Universidad Carlos III de Madrid

Master in Telecommunications Engineering

2016-2017

Master Thesis

"Statistical and graphical tool for network cost analysis and support for CAPEX allocation"

Jorge Rubio Álvarez

Tutores

Adrián Maeso Martín-Carnerero (Telefónica Investigación y Desarrollo S.A.U.)
Pablo Basanta Val (Universidad Carlos III de Madrid)

Leganés, 9 de febrero de 2017

Palabras clave: Indicador clave de rendimiento (KPI), análisis de datos, tráfico cursado, costes, R.

Resumen: Este trabajo consiste en el desarrollo de una herramienta en el lenguaje R para realizar un análisis estadístico y una representación gráfica de los costes de red históricos en los que se ha incurrido, así como del tráfico cursado en estas redes, tanto en fijas como en móviles. Además, la herramienta permitirá dar soporte a la decisión de asignación de inversiones mediante la proyección de los tráficos futuros de cada red en función de diversos parámetros de entrada. Todo ello se podrá ver reflejado gráficamente mediante una aplicación web generada en R. Para el desarrollo de la herramienta, se van a utilizar datos de costes y tráficos de red de las distintas operaciones/países del grupo Telefónica.

ABSTRACT

In the telecommunications field, there is a selection of relevant information referred as key performance indicators due to the fact that they provide the chance to visualize the network communications performance. This data's disadvantage is that they can be confined in massive data sets in which it is complex to elucidate at a glance the performance of the different parts of a communications system.

This project performs an analysis and a graphic representation of these relevant performance indicators, which allows studying the mobile and fixed networks of a telecommunications company. In specific, the implementation of this tool is displayed by using all traffic data and network costs from different operations and countries of Telefónica footprint. This implementation gives the opportunity to check that the data analysis is done in an accurate way.

In order to obtain these graphic illustrations of network costs and traffic carried over all networks, both historical and predictions, diverse methodologies are applied to key performance indicators in order to acquire its calculus and estimation. Afterwards, based on the information obtained with these procedures, we can give support in order to make decisions of investment allocations in the company's networks.

Finally, this tool has been performed with R programming language as well as for the calculation of the different data as to the development of the web application to represent this data in a graphic way with the Shiny package. Telefónica's data has been modified in order to assure its confidentiality.

KEYWORDS

Key performance indicator (KPI), data analysis, traffic, costs, R.

RESUMEN

En el mundo de las telecomunicaciones, hay información relevante denominada indicadores clave de rendimiento que nos proporcionan la posibilidad de visualizar el rendimiento de las redes de comunicaciones. El inconveniente de estos datos es que pueden estar en grandes conjuntos de datos de los que es complicado dilucidar a simple vista el rendimiento que tienen las diferentes partes del sistema de comunicaciones.

En este Trabajo Fin de Máster se ha realizado un análisis y representación gráfica de los indicadores más relevantes de rendimiento que permite el estudio de las redes móviles y fijas de las que se compone una empresa de telecomunicaciones. En concreto, se presenta la implementación de la herramienta mediante los datos de tráfico y de los costes de red de las distintas operaciones y países del grupo Telefónica. Esta implementación nos permite la posibilidad de comprobar que el análisis de los datos se realiza de manera correcta.

Para obtener las representaciones gráficas de los costes de red en los que se ha incurrido, así como del tráfico cursado en todas las redes tanto históricos como predicciones, se aplica a los indicadores clave de rendimiento diferentes metodologías para su cálculo y estimación. Con posterioridad, a partir de la información obtenida por estos métodos, se puede dar soporte a la decisión de asignación de inversiones en estas redes dentro de la empresa.

Finalmente, esta herramienta se ha realizado mediante el lenguaje de programación R tanto para el cálculo de los diferentes datos como para el desarrollo de la aplicación web para la representación gráfica de los mismos. Para mostrar este trabajo se han alterado los datos proporcionados por el grupo Telefónica para asegurar la confidencialidad de los mismos.

PALABRAS CLAVE

Indicador clave de rendimiento (KPI), análisis de datos, tráfico, costes, R.

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a todas aquellas personas, tanto de la Universidad Carlos III de Madrid como de Telefónica, que me han ayudado en la realización de este Trabajo Fin de Máster y a lo largo de todo mi tiempo como estudiante universitario. Concretamente me gustaría agradecer la ayuda y orientación de mis tutores, Adrián Maeso en Telefónica y Pablo Basanta en la Universidad Carlos III. Gracias a sus consejos y recomendaciones he logrado superar todas las dificultades y retos para conseguir desarrollar este Trabajo Fin de Máster.

Agradecer a todos mis compañeros y profesores que han hecho posible que haya podido realizar los estudios de Máster en Ingeniería de Telecomunicación en su edición bilingüe, ya que ha sido una gran oportunidad para mí tanto el poder formarme académicamente como también el formar parte de esta universidad. Además, muestro mi agradecimiento a todas las personas con las que tuve contacto en la Universidad de Valladolid durante mis estudios de Grado ya que gracias a ellos he obtenido una base de conocimientos para poder realizar este Máster y este Trabajo Fin de Máster.

Por otra parte, agradecer a mis padres todo su apoyo, su ayuda y su paciencia a lo largo de toda mi etapa como estudiante y de mi vida en general. También a todos los compañeros que he tenido en la Universidad, por todas las horas de estudio juntos que me han llevado a superar todos los obstáculos que aparecían en mi camino. En especial a mis amigos, tanto los conocidos en mi estancia en Madrid como en Valladolid y también agradecer el apoyo de todos mis amigos de mi ciudad natal, Ponferrada, sin ellos no hubiera podido alcanzar este momento.

A todos, muchas gracias.

Index

Table of Figures

Chapter 1

Introduction

1.1.   Motivation

Today, technology and capital investment are closely linked. The technology is regularly used by experts in the field of data analysis for the storage and usage of companies' information. In addition, it is used as an essential tool in order to know how well is the money being spent, as well as for the forecast of different traffic measurements in order to know what growth each network is expected to have.

Telecommunications companies seek to extract relevant information from these measurements of their networks in order to gain a competitive advantage over other companies by taking advantage of the usefulness of the information obtained, so they can draw business strategies in order to reduce their costs and maybe, as an indirect effect, increase their profits.

Currently, the processing of networks' data has been transformed into a primordial aid tool to support the decisions of investment allocations in the company's networks. A good method for investment on the networks could be improving their maintenance, besides with the right choose of CAPEX allocations the company would acquire cost optimization.

Lately, there is a trend of using programming languages like R in order to process all the available information of a company and convert these measurements into a useful data in order to take conclusions of them and to have a big picture of the company, so it can be known where there is a need to improve the network attending to the predictions of traffic usage. R is being used due to the fact that you can do data extraction, cleansing and processing, all in one script in order to get an analysis of all the measurements and information and to make representations in charts to analyze it and take some deductions.

The proposal of this project has been defined around this question, and it has been defined as the development of a statistical and graphical tool for network cost analysis, which allows giving support for CAPEX allocation in telecommunications companies.

The main motivation for this master thesis has been to apply the skills obtained throughout the master and bachelor's degree by being involved in a real project inside a company in the area of research and development.

This tool is going to show the evolution of networks' traffic using historical data and predict how is going to be in the following years in order to determine where the company should invest. Besides, networks' costs are going to be visualized and split into CAPEX or OPEX, countries, types of networks and concepts of these costs in order to analyze the deviations from the target cost inside the quarterly forecast closure (QFC), the budget (BGT) or the strategic plan (SP).

1.2.   Targets

Once the context of the work has been pointed out, the main purpose that this project is trying to obtain, that is shown in this report, is the analysis and the graphic representation of the key performance indicators of mobile and fixed networks of a telecommunications company. In addition, based on the results achieved with these analysis and representations, another purpose that we aim is the support to make decisions of investment allocations in the company's networks. For this purpose, a series of previous objectives are proposed:

- Description of the principles of network cost analysis. In this way, the necessary prior knowledge can be formed in order to discern what the costs of a network mean.

- Description of the principles of CAPEX allocation. In this manner, the essential knowledge can be gotten in order to know how to allocate CAPEX.

- Description of the principles of carried traffic in fixed and mobile networks. Like this, the required previous knowledge can be established in order to determine how the networks' traffic is evolving.

- Design of an analysis of the evolution of networks' traffic as realistic as possible and highly configurable in order to be able to adapt it to different studies that could be done with different telecommunications companies.

- Design of an analysis of the networks' costs of a company in order to be able to see where all the company's expenses are being spent.

- Design of a web graphic application in order to be able to make different comparisons of carried traffic between different countries, types of networks and evolutions amongst months and thus be able to appreciate the differences that arise in the study of the progression that traffic makes in very different situations.

- Design of a web graphic application in order to be able to make different comparisons of networks' costs between different types of cost such as CAPEX or OPEX, diverse countries or organizations, types of networks and deviations from the predicted cost inside the quarterly forecast closure (QFC), the budget (BGT) and the historical data.

- Graphic representations of the key performance indicators inside the web application in order to be able to study mobile and fixed networks from the perspective of costs and carried traffic.

- Development of a model to forecast the evolution of networks' traffic. A good predictive formulation will improve the traffic forecast and will be more realistic.

A complementary objective is to provide a guide of the methodology used throughout the development of this project, well described, documented and illustrated, so that it can be used in a simple way in future analyses with this tool that could be provided for the companies.

1.3.   Phases

This project is distributed in several phases. First of all, a study of the fundamentals of network analysis and CAPEX allocation, as well as a phase to study the technologies and software tools necessary to carry these analyses out.

The next phase deals with the utilization of historical network cost data in order to make the analysis. This development is composed by the use of quarterly forecast closure data, budget data, strategic plan data. From now on, QFC, BGT and SP. Besides these objectives or predictions, it is used the historical data in order to compare them all.

The final phase is to perform the representation of carried networks' traffic on fixed and mobile networks between different countries, types of networks and evolutions through the time. Besides, in the last phase the traffic forecast for the following three years is studied in order to know where the company has to invest and use its CAPEX efficiently.

1.4.   Resources

Lots of data are going to be used from different tables of a database in order to carry the study out and different technologies and software are going to employed to analyze all the data and to represent it in a web application.

1.4.1. Input data for analysis and representation

To carry out the study in this project, data from different tables of a database is going to be used, which has the following information: QFC data, BGT data, SP data and historical data of costs and mobile and fixed data. For example, fixed and mobile download and upload traffic intensity, total 2G/3G/LTE traffic, fixed broadband (FBB) and fiber to the home (FTTH) accesses in order to make the traffic forecast for the following three years.

In addition, there is a need to define some parameters to make the forecast such as FTTH consumption download and upload factor, percentage growth of uplink and downlink in the following years, LTE coverage forecast and more that are going to be used in the development of the project.

1.4.2. Employed technologies and software

This project has been done on a desktop computer that works with Windows 7 Enterprise 64-bit operating system. The computer has an Intel Core i5 processor at 3.20 GHz and 4 GB of RAM.

Besides, two servers have been used as back-end (named WTAC) and front-end (named Bazan) which work with CentOS operating system and have 2 TB of filesystem space each machine, 24 Intel Xeon CPU E5-2630 v2 processors at 2.60 GHz each of them and 132 GB of RAM for the back-end server and 66 GB of RAM for the front-end server.

All the scripts that have been developed for the execution of the analysis and representations of this project have been programmed in the R language. In order to use all the power of R, the RStudio environment has been used, which can be used through an application on an operating system or through a web browser.

For the representation of the results, the Shiny package of R has been used, which allows making web applications in an intuitive way, and they can be done without knowledge of HTML, CSS or JavaScript.

All the data is stored in a MariaDB database, and in order to access to this data the RODBC package of R has been used and to see the values and check all the queries made by the R code the HeidiSQL software has been employed as a structured query language client for these type of databases.

1.5.   Memory structure

The current project is established as follows: in the first chapter, the problem to deal with in this project has been introduced, as well as the motivation, resources and objectives.

Then, the second chapter deals with the foundations on data analysis, what CAPEX and OPEX mean and why they are important for a company and the importance in the art of CAPEX allocation is mentioned, this means how to invest the money in a proper way, as well as the significance of network and costs analysis.

The next step, discussed in chapter three, will be the foundations on the R programming language, its best environment to develop R code, which is RStudio and the Shiny package for R, which allows to elaborate web applications without any knowledge of HTML, PHP or other web technologies. There is some information about the structured query language (SQL) and the type of databases used for this project (MariaDB). In the last section of this chapter the client (HeidiSQL) is going to be studied, which will give the opportunity to execute queries and check that the R code accomplish the results that it is expected to get.

The fourth chapter explains the process followed for the accomplishment of network cost analysis thanks to the use of historical costs data and the comparison between those values and the forecast values given by the QFC, BGT and the SP. These values allow seeing if the money is being spent as expected or not, due to the fact that for a company it is important to have a prediction for each month and year costs.

Subsequently, in chapter five, it is a question of studying how carried traffic affects fixed and mobile networks on a telecommunications company by using the stored values in databases and representing them in graphics and tables so it can be compared country-by-country, monthly or yearly among other options.

Besides, in this chapter, it is also explained how the forecast for carried traffic is done, with which formulation and how these forecasts for the next three years can help investors to allocate their CAPEX in the best or needed network in order to achieve their aims, their purposes for the evolution of the company.

Finally, in chapter six, the time of development on the different areas of the project is shown, an economic budget is gathered for the accomplishment of this study, the conclusions obtained from the results of making and executing this tool, highlighting the main contributions of this project, which will correspond to the

targets pursued from the beginning of it and the proposal of possible future lines of work as a result of it.

Chapter 2

State of the art of network analysis and CAPEX allocation

2.1.   The usefulness of data analysis

Information fills the world nowadays. The challenge is to understand this data in a rational and functional way.

A good policy in order to make decisions about which business prospects to pursue is to learn to use all the available information. This policy will help to evaluate the viability and capacity to grow a business [1].

The best approach is to make a smart summary of the data collected, not to browse over a mass of raw numbers. It is implied the need to decrease the mass of values to a few illustrative measures that will provide an overall knowledge of what that data is describing and what deductions are inferred from those values [2].

A line of attack for investment is refining the network maintenance process and the renewal strategy in order to acquire cost optimization [3].

Companies need to analyze how their products are in order to use this information to make investments. For example, for telecommunications companies, fixed and mobile download and upload traffic intensities are relevant so they can know if the networks need an improvement or not, depending on this analysis.

Moreover, these companies have to check if all the countries and organizations of which they are composed are spending the CAPEX and OPEX in a correct way and adjusting their investments to the investment plans showed in the QFC, BGT or in the SP. Therefore, they could take some actions if these plans are not being followed as they should be.

2.2.   CAPEX and OPEX definitions

It is important to define capital expenditures (CAPEX) and operating expenditures (OPEX) in order to understand how the money is being spent in a company and to evaluate the profitability of its services. These type of expenditures are very different from each other.

On the one hand, capital expenditures impact on the fixed infrastructure of the company and they are devalued over time, but CAPEX are needed to enlarge the telecommunications services that the company can offer to its clients [4].

On the other hand, operational expenditures do not impact on the infrastructure itself and therefore they are not expose to devaluation. OPEX correspond to the cost of preserving the company operational and include costs for technical and commercial operations or administration [5].

Nevertheless, CAPEX and OPEX are interrelated matters. For example, a network technology letting perform a lot of preservation and provisioning tasks automatically, will possibly have higher acquirement cost (CAPEX), but will be low-priced to work (OPEX) [4]. So it is well defined that, for a given volume of equipment, the more operation is mechanized, the more employment costs can be set aside [6].

Lastly, the fact whether a current operator or a new competitor is being studied will have a significant influence on the ratio CAPEX/OPEX.

Most of the current operators have acquired their fiber infrastructure; this is taken for granted, so it is assigned to CAPEX, while new competitors possibly will rent their infrastructure.

2.3.   Importance of CAPEX allocation in telecommunications companies

It is important to describe the precise allocation of capital expenditures. CAPEX optimization is accomplished by finding the correct balance among maintenance and renewal activities, likewise the network installation plan and selection of suitable network components. This optimization must be grounded on robust principles [7].

These principles can be as diversity as the average lifespan of equipment, the localization of the equipment and the access limitations as antennas located in isolated regions [8].

Due to the rising request for consistent service delivery, fault supervision is turning out to be more important in all parts of communications networks. Nevertheless, there is a trade-off among the protection costs and the level of service reliability. Enhancing reliability performance by doubling network resources and capital expenditures would be too expensive [9].

Capital expenditures are founded on three categories for a network operator or network service supplier: network infrastructure like buying equipment, network software like network management systems and non-telecommunications specific infrastructure like purchasing of property and structures [4].

The allocation of capital expenditures consists on identifying where capitals can be released for other developments and establishing clear accountability among others. Making these alterations to an old allocation plan is neither stress-free nor fast; however, the rewards will pay it off [10].

To avoid a deterioration in the quality of service, an appropriate resource allocation procedure need to be set up [8].

Usually an investment scenario is supported by the need of it as a consequence of the network's congestion. This can happen, since capacity problems are a fact for all telecommunications operators. Nonetheless, maybe this investment can be put off in order to invest in a more vital scenario.

The main issue in giving anywhere, anytime mobile access is the towering infrastructure cost as it is fundamentally proportional to the bandwidth offered. The infrastructure cost is not dominated by electronic equipment, but rather by other related costs such as towers, wiring, building and maintenance costs [11].

In order to make a decision about allocation, statistical data analysis has been employed [12]. With these analyses, the data can be transformed into significant and valuable information that will be used to allocate appropriately CAPEX.

Statistical data analysis has to be used because poor decision-making can cost a lot, for example, sometimes CAPEX organization is determined by technological considerations rather than business purposes [13].

2.4.   Analyzing the available information

It is important to accomplish viable improvements and in order to do that an effective approach is required. Therefore, network costs can be optimized on a long-term basis.

Decomposing the available information helps the company to invest its assets where they are most necessary. Besides, with that information the company can be ready for the future in a more successful way.

In order to do that, a web application is going to be designed to be able to make different comparisons of networks' costs between CAPEX and OPEX, diverse countries or organizations and to observe graphically deviations from the predicted cost inside the QFC, BGT or SP. The interaction among all of the different costs accentuates the importance of good planning.

Besides, with some graphic representations of the key performance measurements of traffic, fixed and mobile networks are going to be able to be studied from the perspective of carried traffic and the evolution of these networks' traffic is going to be forecasted.

Costs can be divided into fixed and variable. Variable costs will fluctuate in ratio to transformations in the level of an activity. Fixed costs are important in order to evaluate the scalability of a business proposal. There are many types of fixed costs, for example, salaries, taxes, networking infrastructure maintenance and more [1].

The technique for this is to coordinate operations in order to get the most out of the profits of each variable and fixed cost. However, revenue opportunities also have to be considered [1].

A tool for representing all the data obtained are charts. They are effective for detecting problems or remarkable data points in the data set. When a fact has to be clarified, they are very useful [2]. Charts are supposed to be used as an alternative to tables when there are too many data to be showed or to be understood.

