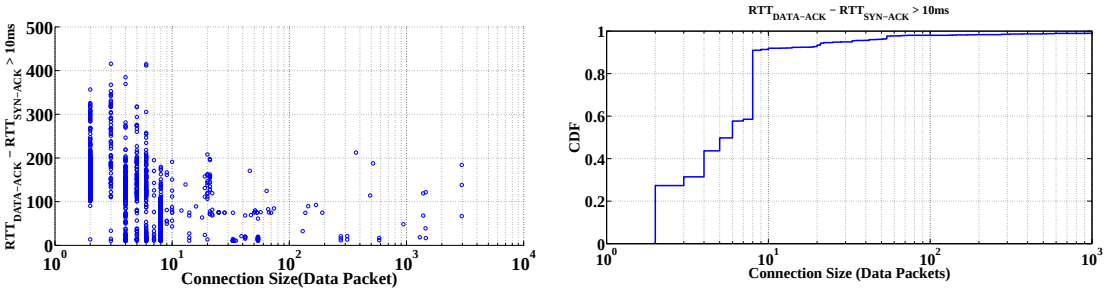
We focus in this paragraph on connections where the $RTT_{\text{DATA-ACK}} - RTT_{\text{SYN-SYN/ACK-ACK}}$ is larger than 10 ms. Our purpose is to validate the hypothesis presented in the previous paragraph : observed differences between RTT estimations is especially present for short TCP connections.

We report in Figure 8.8(a) the scatter plot of $RTT_{\text{DATA-ACK}} - RTT_{\text{SYN-SYN/ACK-ACK}}$ more than 10 ms on the x-axis and corresponding connection size in terms of data packets.

Figure 8.8(b) shows the CDF of connections size in terms of data packets, with $RTT_{\text{DATA-ACK}} - RTT_{\text{SYN-SYN/ACK-ACK}}$ more than 10 ms.

A first observation from Figure 8.8(b) was that 92 % of connections with $RTT_{\text{DATA-ACK}} - RTT_{\text{SYN-SYN/ACK-ACK}}$ more than 10 ms are less than 10 data packets. It confirms the assumption for RTT over estimation using DATA-ACK method for short transfers. Figure 8.8(a) shows two high values corresponding to the difference $RTT_{\text{DATA-ACK}} - RTT_{\text{SYN-SYN/ACK-ACK}}$ with 425 ms. These values are obtained with connections of 3 and 6 data packets.

At this stage of analysis we can conclude that RTT estimation is more accurate with three way hand shake method. To fully validate that RTT was better estimated using the 3-way handshake method, we compared values obtained with this method to actively inferred latency measurements with ping.



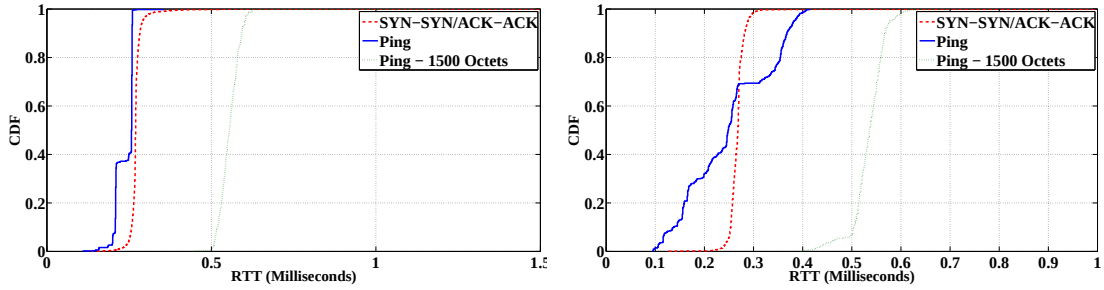
(a) Scatter Plot (b) CDF

FIGURE 8.8 – RTT Estimation Methods and Connections Size

8.3.2 A comparison with Active Measurements

In this paragraph we compare controlled experiments of RTT estimation with Internet Control Message Protocol (ICMP) messages and three way handshake method, to examine the accuracy of our selected RTT estimation method.

We base our experiments on the estimation of RTT using ping message sent from client to server. The purpose here is to estimate RTTs from client to a remote host using ICMP packets. This measurement technique involves sending an ICMP Echo Request packet, receiving an ICMP Echo Reply packet, and recording the elapsed time between the two events. We applied this technique, for the case of two different pairs of client and server, where we had, for each case, enough samples obtained with the SYN/SYN-ACK-ACK method (i.e., cases for which we observed a large number of connections between the client and one server).



(a) Pair 1 (b) Pair 2

FIGURE 8.9 – A comparison with Active Measurement

We plot in Figure 8.9 RTT estimations with tree way handshake, a classical ping and a ping message with 1500 octets of payload. The first observation here is that RTTs are very low and close to the precision of tcpdump (10 microseconds). RTT estimation with three way handshake and classical ping are similar for the two observed couples of clients and servers. Experiments show different results for full sized ICMP message with largest RTT values. We believe that the observed difference is due to data sending and receiving process.

In summary, RTT estimation is challenging in an enterprise environment when the estimation is done using legacy NIC and tcpdump like methods as the values are close to the precision of the timestamping achieved with this technique. The latency is also such that the packet processing time plays a role as well as the delay-ack mechanism whose timer value is very large as compared

to the observed RTT. At the end of the day, we obtained (see Figure 8.1(b) for instance) that the RTT estimation can lead to errors up to 100%. Still, the order of magnitude (a fraction of a ms) is correct. When applying our profiling technique based on a data time breakdown, the precision will turn out to be large enough as the other phenomena that we want to measure, esp. the application and users delays are working at a time scale of a few ms or even 10s of milliseconds, ie. one order of magnitude larger than the RTT. Clearly more accurate method of RTT estimation should be developped for RTT estimation in enterprise networks, that we leave for future work.

8.4 Service Profiling

We first take in Section 8.2 a broad look at the protocols present in our trace. We examined in previous sections of this Chapter several parameters of enterprise traffic performance.

We noticed from Table 1.6 that SYMANTEC and LDAP(S) generate the largest number of TCP connections, but not the largest volume of data. On the other hand largest volumes of data were obtained respectively with NFS and SMB, used by client to access files over a network in a manner similar to the way local storage is accessed.

Before proceeding further, we need to present statistics about machines and classes of clients for the present trace. Next in our analysis, we focus only on client/server traffic. Table 8.2 shows the percentage of generated connections per class of users. It depicts that the enterprise trace consists of 3 categories of users (based on IP address identification) : staff with 73.9% , student with 21.6% and finally nomad users and external users with the lowest percentage - we group the last classes into the Others class.

Client	Connection (%)
Student Eurecom	21.6
Staff Eurecom	73.96
Others	4.42

TABLE 8.2 – Eurecom Clients

In addition to the knowledge of clients classes, we succeeded to identify clients operating system and the nature of his machine : personal computer or laptop. We figured out that the majority of connections were established from Windows machine with 66%, Linux with 16%, laptop with 4.6% and the remaining is a mix of non classified connections.

We report in Figure 8.10 data time break down for traffic between eurecom's client and servers. We plot results for all transfers. Figure 8.10 shows the breakdown per direction and per access technology with, for each case, the median of each component in relative (left y axis - relative to total data time) and absolute values (right y axis - in seconds). The first observation is that data time transfer is dominated by data preparation time at the server side. Median data transfers time are very short, due to access characteristics.

A comparison of data time beak down results in Figure 8.10 and results with Internet traffic in Figure 5.2 shows different transfer profiles. First, data transfers for enterprise traffic are shorter than Internet ones, due to each access characteristic.

While for enterprise and Internet traffics, we observe that Theoretical times represent approximatively between 35% and 50% of data transfers time, we notice different impact for server side. In fact, enterprise traffic is dominated by data preparation time at the server side and without Pacing A and B, while for Internet Traffic Warm-up A is larger than Warm-p B. An explanation for this observation is that Web traffic, which implies interaction between client and servers (some

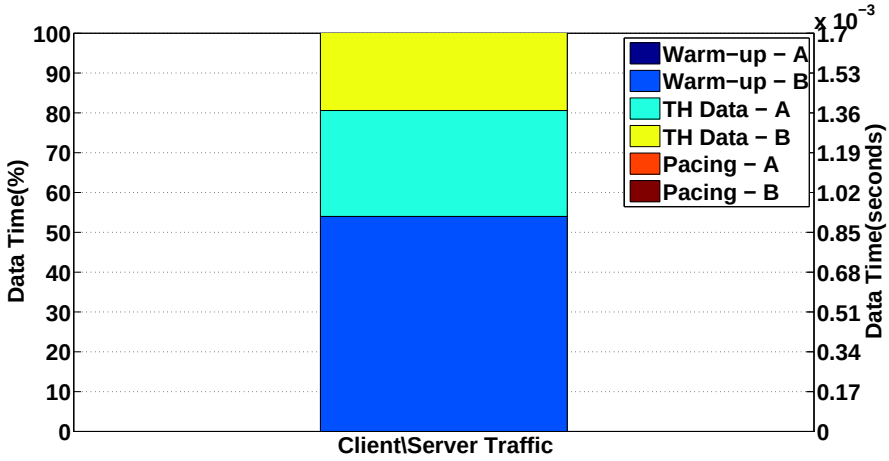


FIGURE 8.10 – Data Time Break Down for Client/Server Traffic

time clients take large time to think and perform a request), dominates Internet traffic. Also, for Internet traffic we showed that Pacing represents between 20% and 42% of data transfer time

To go deeper in enterprise characteristics and to highlight server, application and usage impacts, we focus in the next paragraphs on LDAP and SMB traffic.

8.4.1 LDAP

In our trace, LDAP(S) is a key protocol that email and other programs use to look up information from a server. LDAP is not limited to contact information, or even information about people. It is used to look up encryption certificates, pointers to printers and other services on a network, and provide "single sign-on" where one password for a user is shared between many services. We shows in Section 8.2 that it represent a large amount of TCP connections due to its importance and its usage by key enterprise applications.

To investigate the performance of LDAP protocol our strategy was to apply our data time break-down and clustering approaches, in order to propose a fine grained study an to shed light on the interplay between service, access and usage, for the client and server side.

Figure 8.11 depicts the 4 clusters obtained by application of Kmeans. The value of 4 clusters was obtained by inspection of the projection obtained with t-SNE. We indicate, on top of each cluster, the median connection size, the percentage of involved connections and clients.

In order to have an idea about how LDAP exchanged data is allocated over identified clusters, we present in Figure 8.12 data volume distribution per cluster. We observed, in the one hand, two dominant clusters : 1 and 2 with 99% of data and in other hand clusters 3 and 4 with less than 1% of data.

A first observation from Figure 8.11 is that three of the identified clusters (Cluster 1, 2 and 3) are characterized by a dominance of data preparation time at the server side, what we defined as Warm-up B. Inside these clusters we identified 2 categories of servers. In fact in clusters 1 and 2, with majority of clients, we observed that clients establish connections to Active Directory Domain Controller, while in cluster 3 we identified only LDAP servers for Linux machines.

Cluster 4 contains only 1% of LDAP connections and 8% of clients and it was characterized by large Theoretical times A and B. Clients in this cluster corresponds to ones with Wifi and VPN access, which explains high theoretical times.

A characterisation of LDAP traffic reveals a strong correlation with the target servers. Data

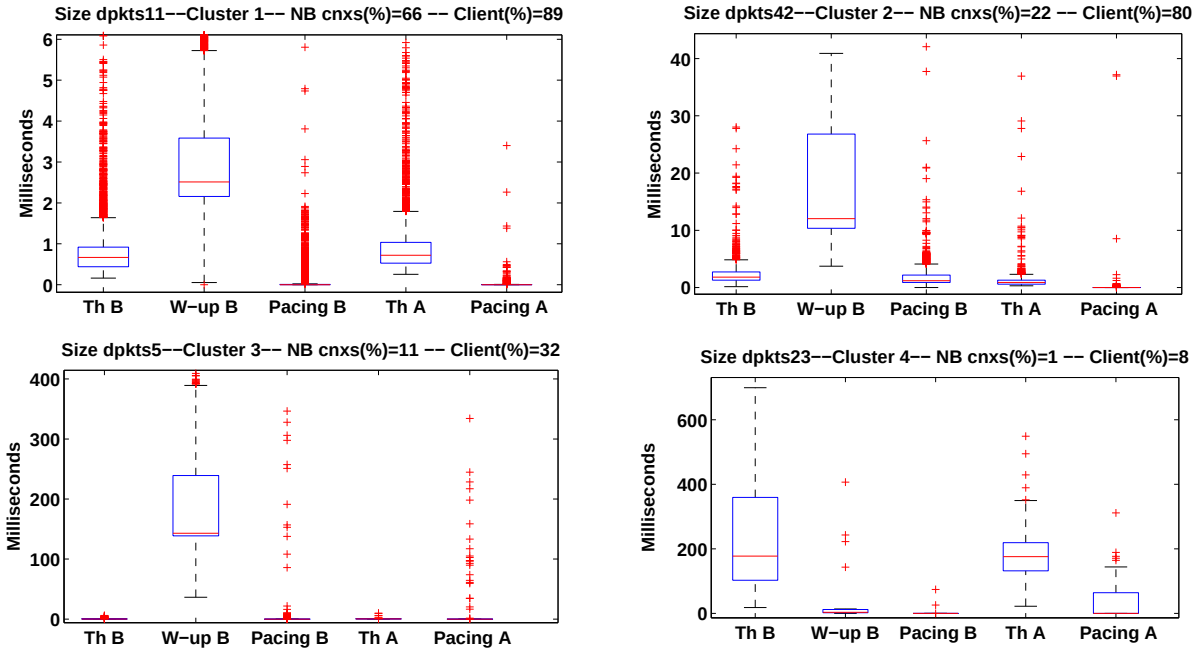


FIGURE 8.11 – K-means Clusters : LDAP

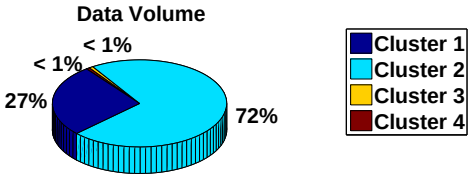


FIGURE 8.12 – Data Distribution per Cluster : LDAP

Warm-up times on the server side dominate data transfers times for the majority of transfers in clusters 1, 2 and 3. Connections to LDAP servers from Linux machines in clusters 3 are short compared to ones in the remaining clusters, which can highlight different LDAP policies between Linux and Windows machines. We summarize in Table 8.3 the characteristics of each identified clusters.

Cluster 3	Cluster 1	Cluster 2	Cluster 4
LDAP Server for Linux	Domain Controller - Active Directory		
	Majority of Connections	Large Transfers	Large RTT

TABLE 8.3 – Clusters Characteristics : LDAP

In summary, our data time breakdown and clustering technique reveals that for internal clients, the major source of delay was the server data preparation time. Still, the values of the warm-up are small in most cases, and we do not see any highly visible performance anomaly in LDAP for the Eurecom network.

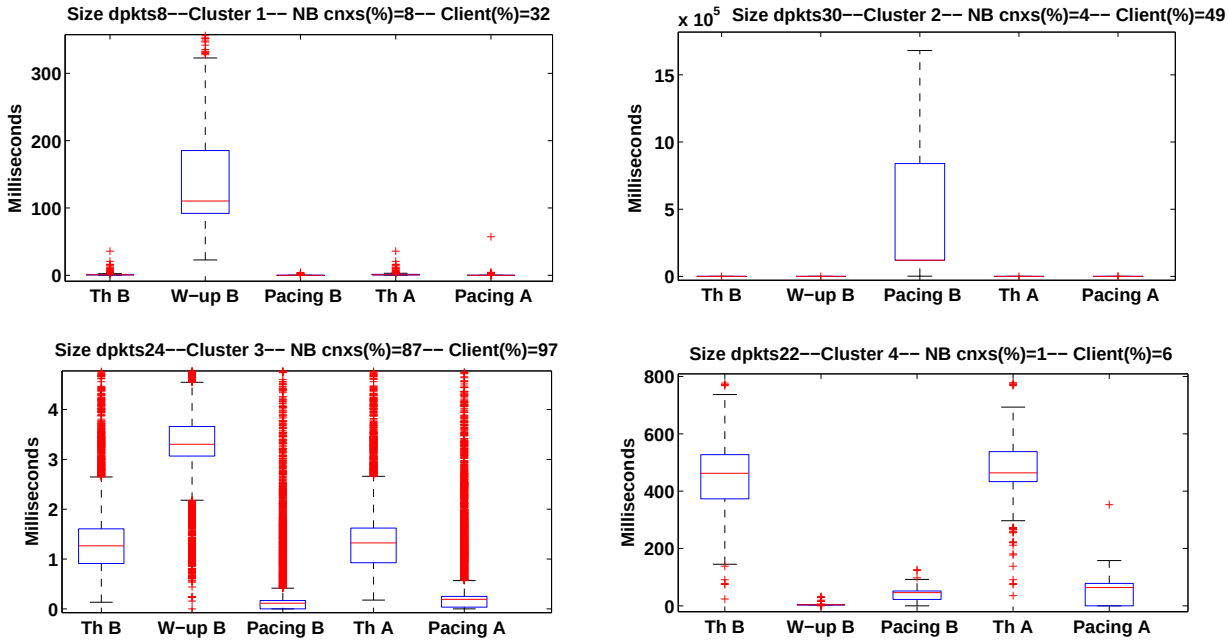


FIGURE 8.13 – K-means Clusters : SMB

8.4.2 SMB

Server Message Block (SMB) traffic is an application-level network protocol typically used for file and printer sharing. It represents the largest volume of data in our trace. It also provides an authenticated inter-process communication mechanism. Dominant usage of SMB involves computers running Microsoft Windows. We present in the following paragraph an analysis of regimes that we observe in SMB traffic. Figure 8.13 shows clustering results for SMB connections. As for LDAP clustering cluster 4 corresponds to connection with large RTT, and groups SMB users connected via Wifi and VPN accesses.

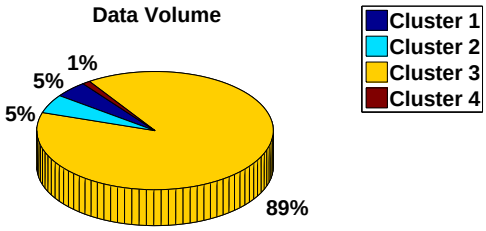


FIGURE 8.14 – Data Distribution per Cluster : SMB

The study of targeted servers shows two categories of clusters. Cluster 1 corresponds to servers that contain client data such as homes folders and data. On the other hand, clusters 2, 3 and 4 correspond to connections towards Active Directory domain controllers. Figure 8.13 depicts large Warm-up B for cluster 1. We noticed large Pacing B for cluster 2 - a median over 10 s, probably due to the chatty nature of SMB protocol [100]. Cluster 3, that contains the majority of exchanged bytes, as plotted in Figure 8.14, involves most of SMB clients. It presents reasonable values of data time break down with a dominance of Warm-up B.

The study of SMB application shows 4 categories of clusters. We observed that cluster 4 corresponds to clients with large RTT via VPN and Wifi accesses. On the other hand server characteristic plays an important role for the remaining clusters. We summarize in Table 8.4 the main characteristics of each SMB clusters.

Cluster 1	Cluster 2	Cluster 3	Cluster 4
Server for Homes Data	Domain Controller - Active Directory		
	High Number of Trains	Moderate values	Large RTT

TABLE 8.4 – Clusters Characteristics : SMB

8.4.3 Discussion

To study the performance of enterprise traffic we selected two interesting protocols in terms of service presented to client and their usage in enterprise environment. While SMB and LDAP applications have different strategies we noticed similarities in terms of behaviours. In particular we identified between 6 and 8% of clients with VPN and Wifi access localized in cluster 4 for LDAP and SMB. It does not reveal an RTT anomaly but it highlights the impact of low access bandwidth. For the case of LDAP, we shows that the Domain Active Directory Controller plays an important role for these protocols in the way that it was characterized by short think times compared to Linux ones. Finally, we noticed large Pacing B values for SMB traffic in cluster 2 ; It could be classified as an application anomaly ; SMB is a very chatty protocol and performs a large number of data exchanges. In summary, our methods help to identify key regimes due to the characteristics of the access technology used by the end user or the server type/provisioning.

8.5 Conclusion

The study of Enterprise network performance has been neglected in the modern literature compared to Internet accesses measurements. Our major contribution in this Chapter is to provide a first characterization for several aspects of enterprise network traffic.

Our investigation covers topics previously studied for wide-area traffic. Through the study of traffic stability we pinpointed the impact of backup process with an increase of exchanged data volume and RTT. At the connection level, we concentrated on several key indicators such as data transfers, throughput and RTT. Through the study of RTT we highlight the difficulty observed in order to accurately estimate this metric, first due to short absolute RTTs (close de the timestamping accuracy) and second the impact of TCP mechanism like duplicates ACK, which can introduce biases especially for short transfers. For RTT estimation we selected the most accurate method (three way hand shake) that we can use for our case, but for future measurements we recommend to perform measurements with DAG cards.

Then we investigated performance of main used application in our enterprise network using our break-down and clustering approaches.

Our investigation is only an initial step in enterprise traffic analysis. Our in depth profiling of two key services has underscored the ability of our breakdown/clustering approach to pinpoint the different types of usage of these applications. It enables us to obtain clusters that correspond to the baseline regime of the application, e.g. clusters 1 and 2 for LDAP and cluster 3 for SMB. Also, our technique has enabled to identify a potential anomaly - cluster 2 in SMB. There exist two natural extensions to this work.

First, one could collect several days of data, cluster each active period (day time) separately and seek if the baseline regimes persist and identify anomalies as minor clusters, which feature high values on some our metrics. A second extension would be to focus specifically on anomalies by filtering out potential candidate connections, e.g. connections that feature high values on some or all of the metrics obtained during the data time breakdown. Due to the lack of time the first extension, will be presented as a future work for this thesis. On the other hand, we investigated further the case of anomalies for Internet, but also enterprise traffic in the next chapters.

Conclusion of Part II

In this part, we compared the performance from different access technologies such as Cellular, FTTH and ADSL. We showed that this task becomes difficult as the transport layer is interacting with the application layer above and network layer below. We explored several factors that are classically used to assess the performance of TCP connection, namely RTT and losses. The crucial impact of those parameters is formally known since the derivation of the well-known TCP throughput formula. We discussed the derivation of those parameters for the case of our traces. We illustrated shortly the fact that RTT and losses are not enough to characterize TCP connection in the wild.

To overcome to this problematic, we propose a new analysis method that uncovers the impact of specific factors to inform the comparison of different access technologies. The analysis method that we used consists of two steps. In the first step, the transfer time of each TCP connection is broken down into several factors that we can attribute to different causes, e.g., the application or the end-to-end path. In a second step, we used a clustering approach to uncover the major trends within the different data sets under study.

Application of this technique to the Google Web search service demonstrated that it provides easily interpretable results. It enables for instance to pinpoint the impact of usage or of raw characteristics of the access technology. We demonstrated that user behavior dominates clusters with large volume of data packets and connections. This explains the similar behavior of FTTH and ADSL as response time is dominated by Warm-up A.

Using the example of the Eurecom network, we also characterized a number of the most salient aspects of enterprise traffic, and we presented a fine-grained profiling of two popular applications. We obtained clusters that correspond to the baseline regime of the application, e.g. clusters 1 and 2 for LDAP and cluster 3 for SMB. Our technique has allowed to identify a potential anomaly, cluster 2 in SMB.

The next part of this thesis presents how our fine grained approach of performance analysis can be used to detect anomalous TCP connections. We aim at detecting and uncovering the reason behind ill-behaved TCP transfers, where a ill-behaved connection here is a functionally correct TCP connection – normal set-up/tear-down and actual data transfer – that experienced performance issues, e.g. losses or abnormally long waiting times at the server side.

Part III

Profiling Anomalous TCP Connections

Overview of Part III

Traffic anomaly detection has received a lot of attention during last years, but understanding the nature of these anomalies and identifying the flows involved is still a manual task, in most cases. Several traffic anomaly detection methods have been proposed, (i) e.g. DDoS [48, 49, 50, 51, 52], or (ii) traffic feature distributions [53], or (iii) segments that have a sequence number different from the expected one [55], etc.

On the other hand only few works [3, 57] have tried to address the problem of detecting traffic anomalies introduced by performance problems of distant server, upper layer application or service usage.

In this part we focus on the issue of profiling anomalous TCP connections that are defined as functionally correct TCP connections but with abnormal performance. Our method enables to pinpoint the root cause of the performance problem, which can be either losses or some idle times during data preparation or transfer. To study the latter type of anomaly, we use a variant of the method developped in 5 to profile all connections of some traces, irrespectively of their anomalous nature.

In Chapter 9 we apply our method to the case of residential traffic, using the same set of traces (FTTH, ADSL,Cellular) as before.

In Chapter 10 we apply a methodology similar to the one proposed in Chapter 9 to the case of TCP traffic anomalies for enterprise traffic.

Chapter 9

Pinpointing and Understanding Anomalous TCP Connections in Residential Traffic

9.1 Introduction

Several access technologies are now available to the end user for accessing the Internet, e.g., ADSL, FTTH and Cellular. Those different access technologies entail different devices, e.g., smartphones equipped with dedicated OS like android. In addition, a different access technologies also imply a different usage, e.g., it is unlikely that p2p applications are used as heavily on Cellular than on wired access. Even if we consider ADSL and FTTH, which are two wired technologies, some differences have been observed in terms of traffic profile [27].

Despite this variety of combinations of usage and technology, some constant factors remain in all scenarios like the continuous usage of email or the use of TCP to carry out the majority of user traffic. This predominance of TCP constitutes the starting point of our study and our focus in the present work is on the performance of TCP transfers.

In this chapter, we aim at detecting and uncovering the reason behind ill-behaved TCP transfers, where a ill-behave connection here is a functionally correct TCP connection – normal set-up/tear-down and actual data transfer – that experienced performance issues, e.g. losses or abnormally long waiting times at the server side. Note that this a different objective from the detection of traffic anomalies, where the focus is to detect threats against the network, e.g. DDoS [49, 50, 48, 51, 52].

Our main contributions are as follows (i) we demonstrate that wired (ADSL and FTTH) and wireless (Cellular) technology adopt different strategies to recover from packet losses, especially under time out conditions (time outs being prevalent in all environments over fast retransmits), and that the strategies observed on the Cellular technology seem more efficient than on ADSL and FTTH, (ii) we show that our methodology for profiling the transfers (or parts of transfers) unaffected by losses is able to uncover various types of anomalies, some being related to the configuration of servers and some other being shared by several services.

9.2 On the Impact of Losses

TCP implements reliability by detecting loss of data segments and retransmitting lost segments. Unfortunately, the loss detection/recovery mechanism can be penalising due to high re-

transmission times : it is generally known that segment losses can adversely impact the duration of TCP connections,especially short ones.

In this section, we address this issue by evaluating the impact of TCP loss detection/recovery mechanisms (presented in Section 4.3.3), with the study of loss recovery approaches, on the performance of real-world TCP connections collected from different Internet accesses.

9.2.1 Identifying RTO and FR/R

TCP detects and recovers from losses using two basic types of mechanisms : retransmission-timeouts (RTO) and fast retransmit/recovery (FR/R).

In a nutshell, RTO can be seen as a safety mechanism that is slow but can recover from losses in any scenario, provided that the network offers enough resources to carry the segment. On the other hand, FR/R does not work in any situation but in the majority of them and provides faster recovery time than RTO. FR/R triggers retransmission as soon as 4 ACKs with the same sequence number are observed. This figure of 4 ACKs, represents a trade-off between accuracy and speed of reaction as a lower value would lead to false positives if packet reordering occurred.

Two important parameters guide the design of TCP loss detection/recovery mechanisms. First, TCP should accurately identify segment losses. In particular, if TCP erroneously inferred that a segment was lost, it would unnecessarily invoke loss recovery and increase the connection duration. Second, TCP should quickly identify segment losses. A longer detection period adversely impacts connection duration as well. However, a quick inference of segment loss would also be erroneous when segments (or their ACKs) are not lost but merely delayed or reordered in the network. To achieve high loss-estimation accuracy, therefore, TCP has to wait longer for ACKs that may merely be delayed.

More generally, this fundamental trade-off between accuracy and timeliness is controlled by several design parameters associated with RTO and FR/R based loss detection. These include the duplicate ACK threshold, the minimum RTO, the RTT-smoothing factor, the weight of RTT variability in the RTO-estimator, and the RTO estimator algorithm itself. While the proposed standards for TCP recommend values for each of these design parameters, TCP implementations in prominent operating systems differ, sometimes significantly, in the values used.

The invoking of loss detection/recovery can thus be quite costly in terms of connection duration. The exact cost depends on the choice of values for each of the parameters associated with loss detection such number of duplicate ACK, minimum RTO and TCP stack parameters on used client and servers.

9.2.2 Retransmissions in the Wild

We first report, in Table 9.1, on two metrics : the average loss rate and the average fraction of connections affected by loss events for the three traces (results are based on loss detection/recovery algorithm presented in Section 3.4.3).

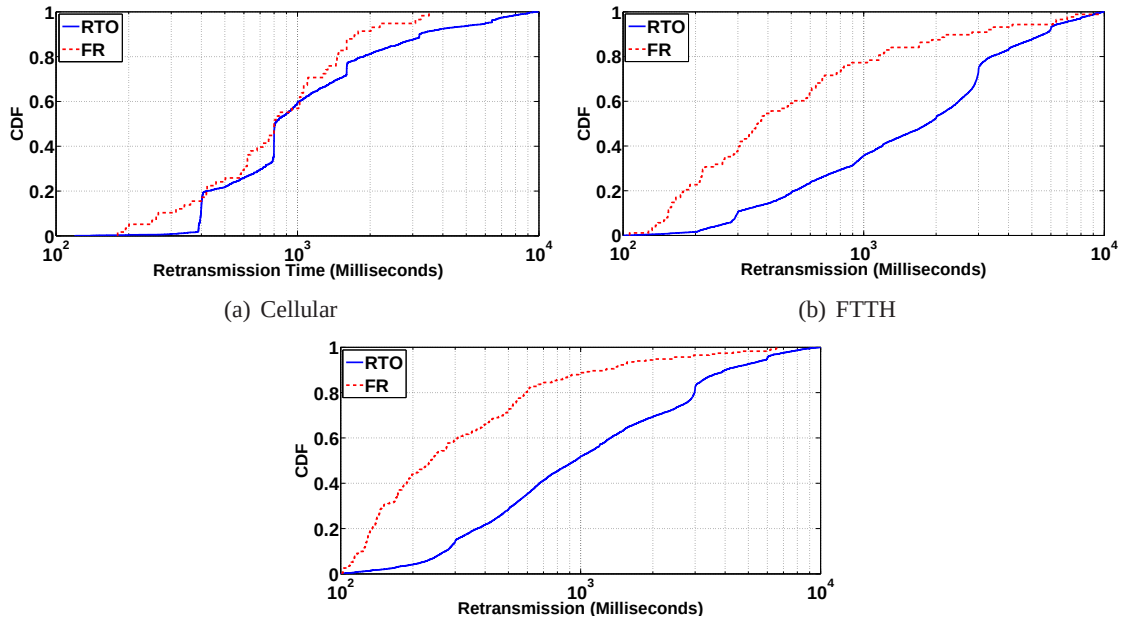
	Cellular	FTTH	ADSL
Loss rate	4%	2%	1.2%
% of connections	29%	9%	5%

TABLE 9.1 – Overall Loss Rates

We observe from Table 9.1 that while loss rates are quite low (our traces are too short to draw general conclusions on the loss rates in each environment), the fraction of connections affected

by losses are quite high, esp. for the Cellular technology. A possible reason is that losses are due to actual wireless channel conditions with the Cellular technology, which may result in small loss episodes that affect connections irrespectively of their status (duration, rate).

