MEMOIRE DE FIN D’ETUDE ESIEA

ESTIMATION & SYNTHESIS OF
INTERNET TRAFFIC MATRICES

ESTIMATION & SYNTHESE DES MATRICES DE TRAFIC
SUR LE RESEAU INTERNET

HENNAULT Jérémy

Foreman of the Jury:  
Président de Jury  
Peter WILSON

Academic Supervisor:  
Tuteur Pédagogique  
Philippe DUROUCHOUX

Internship Supervisor:  
Maître de Stage  
Edward CHLEBUS

12/09/2010
# Table of Contents

1. Acknowledgement .................................................................................................................. 1

2. Abstract .................................................................................................................................. 2

3. Résumé .................................................................................................................................... 3

4. Introduction .............................................................................................................................. 4

   4.1. My host Institution: Illinois Institute of Technology ......................................................... 5

   4.1.1. History and Location ................................................................................................. 5

   4.1.2. Organization ............................................................................................................. 6

   4.1.3. Placement and choice of the project ......................................................................... 8

   4.2. Genesis of the project ..................................................................................................... 9

   4.2.1. Definition of a Traffic Matrix .................................................................................. 9

   4.2.2. Why is it important to estimate Traffic Matrices? ................................................. 13

   4.2.3. Obstacles and issues in computing Traffic Matrices ............................................. 13

5. State of the art ........................................................................................................................ 14

   5.1. Data available for study ................................................................................................. 15

   5.2. Existing Techniques ....................................................................................................... 17

      5.2.1. Parametric models ............................................................................................... 17

      5.2.2. Non-parametric models ....................................................................................... 17

6. Technical aspects and progress ............................................................................................ 18

   6.1. Understanding the problematic and defining a working plan .................................. 18

   6.2. Extraction and manipulation of the raw data .............................................................. 19

      6.2.1. Choice of a programming language and environment .................................... 20

      6.2.2. Unifying both datasets to the same format ....................................................... 23

      6.2.3. Building a toolbox for batch data manipulation .............................................. 25

   6.3. Analysis of the data and research of better models ..................................................... 28

      6.3.1. Mathematical background ................................................................................... 28

   6.4. Results of distribution fitting ....................................................................................... 39
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4.1. Choice of the data to analyze</td>
<td>39</td>
</tr>
<tr>
<td>6.4.2. Distribution fitting</td>
<td>41</td>
</tr>
<tr>
<td>6.5. Conclusion</td>
<td>51</td>
</tr>
<tr>
<td>6.6. Further work</td>
<td>51</td>
</tr>
<tr>
<td>7. Human aspects of the project</td>
<td>52</td>
</tr>
<tr>
<td>8. Conclusion</td>
<td>53</td>
</tr>
<tr>
<td>References</td>
<td>55</td>
</tr>
<tr>
<td>Credits</td>
<td>57</td>
</tr>
<tr>
<td>Appendices</td>
<td>58</td>
</tr>
<tr>
<td>Appendix A – ABILENE XML formatting Source Code</td>
<td>58</td>
</tr>
<tr>
<td>Appendix B – A Versatile Probability Distribution for Light and Heavy Tails of Web File Sizes</td>
<td>62</td>
</tr>
<tr>
<td>Appendix C – Truncated Pareto distribution algorithms (fitting and CDF) – Matlab Code</td>
<td>69</td>
</tr>
<tr>
<td>Appendix D – Algorithms for Chlebus distribution fit and Chlebus CDF – Matlab Code</td>
<td>70</td>
</tr>
</tbody>
</table>
1. Acknowledgement

I would like to acknowledge the following people for their assistance during this project.

From ESIEA\textsuperscript{1}, I would like to thank Dr. P. Durouchoux who kindly offered to be my mentor during for this project.

Many classmates and professors at IIT\textsuperscript{2} have been helping me along these few months. I would like to thank my professor Dr. Gady Adam who taught me some basics of Machine Learning, my friend Pierre Khoury who helped me to handle the \textit{Matlab} software and my friend Paul Liverneaux who was patient enough during our team projects when I was busy working on this project.

I would like to show my appreciation to Mathew Roughan, professor at the University of Adelaide who genuinely accepted to answer my many questions about his research papers when I was in the early stages of this project.

Special thanks go to my parents and family who have been supporting me both morally and financially in my dream of studying in the United States.

And finally I would like to sincerely express my gratitude to Dr E. Chlebus who accepted to give me this project and offered me more help and support I could have ever wished.

\textsuperscript{1} \textit{Ecole Supérieure d'Informatique, Electronique, Automatique}: a French engineering school based in Paris, France

\textsuperscript{2} \textit{Illinois Institute of Technology}, University based in Chicago, IL, USA where I am completing my MSc degree
2. Abstract

This research project gave me the unique opportunity to discover new fields of study and apply my knowledge acquired throughout my graduate education both at ESIEA and IIT. The problematic is the following:

“How can we optimize the modelization, and estimation, of internet traffic through the use of traffic matrices?”

The challenge is thus to find accurate mathematical models, or mechanisms, that will allow us to generate synthetic traffic matrices (i.e. traffic matrices obtained from models inference). This is a critical issue since it is, to this date, impossible to effortlessly obtain traffic matrices from internet networks. There are two reasons for this: implementing tools to measure such data is not easy and can affect the networks performance; companies owning private networks have strategic interests and publicizing such data could harm their competitive assets.

Traffic matrices represent a very valuable source of data as they allow network administrators to simulate routing algorithms, predict networks loads and thus enable them to better design the structure of such networks. The purpose is ultimately to optimize infrastructure deployment, to better anticipate flows repartition, and to minimize the costs.

This project consists in searching and evaluating various probabilistic distributions to estimate traffic matrices, although a sizeable period has been dedicated to perform prep work.

To this date, only two datasets of traffic matrices have been released to the research community. One has been measured on ABILENE, a former North American backbone network for 24 weeks in 2004, and another has been extracted from GEANT European backbone network.

Therefore, the finality of this memoir is to first explain how the research data has been extracted and reformatted for the use of the project, and then to try to discover and analyze adapted and efficient methods for traffic matrix estimation that are consistent to both networks.
3. Résumé

Ce projet de recherche a été l’occasion de découvrir de nouveaux domaines et d’appliquer mes connaissances acquises lors de mon éducation à l’ESIEA et à l’IIT. La problématique est la suivante :

« Comment modéliser et estimer au mieux le trafic Internet par l’intermédiaire des matrices de trafic ? »

Le défi est donc de trouver des modèles mathématiques, ou des mécanismes permettant de générer des matrices de trafic synthétiques (c’est-à-dire obtenues à l’aide de modèles). L’enjeu est important puisqu’il est aujourd’hui impossible de disposer facilement de ces dites matrices pour les réseaux d’Internet ; ce pour des raisons de lourdeur d’implémentation, mais également pour des raisons stratégiques car les différents possesseurs de réseaux privés (Fournisseurs d’Accès à Internet, Réseaux Telecom ...) protègent farouchement les statistiques de leurs infrastructures.

Les matrices de trafic constituent une donnée de statistique très intéressante puisqu’elles permettent de simuler des algorithmes de routage, de prévoir la charge des réseaux et donc de savoir comment mieux structurer les réseaux. Elles permettent finalement d’optimiser les couts de déploiement en anticipant les mouvements de flux. On comprend alors aisément l’atout non négligeable de disposer de cette donnée statistique pour un administrateur de réseaux.

Dans ce projet, il est donc question de rechercher et d’évaluer différentes distributions probabilistes pour estimer les matrices de trafic, mais une grande partie du projet a consisté en un travail préparatoire conséquent.

A l’heure actuelle, les chercheurs ne disposent que de très peu de données pour travailler. Il n’y a en fait que 2 jeux de données rendues publiques pour le monde de la recherche au moment de la rédaction de ce mémoire. L’un est un ensemble de matrices de trafic mesurées sur le réseau européen (dorsale Internet) GEANT pendant 4 mois en 2005, et l’autre est un ensemble de matrices de trafic mesurées sur le feu réseau américain ABILENE pendant 24 semaines).

L’objet de ce mémoire est donc, après avoir expliqué comment les données à disposition des chercheurs ont été extraites et reformatées, de trouver des méthodes d’estimation de ces matrices de trafic efficaces et performantes.
4. Introduction

I was able to benefit from great conditions in the accomplishment of this project, thanks to a productive environment, and ideal conditions of study.

After attending to my Master’s classes for a semester, I wanted to tackle a new opportunity to use my knowledge acquired both at ESIEA and IIT. Doing a research project was the best challenge I could hope for and when Dr E. Chlebus offered me to work with him, I was honored to accept it.

In this introduction I will introduce the Illinois Institute of Technology: aside from being the host institution for my project, it is the University I have been a student in for now a year and a half. This has been a place where I gained an invaluable experience and met incredible people. I will then trace back the history of this project, explaining where it comes from, why I chose it, what the problem to solve is, and why it is important to study it.
4.1. My host Institution: Illinois Institute of Technology

4.1.1. History and Location

**Illinois Institute of Technology** is a University based in Chicago, Illinois, that offers a wide variety of specialties, including some renowned programs such as Architecture, Electrical Engineering or Computer Science.

The institution was formed in 1940 by the merger of Armour Institute of Technology and Lewis Institute that were both founded in the 1890s.

The campus offers a rare combination of a unique studying work environment and the proximity of the beautiful city of Chicago (when it is not freezing cold!).

*IIT* has some brilliant alumni, such as Marvin Camras, one of the fathers of the magnetic tape, Dr. Martin Cooper, the inventor of the mobile phone, or Susan Solomon, who shared the Nobel Peace Prize as a member of IPCC\(^4\) with Al Gore in 2007 for her work on man-made climate change.

---

\(^3\) Image Credit: Lawrence Biemiller - http://www.iceandcoal.org/

\(^4\) IPCC is the acronym for *Intergovernmental Panel on Climate Change* - http://www.ipcc.ch/

4.1.2. **Organization**

Before describing my environment of study, we shall have a quick overview of IIT organization: it is a rather small university as the institution only unrolls around 7,700 students spread over 5 campuses in the Chicago area.

The university is divided in 8 units, composed of colleges and institutes, each supervised by a Dean. Some of IIT’s most renowned colleges are the *College of Science & Letters* where I was studying, the *Armour College of Engineering*, and the famous *College of Architecture* from which many students and professors participated in the development of the unique and beautiful Chicago skyline.

Each of these colleges is then subdivided in departments. The people at the head of each department are called the Chairs. The organization of IIT being relatively complex, I will just detail the structure of my unit, the *College of Science & Letters (CSL)*. The CSL is supervised by Dean Dr. R. Russel Betts and consists of 6 departments: *Applied Mathematics*; Biological, Chemical & Physical Sciences; *Computer Science*; *Humanities*; *Mathematics & Science Education*; *Social Sciences*. Each of these departments is supervised by a chair. The chair for the department of Computer Science which was the academic department in charge of my Master’s is Dr Xian-He Sun.

A comprehensive organizational chart of IIT academic units is shown in the next page.

---

6 Image Credit: Lawrence Biemiller - http://www.iceandcoal.org/
7 Mies van der Rohe, one of the Hall of Famers’ professor from IIT College of Architecture is widely regarded as one of the pioneering masters of Modern Architecture, (see http://www.iit.edu/giving/mies/about_mies/influence/)
**IIT Organizational chart of the academic offices:**

![IIT Academic Organizational Chart](image_url)

*Figure 1 - IIT Academic Organizational Chart*
4.1.3. Placement and choice of the project

For the year of 2009-2010, I was enrolled in the department of Computer Science at IIT to complete a Master’s of Science in Computer Science.