2.5. Network and costs data

There is a lot of information about a company, which is why there is a need to understand some key performance indicators (KPIs). The BGT and SP report some of these KPIs for the future based on objectives of the operators.

On the one hand, information of fixed and mobile networks is available with some KPIs. These KPIs can be classified in two groups: fixed and mobile. For fixed networks, there are key performance indicators such as installation activity, disconnection activity, and for mobile networks, there are KPIs such as size, rollout, and towers.

There are some KPIs that will give information about both types of networks such as customer, traffic, capacity, network fault management, quality, energy and others.

On the other hand, information of costs is available, they can be divided in CAPEX and OPEX, and inside these divisions there can be KPIs about access, infrastructure, control, platform, transportation, efficiency, rental, energy, external contracts, radio frequency licenses and more, both for mobile and fixed networks.

2.5.1. Fixed and mobile networks

At this point, measurements of the different fixed and mobile networks are being taken by different operations support systems (OSSs) such as deep packet inspection DPI technologies, which make complete packet inspection and then they extract all the possible information from these inspections. There are many companies offering these services, such as Cisco, Huawei or Sandvine among others.

Many of the traffic measurements are referred to the "Busy Hour" (BH). The BH is generally understood as the hour in which the traffic level measurement is

representative of the maximum load that the network has to support maintaining the adequate grade of service [14]. Nevertheless, there is not a unique methodology to determine the value for the Busy Hour, since it depends on several factors such as maturity of the service, variability of the daily traffic patterns, variability of traffic inside the day and situations "out of norm", for example massive calls in special days like Christmas or New Year's eve.

This has been reflected in different BH methods that have been recommended historically in ITU-T Recommendation E.500 [14]. These evolutions, and the local conditions of each operation, have led to the present situation, in which there is just one method for calculating the BH in different operations and for different services inside a company.

There are some specific remarks for fixed and mobile networks about Busy Hour measurements. BH in fixed networks is not expected to have great differences among the different methods used to calculate the traffic measurements. For BH in mobile networks at the network level the method proposed by the Radio Design Council inside the company is being used: the network level aggregates weekly maximum value averaged across the weeks of the month and a week is considered part of a specific month if four days or more fall within the month in consideration.

The agreement regarding this method has been reached after analysis of different possibilities within the different operations in Telefónica around the world and against actual network traffic. It is offered to be a reasonable compromise between consistent growth in traffic without matching sporadic traffic peaks both at time and cell levels.

2.5.1.1. Fixed Broadband Clients

The objective is to measure the accumulated number of connected properties using each fixed access technology. This measurement is applied by fixed operators in unitary.

This KPI refers to the number of properties that are connected and in service by a given fixed access network. Technologies that have to be considered are the following: ADSL, ADSL2+, Bitstream, VDSL2 from central office, FTTN/B with ADSL2+, FTTN/B with VDSL2, FTTH connected properties and Cable (DOCSIS1/2 and DOCSIS3) [15].

ADSL refers to "Asymmetric Digital Subscriber Line"; ADSL2+ extends the capability of basic ADSL; Bitstream is the service provided by a third party for both

ADSL and VDSL; VDSL2 denotes "Very-high-bit-rate Digital Subscriber Line" of second generation; FTTN symbolizes "Fiber-To-The-Node" or "Fiber-To-The-Neighborhood"; FTTH indicates "Fiber-To-The-Home" and DOCSIS is the "Data Over Cable Service Interface Specification".

The recurrence of this measurement is by the end of period, and it can be the end of year or month.

2.5.1.2. Fixed Data Traffic in Busy Hour of downlink/uplink

The objective is to measure the traffic in the busy period for the fixed data network in downlink/uplink direction (network to access or access to network), for dimensioning purposes. This measurement is applied by fixed operators in Gigabit/s (10^9 bits per second).

This KPI refers to the maximum of the sum of the downlink/uplink traffic values registered in all points of presence (POPs) during the network wide Busy Hour, which may or may not be coincident with the Busy Hour in each POP.

It includes all types of data traffic: residential broadband and business data [15].

The recurrence of this measurement is by the end of period, and it can be the end of year or month. In the case of yearly reporting, the maximum value of the monthly values is reported.

2.5.1.3. Mobile Data Traffic Intensity in Busy Hour

The objective is to measure the traffic in the busy period for all mobile data networks in both directions (uplink and downlink), for dimensioning purposes. This measurement is applied by mobile operators in Gigabit/s.

This KPI refers to the traffic intensity in the BH, which is the maximum of the sum of the uplink and downlink cell level traffic values registered in all radio access elements during the network Busy Hour, which may or may not be coincident with the Busy Hour in each radio access element. It is measured for 2G, 3G and LTE [15].

The recurrence of this measurement is by the end of period, and it can be the end of year or month.

2.5.1.4. Mobile Data Traffic volume

The objective is to measure the accumulated usage of all mobile data networks in both directions (uplink and downlink). This measurement is applied by mobile operators in Terabytes.

This KPI refers to the total monthly traffic volume (downlink and uplink), which is the sum of the downlink and uplink traffic volumes at the POPs level registered in all the Node B of the network during the month [15].

The Node B refers to the node that provides the connection between mobile phones and the wider network.

The recurrence of this measurement is by the end of period, and it can be the end of year or month.

2.5.1.5. Outdoor Population Coverage

The objective is to assess how much of the static population could in theory have access to the operator's services. This measurement is applied by mobile operators in percentage of population in the country.

This KPI refers to use standard coverage planning parameters, which means the proportion of the population in the country under consideration that is capable of accessing the operator's services outdoors.

The population coverage is provided for the following technologies: 2G (voice service), 3G (voice service), HSPA (at least 500 K Kbps service) and 4G (at least 1Mbps service) [15].

The recurrence of this measurement is by the end of period, and it can be the end of year or month.

2.5.2. CAPEX and OPEX data

For CAPEX and OPEX available information, two concepts have to be taken into account: Year-over-Year (YoY) and Year-to-Date (YTD).

On the one hand, Year-over-Year is the comparison with the same period on the previous year. On the other hand, Year-to-date is the period of time starting from the beginning of the current year continuing up to the current day.

CAPEX and OPEX can be divided in access, infrastructure, control, platform, transportation, efficiency, rental, energy, external contracts, radio frequency licenses and more, both for mobile and fixed networks.

In the access classification, there are some values about 2G, 3G, single radio access network (RAN), CDMA, LTE, Wi-Fi or efficiency.

In the infrastructure classification, there are some values about buildings, refrigerants, energy and others.

In the control classification, there are some values about circuit switched cores, packet switched cores, evolved packet cores (EPC), next generation networking (NGN) or physical control fields (PCF) among others.

In the platform classification, there are some values about data, global services, messaging or voice.

In the transportation classification, there are some values about asynchronous transfer mode (ATM), X.25, frame relay (FR), circuit connections, fiber, backhaul, IP, plesiochronous digital hierarchy (PDH), photonic, synchronous digital hierarchy (SDH) or satellite hubs.

In the rental classification, there are some values about own sites, shared sites and other technical buildings.

In the energy classification, there are some values about data centers, sites and offices.

In the external contracts classification, there are some values about customer service, consulting service, camp maintenance, network operation management center, support agreement, spare parts and warehouse management, supervision and other activities.

In the radio frequency licenses classification, there are some values about satellites and national spectrum.

There are some other categories for the costs such as human resources, indirect costs, maintenance and more.

In short, the CAPEX table as follows.

CAPEX				
Mobile networks			**Fixed networks**	
Access:	2G		Access:	Cable
	3G			DSL
	Single RAN			FTTB
	CDMA			FTTN
	LTE			FTTH GPON
	Wi-Fi			Fiber
	Efficiency			Efficiency
Infrastructure:	Buildings		Infrastructure:	Buildings
	Refrigerants			Cooling
	Energy			Energy
	Other			Other
Control:	CS Cores		Control:	IMS
	PS Cores			PSTN
	EP Cores			NGN
	NGN			PCF
	PCF			Other
Platform:	Data		Platform:	Data
	Global services			IPTV
	Messaging			Messaging
	Voice			Voice
Transportation:	ATM/X.25/FR		Transportation:	ATM/X.25/FR
	Circuit connections			Circuit connections
	Fiber			Fiber
	Backhaul			Backhaul
	IP			IP
	PDH			PDH
	Photonic			Photonic
	SDH			SDH
	Satellite hubs			Satellite hubs
…	…		…	…

Figure 1: CAPEX classification.

It is shown the OPEX classification too in the following table.

OPEX				
Mobile networks			**Fixed networks**	
Rental:	Own sites	Rental:	Technical sites	
	Shared sites		Other	
	Other technical buildings	Energy:	Data centers	
Energy:	Data centers		Technical sites	
	Sites		Offices	
	Offices	External contracts:	Customer service	
External contracts:	Customer service		Consulting service	
	Consulting service		Camp maintenance	
	Camp maintenance		Network operation management center	
	Network operation management center		Support agreement	
	Support agreement		Supervision	
	Supervision		Other activities	
	Other activities	OSS:	SEP	
RF license:	Satellites		Other	
	National spectrum	Human Resources:	Network support	
Human Resources:	Network support		Planning	
	Planning		Maintenance	
	Maintenance		Efficiency	
…	…	…	…	

Figure 2: OPEX classification.

2.5.3. Quarterly Forecast Closure (QFC), Budget (BGT) and Strategic Plan (SP)

This subsection is going to explain the target values of the company's costs from the less important to the most important one.

First, the strategic plan (SP) has to be taken into account, which is a forecast of all the costs for the following 3 years. It can be seen as the targets that the company wants to reach in these next years.

Then, the budget (BGT) has to be taken into account, which is a forecast of all the costs for the following year and it is an adjustment of the SP.

Finally, the quarterly forecast closure (QFC) has to be taken into account, which is a forecast of all the target costs at the end of the current year, it is revised twice throughout the year and it is an adjustment of the BGT, so these values would be the more realistic ones.

With all of these targets or forecasts, the impact of major and minor alterations to the plan that the company has in mind can be understood. Their determination is to deliver an image for the upcoming time and support better choices in order to allocate CAPEX and invest.

However, these forecasts cannot be used by supervision part of the company as an instrument for re-evaluating performance marks.

It has to be transmitted that there is a need to have forecasts and objectives separate in order to obtain success inside the company. Forecasts allow the company to analyze threats and chances to invest and to take counteractive acts, which may be in disagreement with the objective but it is going to be beneficial for the company.

Chapter 3

State of the art of R, RStudio, Shiny and SQL

3.1.   R programming language

Long time ago, data analysis, visualization and the treatment of intricate statistical problems were set aside just for universities and limited organizations. R was born in the nineties and since then it has been getting better. R has been used above all in data science and it has been the most used programming language for this area because it has an open source nature [16].

R is a dialect of S, which is a language that was created by John Chambers and others at the old Bell Telephone Laboratories. S was created to be used as an internal statistical analysis environment in 1976. In fact, initial versions of S did not even have functions for statistical modeling. It is significant that the S language had its origins in data analysis due to the fact that its creators were fixated on making data analysis easier for themselves and for others [17].

The R language was developed due to the limitations of S. One of these limitations was that it was only accessible in a commercial package. Nevertheless, R can run on nearly any platform and operating system. The R open source nature indicates that you can adjust the software to any kind platform for free. As a matter of fact, R has been reported to be able to run on tablets, smartphones and game consoles [17].

As for the year 2017, R is rising and acquiring more relevance among data scientists. The R programming functionalities can be improved with the installation

of packages and libraries. You can cover almost everything you can possible imagine with over 7,000 packages that R has right know.

In the area of data analysis, you can do data extraction, cleansing and processing in order to get an advanced analysis, create a model data or picture this data in graphics and charts [18]. That is why R is describes as a programming language and environment for statistical computing and graphics.

On the one hand, the main characteristic of this programming language is that it is object oriented, this means that it is constituted mostly of functions and objects. Besides, R is continuously being improved by users' contributions thanks to the publishing of their R scripts or thanks to their packages creations [18].

On the other hand, the main difference with classical programming languages such as Java or C is that in R you do not need to compile, there is no compilation.

As can be seen in the following figure, the graphical user interface of R is very simple.
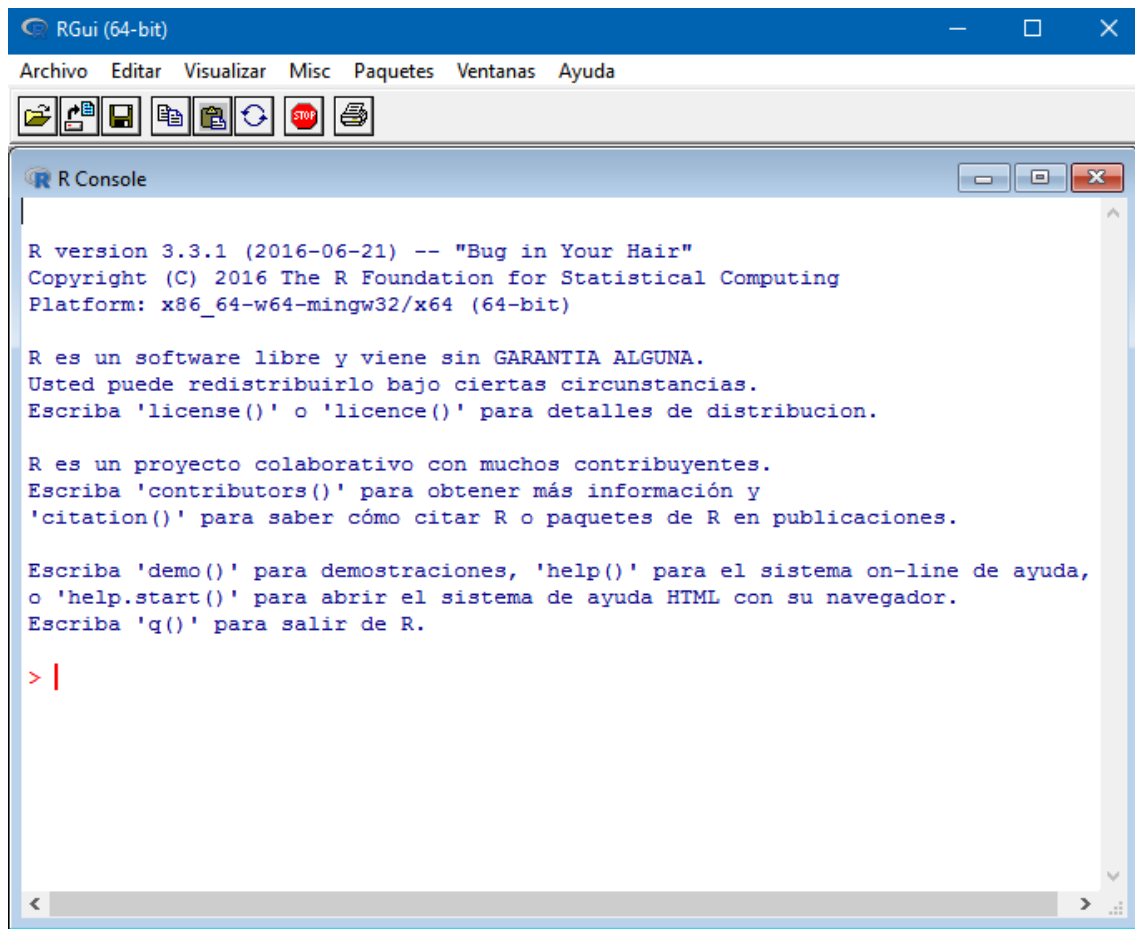
Figure 3: R Graphical User Interface.

3.2.   RStudio environment

In order to develop code in R language, the R application does not offer an easy environment. That is why an Integrated Development Environment (IDE) is needed to develop scripts in a more comfortable way.

One of the main factors of the success of R is the accessibility of a sophisticated Integrated Development Environment (IDE) named RStudio. The development of this environment was began in 2010 and it has been converted in the de facto integrated development environment for everyone using R. The main objective of RStudio is "to provide the most widely used open source and enterprise-ready professional software for the R statistical computing environment" [16].

This RStudio environment delivers more than just an easy way to develop R code in scripts; it has been developed in order to have a diversity of functionalities such as packages, applications and more. Having these characteristics, the RStudio environment has accomplished to take data analysis to a bigger audience thanks to all of this easiness [16].

RStudio joins the robust statistical capacity of the R programming language, all the users and developers that use R and the open source nature with new tools of user interface development. This is what made RStudio more than another environment to develop R code; it turns out to be the platform for everybody who deals with data [16]. Due to its unceasing wish to revolutionize the R language and all of its potentials, it could be perceived as an additional development of R.

RStudio environment is suggested when someone has to create web pages or web applications in R programming language with the Shiny package due to the fact that this environment holds features particularly intended for this reason.

Furthermore, RStudio offers capabilities to write in other programming languages such as C++, Latex, or HTML and fit in all of these languages to the R code. RStudio also provides version control, project management, and debugging characteristics amongst several others [18].

The RStudio project delivers most of the looked-for characteristics for an R environment and it makes the R language easier to use and it gives more efficiency.

Some of these looked-for characteristics that everyone wants in an IDE are the following: the main components of this environment are all well combined into a four-panel layout that includes a console for interactive R sessions, a tabbed

More information about RStudio [24].

source-code editor to organize a project's files and panels with the environment available variables and the packages installed and used.

The source-code editor is feature-rich and integrated with the built-in console and they are strongly connected to R's internal help system. Besides, in the RStudio environment you can set up different projects and change amongst them with just a click [17].