We next turn our attention to the way losses are recovered by TCP. We suppose that RTO (resp. FR/R) correspond to recovery periods with strictly less than (resp. greater or equal to) 3 duplicate acknowledgments. This definition leads to a striking result : for our traces, more than 96% of loss events are detected using RTO. Two factors contribute to this result. First, most transfers are short and it is well-known that short transfers, which do not have enough in flight packets to trigger a FR/R revert to the legacy RTO mechanism. Second, long connections must often rely on RTO as the transfer, while large, consists of a series of trains (questions and answers of the application layer protocol) whose size is not large enough, in almost 50% of the cases in our traces, to trigger a FR/R (see next section for more details).



(a) Cellular

(b) FTTH

(c) ADSL

FIGURE 9.1 – Retransmission Time

Figure 9.1 plots the distribution of data retransmission time for FR/R and RTO based retransmissions (only for connections that experience losses). A first observation is that ADSL and FTTH show similar behavior to recovery from losses. We observe that :

– FR/R retransmission times are shorter than RTO, as expected, for all access technologies. Still, the difference is larger for the FTTH trace than for the Cellular trace. This apparently reveals different implementation strategies in a cellular environment.
– A significant number of Cellular TCP retransmission result from losses at the beginning of the respective TCP connections, where the RTO is primarily governed by the initial RTO. It results in 5 easily identifiable peaks in the RTO values at 400ms, 800ms, 1600ms, 3200ms and 6400ms.

The research question we target is the identification of anomalous TCP connections in a set of environments that reflect user typical experience nowadays. We observe from the above results that manufacturers apparently take advantage of the leeway in the specification of TCP to try to optimize performance in some environments, esp. the Cellular environment. For the latter case, it

results in RTO performance close to the FR/R performance in cellular environment. As cellular environment features higher RTTs, in the order of 200 ms as compared to less than 70 for ADSL and FTTH cases, the optimization process has possibly reached its limit in this environment with the current technology constraints. Optimizing the RTO mechanism is a strategy that pays offs as the vast majority of TCP transfers rely on RTO. If we arbitrarily set a threshold in terms of anomaly to 1s of recovery period, we observe that with the current optimization, the fraction of anomalous recovery time is about 20% smaller in Cellular than in ADSL and FTTH scenarios.

9.2.3 Studying Impact on Short and Large Transfers

It is well-known that packet losses can adversely affect the connection duration of TCP connections. However, what is not fully understood is how short and large transfers deal with losses.

Figure 9.2 plots the distribution of retransmission time for ADSL trace and for short and large transfers. We find that : a significant fraction of large transfers retransmission time is similar to short transfers retransmission time. Moreover, the tail of distributions show that large transfers are more penalized by consecutive retransmissions.

This is against intuition as it suggests that large transfers should recover from loss using FR/R. Also, when focusing on short transfers, we notice that more than 50% of RTO are less than 1s which is the recommended minimum RTO threshold [13]. Also, short and large transfer retransmission times apparently show new RTO thresholds. For instance, Figure 9.2 shows that 7% of RTO short transfers are equal to 300 ms which suggests a new RTO threshold implemented in new TCP stack generation. The explanation behind the bad performance of long transfers lies, as already pinpointed in Chapter 3 in the impact of the application on top of TCP.

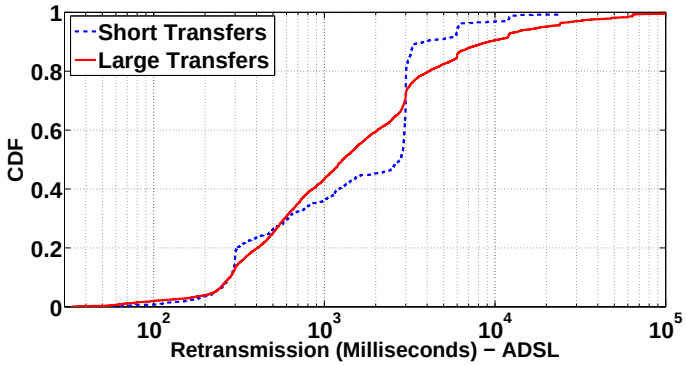


FIGURE 9.2 – Short vs Large Transfers

Indeed, one often observes that even if the server is sending a large amount of bytes/packets, the actual exchange is fragmented : the server sends a few packets that we called a train of packets, then waits for the client to post another request and then sends its next answer. If such a behavior is predominant in TCP transfers, it can have a detrimental impact if ever the train size is too small as it might prevent TCP from performing FR/R in cases of losses.

Table 9.2 summarises the distribution of train sizes for short and large transfers. We use our definition presented in Section 2.3 to classify short and large transfers, . We distinguish between the initiator of the connection, which is generally the client and the remote party which corresponds to the server. We differentiate between trains less or more 3 data packets. In fact trains with more than 3 data packet are able to recover using FR from loss only when the first data packet is lost and the recommended duplicate ACK is equal to 2 duplicate ACK. This depends on the actual implementation of the OS as the default standard is 3 duplicate ACK. Here we observe that :

	Short Transfers				Large Transfers			
	Initiator		Remote		Initiator		Remote	
Trace	<=3	>3	<=3	>3	<=3	>3	<=3	>3
CELL	98%	2%	98%	2%	65%	35%	68%	32%
FTTH	90%	10%	92%	8%	47%	53%	49%	51%
ADSL	92%	8%	92%	8%	64%	36%	76%	24%

TABLE 9.2 – Train Size Distribution

– Trains sent by servers (remote party) are larger than those sent by the initiator (local client), in line with our hypothesis that the remote party is the server ; Remember that in the Cellular network we observe only the mobile hosts can initiate connections to the outside : they can not be reached from the outside,
– More than 90% of short Remote and Initiator transfers are less than 3 data packets, which confirms our definition of short transfers (see Chapter 2). Thus long connections are often not able to trigger FR/R,
– The focus on large transfers shows that more than 47% of observed trains are less than 3 data packets. This again leaves TCP unable to trigger a fast recovery/retransmit, even if Limited Transmit is used. Hence, large transfer with short trains size can be penalizing and have a detrimental impact on recovery time.

The main conclusion from this study is that while short TCP transfers are penalized by TCP strategy which requires enough duplicates ACK to trigger its fast retransmission strategy (note that we use the term 'penalyzing' but no better strategy has been proposed in the literature). Long transfers are penalized by the application that sends too small burst of packets. Again, for the latter case, it is difficult to prescribe any improvement to this problem.

9.3 Anomalies within Data Transfers

9.3.1 Methodology

We next turn our attention to connections (or part of connections) that are not affected by retransmissions. We are left with the set-up, data transfer and tear-down times. We did not observe, in our data sets long set-up times, due, for instance, to the loss of the SYN or SYN-ACK packets. We thus do not consider this portion of connections in our analysis. On the other hand, tear-down durations arguably do not affect client perceived performance, though their actual value might be extremely large as compared to the set-up time. To highlight this fact, we present in Figure 9.3 the legacy throughput (total amount of bytes divided by total duration including tear down) and what we call the Application-Layer (AL) throughput where tear-down is excluded. We already see a major difference between those two metrics. If we are to reveal the actual performance perceived by the end user, we further have to remove the durations from the epochs where the user has received all data she requested from the server (which we detect as no unacknowledged data from the server to the client in flight) and the epochs where she issues her next query. We call this metric the Effective Exchange (EE) throughput. Those three metrics (throughput, AL throughput and EE throughput) are presented in Figure 9.3 and we can see that they present highly different views of the achieved performance and question the choice of a threshold to consider if we are to select anomalous connections based on a throughput metric only.

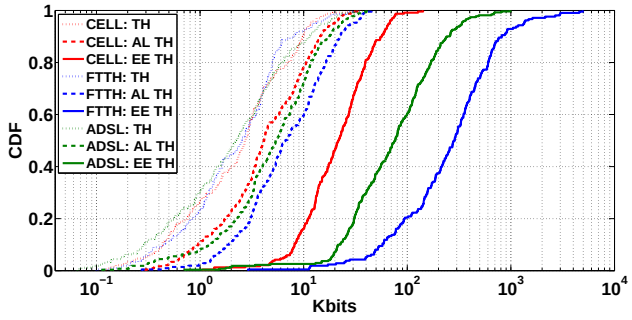


FIGURE 9.3 – Various Ways of Computing Connection Throughput

We build on the previous observation to derive a methodology that takes as input the actual duration of transfers and output 6 durations that sum the total transfer durations (complete partition). These are first the client [1] and server **Warm-up** times, where either the client is thinking or the server is crafting data, and the **Theoretical times** computed on the client and server side which represent the time an ideal TCP connection acting on the same path (same RTT but infinite bandwidth) would take to transfer all data from one side to the other. The difference between the transfer in one direction (say client to server) and the sum of thinking time and Theoretical time is due to some phenomenon in the protocol stack, e.g. the application or the uplink/downlink capacity that slowed down the transfer. We call **Pacing** those remaining durations.

The above methodology was presented in Chapter 5 to profile all client transfers. We aim here at using it to isolate abnormal connections. We adopt a simple approach : we isolate as potential anomalies connections that feature high values in (at least) one of the above dimensions. We exclude from our next analysis the think time at the client side, as client can spend large times to interact with the application on top of TCP. All the results presented here were obtained with a threshold vector formed by the 85-th quantile in all dimensions, i.e., a connection is flagged if its values in one or several dimensions is higher than the 85-th quantile in those dimensions. With this threshold, we restrict the analysis to 5% of the initial data volume. To discuss the choice of the quantile we report in Figure 9.4 the volume of data associated to each threshold.

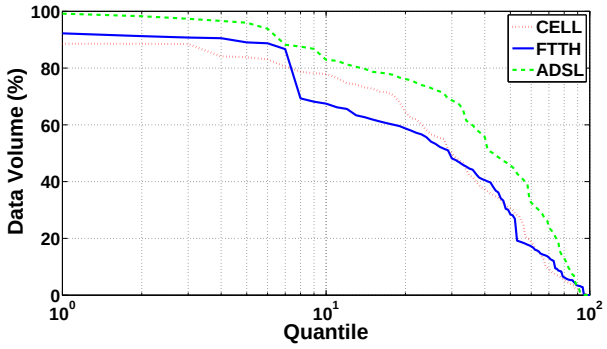


FIGURE 9.4 – Data time Break Down Quantile vs Data Volume

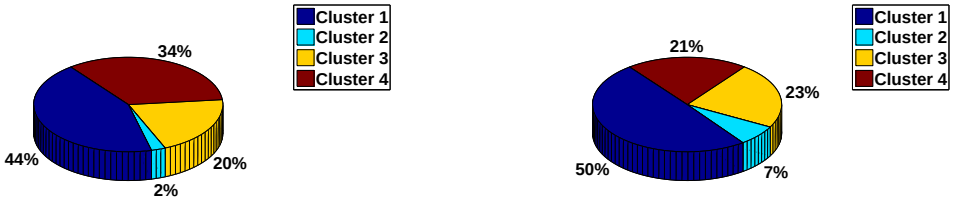
The application of the methodology to our traces will thus isolate connections that apparently miss-perform in (at least) one of the above dimensions. We need to make two important remarks : first, when a connection is flagged, it is not necessarily a real anomaly and some further analysis

1. The client is for us, the initiator of the transfer.

might be required. Second, to soften the previous point, the methodology we apply has the merit to not only isolate blindly a set of potential anomalous candidates but also to uncover the origin of the problem through the set of dimensions that were affected. We build on this richness of the methodology to cluster the a priori anomalies together, mixing all candidate connections on all access technologies. The idea is to observe if anomalies are shared or not and to categorize them. We use Kmeans to cluster those connections (again, for space constraint, we leave apart important details like the choice of the number of clusters) and represent the clusters we obtained in terms of boxplots - see Figure 9.6 – enriched with additional information on top of each plot like the fraction of connection for each access technology and also the median size of the transfers.

9.3.2 Results

Figure 9.5 enables a quantitative comparison of the clusters : one clearly sees that we have three large clusters in terms of bytes and connections, and one smaller cluster (cluster 2).



(a) Distribution of Volumes

(b) Distribution of Connections

FIGURE 9.5 – Overall Characteristics of Clusters

Figure 9.6 depicts boxplot representations of the clusters and the distributions of port numbers in each cluster. It enables a qualitative comparison of the clusters by comparing the relative size and position of boxplots. With this approach, we observe two groups of clusters : clusters 1 and 3 and clusters 2 and 4. The main factor that differentiate clusters within each group is the Pacing A value, which are higher in clusters 1 and 4.

Let us first focus on clusters 2 and 4. There are only few connections in those clusters for Cellular and ADSL technologies, while for FTTH, it is only cluster 2 that is small as cluster 4 aggregates 42% of FTTH samples. Let us first consider cluster 4, where TCP connections are characterized by extremely high Warm-up B and Pacing B values (median of several seconds). This suggests that the anomaly is located at the server side. However, we observe a predominance of port 1863 that corresponds to Microsoft Messenger. This application is highly interactive as it involves two humans, and we can conclude here that the anomaly is a false positive as what we believe to be servers or an application with low response times is in fact a human with long response times (time to read, think and write).

On the other hand, cluster 2 corresponds mostly to IMAP traffic. IMAP is used to download messages, hence there is little traffic from the client to the server, which results in negligible Pacing A in cluster 2. Cluster 2, unlike cluster 4, could thus represent a problem (cluster 4 with a majority of Microsoft Messenger connections corresponds to a false positive) where insufficient server resources have been allocated or alternatively a protocol anomaly.

Let us now turn our attention to clusters 1 and 3, which gather connections from the three access technologies. They are characterized by quite large values – median in the order of one to two seconds – on every dimension, except Pacing A for cluster 3. When looking at the port number,

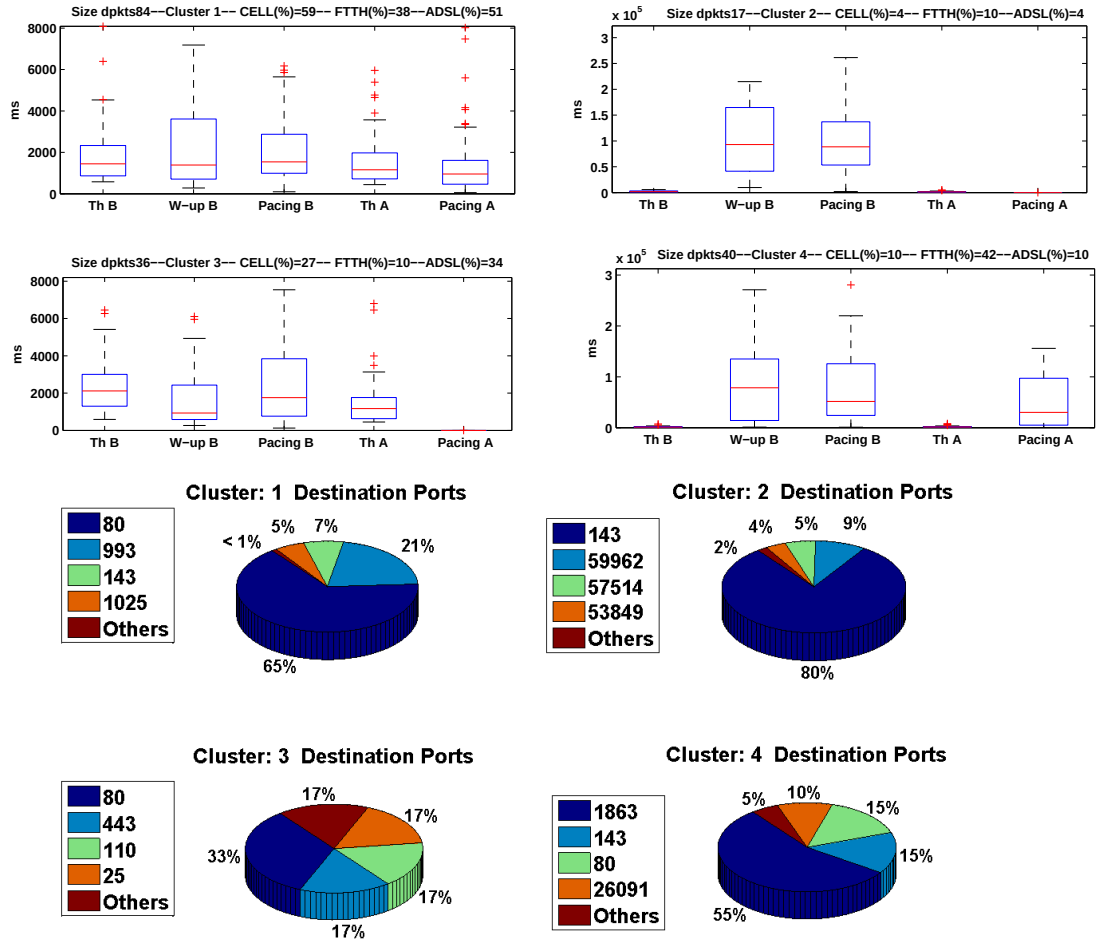


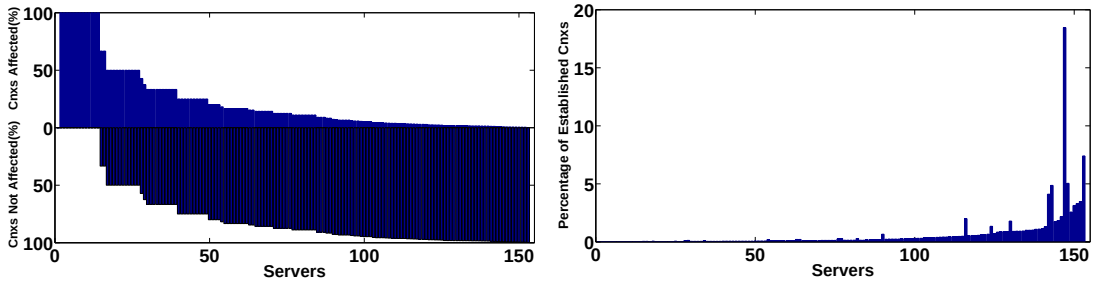
FIGURE 9.6 – Clusters : Threshold at 85-th Quantile

we can see that cluster 1 corresponds mostly to HTTP traffic while cluster 3 corresponds mostly to a mix of applications (based on port numbers) : HTTP, HTTPs, POP and SMTP. Large Warm-up B and Pacing B values indicates that the server is taking some time to prepare its response and also is slow to send packets. Again, this is a hint that there is some performance problem on the server side.

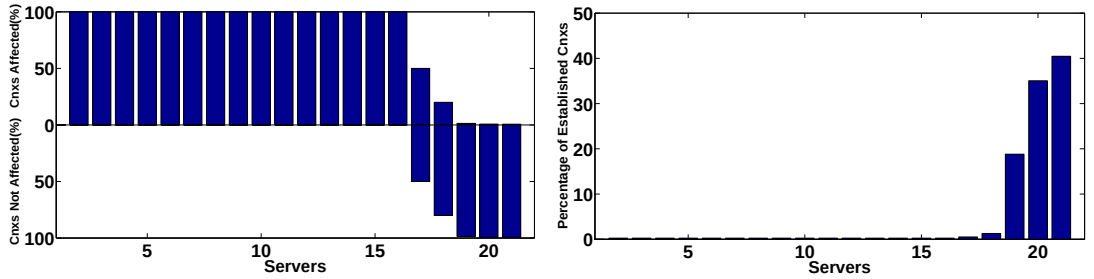
As for cluster 1, we observe that Th A and Th B on one side, and Pacing A and Pacing B on the other side are similar, which hints that the Web usage consists not only of pure downloads from the server to the client but a mix of uploads and downloads. We indeed observe a significant fraction of Facebook transfers in this cluster, which is a typical Web site involving more symmetric transfers.

9.3.3 Zoom on clusters 1 and 3

Clusters 1 and 3 have both a dominant application : HTTP for cluster 1 and IMAP for cluster 2. The question we address here is to quantify the extent of the anomaly. We can only rely on indirect means as we do not have access to those Web and IMAP servers. We proceeded as follows : we count, for each server, the total number of connections in our traces and the fraction of these



(a) Cluster1 : Web Servers



(b) Cluster2 : IMAP Servers

FIGURE 9.7 – Anomalies Locality

connections that have been declared anomalous. Results are reported in Figure 9.7. Let us first focus on the IMAP case. Focusing on the left plot, where each bar corresponds to a server, we can see it is a all or nothing situation : either all the connections of the server are anomalous according to our algorithm, or only a small fraction. This suggests the existence of configuration problems on the servers where almost all connections affected. For the other cases, this might just be transient phenomena. The right plot for IMAP shows that those servers (with 100 % of anomalies) are clearly unpopular as compared to the ones with only a small fraction of connection affected. We observe a similar situation for the HTTP case where have a set of fully anomalous servers, a set of transient anomalies and a class of servers in between (roughly, servers with id 20 to 40). Again, we observe that the fully anomalous servers are the less popular ones.

9.4 Conclusion

In summary, our clustering approach enables to narrow down the set of candidates when looking for anomalies in residential traffic. In addition, it provides for every connection flagged as an anomaly , a precise identity card of the connection to understand if the problem comes from the client side, the server side or the network. Clustering further enables to detect if groups of anomalies are spanning over several access technologies and applications or not. Clearly, our approach is not immune to false positives and requires to hand over to experts, once the location of the problem has been identified, but we believe that it is already a valuable tool that does a good job at narrowing down the crime scene. It could constitute a valuable tool for ISP and network engineers to quickly detect potential weaknesses in their network.

Chapter 10

Proposal to Locate Application Layer Anomalies in an Enterprise Environment

10.1 Introduction

Localizing performance problems in enterprise networks is challenging and difficult because little is known about enterprise characteristics and also available studies are made for specific enterprise networks. On the other hand, establishing ground truth, for some applications is complex as they are often not well documented and unlike Internet protocols. A typical example is Windows services and Enterprise Resource Planning (ERP) applications.

Nowadays, modern networks have many components/services that interact in complex ways. Client connectivity (VPN, Wifi, Ethernet,etc) and network architecture makes the task of traffic analysis more complex, since we have to deal with each case and to take into account constraints of network topology.

As Little is known about anomalies detection inside enterprise networks [58, 43], we propose in this chapter approaches to detect performance anomalies for enterprise networks. After a first overview of our Eurecom enterprise traces study in Chapter 8 we propose two approaches to differentiate anomalous connection from normal ones.

We show that the most popular applications such as SYMANTEC, SMB, LDAP and mail applications present problems of performance due mainly to the server side. We discuss results from each anomalies detection approaches.

10.2 Study Challenge

10.2.1 Client Access

We presented in Chapter 8 an overview of traffic analysis of the traffic traces we collected from the Eurecom network. We distinguished between 3 classes of traffics : client/server, DMZ and server to server. In this chapter we focus on client/server traffic analysis, since it corresponds to the majority of enterprise connections and traffic volume (see Section 1.7.3).

For the case of our trace, we identified 4 categories of clients, according to the user class or the type of its access link. Inside Eurecom we have student users, Staff users and Wifi users with their laptops ; On the other hand employers located outside the enterprise can use VPN connections to

access the enterprise servers and Intranet. Staff and students represent the majority of clients with respectively 70% and 26% of clients, while VPN and Wifi clients represent 4% of clients.

Figure 10.1 shows RTT distribution for the different clients classes. We group students and staff in the same category, since they use similar Ethernet access. We observe different RTT distributions for each class of client access. Inside the enterprise students and staff are characterized by short RTT, while VPN users have large ones similar to RTT for Internet traffic (see Chapter 4).

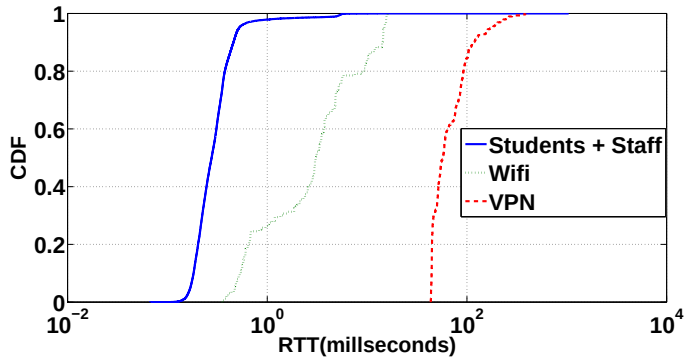


FIGURE 10.1 – Several Client Accesses

High RTT for VPN and Wifi users can bias the anomaly detection process, since we reproduce the approach presented in Chapter 9. In this case we will observe connections with high theoretical times A and B, in the same cluster could mistakenly be taken as an anomaly. To avoid this false positive impact, we concentrate our next analysis on traffic inside the enterprise from wired students and staff classes.

10.2.2 How to Define Anomalous Behavior

Before presenting results we note that, as we did for the Internet traffic, we have no prior knowledge of the threshold to consider in order to identify anomalies. As the Eurecom network is well dimensioned (at least short RTTs) we adopted an aggressive approach and we set the anomalies threshold to the 99-th quantile .

The research question that we target is the identification of anomalous TCP connections in enterprise environments that highlight factors that influence client perceived performance. The first parameter to investigate is connection reliability in terms of loss rate. We have observed in Chapter9 that invoking of loss detection/recovery mechanisms can be quite costly for TCP in terms of connection duration.

We observed for our trace that 0.5% of TCP connections experience loss/retransmission, which is a low compared to loss Internet accesses ratios presented in Section 9.2.2. In fact, enterprise and Internet traffic present different characteristics in terms of architecture and traffic load. This figure is in line with what has been observed in other studies [35].

We can conclude that only a small amount of traffic is affected by losses, and loss does not constitute the first factor that penalizes users in our enterprise trace. We next turn our attention to connections (or part of connections) that are not affected by retransmissions.

To investigate anomalies on application layer for enterprise traffic, we base our study on the methodology presented in Chapter 5 to profile all client transfers. We aim here at using it to isolate abnormal connections.

In Chapter 9, we adopted a basic approach to isolate a normal connection from an anomalous one. A Connection with potential anomalies corresponds to one that features high values in (at

least) one of the dimensions : : theoretical A/B, pacing A/B, warm-up A/B. We exclude the client side thinking time (warm-up A) as large thinking time at the client side do not mean anomalies.

10.3 High Quantile Metric

All the results presented here have been obtained with a threshold vector formed by the 99-th quantile in all dimensions, i.e., a connection is flagged if its values in one or several dimensions is higher than the 99-th quantile in those dimensions. With this threshold, we restrict the analysis to 9% of the initial data volume.

Figure 10.2(a) depicts the 6 clusters obtained by application of Kmeans and the distributions of port numbers in each cluster. We indicate, on top of each cluster, median connection size, percentage of samples in the cluster (compared to connection higher than 99-th quantile) and percentage of servers and clients. We notice 3 clusters (3, 4 and 6) with more than 28 data packets, while clusters 1, 2 and 6 correspond to short transfers with less than 8 data packets of connection size.

Before beginning to the interpretation of the individual clusters, we observe that clusters 3, 5 and 6 are identified by higher break-down values, while clusters 1, 2 and 4 present moderate ones. We use the percentage of active clients and servers to evaluate the popularity of the identified phenomenon and if it affects isolated clients/servers or not.

Let us focus on clusters 3, 5 and 6 where data transfers time is dominated by data break-down values on the server side. In cluster 5, Warm-up B dominates data transfers, while in clusters 3 and 6 we notice that Pacing B dominates.

Connections in these clusters represent 48% of connections identified with anomalies (higher than 99-th quantile).

From Figure 10.2 we observe that Cluster 5 corresponds to short connections with 2 data packets, it represents connections with large data preparation time at the server side. Destination ports results in Figure 10.2(b) shows that connections in cluster 5 corresponds to SYMANTEC traffic. Clients connect to the antivirus server in order to load updates for local virus list or to perform a check for their status, which generates several operations at the server side.

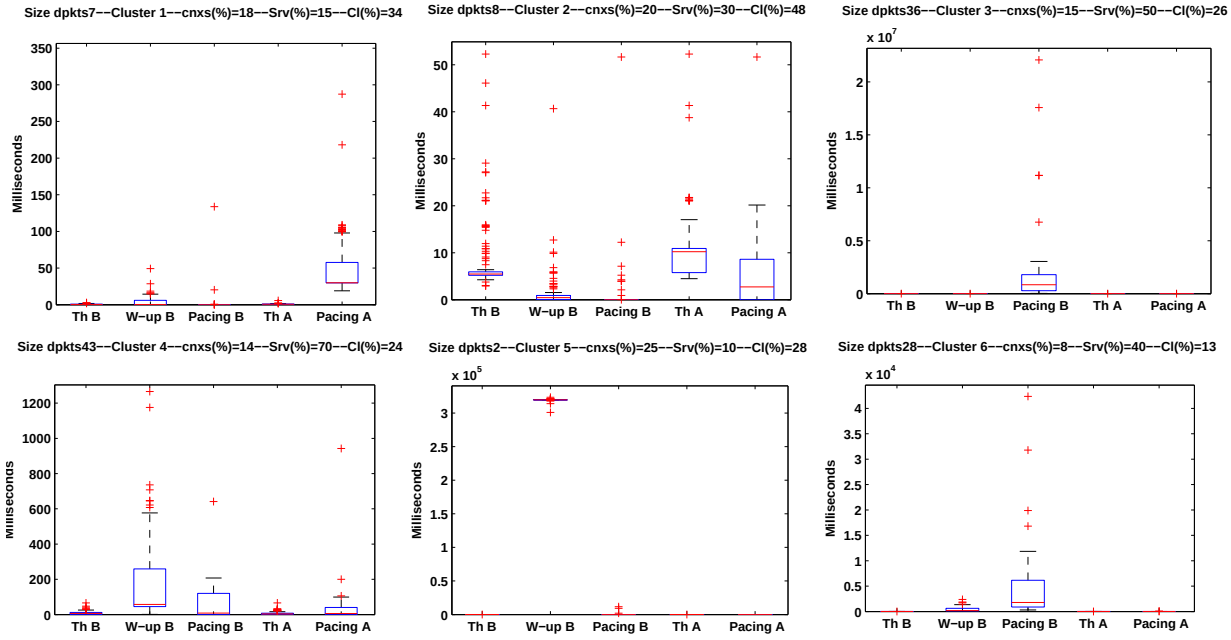
Clusters 3 depict large connections with higher Pacing B values. It corresponds mostly to LDAPS, SMB and IMAP traffics. These large Pacing B could thus represent a problem of performance, especially for IMAP where client need to upload mail without high waiting time. One explanation for this observation are application impact (large transfers with short train size) or insufficient server resources have been allocated for clients.