The partnership between ESIEA and IIT offers a unique opportunity to complete our “end of studies” internship under the authority of the host university IIT, as a Research Project or an internship.

At the end of the semester of fall 2009, I felt motivated to work on a research project and I decided to go meet with some of my professors to ask them if they were interested in working with me.

I first met with my professor of Algorithms, Dr Gruia Calinescu who proposed me to work on multi-path routing algorithms, which I didn’t really feel comfortable with. I then met with my professor and now project supervisor Dr Edward Chlebus, who is also the team head of the research lab NEMTAK\(^8\) on network modelization. He offered me to work on this project I am reporting on now.

I decided to choose this project because it dealt with some domains I wasn’t very familiar with and felt the need of studying deeper, precisely statistics and estimation techniques; and because of the challenging aspect of the task.

\(^8\) NEMTAL is the acronym for Network Modeling and Teletraffic Analysis Lab - http://www.cs.iit.edu/~nemtal/
4.2. Genesis of the project

During our first meeting, Mr. CHLEBUS gave me some literature to read and told me to come back to him if I was interested in the subject. I had a couple research papers to go over, treating about Internet Traffic modelization and more precisely Traffic Matrices. After I told my professor I was interested in the project, he explained me that the goal of the project is to find accurate models for estimating Internet Traffic Matrices.

4.2.1. Definition of a Traffic Matrix

Let $A$ be a square matrix. If $T$ is a Traffic Matrix (TM), then $T$ provides, for every ingress point $i$ into a network and egress point $j$ out of the network, the volume of traffic $T_{i,j}$ that transited from $i$ to $j$ over a given time interval.

As a picture is often worth a thousand words, let’s take a visual example. We consider the following network with 4 interconnected routers:

![Network Topology Diagram](lovelycharts.com)

Figure 2 - Example of a network topology
Let’s assume that the time interval considered is 10 minutes, and that the unit for the traffic volume is the mbps. Let $A = (a_{i,j})_{4x4}$ be the Traffic Matrix of this network for a given interval with:

$$
A = \begin{pmatrix}
5 & 4 & 11 & 1 \\
7 & 15 & 2 & 9 \\
10 & 10 & 4 & 9 \\
15 & 2 & 1 & 8 \\
\end{pmatrix}
$$

This means that during the 10 minutes time interval when $A$ was measured, the amount of traffic that entered the network from router R1 and left the network at router R2 was 4mbps; the amount of traffic that entered the network from router R3 and left the network at router R4 was 9mbps; and so on ...

Notice also that the total traffic $T$ that transited within the network during the 10 minute interval is $T = \sum_{1 \leq i,j \leq n} a_{i,j}$ (in that case $T=113mbps$).

This whole memoir is going to deal with Traffic Matrices (TMs), so to make sure the notion is clear. I will complete the definition with some properties on the TMs. The following presents 3 observations about TMs.
1) **A TM is not necessarily symmetric.**

This property is easy to see in the example above, and it is fairly trivial to understand the reasons of it, without getting too technical.

Let’s go back to our previous network topology and assume the following additions: a personal computer is connected to R1 and a web server is connected to R2 as follow:

![Network Topology Diagram](image)

Figure 3 - Example of a network topology, continued

If the person working on the personal computer connected to R1 requests a page that is hosted on the web server connected to R2, it is obvious that the exchange of data is going to be asymmetric. A web request has an order of magnitude measured in bytes (a request message contains a request line with the address of the page requested and a few more information) while the page served in return will have an order of magnitude measured in kB or even MB if the page is rich in multimedia content (images, videos ...). In that case we will have for example $a_{1,2}=10$ bytes (the PC request a page to the web server on R2) and $a_{2,1}=30$kB (the web server on R2 serves the page to the PC on R1) and thus $a_{2,1} >>> a_{1,2}$. 
2) The diagonal elements of a TM are generally non zero

Again this property will be a lot easier to understand with an example. Let’s now assume that 3 computers are connected to R1:

![Network Topology Diagram](lovelycharts.com)

If PC1 needs to download a file from PC2, the information is going to transit through R1, therefore if we zoom out to the scale of the whole network, the traffic is going to enter the network from R1 (request from PC1) and leave the network also at R1 (request will be forwarded to PC2). Let’s also notice that the eventual asymmetric nature of the data exchange will be hidden in the value of $a_{1,1}$ as we have no information on the volume of data that traveled from PC1 to PC2 and the volume that travelled the opposite way.

A case where a diagonal element would be zero is if there was only one device connected to a router, thus there would be no internal traffic.

3) The nodes of a TM are not necessarily routers

A TM can be of various natures, as observed in (Alderson, et al. 2006):

“We can study traffic matrices at various granularities: e.g., computer to computer, router to router, or PoP (for Point-of-Presence) to PoP, and from varying points of focus.”

---

9 “A Point of Presence or PoP is a geographic location of equipment and interconnection to the network.” - [http://arthurdejong.org/aaa/aaaterms.html](http://arthurdejong.org/aaa/aaaterms.html)
4.2.2. Why is it important to estimate Traffic Matrices?

We have spent a lot of time on defining in detail what a TM exactly is, but now let’s explain why being able to **correctly estimate and synthesize a TM is important**. To a further extent, this also explains the purpose and finality of doing such research project.

In (Gunnar, Johansson and Telkamp 2004) and (Medina, et al. 2002), different arguments are presented that underline the importance of Traffic Matrices in designing networks and maintaining them efficiently. TMs are indeed a **key statistic for many traffic engineering tasks**, because knowing the size and locality of flows is essential to optimize the structure of a network, to choose the routing algorithms, or to handle failure strategies. It can enable a network administrator to anticipate network growth and links congestion.

All these arguments put together, we can understand why TMs are so crucial, because being able to compute all the above statistics from traffic matrices can ultimately enable network engineers to **estimate costs** and **provide better QoS**\(^\text{10}\).

4.2.3. Obstacles and issues in computing Traffic Matrices

Unfortunately, current network devices (such as routers) offer very little statistics for network measurement. The only statistics widely available are the link-loads (resource utilization at network nodes) and routing configurations. It is thus impossible to compute traffic matrices directly. In addition, such traffic data is very complicated to obtain as most ISPs\(^\text{11}\) and other ASs\(^\text{12}\) keep their statistic and logs information private for strategic regards.

---

10 QoS is the acronym for **Quality of Service**: “the idea that transmission rates, error rates and other characteristics can be measured, improved and, to some extent, guaranteed in advance.” - [http://www.frso.org/index.php?option=com_content&view=article&id=1726](http://www.frso.org/index.php?option=com_content&view=article&id=1726)

11 ISP (or IAP) is the acronym for **Internet Service (Access) Provider**: a company that provides internet access for its customers.

12 AS is the acronym for **Autonomous System**: “a network (or group of networks) that is under the control of a common administrator (or group of administrators) on the behalf of ONE entity (like a school or business)” - [http://searchnetworking.techtarget.com/definition/autonomous-system](http://searchnetworking.techtarget.com/definition/autonomous-system)
5. State of the art

Very little research has been done on this matter to this date. Indeed all the literature I have been able to find and study has been published during the last decade. I see two reasons that can explain this fact:

- Internet massive expansion is still very new, its growth became exponential in the late 90s, no later than 15 years ago. The chart below shows the evolution of Internet Hosts\(^\text{13}\) since 1981:

![Internet hosts 1981 - 2009](image)

\(^{13}\) A host computer is a computer that has full two-way access to other computers on the Internet. A host computer has a unique IP address.

- As we will see in the next paragraph, the quantity of available data researchers are able to access is ridiculously low.
5.1. Data available for study

At the time I am writing this memoir, only 2 sets of Traffic Matrices are publicly available to the research community.

The first set has been computed and measured by Dr Yin Zhang in 2004 and offers traffic matrices measured every 5 minutes for 6 months (not continuously, many days are missing) on the Abilene Network. The Abilene Network was a high performance backbone network\(^{14}\) that connected mostly universities in the United States.

The following image depicts the topology of this network:

![Abilene Network Topology](http://www.internet2.edu/2004AR/abilene_map_large.cfm)


---

\(^{14}\) A backbone network is basically a high capacity network of lower capacities networks. Wikipedia contributors define it as “a part of computer network infrastructure that interconnects various pieces of network, providing a path for the exchange of information between different LANs or sub-networks.”
The second dataset has been made available in the research paper “Providing Public Intradomain Traffic Matrices to the Research Community” (Uhlig, et al. 2006). This paper and its associated study publish 4 months of Traffic Matrices, measured every 15 minutes between January 1st, 2005 and April 29th, 2005 on the GEANT network (dates and topology have been anonymized). The GEANT network is a pan-European high-bandwidth backbone network that interconnects National Research and Education networks across Europe. The topology of this network is given in the image below:

![Topology of GEANT network](http://www.geant.net/Network/NetworkTopology)

The GEANT network traffic matrices are available at [http://totem.info.ucl.ac.be/dataset.html](http://totem.info.ucl.ac.be/dataset.html).

It seems like a third dataset from the Sprint backbone has been computed but is not publicly available. The research paper “The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations” (Nucci, Sridharan and Taft 2005) uses Sprint data for study. I have been searching for this dataset without success.
5.2. Existing Techniques

The literature provides a good amount of methods to generate and estimate traffic matrices. An early research publication already listed a partial state of the art in 2002: “Traffic Matrix Estimation: Existing Techniques and New Directions” (Medina, et al., 2002). Let’s review the major contributions to TMs estimation to date.

5.2.1. Parametric models

In the article “The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations” (Nucci, Sridharan and Taft 2005), the authors widely discuss the problem of fitting the real data to generic distributions such as the lognormal distribution, the loglogistic distribution, and the inverse Gaussian distribution.

An earlier paper already dealt with the matter, only focusing on the lognormal distribution as a tool of comparison with another model that we discuss in the next paragraph. This paper is titled “Simplifying the synthesis of Internet Traffic Matrices” (Roughan 2005).

5.2.2. Non-parametric models

A common model generally accepted in Traffic Matrix estimation is the Gravity Model. This model is discussed in the papers “Simplifying the synthesis of Internet Traffic Matrices” (Roughan 2005) and “Traffic Matrix Estimation on a Large IP Backbone – A Comparison on Real Data” (Gunnar, Johansson and Telkamp 2004).

It is improved to a Generalized Gravity Model in “Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads” (Zhang, Roughan, et al. 2003). This last technique combines the Gravity Model approach with the use of widely available network statistics (link loads and network topology).
6. Technical aspects and progress

6.1. Understanding the problematic and defining a working plan

The first and not least challenge of this project was the understanding of the problematic itself. When Mr. Chlebus first introduced me to the subject I first felt somewhat lost as it is something I had never dealt with before. Both ESIEA and IIT provide a detailed and well designed class on Network Fundamentals (INF4032 – Réseaux Informatiques and CS542 – Computer Networks I: Fundamentals) as well as a Statistics / Estimation / Modelization Class (MAT4056 - Estimation and CS584 – Machine Learning). However, the problem here was to put 2 and 2 together and apply statistics techniques to Network theory.

I believed I could achieve this goal by reading over and over the literature on both topics until I felt comfortable enough. I spent a few weeks, only dedicating to this task. In addition, the beginning of this project was quite hectic because I also had to deal with the Spring Semester 2010 from which I was enrolled in 4 Master’s classes.

After doing all the above prep work, I still wasn’t 100% understanding the whys and wherefores of the project but I knew I couldn’t advance any further without getting my hands dirty.