Another characteristic is that RStudio delivers numerous administrative tools for managing packages, the workspace among others. This environment of R programming is offered for various operating systems such as Windows, Mac OS X, Ubuntu and Fedora [19]. Besides, it can be run through a web browser for remote access with RStudio Server.
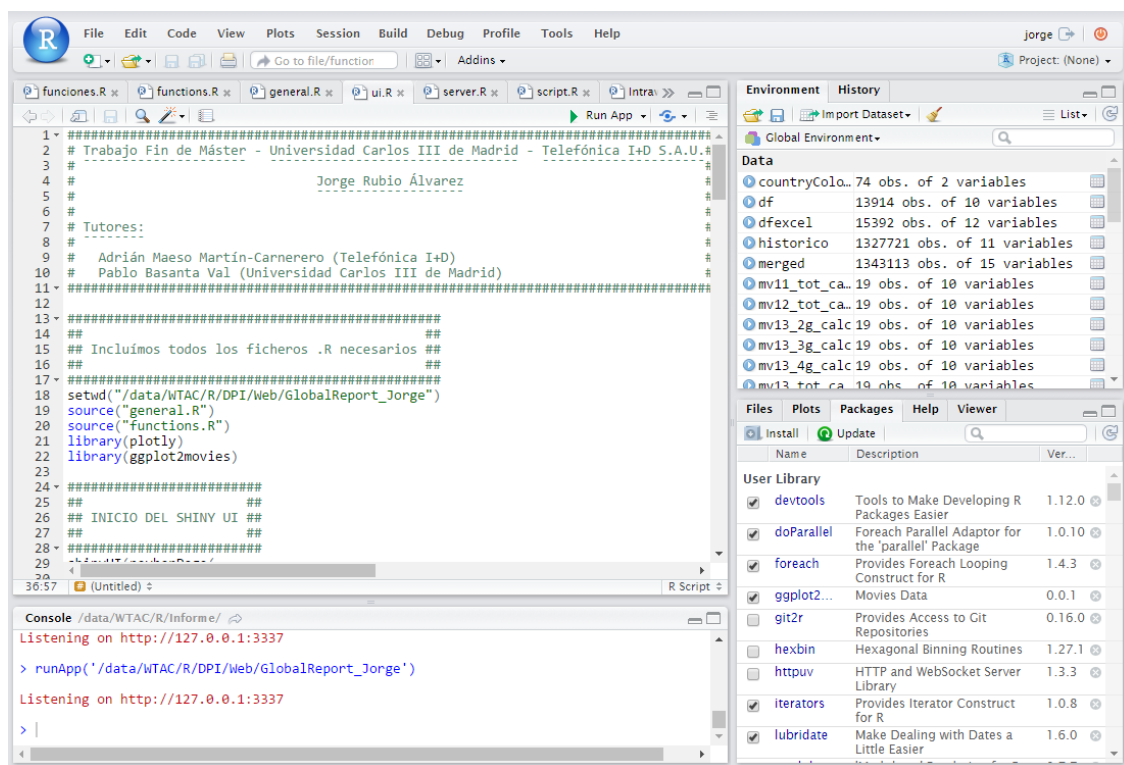
Figure 4: RStudio environment on a web browser.

All of these characteristics can be observed in the following figure of RStudio running on Windows 7.
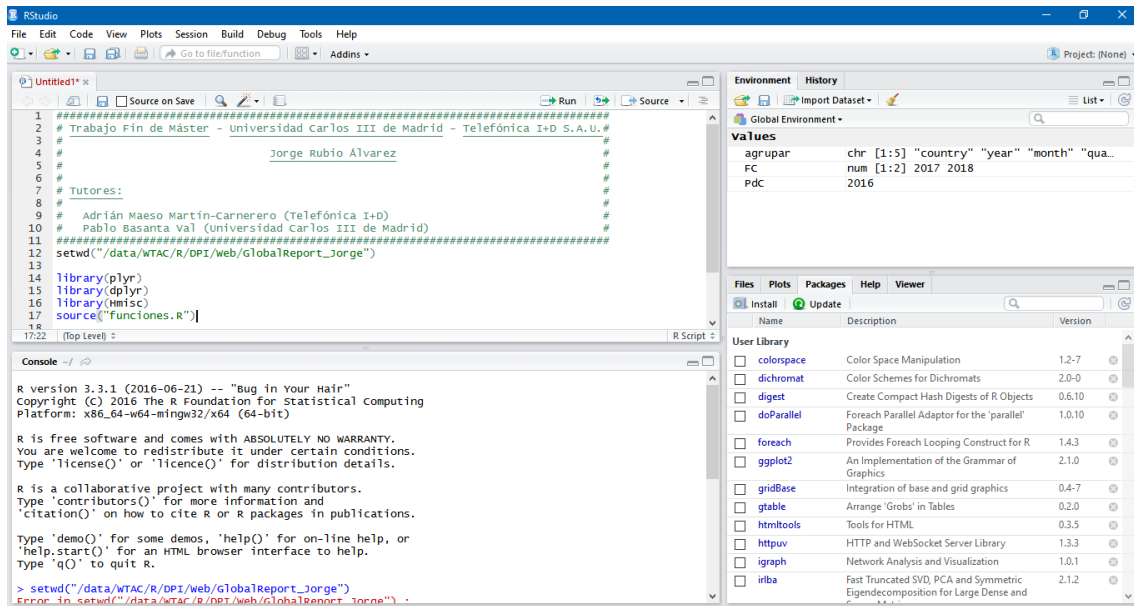
Figure 5: RStudio environment on Windows.

This environment performs equally throughout all platforms thanks to the use of different web technologies in its design. Therefore, programmers can have a feature rich and constant environment for R [19].

Moreover, all of these characteristics are free. You can pay a commercial license to use the RStudio Server Pro, which gives you more options such as enterprise security, project-sharing management, the use of multiple R sessions and different versions of R, administration dashboard to see what users are using the environment and a tool for load balancing. Telefónica uses this commercial version and it is the tool used for this project although it can be executed in an RStudio free version or in another R environment.

3.3.   Shiny package

Shiny is the package of R that can make a code improve due to the fact that you will be able to share all the results of R code with anyone through a web application. With Shiny, you can see hidden information of your data so it is not just a dashboard tool and it can be used without any HTML knowledge although it can help experts even for no expert users [18]. This means that using the Shiny package you can deliver applications and give access to everyone to your R data developments.

More information about Shiny [23].

RStudio inventors created this package. That is why Shiny allows to effortlessly using R with a web browser.

As specified in its documentation, Shiny is a web application framework for R that makes it easy and effortless to create interactive web applications with the use of the R programming language [18].

The first thing to know in order to develop a Shiny web application is that it has two components: a user interface script which manages the layout and the appearance of the application by converting R code into HTML code, and a server script which holds the code needed to build the application [18]. That is why there is no need to know about HTML, JavaScript or CSS but you can use them in order to take advantage of them. Because of that, every Shiny application has two R scripts named ui.R and server.R and they are in the same directory.

How often the R code that gets the results displayed on the web application is run rests on where this code is placed within the server script.

If there is a need to use some of HTML, JavaScript or CSS characteristics, you can look first through the Shiny framework due to the fact that it already covers prebuilt features that include the frequently used functions in a web application of these characteristics.

In order to decide to use Shiny the main advantages of using the combination of R and this package had been looked for, which are the following: the use of R and the Shiny package is free in terms of money and in terms of freedom and free software is a matter of importance to the people who use this software due to the fact that they can run, copy, distribute, study, change, and improve it [18].

Besides, with a web application created by a Shiny package you can have all the functionalities that R has. You can make a web application that will be able to display any content that your R code has generated which is great because you can run complex statistical models and return the output in a friendly way with graphics and charts in a web application [18].

Furthermore, thanks to the power of R, you can get and mix data from numerous sources and formats such as CSV, XSLX, SQL or JSON in order to use it all together to process all the information and extract results from it.

3.4.   SQL database language

SQL is named after Structured Query Language and it is a database language designed for managing data in relational database management systems, which is the type of databases that it is going to be used in this project. SQL was originally created by IBM for querying, altering and defining relational databases [20].

There are lots of functions of SQL, which make it one of the most used languages for interacting with databases.

Some of the main characteristics of SQL are the following: it can perform queries against a database and retrieve data from it and it can insert values in a database, update or delete them. In addition, SQL can create new databases and tables inside a database with the permissions that we want to give to these tables [20].

There are many different database systems such as MySQL, which is free, Microsoft Access, IBM DB2, Oracle or Microsoft SQL Server among others.

A point to take into account is that several database systems apply their own version of the Structured Query Language although it is a standard and it should be the same for every database system.

The Data Manipulation Language (DML) is the language of SQL that helps to add, update and delete values from a table in a database [20]. Some of the main functions that is a requirement in a relational database are the following: create implemented by the function "INSERT INTO" which inserts new values into the database, retrieve implemented by the function "SELECT" which obtains values from the database, update implemented by the function "UPDATE" which renews the values that are available and destroy implemented by the function "DELETE" which removes values.

There are some features that can be applied to tables or columns of a table such as the constraint "UNIQUE" which distinctively recognizes each value in a table. In order to provide an assurance for individuality for a column or set of columns you would have to use some of these SQL constraints.

Most SQL servers contain a database engine which is a service running in the background of the server and a management tool which is a graphical instrument in order to configure and see all the available information of the databases [20].

3.4.1. MariaDB database

MariaDB is an improved fitting substitute for MySQL created by numerous of the previous essential creators of MySQL. MySQL grew out of a necessity in the early 1990s for a quick and adaptable database for use in web-based applications [21]. MySQL database is stable and reliable and it is being used by lots of companies.

The likenesses between MariaDB and MySQL are a lot due to the fact that MariaDB is founded on the open-source MySQL code, which means that MariaDB is a fork of the MySQL source code. Developers took the freely accessible MySQL code and then added to it. In fact, as variations are made to MySQL, the MariaDB developers evaluate and integrate those equivalent variations into MariaDB [21].

MariaDB has some characteristics, performance enhancements, improved testing and bug fixes that differentiates it from MySQL. Some of these were developed by the MariaDB developers and others come from companies such as Facebook, Google and others [21]. The developers have also developed some NoSQL-style functions.

You can be able to comprehend the server activity better and to find the causes of your database's load thanks to the INDEX and TABLE statistics that MariaDB integrates. Besides, the upgraded optimizer in MariaDB accomplishes meaningfully quicker than MySQL on complex assignments [21].

The increasing result of the extended and boosted MariaDB test set-up is that errors are observed earlier and resolved quicker so everything is widely tested long before they are published [21].

In consequence, MariaDB is going to be employed as the main database in this Telefónica's project due to the fact that it is going to have all the good characteristics that MySQL has, as well more features, performance enhancements, improved testing, and less errors in the databases.

3.4.2. HeidiSQL client

HeidiSQL is a suitable instrument in order to manage some of the most popular databases such as MariaDB, MySQL, Microsoft SQL or PostgreSQL. This software is going to be used in this project due to the high capacities that it enables

More information about MariaDB [25] and HeidiSQL [26].

such as browse and edit data, create and edit tables, views, scheduled events and more [20]. Besides, with this SQL client you can transfer the data of the database to the clipboard, to a SQL file and other options.

HeidiSQL runs satisfactory on Windows 10, which is the main operating system in this project, but it also works well on Windows XP, Vista and 7 and on Linux through the use of Wine.

The main characteristics of HeidiSQL is that it is free and open source. Some of the functions that you can use with this client are the following: connect to multiple servers in a single window or connect to them via command line through a SSH tunnel, create and edit tables and values, generate SQL file exports of the tables and data on the databases, compress them or place them on the clipboard, import text files and export table rows as CSV, HTML, XML, SQL, LaTeX, Wiki Markup or PHP Array [20].

Furthermore, with this client you can manage user privileges, write SQL queries with syntax highlighting and code completion. A great characteristic is that it can help you to find a text in all tables of all databases of one server [22]. It can be seen the HeidiSQL interface in the following figure.
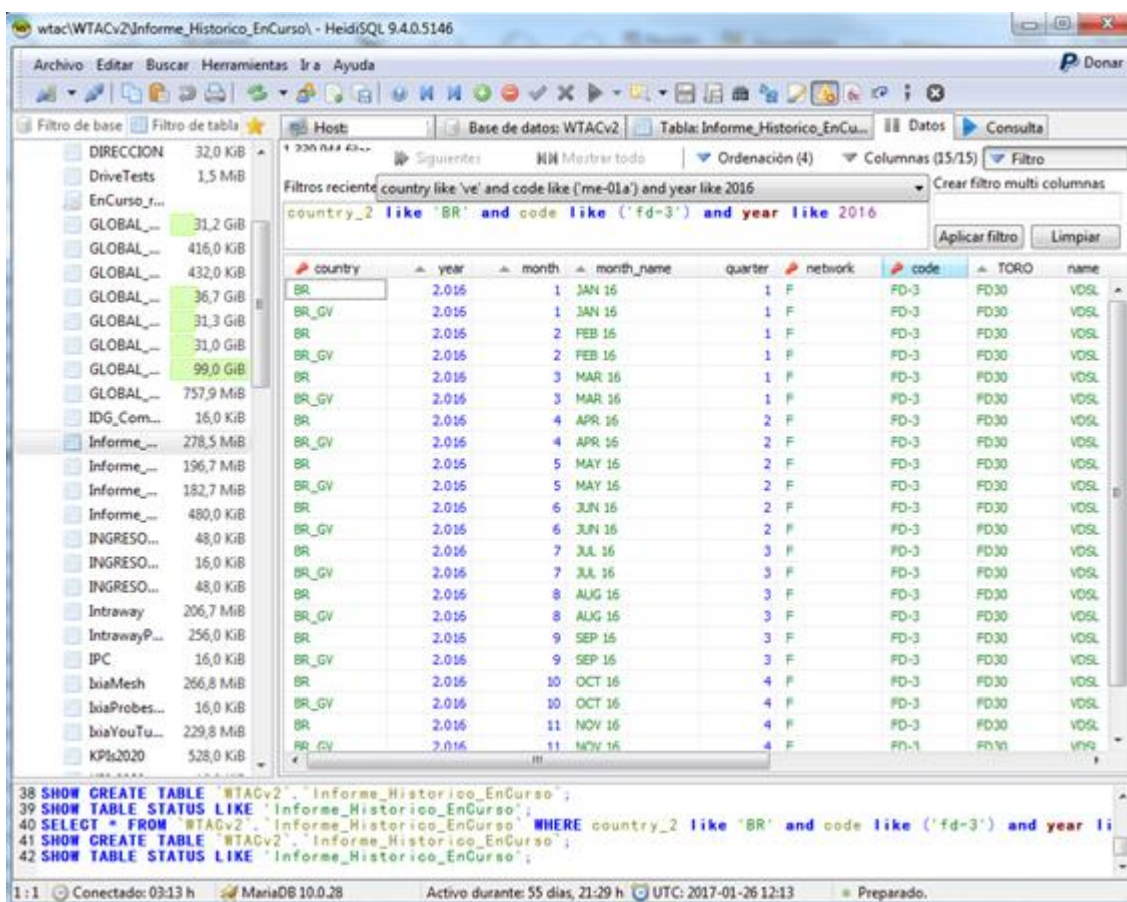
Figure 6: HeidiSQL interface.

It can be seen on the left of the interface all the databases and their structures; on the top right we can go through the objects of the current database. On the lower part of the interface, it can be seen the last commands that the HeidiSQL client has made to the database and the results of those commands or queries are presented in the central part of the window.

Chapter 4

Network costs

4.1.   Obtaining costs data

The CAPEX and OPEX information has been obtained from MariaDB databases where is all stored. Due to confidentiality, the data has been anonymized for the figures and tables that are going to be shown after. However, it has been previously tested that the web application showed every data correctly with the true values of Telefónica's costs.

4.1.1. Input data

There are three tables in the database: "ACTUAL_VALUES", "QFC_VALUES" and "BGT_VALUES" that correspond to the actual costs, the QFC costs and the BGT costs prediction. In order to get all the fields that are requested, some calls to the database are going to be made inside the server.R script of the Shiny web application just like the following code.

```
odbcDataSources("all")

# We chose the WTAC connection
WTACv2 <- odbcConnect(dsn = "WTAC", uid = "user", pwd = "password")

# We import all data from the database
queryACTUAL_VALUES <- paste("select COST_TYPE, RED, COUNTRY, YEAR, MONTH, EURO, DATETIME, CONCEPT from
ACTUAL_VALUES GROUP BY COUNTRY, CONCEPT, YEAR, MONTH")
ACTUAL_VALUES <- as.data.frame(sqlQuery(WTACv2,queryACTUAL_VALUES))
```

```
queryQFC_VALUES <- paste("select COST_TYPE, RED, COUNTRY, YEAR, MONTH, EURO, DATETIME, CONCEPT from QFC_VALUES
GROUP BY COUNTRY, CONCEPT, YEAR, MONTH")
QFC_VALUES <- as.data.frame(sqlQuery(WTACv2,queryQFC_VALUES))

queryBGT_VALUES <- paste("select COST_TYPE, RED, COUNTRY, YEAR, MONTH, EURO, DATETIME, CONCEPT from BGT_VALUES
GROUP BY COUNTRY, CONCEPT, YEAR, MONTH")
BGT_VALUES <- as.data.frame(sqlQuery(WTACv2,queryBGT_VALUES))

# We disconnected the connection with the database
odbcClose(WTACv2)
```

Figure 7: R code to retrieve costs data from the database.

Now all the information is available in R data frames, which are easily manageable and ready to show in any part of the Shiny web application.

4.1.2. Quarterly Forecast Closure (QFC) and Budget (BGT) data

This information about costs are predictions made by the company in order to estimate what each operator/country should spend in the following months or years and in what concepts this money should go.
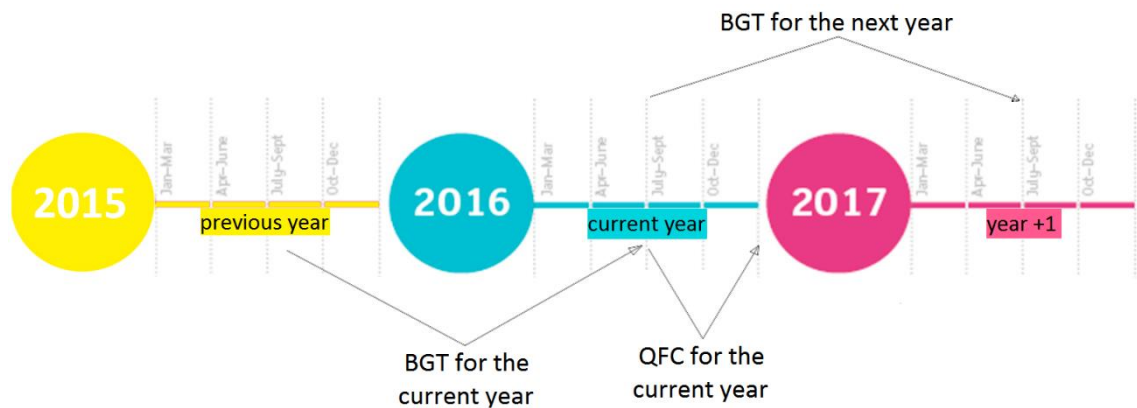
Figure 8: Quarterly Forecast Closure and Budget in a timeline.

Therefore, comparisons are going to be made between the actual values of, for example, the year 2016 against the predicted values of the QFC of this year and the BGT that had been estimated on 2015 in order to have a prevision of the costs of 2016.

4.2.   Understanding costs definitions

A document has been defined in order to make understandable all the concepts relative to costs to everybody who uses this web application.

In the document are explained some definitions such as Year-over-Year (YoY) or Year-to-Date (YTD) that have been explained in this report in the state of the art. It is also defined how CAPEX and OPEX are divided in access, infrastructure, control, platform, transportation and more concepts and that inside each classification, there are some subdivisions such as in access classification there are costs values about 2G, 3G, etc.