On the other hand, cluster 6 corresponds to connections with large Pacing B, with mostly LDAPS transfers. In this cluster we identified 8% of anomalous connections. While cluster 3 and 6 have are characterized by large Pacing B values, cluster 6 is identified by Warm-up B values.

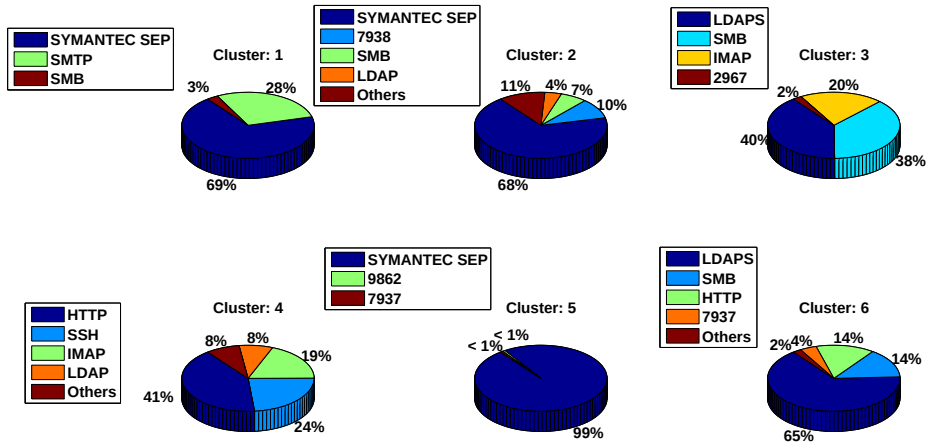
Let us now turn our attention to clusters 1, 2 and 4, which gather short and large transfers. A common characteristics of these clusters is the quite short median values on each dimension, compared with cluster 3, 5 and 6. We wonder if these clusters correspond to anomalies.

We noticed that cluster 1, with a median size of 7 data packets, is characterizes by large Pacing A. Port distribution in Figure 10.2(b) shows that traffic in cluster 1 corresponds mainly to SYMANTEC and SMTP. Probably large Pacing A in SMTP traffic corresponds to an application anomaly.

Cluster 4 corresponds to large connections, from a diversity of Intranet applications such as HTTP, SSH and IMAP. It is characterized by large values of Warm-up/Pacing B and Pacing A, but with median values clearly smaller in clusters 3, 5 and 6.



(a) K-means



(b) Port Number

FIGURE 10.2 – Anomalies Clustering

In Cluster 2 we observe shortest break-down values, compared to previous ones. It corresponds to mainly to SYMANTEC traffic. Median values of data times values do not exceed 15 milliseconds, a priori it does not influence client perceived performance.

To quantify the extent of impacted clients and servers with anomalies in each cluster, we proceeded as follows : for each cluster we compute the percentage of clients or servers that are present in a considered cluster and are not in the rest of samples. We obtain the two following vectors, one for clients and one for servers :

$$\text{Client} \begin{pmatrix} Cluster1 & Cluster2 & Cluster3 & Cluster4 & Cluster5 & Cluster6 \\ 1.6949 & 0 & 2.22 & 2.38 & 0 & 0 \end{pmatrix}$$

$$\text{Server} \begin{pmatrix} Cluster1 & Cluster2 & Cluster3 & Cluster4 & Cluster5 & Cluster6 \\ 0 & 0 & 20 & 21.42 & 0 & 0 \end{pmatrix}$$

Results show that a small fraction of clients in clusters 1, 3 and 4 does not exist in the rest of data, while clients in clusters 2, 5 and 6 are included in the rest of traffic. It does not seem significant enough to draw any conclusions.

On the other hand, the server vector shows interesting results ; We notice that 20% of servers in cluster 3, where connections are characterized by large Pacing B (several seconds of magnitude), are not included in the rest of transfers. As in clusters 3 and 6 users use approximatively same applications (mainly LDAPS and SMB) results in server matrix are different, one explanation is that this is the IMAP servers in cluster 3 suffer from an anomaly that generates large Pacing B.

Also, for cluster 4 we notice that about 21% of servers are unique and not included in the rest of data. This cluster includes different applications such as HTTP, SSH, IMAP and LDAP, with high values of Warm-up and Pacing B.

Those results would need more investigations. However, we ran out of time and were not able to confirm those results with the IT service of Eurecom.

Results from a comparison of affected clients and servers show that identified anomalies can be identified on specific servers, such as in clusters 3 and 4, while others not and observed anomalies depends more on others parameters (such as traffic load or others factors) that needs a more fine grained approach.

Finally, we report in Figure 10.3 the client OS or operating system, that we are able to distinguish using client IP address (a specific naming convention at Eurecom). The striking conclusion here is that, except for cluster 6, the majority of connections are established from laptop machines connected through their wired connections. This can suggest that laptop users are more prone to performance problems due to limited capacity of laptops in terms of processing compared with personal computers for the case of Eurecom users.

10.4 Outliers in Data Time Beak-down Values

In this section we discuss a second approach to detect anomalous connections. We adopt an approach similar to the one presented in the previous section with some differences in terms of computation of the upper bound limit of the normal behavior.

We define anomalous connections here as, outliers that feature, at least, along one of the dimensions a values higher than the upper bound limit (75-th quantile + 1.5 * interquartile range of the dimension to study) for each dimensions from data time break-down that identify each TCP connection. Note that this approach is similar to the one used in boxplot representations to identify outliers.

Also, as the same as the first approach, we exclude the client side thinking time as large thinking time at the client side do not mean anomalies.

Through this definition of abnormal connections, we intend to extract only high extreme behaviors that deviate from median and the range of most available values. With this threshold, we restrict the analysis to 5% of connections and 10% of the initial data volume.

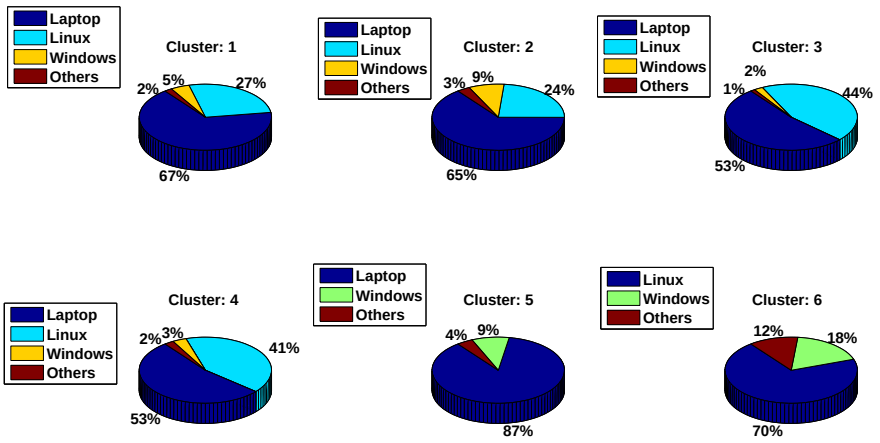
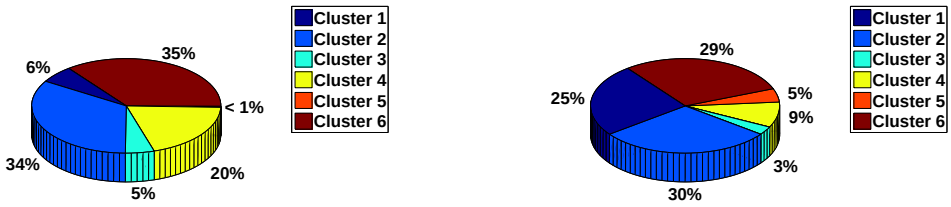


FIGURE 10.3 – Operating Systems and Client Machines

Figure 10.4 shows data volume and number of connections per each cluster. We notice that clusters 2, 4 and 6 represent majority of data volume, while cluster 1 aggregates 25% of connections and only 6% of data volume.



(a) Distribution of Volumes (b) Distribution of Connections

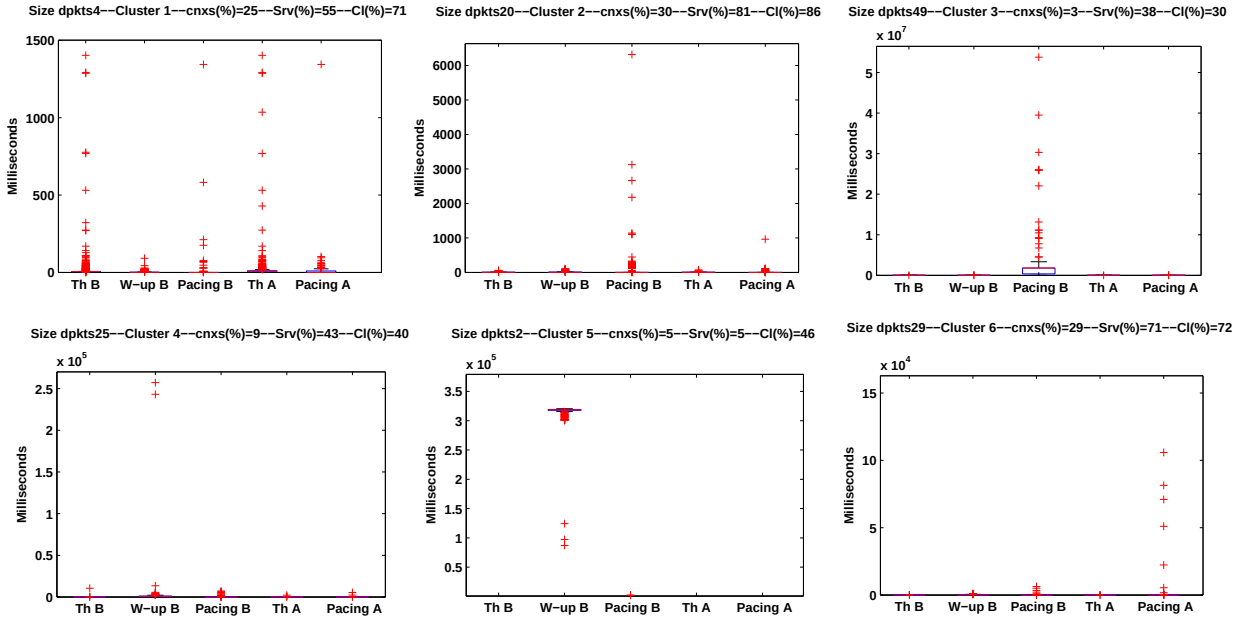
FIGURE 10.4 – Overall Characteristics of Outliers Clustering

We report in Figure 10.5(a) clusters obtained by the application of Kmeans and the distributions of port numbers in each cluster, for identified anomalous connections.

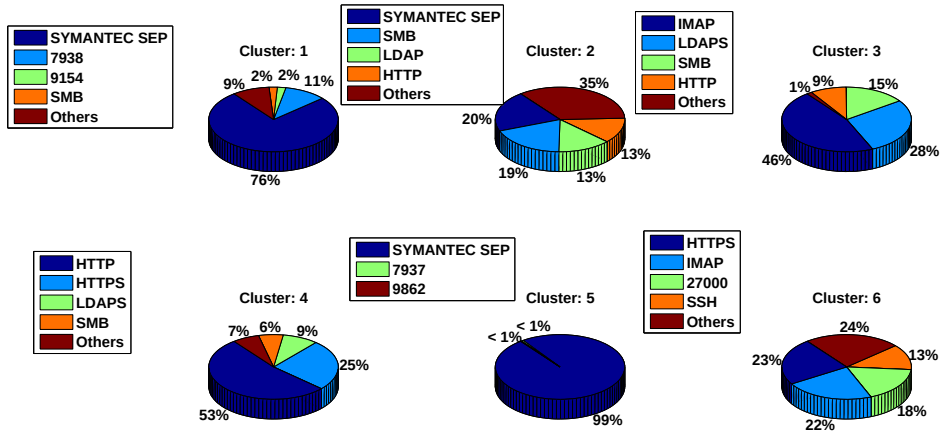
The first observation here is that we recognize in Figure 10.5 clusters 3 and 5 already identified with the first approach in Section 10.3, also in clusters 3 and 5. They are characterized respectively by (i) large Pacing B for the case of LDAP, SMB and IMAP and (ii) large Warm-up B for SYMANTEC flows.

On the other hand we find in cluster 1 connections identified by large theoretical times for a short amount of data packets (median size of 4 data packets), which suggests large RTT for these connections. A further investigation show that these connections correspond to ones established during backup process (see Section 8.2.1). This can highlight a problem of latency due to traffic over load. This phenomena covers 25% of selected anomalous connections and a majority of clients.

Cluster 2 is characterized by several hundreds of milliseconds of Pacing. For the case of this cluster we notice from Figure 10.5(b) that some of the popular applications in our trace are affec-



(a) K-means



(b) Port Number

FIGURE 10.5 – Clusters : Threshold at 75-th quantile + 1.5 * interquartile range

ted, e.g. SYMANTEC, SMB, and LDAP. In this cluster data time values appear reasonable and do not reveal, according to us, real performance problems.

Cluster 4 corresponds to a majority of Web applications, with less than 5% of data volume. This clusters have large preparation times and Pacing at the server side. While this behavior covers 9% of connections, it affects more than 40% of clients and servers. This cluster is in line with our experience of the internal Web server of Eurecom, which is connected to complex back-end applications and which is often quite slow.

Finally, in clusters 6 we have connections with large volume of data, with a majority of HTTPS, IMAP and IMAP transfers characterized by large Pacing A. This cluster involves 70% of clients ad servers. We hypothesize that these are fat transfers that are identified because of their

size and not because of their performance problem.

The study of client operating systems and machines, shows that in all clusters the majority of connections are established from laptop machines ; It confirms observations presented in Section 10.3.

10.5 Discussion

We proposed in this chapter 2 approaches based on different definitions of threshold in order to detect anomalous connections in enterprise environment. The first approach was based on selecting high quantiles for vector formed by data time break down values. We defined an anomalous connection, as connection where one or several dimensions is higher than the 99-th quantile in those dimensions. This approach allowed us to identify in clusters 3, 5 and 6 large times of Pacing and Warm-up in the server side.

The second approach is based on the study of outliers using the same approach as in boxplot representation. An anomalous connection features at least, one of data time break-down dimension values, that identify each TCP connection, higher than the upper bound limit (75-th quantile + 1.5 * interquartile range of the dimension to study). With this approach we recognised in cluster 3 and 5 two clusters identified with the first approach. Also, the rest of clusters depicts the impact of backup process in increasing RTT values, high Pacing A due limited performance for uplink traffic to servers : HTTPS and IMAP, and on the other hand client in cluster 4 with HTTP(S), LDAPS and SMB are penalized by large waiting time for response from the servers.

To wrap up, with these two anomalies detection approaches we identified different phenomena, several ones can be assimilated to anomalous service primitives, e.g. large Warm-up B for SYMANTEC traffic, and others with less degree of criticality, where the total connection time does not exceed hundred of milliseconds, e.g cluster 1 in Figure 10.2.

The impact of large Warm-up B in cluster 5 in Figures 10.2 and 10.5 can be expected not to be be very penalizing for client performance if the interaction with the SYMANTEC server is executed as the background tasks. On the other hand if this large treatment time on the server, is recurrent for the case of several requests at the same time for different users, it can globally increase server response time and the server/application setting needs to be re-evaluated

On the other hand, others identified phenomena can be more critical. We show that large mail (using imap), LDAP and file sharing (SMB) traffics depicted in cluster 3 in Figures 10.2 and 10.5 present large Pacing B. This suggests problem of performance on the server side in addition of the chatty characteristics of SMB protocol [100]. Cluster 6 in Figure 10.5, with higher data volume, shows large Pacing A values for several key applications, such as HTTPS and IMAPS. We highlighted in Figure 10.5, with a majority of SYMANTEC traffic, the impact of the traffic load, which generated large RTT.

At this stage, we can conclude that with the presented anomalies detection approaches we succeed to identify several behaviors of anomalous connections, with different degrees of criticality. The task of the definition of anomalous behavior was complex compared to the one presented in Chapter 9 mainly due to the enterprise environment characterized by specific applications,e.g SYMANTEC, RPC,etc. The main difference with Internet traffic is that the knowledge of each application behavior is very important (executed in background or not, etc). Presented approaches are not immune to false positives and require to hand over to experts, once the location of the problem has been identified, but we believe that it is already a valuable tool that does a good job at narrowing down several abnormal behaviors.

10.6 Conclusion

We tried in this chapter to propose approaches to detect anomalous connections, where an anomalous connection here is a functionally correct TCP connection – normal set-up/tear-down and actual data transfer – that experienced performance issues, e.g. losses or abnormally long waiting times at the server side. We succeeded to identify several phenomena, where some were common between the two approaches, especially large Pacing and Warm-up B and others not. We pinpointed factors that explain large observed RTT, Pacing and Warm-up B.

In the future, it would be interesting to consolidate the choice of anomaly threshold and to automatize the process of selecting anomalous clusters from normal ones. Also, it would be better to perform the study on a larger set of traffic traces, for example that spans over several days. Such analysis would help to understand how the client behavior and their performance limitations evolve over a large time scale. While the study over a single day presented in this chapter brings many useful insights and highlights performance problems for non trivial applications, it cannot be considered, alone, fully representative. The main goal of this study was to show how proposed approaches can be applied to produce the first results to guide further research in enterprise traffic analysis.

Conclusion of Part III

In Part III we focused on the issue of profiling anomalous TCP connections. Our method enabled to pinpoint the root cause of the performance problem, which can be either losses or some idle times during data preparation or transfer. We applied this methodology to several traces corresponding to Internet and enterprise traffics. We demonstrate the existence of specific strategies to recover from losses on Cellular network that seem more efficient than what is done currently in wired networks. When focusing on the transfers or parts of the transfers that are not affected by losses, we demonstrate that our approach is able to detect and classify different classes of anomalies, especially anomalies due to transient or persistent - provisioning - problems at the server side.

In the last concluding chapter of the thesis, we reevaluate the thesis claims made in the beginning and evaluated how we fulfilled those claims. We also discuss the possible future directions of research for which this thesis has laid the basis.

Thesis Conclusions and Perspectives

Internet performance has been measured in various ways since its inception as the ARPANET in 1969. There have been a number of trends that have affected the way the Internet has been measured over this span of time. Some trends depend on the technology improvement : Internet technology has changed over time, which has made some measurements more difficult and some measurement easier to obtain. Other trends are matters of scaling : the prodigious growth of the Internet has changed metrics to measure for performance evaluation, and has triggered the development of new measurement methods and statistical tools. Finally social trends, the transition of the Internet from government funding to private operation and the economic significance of Internet communication have altered the kinds of measurements needed and the extent to which certain measurements can be made.

These trends have been driven by the interplay between measurement goals and measurements difficulties. In this thesis, we reviewed different difficulties that can face experts when collecting data with new available architectures (Internet and enterprise environments) and then, we proposed a new methodology to highlight new parameters that can influence client perceived performance. Finally, we discussed approaches to detect anomalies in Internet and enterprise environments.

In this final chapter we seek to synthesize some salient characteristics of Internet and enterprise traffic measurement to show problematics where we succeed to progress, and where more efforts have to be performed. We now revisit the thesis claims and discuss the thesis work in general, highlighting the main contributions. Finally, we give our vision on how this research could be extended in the future.

Short Transfers and Application.

While analyzing the performance of TCP transfers, we focused on the connections that correspond to valid and complete transfers, from the TCP perspective, that fulfill the following criteria : a complete three-way handshake, at least one TCP data segment in each direction, and the connection must finish either with a FIN or RESET flag.

In Chapter 2, we introduced a first definition of a short TCP connection, which is commonly used in the literature. *A short TCP connection is a well behaved connection unable to perform fast retransmit/recovery (FR/R), after a packet loss detection.* We presented an overview of the impact of the application, on the TCP transfers. We showed that while losses can have a detrimental impact on short TCP transfers, the application significantly affects the transfer time of almost all short - and even long - flows in a variety of way.

We demonstrated that the sensitivity to loss concerns also many long transfers as many of them are a sequence of alternate exchanges and the vast majority of these bursts are less than 3 packets. Such a feature has a direct influence on the ability of TCP to recover from a loss using Fast Retransmit. We observed that the application can induce extremely large tear-down times and it can also slow the rate of TCP transfers.

ISP Architecture have to be Taken in to Account. In Chapter 3, we highlight that in modern Cellular networks, estimating latency turns out to be a complex task. We demonstrated that latency can be under estimated due to the use of new mechanisms or services, like proxies for content adaptation or applications acceleration. We investigate how these mechanisms impact our measurements and the performance perceived by end users. The key message here, was that several specific devices might affect classical performance metrics in Cellular networks, which should be taken into account when performing measurement studies.

Usual suspects are not enough to explain Performance. We used in Chapter 4 a classical approach to compare performance of different access technologies : Cellular, FTTH and ADSL in order to conclude if clients fully benefit from their broadband access. We focused on the two key factors that influence the throughput of TCP transfers (TCP throughput formula [89]), namely loss rate and RTT, that suggest that the performance over FTTH should significantly outperform the one of ADSL, which should in turn outperform the one of Cellular. It turned out that reality is slightly more complex. While the Cellular technology offers significantly smaller throughput, in line with RTT and loss factors, FTTH and ADSL have much closer performance that RTT and loss were suggesting. We conclude that focusing on classical parameters of performance analysis does not lead to a full understanding of client perceived throughput.

Fine Grained Analysis. We proposed mainly in Part II a method that drills down into the data transfer of each well-behaved connection. The developed approach is exemplified with the set of traces collected on the Cellular/FTTH and ADSL backbones of Orange. Proposed data time breakdown approach automatically extracts the application, access, server and client behavior impacts from passively observed TCP transfers and then group together, with an appropriate clustering algorithm, the transfers that have experienced similar performances.

Application of this technique to the Google Web search service demonstrated that it provides easily interpretable results. It enables for instance to pinpoint the impact of usage or of raw characteristics of the access technology. We demonstrated that user behavior dominates clusters with large volume of data packets and connections. This explains the similar behavior of FTTH and ADSL as response time is dominated by Warm-up A.

Also, we further compared Yahoo and Google Web search traffic and provided evidences that they are likely to adapt content to the terminal capability for Cellular clients which impacts the performance observed. Cellular clients featuring a laptop/desktop Windows operating system (Vista/XP/2000) experience similar warm-up B as ADSL clients while clients using Iphones or a Windows-CE operating system experience way higher warm-up B.

Proposal of Approaches to detect Anomalous Behaviors. We profiled in Part III anomalous TCP connections that are defined as functionally correct TCP connection but with abnormal performance. Our method enabled to pinpoint the root cause of the performance problem, which can be either losses or some idle times during data preparation or transfer. We applied this methodology to several traces corresponding to Internet and enterprise traffics.

We demonstrated that common wired (ADSL and FTTH) and wireless (Cellular) technology adopt different strategies to recover from packet losses, especially under time out conditions (time outs being prevalent in all environments over fast retransmits), and that the strategies observed on the Cellular technology seem more efficient than on ADSL and FTTH.

We showed that our methodology for profiling the transfers (or parts of transfers) unaffected by losses is able to uncover various types of anomalies, some being related to the configuration of servers and some other being shared by several services.

On the other hand, through the study of an enterprise environment, we argued two anomalies detection approaches. We succeed to identify several behaviors of anomalous connections, with different degrees of criticality. The task of the definition of anomalous behavior was complex compared to the Internet one, mainly due to the enterprise environment characterized by specific applications,e.g SYMANTEC, RPC,etc. The main difference with Internet traffic is that the knowledge of each application behavior is very important (executed in background or not,etc).

Finally, the approaches presented are not immune to false positives and requires to hand over to experts, once the location of the problem has been identified, but we believe that it is already a valuable tool that does a good job at narrowing down several abnormal behaviors.

On the other hand, we identify future research tasks and directions in three categories : first, related to the methodology, second, the scale of analysis and, finally, the architecture of the used approach.

Leaving the connection level of analysis. In this thesis and in the thesis of Matti Siekkinen [101], the emphasis was put on the analysis of individual connections. While it turned out to be a rich and complex topic, which enables to obtain many insights concerning the performance perceived by the end users, it bears specific limitations. A crucial one is that the dependency between flows is not taken into account. Trivial but important examples are the case of DNS queries prior to a lot of application specific connections in IP context or parallel HTTP or P2P transfers. Some techniques [66] have been proposed to automatically extract connections inter-dependency, which is a valuable approach as, similarly to what we do, the application semantic can be ignored, i.e. no details about the specific application needs to be cast in the algoritm. Also, a lot of works have proposed graph approaches [65, 102, 83] to identify application or user behaviors. Such approaches are interesting as they provide high level overview of clients and application behaviors. However, it is difficult to troubleshoot the network with those techniques. An interesting continuation of this work, could be to combine those types of approaches (at the session or application or user level) with our low level approach at the connection level to better inform the results obtained during the clustering process we use.

Large scaled analysis. We faced, in our work a problem that is common to a lot of traffic analysis study : we spent a lot of time developing and calibrating our analysis techniques and due to the limited size of our traces, our results are not established on a fully solid ground. We mean that it is difficult to know if our results are limited to the setting of the network for which we have traces or if it general for the technology we consider, e.g., ADSL or Cellular. In our opinion, this weakness does not affect enterprise as the latter is by definition, specific to a location. However, in the latter case, the advantage is that, in some cases (like the Eurecom network) all the network and all the traffic can be simultaneously observed, which is obviously more difficult at the Internet scale. We expect that a continuation of the present work will be on applying the methods we developed on large variety of traces, e.g. several Cellular traces from the same GGSN or several days/weeks of enterprise traffic.

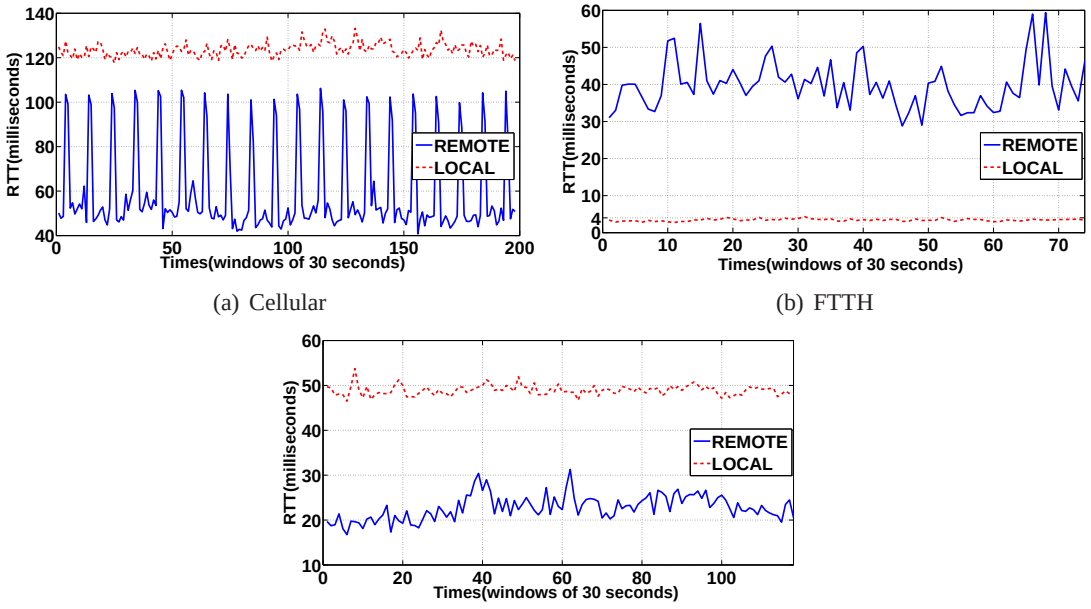
Cloud computing. Cloud computing is not only a buzz word of the moment but is likely to become the future of the data network in a lot of scenarios. In such context, remote servers are accessed by end users and pinpointing the performance problems becomes crucial in these complex environments both from a networking but also from a system point of view, e.g. server consolidation with virtualization. We hope and expect that the method we developed could constitute valuables tools to diagnose the performance issues in this context.

APPENDIX

Appendix A

RTT Stability for the Cellular, FTTH and ADSL Traces

The RTT stability measurement methodology works as follows : we compute for each slice of 30 seconds, the median values of local and remote RTT, we use data/ack estimation method. Remember that the syn/acka approach was not appropriate for the Cellular scenario and that syn/ack and data/ack approaches were giving comparable results for the other access technologies.



(a) Cellular

(b) FTTH

(c) ADSL

FIGURE A.1 – RTT median : over time windows of 30 seconds

FigureA.1 s that local RTT values constitute a signature of each network access impact. For instance, lower values of local RTT were detained respectively by FTTH, ADSL and finally Cellular.

Unfortunately we can not draw conclusion from the comparison of remote RTT due to several factors, the fact that traces haven't been simultaneously collected (not the same client and server load) and specific handling in Core Network between FTTH/ADSL on one hand and Cellular on the other hand.

Figure A.1 shows that 3G remote RTT is more unstable and variable than in FTTH and ADSL access. In fact we can perceive high and periodic fluctuations of 3G remote RTT in the temporal profile : for each slice of 10 minutes we observe two peaks of 105ms. After further explorations we found that this phenomenon was previously observed in 3G Cellular networks [103] : They found that the primary causes of remote RTT spikes are scanners. The probe traffic generated by a sequential high rate scanning source causes an arrival rate pattern at the peering link. It mirrors the address space allocation of the local network. If the scanner source keeps cycling into the address space, such patterns will appear periodically. We did not investigate further this problem and we were not able to confirm or deny this hypothesis of scanning activity.

After focusing at local RTT, it is interesting to note that FTTH access offer shorter and stable local RTT (between 3 and 4ms) compared to ADSL and radio access, which is a consequence of the used architecture in each access. Fiber client access is based on Ethernet unlike ADSL which uses ATM until Orange's core network.

Appendix B

Data Time Break-down for Mail and Webmail Traffic

B.1 Webmail : Clustering Results

Figures B.1(a) and B.1(b) present results for Orange Webmail clusters, with Kmeans and a graphical representation using T-SNE method.