In consequence, I decided to build a working plan in order to efficiently solve my Traffic Matrix estimation problem. The working plan is presented in the next page.
Working plan

<table>
<thead>
<tr>
<th>Task definition</th>
<th>Difficulty</th>
<th>Time Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtain ABILENE and GEANT Traffic Matrices datasets</td>
<td>Easy</td>
<td>1 day</td>
</tr>
<tr>
<td>Get familiar with the datasets and understand their formatting</td>
<td>Easy</td>
<td>1 week</td>
</tr>
<tr>
<td>Unify the format of both datasets (so that a future program can handle both sets with no additional work)</td>
<td>Easy</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Build a toolbox that can manipulate traffic matrices on the fly and execute a various set of tasks</td>
<td>Medium</td>
<td>1-2 months</td>
</tr>
<tr>
<td>Work on the core problem: TM estimation - parametric models</td>
<td>Hard</td>
<td>?</td>
</tr>
<tr>
<td>Work on the core problem: TM estimation - Gravity Model</td>
<td>Hard</td>
<td>?</td>
</tr>
<tr>
<td>Look for new directions</td>
<td>Hard / Challenging</td>
<td>? - if extra-time</td>
</tr>
<tr>
<td>Work on my memoir</td>
<td>Medium</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Prepare presentation</td>
<td>Medium</td>
<td>2 days</td>
</tr>
</tbody>
</table>

Note: I wasn’t able to estimate how long I would need to work on the core problem at the time I wrote this working plan because I was still in the process of getting familiar with the topic. However, I knew with a simple subtraction that I would have around 3 months to work on these tasks.

6.2. Extraction and manipulation of the raw data

The ABILENE dataset contains 48,384 5-minute traffic matrices measured from March 1st, 2004 to September 10th, 2004. These TMs are presented on 24 different files where each file contains 2016 TMs worth one week of measurement. The formatting of each file is quite complicated and contains more data than the TMs alone.

On the other hand, the GEANT dataset contains 10,772 15-minute traffic matrices measured from January 1st, 2005 to April 29th, 2005. These TMs are presented as 10,772
different files where each file is named after the date and time of measurement and contains only one TM in XML\textsuperscript{15} format.

The two datasets previously described raise two issues:

- The amount of data to be manipulated is very large and the design of a tool to manipulate the data easily and on the fly is mandatory. This will be solved by creating an efficient GUI\textsuperscript{16}. This GUI will enable me to easily extract and plot one TM, or compute its distribution ...

- The format of each dataset is completely different and it is impossible to efficiently manipulate the data without defining a common format for both datasets.

To sum up, before thinking of modelizing the data and estimating the TMs, we need an efficient tool to automatize the batch treatment of our data.

### 6.2.1. Choice of a programming language and environment

To implement my “data manipulation toolbox” and unify the formatting of both network datasets. I had to design and write 2 different programs. A very large choice of programming languages is available nowadays and I am already familiar with many of them. Even if this step of my project wasn’t the most important, I decided to sacrifice a little time on it by choosing a programming language I had never used before. My purpose was to broaden my programming skills and languages variety.

As I have also been preparing my future and looking for opportunities and job offerings throughout this project, I have realized that Microsoft .NET framework is a highly demanded skill on today’s job market.

As a future engineer, we have a duty of keeping ourselves up-to-date with the latest technologies by constantly self-learning new approaches and techniques. As a consequence, I decided to use the .NET framework for this first part of the project.

\textsuperscript{15} XML is the acronym for Extensible Markup Language: “a flexible text format for creating structured computer documents” - [http://en.wiktionary.org/wiki/xml](http://en.wiktionary.org/wiki/xml)

\textsuperscript{16} GUI is the acronym for Graphical User Interface: a way for humans to interact with computers (using windows, buttons, forms, list boxes ...)

Quick Presentation of Microsoft .NET framework

.NET (pronounced “dot NET”) is the framework created by Microsoft (it is not a programming language!) that enables developers to create hassle-free applications in a unified environment. It is designed to greatly simplify the developers’ work and allow them to gain in productivity by going faster on redundant tasks where silly mistakes are easy to make and long to debug.

.NET framework consists in a set of libraries for developers to be used in any of the compatible languages. Developers have thus the choice in the language they want to use. The most popular languages are VB.NET (very close to initial Visual Basic) and C# (Very close to C++). ASP.NET is another widely used language that is more web-oriented.

Again, developing in any of these languages is strictly equivalent because they all have access to the same .NET libraries. This advantage is the result of the way the framework structure has been designed. As we see in figure 8, Microsoft has produced a compiler for each .NET supported language. Theses compilers will transform the code from its original language (C#, VB.NET …) to a Common Intermediate Language (CIL).

Once the program is compiled to CIL, the project can be easily built. Notice that virtually any programming language can be compatible with .NET providing Microsoft (or the open source community) has written a compiler to transform the source code into CIL. Many projects are indeed in process to make different languages “.NET friendly”, such as PHP (php4mona project), JavaScript (Jscript .NET project), and many more.

17 A non exhaustive list of .NET supported languages is available here: http://en.wikipedia.org/wiki/List_of_CLI_languages
Figure 8 - Overview of the connect language infrastructure (Source: http://en.wikipedia.org/wiki/.NET_Framework)
6.2.2. Unifying both datasets to the same format

Each dataset has been measured by different contributors, on different networks, and at different dates, which unfortunately implies that the file structure and formatting on each set is completely different. In order to efficiently manipulate the data and simplify the programming of my toolbox I decided that it would be much more efficient to only have one “file formatting” style for the data extracted from both networks.

After analyzing the structure of both ABILENE and GEANT files, I would then pick the best option to use for my experiments: whether choosing one of the 2 existing structures, or by creating my own. Let’s now review these two steps.

6.2.2.1. Analysis of each dataset file structure

The ABILENE data formatting is rather confusing and unnatural. Indeed each file contains 2016 5-minute traffic matrices. And on top of these traffic matrices the files contain other computed data, not directly measured from the network. In the end, for each traffic matrix, we are given the real Origin/Destination (OD) values of the flows, and 4 other values computed using different models (Gravity model, general gravity model ...), all mixed up together. ABILENE formatting is definitely not a keeper.

GEANT data formatting uses XML standard and is much better organized. Each file consists in only one TM, and is named as follows: “IntraTM-YYYY-MM-DD-HH-mm.xml”. For example “IntraTM-2004-11-23-14-45.xml” contains the TM measured on November 23rd, 2004 at 2:45pm. The file consists of a list of OD flows as shown in figure 9.

As we can see in figure 9, flows (in kbps) are classified by source (or origin) in the tag “<src>...</src>” and destination in the tag “<dst>...</dst>”. Each file contains a header marked by the tags “<info></info>” containing a title, the date and time of measurement, the duration in seconds, the author, and the unit.

For example if we want to get the OD flow {12; 8}, i.e. the amount of traffic entering the network at node 12 and exiting at node 18 we first find the tag “<src id="12">... </src>”. Within this tag we now look for the tag “<dst id="8">... </dst>” and we read the value. Let T be the name of that matrix, in that case we would have \( T(12; 8) = 6552.9333 \) (see figure 9).
After making all the previous observations, I decided that my best option was to transform the ABILENE files to make it to the same format as the GEANT files. There are a few advantages that oriented my choice:

- By choosing GEANT formatting and not creating my own file structure, I only need to modify one dataset (ABILENE), which saves time and avoid an additional source of error (by corrupting the GEANT dataset).
- The XML format is a very common standard, widely use in modern applications. As a matter of fact, the .NET framework offers a library to easily manipulate and create XML documents.

Writing the code to transform the ABILENE files into XML took me a few hours (see *Appendix A*). It consisted in separating the 2016 traffic matrices in each file into 2016 XML files containing one traffic matrix, making sure I only kept the real OD flows for each TM, and got rid of useless information. Although the formatting script was produced in little time, I took a couple days to make sure the script wasn’t bogus and the TMs were not mistaken. This verification step was absolutely necessary and critical since all the later research would be based on these files.
6.2.3. Building a toolbox for batch data manipulation

Dealing with thousands of TMs for each network is impossible without building tools to efficiently manipulate and extract information automatically. The creation of such toolbox was thus crucial before I could start studying and estimating the TMs.

6.2.3.1. Requirements Elicitations

I spent a few days listing all the requirements of the toolbox and designing the structure of my application. Below is the UML Use Case Diagram that contains all the requirements.

![Figure 10 - Use Case Diagram for TM Toolbox project](image-url)
6.2.3.2. Structure of the program

One of the most accepted good practice in programming is to isolate functionalities in different modules in order to split difficult problems into simpler problems. This method is often referred as “divide and conquer”.

From the analysis of the requirements, I isolated 3 aspects that my software would need to deal with:

- File manipulation (dealing with the File system, reading and writing in files)
- Data manipulation (computing distributions, values ...)
- Graphic User Interface (I also could have made my program as a Console application meaning that each option would have been accessible with a command line, but for very limited extra time I could make a neat user friendly interface)

This list is pretty much the basis of my program structure, the class diagram I designed is entirely inspired from it (see figure 11).

It is composed of 3 classes:

- The FileManager class: this class is only in charge of manipulating the files – i.e. reading/writing/file creation. This class provides data to the DataControl class that can then use it to return the appropriate information to the user. The main methods consist in:
  o Returning one traffic matrix of a certain date on a certain network (ABILENE or GEANT)
  o Returning the computed traffic matrix of the average values of each OD destination on a time interval (more on that in paragraph 6.4.1.)
  o Return the total traffic of a TM from a certain date / time interval
  o Return the list of values of a particular OD flow on a certain time interval
- The DataControl class: this class is in charge of manipulating the data returned by the FileManager class in order to compute a Cumulative Distribution Function, sums up the OD flows of one TM to get the total traffic ... It basically uses the raw data passed by the FileManager class to make it understandable by the user.
- The GUI class: this class, almost entirely generated automatically with Microsoft Visual Studio, allows the user to manipulate and extract the information he needs in a few clicks.
Figure 11 - Class Diagram for TM Toolbox project
6.3. Analysis of the data and research of better models

This section describes the core aspects of this research project. Although the prep work described in 6.1 and 6.2 took me a couple months, it only consisted in designing a program and implementing it. These tasks are technical but are not harder than my technical internship I completed at the end of my third year at ESIEA. The ambition I gave myself for this project is much larger than that. Analyzing the traffic matrices and researching statistical models and other methods allowed me to reach another dimension and fulfill my expectations.

Let’s review the different steps I took to analyze Internet traffic matrices.

6.3.1. Mathematical background

Most of the models I analyzed are parametric distributions. All my work makes a great use of an estimation method called Maximum Likelihood Estimation (MLE).

I studied MLE method in my class of Machine Learning at IIT and deepened the subject with the book “Simulation Modeling and Analysis” (Law and Kelton 2000) on my professor advice. Before detailing this method, we shall expose a few definition and notations.
6.3.1.1. Definitions and notations

1) Statistical Hypothesis testing

Modelizing an observed phenomenon is always based on the assumption that the measured data follows a certain model. We call this process Statistical Hypothesis testing.

Statistical Hypothesis testing consists in the following process:

- Formulation of a hypothesis that the data is distributed (in our case) according to a certain law $L$. Let $H_0$ be the hypothesis that the data is distributed according to $L$. We state the alternative hypothesis $H_1$ that the data doesn’t follow the distribution $L$.
- Decide which test is appropriate and choose a relevant test statistic $T$
- Derive the distribution of the test statistic $T$ from the assumptions that the data follows the distribution law $L$
- Compute $t_{obs}$ from the test statistic $T$ using the data sample
- Decide to either fail to reject $H_0$ (“fail to reject” $H_0$ is slightly different from “accepting” $H_0$ as there is always a non null probability of error) or to reject $H_0$ in favor of $H_1$.