In the following figure, it can be seen how you can download this document from the "Tools" tab of the web application making a click in "Download definitions".
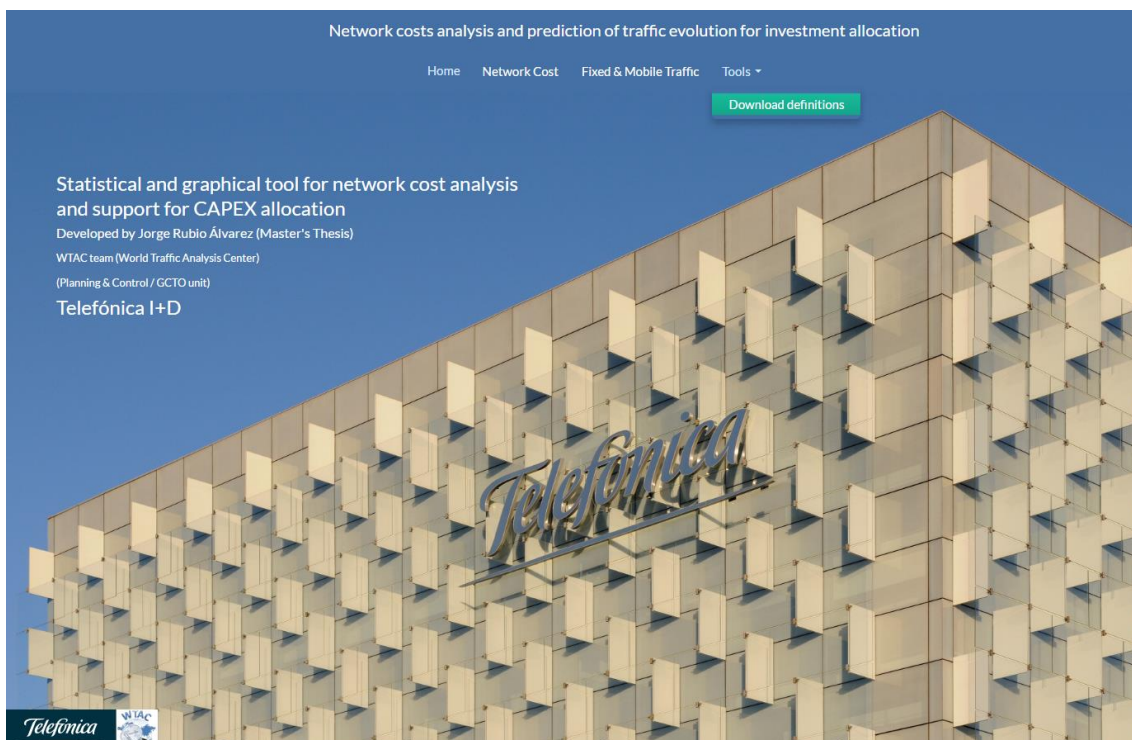
Figure 9: Tools tab to download definitions from the web application.

Inside this page a button to download the definitions document has been defined as can be seen in the following picture.
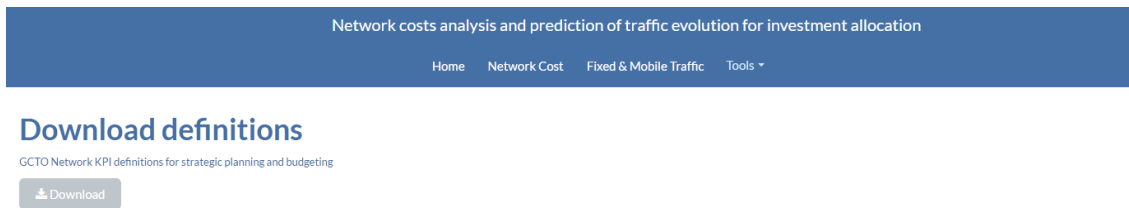
Figure 10: Download button of definitions document in the web application.

In this document has been included some definitions about KPIs of traffic too in order to have all the information in one only document, but two documents could have been put and two buttons to download them separately.

4.3.   Distinction of different types of costs

Inside each analysis of costs in the web application, it is going to be able to filter or to gather costs information about different types.

It can be seen each type of data filtering in the following picture.
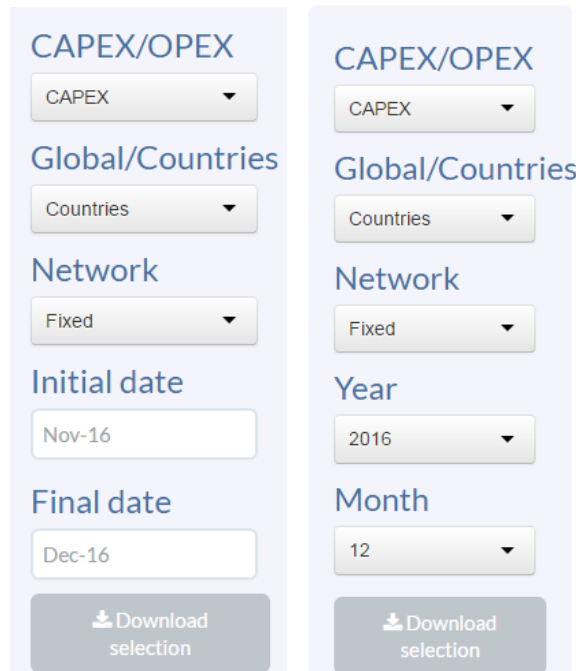
Figure 11: Costs filtering inside the web application.

Costs data can be filtered by CAPEX or OPEX; by country, organization or the global company's costs; fixed or mobile network and also by year and month or a range of dates in the case of costs summary.

Moreover, inside tables and graphs, they can filter by concept or classification of costs such as energy, infrastructure and more.

4.4.   Representation in the web application

In order to access to all the data in order to make analysis and comparisons of the costs of the telecommunications company, we are going to click in the "Network Cost" tab of the Shiny web application as can be seen.
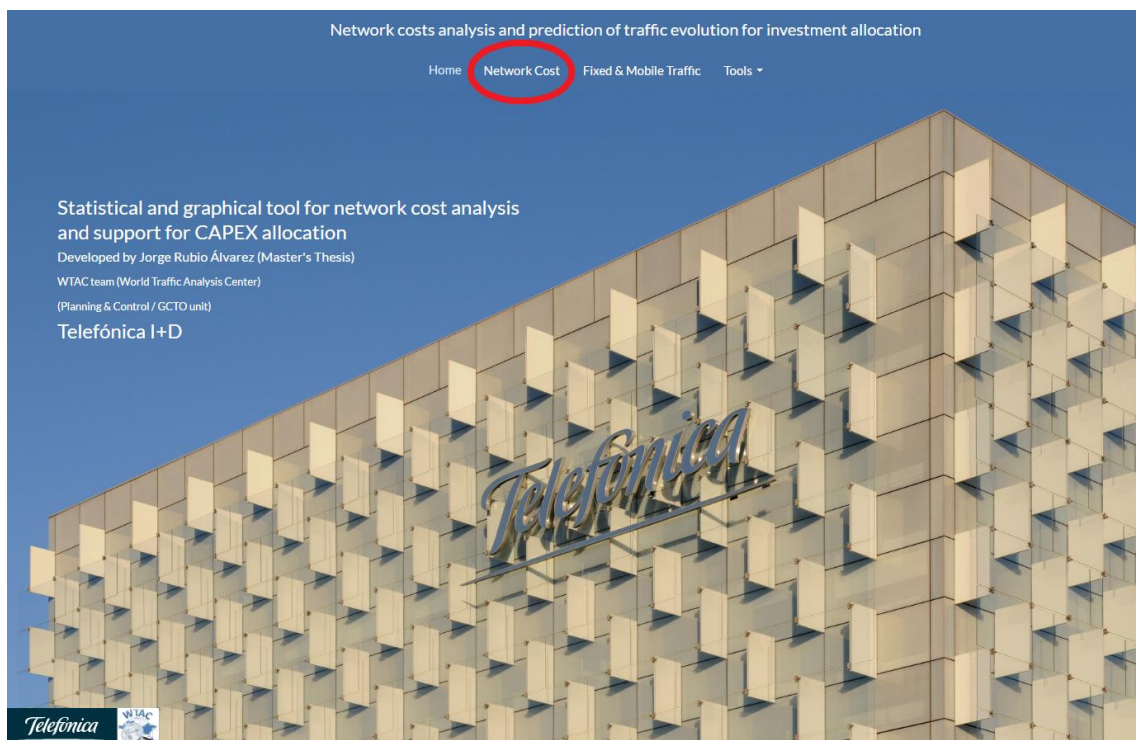
Figure 12: "Network Cost" tab view from the index webpage.

This costs section has been divided into six subsections: summary costs, executed and deviations costs by concept, global executed and deviations costs, executed versus QFC and BGT costs graph, CAPEX and/or OPEX by concept graph and year-to-date monthly costs evolution.

It can be seen how all these sections look like in the following picture.
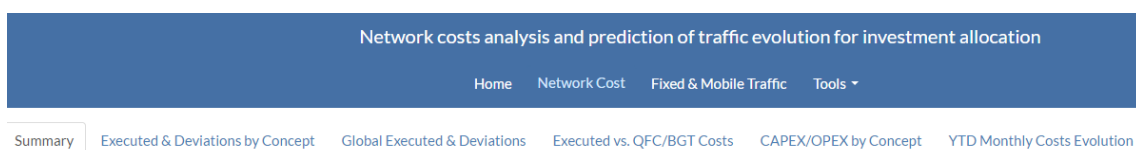
Figure 13: Subsections of network costs.

4.4.1. Accessible and filterable costs data

The first developed subsection of network costs is a table containing all the information without processing. It can be seen all the types of costs, each cost in the local currency (EURO), to which network it belongs and more.

This useful representation of the costs data has been achieved by writing in R code. The user interface script (ui.R) manages the look and feel of the web application, while the server script (server.R) manages the data processing.

Inside the ui.R, which helps to divide the sections of the web application and to add all the selectable inputs to filter the data, this code has been written.

```r
tabPanel("Network Cost",
includeHTML("www/FondoGlobalReport.html"),
    tabsetPanel(
      id = "Network Cost",
      tabPanel("Summary"
          , fluidPage(
            h2("Network Costs"),
            h3(paste(
              month(
                ymd(011201) + months(CURRENTMONTH),
                label = TRUE,
                abbr = FALSE
              ), CURRENTYEAR
            )), #Instruction so that the values in the table are
aligned to the right
                tags$head(tags$style(".table  .alignRight  {text-
align:right;}")),
sidebarLayout(position = "right",
sidebarPanel(
width = SIDEBARPANELWIDTH,
selectInput(
 inputId = "capexopexNetworkCosts",
label = h4("CAPEX/OPEX"),
 choices = c('CAPEX+OPEX', 'CAPEX', 'OPEX')),
```

```r
selectInput(
 inputId = "typeNetworkCosts",
 label = h4("Global/Countries"),
 choices = c('TEF', 'Countries', 'Organization')),
selectInput(
 inputId = "networkNetworkCosts",
 label = h4("Network"),
 choices = c('Fixed+Mobile', 'Fixed', 'Mobile')),
dateInput2(
 "InitialDateNetworkCosts",
 h4("Initial date"),
 value = as.Date(InitialDate)),
dateInput2(
 "FinalDateNetworkCosts",
 h4("Final date"),
 value = as.Date(ActualFinalDate)),
downloadButton('downloadNetworkCosts', 'Download
selection')),
sidebarPanel(
width = MAINPANELWIDTH,
h3(textOutput("myTitleNetworkCosts"), align = "center"),
dataTableOutput("myTableNetworkCosts"))
)
)),
```

Figure 14: R code of the user interface to enable the costs summary section.

Inside the server.R, which allow to treat all the data and filter it depending on the selectors that appear on the web application, this code has been written.

```r
output$myTableNetworkCosts <- renderDataTable({

initial_date <- input$InitialDateNetworkCosts
final_date <- input$FinalDateNetworkCosts

ACTUAL_VALUES_subset <- subset(ACTUAL_VALUES, (YEAR >=
year(initial_date) & MONTH >= month(initial_date)) & (YEAR <=
year(final_date) & MONTH <= month(final_date)))
    ACTUAL_VALUES_subset <- subset(ACTUAL_VALUES_subset, EURO >
0)

if (input$capexopexNetworkCosts == 'CAPEX+OPEX')
{
groupBy <- c('RED', 'COUNTRY','YEAR','MONTH', 'DATETIME', 'CONCEPT')
ACTUAL_VALUES_subset <- ddply(ACTUAL_VALUES_subset, groupBy,
numcolwise(sum))
ACTUAL_VALUES_subset$COST_TYPE <- 'CAPEX+OPEX'
}
```

```r
}
if (input$networkNetworkCosts == 'Fixed+Mobile')
{
groupBy <- c('COST_TYPE',
'COUNTRY','YEAR','MONTH', 'DATETIME',
'CONCEPT')
ACTUAL_VALUES_subset <-
ddply(ACTUAL_VALUES_subset, groupBy,
numcolwise(sum))
ACTUAL_VALUES_subset$RED <- 'Fixed+Mobile'
}
else if (input$networkNetworkCosts == 'Fixed')
{
ACTUAL_VALUES_subset <-
subset(ACTUAL_VALUES_subset, RED == 'Fixed')
} else if (input$networkNetworkCosts ==
'Mobile'){
```

```
else if (input$capexopexNetworkCosts == 'CAPEX')
{
ACTUAL_VALUES_subset <- subset(ACTUAL_VALUES_subset, COST_TYPE
== 'CAPEX')
}
else if (input$capexopexNetworkCosts == 'OPEX')
{
ACTUAL_VALUES_subset <- subset(ACTUAL_VALUES_subset, COST_TYPE
== 'OPEX')
}


if (input$typeNetworkCosts == 'TEF'){
heading <- 'Network Global Costs'
countries <- 'TEF'
ACTUAL_VALUES_subset <-
ACTUAL_VALUES_subset[ACTUAL_VALUES_subset$COUNTRY %in%
countries,]
} else if (input$typeNetworkCosts == 'Countries'){
heading <- 'Network Costs by Country'
countries <-
c("OB1","OB2","OB3","OB4","OB5","OB6","OB7","OB8","OB9","OB10","
OB11","OB12","OB13","OB14","OB15","OB16","OB17")
ACTUAL_VALUES_subset <-
ACTUAL_VALUES_subset[ACTUAL_VALUES_subset$COUNTRY %in%
countries,]
} else if (input$typeNetworkCosts == 'Organization') {
heading <- 'Network Costs by Organization'
countries <- c("OB3","OB4","OB7","OB14","ORG5","ORG6")
ACTUAL_VALUES_subset <-
ACTUAL_VALUES_subset[ACTUAL_VALUES_subset$COUNTRY %in%
countries,]
```

```
ACTUAL_VALUES_subset <-
subset(ACTUAL_VALUES_subset, RED == 'Mobile')
}

ACTUAL_VALUES_subset$DATETIME <- NULL # We
delete this column because we do not need it in
the representation of the table
ACTUAL_VALUES_subset <-
ACTUAL_VALUES_subset %>% select(-EURO,
everything()) # We put the EURO column at the
end

ACTUAL_VALUES_subset$EURO <-
ACTUAL_VALUES_subset$EURO/1000000
colnames(ACTUAL_VALUES_subset)[which(names(
ACTUAL_VALUES_subset) == "EURO")] <- "EURO
(Millions)"

output$downloadNetworkCosts <-
downloadHandler(
filename =
paste('NetworkCosts',input$typeNetworkCosts,in
put$capexopexNetworkCosts,'.csv',sep = "")
,content = function(file) {
  write.csv2(ACTUAL_VALUES_subset,
file,row.names = FALSE)
})

output$myTitleNetworkCosts <-
renderText({heading})
return(ACTUAL_VALUES_subset)
})
```

Figure 15: R code of the server to enable the costs summary section.

It can be seen how all this R code is transformed to a beautiful and useful table with costs filtering thanks to the Shiny package.

Figure 16: Table to access and filter costs information.

It can be seen an example of how to filter the costs data with the following picture. All the CAPEX from all the countries is going to be shown and only their fixed network costs from November to December 2016. Then, it is going to be filtered again (on the lower part of the table) to show only the data from one OB, in this case OB7.
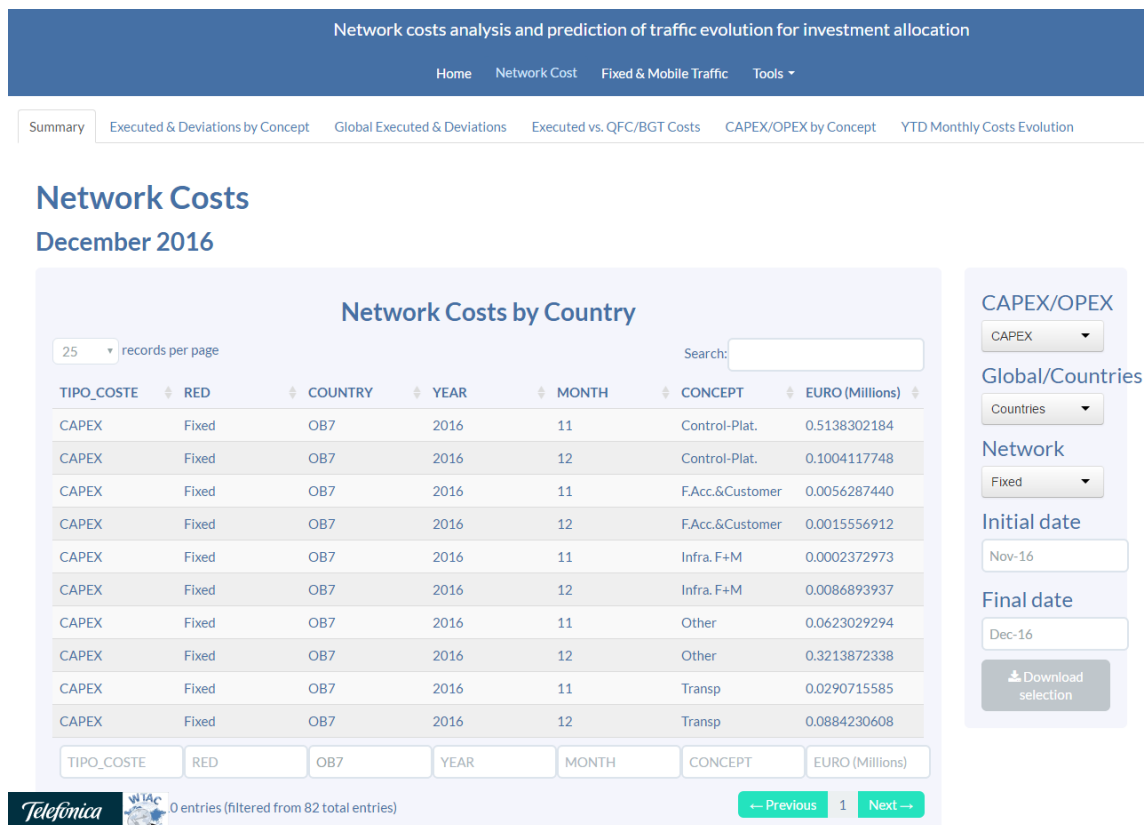
Figure 17: Example of filtering network costs by CAPEX, country and network.

Furthermore, a "Download selection" button has been implemented, with the downloadHandler function in R, in order to let anyone, who has access to this web application, to download all the data in a CSV format file in order to use it with Excel or even with R to process it in their way. It can be seen how this information would appear in this file in the following table.