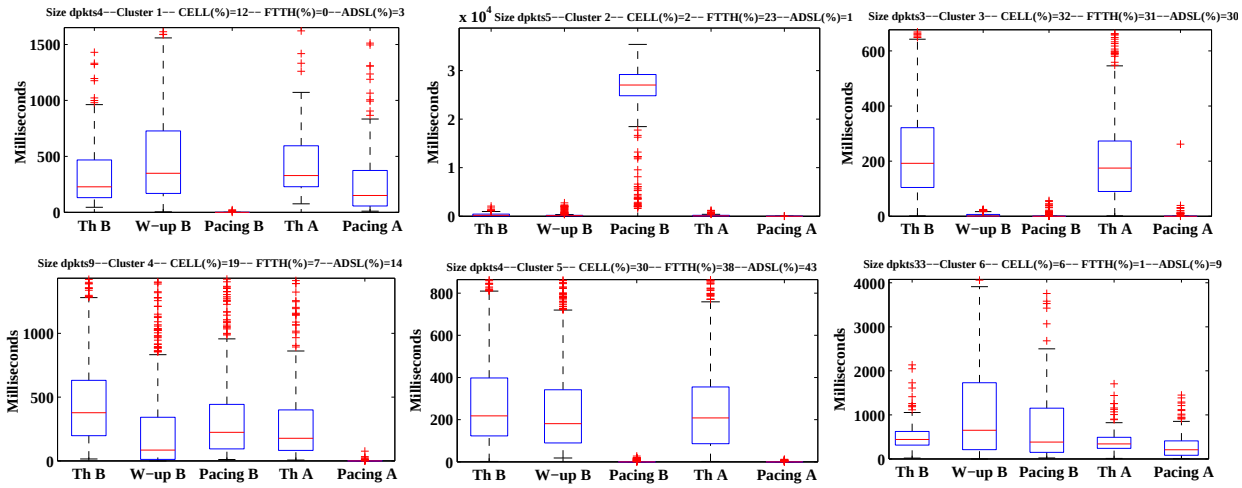
As a general remark, we noticed a good match between clusters identified with Kmeans algorithm and T-SNE, presented in Figure B.1(b). In fact, we observe that Kmeans clusters, showed in the left plot of Figure B.1(b) are easily identified using their cluster ID from Kmeans. We noticed a number of clusters between 4 and 6. After several trails attempts we fixed the number of clusters to 4.

The main observation from this clustering is that the majority of clusters are short sized, except cluster 6 with 33 data packets of median connection size. We compare in Figure B.2(a) CDFs of Warm-up B for obtained clusters. We noticed similar values of Warm-up B (which approves our break down methodology : for the same service we have a high probability to obtain the same server response time for different client accesses). It was not the case of cluster 3, because it corresponds to shortest connections with a median size of three data packets.

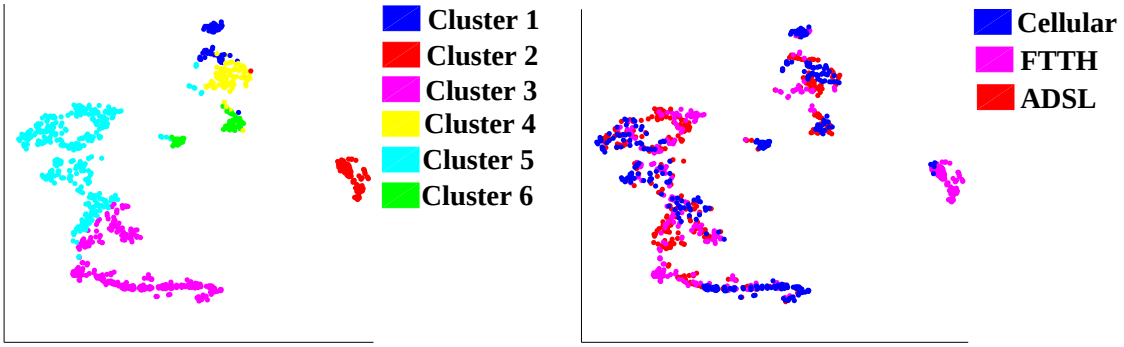
To better understand the main discriminant parameters that influence the obtained clusters, we investigate clusters obtained with t-SNE algorithm and depicted in Figure B.1(b). This figures indicate three categories of clusters according to the spatial distributions of plotted connections. We see that clusters 1, 4 and 6 are situated on the same area. Then, clusters 3 and 5 are very close and have common border. While cluster 6 is far from others clusters which leads us to think that it can reveal an non-trivial behavior.

From the study of clusters 3 and 5, as we observe from the median connection size, they corresponds to the shortest connections respectively with 3 and 4 data packets and distributed equitably from each used trace, for the two clusters. Next, we compare in Figure B.2(a) Warmu-up B values for each cluster, we see that cluster 3 is identified by shortest ones - that suggest a common usage of this class of connections for different users and it is not correlated with the used access. In fact after further investigation we noticed that connections in this clusters correspond to very short connections used for users authentication or images download.

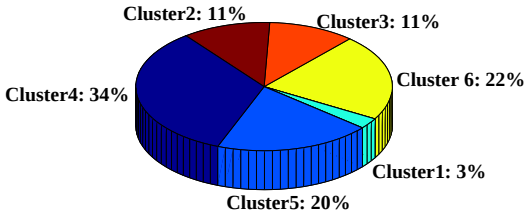
To describe clusters 1, 4 and 6 we will base our analysis on pacing values. In fact these clusters have similar Warm-up B and theoretical transfers times. Cluster 1 and 6, with a majority of connections from Cellular and ADSL accesses, group Webmail connections with Pacing A values. Which highlights throughput limitation over the uplink access, while cluster 4 shows no Pacing A.



(a) K-means



(b) TSNE



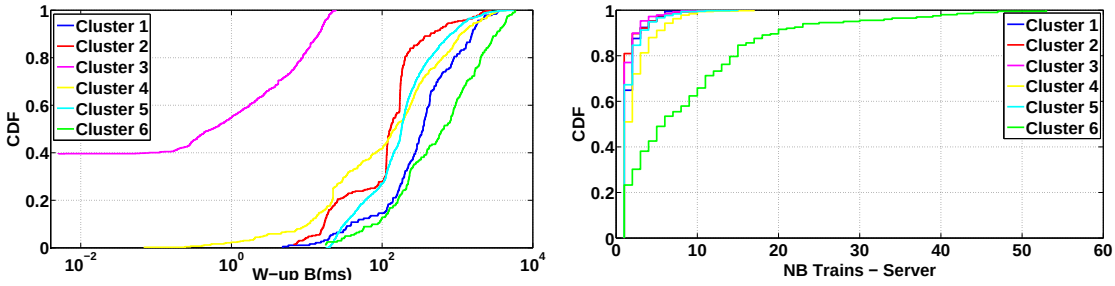
(c) Data Volume per Clusters

FIGURE B.1 – Clusters : Webmail Traffic

Cluster 1 and 6 have similar behavior in terms of Theoretical times, Warm-up B and Pacing A, but on the other hand they present different connections size with large connection size for cluster 6, and different Pacing B values (no Pacing B for cluster 1).

Finally, cluster 2 is an interesting cluster, because it groups connections from all accesses, characterized by a high Pacing B.

We noticed that connections in this cluster are mainly from Cellular and FTTH accesses and



(a) W-up B (b) Trains

FIGURE B.2 – Webmail : Warm-up B and Trains

target Gmail servers. Then in order to understand the observed high values of Pacing times, located on only Gmail Webmail server, we analyse connections with this feature. We observed that Gmail server adds a large delay between TCP segments in a same train of data, which highlights the service impact.

We summarize in Figure B.3 the main characteristics of each identified clusters and the common features and differences.

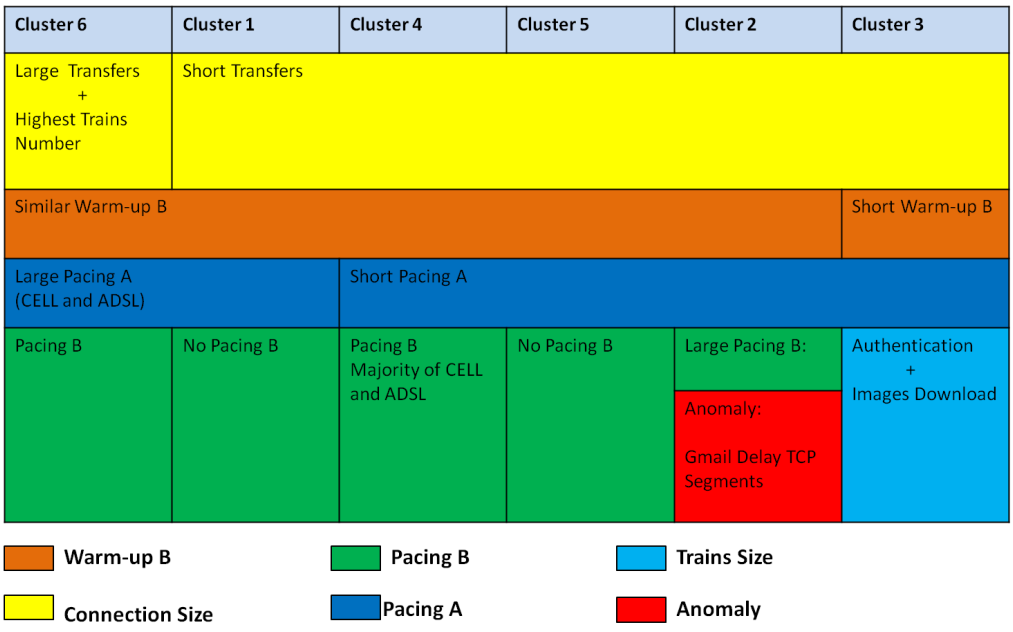


FIGURE B.3 – Overview of Webmail Clusters

We present in Figure B.4 the scatter plot of each connection start time versus the cluster ID. It shows that Webmail connections are spread over the capture time of each considered trace. It indicates that clusters are not located over a specific slice of time but equitably distributed in Celllar, FTTH and ADSL captures.

Figure B.4 shows that anomaly corresponding to connections with large Pacing B (Gmail delay TCP segments) in cluster 2 are displayed over all captures. It excludes that this performance problem is correlated with the server or network load.

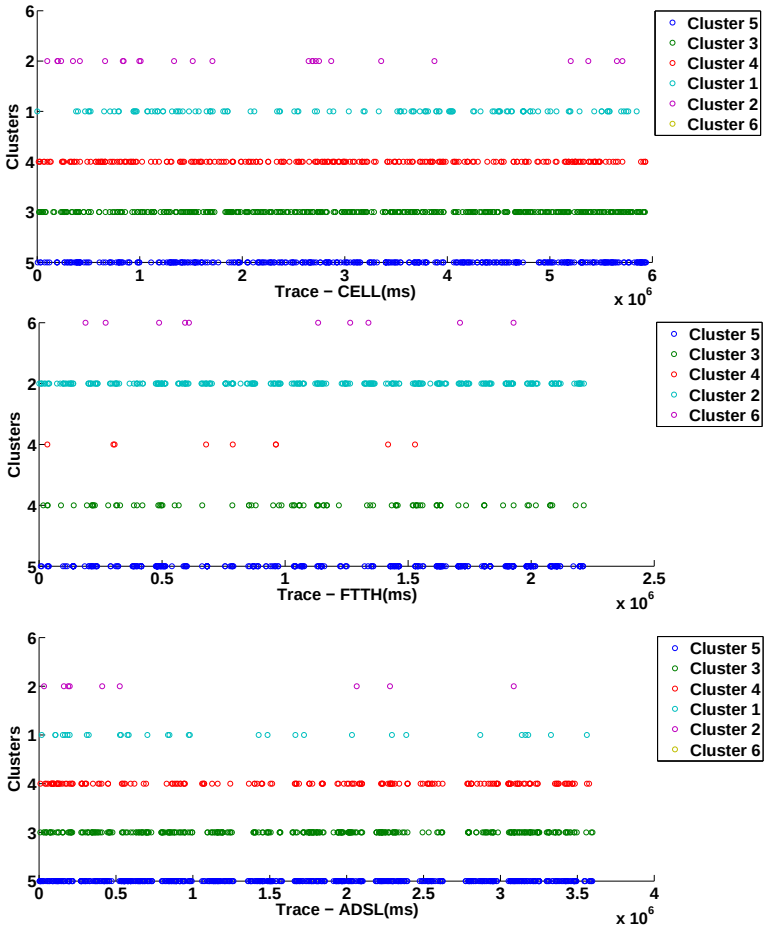


FIGURE B.4 – Webmail : Clusters vs Time-stamp

B.2 Orange Mail Service

We illustrate the issue of comparing access technologies with the case of email traffic. Before going into the details of email traffic in our traffic traces, we note that :

- Mail is a key application from the end user point of view and while most of the work has focused on trendy applications, e.g., p2p or social networks, mail has received little attention in previous works ;
- Mail is a versatile application as it can run over HTTP (Webmail) or through direct interactions between the users and the mail servers using POP3 or IMAP (from the server to the client) and SMTP (in the opposite direction). From now on, we will refer to mail traffic to denote POP3/IMAP/SMTP only and distinguish it from Webmail.
- Mail is a rich application from the traffic analysis point of view as the interactions between the user and the mail server can be interactive (mail checking with no available mail, mail headers download via IMAP or small mail downloads via POP), or can be considered as bulk transfers (large mail uploads/downloads).

We take a stance in this work to focus on mail traffic because (i) Mail traffic can be readily identified using port numbers unlike Webmail, which needs further filtering heuristics to delineate Webmail from Web traffic and (ii) it is easy to categorize mail exchanges from client to server and from server to client because different protocols are used. The characteristics of mail traffic

in our traces are presented in Table B.1. We restricted our study to POP3 traffic in the down link direction and SMTP in the up link direction. Indeed, IMAP and IMAPS are popular on the Cellular access only. We also notice that SMTP traffic appears more popular for FTTH and ADSL than for Cellular access. This might be attributed to the fact that the users tend to use their Cellular access to check their mail but defer the answering of those mails to the moment in which they will have a more convenient wired access.

	Cellular	FTTH	ADSL
SMTP	4988 (4.05%)	11364 (65.52%)	7555 (29.69%)
POP3	44200 (35.92%)	5125 (29.55%)	17295 (67.96%)
IMAP	35431 (28.79%)	97 (0.55%)	254 (0.99%)
SMTPS	2342 (1.9%)	37 (0.21%)	4 (0.01%)
IMAPS	29062 (23.62%)	172 (0.99%)	153 (0.60%)
POP3S	7002 (5.69%)	547 (3.15%)	185 (0.72%)

TABLE B.1 – Mail Traffic

B.2.1 ASP Mail service : a First Look

Throughput is an appealing candidate to compare implementations of a service over different access technologies. Even though we pinpointed that email is a complex application that mixes interactive and bulk-transfers usage, we can expect that throughput allows us to draw first conclusions on the impact of the access technology.

Figure B.5 shows cumulative distribution functions of the application level throughput, for the considered traces, in both uplink (SMTP) and downlink (POP3) directions. We also present in Figure B.6, the size of transfers in data packets (bytes profiles are similar to packet profiles). Based on these figures, we formulate two hypotheses :

Hypothesis 1 : Distributions of the amount of packets transferred per connection in the down direction (over POP3) are similar for the three access technologies. Therefore, the observed difference in throughput should be a function of the latency of the path as well as of the application. Indeed, the low rates observed clearly suggest that the raw capacity of the access technology is not the dominant factor that is limiting the performance. Also, as transfers are small in size, we cannot expect TCP to fully utilize the link capacity.

Hypothesis 2 : Distributions of the amount of packets transferred per connection in the uplink direction (over SMTP) differ between the three technologies. Transfers are small for the Cellular and FTTH traces, while they can be fairly large over ADSL. This means that a priori three factors influence the observed throughputs : Latency and the application for all three technologies and possibly the raw capacity for the case of ADSL.

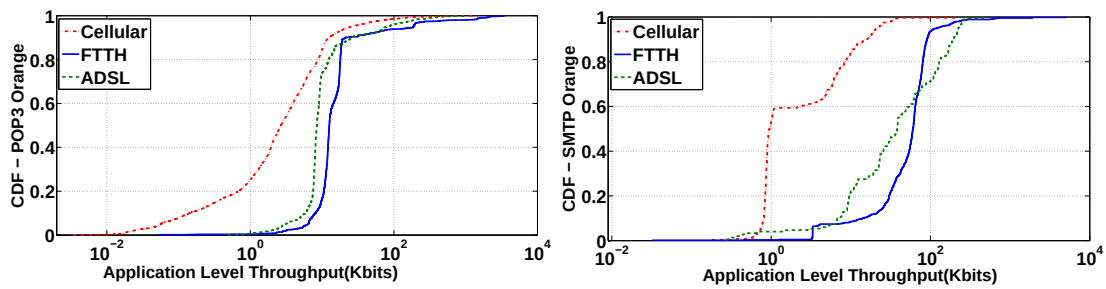


FIGURE B.5 – POP3 vs SMTP for Orange server : AL Throughput

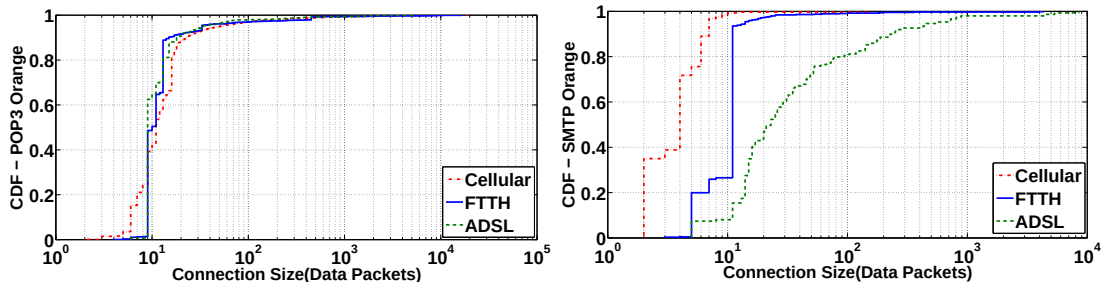


FIGURE B.6 – POP3 vs SMTP for Orange server : Connection Size

Throughput and transfer sizes alone are clearly not enough to validate Hypotheses 1 and 2. Scatter plots of throughput and transfer sizes could help to take usage into account, but they are difficult to interpret in practice. We apply in the next section, our break-down method presented in Chapter 5 that is fully application agnostic but nevertheless allows to assess the relative impacts of application and access technology, at the two sides (client and server) of a transfer.

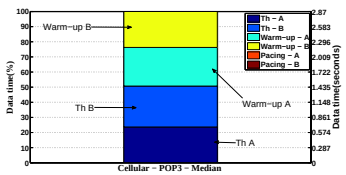


FIGURE B.7 – POP3 - Cellular

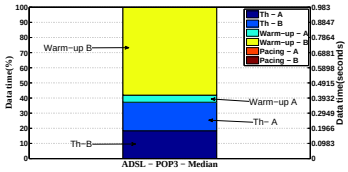


FIGURE B.8 – POP3 - ADSL

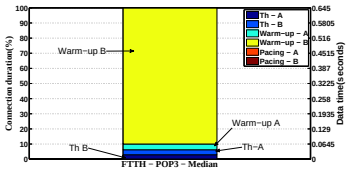


FIGURE B.9 – POP3 - FTTH

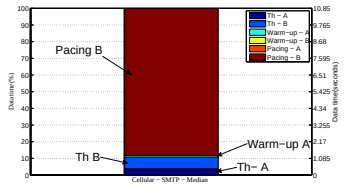


FIGURE B.10 – SMTP - Cellular

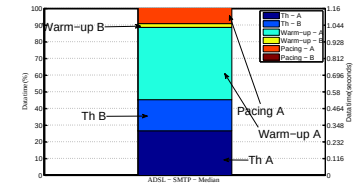


FIGURE B.11 – SMTP - ADSL

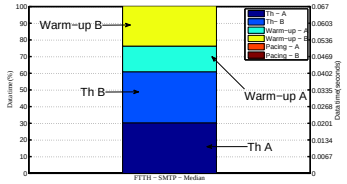


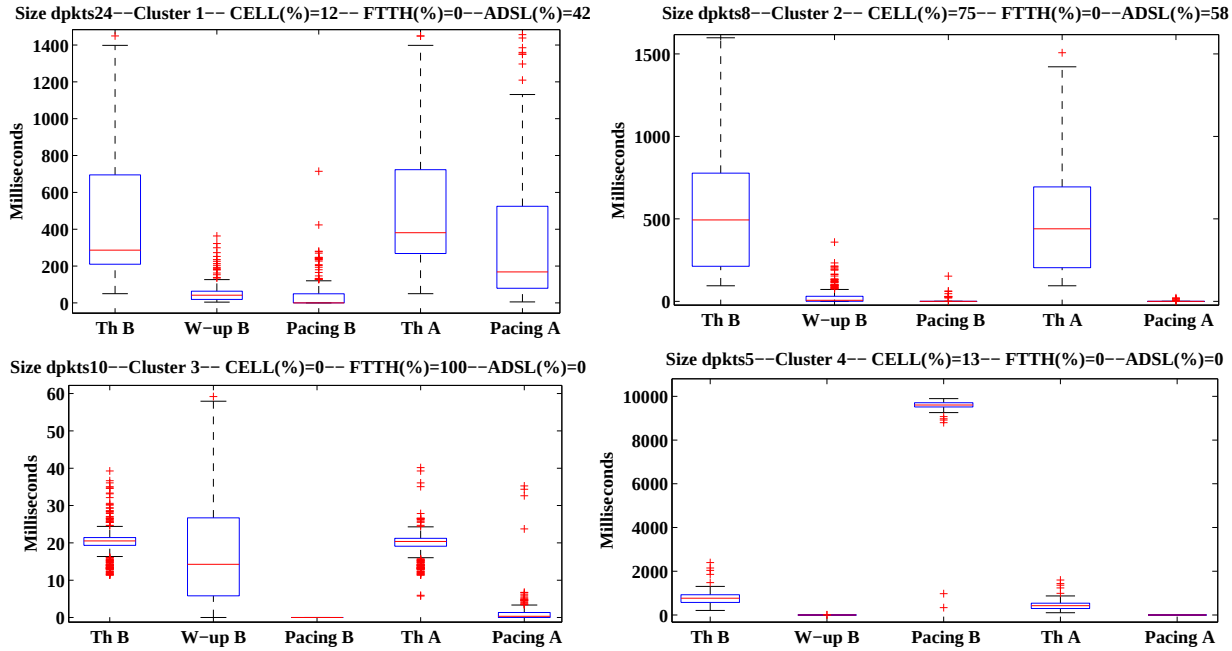
FIGURE B.12 – SMTP - FTTH

Let us now focus on the relative values observed per access technology and mail direction. For the case of POP3, it is almost exclusively the theoretical and warm-up times that explain the total data time. As noted above, warm-ups are similar for the three traces. Our methodology enables to observe the relative shares of theoretical times and warm-ups. Clearly, these relative values vary according to the latency of the access technology : For FTTH, the warm-up at the server side dominates, while in the case of Cellular access, the theoretical times are much higher because of larger latencies. The share of Pacing is negligible, most likely because a majority of downloaded mails are short or there is no mail to transfer. In summary, path latency (a clear function of the access technology) and server response times (i.e., the application) dominate in POP3 transfers.

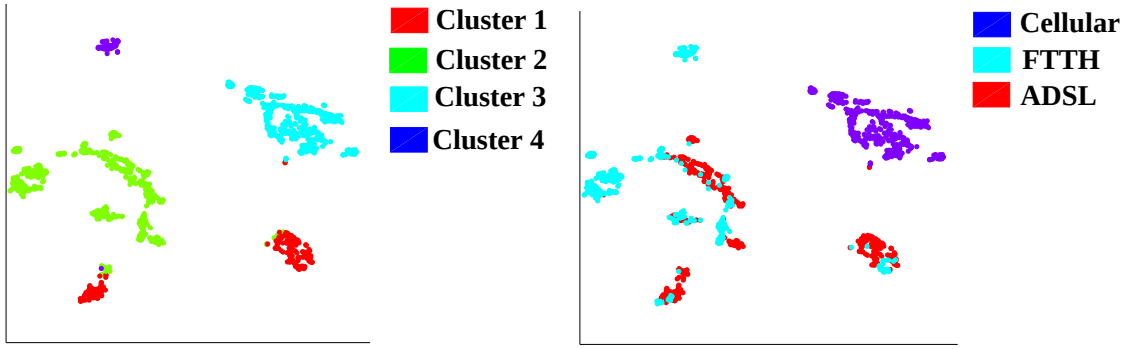
As for SMTP, one would expect theoretical times and possibly pacing on the client side (A) to play a more important role, especially for ADSL and Cellular access due to the low uplink capacity offered by those technologies. Also warm-ups on the server side should be significantly smaller as this is mostly the client side that does the job, i.e.,pushing the mail up to the server. This is indeed what we observe with ADSL where the Pacing-A is increased compared to POP3. In the case of FTTH, latency dominates (note that mail transfers are small over FTTH as can be seen in Figure B.6) as opposed to the other costs, which is visible as high theoretical times. Cellular access, on

the other hand, provides a very different picture with a high and unexpected cost due to pacing on the server side.

B.2.2 SMTP : Clustering Results



(a) K-means



(b) TSNE

FIGURE B.13 – SMTP Orange Clusters

Figure B.13(a) depicts 4 clusters obtained with Kmeans. We first observe that, theses clusters coincide with the projection obtained by t-SNE as indicated in the left plot of Figure B.13(b), where data samples are indexed using their cluster ID in Kmeans.

Before going into the interpretation of the individual clusters, we observe that two of them carry the majority of the bytes. Indeed, Figure B.2.2 indicates that clusters 1 and 2 represent 89% of bytes. Let us first focus on these dominant clusters.

The first observation from clustering results is that client access is the main discriminant parameter. For instance, we find all FTTH connections in cluster 3, 13% of Cellular connections in cluster 4, while in clusters 1 and 2 we identify ADSL and Cellular connections.

Cluster 1 corresponds to the largest connections, comparing to the reset of clusters with a mean value of 24 data packets. Connections in this cluster were identified by a large Pacing A. In fact, we expected to find pacing values on the uploading (data sent from the client to the server) due to the limited capacity of the Cellular and ADSL accesses. Also, one other parameter can explain this Pacing, if we look to the number of exchanged trains in Figure B.15(a) we notice that connection in this cluster were characterized by highest number of exchanged trains. In other hand, cluster 2 represents the majority of Cellular and ADSL short connections, when connection is going well : no pacing A and B. Warm-up B are similar to the ones obtained for clusters 1 and 3, showed in Figure B.15(b).

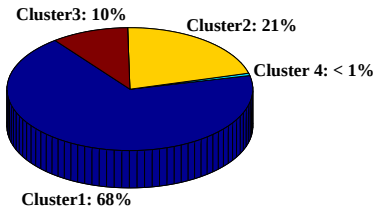
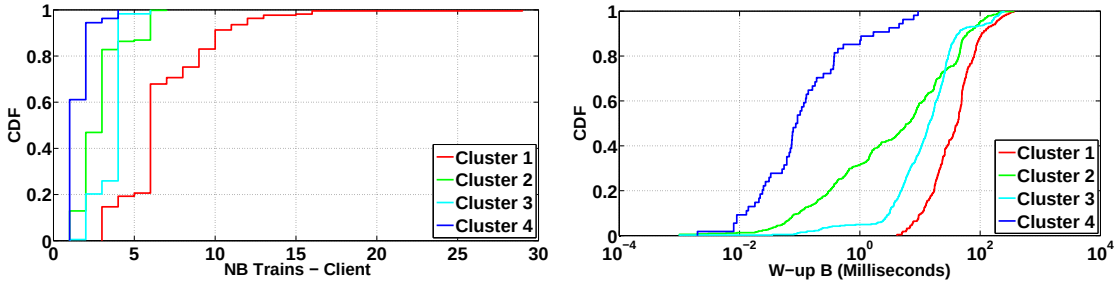


FIGURE B.14 – SMTP : Data Volume per Cluster

Cluster 3 corresponds only to FTTH connections. As we can notice connections in this cluster are penalized only by the response time on the server side, because large FTTH throughput allows users to send mails more faster than ADSL and Cellular users. In fact, Figure B.16(a) shows that RTT are very low for these clusters, which generates low Theoretical times A and B. Identified Warm-up B are similar to ones in clusters 1, 2 and 3 (Figure B.15(b))

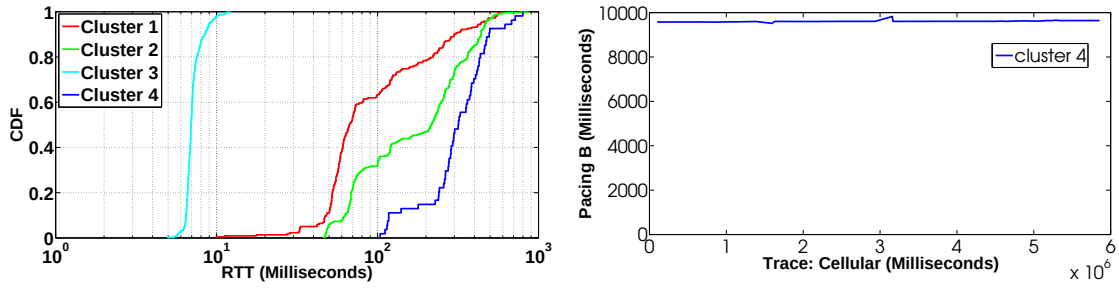


(a) Trains (b) W-up B

FIGURE B.15 – SMTP : Warm-up B and Trains

We can further observe that Cluster 4 is only from Cellular connections with 10 seconds of Pacing B. After the investigation of exchanged data packets in theses connections, we found that phenomenon was due to an anomaly in Orange SMTP servers.

Investigation results shows that problem was not related to one server, but servers with different IP addresses and only for Cellular connections. Figure B.16(b) shows pacing B values for connections in this cluster over the Cellular trace. We observe that high Pacing B values were not limited to a period of the capture.



(a) Trains (b) Pacing B : cluster 4

FIGURE B.16 – SMTP : RTT and Pacing B

We summarize in Figure B.17 the characteristics of SMTP clustering. We observe two categories of clusters corresponding to short and large transfers. Then we show that access technology is the main discriminant factor while since cluster 3 corresponds to FTTH connections with short RTT and cluster 4 correspond to Cellular connections. On the other hand, clusters 1 and 2 present a mix of ADSL and Cellular connections.