In my case, Hypothesis testing will consist in:

- Assuming the traffic matrix OD flows follow a distribution $D$ with parameters \( \theta = \{ \theta_1, \theta_2, \ldots, \theta_n \} \). Then $H_0 = \{ \text{the distribution of the elements of a traffic matrices follows } D(X|\theta) \text{ where } X \text{ is a random variable} \}$, and $H_1 = \{ \text{the distribution of the elements of a traffic matrices does not follow } D(X|\theta) \text{ X is a random variable } \}$
- Deriving the formula of the likelihood (joint probability that all the observations \( (X_i)_{1}^{n} \) are issued from $D(X|\theta)$) to compute $\theta$ using Maximum Likelihood Estimation method.
- Computing $\theta$ using the formula of the previous step on the training data.
- Decide whether to “accept” (fail to reject) $H_0$ or reject it by testing my theoretical distribution versus the empirical distribution\(^{18}\).

\(^{18}\) Empirical distribution function is the distribution associated with the empirical measure of the traffic matrices
2) Notations
For the rest of this report, we will focus on how the OD flows of a traffic matrix are distributed, that is how the elements of a traffic matrix are distributed.

Let \( A \) be the following traffic matrix:
\[
A = (a_{i,j}) = \begin{pmatrix}
a_{1,1} & a_{1,2} & a_{1,3} \\
a_{2,1} & a_{2,2} & a_{2,3} \\
a_{3,1} & a_{3,2} & a_{3,3}
\end{pmatrix}
\]
Since we don’t need spatial positions of the \( a_{i,j} \) to study the distribution of the elements, we reformat \( A \) to \( X \):
\[
X = \{ a_{1,1} ; a_{1,2} ; a_{1,3} ; a_{2,1} ; a_{2,2} ; a_{2,3} ; a_{3,1} ; a_{3,2} ; a_{3,3} \} = (X_i)_{1 \leq i \leq 3 \times 3}
\]
Every \( X_i \) is an empirical example. Note that our examples have only one feature (one dimension): the value in kbps of the flow that transited during a period of time (15 minutes for GEANT, 5 minutes for ABILENE).

3) Maximum Likelihood Estimation
Wikipedia defines MLE as “popular statistical method used for **fitting a statistical model to data**, and providing estimates for the model’s parameters.”

As we described before, let’s suppose we have a sample \( X \) of \( n \) observations \( X_1, X_2, ..., X_n \). and let’s supposed that \( X \) is distributed by the parametric model \( f \) of parameter \( \theta \).

The joint density function, i.e. the probability that all the observations \( X_i \) are issued from \( f \), is:
\[
f(X_1, X_2, ..., X_n | \theta) = f(X_1 | \theta) \cdot f(X_2 | \theta) \cdot ... \cdot f(X_n | \theta)
\]
Now if we change the point of view and consider our observation values \( X_i \) as fixed parameters (training data) and the parameter \( \theta \) to be estimated as a variable, we get the function called the **likelihood** \( L \):
\[
L(\theta | X_1, X_2, ..., X_n) = f(X_1, X_2, ..., X_n | \theta) = \prod_{i=1}^{n} f(X_i | \theta)
\]
The purpose here is to find $\theta_{\text{MLE}}$ that maximizes this probability, i.e. maximizes the likelihood that all the examples from the observation are issued from the model $f$ of parameter $\theta_{\text{MLE}}$. In other words we want to find the value of $\theta$ (called $\theta_{\text{MLE}}$) that maximizes the likelihood $L$.

For ease of calculations we shall define the log-likelihood $\ln L$ in order to manipulate sums instead of products:

$$\ln(L(\theta|X_1, X_2, \ldots, X_n)) = \ln(f(X_1, X_2, \ldots, X_n|\theta)) = \sum_{i=1}^{n} \ln(f(X_1|\theta))$$

Then $\theta_{\text{MLE}}$ is:

$$\theta_{\text{MLE}} = \arg\max(\ln(L(\theta|X_1, \ldots, X_n)))$$

We finally find $\theta_{\text{MLE}}$ by deriving the value of the log-likelihood and solving the resulting system of first order conditions.

For the sake of understanding, we will apply MLE for the normal probability density function (PDF) (see next paragraph).

4) Different parametric models tested during this project

To be sure not to miss any valuable results, I have decided to test a wide number of distributions versus my empirical data (the TMs). I will now list all the probability functions I have tested and provide (when a simple form exists) the formulas of the CDF\(^{19}\), the PDF\(^{20}\), and the parameters estimated by the MLE method. The results of these models on ABILENE and GEANT TMs are left for a further paragraph. I will fully demonstrate the parameter estimation by MLE only once, for the Normal distribution.

**Note:** most the CDFs, PDFs, and parameters are given in my book (Law and Kelton 2000), but I copied most of the formulas from Wikipedia for the sake of simplicity.

---

\(^{19}\) CDF is the acronym for **Cumulative Density Function**, it describes “the probability that a real-valued random variable $X$ with a given probability distribution will be found at a value less than or equal to $x$."


\(^{20}\) PDF is the acronym for **Probability Density Function**, it describes “the relative likelihood for a random variable to occur at a given point." [http://en.wikipedia.org/wiki/Probability_density_function](http://en.wikipedia.org/wiki/Probability_density_function)
**Normal distribution:**

- **PDF:** 
  \[ f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]
  where \( \mu \) is the mean and \( \sigma^2 \) is the variance (measure of the width of the distribution)

- **CDF:** no simple form

- **Estimation of parameters by MLE:**
  \[ \hat{\mu} = \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i, \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2. \]

**Proof:**

We first take the log-likelihood function:

\[ l(\mu, \theta^2) = \sum_{i=1}^{n} \ln f(x_i; \mu, \theta^2) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2 \]

We now maximize \( l(\mu, \theta^2) \) by solving the system:

\[ \nabla(l(\mu, \theta^2)) = \begin{bmatrix} \frac{\partial l(\mu, \theta^2)}{\partial \mu} \\ \frac{\partial l(\mu, \theta^2)}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

- Let’s solve the first equation:

\[ \frac{\partial l(\mu, \theta^2)}{\partial \mu} = 0 \]

\[ \frac{\partial (-\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2)}{\partial \mu} = 0 \]

\[ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (-2(x_i - \mu^*)) = 0 \]

\[ \sum_{i=1}^{n} (x_i - \mu^*) = 0 \]

(by multiplying both sides by \( \sigma^2 \) and simplifying the fraction)

... Continuing on next page ...
\[
\sum_{i=1}^{n} (x_i) - n\mu^* = 0
\]

\[
\mu^* = \frac{\sum_{i=1}^{n} x_i}{n}
\]

Let’s solve the second equation:

\[
\frac{\partial l(\mu, \sigma^2)}{\partial \theta} = 0
\]

\[-n \frac{2\sigma}{2 \sigma^2} - \sum_{i=1}^{n} (x_i - \mu)^2 \left(\frac{-4\sigma}{4\sigma^4}\right) = 0
\]

\[-n + \sum_{i=1}^{n} (x_i - \mu)^2 \left(\frac{1}{\sigma^3}\right) = 0
\]

\[-n + \frac{1}{\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2 = 0
\]

(by multiplying both sides per \(\sigma\))

\[
\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}
\]

Q.E.D.

**Lognormal distribution:**

\[
f_X(x; \mu, \sigma) = \frac{1}{x\sigma \sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, \quad x > 0
\]

- **PDF:** where \(\mu\) is the mean and \(\sigma^2\) is the variance (measure of the width of the distribution)
- **CDF:** no simple form
- **Estimation of parameters by MLE:**

\[
\hat{\mu} = \frac{1}{n} \sum_{k=1}^{n} \ln x_k, \quad \hat{\sigma^2} = \frac{1}{n} \sum_{k=1}^{n} (\ln x_k - \hat{\mu})^2.
\]
Weibull distribution:

- **PDF:**
  \[
  f(x;\lambda, k) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0, \end{cases}
  \]
  Where \( k > 0 \) is the shape parameter and \( \lambda > 0 \) is the scale parameter of the distribution.
- **CDF:**
  \[ F(x; k, \lambda) = 1 - e^{-(x/\lambda)^k} \]
- **Estimation of parameters by MLE:** no simple form

Gamma distribution:

- **PDF:**
  \[
  f(x; k, \theta) = x^{k-1} e^{-x/\theta} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)} \text{ for } x \geq 0 \text{ and } k, \theta > 0.
  \]
  Where \( k > 0 \) is the shape parameter and \( \theta > 0 \) is the scale parameter of the distribution, and the gamma function is for any \( n \):
  \[ \Gamma(n) = (n - 1)! \]
- **CDF:**
  \[ F(x; k, \theta) = 1 - \sum_{i=0}^{k-1} \frac{(x/\theta)^i}{i!} e^{-x/\theta} = \sum_{i=k}^{\infty} \frac{(x/\theta)^i}{i!} e^{-x/\theta} \]
- **Estimation of parameters by MLE:**
  \[
  \hat{\theta} = \frac{1}{kN} \sum_{i=1}^{N} x_i. \\
  k \approx \frac{3 - s + \sqrt{(s - 3)^2 + 24s}}{12s} \quad \text{where} \quad s = \ln \left( \frac{1}{N} \sum_{i=1}^{N} x_i \right) - \frac{1}{N} \sum_{i=1}^{N} \ln(x_i). 
  \]
**Exponential distribution:**

- **PDF:**
  \[
  f(x; \lambda) = \begin{cases} 
  \lambda e^{-\lambda x}, & x \geq 0, \\
  0, & x < 0 \end{cases}
  \]
  where \( \lambda > 0 \) is the parameter of the distribution.

- **CDF:**
  \[
  F(x; \lambda) = \begin{cases} 
  1 - e^{-\lambda x}, & x \geq 0, \\
  0, & x < 0 \end{cases}
  \]

- **Estimation of parameters by MLE:**
  \[
  \hat{\lambda} = \frac{1}{\bar{x}}, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i
  \]

**Beta distribution:**

- **PDF:**
  \[
  \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} f(x; \lambda) = \begin{cases} 
  \lambda e^{-\lambda x}, & x \geq 0, \\
  0, & x < 0 \end{cases}
  \]
  where \( \alpha \) and \( \beta \) are 2 shape parameters of the distribution,
  \[
  B(\alpha, \beta) = \int_{0}^{1} x^{\alpha-1}(1-x)^{\beta-1} dx
  \]
  and \( B(\alpha, \beta) \) is the Beta function with

- **CDF:**
  \[
  F(x; \alpha, \beta) = \frac{B_x(\alpha, \beta)}{B(\alpha, \beta)} = I_x(\alpha, \beta)
  \]

- **Estimation of parameters by MLE:**
  \[
  \alpha = \bar{x} \left( \frac{\bar{x}(1-\bar{x})}{v} \right), \quad \beta = (1-\bar{x}) \left( \frac{\bar{x}(1-\bar{x})}{v} \right),
  \]
  \[
  \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad v = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2
  \]
  Where
Pareto and Truncated Pareto distribution

For these 2 distributions, I designed my algorithms (PDF, CDF, and parameter estimation algorithm) by using the formulas from the following publications:

- “Favorable estimators for fitting Pareto models: A study using goodness-of-fit measures with actual data” (Brazauskas and Serfling 2004)
- “Parameter Estimation for the Truncated Pareto Distribution” (Aban, Meerschaert and Panorska 2006)

The algorithm I designed for fitting Truncated Pareto distribution and computing the CDF can be found in Appendix C (Matlab language).