COST_TYPE	RED	COUNTRY	YEAR	MONTH	CONCEPT	EURO (Millions)
CAPEX	Fixed	OB1	2016	12	F.Acc.&Customer	0,15486775
CAPEX	Fixed	OB1	2016	12	FTTX	0,58460667
CAPEX	Fixed	OB1	2016	12	Control-Plat.	0,61874495
CAPEX	Fixed	OB1	2016	12	F.Acc.&Customer	1,45618205
…	…	…	…	…	…	…

Figure 18: CSV example of download costs function.

4.4.2. Executed costs, deviations and coverages

The second subsection shows the executed costs, deviations and coverages. A table containing the entire information group by concept of cost has been shown.

Besides, it shows the comparison between the actual executed costs, its predicted QFC value and its estimated BGT value.

The deviations from these QFC and BGT predictions have been calculated as the difference of the actual value and its forecast. Then the coverages represent the percentage of that allow or predict cost that the company has spent in a particular month. If this percentage is above 100%, then the expectations have been overcome and because of that, the predicted values.

This representation of the executed costs has been realized with the following code. Inside the ui.R, this code has been written.

```
tabPanel("Executed and QFC/BGT Deviation by Concept"
, fluidPage(
h2("Executed Costs and Deviations by Concept"),
#Instruction so that the values in the table are aligned to the
right
tags$head(tags$style(".table .alignRight {text-align:right;}")),
sidebarLayout(position = "right",
sidebarPanel(
width = SIDEBARPANELWIDTH,
selectInput(
 inputId = "capexopexExecutedCosts",
 label = h4("CAPEX/OPEX"),
 choices = c('CAPEX+OPEX', 'CAPEX', 'OPEX')),
selectInput(
 inputId = "typeExecutedCosts",
 label = h4("Global/Countries"),
 choices = c('TEF', 'Countries', 'Organization')),
selectInput(
 inputId = "networkExecutedCosts",

label = h4("Network"),
 choices = c('Fixed+Mobile', 'Fixed', 'Mobile')),
selectInput(
 inputId = "yearExecutedCosts",
 label = h4("Year"),
 choices = rev(2012:CURRENTYEAR)),
selectInput(
 inputId = "monthExecutedCosts",
 label = h4("Month"),
 choices = rev(1:CURRENTMONTH)),
downloadButton('downloadExecutedCosts', 'Download
selection')),
sidebarPanel(
width = MAINPANELWIDTH,
h3(textOutput("myTitleExecutedCosts"), align = "center"),
dataTableOutput("myTableExecutedCosts"))
)
)),
```

Figure 19: R code of the user interface to enable the executed costs and deviations by concept section.

Inside the server.R the costs data has been gotten and filtered with this code.

```
output$myTableExecutedCosts <- renderDataTable({

ACTUAL_VALUES_subset <- subset(ACTUAL_VALUES,
MONTH >= 1 & MONTH <=
as.numeric(input$monthExecutedCosts))
ACTUAL_VALUES_subset <-
subset(ACTUAL_VALUES_subset, YEAR ==
as.numeric(input$yearExecutedCosts))
QFC_VALUES_subset <- subset(QFC_VALUES, YEAR ==
as.numeric(input$yearExecutedCosts))
QFC_VALUES_subset <- subset(QFC_VALUES_subset,
MONTH >= 1 & MONTH <=
as.numeric(input$monthExecutedCosts))
BGT_VALUES_subset <- subset(BGT_VALUES, YEAR ==
as.numeric(input$yearExecutedCosts))
BGT_VALUES_subset <- subset(BGT_VALUES_subset,
MONTH >= 1 & MONTH <=
as.numeric(input$monthExecutedCosts))
# We call the new column: Accumulated/Executed

ACTUAL_VALUES_PDC_BGT$QFC_Coverage <-
(ACTUAL_VALUES_PDC_BGT$EXECUTED/ACTUAL_VALUES_PDC_B
GT$QFC)*100
ACTUAL_VALUES_PDC_BGT$BGT_Coverage <-
(ACTUAL_VALUES_PDC_BGT$EXECUTED/ACTUAL_VALUES_PDC_B
GT$BGT)*100
# Units in millions
ACTUAL_VALUES_PDC_BGT$EXECUTED <-
ACTUAL_VALUES_PDC_BGT$EXECUTED/1000000
ACTUAL_VALUES_PDC_BGT$QFC <-
ACTUAL_VALUES_PDC_BGT$QFC/1000000
ACTUAL_VALUES_PDC_BGT$BGT <-
ACTUAL_VALUES_PDC_BGT$BGT/1000000
ACTUAL_VALUES_PDC_BGT$QFC_Deviation <-
ACTUAL_VALUES_PDC_BGT$QFC_Deviation/1000000
ACTUAL_VALUES_PDC_BGT$BGT_Deviation <-
ACTUAL_VALUES_PDC_BGT$BGT_Deviation/1000000
# Truncate decimals
```

```
colnames(ACTUAL_VALUES_subset)[which(names(ACTU
AL_VALUES_subset) == "EURO")] <- "EXECUTED"
colnames(QFC_VALUES_subset)[which(names(QFC_VAL
UES_subset) == "EURO")] <- "QFC"
colnames(BGT_VALUES_subset)[which(names(BGT_VAL
UES_subset) == "EURO")] <- "BGT"
# We put the EURO column at the end
ACTUAL_VALUES_subset <- ACTUAL_VALUES_subset
%>% select(-EXECUTED, everything())
QFC_VALUES_subset <- QFC_VALUES_subset %>%
select(-QFC, everything())
BGT_VALUES_subset <- BGT_VALUES_subset %>%
select(-BGT, everything())

// if-elseif parts as shown in code of the previous
section
groupBy <- c('COST_TYPE', 'COUNTRY', 'CONCEPT', 'RED')
ACTUAL_VALUES_subset <-
ddply(ACTUAL_VALUES_subset, groupBy,
numcolwise(sum))
QFC_VALUES_subset <- ddply(QFC_VALUES_subset,
groupBy, numcolwise(sum))
BGT_VALUES_subset <- ddply(BGT_VALUES_subset,
groupBy, numcolwise(sum))

ACTUAL_VALUES_subset$YEAR <- NULL
ACTUAL_VALUES_subset$MONTH <- NULL
QFC_VALUES_subset$YEAR <- NULL
QFC_VALUES_subset$MONTH <- NULL
BGT_VALUES_subset$YEAR <- NULL
BGT_VALUES_subset$MONTH <- NULL

ACTUAL_VALUES_PDC_BGT <-
merge(ACTUAL_VALUES_subset, QFC_VALUES_subset)
ACTUAL_VALUES_PDC_BGT <-
merge(ACTUAL_VALUES_PDC_BGT,
BGT_VALUES_subset)

ACTUAL_VALUES_PDC_BGT$QFC_Deviation <-
(ACTUAL_VALUES_PDC_BGT$EXECUTED -
ACTUAL_VALUES_PDC_BGT$QFC)

ACTUAL_VALUES_PDC_BGT$BGT_Deviation <-
(ACTUAL_VALUES_PDC_BGT$EXECUTED -
ACTUAL_VALUES_PDC_BGT$BGT)
```

```
ACTUAL_VALUES_PDC_BGT$EXECUTED <-
format(round(ACTUAL_VALUES_PDC_BGT$EXECUTED, 2), nsmall =
2)
ACTUAL_VALUES_PDC_BGT$QFC <-
format(round(ACTUAL_VALUES_PDC_BGT$QFC, 2), nsmall = 2)
ACTUAL_VALUES_PDC_BGT$BGT <-
format(round(ACTUAL_VALUES_PDC_BGT$BGT, 2), nsmall = 2)
ACTUAL_VALUES_PDC_BGT$QFC_Deviation <-
format(round(ACTUAL_VALUES_PDC_BGT$QFC_Deviation, 2),
nsmall = 2)
ACTUAL_VALUES_PDC_BGT$BGT_Deviation <-
format(round(ACTUAL_VALUES_PDC_BGT$BGT_Deviation, 2),
nsmall = 2)
ACTUAL_VALUES_PDC_BGT$QFC_Coverage <-
format(round(ACTUAL_VALUES_PDC_BGT$QFC_Coverage, 2),
nsmall = 2)
ACTUAL_VALUES_PDC_BGT$BGT_Coverage <-
format(round(ACTUAL_VALUES_PDC_BGT$BGT_Coverage, 2),
nsmall = 2)
# We add "M" of millions
colnames(ACTUAL_VALUES_PDC_BGT)[which(names(ACTUAL_VAL
UES_PDC_BGT) == "EXECUTED")] <- "Executed (Millions)"
colnames(ACTUAL_VALUES_PDC_BGT)[which(names(ACTUAL_VAL
UES_PDC_BGT) == "QFC")] <- "QFC (Millions)"
colnames(ACTUAL_VALUES_PDC_BGT)[which(names(ACTUAL_VAL
UES_PDC_BGT) == "BGT")] <- "BGT (Millions)"
colnames(ACTUAL_VALUES_PDC_BGT)[which(names(ACTUAL_VAL
UES_PDC_BGT) == "QFC_Deviation")] <- "QFC Deviation (Millions)"
colnames(ACTUAL_VALUES_PDC_BGT)[which(names(ACTUAL_VAL
UES_PDC_BGT) == "BGT_Deviation")] <- "BGT Deviation (Millions)"
# Add percentage symbol
ACTUAL_VALUES_PDC_BGT$QFC_Coverage <-
paste(ACTUAL_VALUES_PDC_BGT$QFC_Coverage, "%")
ACTUAL_VALUES_PDC_BGT$BGT_Coverage <-
paste(ACTUAL_VALUES_PDC_BGT$BGT_Coverage, "%")

output$downloadExecutedCosts <- downloadHandler(
filename =
paste('ExecutedCosts',input$typeExecutedCosts,input$capexopexE
xecutedCosts,'.csv',sep = "")
,content = function(file) {
  write.csv2(ACTUAL_VALUES_PDC_BGT, file,row.names = FALSE)})

output$myTitleExecutedCosts <- renderText({heading})
return(ACTUAL_VALUES_PDC_BGT) })
```

Figure 20: R code of the server to enable the executed costs and deviations by concept section.

The part of the filtering with the condition statements if-else has been omitted due to its similarity with the code of the previous section.

It can be shown in the following picture how the executed costs and its deviations appear on the web application.

Figure 21: Table of executed costs, deviations and coverages by concept.

As in the previous section, it can be seen an example of how to filter the executed and deviations costs data with the following picture. All the executed costs are going to be shown, deviations and coverages from one OB, in which the concept is 3G. It can be seen how the coverages are below 100% in this case, so it could be said that this operator has not reached the expectative or objectives of the 3G costs.
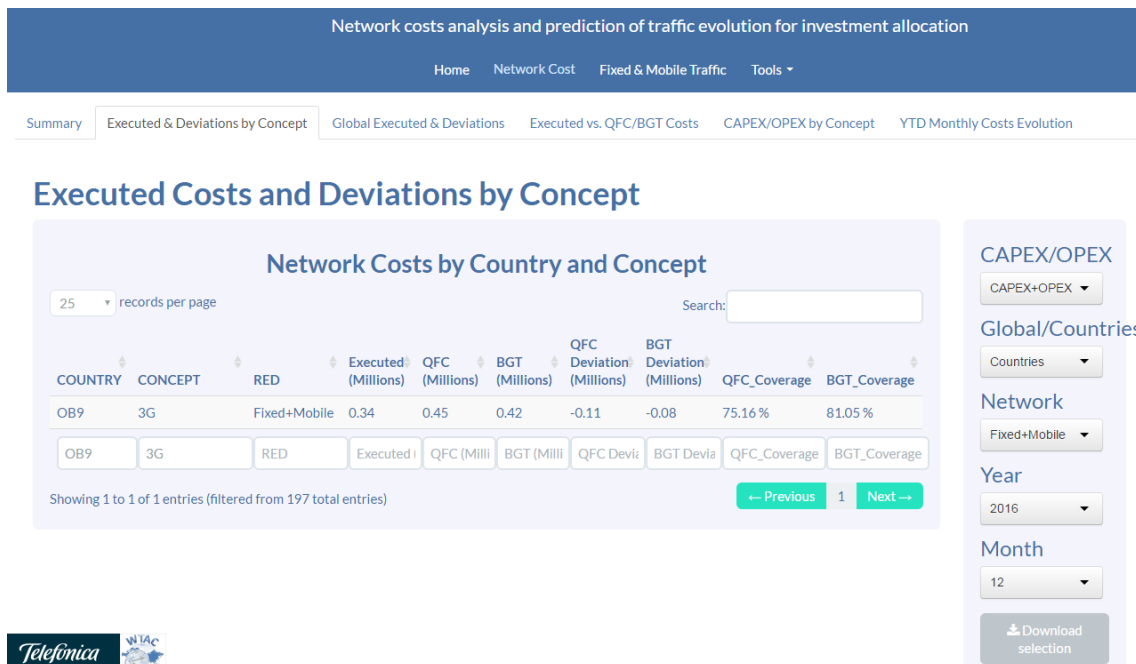
Figure 22: Example of filtering executed costs by concept and country.

Besides, a "Download selection" has been implemented button also in this section to download all the showed information in a CSV format file.

COU NTRY	CONCEPT	RED	Exec uted (Milli ons)	QFC (Milli ons)	BGT (Milli ons)	QFC Devia tion (Milli ons)	BGT Devia tion (Milli ons)	QFC_Co verage	BGT_Co verage
OB2	4G	Fixed+ Mobile	0.92	0.73	0.76	0.19	0.16	125.58%	120.73%
OB2	Control-Plat.	Fixed+ Mobile	4.56	6.60	7.89	-2.04	-3.33	69.15%	57.85 %
OB2	F.Acc.&Cu stomer	Fixed+ Mobile	1.47	12.64	8.51	-1.17	2.96	90.75%	134.75%
…	…	…	…	…	…	…	…	…	…

Figure 23: CSV example of download executed, deviations and coverages costs function.

4.4.3. Year-To-Date (YTD) costs, deviations and coverages

The third subsection shows the year-to-date executed costs, deviations and coverages. The year-to-date is the sum of the costs from the first month of the year (January) to the actual month in study.

This illustration of the year-to-date executed costs data has been achieved by writing in R the following code in the user interface and server in the Shiny package.

Inside the ui.R, this code has been written.

```
tabPanel("Globally Executed and QFC/BGT Deviation"
, fluidPage(
h2("Globally Executed Costs and Deviations"),
#Instruction so that the values in the table are aligned to the
right
tags$head(tags$style(".table .alignRight {text-align:right;}")),
sidebarLayout(position = "right",
sidebarPanel(
width = SIDEBARPANELWIDTH,
selectInput(
 inputId = "capexopexGloballyExecutedCosts",
 label = h4("CAPEX/OPEX"),
 choices = c('CAPEX+OPEX', 'CAPEX', 'OPEX')),
selectInput(
 inputId = "typeGloballyExecutedCosts",
 label = h4("Global/Countries"),
 choices = c('Organization','Countries','TEF')),
selectInput(
 inputId = "networkGloballyExecutedCosts",

label = h4("Network"),
 choices = c('Fixed+Mobile', 'Fixed', 'Mobile')),
selectInput(
 inputId = "yearGloballyExecutedCosts",
 label = h4("Year"),
 choices = rev(2012:CURRENTYEAR)),
selectInput(
 inputId = "monthGloballyExecutedCosts",
 label = h4("Month"),
 choices = rev(1:CURRENTMONTH)),
downloadButton('downloadGloballyExecutedCosts',
'Download selection')),
sidebarPanel(
width = MAINPANELWIDTH,
h3(textOutput("myTitleGloballyExecutedCosts"), align =
"center"),
dataTableOutput("myTableGloballyExecutedCosts"))
)
)),
```

Figure 24: R code of the user interface to enable the Year-to-date executed costs and deviations section.

Inside the server.R, this code has been written for the YTD executed costs.

```
output$myTableGloballyExecutedCosts <-
renderDataTable({

ACTUAL_VALUES_subset <-
subset(ACTUAL_VALUES_singroupBy, MONTH >=
1 & MONTH <=
as.numeric(input$monthGloballyExecutedCosts))
ACTUAL_VALUES_subset <-
subset(ACTUAL_VALUES_subset, YEAR ==
as.numeric(input$yearGloballyExecutedCosts))

QFC_VALUES_subset <-
subset(QFC_VALUES_singroupBy, YEAR ==
as.numeric(input$yearGloballyExecutedCosts))
QFC_VALUES_subset <-
subset(QFC_VALUES_subset, MONTH >= 1 &
MONTH <=
as.numeric(input$monthGloballyExecutedCosts))

BGT_VALUES_subset <-
subset(BGT_VALUES_singroupBy, YEAR ==
as.numeric(input$yearGloballyExecutedCosts))
BGT_VALUES_subset <-
subset(BGT_VALUES_subset, MONTH >= 1 &
MONTH <=
as.numeric(input$monthGloballyExecutedCosts))

# We call the new column:
Accumulated/Executed
colnames(ACTUAL_VALUES_subset)[which(name
s(ACTUAL_VALUES_subset) == "EURO")] <-
"EXECUTED"
colnames(QFC_VALUES_subset)[which(names(QF
C_VALUES_subset) == "EURO")] <- "QFC"
colnames(BGT_VALUES_subset)[which(names(B
GT_VALUES_subset) == "EURO")] <- "BGT"

ACTUAL_VALUES_subset <-
ACTUAL_VALUES_subset %>% select(-EXECUTED,
everything()) # We put the EURO column at the
end
QFC_VALUES_subset <- QFC_VALUES_subset
%>% select(-QFC, everything())
BGT_VALUES_subset <- BGT_VALUES_subset
%>% select(-BGT, everything())
```

```
// if-elseif parts as shown in code of the previous section

groupBy <- c('COST_TYPE', 'COUNTRY', 'RED')
ACTUAL_VALUES_subset <- ddply(ACTUAL_VALUES_subset, groupBy,
numcolwise(sum))
QFC_VALUES_subset <- ddply(QFC_VALUES_subset, groupBy,
numcolwise(sum))
BGT_VALUES_subset <- ddply(BGT_VALUES_subset, groupBy,
numcolwise(sum))

ACTUAL_VALUES_PDC_BGT <- merge(ACTUAL_VALUES_subset,
QFC_VALUES_subset)
ACTUAL_VALUES_PDC_BGT <- merge(ACTUAL_VALUES_PDC_BGT,
BGT_VALUES_subset)

ACTUAL_VALUES_PDC_BGT$QFC_Deviation <-
(ACTUAL_VALUES_PDC_BGT$EXECUTED -
ACTUAL_VALUES_PDC_BGT$QFC)
ACTUAL_VALUES_PDC_BGT$BGT_Deviation <-
(ACTUAL_VALUES_PDC_BGT$EXECUTED -
ACTUAL_VALUES_PDC_BGT$BGT)
ACTUAL_VALUES_PDC_BGT$QFC_Coverage <-
(ACTUAL_VALUES_PDC_BGT$EXECUTED/ACTUAL_VALUES_PDC_BGT$QFC
)*100
ACTUAL_VALUES_PDC_BGT$BGT_Coverage <-
(ACTUAL_VALUES_PDC_BGT$EXECUTED/ACTUAL_VALUES_PDC_BGT$BGT
)*100

// all the following processing is done as the previous code of server.R for
the other section

# Units in millions
# Truncate decimals
# We add "M" of millions
# Add percentage symbol

output$downloadGloballyExecutedCosts <- downloadHandler(
filename =
paste('GloballyExecutedCosts',input$typeGloballyExecutedCosts,input$ca
pexopexGloballyExecutedCosts,'.csv',sep = "")
,content = function(file) {
  write.csv2(ACTUAL_VALUES_PDC_BGT, file,row.names = FALSE)})

output$myTitleGloballyExecutedCosts <- renderText({heading})
return(ACTUAL_VALUES_PDC_BGT)
})
```

Figure 25: R code of the server to enable the Year-to-date executed costs and deviations section.