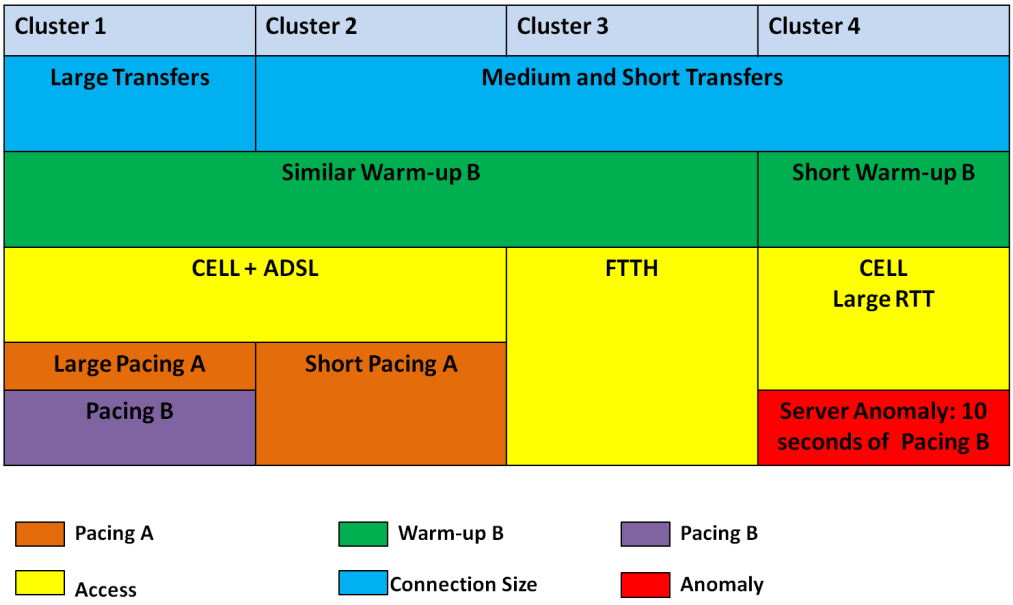
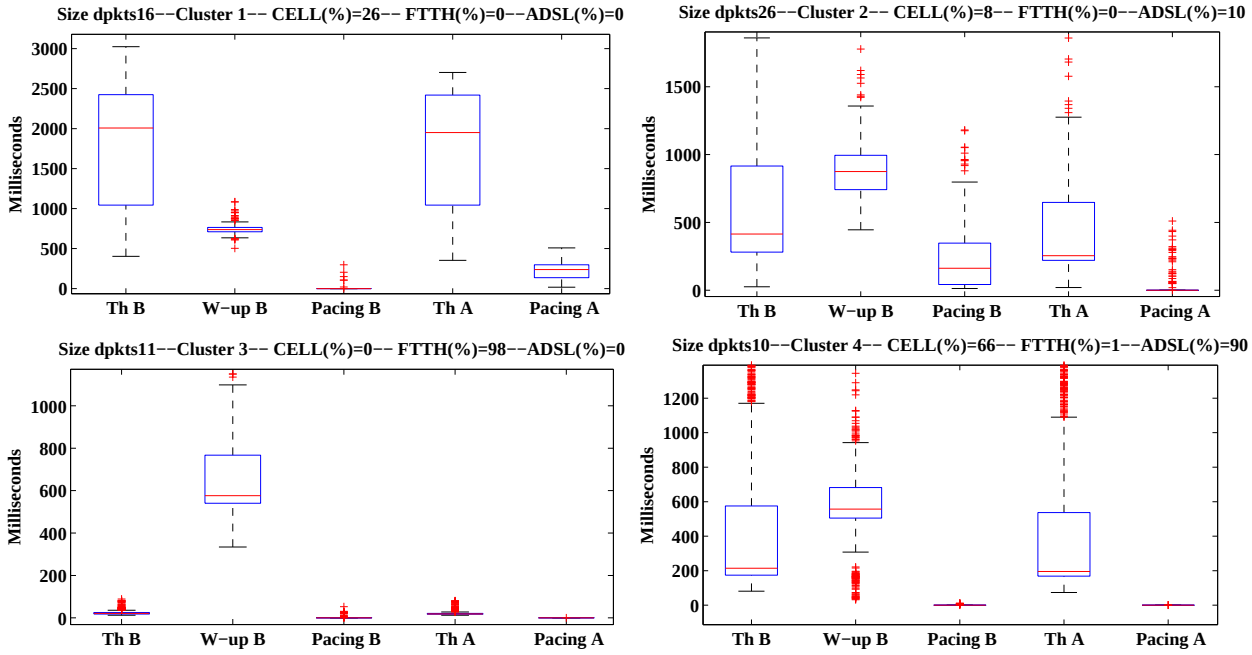


FIGURE B.17 – Overview of SMTP Clusters

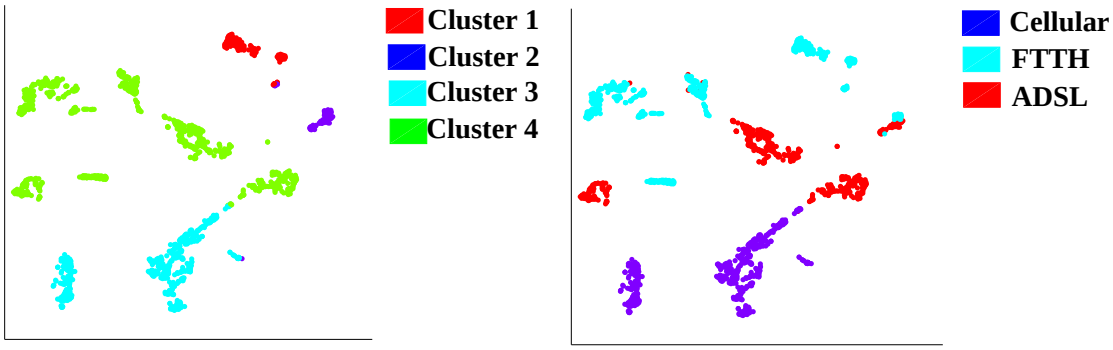
B.2.3 POP3 : Clustering Results

Figure B.18(a) shows boxplots of 4 clusters obtained with Kmeans algorithm. We obtain the same number of clusters, like computed for SMTP Orange traffic. We have also reasonable results with the t-SNE clustering method. In fact, clusters obtained with Kmeans are in good agreement with the projection obtained with t-SNE as indicated in the left plot of Figure B.18(b), with data samples indexed using their cluster ID in Kmeans.

Figure B.2.3 indicates that clusters 2 and 3 correspond to the majority of data with 81% of bytes. While, cluster 4 has more connections and only 15% of exchanged bytes.



(a) K-means



(b) TSNE

FIGURE B.18 – POP Orange Clusters

Overall, clusters were divided in two categories : Cluster 1 and 2 correspond to large connections, while 3 and 4 correspond to short ones. Figure B.20(a) depicts Warm-up B distribution for POP traffic. It shows approximatively the same Warm-up B distributions for identified clusters.

Based on the presented results in Figure B.18(a), we can draw some conclusions for the main clustering parameters. We have seen that cluster 1 and 3 correspond exclusively to Cellular and FTTH connections, while cluster 2 and 4 group ADSL and Cellular ones. This first observation highlights the access impact for clustering results. Cluster 3 was identified by short Theoretical times A/B and null Pacing A and B. It shows that due to high throughput available in FTTH access, users are able to download data from POP3 server more faster than in ADSL and Cellular without Pacing values.

In Cluster 1, we show large Cellular connections, with a mean size of 16 data packets, characterized by a large Pacing A (median value=240 ms). Figure B.20(b) shows that cluster 4 presents

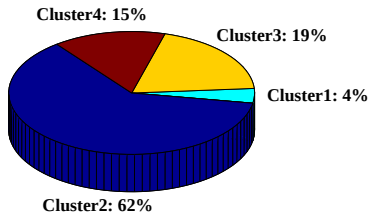
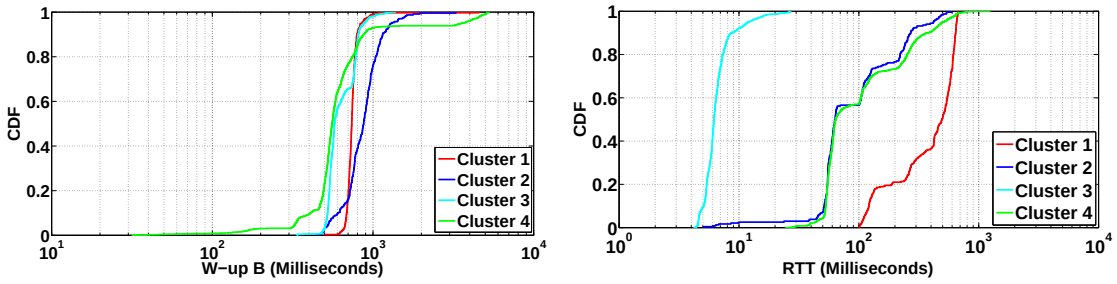


FIGURE B.19 – POP : Data Volume per Cluster

the largest RTT, which in part explains the noticed Pacing A. As we can notice from Figure B.21, more than 98% of connections in this cluster exchange seven trains. We observe in Figure B.18(a) that they represent only 4% of all generated Orange POP data traffic. An explanation for this observation is that the observed traffic corresponds to a specific one like the one used for the authentication step or exchanged POP messages used in order to check mail boxes.

Clusters 2 and 4 with ADSL and Cellular connections, two clusters have similar RTT values as it is shown in Figure B.20(b), but with different connections size. The main conclusion here is that ADSL and Cellular accesses offer approximatively similar performances.



(a) W-up B (b) RTT

FIGURE B.20 – POP : W-up B and RTT

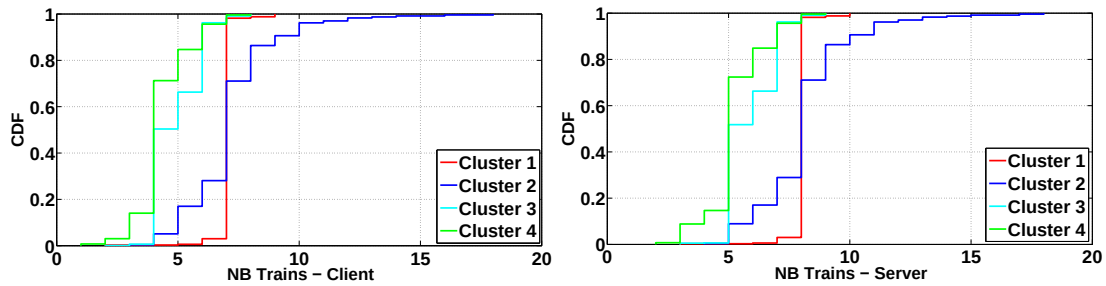


FIGURE B.21 – POP : Client Trains

Although clusters 3 and 4 have the same connection size and Warm-up B, FTTH access seems to be more penalized by data preparation in the server side than the Cellular and ADSL accesses.

We summarize in Figure B.22 the main characteristics of identified clusters. The study of Orange POP traffic shows that RTT is the main parameter when performing clusters, presented in Figure B.20(b). Cluster 3 is distinguished from other clusters by short RTTs because it corresponds

to FTTH connections, while cluster 1 corresponds to Cellular connections with largest RTTs. Then, clusters 2 and 4 with connections from Cellular and ADSL present the same RTT distribution, but different connections size : Large connections in cluster 2 are characterized by the highest number of exchanged trains, contrariwise, the cluster 4 has connections with the smallest ones.

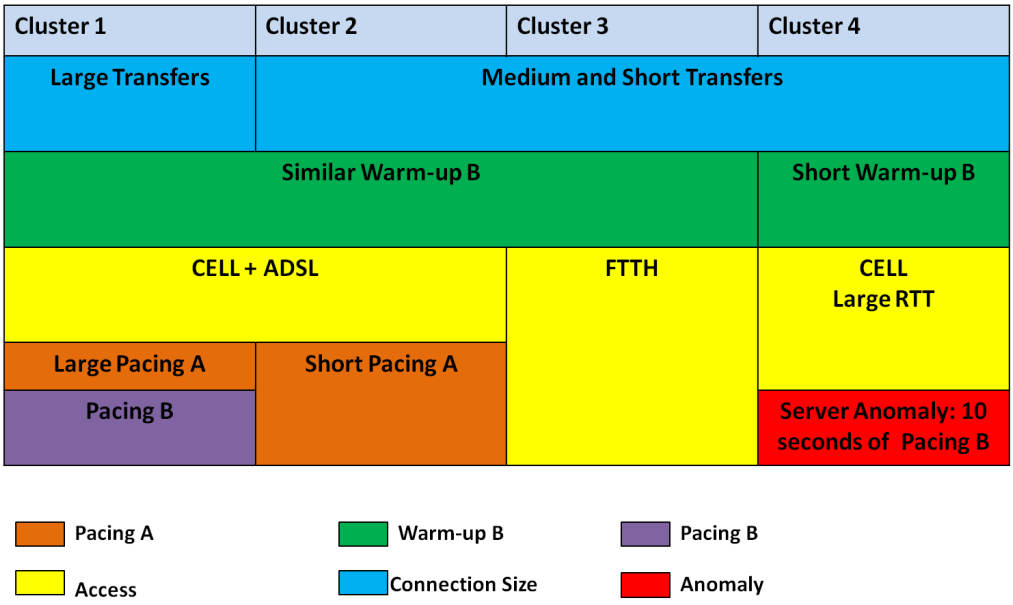


FIGURE B.22 – Overview of POP Clusters

Appendix C

Résumé en Français

C.1 Introduction

De nos jours, nous remarquons que TCP/IP (Transmission Control Protocol / Internet Protocol) est le protocole dominant de transfert des données dans les réseaux locaux et étendus. Les protocoles TCP/IP ont été initialement développés dans le cadre de réseaux de recherche. A la fin des années 1960, l'ARPA (Advanced Research Projects Agency, maintenant appelée DARPA) du département américain de la Défense a mis en place un partenariat avec des universités américaines, la communauté de recherche et certaines entreprise, pour la conception de protocoles et de standards pour le cas de réseaux hétérogènes. Internet et TCP / IP sont si étroitement liés par leur histoire qu'il est difficile de parler de l'un sans parler de l'autre. Ainsi, au fil des années, TCP/IP a continué à évoluer pour répondre aux besoins de l'Internet et également pour le cas de réseaux privés.

Au cours des 15 dernières années, l'intérêt pour la collecte des données, la mesure et l'analyse de trafic ont augmenté d'une façon constante. Des progrés significatifs ont été réalisés sur plusieurs fronts. Ainsi, des aspects importants de l'architecture d'Internet ont été mesurés, pour faciliter la compréhension du fonctionnement de ces réseaux, comme par exemple la mesure de la bande passante disponible, la classification de trafic et la mesure de la capacité disponible. Cependant, il reste encore quelques pièces manquantes dans ce puzzle. En particulier, il y a un besoin pour les fournisseurs de services Internet (FSI) de mesurer la qualité des services offerts à leurs clients finaux. Ceci permettra aux FSIs de diagnostiquer les problèmes de réseau et d'améliorer leurs performances.

Alors que le trafic Internet a été très bien étudié depuis de nombreuses années, les caractéristiques des réseaux d'entreprises restent presque totalement inexplorées. Nous avons observé dans la littérature que pratiquement toutes les études disponibles sur le trafic entreprise se focalisent sur des cas particuliers d'environnement et d'architecture de réseaux. Une des raisons probables pour laquelle le trafic entreprise n'a pas été étudié pendant si longtemps, est qu'il est techniquement difficile à mesurer. Contrairement au trafic Internet, que nous pouvons généralement collecter sur un lien d'accès unique, la collecte du trafic entreprise nécessite un processus de mesure plus complexe.

Dès le début, l'Internet avait pour but de fournir une infrastructure générale sur laquelle une large gamme d'applications pourrait fonctionner. La conception d'Internet prévoit deux sortes d'objectifs en même temps : la capacité à supporter une large variété d'applications est essentielle, mais c'est aussi la capacité de fonctionner sur une gamme de nouvelles technologies émergentes d'accès aux réseaux tels que les réseaux cellulaire, la fibre et l'ADSL classique.

Avec l'émergence de ces nouvelles technologies d'accés à Internet, on constate une émergence

de nouvelles applications et services. Les services qui étaient auparavant conçus pour les lignes ADSL sont maintenant utilisés dans les réseaux avec une latence faible ou élevéé.

Nous développons dans cette thèse une méthodologie globale pour étudier les performances des technologies d'accès hétérogènes et pour connaitre les problèmes de performance perçus par le client, c'est à dire évaluer séparément pour les caractéristiques spécifiques de la technologie d'accès, aux comportementx des serveurs, ou aux comportements des clients. C'est en effet une tâche difficile, étant donné que peu de travaux [2, 3] se sont focalisés sur ce problème et ont développé des méthodes d'analyse, qui permettent aux utilisateurs de déterminer à partir de traces les causes de la limitation de débit. Les travaux de Matti Siekkinen ont constitué un point de départ de notre analyse des performances TCP, ainsi sa méthodologie était dédiée aux connexions TCP qui transportent au moins 130 paquets de données. Comme les petits transferts constituent la majorité des flux TCP, nous avons décidé de concevoir une méthodologie générique de profilage des connexions TCP, indépendamment de leur taille.

Pour s'attaquer au problème de l'analyse de perfomance, nous adoptons une approche "diviser pour régner", où nous nous sommes d'abord concentrés sur les pertes, qui sont sans doute une cause majeure de problèmes de performance pour le protocole TCP. Ensuite, nous analysons les transferts ou les parties de transferts qui ne sont pas affectés par des pertes. Nous utilisons une méthodologie fine et de nous discutons de la façon dont certaines anomalies peuvent être découvertes en appliquant cette technique.

Pour notre travail, nous avons recueilli plusieurs traces de différents milieux : le trafic Internet du réseau d'un ISP européen (cellulaire, FTTH et ADSL), un hotspot sans fil, un laboratoire de recherche et une trace du trafic entreprise. Ces traces ont été recueillies au cours de différentes périodes de temps. L'intérêt de cette diversité est d'éviter que les résultats obtenus soient biaisés par la localisation ou par les aspects temporels. Nous avons l'intention de proposer une approche agnostique et globale d'analyse de performances avec un large champ d'application.

C.2 Description des Traces

Nous avons utilisé, tout au long cette thèse, trois ensembles différents de traces, dont nous présentons un aperçu dans la partie qui suit.

C.2.1 Environnements Hétérogènes

Le tableau C.1 résume les principales caractéristiques des traces, au niveau des paquets, utilisées pour notre travail. Ces traces ont été recueillies dans différents environnements : le réseau DSL à partir d'un FSI européen, un point d'accès sans fil à Portland et notre laboratoire de recherche (Eurecom). Ces traces sont intéressantes en raison de leur diversité en termes de technologies d'accès et également en termes d'applications. Par exemple, les transferts P2P sont interdits sur le réseau Eurécom, alors qu'ils représentent une fraction importante des octets pour la trace DSL. Un point d'accès sans fil devrait différer d'un réseau DSL : dans les réseau DSL les utilisateurs ont tendance à se concentrer davantage soit sur des applications interactives soit à générer d'importants transferts, par exemple, les mises à jour des applications ou des transferts P2P. Comme présenté dans le tableau C.1, ces traces présentent plusieurs différences concernant la date de capture, leur emplacement, la nature du trafic, ainsi que le type d'utilisateurs sélectionnés. Nous détaillons par la suite la définition des connexions TCP bien formées.

	Date de capture	Durée de la capture	Nb de connexions	NB de connexions bien formées	Taille en MB	Nombre de paquets
ADSL	2005-05-31	1 min and 29 s	37790	5873	357.51	743683
Portland Hotspot	2007-09-14	2 h and 20 min	5051	3798	174.13	352569
Research Lab	2008-10-20	1 h and 1 min	32153	26837	1567.42	2867321

TABLE C.1 – Environnements hétérogènes : Description

C.2.2 Des Traces du FSI Orange

Dans cette partie, nous étudions trois traces, au niveau paquet, pour des utilisateurs finaux appartenant à un FSI français, utilisant différentes technologies d'accès : ADSL, cellulaire et FTTH. Les Traces ADSL et FTTH correspondent à l'ensemble du trafic de connexions ADSL et FTTH, tandis que la trace cellulaire est recueillie sur un GGSN de niveau 3, qui est l'interface entre le réseau mobile et l'Internet. Le tableau C.2 résume les principales caractéristiques de chaque trace. Notez que ces mesures ont été effectuées à différentes périodes de la journée afin de comparer la stabilité du trafic et d'obtenir des conclusions indépendantes d'une période de temps ou des comportements des utilisateurs. En conséquence, il est important de noter la grande variabilité et la diversité de nos ensembles de données, accentuée par les différences de comportements des utilisateurs ainsi que les cactéristiques du réseau d'accès.

Par exemple l'accès cellulaire devrait différer de FTTH et ADSL en terme d'utilisation, car les utilisateurs de l'accès cellulaire ont tendance à en faire un usage spécifique et rapide tel que la consultation du courrier électronique ou la navigation Web. On peut s'attendre aussi à de nouveaux changements avec l'introduction des téléphones intelligents (smartphones) et l'utilisation des clefs 3G pour le cas de réseaux cellulaires.

	Cellulaire	FTTH	ADSL
Date	2008-11-22	2008-09-30	2008-02-04
Début de la capture	13 :08 :27	18 :00 :01	14 :45 :02 :03
Durée	01 :39 :01	00 :37 :46	00 :59 :59
NB Connexions	1772683	574295	594169
cnxs bien formées	1236253	353715	381297
Volume UP(GB)	11.2	51.3	4.4
Volume DOWN(GB)	50.6	74.9	16.4

TABLE C.2 – Des traces du FSI Orange : Description

Dans le présent travail, nous nous concentrons sur les applications aux dessus de TCP, ce protocole transporte l'immense majorité des octets dans nos trois traces, et presque 100% pour la technologie cellulaire. Nous limitons notre analyse aux connexions qui correspondent à des transferts a priori compléts, que nous appelons connexions bien formées (qui seront détaillées par la suite). Ces connexions transportent entre 20 et 125 Go de trafic pour le cas de nos traces (voir tableau C.2).

C.2.3 Trafic Entreprise

Notre dernière capture consiste en une seule trace recueillie dans un environnement de réseau d'entreprise, composé d'un ensemble de machines qui peuvent communiquer soit avec des serveurs internes soit avec des machines sur Internet.

La Figure C.1 présente une vue globale de notre réseau. Cette infrastructure se compose d'environ 800 postes de travail équipés d'une variété de systèmes d'exploitation. Le réseau est organisé en plusieurs sous réseaux locaux virtuels (VLAN) : les serveurs, le personnel, DMZ, connectés via un commutateur Cisco. Nous nous focalisons sur les flux TCP, car ils représentent plus de 97% des flux dans chaque trace, et ils transportent plus de 99% des octets.

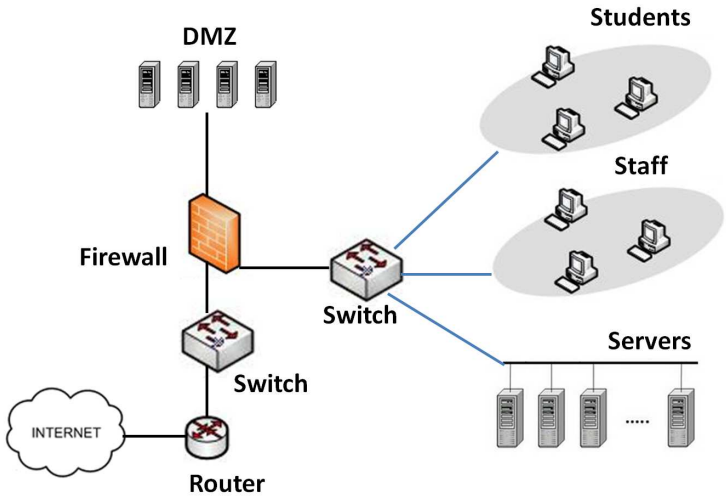


FIGURE C.1 – Architecture de notre réseau d'entreprise

Le tableau C.3 résume les principales caractéristiques de notre trace du réseau d'entreprise. La trace peut être divisée en plusieurs sous-classes de trafic, selon la source et la destination des machines. Comme le montre le tableau C.3, nous remarquons que le trafic client/serveur domine en proportion de connexions bien formées ainsi que pour les volumes de données échangés.

	Serveur/DMZ	Client/Serveur	Serveur/Serveur
Bien formées connections	57348	128237	52333
Volume UP(GB)	8.581	127.061	76.290
Volume DOWN(GB)	6.651	114.054	76.365
Volume UP(data packets)	10,798,530	153,704,391	61,114,981
Volume DOWN(data packets)	9,268,532	145,712,454	61,198,436

TABLE C.3 – Trace Entreprise : Description

C.3 Revisiter les Performances des Transferts TCP

Dans cette partie, nous mettons en évidence l'interaction entre le protocole TCP et l'application. Nous allons d'abord discuter de la définition communément faite des transferts TCP courts,

qui ne peuvent pas compter sur le mécanisme Retransmission Rapide (FR/R) - même avec l'émergence de nouveaux mécanismes comme le 'limited transmit'. Notre principale contribution est de présenter un aperçu de l'impact de l'application, sur les transferts TCP. Nous montrons que si les pertes peuvent avoir un impact négatif sur les transferts TCP courts, l'application affecte de manière significative le temps de transfert de presque toutes les connexions TCP (longs et courts).

En outre, l'application peut aggraver l'impact des pertes en empêchant TCP d'envoyer de gros blocs avec assez de paquets (groupe de paquets de taille suffisante pour que le FR/R puisse s'appliquer). Nous adoptons une approche agnostique d'application : nous ne faisons aucune hypothèse sur la façon dont l'application fonctionne, afin de développer un ensemble de techniques qui délimitent l'impact de l'application à d'autres causes qui expliquent la durée de transfert de données, y compris le transfert de données lui-même et le temps de récupération, le cas échéant.

Nous illustrons nos résultats avec l'ensemble des traces décrites à la section C.2.1, qui incluent la trace ADSL, un point d'accès Wifi et finalement notre laboratoire de recherche.

C.3.1 Connexions Bien Formées

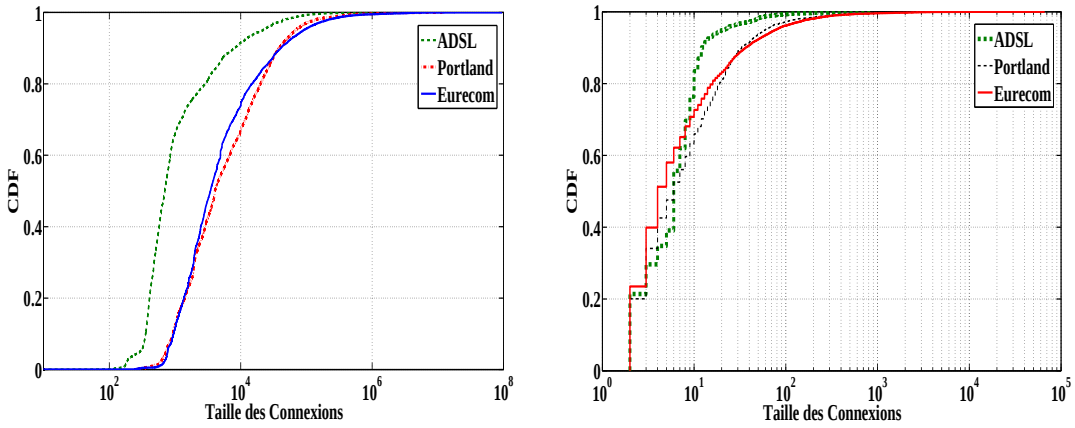
Tout en analysant les performances des transferts TCP, nous nous sommes concentrés sur les connexions qui correspondent à des transferts valables et complets du point de vue TCP. Plus précisément, les connexions TCP bien formées doivent remplir les conditions suivantes : (i) une étape complète d'établissement de la connexion TCP, (ii) au moins un segment de données TCP dans chaque direction, (iii) la connexion doit finir soit avec un drapeau FIN ou RESET.

Lors de l'application de cette heuristique pour nos traces, nous nous retrouvons avec un total de connexions TCP bien formées de plus de 35.000 connexions. La trace DSL est celle qui offre la plus petite fraction des connexions bien formées, plus 377905873 connexions, en raison d'un grand nombre de transferts unidirectionnels (SYN sans réponse). La courte durée de la trace a aussi un impact, car pour un grand nombre de cas, nous n'observons pas le début ou la fin (ou les deux) de la connexion.

Les applications P2P ont tendance à générer de telles connexions anormales (serveur P2P indisponible pour télécharger un contenu) ainsi que des activités malveillantes.

La Figure C.2 représente la distribution cumulative de la taille de connexion pour les connexions bien formées en termes d'octets et de paquets de données pour les trois traces. Nous observons que les traces d'Eurecom et de Portland offrent un profil de connexion similaire qui diffère sensiblement de la trace DSL. Par exemple, 65% des connexions ADSL ont moins de 1 Ko et 25% sont comprises entre 1 Ko et 1 Mo, contrairement à Portland et au trafic Eurecom qui présentent des tailles plus grandes aux mêmes percentiles. Une raison derrière cette observation est de nouveau la petite durée de la trace DSL. Dans cette partie, nous nous concentrons sur les transferts courts. Nous avons aboservé que la trace DSL offre des données différentes des deux autre traces, alors que les traces d'Eurecom et Portland présentent à peu-près la même distribution cumulative des octets.

En se concentrant sur la performance des transferts TCP, le nombre de paquets de données à transférer est un élément clé à considérer, car il impacte la capacité de TCP à récupérer après un évenement de pertes à l'aide du mécanisme Fast Retransmit. Nous pouvons déjà observer dans la Figure C.2 que indépendamment de la trace, une partie importante des connexions (entre 53% et 65%) ont moins de 7 paquets de données.



(a) Taille en Octets (b) Taille en Paquets

FIGURE C.2 – Tailles des Connexions du Milieu Hétérogène

C.3.2 Transferts Courts : Définition

Dans cette section, nous introduisons une première définition d'une connexion TCP courte, ce qui est communément utilisé dans la littérature.

Une connexion TCP courte est une connexion bien formée, incapable d'accomplir un FR/R, après une détection de perte de paquets.

Bien que simple, la définition ci-dessus ne conduit pas à une valeur seuil unique en termes de nombre de paquets de données pour définir un court transfert TCP. En effet, les différentes implémentations TCP ainsi que ses différentes implémentations peuvent affecter cette définition : la fenêtre de congestion initiale, l'utilisation du mécanisme de l'acquittement retardé, le nombre d'acquittements (ACK) dupliqués qui déclenche un FR/R. Par exemple, Windows Vista implémente le Limited Transmit, ce qui signifie que seulement 2 ACK sont suffisants pour déclencher un FR.

Trace	Initiateur			Partie Distante		
	1 pkt	2 pkts	$>$ 2 pkts	1 pkt	2 pkts	$>$ 2 pkts
DSL	99%	1%	0%	80%	18%	2%
Portland	82%	17%	1%	64%	24%	2%
Eurecom	90%	10%	0%	65%	24%	1%

TABLE C.4 – Estimation de la Fenêtre de Congestion Initiale

Nous avons estimé pour les trois traces, le nombre de segments observés dans une durée égale à un RTT, après l'envoi du premier paquet de données, et ce pour chaque la direction - voir le Tableau C.4. La valeur obtenue donne une limite inférieure de la taille de la fenêtre de congestion initiale, puisque dans certains cas l'application au dessus de TCP peut ne peut pas fournir suffisamment de données à TCP pour les envoyer au début du transfert. Cela est particulièrement vrai pour le côté initiateur dans le cas des transferts Web, où la requête GET pourrait tenir dans un seul paquet de données. Globalement, on observe que les valeurs de 1 et 2 (MSS et les valeurs éventuellement plus élevées) semblent être les tailles communes de la fenêtre de congestion initiale. Les tailles de

fenêtres de congestion initiale de plus de 2 MSS (nous avons observé des valeurs allant jusqu'à 12 SMS) pourraient être dues à des optimisations spécifiques des systèmes d'exploitation [71].