Chlebus distribution

This distribution, that I named Chlebus distribution as the name of my professor and mentor that helped me all along this project, is a probability distribution that he designed and improved in the past years.

The CDF, PDF, and parameter estimation of this distribution can be found in his research paper (see appendix B): “A Versatile Probability Distribution for Light and Heavy Tails of Web File Sizes” (Chlebus and Divgi, A Versatile Probability Distribution for Light and Heavy Tails of Web File Sizes 2009).

The algorithm I designed for fitting Chlebus distribution and computing the CDF can be found in Appendix D (Matlab language).
5) Goodness of fit tests

After fitting my empirical data extracted from TMs files to the probability distributions I defined before, the problem was to determine if I could assume that a distribution fits or not my empirical data (accept or reject H0).

In other words, I tried to prove that the distribution of OD flows in an internet traffic matrix is issued from a given probability distribution.

To do so, we have 2 options:

- We plot the empirical data issued from the measured TMs versus the synthetic data generated from a given distribution that we estimated the parameters, and we use our eyes to see if the 2 lines match or not. This method is obviously quite random and highly depends on the settings of the plot, e.g. sometimes 2 lines might seem very close one another, but if we change the scale of the graph, we see that the match is not that good.

- To avoid a wrong judgment, we thus use some statistical tests (called “goodness of fit” tests) designed for helping us in accepting or rejecting a hypothesis.

I decided, on the advice of my professor, to use two different “goodness of fit” tests. Each test has its advantages and weaknesses, which explain the need of using both.

The K-S Test

The K-S Test (named after their inventors Kolmogorov and Smirnov) tries to determine if two datasets differ significantly. It has the advantage of being non-parametric (independent of the distribution to be tested).

The K-S Test involves the CDFs of both datasets to be compared, in our case we will compare the empirical CDF (CDF measured from our TM files) with the CDF of a distribution we want to test. The test quantifies the maximum distance D between the 2 sets.
The following formula gives D:

- Let the empirical distribution of the data be
  \[ F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i \leq x} \]
  where \( I_{X_i \leq x} \) is the indicator function, equal to 1 if \( X_i \leq x \) and equal to 0 otherwise
- Let \( F(x) \) be the CDF of the given distribution to be tested.
- Then the K-S statistic is:
  \[ D_n = \sup_x |F_n(x) - F(x)| \]

To decide whether to accept or reject our null hypothesis \( H_0 \) (i.e. to decide if the empirical data is issued from the given distribution function), we then compare the K-S statistic \( D_n \) with a table of critical values. If \( D_n \) is smaller than the critical value, then we accept \( H_0 \), otherwise we reject \( H_0 \) and accept \( H_1 \).

One disadvantage in using this test is that it doesn’t perform well on the tales of a distribution as it only takes into account the global maximum of the distance between the 2 distributions.

**The A-D Test**

The A-D test (named after Theodore Wilbur Anderson and Donald A. Darling) is another “goodness of fit” test. Such as for the K-S test, it tries to determine if two datasets differ significantly.

The value of the A-D test is given by \( A^2 \) as follows:

\[ A^2 = -n - S, \]

Where

\[ S = \sum_{k=1}^{n} \frac{2k - 1}{n} \left[ \ln F(Y_k) + \ln (1 - F(Y_{n+1-k})) \right]. \]

And where \( F(Y_k) \) is the empirical distribution function of the traffic matrix.

Again the value \( A^2 \) is then compared with a critical value \( CV \) and \( H_0 \) is accepted if \( CV \) is not reached.
Note that by looking at the formulas of both K-S and A-D test, we can infer than the lower the value of the statistic is, the better.

6.4. Results of distribution fitting

After studying all the mathematical background previously described, I was able to test various probability distribution versus my empirical traffic matrices. To do so, I decided to use the software Matlab, specialized in matrix and mathematics programming. One advantage of Matlab is that most of the algorithms I needed were already implemented, which left me more time for analysis.

6.4.1. Choice of the data to analyze

One delicate task is to properly choose the way we test our traffic matrices and try to estimate them. In the paper by (Nucci, Sridharan and Taft 2005), a very smart option is proposed: instead of trying to test a probability distribution based on only one traffic matrix, they propose to average many traffic matrices of a significant period of time and use these new computed matrices as inputs for our research.

The significant period is decided to be the peak hour of the day in term of traffic volume. This is a reasonable choice because “this captures the worst case traffic matrix which is an appropriate one to consider for many traffic engineering tasks”.

In the paper, they state that the peak hour of the day for SPRINT and ABILENE network is noon to 1pm. I have reviewed this statement for ABILENE and found out that this is actually not true.
Peak hour of the day for GEANT network is 10-11pm as shown below:

Figure 12 - Evolution of the total traffic volume on the GEANT network, averaged on 21 days for each interval of 15 minutes

Peak hour of the day for ABILENE network is 7-8pm as shown below:

Figure 13 - Evolution of the total traffic volume on the ABILENE network, averaged on 21 days for each interval of 15 minutes
The 12-1pm peak hour described in (Nucci, Sridharan and Taft 2005) is indeed the least busy hour of the day on both networks!

From the previous observation, I decided to compute two mean matrices of the respective peak hour of each network, that is: for each flow, I extracted each 15 minute sample (for GEANT, 5 minute for ABILENE) from the peak hour (10-11pm for GEANT, 7-8pm for ABILENE) of each day during one week. All these samples of each OD flow were then averaged to compute one rate value for each flow; the ensemble of the mean rates for all OD flows were used to create my datasets submitted for distribution fitting.

- The mean matrix for ABILENE is computed from May 8th, 2004 to May 14th, 2004, on the 7-8pm time interval.
- The mean matrix for GEANT is computed from February 1st, 2005 to February 7th, 2005, on the 7-8pm time interval.

### 6.4.2. Distribution fitting

I have tested many different distributions, (all the ones described in 6.3.1.1.) on each dataset and it turns out that many distributions are a good candidate for traffic matrix estimation. The K-S statistics are indeed better for some distribution than the ones obtained by (Roughan 2005) in his paper “Simplifying the Synthesis of Internet Traffic Matrices”.

The next pages expose the results obtained for fitting distributions to ABILENE traffic matrix (charts and KS statistic value). The gravity model is not taken into consideration yet.
Figure 14 - Distribution fitting of ABILENE mean traffic matrix, KS test values provided on graphs
Figure 15 - Distribution fitting of ABILENE mean traffic matrix, KS test values provided on graphs -- continued
Figure 16 - Distribution fitting of ABILENE mean traffic matrix, KS test values provided on graphs -- end
**Observations and remarks on ABILENE results**

It turns out that many distributions are a good candidate to fit our ABILENE traffic matrix. (Roughan 2005) tested the lognormal distribution for his research. Although the hypothesis H0 is accepted for this distribution, we notice that it is far from being the best option. As a matter of fact, if we focus on the detail of the lognormal distribution, we can see that it is quite far from the empirical data. To better show this, we re-plot the chart comparing the lognormal distribution with the lognormal distribution using a log scale on the x axis:

![Figure 17 - comparison of the lognormal distribution with the empirical data (log-linear scale)](image)

It turns out that the best distribution for my ABILENE data is the Weibull distribution since the KS statistic $D = 0.0496$ and the AD statistic $AD = 0.5130$ (far under the critical value 2.492 to accept the test on the 95th percentile) are the lowest of all distributions.

If we plot the graph again using log scale on x axis we see a much better fitting than for lognormal distribution:

![Figure 18 - comparison of the Weibull distribution with the empirical data (log-linear scale)](image)
To review the results, let us do a short list of the distributions that were accepted for this sample of data from ABILENE (ordered by ascendant KS stat):

- Weibull distribution (KS stat = 0.0496)
- Gamma distribution (0.0648)
- Chlebus (KS stat = 0.0687)
- Lognormal distribution (KS stat = 0.114)

We shall now analyze the results of distribution fitting on our GEANT dataset (see next pages). The gravity model is not taken into consideration yet:
Figure 19 - Distribution fitting of GEANT mean traffic matrix, KS test values provided on graphs
Figure 20 - Distribution fitting of GEANT mean traffic matrix, KS test values provided on graphs -- continued
Figure 21 - Distribution fitting of GEANT mean traffic matrix, KS test values provided on graphs -- end
Observations and remarks on GEANT results

The observations on this dataset, although slightly different than from the ABILENE dataset, are consistent with the previous conclusions. Most distributions that performed well perform well again on GEANT. We also notice that lognormal is not accepted for distribution (it was already closed to be rejected previously). On the other hand we notice the flexibility of the Chlebus distribution that performs well on this test.

Let’s have an overview of the distribution that were accepted, order by ascendant KS-value:

- Chlebus (KS stat = 0.0361)
- Weibull distribution (KS stat = 0.058)
6.5. Conclusion

We found 2 good candidates for estimating internet traffic matrices: the Weibull distribution and the Chlebus distribution. Both perform very well and show adequate flexibility in term of parameters estimation.

I haven’t had time yet, but my professor improved the parameter estimation of the Chlebus distribution using moment matching, we might obtain even more promising results implementing this improvement.

- **Note 1**: for the sake of the simplicity of this report, I have only used one dataset on each network. This obviously is not enough to draw conclusions. But during this project I have been testing and validating these models on more datasets and the results are consistent.

- **Note 2**: on the graphs presented for distribution fitting, we can see one that tests the Gravity Model. I leave the definition and explanation of this model for further work (probably the project presentation).

6.6. Further work

Although this internship is administratively over, I have been continuing to work on it with my professor until now and plan to keep studying it for a while.

My professor wants to write a research paper on it and this is a very big motivation for me to complete this project. This would help in my personal realizations and accomplishment as well as on the job market, as this experience is very valued by head hunters.

We still have many ideas to try on and the early results of these new tests are quite promising and exciting.
Here are a few ideas that I am currently testing or will in a near future:

- Try different data distributions to generate the gravity model
- Work on the generalized gravity model
- Try to estimate and modelize other data such as:
  - The sums of the $a_{i,j} + a_{j,i}$ (sum of the traffic from $i$ to $j$ and the traffic from $j$ to $i$).
  - The value of one OD flow over a period of time (Temporal modelization)
  - View the couples $(a_{i,j}, a_{j,i})$ as points coordinates. Plot them and try to fit a model to this plot. (It seems that the log-log plot of the points $(a_{i,j}, a_{j,i})$ might be linear)

7. Human aspects of the project

There is little to say about human aspects for this project as the only collaborators I had to deal with were Mr. Durouchoux (to provide him a few updates on the project) and Mr. Chlebus.

I had a really great time working on this project with Professor E. Chlebus because, on top of being very understanding when I was overloaded with class work, he provided me with valuable feedback every time I would show him some results and helped me improve and work with for myself with better efficiency.

This was the first time I was totally immerged in a research environment and exchanging with Dr Chlebus, and to some extent, with other classmates made me become more self-driven and independent. I also could exchange e-mails with Matthew Roughan from the University of Adelaide in South Australia. This experience made me realized how supportive and helping the research community is.

I also had to work a big part of this project away from IIT, which made it harder for me to obtain immediate help when I would face obstacles. This forced me in searching the solutions alone, and to only contact my professor when I felt I was in a dead-end. I feel like this has empowered my ability of synthesizing information and extracting only important results or issues.
8. Conclusion

This Research Project really fits to its name as I was able to deepen both aspects of it and work hard on this challenging task, which I believe really changed me and offered me an invaluable experience.

➢ “Project” it was without doubt. Also this is not the first and last project I have done; I could still improve my organizational skills, and had to face with some decisional choices. The “project” aspect was certainly not the main part of this internship but throughout the years, I have learnt to value every small opportunity and I wouldn’t have want to miss improving my abilities on project management.