The part of the filtering with the condition statements if-else has been omitted and the part of putting the values in millions units, adding a M character and the percentage symbol % to the values of the data frame in order to show them in the web application too, due to its similarity with the code of the previous section.

It can be seen the year-to-date executed costs; this means the sum of the costs from January to the actual month, the year-to-date deviations and its coverages as follows in the picture.
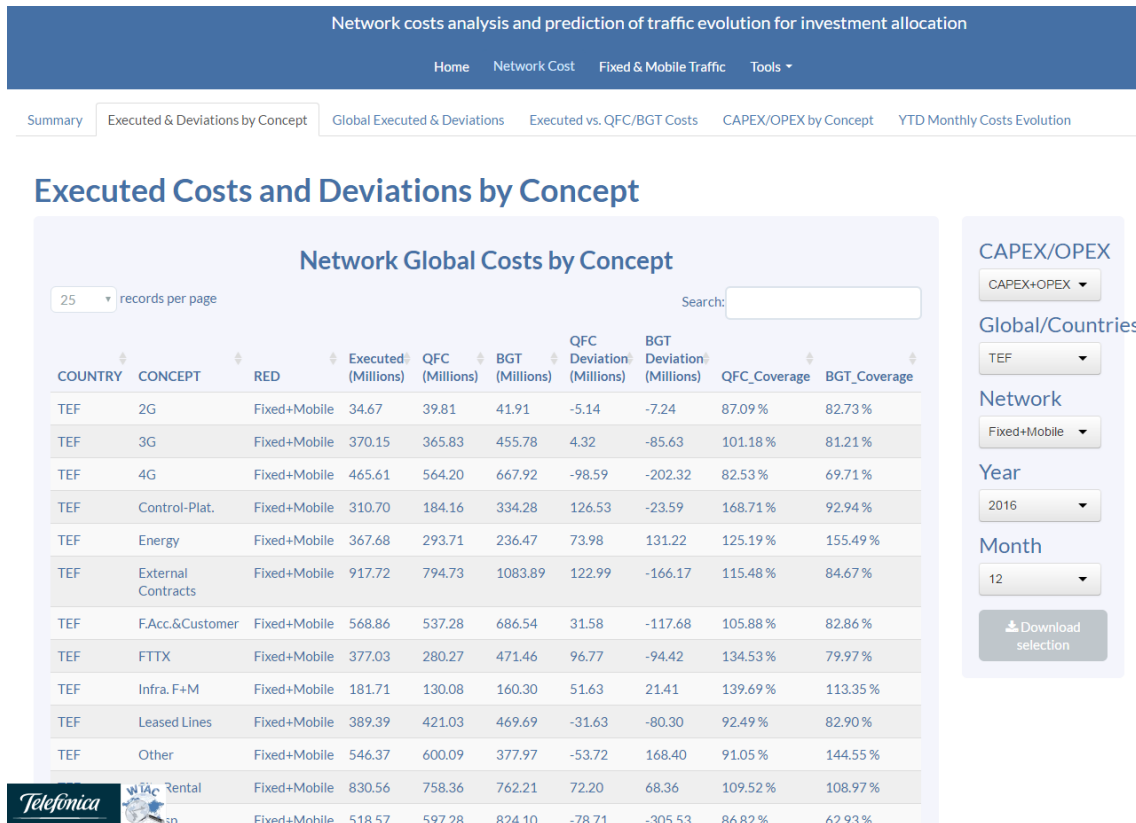
Figure 26: Table of Year-to-date executed costs, deviations and coverages.

As in the previous sections, it can be seen an example of how to filter the year-to-date executed costs, for example, showing the information of CAPEX from all operating businesses and only about the fixed network from January to December 2016.
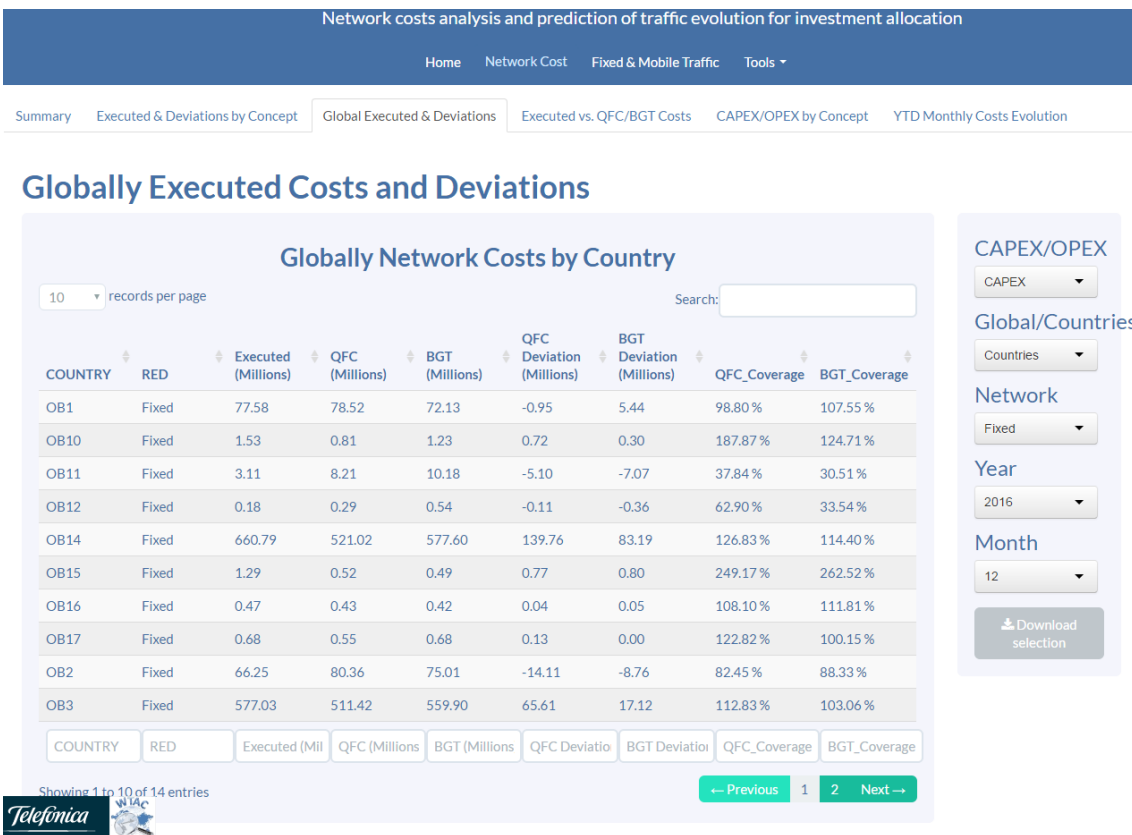
Figure 27: Example of filtering Year-to-date executed costs by country and network.

As well, a "Download selection" button has been put to download all the showed data in a CSV format file.

COUNTRY	RED	Executed (Millions)	QFC (Millions)	BGT (Millions)	QFC Deviation (Millions)	BGT Deviation (Millions)	QFC_Coverage	BGT_Coverage
OB1	Fixed	77.58	78.52	72.13	-0.95	5.44	98.80 %	107.55%
OB2	Fixed	66.25	80.36	75.01	-14.11	-8.76	82.45 %	88.33%
…	…	…	…	…	…	…	…	…

Figure 28: CSV example of download YTD executed, deviations and coverages costs function.

4.4.4. Comparison between executed and QFC/BGT costs

The forth subsection shows a graph of the executed costs in a concrete month and its comparison with the QFC and BGT predictions, which is very helpful in order to see easily if the countries are spending more or less money than expected.

On the one hand, the ui.R code part is very similar to the previous sections, just with a small change, because in this case a graph is going to be shown, not a table, so the last lines of code have to be changed.

```
sidebarPanel(
  width = MAINPANELWIDTH,
  h3(textOutput("myTitleGraphGloballyExecutedCosts"), align = "center"),
  showOutput("myChartGraphGloballyExecutedCosts", "nvd3")
)
```

Figure 29: R code of the user interface to enable the graphical executed vs. QFC/BGT costs section.

On the other hand, the server.R code part is very similar to the previous sections too, just with small changes to show a graph, the last lines of code are going to be changed.

```
p <- pintaKPITimeEvolByCountry(ACTUAL_VALUES_PDC_BGT, columnX = 'COUNTRY', columnY = 'VALUE', grupo = 'TYPE',
graphType = 'multiBarChart', units = 'Millions of €', chartName = 'myChartGraphGloballyExecutedCosts',  melt = FALSE,
datetimeFunction = FALSE)
p$chart(reduceXTicks = FALSE)
p$yAxis(tickFormat = "#! function(d) {return d3.format(',.3f')(d)} !#")
```

Figure 30: R code of the server to enable the graphical executed vs. QFC/BGT costs section.

Being pintaKPITimeEvolByCountry a function created by ourselves, which is defined in functions.R, which is called from the server.R code, in order to manage all the possible graphs in this project: line chart, bar chart and more, by using nPlot function of R.

It can be seen this graphical representation of the costs versus its predictions.
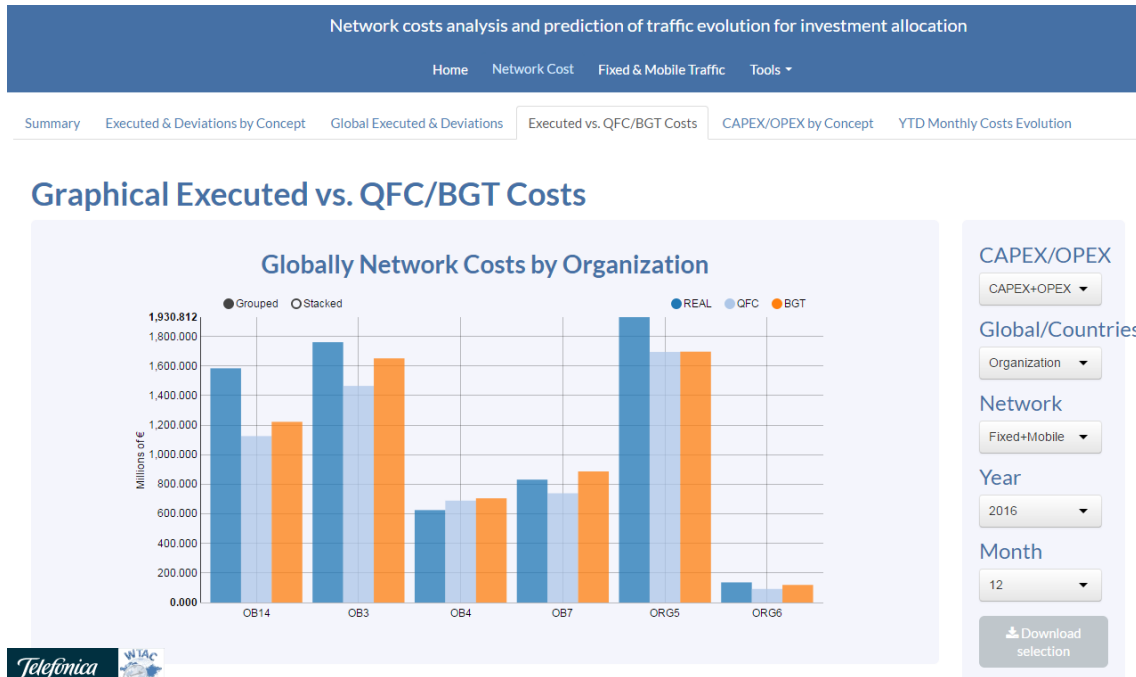
Figure 31: Graph of executed vs. QFC and BGT costs.

In this picture, the sum of CAPEX and OPEX and its predictions can be seen for the organizations that compose Telefónica. Although they are unreal data, it could be explained that most of them are spending more than the target thanks to this chart.

An example of filtering this data is, for example, taking only the actual values and QFC costs objectives for all the countries in order to know how much each operating business is supposed to spend on December 2016.
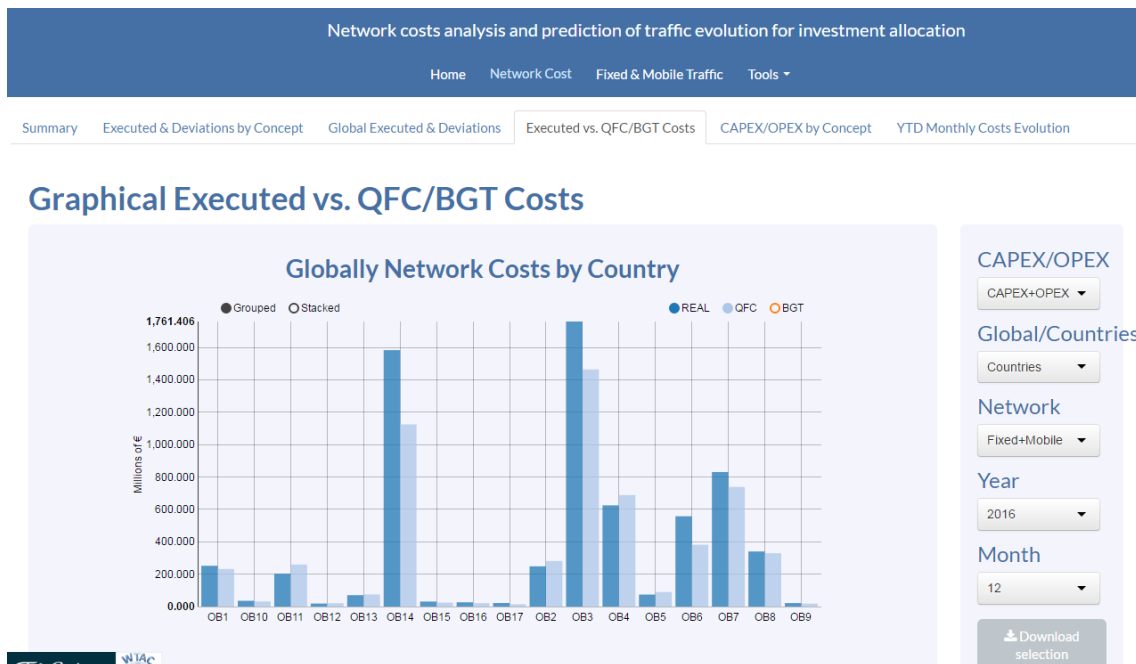
Figure 32: Example of filtering executed costs vs. QFC/BGT in a graph.

The download functionality has been also put to extract the information that appears in the graph, so anybody can use it for his or her own purposes.

COUNTRY	TYPE	VALUE
OB1	ACTUAL	251,621758
OB1	QFC	231,67535
OB1	BGT	248,451897
…	…	…

Figure 33: CSV example of download actual, QFC and BGT costs function.

4.4.5. Comparison of CAPEX and OPEX by concept

The fifth subsection shows a graph of the CAPEX and OPEX divided by concept, which gives a better view on where all the expenses are going.

On the one hand, the ui.R code part is very similar to the previous section that have a chart too, just with changes in the names of variables as can be seen.

```
sidebarPanel(
  width = MAINPANELWIDTH,
  h3(textOutput("myTitleGraphCAPEXOPEXbyConcept"), align = "center"),
  showOutput("myChartGraphCAPEXOPEXbyConcept", "nvd3")
)
```

Figure 34: R code of the user interface to enable the graphical comparison of CAPEX/OPEX by concept.

On the other hand, with the server.R occurs the same, it is very similar to the previous section with a chart, so it just changes the subdivisions of the data acquired and the presentation of this data by concept.

```
p <- pintaKPITimeEvolByCountry(ACTUAL_VALUES_subset, columnX = 'COUNTRY', columnY = 'VALUE', grupo = 'CONCEPT',
graphType = 'multiBarChart', units = 'Millions of €', chartName = 'myChartGraphCAPEXOPEXbyConcept', melt = FALSE,
datetimeFunction = FALSE)
p$chart(reduceXTicks = FALSE)
p$yAxis(tickFormat = "#! function(d) {return d3.format(',.3f')(d)} !#")
```

Figure 35: R code of the server to enable the graphical comparison of CAPEX/OPEX by concept section.

The first view of this subsection is the CAPEX and OPEX divided by concept for each organization of the telecommunications company for a concrete month and year.
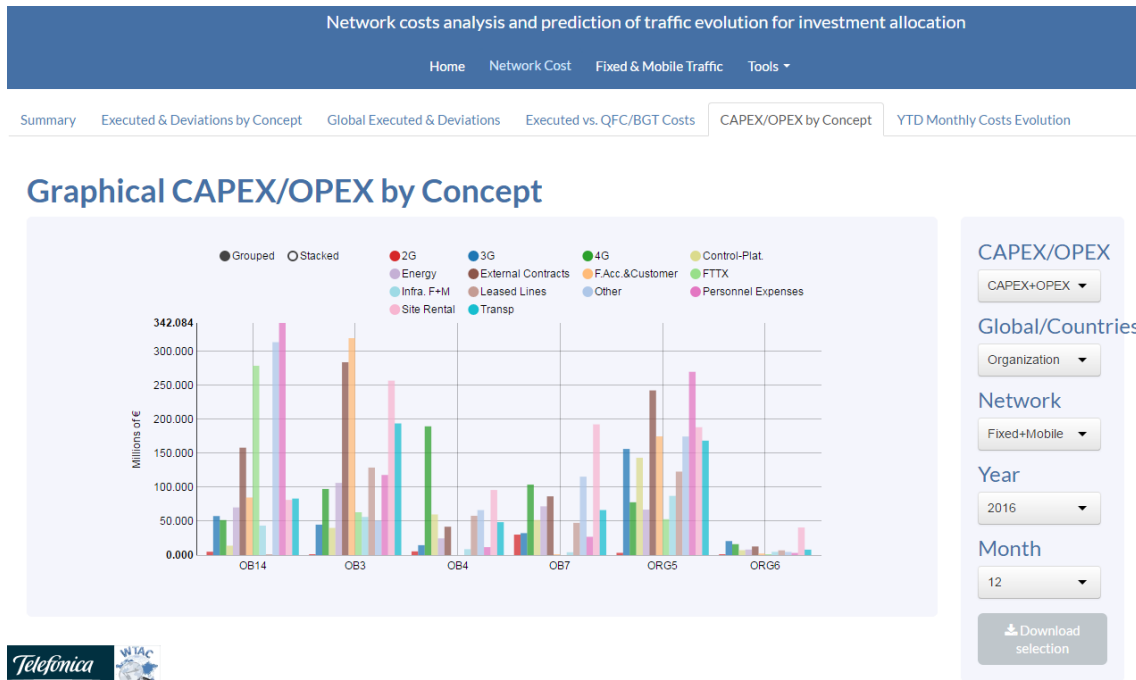
Figure 36: Graph of CAPEX/OPEX by concept.