Compte tenu de la fenêtre de congestion initiale estimée du Tableau C.4, nous présentons dans le Tableau C.5 les principaux scénarios pour trouver le seuil pour le nombre de paquets de données qui déclenche un FR/R. Une connexion courte est donc, pour chaque scénario, une connexion avec un nombre de paquets strictement inférieur au seuil. Ces scénarios couvrent, étant donné nos connaissances actuelles, tous les cas de figures les plus fréquents.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
cwnd Initial	1	1	2	2
Delayed ACK	no	yes	yes	yes
ACK Duppliqués	3	3	3	2
Taille de connexion Minimum size (paquets de données)	7	9	8	7

TABLE C.5 – Taille de Connexion minimum necessaire pour un FR/R

Sur la base des résultats présentés dans le tableau C.5, nous observons que :
- différents scénarios conduisent à des seuils différents, de 7 à 9 paquets de données ;
- Une connexion de moins de 7 paquets de données ne peut pas appliquer le FR après une perte de paquets, quel que soit le scénario ;
- Lorsque l'on considére un scénario donné et dont la taille de connexion est plus grande que le seuil, on observe que cette connexion est en mesure d'effectuer un FR/R pour seulement un seul paquet dans son dernier bloc de données. La perte de tout autre paquet ne mènera pas à un FR/R. Une connexion n'est donc pas toujours en mesure d'effectuer un FR/R si sa taille est supérieure au seuil.

Basé sur le résultat obtenu à partir de cette section, nous adoptons une première définition des transferts TCP courts, c'est un transfert dont la taille est inférieure à 7 paquets de données. Cette définition, bien que simple, repose sur l'hypothèse implicite que l'application sur le dessus de TCP n'a pas d'impact sur la façon dont TCP envoie des paquets. Comme nous le verrons dans la section C.5, cette hypothèse peut être trop forte, en pratique, puisque même les long transferts TCP peuvent être divisés en petits blocs (dus à l'application au dessus de TCP) qui empêchent le déclenchement du FR/R en cas de pertes.

C.4 Décomposition des Délais de Transfert

Pour comprendre les facteurs qui affectent les performances des transferts TCP, nous nous appuyons sur la décomposition suivante dans la figure C.3 de chaque transfert TCP en 3 phases :

Le délai d'établissement : C'est le temps entre le premier paquet de contrôle et le premier paquet de données. Puisque nous ne considérons que les transferts qui ont une étape d'établissement de connexion complète, le premier paquet est un paquet SYN alors que le dernier est un ACK pur en général. Le temps d'établissement de la connexion est fortement corrélé au RTT de la connexion. Pour les trois traces que nous considérons, nous avons un coefficient de corrélation de 70 % pour la trace DSL, 60 % pour la trace de Portland, et 39 % pour la trace Eurecom.

Le délai de transfert des données : C'est le temps entre le premier et le dernier paquet de données de la connexion. Il comprend aussi les durées de recouvrement des pertes, le cas échéant.

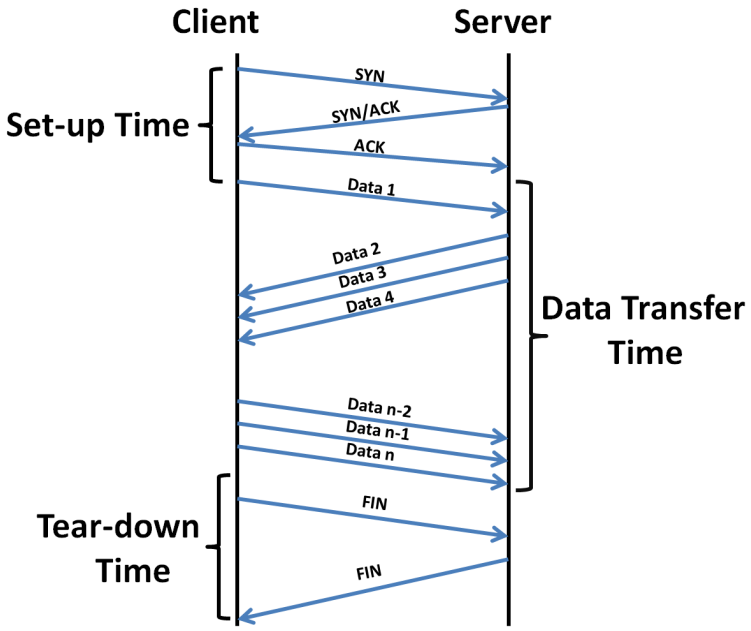


FIGURE C.3 – Décomposition du Temps de Transfert

Le délai de libération : C'est l'intervalle de temps entre le dernier paquet de données et le dernier paquet de contrôle de la connexion. Nous imposons, comme expliqué dans la section C.3.1, qu'au moins un FIN ou un RESET soit observé, mais il peut y avoir de multiples combinaisons de ces drapeaux à la fin du transfert. Contrairement à l'initialisation, la phase de libération de la connexion TCP ne dépend pas seulement du RTT de la connexion, mais aussi de l'application au dessus de TCP. Par exemple, le réglage par défaut d'un serveur Web Apache est de permettre des connexions persistantes, mais avec une durée maximale d'inactivité de 15 secondes, ce qui signifie que si l'utilisateur ne poste pas une nouvelle requête GET après 15 secondes, la connexion est fermée. On trouve dans nos traces une faible corrélation entre le temps de libération de la connexion TCP et son RTT : 40% pour la trace DSL (qui est encore assez élevé), 0,7% pour la trace de Portland, et -2% pour la trace Eurecom.

En utilisant la décomposition ci-dessus, nous analysons ensuite l'impact des pertes (section 2.4.1) et de l'application (section C.5) sur le temps de transfert des données.

C.4.1 Retransmission et Libération des Connexions TCP

Comme expliqué plus haut, le temps de transfert de données comporte éventuellement les délais de retransmission des paquets perdus. Nous estimons le temps passé par le protocole TCP dans la récupération des pertes en mesurant les délais de recouvrements .

Plus précisément, pour un transfert donné, chaque fois que le numéro de séquence dans le flux de paquets de données diminue, on enregistre la durée entre cet évènement et l'observation du premier paquet de données dont le numéro de séquence est plus grand que le plus grand numéro de séquence observée jusqu'ici. Par exemple, nous présentons dans la Figure C.4 un exemple d'une connexion TCP avec la perte de deux paquets de données. En supposant que nous associons un numéro de séquence unique à chaque paquet, si l'on observe la séquence 1,2,3,4,7,6,5,6,8, nous

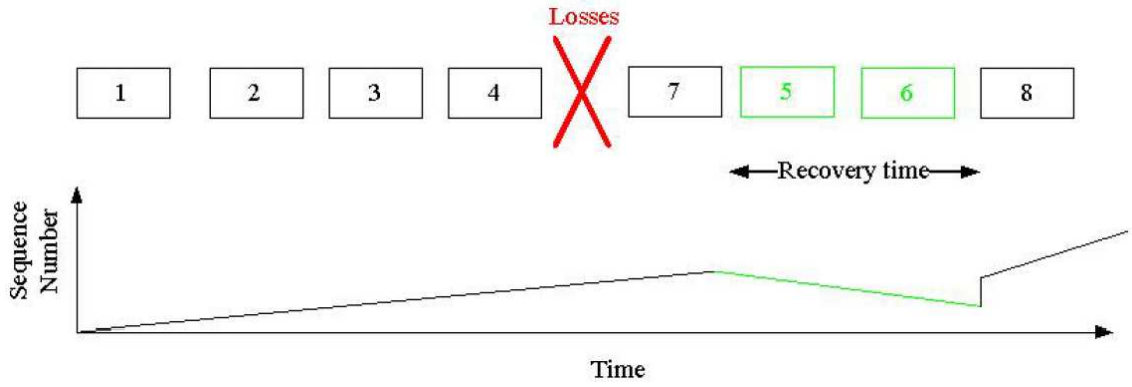


FIGURE C.4 – Récupération suite à la détection de Perte

allons enregistrer la durée entre le paquet 7 et le paquet 8. Cette durée est ajoutée à la période de récupération du transfert. Pour filtrer les phases de ré-ordonnancement dans le réseau, nous éliminons chaque fois les temps de récupération inférieur à un RTT. Rewaskar et al.[72] ont développé des algorithmes pour évaluer si les évènements de perte peuvent être attribués à un temps-mort ou un FR/R. Nous n'étions pas en mesure d'utiliser cette technique car elle nécessite d'effectuer une connaissance de l'OS de l'expéditeur des données. Cependant, dans nos traces, la plupart des pertes sont survenues dans les flux de données émis par la partie distante et non pas pour les clients locaux.

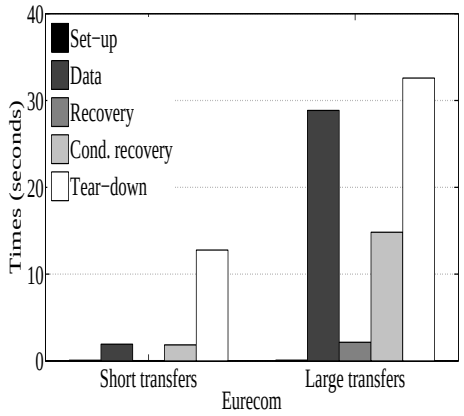
La Figure C.5 présente les résultats de décomposition des transferts TCP pour les petites et les grandes connexions, pour le cas de nos trois traces. Nous observons tout d'abord dans la Figure C.5 que les temps d'établissements sont toujours petits pour toutes les traces et les tailles de transferts ; Les temps de libération peuvent atteindre des valeurs trés élevées, entre 2,5 et 27,5 secondes en moyenne.

La phase de libération, représente souvent la majorité du temps de connexion. Notez cependant, que le temps de libération de la connexion TCP ne devrait avoir aucun impact sur la performance perçue de l'application quand le transfert des données est terminé.

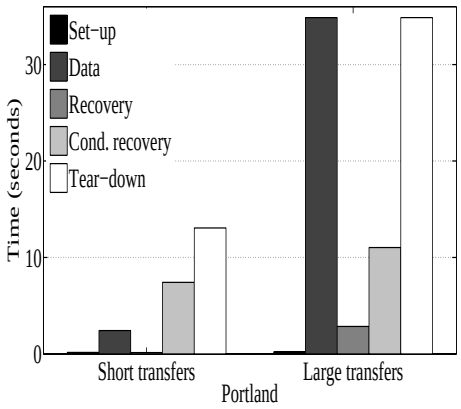
Quant aux pertes, nous présentons deux valeurs distinctes pour le temps de récupération : le temps moyen de récupération conditionnel et le temps moyen de récupération. Ce dernier est calculé sur tous les transferts de la catégorie alors que le premier est calculé uniquement pour les transferts avec au moins un évènement de récupération de perte. Etant donné que seulement une petite fraction de transferts ont des pertes (9,4% pour DSL trace, 13,2% pour Portland et 6,8% pour Eurecom), le temps moyen de récupération conditionnelle est souvent beaucoup plus grand que le temps de transfert moyen. Cet impact est nettement plus marqué pour les petits que pour les grands flux, dans les trois traces, probablement en raison de la prédominance de temps morts pour les transferts de courte durée.

Pourtant, d'un point de vue du serveur qui sert simultanément un grand nombre de clients, comme un serveur Web, des temps longs de fin de connexion, qui peuvent affecter la qualité de service. Un effet secondaire de ces grands temps de fin de connexions est lors de l'estimation du débit des transferts.

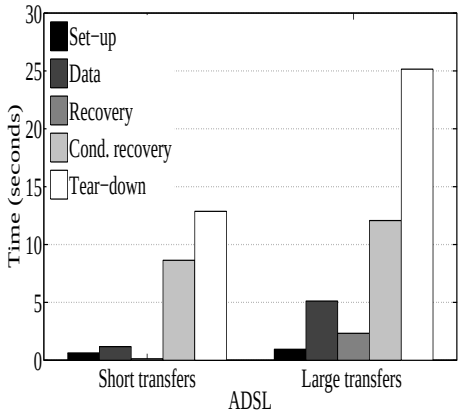
Si on divise le nombre total d'octets de données par la durée totale de la connexion TCP, on peut sous-estimer le débit réel perçu par l'utilisateur et l'application. La figure C.6 présente pour le cas de la trace Eurecom, le débit calculé lorsqu'on considère le temps total de connexion d'un coté et d'un autre côté le débit calculé lorsqu'on considère uniquement le temps d'établissement et le temps de transfert de données. On appelle ce dernier "débit au niveau applicatif" (il est



(a) Eurecom



(b) Portland



(c) ADSL

FIGURE C.5 – Décomposition du Temps Total de Transfert

étiqueté AL), car il représente le taux auquel les données sont envoyées ou reçues au niveau de la perspective de l'application. La figure C.6 montre une différence significative entre les deux métriques pour le cas des courts et longs tranferts.

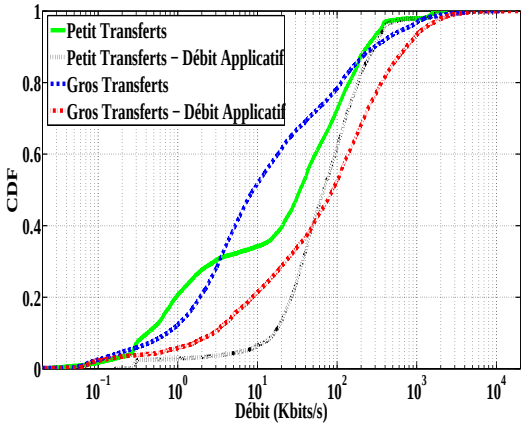


FIGURE C.6 – Comparaison de méthodes d'Estimation de Débits pour la Trace Eurecom

La principale conclusion de l'étude ci-dessus est que les pertes se produisent rarement, mais ont un effet trés préjudiciable. Une deuxième remarque est que le temps de libération de la connexion TCP doit être retiré lors du calcul du débit car il peut conduire à une sous-estimation considérable du débit perçu au niveau de l'application. Par exemple pour le cas de la trace Eurecom, le débit médian des petits (respectivement grands) transferts obtenus lorsqu'on considére le temps de libération, est de 34 kbits/s (resp. 8,7), tandis qu'elle est de 67 kbits/s (respectivement 88) lorsque le temps de libération de connexion est éliminé du temps total de connexion.

C.5 Impact de l'Application

Dans cette section, nous allons évaluer l'impact de l'application sur le temps de transfert d'une connexion TCP. Il y'a plusieurs façons grâce auquelles l'application peut influencer le débit auquel les flux de données sont acheminés dans un réseau. Premièrement, l'utilisateur peut être impliqué dans le transfert, comme dans le cas d'une connexion persistante HTTP, où le téléchargement d'une nouvelle page est déclenché par une requête HTTP GET émis par le client. Deuxièmement, l'application peut limiter la vitesse à laquelle les informations sont envoyées à la couche TCP. C'est typiquement ce que les applications P2P font pour limiter l'encombrement sur la liaison montante de l'utilisateur. Une troisième possibilité est que la génération de données se fasse en ligne. Par exemple, lors de l'interrogation de Google pour un mot clé spécifique, implique plusieurs dizaines de machines. De la discussion ci-dessus, nous observons que l'application peut affecter le transfert des données de différentes façons. Une première évaluation simple de l'impact de l'application sur un transfert TCP, est de calculer la fraction des paquets avec des drapeaux PUSH [73]. Le drapeau PUSH est une façon pour l'application de spécifier qu'elle n'a pas d'octets à transmettre pour le moment à la couche TCP et que celle-ci peut envoyer les données sur le réseau. Nous présentons dans la Figure C.7 le ratio de drapeaux PUSH en fonction de la taille des transferts pour les trois traces. Nous observons que l'impact de l'application diminue avec la taille de transfert jusqu'à un certain seuil dépendant de la taille de connexion. Pour les connexions courtes, la proportion des drapeaux PUSH est extrêmement élevéé, entre 74% et 86%.

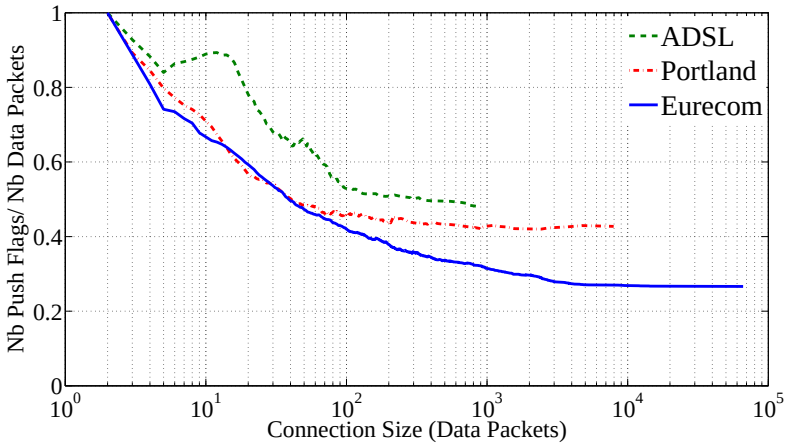


FIGURE C.7 – Ratio conditionnel de drapeaux Push

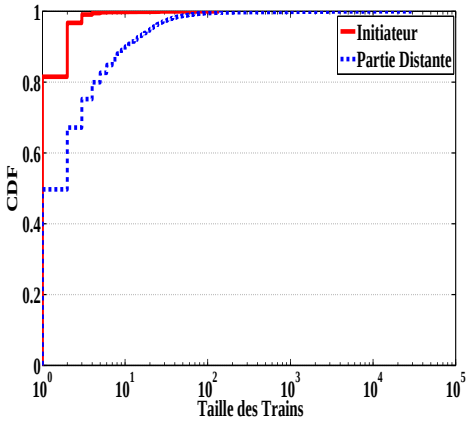
Dans les prochaines sections, nous évaluons plus en détails la façon dont l'application influence le temps de transfert. Nous montrons que l'application a tendance à fragmenter le transfert en petits groupes de paquets TCP qui empêchent de se fonder sur FR / R en cas de pertes.

C.6 Notions de Synchronisme et de Pertes de Paquets

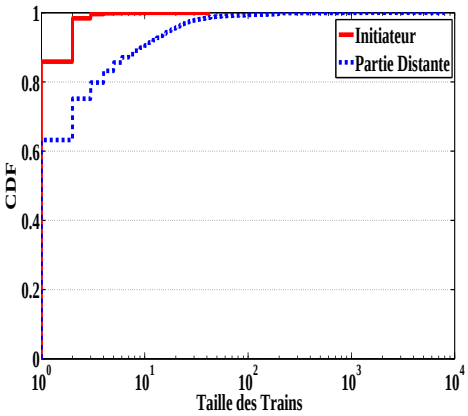
Pour les applications client/serveur, on constate souvent que même si le serveur envoie une grande quantité d'octets/paquets, l'échange réel est fragmenté : le serveur envoie un groupe de quelques paquets (appelés ci-après train de paquets), puis attend que le client poursuive par une autre requête et envoie ensuite sa réponse suivante.

Si un tel comportement est prédominant dans les transferts TCP, il peut avoir un impact néfaste sur les performances lorsque la taille des trains est trop petite, car il peut empêcher d'accomplir le FR/R en cas de pertes de paquets.

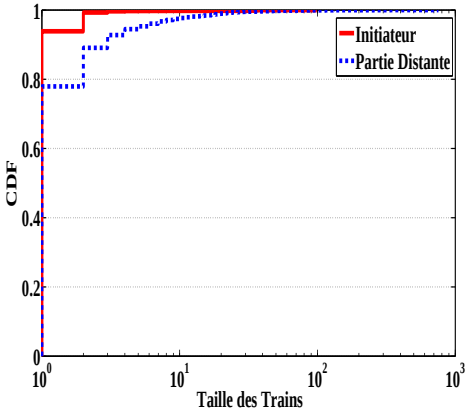
Quand on observe passivement une connexion, nous voyons des données circulant alternativement dans les deux directions ; chaque direction envoie à son tour un train de paquets. Ce n'est pas nécessairement dangereux si les deux parties ne sont pas synchrones, c'est à dire si une partie n'a pas besoin de recevoir des paquets de l'autre partie avant d'envoyer son prochain train de paquets. Toutefois, nous avons observé que les deux parties sont apparemment pour la plupart du temps synchronisées, c'est à dire qu'elles n'envoient pas simultanément des trains de paquets.



(a) Eurecom



(b) Portland



(c) ADSL

FIGURE C.8 – Distribution cumulative de la taille des trains de données

La question que nous soulevons est donc la suivante : Est-ce que les deux parties impliquées dans un transfert sont synchronisées ou non ? Prouver cette synchronisation nécessite une connaissance a priori de la sémantique de l'application. On peut cependant prouver que l'hypothèse de

synchronisation ne peut pas être rejetée comme suit : pour un transfert de donnée, chaque fois que nous observons une transition d'un côté qui envoie des paquets, par exemple A, à l'autre côté, disons B, si le premier paquet de B valide la réception du dernier paquet de A. Si ce n'est pas le cas, alors il n'y a pas de synchronisation, sinon, le synchronisation ne peut pas être rejeté. En appliquant cette méthodologie pour les trois traces, nous avons obtenu que pour chaque trace, la fraction de connexions pour lesquelles le synchronisme ne pouvait pas être rejeté est extrêmement élevée : 88,6% pour la trace ADSL, 94,4% pour la trace de Portland et de 95,3% pour la trace d'Eurecom.

Pour les connexions pour lesquelles la synchronisation ne pouvait pas être rejetée, nous avons examiné la distribution de la taille des trains de paquets. Nous avons distingué entre l'initiateur de la connexion et la partie distante. Comme nous attendons pour ce dernier, qui est aussi dépendant du type de la connexion, dans une grande partie des cas il doit correspondre à une sorte de serveur qui envoie habituellement une plus grande quantité de paquets que l'initiateur de la connexion qui fait juste des requêtes. Comme illustré par la Figure C.8 :

- Les tailles de trains envoyées par l'hôte distant sont plus grandes que ceux envoyés par l'initiateur, ce qui concorde avec notre hypothèse (la partie distante correspond à un serveur) ;
- Plus de 97 % des trains de l'initiateur ont moins de 3 paquets de données, ce qui laisse TCP incapable de déclencher une retransmission rapide, même si Limited Transmit est utilisé ;
- Plus de 75 % des trains de l'hôte distant ont moins de 3 paquets de données, ce qui laisse à nouveau TCP incapable de déclencher la retransmission rapide, même si le Limited Retransmit est utilisé.

C.7 Approche Classique pour la Comparaison de Performance

L'étude du comportement de TCP, en particulier ses performances en termes de délais, des pertes et de débit, a été étudié depuis son émergence dans des environnements spécifiques et différents type d'utilisateurs.

Toutefois, la comparaison et la compréhension des paramètres clés qui influencent les performances perçues de différentes technologies d'accès telles que le cellulaire, le FTTH et l'ADSL deviennent difficiles quand il est en interaction avec la couche application au dessus de TCP.

Ci-après, nous commençons par présenter une approche classique pour comparer les performances des différentes technologies d'accès, afin de conclure si les clients profitent pleinement de leur accès Internet offert par leur FSI.

Ensuite, nous proposons une nouvelle méthode d'analyse qui permettera de révèler l'impact de certains facteurs spécifiques, comme l'application au dessus de TCP, son interaction avec l'utilisateur pour faciliter la comparaison des performances de différentes technologies d'accès.

La méthode d'analyse que nous utilisons consiste en deux étapes. Dans la première étape, le temps de transfert de chaque connexion TCP est décomposé en plusieurs détails que nous pouvons attribuer à des causes différentes, par exemple, l'application ou le chemin de bout en bout. Dans une deuxième étape, nous classons les connexions pour découvrir les grands types de connexions présentées dans nos traces.

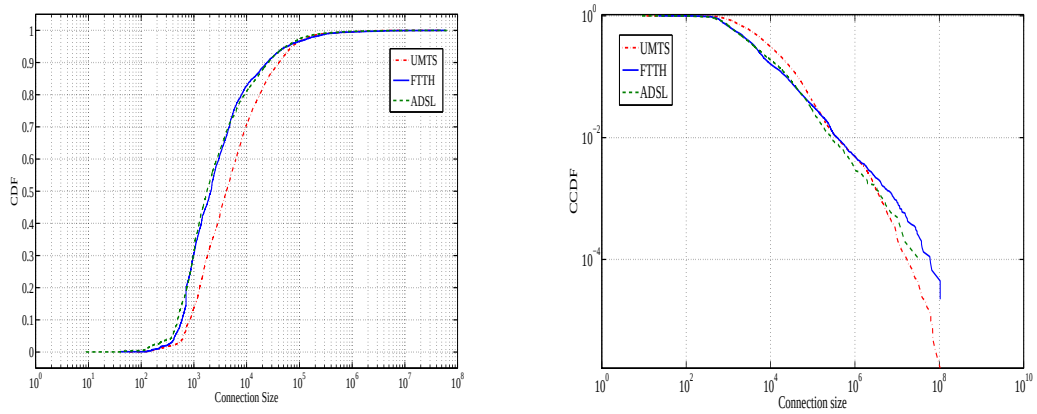
C.7.1 Principaux Suspects

C.7.1.1 Volume de Données

La Figure C.9 présente la CDF (Fonction de répartition) et la CDF complémentaire (CCDF) de la taille des connexions en termes d'octets, pour les traces cellulaires, ADSL et FTTH. Seules

les connexions bien formées sont prises en considération. Nous observons que les traces FTTH et ADSL offrent des profils similaires qui diffèrent sensiblement de l'accès radio. Par exemple 30 % de traces ADSL et FTTH sont inférieurs à 1 koctets et 55 % ont entre 1koctet et 10 koctets, contrairement à la trace cellulaire qui offre des valeurs plus grandes aux même percentiles.

L'étude de la CCDF, montre que la probabilité d'obtenir des transferts avec 1 Megaoctets est trés faible (moins de 0,01). Ces résultats révèlent aussi que, tandis que la majorité des clients cellulaires ne font pas du P2P (limitation au niveau des téléphones), ils sont capables de générer des connexions larges comme pour les accès filaires (FTTH et ADSL).



(a) CDF (b) CCDF

FIGURE C.9 – Taille des Connexions(Octets)

En fait, plusieurs explications peuvent être trouvées pour ces observations. Par exemple, l'utilisation de connexions HTTP persistantes (plus de 84% des connexions cellulaires ciblent des ports HTTP (s)). En outre, l'utilisation de nouvelles applications ou de nouveaux services avec l'émergence des nouveaux mobiles, tels que le téléchargement d'applications de 'Apple Store' ou 'Android Market' et l'augmentation des applications de streaming (Youtube, etc), expliquent les valeurs des grandes tailles de connexions pour l'accès cellulaire par rapport à FTTH et ADSL.

La principale conclusion de ce paragraphe est que, de nos jours, les utilisateurs du réseau cellulaire ont tendance à utiliser leurs mobiles pour un usage différent du simple appel téléphonique ou du SMS à envoyer, en concordance avec les améliorations de l'affichage et la capacités des smartphones. Cela signifie que l'accès cellulaire n'est pas limité à un usage pour de courtes périodes ou à une utilisation nomade, mais pour un usage similaire à celui d'un accès fixe.

C.7.1.2 La Latence de l'accès

Nous avons observé que les deux méthodes d'estimation du RTT avec SYN/SYN-ACK ou DATA-ACK conduisent pratiquement à une même estimation du temps d'aller-retour pour les traces l'ADSL et FTTH, tandis que nous observons des différences pour l'accès cellulaire à cause du 'Performance Enhancing Proxy' (PEP) et d'un APN (Access Point Name).

Nous allons donc nous fonder sur la méthode de DATA-ACK pour estimer la latence sur les traces considérées. La Figure C.10 représente les estimations de RTT pour les trois traces. Elle met clairement en évidence l'impact de la technologie d'accès sur la latence de chaque accès. L'accès FTTH offre un RTT faible en général - à moins de 110 ms pour les plus de 60 % des

connexions. Cette conclusion est en accord avec les caractéristiques généralement annoncés pour la technologie d'accès FTTH. D'un autre côté, la latence sur la l'accès cellulaire est notamment plus longue que pour l'ADSL ou FTTH.

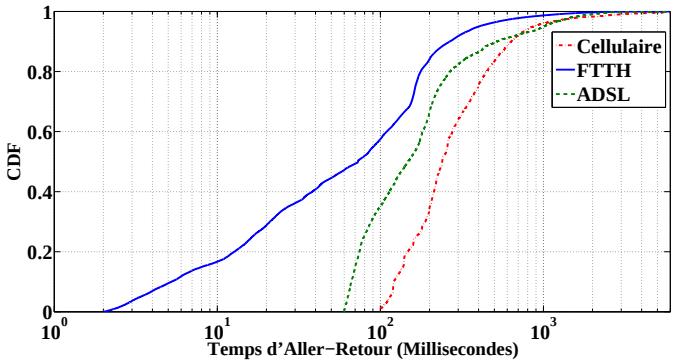


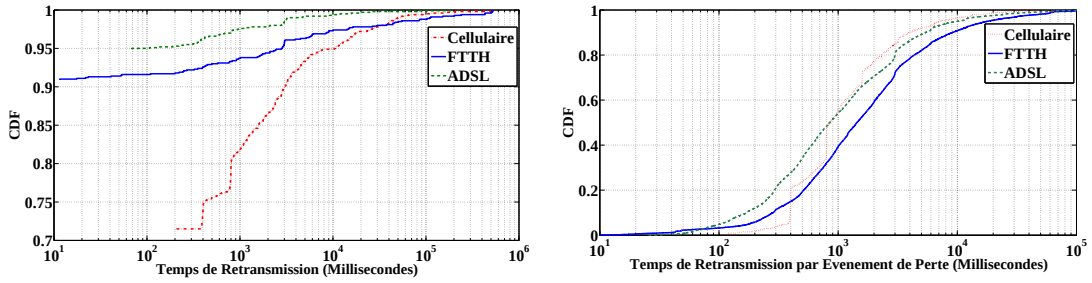
FIGURE C.10 – Estimation du Temps d'Aller-Retour(RTT)

C.7.1.3 Temps de Retransmissions

Nous avons développé un algorithme pour détecter les paquets de données retransmis, qui se produisent entre le point de capture et le serveur ou entre le point de capture et le client.

Cet algorithme [1] est similaire à celui développé dans [75].

Si jamais la perte a eu lieu après le point d'observation, nous pouvons observer le paquet initial et sa retransmission. Dans ce cas, le délai de retransmission est tout simplement la durée entre ces deux instants [2]. Lorsque le paquet est perdu avant la sonde, nous en déduisons l'instant auquel il aurait dû être observé, sur la base des numéros de séquence des paquets. Notez que les calculs de toutes ces durées sont effectués, côté expéditeur, nous nous basons sur les séries chronologiques (nous décalons nos calculs selon nos estimations du RTT). Pour nos traces, il est plus facile de détecter des pertes où le paquet de données est vu à deux reprises. Mais, lorsque la perte s'est passé entre le point de capture et le serveur distant, nous sommes seulement en mesure de détecter un paquet dé-séquencé.