➢ “Research” was the most interesting and challenging aspect of this internship. For the first time of my life I felt like I had to deal with things I had never seen before. I could test and improve the limits of my research competences. I could make a practice of the knowledge acquired throughout the years of my graduate education. I could understand that the best asset for a researcher is to be able to assimilate unknown domains and put them into perspective with his own background and knowledge. And ultimately I could live the exciting feeling of finding new results that improve not only my personal knowledge but also the global common knowledge.

I hope, that, through this “Mémoire”, I could share with you some of the passion I had - and still have - in working on this project, and that you could read this report with great interest.

This experience really helped me defining more precise goals for my future work life. I really hope I will be able to find a position in a company that allows me to continue focusing on research, and to participate in tomorrow’s exciting discoveries.

Mr Chlebus offered me to continue working with him on this project with the goal of publishing a research paper, which will be a thrilling experience for me. He also offered me, and this is what I really look forward to the most, to continue and extend this project in a future Ph. D.
To conclude, this project has brought me more experience than I could ever have expected; and most importantly, has provided me great perspectives for my future. Now the ball is in my court and I am ready to seize the opportunity.
References


Zhang, Yin. *@ UTCS.* http://www.cs.utexas.edu/~yzhang/.

Credits

Illustration 1 & Illustration 3: Lawrence Biemiller, with his permission - http://www.iceandcoal.org/


Figure 1: IIT official website - http://www.iit.edu/provost/acad_offices.shtml

Figure 2, Figure3 & Figure 4: network diagrams created with the online diagramming application Lovely Charts - http://lovelycharts.com/

Figure 5: Created with Microsoft Excel 2007 using statistics from the Internet Systems Consortium - http://www.isc.org/

Figure 6: courtesy of Internet2 Consortium - http://www.internet2.edu/

Figure 7: courtesy of the GEANT project - http://www.geant.net

Figure 8: Jarkko Piirainen, under Wikimedia Commons license - http://en.wikipedia.org/wiki/File:Overview_of_the_Common_Language_Infrastructure.svg

Figure 9: Image extracted from an XML file on open source software Notepad++ - http://notepad-plus-plus.org/

Figure 10 & Figure 11: UML diagrams created with Visual Paradigm for UML Community Edition - http://www.visual-paradigm.com/

Figure 12 & Figure 13: Charts created with Microsoft Visual Studio 2010 Professional Edition

Figures 14 to 21: Charts created with Matlab
Appendices

Appendix A – ABILENE XML formatting Source Code

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Globalization;
using System.Xml;

/*****************************************/
* Project: Abilene Formatting
* Source: Program.cs
* Date: 2010/07/03
* Author: Jeremy Henault
* This program formats the datasets of ABILENE to the same XML format used for the
datasets of the GEANT network
* ****************************************/

namespace AbileneFormating
{
    class Program
    {
        static void Main(string[] args)
        {
            List<DateTime> dates = new List<DateTime>();
            List<List<double>> matrix = new List<List<double>>();

            NumberFormatInfo provider = new NumberFormatInfo();