There is an option in the chart to view the bars stacked in order to have the sum for each organization. As an example, the concepts for just 2G, 3G and 4G comparison by countries have been stacked as can be seen in the next picture.
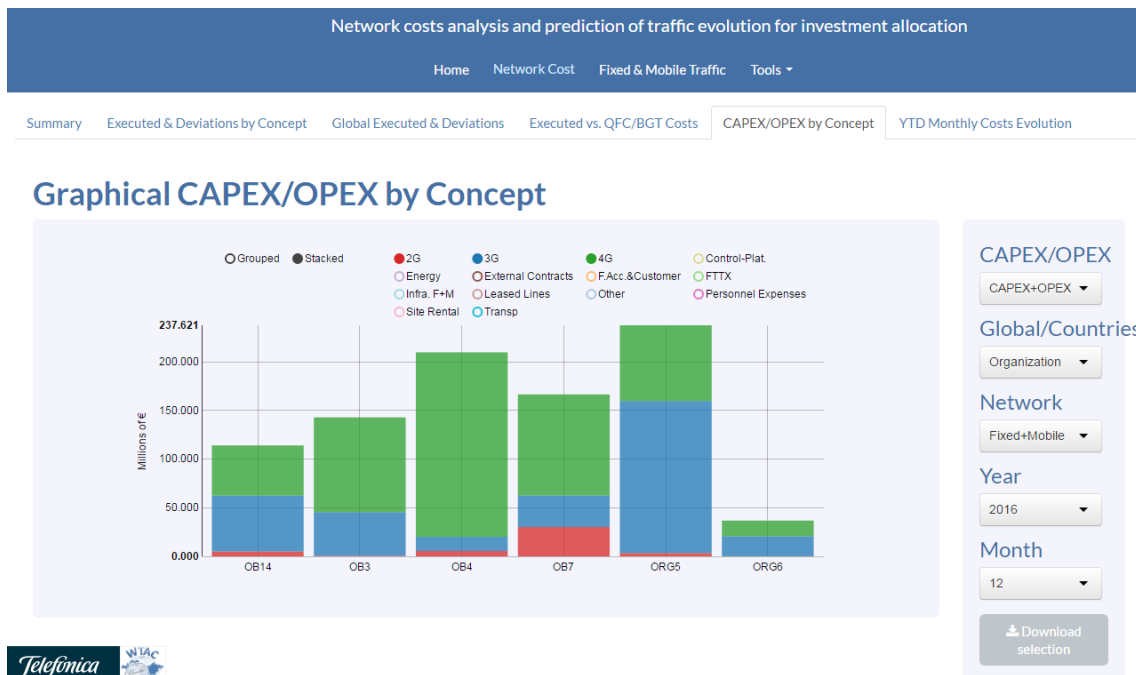
Figure 37: Example of stacking 2G, 3G and 4G costs by country.

In this graph, it could be concluded that the operating business OB4 is spending a lot of money in the 4G technology, whereas the OB7 is still spending a considerable percentage of expenses in 2G, although this data is not the true information of costs of Telefónica.

Another example could be to filter just organizations and fixed network, in order to see only the CAPEX and OPEX related to fixed networks.
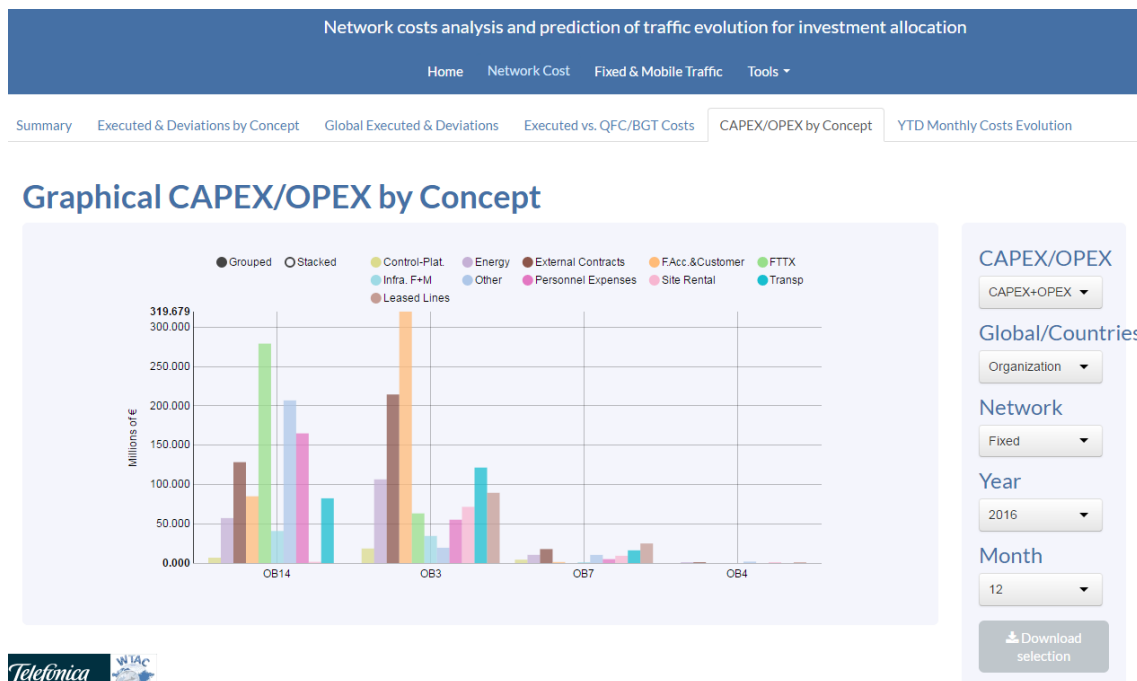
Figure 38: Example of filtering CAPEX/OPEX of fixed networks by organization.

It could be deducted from this chart that in organization OB14 the most important part of the expenses would be spent on FTTX while in organization OB7 would be the leased lines.

In addition, the download functionality has been put to this graph too.

4.4.6. Comparison of YTD monthly evolution vs. QFC/BGT

The last subsection of network costs shows a monthly evolution of the CAPEX and OPEX of the whole company with year-to-date, so it is aggregating each CAPEX and OPEX value of each month and the previous ones.

On the one hand, the ui.R code part is very similar to the previous sections that have a chart too, just with some changes due to the fact that only two selectors are going to be employed to filter the data showed in the graph.

```
tabPanel("YTD Monthly Costs Evolution"
, fluidPage(
h2("Graphical CAPEX/OPEX Monthly Evolution"),
#Instruction so that the values in the table are aligned to the right
tags$head(tags$style(".table .alignRight {text-align:right;}")),
sidebarLayout(position = "right",
 sidebarPanel(
  width = SIDEBARPANELWIDTH,
  selectInput(
          inputId = "capexopexGraphMonthEvolution",
          label = h4("CAPEX/OPEX"),
          choices = c('CAPEX+OPEX', 'CAPEX', 'OPEX')   ),
  selectInput(
          inputId = "yearGraphMonthEvolution",
          label = h4("Year"),
          choices = rev(2012:CURRENTYEAR)   ),
  downloadButton('downloadGraphMonthEvolution', 'Download selection') ),
 sidebarPanel(
  width = MAINPANELWIDTH,
  h3(textOutput("myTitleGraphMonthEvolution"), align = "center"),
  showOutput("myChartGraphMonthEvolution", "nvd3") )
)
))
```

Figure 39: R code of the user interface to enable the graphical comparison of CAPEX/OPEX YTD evolution.

On the other hand, with the server.R occurs the same, it is very similar to the previous section with a chart, although in this case just the 'TEF' values are taken, which are the global values of CAPEX and OPEX for the telecommunications company.

There is only an if-else statement in the code of this subsection for the selection of CAPEX+OPEX, only CAPEX or only OPEX.

```
p <- pintaKPITimeEvolByCountry(ACTUAL_VALUES_PDC_BGT, columnX = 'MONTH', columnY = "VALUE", grupo = "TYPE",
graphType = 'lineChart', units = 'Millions of €', chartName = 'myChartGraphMonthEvolution', melt = FALSE, datetimeFunction =
FALSE)
p$yAxis(tickFormat = "#! function(d) {return d3.format(',.3f')(d)} !#")
```

Figure 40: R code of the server to enable the graphical comparison of CAPEX/OPEX YTD evolution section.

The first view of this last subsection is the sum of CAPEX and OPEX (CAPEX+OPEX) for each month but it is shown the value of the year-to-date, so for March, the CAPEX and OPEX values of January, February and March itself are being aggregated, so obviously it is going to be a crescent line.

In this case, which is not with true values, it can be seen that the sum of CAPEX and OPEX expenses are going above all predictions (QFC and BGT) since the first month of the year 2016, which is something that should not occur.
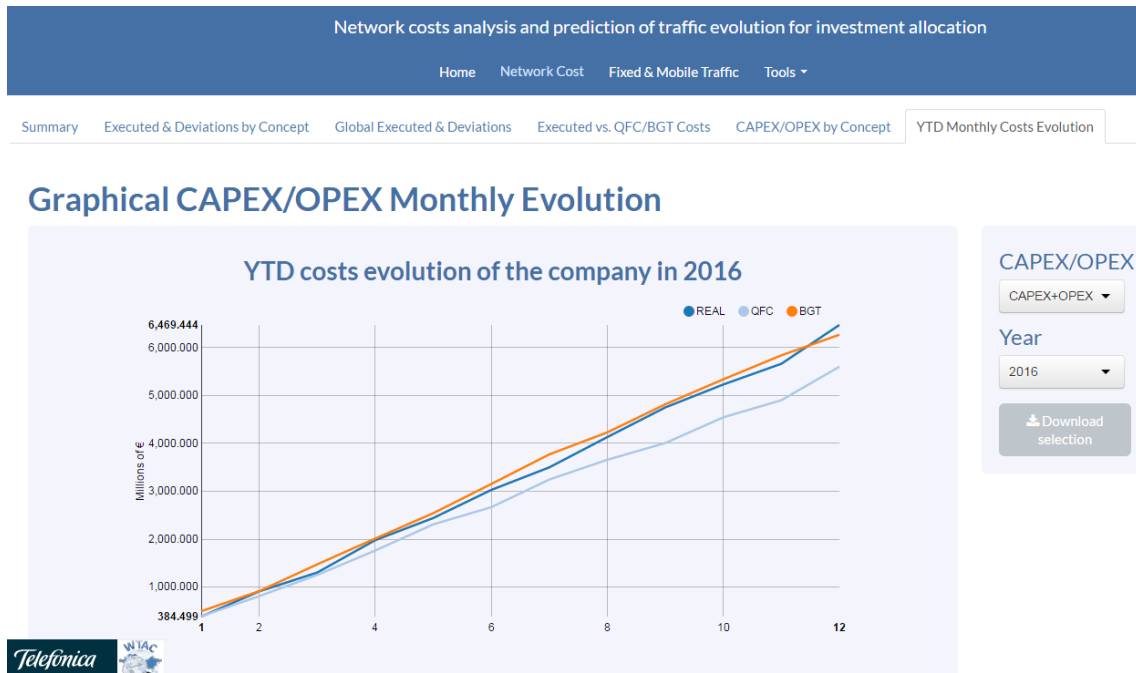
Figure 41: Graph of CAPEX+OPEX monthly evolution.

One example could be showing the year-to-date CAPEX of Telefónica in 2016.
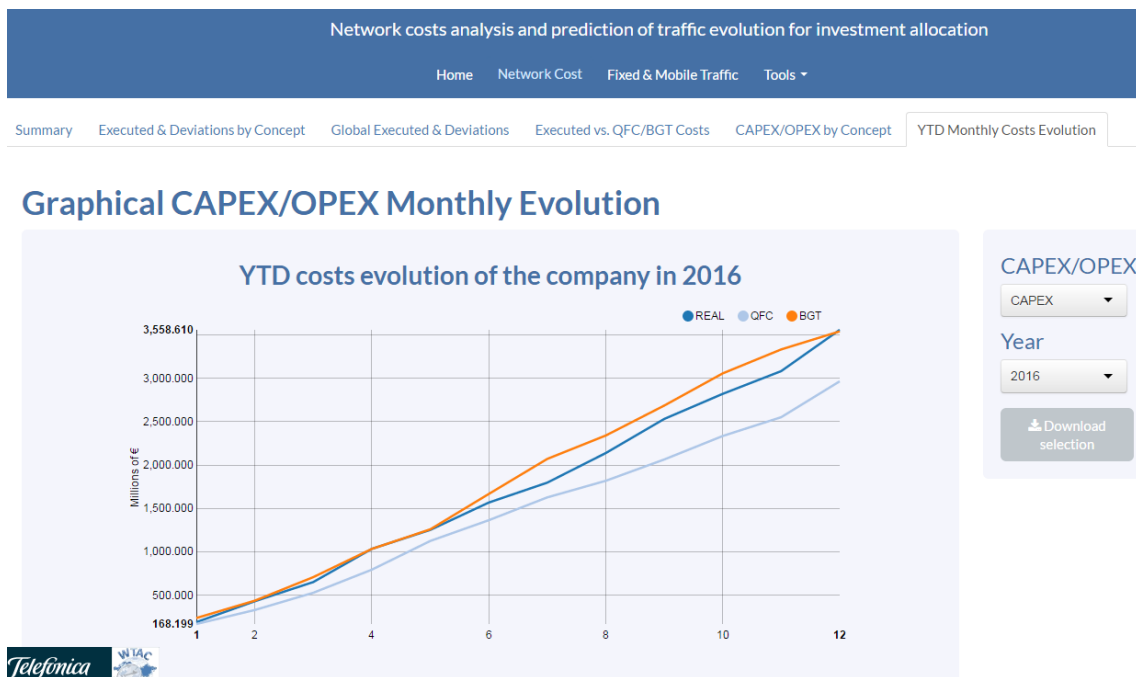
Figure 42: Example of CAPEX monthly evolution.

It can be seen that in this particular case, the CAPEX had been below the BGT estimated value, which could be good because it is not a huge difference, so it seems that the previous year the department in charge of estimating the budget (BGT) had been very precise and forecasted very well the CAPEX of the company for the following year. Nevertheless, the next revision of these values, which is the QFC, it

had been not as realistic as it should have been or something happened with the costs in the last months of 2016 that they had to spend more CAPEX.

Another example could be showing the year-to-date OPEX of Telefónica.
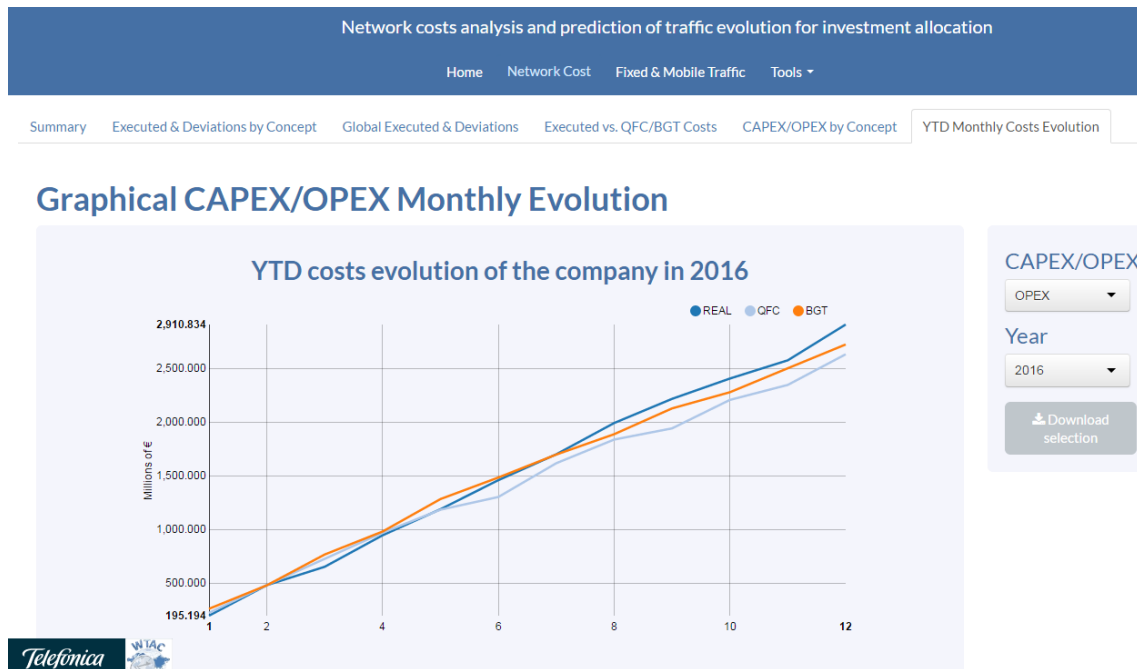
Figure 43: Example of OPEX monthly evolution.

In this case, it can be seen how the actual OPEX of the company is almost the same as the expected for the QFC and the BGT forecasts. This should happen, if this was a real-life case, the OPEX had been presumably well forecasted.

Furthermore, the download functionality has been put to this chart too.

4.5.   Costs analysis

Thanks to all the charts and tables of the web application that have been shown, an analysis of how all the expenses are being spent in a company can be done and it can be known if the estimations of QFC and budget BGT had been well forecasted or if there is a high deviation from these values. Besides, it can be seen if the company is achieving the coverage of these predictions each month of the year in study.

4.5.1. QFC and BGT deviations and coverages

The deviations and coverages of the actual expenses versus the QFC and BGT predictions can be analyzed with the table that gives access and the possibility to filter all the executed costs, deviations and coverages information of each concept of the expenses and, for example, filter this information by CAPEX, country or network.

In addition, it can be seen all this information in a table with the relation of the year-to-date executed costs, deviations and coverages and filter it too. With this information, it can be seen how is being the monthly evolution.

Besides, all this information in all the tables and charts can be downloaded in a CSV file in order to treat it in R or Excel or another programming language.

4.5.2. Examination of investment allocation

There is a graphical tool that shows executed costs versus its expected value from the QFC and the BGT. With this graph, it can be visually compared if each costs are being what had been expected or not.

As well, there is another graph tool with the comparison of CAPEX/OPEX by concept. For example, it can be filtered by concepts of 2G, 3G and 4G and see how much the company is spending in each mobile technology.