(a) Per Connection (b) Per Loss Event

FIGURE C.11 – Temps de Retransmission des Paquets de données

1. L'algorithme de détection de perte utilisée est disponible sur http ://intrabase.eurecom.fr/tmp/papers.html. Les lecteurs sont invités à vérifier l'exactitude de notre algorithme pour détecter les pertes

2. Ces instants sont calculés du point de due émetteur en décalant les séries chronologiques selon nos estimations RTT.

Pour le calcul des temps de retransmissons, nous ne distinguons pas entre les séquences de paquets hors-séquencé et les paquets de données sont observées plus d'une fois. Nous utiliserons pour ces deux cas de figures le terme "retransmission". Nous essayons de séparer les retransmissions réelles des fausses retransmissions en éliminant les délais plus petits que le RTT de la connexion. Une fois que les pertes sont identifiés avec (i) la retransmission de paquets ou (ii) les paquets hors-séquencés, nous calculons le temps de retransmission total pour chaque connexion TCP.

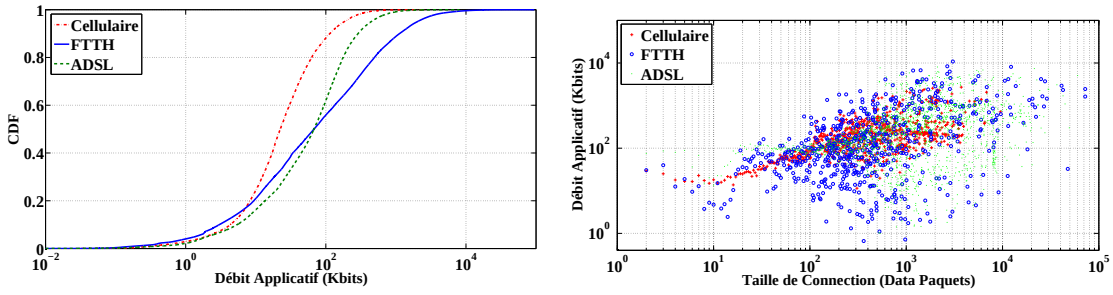
La Figure C.11 représente la distribution cumulative des délais de retransmission par connexion, pour les accès considérés. La principale observation est que la proportion des connexions observant des pertes est plus élevé dasn la trace cellulaire avec plus de 28,6 % et seulement moins de 9 % pour ADSL et FTTH. Ceci démontre que la proportion des connexions observant des pertes diminue lorsque la capacité augmente. Une explication pour cette observation peut résider dans la différence de fiabilité entre les accès cellulaire et filaires.

Dans les précédents travaux, nous avons remarqué que les auteurs ont présenté plusieurs facteurs qui influencent les taux de pertes pour l'accès cellulaire. En fait, dans [90] les auteurs recommandent d'utiliser un algorithme de détection des pertes, qui utilise des traces de chaque connexion (cet algorithme n'est pas adapté à notre cas, parce que nos mesures ont été recueillies au un niveau d'GGSN) pour éviter de fausses retransmissions de TCP. Aussi, les auteurs dans [56] ont montré que le taux de retransmission dans les réseaux cellulaire est plus élevé pour les trafics Google que les autres, en raison de courts délais d'attente mis en oeuvre dans les serveurs de Google.

C.7.2 Comment Comparer les Performances ?

Notre but ici est de montrer que la technologie d'accès influce le débit, mais que ce n'est pas le seul facteur. La congestion, les détails de la couche transport ou de l'application (par exemple des limiteurs de débit, dans les applications P2P) peuvent également influencer sur le débit observé.

Nous fondons notre estimation de débit sur la définition présentée dans la section C.4.1 où le débit correspond à la quantité d'octets transférés à la couche TCP, divisé par la durée totale entre le premier paquet (premier SYN) et le dernier paquet du transfert. Formellement, c'est ce que nous appelons le débit applicatif.



(a) CDF (b) Median

FIGURE C.12 – Débit au Niveau Applicatif

Dans la Figure C.12(a), nous présentons la CDF de débit applicatif (AL) pour nos traces. Une première observation frappante est que les accès FTTH et ADSL offrent des débits nettement plus élevés que dans l'accès cellulaire. Comme nous l'avons présenté précédemment dans la section C.7.1.2, nous pouvons confirmer que cette observation est une conséquence des différences de RTT disponibles pour chaque accès utilisé. D'autre part, on peut remarquer que les débits applicatifs

de la trace ADSL et pour FTTH sont souvent similaires (jusqu'au 50ème percentile), en contraste avec que les utilisateurs finaux peuvent s'attendre. Une première explication de ce fait, est la distribution de RTT pour l'ADSL et la FTTH.

Afin de mieux comprendre ces courbes de débits pour les connexions courtes et longues, nous avons tracé dans la Figure C.12(b) les valeurs médianes du débit applicatif relatif à chaque taille de connexion. Ceci montre des valeurs plus élevées de l'AL obtenues avec des connexions FTTH. Mais en revanche, elle confirme les résultats observés dans la Figure C.12(a) : les débits pour le accès FTTH, ADSL et cellulaires ne sont pas aussi différents que l'on aurait pu s'y attendre, lorsque nous nous concentrons uniquement sur les RTTs, la perte et la taille de connexion.

Pour comparer les performances d'Internet pour différentes technologies d'accès, nous avons commencé avec une approche classique basée sur l'étude des deux principaux facteurs qui influencent le débit des transferts TCP (voir la formule débit TCP [89]), le taux de perte et l'RTT. Ceci suggère que la performance sur FTTH devrait sensiblement surpasser celle sur ADSL, qui devrait à son tour surpasser celle du cellulaire. Mais, il s'avère que la réalité est plus complexe comme on peut le voir sur la Figure C.12(a). En effet, tandis que la technologie cellulaire offre débits applicatif sensiblement plus petits, en ligne avec les facteurs : RTT et perte, FTTH et ADSL ont des performances beaucoup plus proches que ce que le RTT et les pertes le suggèrent.

Dans ce qui suit, nous présentons une nouvelle méthode pour découvrir l'impact de l'application et pour mieux expliquer les différences ou l'absence de différences entre les technologies d'accès. Par application, nous entendons la manière dont les applications fonctionnent, et aussi la façon dont l'utilisateur interagit avec l'application. En plus du comportement des utilisateurs, qui est fonction de la technologie d'accès. Par exemple, les téléchargements de fichiers volumineux peuvent être rares sur la technologie cellulaire, contrairement aux technologies filaires.

C.8 Méthodologie Proposée : Etude de l'Interaction entre l'Application, le Comportement et l'Utilisation

C.8.1 Décomposition du Temps de Transfert de Données

Dans ce paragraphe, nous introduisons une méthodologie qui complète ce qui a été introduit dans la section C.4. L'objectif ici est de révèler l'impact de chaque couche qui contribue aux délais de transferts de données, à savoir l'application au-dessus de TCP, le transport, et le chemin de bout en bout entre le client et le serveur.

Nous effectuons par la suite une décomposition de la durée de la phase de transfert de données d'une connexion TCP, que nous appelons *temps de effective de transfert des données*, c'est à dire, nous excluons le temps d'établissement et de libération des connexions.

Le point de départ de notre étude est que la grande majorité des transferts se composent de dialogues entre les deux parties d'une connexion à tour de rôle. Cela signifie que les instances d'application parlent rarement simultanément sur la même connexion TCP [91]. Nous appelons les phrases de ces dialogues des *trains*.

Par exemple, comme expliqué dans la section C.6 on observe que même si le serveur envoie une grande quantité d'octets/paquets, l'échange de données réel est fragmenté : le serveur envoie quelques paquets de données (un train), attend ensuite que le client formule une autre demande, puis envoie sa réponse suivante, à savoir la prochaine série de paquets (un autre train).

Nous appelons A et B les deux parties impliquées dans le transfert (A est l'initiateur du transfert) et on décompose le transfert des données en trois composantes : le délai de préparation, le délai théorique et le délai résiduel. La Figure 5.1 illustre cette rupture dans le cas d'une recherche sur Google, où A est un client et B est un serveur de Google.

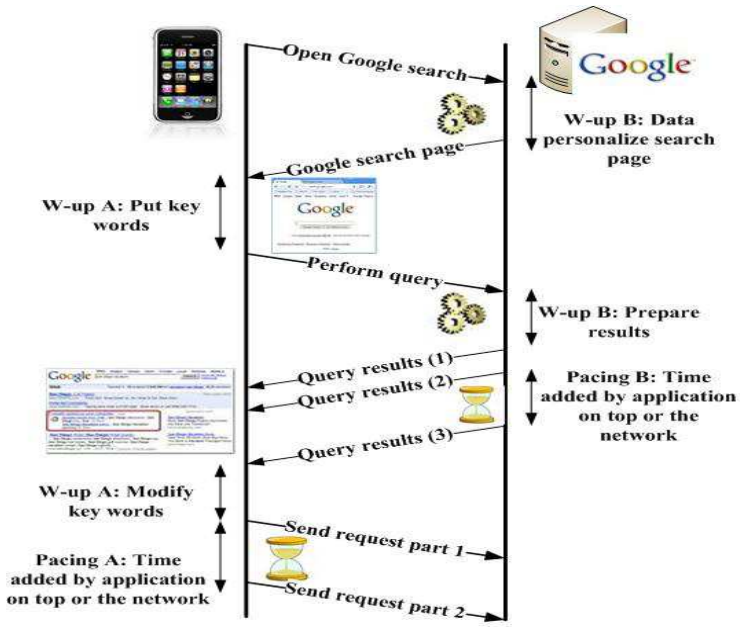


FIGURE C.13 – Décomposition du Temps Effectif de Transfert de données

Le délai de préparation (Warm-up) correspond au temps pris par A ou B avant de répondre à l'autre partie. Il comprend des durées telles que le temps pris par l'utilisateur pour réfléchir ou pour préparer les données côté serveur. Pour notre cas d'utilisation, un warm-up de A correspond au temps passé par le client pour taper une requête et á naviguer à travers les résultats avant d'émettre la requête suivante (le cas échéant) ou en cliquant sur un lien, alors que warm-up B correspond au temps passé par le serveur de Google pour préparer la réponse appropriée à la demande.

Le temps théorique est la durée idéale qu'un transfert TCP mettrait pour transférer un nombre de paquets donné de A vers B (ou de B vers A) égal au nombre total de paquets échangés pendant le transfert complet. Le temps théorique peut être vu comme le temps de transfert total de cette connexion TCP théorique. Pour ce transfert théorique, nous supposons en outre que la capacité du lien n'est pas limité.

C.8.2 Présentation des Résultats

Lors de notre analyse, nous avons eu recours à des techniques de clustering pour obtenir une image globale de la relation entre le service, la technologie d'accès et l'usage.

Après avoir décomposé le temps effectif de transfert, chaque connexion bien formée est représentée par un point dans un espace de 6 dimensions (le temps résiduel, le temps théorique et le temps de préparation des données au niveau du client et du serveur). Pour comprendre ces données, nous utilisons une techniques de regroupement de connexions pour assembler les connexions avec des caractéristiques similaires.

Nous utilisons une approche de clustering non supervisé, à savoir l'algorithme Kmeans. Une question clé lors de l'utilisation Kmeans est le choix des centroïdes initiaux et le nombre de clusters ciblés. Pour évaluer le nombre de clusters, nous utilisons une approche de tests et d'essais. Nous avons commencé avec un nombre important de pôles d'abord, puis nous déduisant ce nombre.

Concernant le choix de centroïdes, nous effectuons une centaine d'essais pour finalement

prendre le meilleur résultat (c'est à dire, celui qui minimise la somme sur tous les clusters des distances entre chaque point et son centre de gravité). Notez que nous utilisons l'implémentation de Kmeans dans Matlab [93].

Pour évaluer le nombre de clusters utilisés, nous nous appuyons sur une technique de réduction de dimensionnalité visuelle, t-Distributed Stochastic Neighbour Embedding (t-SNE) [94].

Les valeurs de chaque dimension ont tendance à être très dépendantes de la taille des connexions. Par exemple, le warm-up est une valeur qui représente la somme de toutes les périodes de préparation sur toute la durée du transfert. Le temps théorique et le temps résiduel dépendent du nombre total de paquets à envoyer. Ensuite, il est important lors de la présentation des résultats de garder un oeil sur la taille des connexions, car il est plus probable que les grandes connexions aient le plus grand temps de préparation des données et le plus grand temps résiduel.

Enfin, pour présenter les résultats, nous utilisons les boîtes à moustaches [3] pour obtenir des représentations compactes des valeurs correspondant à chaque dimension.

Au-dessus de chaque cluster nous avons mis la taille médiane des connexions dans chaque groupe, l'identifiant de chaque accès (ID) du cluster et pour chaque trace le pourcentage de connexions. Ce pourcentage est calculé comme le nombre de connexions dans un cluster sur le nombre total de connexions pour une trace, pour chaque technologie d'accès. Il est important de noter que lors de l'exécution de ce clustering, nous utilisons le même nombre de connexions de chaque trace.

Pour chaque cas de clustering, nous utilisons le même nombre d'échantillons par technologie d'accès (prendre le minimum du nombre de connexions)pour empêcher toute partialité dans le regroupement. Notez que les connexions ont été choisies au hasard parmi celles dans chaque trace.

C.8.3 Validation par des Traces Réelles

Sur des simulations(pour plus de détails, voir Chapitre 6), nous avons observé que les valeurs absolues de Warm-up B ne doivent pas être corrélées ni avec l'utilisateur, ni avec le lien. Ceci est en accord avec ce qui devrait être observé pour le trafic réel : si nous supposons une mise en oeuvre homogène d'un service et des conditions de charge similaires sur le côté serveur, le warm-up sur le côté serveur doit avoir une distribution similaire sur différents accès.

Nous présentons les mesures obtenues par l'étude des traces recueillies par Orange pour différents environnements hétérogènes : ADSL, cellulaire et FTTH. Nous nous concentrons sur l'étude du trafic POP3 pour les clients Orange et les serveurs de messagerie d'Orange.

Nous rapportons dans la Figure C.14, la distribution de chaque temps de préparation des données côté du serveur (le temps de préparer la réponse pour le client) pour chaque technologie d'accès. Ceci montre que, malgré la diversité de la technologie d'accès, en utilisant notre méthodologie de décomposition du temps de transfert, nous observons des distributions similaires de préparation des données pour chaque technologie. Notez que les traces sur lesquelles nous nous concentrons n'ont pas été capturées à la même période de temps et donc, les conditions de charge pourraient expliquer les écarts de différences observées dans les CDF.

Afin de mieux comprendre les causes des pics élevés dans les distributions de la Figure C.14, nous avons inspecté la série temporelle des valeurs de warm-up B. La Figure C.15 représente les séries temporelles de warm-up pour chaque technologie d'accès. Une observation clé est la présence de pics dans la Figure C.14. Ces pics ne semblent pas être dépendants du temps (à cause

3. boxplots sont des représentations compactes des distributions : la ligne centrale est la médiane et la partie supérieure et inférieure de la boîte représente respectivement les 25 ème et 75 ème quantiles. Les valeurs extrêmes - loin de la taille de la distribution - sont signalées par des croix

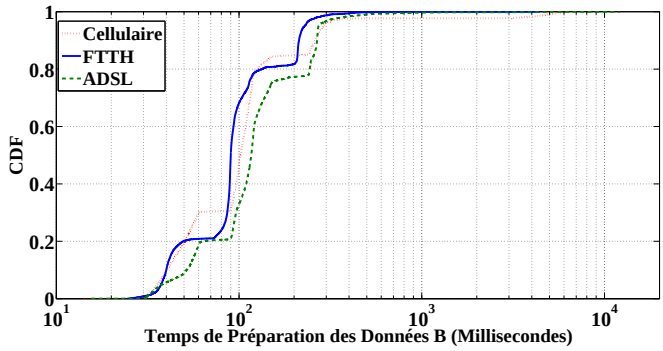
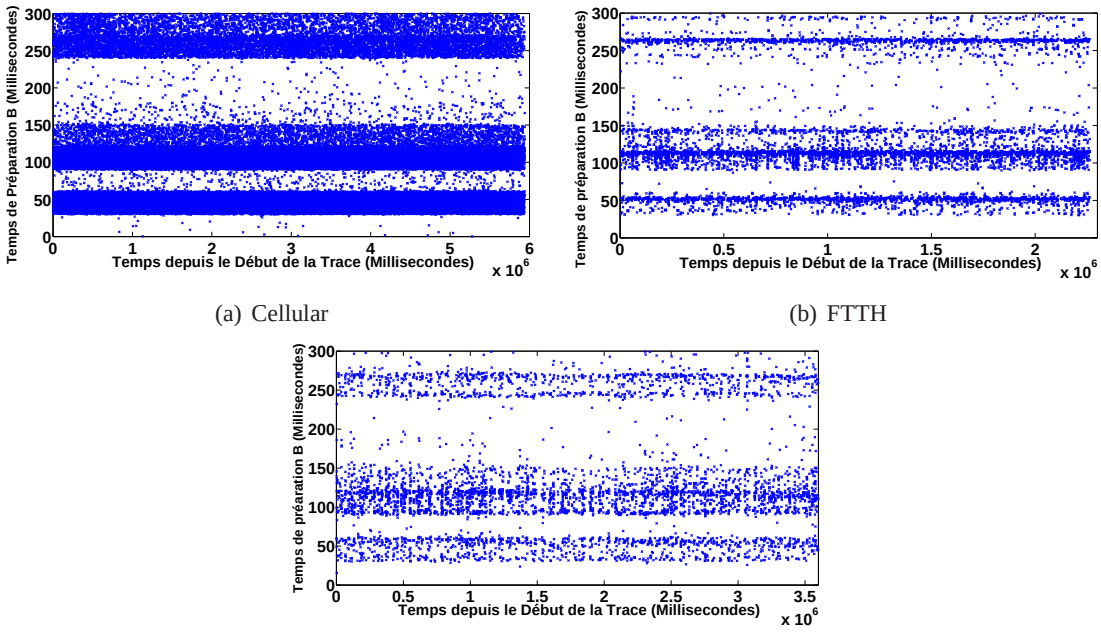


FIGURE C.14 – Temps de Préparation des Donnés : Cas du Service POP de Orange



(a) Cellular (b) FTTH

(c) ADSL

FIGURE C.15 – Series Temporelle des Temps de préparation des donnés : POP3 d'Orange

des variations de charge), mais plutôt de l'application et précisément de la nature de la requête, comme par exemple, l'authentification, vérification de la boîte aux lettres, etc.

Pour résumer, nous avons présenté dans cette partie des résultats différents qui valident, mais indirectement, notre méthodologie de décomposition du temps de transfert de données.

C.9 Application pour le cas de Recherche sur Google

C.9.1 Comparaison des Débits Applicatifs

Notre étude des deux facteurs clés qui influencent le débit des transferts TCP, à savoir le taux de perte et la latence, suggèrent que, puisque les requêtes de recherche Google ont un profil similaire sur les trois technologies d'accèfs, la performance de ce service sur le FTTH devrait sensiblement dépasser celle de l'ADSL, qui devrait à son tour surpasser celle du cellulaire. Il s'avère que la réa-

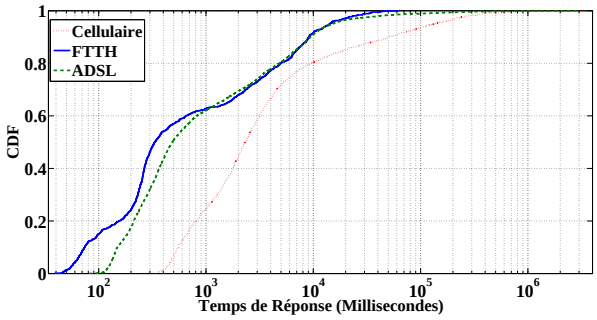


FIGURE C.16 – Google Transfer Time

lité est plus complexe comme on peut le voir sur la Figure C.16 où nous rapportons la distribution des temps de transfert de données.

L'analyse du débit est qualitativement similaire, mais nous préférons nous concentrer sur les temps de transfert de donnés puisque nous regardons un service interactif avec un faible volume de données échangé. En effet, tandis que la technologie cellulaire offre un temps de réponse significativement plus long, en accord avec les facteurs de RTT et de la perte, FTTH et ADSL ont des performances beaucoup plus proches que ne le suggère l'étude du RTT et des pertes.

Dans le prochain paragraphe, nous appliquons une approche plus fine pour la décomposition des temps de transferts, pour le cas du trafic de recherche Google. Le but ici est de découvrir l'impact des facteurs spécifiques comme l'application et l'interaction avec l'utilisateur, et d'informer ainsi la comparaison des technologies d'accès, pour le trafic de recherche Google.

C.9.2 Résultats

La méthode d'analyse que nous utilisons consiste en deux étapes. Dans la première étape, le temps de transfert de chaque connexion TCP est décomposé en plusieurs délais que nous pouvons attribuer à des causes différentes, par exemple, l'application ou le chemin de bout en bout.

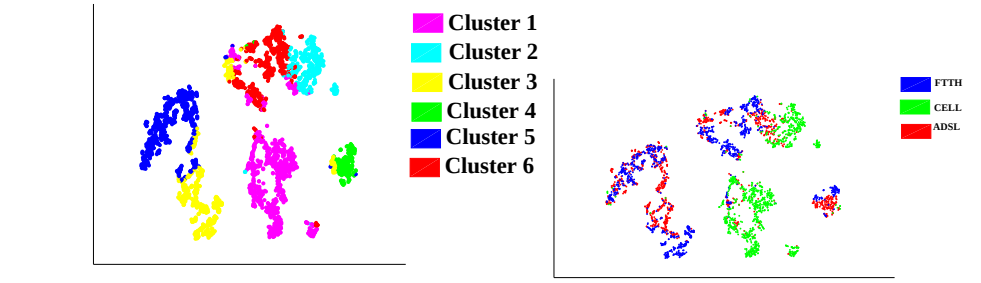
A la fin de l'étape 1, chaque connexion bien formée associée à une recherche Google, elle est transformée en un point dans un espace de 6 dimensions (temps de préparation des données, le temps théorique et le temps résiduel pour chacun du client et du serveur).

Pour comprendre ces résultats, nous utilisons dans une deuxième étape, une approche de groupement pour découvrir les grandes tendances observables sur nos différentes traces.

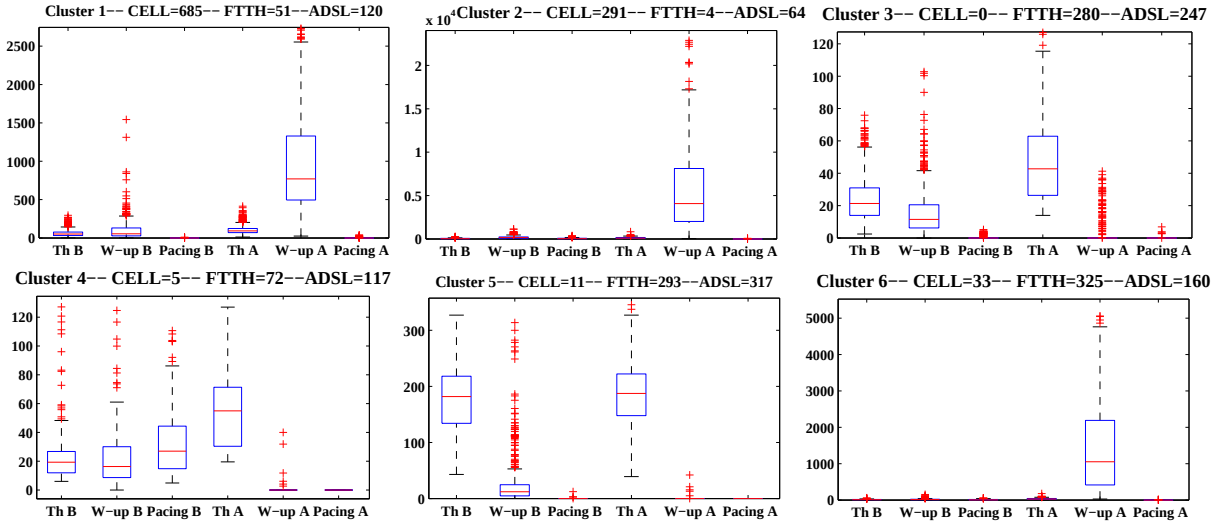
L'application des t-SNE à nos données à 6 dimensions conduit aux résultats de la Figure C.17(a). Ces résultats indiquent qu'un regroupement naturel existe dans nos données. En outre, une valeur raisonnable pour le nombre de clusters se situe entre 5 et 10. La droite de la Figure C.17(a) suggère que certains groupes sont dominés par une technologie d'accès spécifiques tandis que d'autres sont mixtes. Nous avons fixé le nombre de groupes dans l'algorithme de Kmeans à 6.

La Figure C.17(b) illustre les 6 clusters obtenus par application de Kmeans. Nous utilisons les boîtes à moustaches pour obtenir des représentations compactes des valeurs correspondantes de chaque dimension. Nous indiquons, sur le dessus de chaque groupe, le nombre d'échantillons dans le cluster pour chaque technologie d'accès.

Nous utilisons le même nombre d'échantillons pour chaque technologie d'accès pour empêcher toute partialité dans le regroupement, ce qui nous limite à 1000 échantillons, en raison de la courte durée de la trace FTTH. Dans la Figure 7.6(b) nous avons tracé la taille des transferts de chaque cluster et leur débit applicatif.



(a) T-SNE



(b) K-means

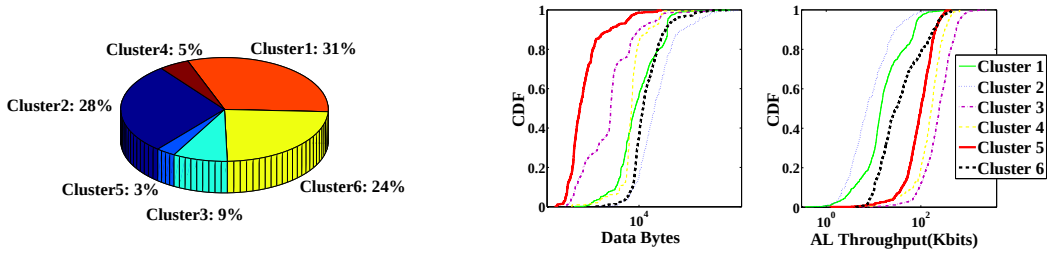
FIGURE C.17 – résultat des Groupement du Traffic de Recherche de Google

Nous observons que les clusters obtenus avec Kmeans correspondent à peu près la projection obtenue par le t-SNE, comme indiqué dans la partie à gauche de la Figure C.17(a), où des échantillons de données sont indexés à l'aide de leur identifiant dans les clusters dans Kmeans.

Avant de plonger dans l'interprétation des différentes groupes, nous observons que trois d'entre eux représentent la majorité des octets. En effet, la Figure 7.6(a) indique que les clusters 1 et 2 et 6 représentent 83 % des octets. Nous commençons d'abord par nous concentrer sur les groupes dominants.

Les groupes 1, 2 et 6 sont caractérisés par des grands temps de préparation des données du côté client, c'est à dire de longs temps d'attente du côté client entre deux requêtes consécutives. Ces valeurs de temps d'attente sont de l'ordre de quelques secondes, ce qui est compatible avec les actions humaines. Ce comportement est en corrélation avec l'utilisation les moteurs de recherche typique, où l'utilisateur soumet d'abord une requête analyse ensuite les résultats avant une poursuite sa requête ou en cliquant sur un des liens de la page de résultat.

Nous pouvons également observer que les clusters 1 et 2 se composent principalement de connexions cellulaires tandis que le cluster 6 se compose essentiellement de transferts FTTH. Cela signifie que l'algorithme de clustering a d'abord basé sa décision sur le warm-up, puis sur la technologie d'accès. Comme l'ADSL offre des caractéristiques intermédiaires par rapport aux traces



(a) Volume en Octets (b) Paramêtres des Clusters

FIGURE C.18 – Principaux Paramêtres du Traffic de Recherche de Google

FTTH et cellulaire, les transferts ADSL offrent des grands warm-up dispersés sur les groupes 1, 2 and 6.

Considérons maintenant les groupes 3, 4 et 5. Ces groupes représentent une petite fraction des transferts, et sont caractérisés par plusieurs caractéristiques notables. Premièrement, nous ne trouvons presque pas de connexions cellulaires dans ces clusters. Deuxièmement, ils totalisent les deux tiers des connexions ADSL et FTTH, même si elles sont plus petites que celles dans les groupes 1, 2 et 6 - voir la Figure C.18(b).

Troisièmement, ces clusters, contrairement aux groupes 1, 2 et 6 ont des valeurs négligeables de temps de préparation des données du coté client.

Une analyse plus approfondie de ces groupes a révélé qu'ils correspondent à des connexions très courtes avec un échange de deux trames HTTP. En fait, le groupe 3 correspond aux cas où un client ouvre la page de recherche Google Web dans son navigateur Internet sans effectuer aucune demande de recherche, puis après un temps-mort de 10 secondes, le serveur de Google ferme la connexion. D'autre part, les clusters 4 et 5 correspondent à des requêtes GET et des réponses OK, correspondant à une recherche effective, la principale différence entre les groupes 4 et 5 étant les valeurs de RTT et la taille de connexion.

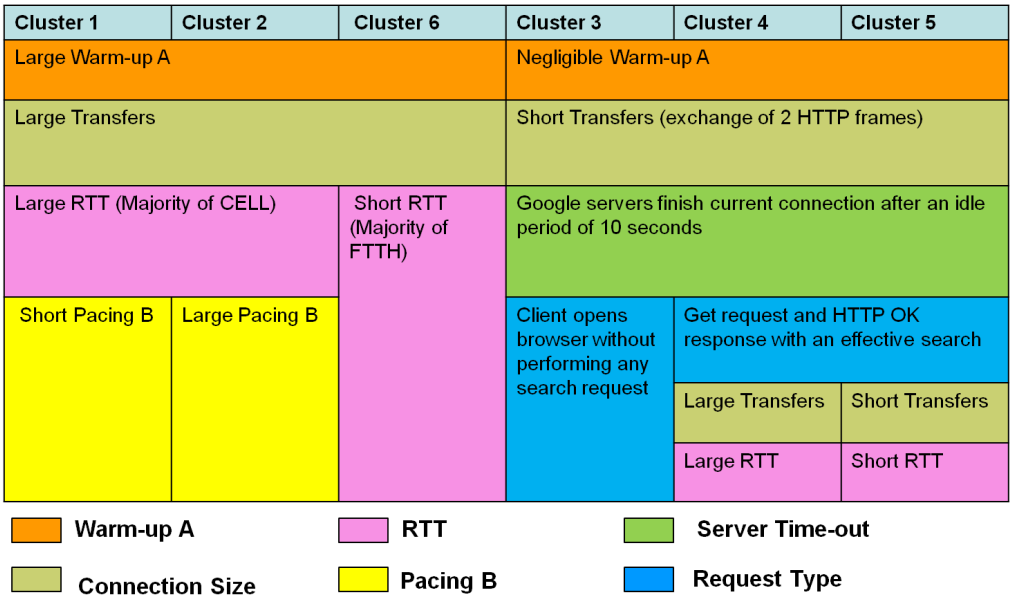


FIGURE C.19 – Vue Global des Groupes

Plus généralement, nous pensons que notre méthode, lorsqu'elle est appliquée au profilage d'autres services, permettra d'identifier des groupes qui peuvent être liés au comportement du service à l'étude, tandis que d'autres concerneront les comportements anormaux ou inhabituels. Ces dernier pourraient nécessiter par la suite une étude plus approfondie. Pour le cas du moteur de recherche Google, nous ne croyons pas que les groupes 3, 4 et 5 correspondent à des anomalies qui affectent la qualité de l'expérience des utilisateurs. Nous avons trouvé que très peu de cas où l'impact du serveur a été dominant sur les performances et sur la qualité d'expérience de l'utilisateur final.