            //Initial Dates of each TM file
            dates.Add(new DateTime(2004, 03, 01, 0, 0, 0));
            dates.Add(new DateTime(2004, 03, 08, 0, 0, 0));
            dates.Add(new DateTime(2004, 04, 02, 0, 0, 0));
            dates.Add(new DateTime(2004, 04, 09, 0, 0, 0));
            dates.Add(new DateTime(2004, 04, 22, 0, 0, 0));
            dates.Add(new DateTime(2004, 05, 01, 0, 0, 0));
            dates.Add(new DateTime(2004, 05, 08, 0, 0, 0));
            dates.Add(new DateTime(2004, 05, 15, 0, 0, 0));
            dates.Add(new DateTime(2004, 05, 22, 0, 0, 0));
            dates.Add(new DateTime(2004, 05, 29, 0, 0, 0));
            dates.Add(new DateTime(2004, 06, 06, 0, 0, 0));
            dates.Add(new DateTime(2004, 06, 12, 0, 0, 0));
            dates.Add(new DateTime(2004, 06, 19, 0, 0, 0));
            dates.Add(new DateTime(2004, 06, 26, 0, 0, 0));
            dates.Add(new DateTime(2004, 07, 03, 0, 0, 0));
            dates.Add(new DateTime(2004, 07, 10, 0, 0, 0));
            dates.Add(new DateTime(2004, 07, 17, 0, 0, 0));
            dates.Add(new DateTime(2004, 07, 24, 0, 0, 0));
            dates.Add(new DateTime(2004, 07, 31, 0, 0, 0));
            dates.Add(new DateTime(2004, 08, 08, 0, 0, 0));
        }
    }
}
```
dates.Add(new DateTime(2004, 08, 13, 0, 0, 0));
dates.Add(new DateTime(2004, 08, 21, 0, 0, 0));
dates.Add(new DateTime(2004, 08, 28, 0, 0, 0));
dates.Add(new DateTime(2004, 09, 04, 0, 0, 0));

//loop on the 24 ABILENE files
for (int i = 1; i <= 24; i++)
{
    matrix.Clear();
    //we read the ith ABILENE file
    matrix = readFile(i);

    // We create the 2016 new TM files from the ith ABILENE file.
    DateTime date = dates[i - 1];
    for (int j = 0; j < matrix.Count(); j++)
    {
        createXml(matrix[j], (date));
        //update date for fileName
        date = date.AddMinutes(5);
    }
}

private static List<List<double>> readFile(int idMatrix)
{
    List<List<double>> matrix = new List<List<double>>();
    List<double> temp = new List<double>();

    NumberFormatInfo provider = new NumberFormatInfo();
    provider.NumberDecimalDigits = 7;
    provider.NumberDecimalSeparator = ".";

    string[] splitValue;
    string formattedNumber;
    string[] lineSplit;
    string strMatrix = "X";
    string line = "";

    if (idMatrix < 10)
        strMatrix += "0" + idMatrix;
    else
        strMatrix += idMatrix;

    string filePath = Path.Combine(Directory.GetCurrentDirectory(), @"..\..\Content\"+strMatrix+"\"");
    string fileName = strMatrix;

    System.IO.StreamReader file = new System.IO.StreamReader(filePath+fileName);
    while ((line = file.ReadLine()) != null)
{  
    lineSplit = line.Split(' ');  
    temp.Clear();  
    
    //get the matrix  
    for (int i = 1; i < lineSplit.Length; i+=5)  
    {  
        //formattedNumber = lineSplit[i].Split('e')[0] + " e" +  
        lineSplit[i].Split('e')[1];  
        formattedNumber=lineSplit[i].Insert(9, " ");  
        splitValue = lineSplit[i].Split('e');  
        temp.Add(Double.Parse(splitValue[0], provider) * Math.Pow(10,  
        Double.Parse(splitValue[1], provider))));  
    }  
    matrix.Add(temp.GetRange(0,temp.Count));  
    
}  

file.Close();  
return matrix;  

//this method takes a line of doubles (matrix) and a date as input and create the XML file  
private static void createXml(List<double> matrix, DateTime date)  
{
    string fileName = "IntraTM-" + date.ToString("yyyy-MM-ddTHH:mm:ss");  
    string filePath = Path.Combine(Directory.GetCurrentDirectory(),  
    @"\..\..\Content\XML\");  
    XmlTextWriter xmlTextWriter = new  
    XmlTextWriter(filePath+fileName+".xml",System.Text.Encoding.UTF8);  
    xmlTextWriter.Formatting = Formatting.Indented;  
    xmlTextWriter.WriteStartDocument();  
    //xmlTextWriter.WriteComment("Creation fichier XML test"); // commentaire  
    //TraficMatrixFile  
    xmlTextWriter.WriteStartElement("TrafficMatrixFile");  
    xmlTextWriter.WriteStartElement("info");  
    xmlTextWriter.WriteElementString("date", date.ToString("yyyy-MM-ddTHH:mm:ss")));  
    xmlTextWriter.WriteElementString("duration", "300");  
    xmlTextWriter.WriteElementString("author", "Jeremy Henault");  
    xmlTextWriter.WriteElementString("unit", "kbps");  
    xmlTextWriter.WriteEndElement();  
    xmlTextWriter.WriteStartElement("IntraTM");  
    int cptSrc = 1;  
    int cptDst = 1;  
    for (int i = 0; i < matrix.Count; i+=12)  
    {  
        xmlTextWriter.WriteStartElement("src");  
        xmlTextWriter.WriteAttributeString("id", "" + cptSrc);  
        xmlTextWriter.WriteEndElement();  
        xmlTextWriter.WriteStartElement("dst");  
        xmlTextWriter.WriteAttributeString("id", "" + cptDst);  
        xmlTextWriter.WriteEndElement();  
    }  
    xmlTextWriter.WriteEndElement();  
    xmlTextWriter.WriteEndDocument();  
}
cptDst = 1;
for (int j = i; j < i+12; j++)
{
    xmlTextWriter.WriteStartElement("dst");
    xmlTextWriter.WriteAttributeString("id", "" + cptDst);
    xmlTextWriter.WriteValue((long)matrix[j]);
    xmlTextWriter.WriteEndElement();
    //xmlTextWriter.WriteElementString("dst id=" + cptDst + ", " + matrix[j]);
    cptDst++;
}
xmlTextWriter.WriteEndElement();
cptSrc++;
}
xmlTextWriter.WriteEndElement();
xmlTextWriter.WriteEndElement();
xmlTextWriter.Flush(); //vide le buffer
xmlTextWriter.Close(); // ferme le document
}
Appendix B – A Versatile Probability Distribution for Light and Heavy Tails of Web File Sizes

A Versatile Probability Distribution for Light and Heavy Tails of Web File Sizes

Edward Chlebus and Gnutzig Drugi
Network Modeling and Teletraffic Analysis Lab
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
Email: {chlebus, drugi}@iit.edu

Abstract—A novel unique probability distribution, which has a lognormal body and either light or heavy tail, has been fitted to various empirical data sets of Web file sizes. The optimal parameters of this distribution have been determined by the maximum likelihood estimation combined with the optimization algorithm minimizing a goodness-of-fit metric specially adopted to provide the best fit to the upper tail. The mirror transformation of the processed original data set with respect to the median has been proposed to improve the fit. The obtained results question the common opinion that the probability distribution of Web file sizes is heavy-tailed. The lognormal fit are given for comparison.

I. INTRODUCTION

Peer-to-peer (P2P) traffic exploded and consumed the largest percentage of the network bandwidth over the last couple of years but according to the recent study based on usage data of about 1 million subscribers in North America, Web traffic has surpassed P2P and become again a dominant Internet load [14]. The recent growth of Web traffic is mainly due to more and more popular streaming video (e.g. YouTube) but the traditional web browsing, i.e. downloading text and images still constitutes 45% of the entire Web traffic. Nowadays mobile users also contribute to its volume as they increased by 13% the total population of Web visitors. US, UK and Italy are leaders in mobile Internet penetration with respective 15.8%, 12.9% and 11.9% of mobile subscribers that are active Internet users. Their number in the US increased by 72% from 23.4 million in May 2006 to 40.4 million in May 2008 [9]. “Full” Web surfing currently represents 77% of all mobile Web traffic. Mobile-friendly WAP and mobile sites attract the remaining 23% of the traffic and this share continues to decline [28].

Determining the Web file size distribution is one of the problems of key importance to understanding Web traffic patterns and what follows, proper performance analysis of the network and capacity planning of its infrastructure. Judging by the aforementioned statistics, nowadays this applies not only to wireline but also to wireless mobile networks.

A file size distribution has been extensively investigated and it appears to have a lognormal body but its tail is usually claimed to be heavy [1-5, 12]. The following distributions have been proposed by different authors to jointly model the lognormal body and heavy tail:

- 5-parameter hybrid of lognormal and Pareto [5];
- 4-parameter biPareto [21];
- 4-parameter double Pareto [39];
- 4-parameter double Pareto-lognormal [26].

Downey [13], contrary to most authors, claims that there is no compelling evidence that the distribution of file sizes is heavy-tailed even in case of the data that have already been reported as such. Unfortunately the lognormal distribution doesn’t provide a good fit to many light-tailed empirical data sets he has examined.

We need a better model for Web file sizes than those developed so far. A desired ideal probability distribution should have a small number of parameters, lognormal body and be versatile enough to fit to the measurement data claimed in the literature as either light or heavy-tailed. Finding such a solution is a real challenge.

In our previous paper [6] we have modeled traffic of a wireless Internet access session by means of a newly introduced distribution. It meets all the specified above criteria and looks like a good candidate for the desirable solution. The objective of this paper is to analyze properties of our distribution in more detail, test whether it is a good model for a variety of empirical data sets of Web file sizes and find an optimal fit to them.

II. A NOVEL PROBABILITY DISTRIBUTION FOR MODELING INTERNET TRAFFIC

In a series of our recent papers [6,7,10,11] we have investigated traffic in a commercial nationwide wireless WiFi network. We have extracted from the measurement log numerical values of three random variables characterizing an Internet access session, namely its duration and the associated up- and downloaded session traffic volumes. All samples of each of these variables X have been sorted in a nonincreasing order

\[ x_1 \geq x_2 \geq \ldots \geq x_n \geq \ldots \geq x_{n-2} \geq x_{n-1} \geq x_n \]  \hspace{1cm} (1)

The index \( r \) denotes the rank of the sample \( x_r \) with \( x_1 = x_\text{max} \) and \( x_n = x_\text{min} \). Presenting values of \( x_r \) on a semi-logarithmic scale as a function of their ranks \( r \) results in a characteristic “lazy” S-shaped curve which is well described by the Lavalle’s law [18,25]

\[ x_r = \left( \frac{r}{n+1} \right)^{\beta} \]  \hspace{1cm} (2)

978-1-4244-2948-6/09/$25.00 ©2009 IEEE
Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on June 02, 2010 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.
with parameters $c$ and $\beta$, originally proposed to relate values of the
journal impact factor $x_j$ with their ranks $r$. In order to
increase fitting flexibility of (1) we have introduced an
additional parameter $k$ in the denominator which gives the
formula [6]

$$x_j = \frac{\ln(1 + r)}{k + r} \beta$$

(3)

The modified Lavatele's law (3) fits very well to all the
wireless traffic data examined in [6]. Applying the
methodology proposed in [8] we have derived from (3) a novel
distribution for $X$. Its probability density function (PDF) $f(x)$
and cumulative distribution function (CDF) $F(x)$ are given by
[6]

$$f(x) = \frac{(1 + x)^{-1 - \alpha}}{\beta}$$

and

$$F(x) = \frac{1}{1 + x}$$

respectively, where $c$, $\alpha$ and $\beta$ are parameters. Sample
CDF (4b) and complementary CDF (CCDF) $F_n(x)=1-F(x)$
are presented in Figs. 1 and 2, respectively. It is clearly seen
from Fig. 2 that our new probability distribution (4) has a unique
property of modeling both light (if $d=0$) and heavy (if $d>0$)
tails. Let's explain this.

Consider the truncated Pareto distribution whose CDF
for $0 < x \leq \infty, k > m$ and $\alpha > 0$ given by [1]

$$F(x) = \frac{1}{1 + x}$$

(5)

is rewritten as

$$F(x) = \frac{1}{1 + x}$$

(6)

For large $x$ and by inference also for large $m$ both the
numerator and denominator of (6) are well approximated by
the first two terms of the Maclaurin series

$$\frac{1}{1 + x^\alpha} \approx \frac{1}{\alpha x^\alpha}$$

(7)

which yields the following approximation of (5)

$$F(x) = 1 - \left(\frac{x}{m}\right)^\alpha$$

(3)

equivalent to the CDF (4b) with $F(x) \sim 1 - \left(\frac{x}{m}\right)^\alpha$.

It's clear now that our model (4) with $d=0$ has the tail of the
truncated Pareto distribution which is light [1]. $d>0$ converts
(5) into the CDF of the Pareto distribution $F(x)=1-(k/m)^\alpha$
whose tail is heavy and can be easily identified by its linear
behavior on a logarithmic scale (cf. Fig. 2). Hence the value of
the parameter $d$ has an absolutely critical effect on the tail of
the distribution (4).

III. MEASUREMENT DATA

We will demonstrate the versatility of our distribution (4)
for modeling six different data sets. The first four of them
contain sizes of documents (files) requested from World Wide
Web servers located at the following sites:

- Department of Computer Science at the University of
  Calgary (a department-level server);
- University of Saskatchewan (a campus-wide server);
- NASA Kennedy Space Center;
- ClarksNet (a commercial Internet service provider in
  the Baltimore-Washington, DC area).

All these data collected in 1994-95 were originally
analyzed by Arlitt and Williamson in [3, 4]. We have
selected these traces for two basic reasons. Firstly, they are publicly
available in the Internet Traffic Archive [17] and have been
also used by other authors, e.g. [13, 20]. Secondly, the papers
[3, 4] have received over 1000 citations (see Google Scholar)
that laid the foundation for the ubiquitous heavy tail paradigm
of the Web file size distribution. Downey [13] and Gong et al.
[15] have recently questioned $x$. This problem is debatable and
we would like to contribute to this discussion.

Since the above traces are somewhat outdated we have
also examined two additional "fresh" sets of Web file sizes
collected in July 2008 at the following websites:

- TWA Photograph Studio (www.twaphoto.com);
- "ShowMe!"
  (www.wisnese.com/weshowme/index.html).

Note that the latter is targeted at mobile subscribers and can also
be accessed through an interface at www.facebook.com which
is one of the most frequently visited websites by mobile
Internet users in US, UK and South Africa [28]. Tab. 1
summarizes all the six data sets.

IV. MAXIMUM LIKELIHOOD ESTIMATION (MLE)
FOR THE DISTRIBUTION (4)

Given the empirical data set \((x_1, x_2, ..., x_n)\) let's write down
the likelihood function (see e.g. [22]) for the distribution

$$L(x_1, x_2, ..., x_n; c, d, \beta) \propto f(x_1)f(x_2)\cdots f(x_n)$$

$$f(x) = \frac{k^\beta x^{\beta - 1}}{\Gamma(\beta)}$$

(4)

The corresponding logarithmic likelihood function has the form

$$L(x_1, x_2, ..., x_n; c, d, \beta) \propto \ln(L(x_1, x_2, ..., x_n; c, d, \beta))$$

$$= -n \ln(1 + d) - \ln(\beta) - \ln\Gamma(\beta) - \frac{1}{\beta} \sum_{i=1}^{n} \ln x_i$$

(10)

Setting the CDF (4b) equal to one for $m$ denoting the upper
bound of $X$, one can easily find

$$d = \frac{\ln(1 + d)}{\ln(\beta)}$$

(11)
Rewriting (10) with (11) we get
\[ x_i \sim F \left( \frac{i}{n+1} \right), \quad F(v) \text{ denotes the CDF of the fitted probability distribution and } x_i \text{ represents the } i\text{th sample of the data sorted in ascending order, i.e. } x_1 = x_{\min} \text{ and } x_n = x_{\max}. \]

Many graphs have been visually tested and the quality criterion (14) appears to provide satisfactory fits to empirical CDFs presented on a logarithmic scale [20].

Since \( q \) is usually very large, in Tab. 2 we have reported \( g_m = q/n \), i.e. the mean value of (14) per sample (in case of identical samples, we replace \( n \) with the number of unique samples (cf. Tab. 1), for which the empirical CDF is defined).

VI. DETERMINING OPTIMAL PARAMETERS OF THE FIT

For a fixed value of \( m \) the MLE estimators of \( c \) and \( \beta \) satisfying (13a-c) can be found by applying the following iterative bisection search algorithm successfully used in [6] for \( m = \hat{m}_{\max} \).