C.10 Conclusion

Les performances d'Internet ont été mesurées de diverses façons depuis sa création par le réseau ARPANET en 1969. Un certain nombre de tendances ont affecté la façon dont Internet a été mesuré dans ce laps de temps. Certaines tendances dépendent de l'amélioration de la technologie : la technologie Internet a changé au fil du temps, ce qui a rendu certaines mesures plus difficiles à obtenir et d'autres plus faciles. D'autres tendances sont des questions d'échelle : le prodigieux essor de l'Internet nous a forcé à faire évoluer les métriques pour mesurer l'évaluation des performances, et a déclenché le développement de nouvelles méthodes de mesure ainsi que les outils statistiques. Enfin les tendances sociales (réseaux p2p, sociaux) et l'importance économique de la communication sur Internet, ont modifié la nature des mesures nécessaires pour l'évaluation de performance.

Principalement, ces tendances découlaient de l'interaction entre les objectifs de mesure et de ses difficultés en même temps. Dans cette thèse, nous avons examiné différentes difficultés que peuvent rencontrer les experts lors de la collecte de données avec les nouvelles architectures disponibles (spécialement avec l'usage d'APN, proxy et les réseaux d'entreprises) et ensuite, nous avons proposé une nouvelle méthodologie pour mettre en évidence de nouveaux paramètres qui peuvent influencer la performance perçue par le client. Enfin, nous avons discuté des approches pour détecter des anomalies dans Internet et les environnements d'entreprise.

Dans ce dernier chapitre, nous cherchons à modéliser quelques caractéristiques importantes d'Internet, la mesure de trafic entreprise et á montrer les problématiques où nous réussissons à progresser, et où plus d'efforts doivent être effectuées. Nous allons maintenant revoir les points abordés par cette thèse et ainsi présenter nos principales contributions. Enfin, nous donnons notre vision sur la façon dont ces recherches pourraient être étendues dans l'avenir.

Défnition des Petits transferts et Impact de l'Application.

Tout en analysant les performances des transferts TCP, nous nous sommes concentrés sur les transferts qui correspondent à des transferts bien formés et complets, du point de vue TCP, et qui remplissent les critères suivants : une étape complète d'établissement de connexion, au moins un segment de données TCP dans chaque direction, et la connexion doit se terminer soit par un drapeau FIN ou RESET.

Nous avons introduit aussi une première définition d'une connexion TCP courte, concept couramment utilisé dans la littérature. *Une courte connexion TCP est une connexion bien formée incapable d'accomplir un FR/R, après une détection de perte de paquets.* Nous avons présenté un aperçu de l'impact de l'application, sur les transferts TCP. Nous avons montré que si les pertes peuvent avoir un impact négatif sur les transferts TCP courts, l'application affecte de manière significative le temps de transfert de presque toutes les connexions.

Nous avons démontré que la sensibilité à la perte concerne aussi les grands transferts dont la

taille des trains échangés fait moins de 3 paquets. Une telle caractéristique a une influence directe sur la capacité de TCP à se remettre d'une perte á l'aide d'une retransmission rapide.

L'architecture de l'opérateur doit être prise en compte Nous avons souligné que, dans les réseaux modernes comme les réseaux cellulaires, l'estimation de latence se révéle complexe. Nous avons démontré que la latence peut être sous-estimée en raison de l'utilisation de nouveaux mécanismes ou de services, comme l'adaptation de contenu ou de l'accélération des applications. Nous étudions comment ces mécanismes impactent nos mesures et la performance perçue par les utilisateurs finaux. Le message clé ici, est que plusieurs dispositifs spécifiques pourraient affecter les mesures de performance dans les réseaux cellulaires classiques. Certains doivent être pris en compte lorsque nous effectuons les études de mesure.

Les suspects habituels ne suffisent pas à expliquer les performances. Lors de notre analyse nous avons utilisé une approche classique pour comparer les performances des différentes technologies d'accès : cellular, FTTH et ADSL, afin d'évaluer si les clients profitent pleinement de leur accès à large bande. Nous nous sommes concentrés sur les deux facteurs clés qui influencent le débit des transferts TCP (formule de débit TCP [89]), le taux de perte et la latence, qui suggèrent que la performance de FTTH devrait sensiblement surpasser celui de l'ADSL, qui devrait à son tour surpasser celui des la trace cellulaire. Il s'est avéré que la réalité est plus complexe. Bien que la technologie cellulaire offre un débit nettement plus petit, en accord avec le RTT et les taux de perte, FTTH et ADSL ont des performances beaucoup plus proches que ce que le RTT et les pertes suggèrent. Nous concluons que se concentrer sur les paramètres classiques d'analyse de performance ne conduisent pas à une pleine compréhension débit perçu par les client.

Analyse plus fine des performance. Nous avons proposé une méthode de décomposition des transferts de données pour chaque connexion bien formée. L'approche développée est illustrée avec l'ensemble des traces recueillies. Notre approche permet d'extraire automatiquement l'impact de l'application, l'accès, le serveur et le comportement du client.

L'application de cette technique pour le service de recherche Google a démontré qu'elle fournit des résultats facilement interprétables. Elle permet par exemple de localiser l'impact de l'utilisation ou les caractéristiques de la technologie d'accès.

Proposition d'approches pour détecter les comportements anormaux. Notre méthode a permis d'identifier les causes de certains problèmes de performance, qui peuvent être soit des pertes soit quelques moments d'inactivité pendant la préparation des données ou du transfert. Nous avons appliqué cette méthodologie à plusieurs traces correspondant à des trafics Internet ou entreprise. Nous avons démontré que les accès filaire (ADSL et FTTH) et sans fil (cellulaire) adoptent des stratégies différentes pour récupérer des pertes de paquets, et que les stratégies observées sur la technologie cellulaire semblent plus efficaces que sur l'ADSL et l'FTTH.

Nous avons montré que notre méthode de profilage des transferts (ou des parties de transferts) affectés par des pertes est en mesure de découvrir différents types d'anomalies, certaines étant liées à la configuration des serveurs et d'autres partagés par plusieurs services.

D'autre part, à travers l'étude d'un environnement d'entreprise, nous avons proposé deux approches de détection d'anomalies. Nous réussissons à identifier les comportements de plusieurs connexions anormales, avec différents degrés de criticité. La tâche de la définition du comportement anormal est plus complexe que dans le cas Internet, principalement à cause de l'environnement d'entreprise caractérisé par des applications spécifiques, par exemple, SYMANTEC, RPC, etc.

La principale différence avec le trafic Internet est que la connaissance du comportement de l'application est très important (exécuté en arrière-plan ou non, etc.) pour pouvoir mieux interpréter les résultats.

Enfin, les approches présentées ne sont pas à l'abri de faux positifs et nécessitent de remettre en oeuvre des experts, une fois le problème identifié. Mais, au final, nous croyons que nous avons déjà un outil qui fait un bon travail pour détecter certains comportements anormaux.

Dans les paragraphes suivants, nous identifions les futurs axes de recherche et les orientations possibles dans trois catégories : premièrement, la méthodologie, deuxièmement, l'échelle d'analyse et, enfin, l'architecture de l'approche utilisée.

Quitter le niveau connexion. Dans cette thèse et la thèse de Matti Siekkinen [101], l'accent a été mis sur l'analyse des connexions individuelles. Alors qu'il s'est avéré être un sujet riche et complexe, ce qui permet d'obtenir de nombreuses réflexions sur la performance perçue par les utilisateurs finaux, ces analyses comportent des limitations spécifiques. Une question cruciale est que la dépendance entre les flux n'est pas prise en compte. Ainsi, beaucoup de travaux ont proposé des approches graphe [65, 102, 83] pour identifier l'application ou les comportements des utilisateurs. De telles approches sont intéressantes car elles fournissent un aperçu de haut niveau des clients et du comportement des applications. Une poursuite intéressante de ce travail, pourrait être de combiner ces types d'approches avec notre approche de bas niveau, au niveau de la connexion, afin de mieux documenter les résultats obtenus pendant le processus de regroupement que nous utilisons.

Analyse à grande échelle. Nous avons été confrontés, dans notre travail à un problème qui est commun à beaucoup d'études d'analyse de trafic : nous avons passé beaucoup de temps à développer et à calibrer nos techniques d'analyse. De plus, en raison de la taille limitée de nos traces, nos résultats ne sont pas établis sur une base entièrement solide. Nous nous attendons à ce que la poursuite de ce travail voit l'application des méthodes que nous avons développées sur une plus grande variété de traces, par exemple, plusieurs traces cellulaires du même GGSN ou plusieurs jours/semaines de trafic entreprise.

Cloud computing. Le cloud computing n'est pas seulement un mot à la mode du moment, mais susceptible de devenir l'avenir du serveurs de données dans un grand nombre de scénarios. Dans un tel contexte, les serveurs ou services distants sont accessibles par les utilisateurs finaux et le problème de performance devient crucial dans ces environnements complexes à la fois au niveau réseau, mais aussi d'un point de vue du système. Nous nous attendons à ce que la méthodologie que nous avons développée constitue un point de départ pour diagnostiquer les problèmes de performance dans ce contexte.

Bibliography

[1] Marcel Dischinger, Andreas Haeberlen, Krishna P. Gummadi, and Stefan Saroiu. Characterizing residential broadband networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 43–56, New York, NY, USA, 2007. ACM.

[2] Matti Siekkinen, Denis Collange, Guillaume Urvoy-keller, and Ernst W. Biersack. Performance limitations of adsl users : A case study. In *In Proceedings of the 8th Passive and Active Measurement Conference (PAM)*, 2007.

[3] Yin Zhang, Lee Breslau, Vern Paxson, and Scott Shenker. On the characteristics and origins of internet flow rates. *SIGCOMM Comput. Commun. Rev.*, 32 :309–322, August 2002.

[4] Marco Mellia, Ion Stoica, and Hui Zhang. Tcp model for short lived flows. *IEEE Communications Letters*, 6 :85–87, 2002.

[5] Nadia Ben Azzouna and Fabrice Guillemin. Analysis of adsl traffic on an ip backbone link. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, volume 7, pages 3742 – 3746 vol.7, dec. 2003.

[6] Urtzi Ayesta, Konstantin E. Avrachenkov, France Telecom RD, and Rue Albert Einstein. The effect of the initial window size and limited transmit algorithm on the transient behavior of tcp transfers, 2002.

[7] Ayesta Ab Avrachenkov, U. Ayesta Ab, K. E. Avrachenkov B, E. Altman, C. Barakat, and P. Dube C. Multilevel approach for modeling short tcp sessions.

[8] Neal Cardwell, Stefan Savage, and Tom Anderson. Modeling the performance of short tcp connections. Technical report, 1998.

[9] Nea Cardwell, Stefan Savage, and Tom Anderson. Modeling tcp latency. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1742 –1751 vol.3, mar 2000.

[10] Chadi Barakat and Eitan Altman. Performance of short tcp transfers. Technical report.

[11] Shirin Ebrahimi-taghizadeh, Ahmed Helmy, and Sandeep Gupta. Tcp vs. tcp : a systematic study of adverse impact. In *of Short-lived TCP Flows on Long-lived TCP Flows ?, IEEE INFOCOM*, pages 926–937, 2005.

[12] Martin Arlitt, Balachander Krishnamurthy, and Jeffrey C. Mogul. Predicting short-transfer latency from tcp arcana : a trace-based validation. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, pages 19–19, Berkeley, CA, USA, 2005. USENIX Association.

[13] Vern Paxson and Marck Allman. Computing tcp's retransmission timer, 2000.

[14] Hari Balakrishnan, Hariharan S. Rahul, and Srinivasan Seshan. An integrated congestion management architecture for internet hosts. pages 175–187, 1999.

[15] Xuan Chen and John Heidemann. Preferential treatment for short flows to reduce web latency, 2003.

[16] Janardhan R. Iyengar, O L. Caro, and Paul D. Amer. Dealing with short tcp flows : A survey of mice in elephant shoes.

[17] Bob Braden. T/tcp – tcp extensions for transactions functional specification, 1994.

[18] Roy Thomas Fielding, U. C. Irvine, J. Gettys, Jeffrey C Mogul, Henrik Frystyk Nielsen, and Tim Berners-Lee. Transfer protocol – http/1.1, 1997.

[19] J. Touch. Tcp control block interdependence, 1997.

[20] Venkata Narayana Padmanabhan and Venkata Narayana Padmanabhan. Addressing the challenges of web data transport, 1998.

[21] Venkata N. Padmanabhan and Randy H. Katz. Tcp fast start : A technique for speeding up web transfers. pages 41–46, 1998.

[22] Yin Zhang. Speeding up short data transfers : Theory, architecture support, and simulation results. In *in Proc. NOSSDAV 2000, Chapel*. Chapel Hill, 2000.

[23] Mark Allman, Sally Floyd, and Craig Partridge. Increasing tcp's initial window, 2002.

[24] Marco Mellia, Michela Meo, and Claudio Casetti. Tcp smart-framing : using smart segments to enhance the performance of tcp. In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 3, pages 1708 –1712 vol.3, 2001.

[25] Liang Guo and Ibrahim Matta. The war between mice and elephants. In *Network Protocols, 2001. Ninth International Conference on*, pages 180 – 188, nov. 2001.

[26] Amit K. Jain, Sally Floyd, and Draft amit-quick-start . Txt Sally Floyd. Quick-start for tcp and ip, 2002.

[27] Gullaume Vu-Brugier. Analysis of the impact of early fiber access deployment on residential internet traffic. In *Teletraffic Congress, 2009. ITC 21 2009. 21st International*, pages 1 –8, sept. 2009.

[28] Kenjiro Cho, Kensuke Fukuda, Hiroshi Esaki, and Akira Kato. The impact and implications of the growth in residential user-to-user traffic. volume 36, pages 207–218, New York, NY, USA, August 2006. ACM.

[29] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. On dominant characteristics of residential broadband internet traffic. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 90–102, New York, NY, USA, 2009. ACM.

[30] Gregor Maier, Fabian Schneider, and Anja Feldmann. A first look at mobile hand-held device traffic. In *Proceedings of the 11th international conference on Passive and active measurement*, PAM'10, pages 161–170, Berlin, Heidelberg, 2010. Springer-Verlag.

[31] Young J. Won, Byung-Chul Park, Seong-Cheol Hong, Kwang Bon Jung, Hong-Taek Ju, and James W. Hong. Measurement analysis of mobile data networks. In *Proceedings of the 8th international conference on Passive and active network measurement*, PAM'07, pages 223–227, Berlin, Heidelberg, 2007. Springer-Verlag.

[32] Philipp Svoboda, Fabio Ricciato, Werner Keim, and Markus Rupp. Measured web performance in gprs, edge, umts and hsdpa with and without caching. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1 –6, june 2007.

[33] Louis Plissonneau and Guillaume Vu-Brugier. Mobile data traffic analysis : How do you prefer watching videos ? In *Teletraffic Congress (ITC), 2010 22nd International*, pages 1 –8, sept. 2010.

[34] Junxian Huang, Qiang Xu, Birjodh Tiwana, Z. Morley Mao, Ming Zhang, and Paramvir Bahl. Anatomizing application performance differences on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 165–178, New York, NY, USA, 2010. ACM.

[35] Ruoming Pang, Mark Allman, Mike Bennett, Jason Lee, Vern Paxson, and Brian Tierney. A first look at modern enterprise traffic. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, pages 2–2, Berkeley, CA, USA, 2005. USENIX Association.

[36] Godfrey Tan, Massimiliano Poletto, John Guttag, and Frans Kaashoek. Role classification of hosts within enterprise networks based on connection patterns. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 2–2, Berkeley, CA, USA, 2003. USENIX Association.

[37] Yu Jin, Esam Sharafuddin, and Zhi-Li Zhang. Unveiling core network-wide communication patterns through application traffic activity graph decomposition. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, SIGMETRICS '09, pages 49–60, New York, NY, USA, 2009. ACM.

[38] Theophilus Benson, Ashok Anand, Aditya Akella, and Ming Zhang. Understanding data center traffic characteristics. *SIGCOMM Comput. Commun. Rev.*, 40 :92–99, January 2010.

[39] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *In Proceedings of ACM Mobicom*, pages 107–118. ACM Press, 2002.

[40] Ramana Rao Kompella, Jennifer Yates, Albert Greenberg, and Alex C. Snoeren. Ip fault localization via risk modeling. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 57–70, Berkeley, CA, USA, 2005. USENIX Association.

[41] Yu-Chung Cheng, Mikhail Afanasyev, Patrick Verkaik, Péter Benkö, Jennifer Chiang, Alex C. Snoeren, Stefan Savage, and Geoffrey M. Voelker. Automating cross-layer diagnosis of enterprise wireless networks. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pages 25–36, New York, NY, USA, 2007. ACM.

[42] Amit P. Jardosh, Krishna N. Ramachandran, Kevin C. Almeroth, and Elizabeth M. Belding-Royer. Understanding link-layer behavior in highly congested ieee 802.11b wireless networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, E-WIND '05, pages 11–16, New York, NY, USA, 2005. ACM.

[43] Srikanth Kandula, Ratul Mahajan, Patrick Verkaik, Sharad Agarwal, Jitendra Padhye, and Paramvir Bahl. Detailed diagnosis in enterprise networks. pages 243–254, 2009.

[44] Sangho Shin and Henning Schulzrinne. Measurement and analysis of the voip capacity in ieee 802.11 wlan. *IEEE Transactions on Mobile Computing*, 8 :1265–1279, September 2009.

[45] Frederic Giroire, Jaideep Chandrashekar, Gianluca Iannaccone, Konstantina Papagiannaki, Eve M. Schooler, and Nina Taft. The cubicle vs. the coffee shop : behavioral modes in enterprise end-users. In *Proceedings of the 9th international conference on Passive and active network measurement*, PAM'08, pages 202–211, Berlin, Heidelberg, 2008. Springer-Verlag.

[46] Boris Nechaev, Vern Paxson, Mark Allman, and Andrei Gurtov. On calibrating enterprise switch measurements. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 143–155, New York, NY, USA, 2009. ACM.

[47] Boris Nechaev, Mark Allman, Vern Paxson, and Andrei Gurtov. A preliminary analysis of tcp performance in an enterprise network. In *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, INM/WREN'10, pages 7–7, Berkeley, CA, USA, 2010. USENIX Association.

[48] Augustin Soule, Kavé Salamatian, and Nina Taft. Combining filtering and statistical methods for anomaly detection. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, pages 31–31, Berkeley, CA, USA, 2005. USENIX Association.

[49] Paul Barford, Jeffery Kline, David Plonka, and Amos Ron. A signal analysis of network traffic anomalies. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, IMW '02, pages 71–82, New York, NY, USA, 2002. ACM.

[50] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. In *In ACM SIGCOMM*, pages 219–230, 2004.

[51] Bernhard Tellenbach, Martin Burkhart, Didier Sornette, and Thomas Maillart. Beyond shannon : Characterizing internet traffic with generalized entropy metrics. In *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, PAM '09, pages 239–248, Berlin, Heidelberg, 2009. Springer-Verlag.

[52] Fernando Silveira, Christophe Diot, Nina Taft, and Ramesh Govindan. Astute : detecting a different class of traffic anomalies. In *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, SIGCOMM '10, pages 267–278, New York, NY, USA, 2010. ACM.

[53] Antoine Scherrer, Nicolas Larrieu, Philippe Owezarski, Pierre Borgnat, and Patrice Abry. Non-gaussian and long memory statistical characterizations for internet traffic with anomalies. *IEEE Trans. Dependable Secur. Comput.*, 4 :56–70, January 2007.

[54] Thomas Dubendorfer and Bernhard Plattner. Host behaviour based early detection of worm outbreaks in internet backbones. In *In Proceedings of 14th IEEE WET ICE / STCA security workshop. IEEE*, 2005.

[55] Marco Mellia, Michela Meo, Luca Muscariello, and Dario Rossi. Passive analysis of tcp anomalies. *Comput. Netw.*, 52 :2663–2676, October 2008.

[56] Peter Romirer-Maierhofer, Fabio Ricciato, Alessandro D'Alconzo, Robert Franzan, and Wolfgang Karner. Network-wide measurements of tcp rtt in 3g. In *Proceedings of the First International Workshop on Traffic Monitoring and Analysis*, TMA '09, pages 17–25, Berlin, Heidelberg, 2009. Springer-Verlag.

[57] Matti Siekkinen, Guillaume Urvoy-Keller, Ernst W. Biersack, and Denis Collange. A root cause analysis toolkit for tcp. *Comput. Netw.*, 52 :1846–1858, June 2008.

[58] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A. Maltz, and Ming Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. *SIGCOMM Comput. Commun. Rev.*, 37 :13–24, August 2007.

[59] Simona Brugnoni, Guido Bruno, Roberto Manione, Enrico Montariolo, Elio Paschetta, and Luisella Sisto. An expert system for real time fault diagnosis of the italian telecommunications network. In *Proceedings of the IFIP TC6/WG6.6 Third International Symposium on Integrated Network Management with participation of the IEEE Communications Society CNOM and with support from the Institute for Educational Services*, pages 617–628, Amsterdam, The Netherlands, The Netherlands, 1993. North-Holland Publishing Co.

[60] Mike Chen, Emre Kiciman, Eugene Fratkin, Armando Fox, and Eric Brewer. Pinpoint : Problem determination in large, dynamic internet services. In *DSN*, pages 595–604. IEEE Computer Society, 2002.

[61] Alice X. Zheng, Jim Lloyd, and Eric Brewer. Failure diagnosis using decision trees. In *Proceedings of the First International Conference on Autonomic Computing*, pages 36–43, Washington, DC, USA, 2004. IEEE Computer Society.

[62] William Aiello, Charles Kalmanek, Patrick Mcdaniel, Subhabrata Sen, Oliver Spatscheck, and Jacobus Van Der Merwe. Analysis of communities of interest in data networks. In *In PAM*, 2005.

[63] Su Chang and Thomas E. Daniels. Correlation based node behavior profiling for enterprise network security. In *Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies*, SECURWARE '09, pages 298–305, Washington, DC, USA, 2009. IEEE Computer Society.

[64] Patrick Drew McDaniel, Subhabrata Sen, Oliver Spatscheck, Jacobus E. van der Merwe, William Aiello, and Charles R. Kalmanek. Enterprise security : A community of interest based approach. In *NDSS'06*, pages –1–1, 2006.

[65] Marios Iliofotou, Prashanth Pappu, Michalis Faloutsos, Michael Mitzenmacher, Sumeet Singh, and George Varghese. Network monitoring using traffic dispersion graphs (tdgs). In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 315–320, New York, NY, USA, 2007. ACM.

[66] Srikanth Kandula, Ranveer Chandra, and Dina Katabi. What's going on ? : learning communication rules in edge networks. *SIGCOMM Comput. Commun. Rev.*, 38 :87–98, August 2008.

[67] Siekkinen Matti. Intrabase. `http://intrabase.eurecom.fr/tmp/index.html`.

[68] Matti Siekkinen, Ernst W. Biersack, Guillaume Urvoy-Keller, Vera Goebel, and Thomas Peter Plagemann. Intrabase : integrated traffic analysis based on a database management system. In *End-to-End Monitoring Techniques and Services, 2005. Workshop on*, pages 32 – 46, may 2005.

[69] Marcin Pietrzyk, Jean-Laurent Costeux, Guillaume Urvoy-Keller, and Taoufik En-Najjary. Cha0llenging statistical classification for operational usage : the adsl case. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 122–135, New York, NY, USA, 2009. ACM.

[70] Hyun chul chul Kim, kc c claffy, Marina Fomenkov, Dhiman Barman, Michalis Faloutsos, and KiYoung Lee. Internet Traffic Classification Demystified : Myths, Caveats, and the Best Practices. In *ACM SIGCOMM CoNEXT*, New York, NY, Dec 2008. ACM SIGCOMM CoNEXT.

[71] Tom Dunigan. Tcp auto-tuning zoo. 2006. `http://www.csm.ornl.gov/~dunigan/netperf/auto.html`.

[72] Sushant Rewaskar, Jasleen Kaur, and F. Donelson Smith. A passive state-machine approach for accurate analysis of tcp out-of-sequence segments. *SIGCOMM Comput. Commun. Rev.*, 36 :51–64, July 2006.

[73] Transmission control protocol, 1981.

[74] Srinivas Shakkottai, R. Srikant, Nevil Brownlee, Andre Broido, and kc claffy. The rtt distribution of tcp flows in the internet and its impact on tcp-based flow control, 2004.

[75] Sharad Jaiswal, Gianluca Iannaccone, Christophe Diot, Jim Kurose, and Don Towsley. Measurement and classification of out-of-sequence packets in a tier-1 ip backbone. *IEEE/ACM Trans. Netw.*, 15 :54–66, February 2007.

[76] Hao Jiang and Constantinos Dovrolis. Passive estimation of tcp round-trip times. *SIGCOMM Comput. Commun. Rev.*, 32 :75–88, July 2002.

[77] Bryan Veal, Kang Li, and David Lowenthal. New methods for passive estimation of tcp round-trip times. In *In Proceedings of the Passive and Active Measurement Workshop*, 2005.

[78] ZhenYu Liu, ShengLi Xie, and Yue Lai. A fast bloom filters method in apn filtering. In *Proceedings of the 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application - Volume 02*, pages 145–150, Washington, DC, USA, 2008. IEEE Computer Society.

[79] Naimul Basher, Aniket Mahanti, Anirban Mahanti, Carey Williamson, and Martin Arlitt. A comparative analysis of web and peer-to-peer traffic. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 287–296, New York, NY, USA, 2008. ACM.

[80] Atif Nazir, Saqib Raza, and Chen-Nee Chuah. Unveiling facebook : a measurement study of social network based applications. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, pages 43–56, New York, NY, USA, 2008. ACM.

[81] Sirikarn Pukkawanna, Vasaka Visootfiviseth, and Panita Pongpaibool. Classification of web-based email traffic in thailand. In *Communications and Information Technologies, 2006. ISCIT '06. International Symposium on*, pages 440 –445, 18 2006-sept. 20 2006.

[82] Marcin Pietrzyk, Guillaume Urvoy-Keller, and Jean-Laurent Costeux. Revealing the unknown adsl traffic using statistical methods. In *Proceedings of the First International Workshop on Traffic Monitoring and Analysis*, TMA '09, pages 75–83, Berlin, Heidelberg, 2009. Springer-Verlag.

[83] Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. Blinc : multilevel traffic classification in the dark. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '05, pages 229–240, New York, NY, USA, 2005. ACM.

[84] Laurent Bernaille, Renata Teixeira, Ismael Akodkenou, Augustin Soule, and Kave Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36 :23–26, April 2006.

[85] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc claffy. Transport layer identification of p2p traffic. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 121–134, New York, NY, USA, 2004. ACM.

[86] Bowei Du, Michael Demmer, and Eric Brewer. Analysis of www traffic in cambodia and ghana. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 771–780, New York, NY, USA, 2006. ACM.

[87] Mark Allman, Hari Balakrishnan, and Sally Floyd. Enhancing tcp's loss recovery using limited transmit, 2000.

[88] Mats Folke, Sara Landstroem, and Ulf Bodin. On the tcp minimum retransmission timeout in a high-speed cellular network. pages 1 –6, april 2005.

[89] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling tcp throughput : a simple model and its empirical validation. *SIGCOMM Comput. Commun. Rev.*, 28 :303–314, October 1998.

[90] Luigi Alfredo Grieco Antonio Barbuzzi. Desrto : An effective algorithm for srto detection in tcp connections. 2010.

[91] Aymen Hafsaoui, Denis Collange, and Guillaume Urvoy-Keller. Revisiting the performance of short tcp transfers. *8th International IFIP-TC 6 Networking Conference, Aachen*, pages 260–273, May 2009.

[92] W. Richard Stevens. Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, 1997.

[93] Mathworks. Marlab product documentation. `http://www.mathworks.com/help/toolbox/stats/kmeans.html`.

[94] Laurens J.P. van der Maaten. t-distributed stochastic neighbor embedding. http://homepage.tudelft.nl/19j49/t-SNE.html.

[95] Scalable Networks. Qualnet. 2011. http://www.scalable-networks.com/products/qualnet/.

[96] François Baccelli and David R. McDonald. A stochastic model for the throughput of non-persistent tcp flows. In *Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, valuetools '06, New York, NY, USA, 2006. ACM.

[97] Sofia Stamou and Lefteris Kozanidis. Impact of search results on user queries. In *Proceeding of the eleventh international workshop on Web information and data management*, WIDM '09, pages 7–10, New York, NY, USA, 2009. ACM.

[98] Abhinav Pathak, Y. Angela Wang, Cheng Huang, Albert Greenberg, Y. Charlie Hu, Randy Kern, Jin Li, and Keith W. Ross. Measuring and evaluating tcp splitting for cloud services. In *Proceedings of the 11th international conference on Passive and active measurement*, PAM'10, pages 41–50, Berlin, Heidelberg, 2010. Springer-Verlag.

[99] Christoph H. Hochstatter. Internet per umts : So falschen deutsche provider webinhalte. http://www.zdnet.de/magazin/41515603/.

[100] SNIA. Common internet file system (cifs)technical reference revision : 1.0. http://www.scribd.com/doc/52514900/CIFS-TR-1p00-FINAL.

[101] Matti Siekkinen. Root cause analysis of tcp throughput : Methodology, techniques, and applications. http://users.tkk.fi/~siekkine/pub/Phd_Thesis_final.pdf.

[102] Marios Iliofotou, Hyun chul Kim, Michalis Faloutsos, Michael Mitzenmacher, and George Varghese. Graption : A graph-based p2p traffic classification framework for the internet backbone. *Comput. Netw.*, 55 :1909–1920, June 2011.

[103] Fabio Ricciato, Eduard Hasenleithner, and Peter Romirer-Maierhofer. Traffic analysis at short time-scales : an empirical case study from a 3g cellular network. *Network and Service Management, IEEE Transactions on*, 5(1) :11 –21, march 2008.