Algorithm 1

Step 1. Set the initial lower \( \beta_i, \) and upper \( \beta_u \) bounds for \( \beta \).

Step 2. Set \( \beta = \beta_{\bar{z}} \).

Step 3. Set the initial lower \( c_i \) and upper \( c_u \) bounds for \( c \).

Step 4. For \( \beta \), \( c \), do the following:

Step 5. If \( h_1(c, m, \beta) < c \) then go to Step 6, else go to Step 5.

Step 6. If \( h_2(c, m, \beta) > 0 \) update the lower bound \( c_i = c \) and go to Step 4, else update the upper bound \( c_u = c \) and go to Step 4.

Step 7. If \( h_3(c, m, \beta) > 0 \) update the lower bound \( \beta_i = \beta \) and go to Step 2, else update the upper bound \( \beta_u = \beta \) and go to Step 2.

This algorithm uniquely determines \( c(m) \) and \( \beta(m) \) for a given \( m \) so to the quality criterion (14) \( q(c(m), m, \beta(m)) \) for the fitted probability distribution (4) becomes \( q(m) \), i.e. a function of the variable \( m \) only. To minimize \( q(m) \) with respect to \( m \) we search the entire domain of \( m \). Thanks to this the algorithm doesn’t eliminate the solution \( m(0) \), i.e. a heavily-tailed fit, if it is the best. In practice the minimum is usually found in close proximity of \( \hat{m}_{\max} \). A sample optimization run is illustrated in Fig. 3. The optimal fit determined with the numerical accuracy \( \varepsilon = 10^{-6} \) are specified in Tab. 2 and depicted in Figs. 4-9.
VII. THE MIRROR TRANSFORMATION (MT) OF DATA

Take a look again at Fig. 1. Judging by it, the CDF (4b) with \( \alpha = 0 \) and median equal to \( c \) has a symmetric center of mass \( S(c, 0.5) \) on a semi-logarithmic scale. Let’s prove this.

Consider two points \( A(x, F(x)) \) and \( A’(x’, F(x’)) \). If they are symmetric with respect to \( S \), then

\[
\frac{\log{x} + \log{x’}}{2} = \frac{\log{c}}{2}
\]

should imply

\[
F(x) + F(x’) = 0.5
\]

From (15) we get immediately

\[
\chi = \frac{c^2}{x'}
\]

Rewriting (16) with (4b) and (17) yields

\[
\frac{1 + \log{x} - \log{x’}}{2} = \frac{1}{2}
\]

which completes this simple proof.

Consider now the data sorted in ascending order

\[
\chi_{\text{min}} = \chi_1, \chi_2, \ldots, \chi_{n-1}, \chi_n, \ldots, \chi_{2n-1}, \ldots, \chi_{3n-1}, \ldots, \chi_{n^2-1}, \chi_{n^2}
\]

Following (17) we replace all samples \( \chi_i \) with \( \chi_i = \chi_{n-i} \) which gives on the logarithmic scale the mirror reflection of the upper tail with respect to the median

\[
\frac{\chi_i}{\chi_{n-i}} = \frac{\chi_{n-i}}{\chi_i}, \ldots, \frac{\chi_{n-1}}{\chi_{n-1}}, \frac{\chi_{n-2}}{\chi_{n-2}}, \ldots, \frac{\chi_1}{\chi_1}
\]

Defining the empirical CDF as \( F_\text{ev}(x) = \frac{i}{n+1} \) one can easily prove that the new transformed data set (20) fulfills the condition (16)

\[
F_\text{ev}(\chi_{n-i}) + F_\text{ev}(\chi_i) = \frac{i}{n+1} + \frac{n-i+1}{n+1} = 1
\]

i.e. it also has a center of symmetry.

By deleting the lower tail of the original data and applying the proposed mirror transformation (MT), we keep unchanged the empirical CDF \( F_\text{e}(x) \) of the upper tail \( x_i > x_\text{med} \) while introducing symmetry to the new transformed data set. The latter may increase goodness of fit provided by (4) to the upper tail since our distribution, as we have just proved, is symmetric.

The initial assumption underlying the MT presented in this section was \( \alpha = 0 \). As we can see from Eq. 6 in most cases \( \alpha \neq 0 \) but its value is so small that this assumption is approximately still valid.

We have run Algorithm 1 (cf Section 5) and estimated the parameters \( c, m \) and \( \beta \) of the fitted distribution (4) for the transformed data set (20) but the quality criterion \( q (14) \) was consistently calculated and minimized for the original data set (19). Our objective is still the same, i.e. to optimize the fit to (19). The MT was used to improve the parameter estimation and to modify the processed data. The fits determined by applying this method are presented in Tab. 1 and Figs. 4-9.

VIII. REFERENCE LOGNORMAL FITS

Generally, there is agreement that the body of the file size distribution is lognormal \([5,11,13,20]\). Some tails of empirical CCDF examined by Downey [13] also show the curvature characteristic for the lognormal distribution; hence we use it for comparison with (4).

The lognormal distribution has the following PDF

\[
f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[ -\frac{\ln{x} - \mu}{2\sigma^2}^2 \right]
\]

and CDF

\[
F(x) = \frac{1}{2} \left( 1 + \text{erf}\left( \frac{\ln{x} - \mu}{\sigma\sqrt{2}} \right) \right)
\]

where \( \mu \) and \( \sigma \) are parameters. Their optimal values, determined by applying both the maximum likelihood estimation (MLE) and the moment matching estimation (MME), are given in Tab. 2. The resulting lognormal fits are illustrated for comparison with (4) in Figs. 10-13.

IX. CONCLUSIONS

It is clearly seen from Tab. 2 and Figs. 4-9 that applying the novel probability distribution (4) to modeling Web file sizes was a good idea. The obtained optimal fits (i.e. for each data set the better of the proposed two) are visually attractive and in five out of six cases they outperform the lognormal model (22). The later works best only in one case, i.e. the MME fit to the NASA data.

The resulting optimal fits provided by (4) to all but one data sets are light-tailed. Only the CCDF of the Saskatchewan fit without MT has a heavy tail. Thus the distribution (4) has fully revealed its practical versatility and usefulness for modeling both light and heavy-tailed actual empirical data.

The proposed MT significantly improves the fits in two cases, namely Calgary and NASA. For all the remaining data the obtained results are worse although the difference is insignificant except ClarkNet. The CDF symmetry is simply not the only factor affecting a successful fit. Employing the MT doesn’t automatically imply better results.

The distribution (4) performs very well although it has only three parameters, i.e. less than each distribution listed in Section 1. The parameter \( \beta \) affects the shape of the tail and can be used to estimate its thickness. All tails for \( \alpha > 0 \) are light, but the higher the value of \( \beta \) the greater the departure from the Pareto distribution and what follows, the higher the tail (cf Fig. 2).

Our results question the evidence and common opinion that the probability distribution of Web file sizes is heavy-tailed; hence this paper supports the findings reported by Downey [13] and Gong et al. [15]. Only for the Saskatchewan data the best fit has \( \alpha = 0 \) and a heavy tail but even in this case the best light-tailed fit, provided by the MT, is only about 6% worse measured by the value of \( q_{opt} \). Since the Saskatchewan data set has the most irregular and asymmetrical shape of the tail compared with other examined traces, one must be very careful with generalizing any statements based on this case.

Note that final conclusions regarding whether or not a certain Web workload component needs a heavy-tailed model may differ depending on the versatility of an applied model.
X. IMPLICATIONS FOR NETWORK PERFORMANCE MODELING

It is widely believed that Internet traffic self-similarity arises from transfer of heavy-tailed files characterizing data at the application layer [24]. Why are files supposed to be heavy-tailed? There is no good answer to that question. This fact is accepted as a sort of axiom underlying Internet traffic analysis and implying other heavy-tailed network features [22]. The opinion that research has provided evidence strong enough to justify this axiom, is common. We do not share it. Jung et al. [15] explicitly argue that the focus on power law tails in the Internet is misguided for various reasons, e.g., it is extremely difficult to statistically characterize an upper tail of a probability distribution based on a data sample of a finite size. Famig et al. have proved in [16] that not only the heavy-tailed Pareto but also the light-tailed lognormal distribution can imply finite-range dependence. References [15] and [16] along with recently cited Donnay’s paper [13] question the status quo of ostensibly ubiquitous heavy tails in the Internet. A possible controversy these papers might cause is probably the main reason why, judging by the number of citations, they have been ignored by the networking community.

The probability distribution (4) that we have introduced supports the claim of the aforementioned authors and provides a model for quantifying an impact of various tails of Web file sizes on network performance. It is shown in [23] that self-similarity deteriorates throughput and leads to increased packet in and retransmission rates, big delays and excessive mean path lengths. An open loop congestion control using UDP-based unreliable transport performs very poorly under such circumstances. Traffic self-similarity has also an analogously degrading effect on TCP performance but due to the employed congestion control and reliability mechanisms, this protocol is much more resistant to an increasing degree of long-range dependence. We expect that eliminating heavy-tails will improve basic network performance metrics, enable less conservative capacity planning and simplify design of efficient congestion control algorithms. Note that our model enables a gradual departure from the heavy tail paradigm as continuously increasing numerical values of the parameter d (cf. (4)) starting with d=0 provides a very smooth transition from heavy to light tails. As such this model is very well suited for detailed quantitative traffic analysis, e.g., TCP throughput determination or optimal buffer dimensioning for a given input file size distribution with specific d.

The bottom line, however, is not to develop a new model but to apply it in practice. Internet is a large-scale complex system and simulation is still a dominant method for evaluating various aspects of its performance. Accuracy of simulation results depends on the capability of generating realistic synthetic workloads. There are two basic approaches to this problem: application-specific and application-independent. SURGE (Scalable URL Reference Generator) [5] is an example of the former. It creates a realistic Web workload matching empirical statistics of different random variables characterizing Web browsing. The server file size is one of them. SURGE implements the 5-parameter hybrid of the lognormal distribution for modeling the body of file sizes and the heavy-tailed Pareto for their tail. Our 3-parameter distribution (4) is a perfect replacement for this complicated and inconvenient hybrid model. Most Internet links carry traffic of many applications and application-specific generators like SURGE are inadequate for modeling it. To investigate combinations of different applications, HARPOON [27] and TMIX [29] have been developed. A series of application-independent file transfers between the endpoint processes of each logical connection is a basic mechanism driving traffic generation. A trace from a network link or alternatively a file size probability distribution derived from it may be taken by these generators as input. A problem of modeling a representative workload arises at this point. We can solve it by determining an empirical file size probability distribution for each major application and then mixing such distributions with the predetermined weights to meet the requirements of different simulation scenarios. Our distribution (4) is again a perfect input to this model.

We presume that applying the newly introduced probability distribution (4) is not to completely eliminate Internet traffic self-similarity but an increasing value of d may limit heavy-tails believed to be a major source of this phenomenon. Since self-similarity is an empirically confirmed fact, a further insight into its origina is needed.

REFERENCES

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.


### Table I

<table>
<thead>
<tr>
<th>Basic Statistical Characteristics of the Examined Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sample size n</strong></td>
</tr>
<tr>
<td><strong>Number of unique samples</strong></td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
</tr>
<tr>
<td><strong>Mean</strong></td>
</tr>
<tr>
<td><strong>Median</strong></td>
</tr>
<tr>
<td><strong>Standard deviation</strong></td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Comparision of Optimal Fits Provided by the Examined Probability Distributions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Calgary</strong></td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>New distribution (4)</td>
</tr>
<tr>
<td>Lognormal (2) with MT</td>
</tr>
<tr>
<td>Lognormal (2)</td>
</tr>
<tr>
<td>Lognormal (2)</td>
</tr>
</tbody>
</table>

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on June 30, 2021 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.
Appendix C – Truncated Pareto distribution algorithms (fitting and CDF) – Matlab Code

% This function fits the truncated pareto to the sample X and returns the parameters of the distribution
function [Xmin Xmax alpha] = truncparetofit(X)
    Xmin = min(X);
    Xmax = max(X);
    n = numel(X);
    % choice of lower and upper bounds for alpha (chosen from observations of the research paper)
    alpha_l = -2;
    alpha_u = 8;
    eps = 0.001;
    lnMin = log(Xmin);
    lnX = log(X);

    % right value of the formula (fixed)
    right = lnX - lnMin;
    right = sum(right);
    XminXmax = Xmin/Xmax;
    alpha = (alpha_l + alpha_u)/2;

    % left value of the formula (to be adjusted)
    left = (n/alpha) + (n*(XminXmax)^alpha*log(XminXmax))/(1-XminXmax^alpha);

    while abs(left - right) > eps
        alpha = (alpha_l + alpha_u)/2;
        left = (n/alpha) + (n*(XminXmax)^alpha*log(XminXmax))/(1-XminXmax^alpha);
        if (abs(left - right)>0)
            if left > right
                alpha_l = alpha;
            else
                alpha_u = alpha;
            end
        end
    end

% this function computes the CDF of the truncated pareto distribution on the sample X using the parameters Xmin, Xmax and alpha
function Y = truncparetocdf(X, Xmin, Xmax, alpha)
    num = 1 - (Xmin ./ X).^alpha;
    den = 1 - (Xmin / Xmax).^alpha;
    Y = num * (1/den);
end
%This function fits the Chlebus to the sample X and returns the 
%parameters of the distribution

function [c, b] = chlebfit(X)
    %precision parameter
    eps = 0.000001;
    %choice of initial bounds for variables c and b (keep those of algo
    %might not converge!)
    min_c_init = 0;
    max_c_init = max(X);
    min_b_init = 0;
    max_b_init = 3;
    %variables c_l, c_u, b_l, b_u
    c_l = min_c_init;
    c_u = max_c_init;
    b_l = min_b_init;
    b_u = max_b_init;
    b= (b_u + b_l) / 2;
    c = (c_u + c_l) /2;
    Hbeta = 1000000;
    Hc = 1000000;
    cpt=0;
    while (abs(Hbeta)> eps)
        cpt=cpt+1;
        b = (b_u + b_l)/2;
        c_l = min_c_init;
        c_u = max_c_init;
        Hc = 1000000;
        while (abs(Hc) > eps)
            c = (c_u + c_l) /2;
            %we compute the new value of Hc with the new value of beta (b)
            Hc = h_c (c, b, X);
            if abs(Hc) > eps
                if Hc >0
                    c_l = c;
                else
                    c_u = c;
                end
            end
        end
        %we compute the new value oh Hbeta with the new value of c
        Hbeta = h_beta(c, b, X);
        if abs(Hbeta) > eps
            if Hbeta >0
                b_l = b;
            else
                b_u = b;
            end
        end
    end
% this function computes the CDF of the Chlebus distribution on the
% sample X using the parameters b and c (the third parameter d depends on b
% and c)

function Y = chlebcdf(X,c,b)
    b_ = 1/b;
    d = (c/max(X))^b_ + 1;

    Y = 1+(c./X).^b_;
    Y = d./Y;
end