In addition, the investment of the company can be examined with the graphical tool of the year-to-date monthly evolution by CAPEX, OPEX or by the sum of all the expenses and versus the year-to-date QFC and BGT forecasts.

All these values with these graphs can be compared with the QFC and BGT predicted values and see, for example, if the company is allocating well its CAPEX and is being what anticipated or not.

Chapter 5

Carried traffic on fixed and mobile networks

5.1.   Obtaining traffic measurements

Some traffic measurements information is going to be obtained from MariaDB databases. However, in order to have the values for the actual traffic in Busy Hour for fixed and mobile networks and the forecast of total traffic in Busy Hour of downlink and uplink for the last month of the year and the following years, some data processing with R is going to be made to calculate and estimate these values.

Due to confidentiality, the traffic information has been anonymized for the demonstrating part of this section.

Nevertheless, it has been previously verified that the web application displayed every traffic information properly with the true values of Telefónica's traffic measurements.

5.1.1. Input data

There is one table in the database "Traffic_Data", which contains all the available information of networks' traffic measurements that are going to be needed to obtain the expected results.

In order to get all the fields that are requested, a call to the database is going to be made inside the script.R, which will make the calculations and the server.R script of the Shiny web application, which will show the information on charts and tables, just with the following code.

```
odbcDataSources("all")
WTACv2 <- odbcConnect(dsn = "WTAC", uid = "user", pwd = "pass")

queryOficial <- paste("select * from Traffic_Data where year>=2010")

# We perform the SQL query
oficial <- as.data.frame(sqlQuery(WTACv2,queryOficial))

odbcClose(WTACv2)
```

Figure 44: R code to retrieve traffic measurements data from the database.

Straightaway, all the data is accessible in an R data frame and it is ready to be manipulated.

5.1.2. Fixed and mobile network parameters

In order to make the estimations of the carried traffic for the following three years some assumptions are going to be made about needed inputs parameters for the forecasts.

Based on historical data, the FTTH consumption downlink factor is considered to be x2 (the growth of the downlink monthly traffic from the previous year to the current year) while in the uplink, this factor is considered to be x4.

Another parameter is the percentage growth in Gbps for the following years, based on previous years it is assumed a growth of 20% in downlink stream while in the uplink is assumed a growth of 10% over to the previous year.

There are more parameters, for example the traffic per small screen 3G client growth for each estimated year.

5.1.3. Forecast for the Strategic Plan (SP)

For network costs, the quarterly forecast closure (QFC) and the budget (BGT) have been used, which had been target values into which the actual costs had to be compared. For the carried traffic evolution, a forecast for the following 3 years is

going to be developed in order to stablish an objective of traffic for each following year, which is called the Strategic Plan (SP).

Figure 45: Strategic Plan (SP) in a timeline.

5.2. Understanding traffic KPIs definitions

A document has been defined in order to make all the concepts relative to carried traffic of networks understandable to everybody who uses this web application.

That document explains some definitions such as Busy Hour, fixed broadband clients, fixed data traffic in Busy Hour of downlink/uplink, mobile data traffic intensity in Busy Hour, mobile data traffic volume and outdoor population coverage between others.

In this document have been included some definitions about costs (CAPEX and OPEX) too in order to have all the information in one only document. It can be downloaded from the "Tools" tab in the web application as it was explained in the networks costs section.

5.3. Acquiring actual fixed downlink & uplink traffic in Busy Hour (BH)

The xDSL clients and the FTTH clients of all countries where the company is present are going to be needed in order to make all the calculations. This information is found in the database.

In order to know how many clients are using the xDSL technology, it is going to be calculated as follows.

$$xDSL_{clients} = FBB_{clients} - FTTH_{clients}$$

This value has to be known due to the fact that it is wanted to obtain traffic in Busy Hour by xDSL and FTTH technologies. This value can be achieved in R with the use of "ddply", "rbind" and "numcolwise" as can be seen in the following piece of code.

```
xDSL_clients <- ddply(rbind(FTTH_clients,FBB_clients), groupBy, numcolwise(diff))
```

Figure 46: R code to obtain xDSL clients.

Besides, each country has the measurements of traffic intensity for the downlink and for the uplink. These measurements can be found in the database too.

The traffic per xDSL in Busy Hour (BH) for the downlink and the uplink is going to be determined as follows.

$$Traffic\ per\ xDSL\ _{BusyHour\ DL/UL} = \frac{Traffic\ intensity\ _{DL/UL}}{\left(FBB_{clients} + \left(FTTH_{consumption\ factor_{DL/UL}} - 1\right) * FTTH_{clients}\right)}$$

These calculations can be done with R using the following code for the downlink:

```
# Downlink:
groupBy <- c('country', 'year','month','quarter', 'datetime')

FTTH_clients_factor_DL <- FTTH_clients

FTTH_clients_factor_DL$value <- FTTH_clients_factor_DL$value*(FTTH_consumption_DL_factor - 1)

FBB_FTTH_clients_DL <- ddply(rbind(FBB_clients,FTTH_clients_factor_DL), groupBy, numcolwise(sum))

FD8_clients_division_DL <- rbind.match.columns.numdenom(Traffic_intensity_FD8_DL_real, FBB_FTTH_clients_DL)

Traffic_per_xDSL_BH_DL <- ddply(FD8_clients_division_DL, groupBy, plyr::summarize, value = (value[role == "numerator"] /
value[role == "denominator"])*1000)
```

Figure 47: R code to obtain traffic per xDSL in Busy Hour.

In the uplink the same code as above is employed but with the traffic intensity for the uplink and its FTTH consumption factor.

The traffic per FTTH in Busy Hour (BH) for the downlink and the uplink is going to be determined as follows.

$$Traffic\ per\ FTTH\ _{BusyHour\ DL/UL} = Traffic\ per\ xDSL\ _{BusyHour\ DL/UL} * FTTH_{consumption\ factor_{DL/UL}}$$

These calculations can be done with R for the downlink as can be seen in the following piece of code.

```
# Downlink:
Traffic_per_FTTH_BH_DL <- Traffic_per_xDSL_BH_DL
Traffic_per_FTTH_BH_DL$value <- Traffic_per_FTTH_BH_DL$value*FTTH_consumption_DL_factor
```

Figure 48: R code to obtain traffic per FTTH in Busy Hour.

Now, the total monthly traffic for the downlink and the uplink is going to be obtained with a simple equation.

$$Monthly\ xDSL/FTTH_{BusyHour\ DL/UL} = Traffic\ per\ xDSL\ _{BusyHour\ DL/UL} * xDSL/FTTH_{clients}$$

This key performance indicator can be achieved with the following code. Besides, it is divided by a factor in order to have this indicator in Gbps.

```
# Downlink:
Monthly_xDSL_BH_DL <- ddply(rbind(Traffic_per_xDSL_BH_DL,xDSL_clients), groupBy, numcolwise(prod))
Monthly_FTTH_BH_DL <- ddply(rbind.match.columns(Traffic_per_FTTH_BH_DL,FTTH_clients), groupBy, numcolwise(prod))

Monthly_xDSL_BH_DL$value <- Monthly_xDSL_BH_DL$value/1000
Monthly_FTTH_BH_DL$value <- Monthly_FTTH_BH_DL$value/1000
```

Figure 49: R code to obtain monthly xDSL/FTTH traffic in Busy Hour.

Then, the sum of the xDSL and FTTH technology monthly traffic is going to be obtained for the downlink and uplink.

$$Total\ Traffic\ (Gbps)_{BusyHour\ DL/UL} = Monthly\ xDSL_{BusyHour\ DL/UL} + Monthly\ FTTH_{BusyHour\ DL/UL}$$

These sums have been achieved with the following instructions.

```
# Downlink:
Total_traffic_Gbps_BH_DL <- ddply(rbind(Monthly_xDSL_BH_DL,Monthly_FTTH_BH_DL), groupBy, numcolwise(sum))
Total_traffic_Gbps_BH_DL$type <- 'DL'
```

Figure 50: R code to obtain total traffic in Busy Hour (Gbps).

Finally, the authentic monthly fixed downlink and uplink traffic in Busy Hour is obtained by aggregating the total traffic in Busy Hour for the downlink and the total traffic in Busy Hour for the uplink in a month.

$$Busy\ Hour\ Actual\ Fixed\ Traffic = Total\ Traffic\ (Gbps)_{BH\ DL} + Total\ Traffic\ (Gbps)_{BH\ UL}$$

It is as simple as this line of code to integrate the final monthly fixed traffic.

```
groupBy <- c('country', 'year','month','quarter', 'datetime')
Total_traffic_Gbps_BH_DL_UL <- ddply(rbind(Total_traffic_Gbps_BH_DL,Total_traffic_Gbps_BH_UL), groupBy, numcolwise(sum))
Total_traffic_Gbps_BH_DL_UL$type <- 'DL+UL'
```

Figure 51: R code to obtain the actual Busy Hour fixed traffic.

5.4.   Fixed traffic forecast

In order to make the fixed traffic prediction there is a need of knowing the xDSL clients and the FTTH clients in order to identify in the following years what clients the company is supposed to have. These forecast or target clients are taken from the QFC.

Again, as with the historical data, xDSL clients have been calculated as follows.

$$xDSL_{Forecast\ clients} = FBB_{Forecast\ clients} - FTTH_{Forecast\ clients}$$

This value in R can be achieved with the use of the following piece of code.

```
XDSL_clients_PdC_FC <- ddply(rbind(FTTH_clients_PdC_FC,FBB_clients_PdC_FC), groupBy, numcolwise(diff))
```

Figure 52: R code to estimate xDSL clients.

The traffic for the last month of the current year, the value for the next year and the traffic within two and three years are going to be forecasted.

All predictions are made based on the value of December of the previous year, as well for the current year as for the following years.

The forecast value for the last month of the current year or QFC has been obtained as follows.

$$QFC\ Traffic\ per\ xDSL_{BusyHour_{DL/UL}} =$$

$$= Last\ Year\ Traffic\ per\ xDSL_{BusyHour_{DL/UL}} * (1 + \%Forecast\ Growth_{DL/UL})$$

As there are historical values for the last year's traffic on December, this forecast value for the last month of the current year can be obtained with the following code.

```
Traffic_per_xDSL_BH_DL_PdC <- subset(Traffic_per_xDSL_BH_DL, month == 12 & year == PdC - 1)

Traffic_per_xDSL_BH_DL_PdC$value <- Traffic_per_xDSL_BH_DL_PdC$value*(1 + Percentage_growth_DL_Gbps_PdC_FC)

Traffic_per_xDSL_BH_DL_PdC$year <- PdC
```

Figure 53: R code to estimate the traffic per xDSL for the QFC.

Then, as the value for the last month of the current year has been obtained, the value of the last month of the following year can be estimated using this value.

$$FC + 1\ Traffic\ per\ xDSL_{BusyHour_{DL/UL}} =$$

$$= Current\ Year\ Traffic\ per\ xDSL_{BusyHour_{DL/UL}} * (1 + \%Forecast\ Growth_{DL/UL})$$

This would be the code in order to obtain that forecast value.

```
Traffic_per_xDSL_BH_DL_FC_1 <- Traffic_per_xDSL_BH_DL_PdC

Traffic_per_xDSL_BH_DL_FC_1$value <- Traffic_per_xDSL_BH_DL_FC_1$value*(1 + Percentage_growth_DL_Gbps_PdC_FC)

Traffic_per_xDSL_BH_DL_FC_1$year <- PdC + 1
```

Figure 54: R code to estimate the traffic per xDSL for the 1st year.

The same occurs with the estimation of the value for two years since now.

$$FC + 2 \, Traffic \, per \, xDSL \, _{BusyHour \, _{DL/UL}} =$$

$$= FC + 1 \, Traffic \, per \, xDSL \, _{BusyHour \, _{DL/UL}} * (1 + \%Forecast \, Growth_{DL/UL})$$

The code for this implementation would be the same as before, just changing the name of variables to use.

```
Traffic_per_xDSL_BH_DL_FC_2 <- Traffic_per_xDSL_BH_DL_FC_1

Traffic_per_xDSL_BH_DL_FC_2$value <- Traffic_per_xDSL_BH_DL_FC_2$value*(1 + Percentage_growth_DL_Gbps_PdC_FC)

Traffic_per_xDSL_BH_DL_FC_2$year <- PdC + 2
```

Figure 55: R code to estimate the traffic per xDSL for the 2nd year.

The same would be for the +3rd year of estimation. Besides, identical process is done for the traffic per FTTH in Busy Hour as can be seen in the following equations.

Firstly, the forecast for the last month of the current year would be as follows.

$$QFC \, Traffic \, per \, FTTH \, _{BusyHour \, _{DL/UL}} =$$

$$= Last \, Year \, Traffic \, per \, FTTH \, _{BusyHour \, _{DL/UL}} * (1 + \%Forecast \, Growth_{DL/UL})$$

Secondly, the forecast for the last month of the following year would be as follows.

$$FC + 1 \, Traffic \, per \, FTTH \, _{BusyHour \, _{DL/UL}} =$$

$$= Current \, Year \, Traffic \, per \, FTTH \, _{BusyHour \, _{DL/UL}} * (1 + \%Forecast \, Growth_{DL/UL})$$

Finally, the forecast for the last month of the second following year would be as follows.

$$FC + 2 \, Traffic \, per \, FTTH \, _{BusyHour \, _{DL/UL}} =$$

$$= FC + 1 \, Traffic \, per \, FTTH \, _{BusyHour \, _{DL/UL}} * (1 + \%Forecast \, Growth_{DL/UL})$$

The same would be for the +3rd year of estimation. Now, the forecast for the total monthly traffic for the downlink and the uplink for both technologies (xDSL and FTTH) is going to be obtained.

$$(QFC/FC + 1 + 2 + 3)\ Monthly\ xDSL/FTTH_{BusyHour\ DL/UL} =$$

$$= Traffic\ per\ xDSL_{\ BusyHour\ DL/UL_{(QFC/FC+1+2+3)}} * xDSL/FTTH_{clients_{(QFC/FC+1+2+3)}}$$

These predictions can be achieved with the following code.

```
# Downlink:
Monthly_xDSL_BH_DL_PdC_FC <- ddply(rbind(Traffic_per_xDSL_BH_DL_PdC_FC,xDSL_clients_PdC_FC), groupBy,
numcolwise(prod))

Monthly_FTTH_BH_DL_PdC_FC <- ddply(rbind.match.columns(Traffic_per_FTTH_BH_DL_PdC_FC,FTTH_clients_PdC_FC), groupBy,
numcolwise(prod))

Monthly_xDSL_BH_DL_PdC_FC$value <- Monthly_xDSL_BH_DL_PdC_FC$value/1000
Monthly_FTTH_BH_DL_PdC_FC$value <- Monthly_FTTH_BH_DL_PdC_FC$value/1000
```

Figure 56: R code to estimate the monthly xDSL/FTTH traffic in Busy Hour.

Then, the forecast of total traffic has been obtained by aggregating of the xDSL and FTTH technology for the downlink and uplink.

$$(QFC/FC + 1 + 2 + 3)\ Total\ Traffic\ (Gbps)_{BusyHour\ DL/UL} =$$

$$= Monthly\ xDSL_{BusyHour\ DL/UL_{(QFC/FC+1+2+3)}} + Monthly\ FTTH_{BusyHour\ DL/UL_{(QFC/FC+1+2+3)}}$$

These sums have been achieved with the following instructions.

```
Total_traffic_Gbps_BH_DL_PdC_FC <- ddply(rbind(Monthly_xDSL_BH_DL_PdC_FC,Monthly_FTTH_BH_DL_PdC_FC), groupBy,
numcolwise(sum))  # Monthly_xDSL_BH_DL + Monthly_FTTH_BH_DL
Total_traffic_Gbps_BH_DL_PdC_FC$type <- 'DL'
```
Figure 57: R code to estimate the total downlink traffic (Gbps) for the Strategic Plan.

Finally, the forecast for the next three years and the last month of the current year for the fixed network in downlink and uplink in Busy Hour is obtained by aggregating the total traffic.

$$(QFC/FC + 1 + 2 + 3)\ Busy\ Hour\ FC\ Fixed\ Traffic =$$

$$= Total\ Traffic\ (Gbps)_{BH\ DL_{(QFC/FC+1+2+3)}} + Total\ Traffic\ (Gbps)_{BH\ UL_{(QFC/FC+1+2+3)}}$$

It is as simple as this line of code to integrate the final monthly fixed traffic.

```
groupBy <- c('country', 'year','month','quarter', 'datetime')

Total_traffic_Gbps_BH_DL_UL_PdC_FC <- ddply(rbind(Total_traffic_Gbps_BH_DL_PdC_FC,Total_traffic_Gbps_BH_UL_PdC_FC),
groupBy, numcolwise(sum))

Total_traffic_Gbps_BH_DL_UL_PdC_FC$type <- 'DL+UL'
```

Figure 58: R code to estimate the total DL+UL fixed traffic for the Strategic Plan.

5.5.   Acquiring actual mobile downlink & uplink traffic in Busy Hour (BH)

The mobile traffic intensity for the downlink and uplink of all countries where the company is present is needed in order to make all the calculations. This information can be found in the database.

```
Traffic_intensity_MD7_DL_real <- subset(mobile, code == 'MD-7-2gD' | code == 'MD-7-3gD' | code == 'MD-7-4gD')
Traffic_intensity_MD7_UL_real <- subset(mobile, code == 'MD-7-2gU' | code == 'MD-7-3gU' | code == 'MD-7-4gU')
Traffic_intensity_MD7_TOTAL_real <- subset(mobile, code == 'MD-7')
```

Figure 59: R code to retrieve mobile traffic intensity measurements from the database.

Although this study could be done with the 2G, 3G and 4G technologies in a separate way, the actual total traffic is requested, so all the technologies for the downlink and for the uplink are going to be aggregated.

```
groupBy <- c('country', 'year','month','quarter', 'datetime')

Traffic_intensity_MD7_DL_real <- ddply(Traffic_intensity_MD7_DL_real, groupBy, numcolwise(sum))
Traffic_intensity_MD7_DL_real$type <- 'DL'

Traffic_intensity_MD7_UL_real <- ddply(Traffic_intensity_MD7_UL_real, groupBy, numcolwise(sum))
Traffic_intensity_MD7_UL_real$type <- 'UL'

Traffic_intensity_MD7_TOTAL_real$type <- 'DL+UL'
```

Figure 60: R code to sum all the traffic intensity of all technologies.

Due to some problems in the database, not all fields have the sum of the three technologies, so in the table where the values are being retrieved, some 'MD-7' code fields could have no value, while its sub codes 'MD-7-2gD', 'MD-7-3gD' and 'MD-7-4gD' could have values, so all of them are going to be aggregated so a complete data frame of total downlink + uplink values can be obtained.

Finally, the total actual mobile traffic in Busy Hour is going to be a data frame which contains the information of total traffic for the downlink, for the uplink, and the sum of both, as can be seen in the following R code.

```
groupBy <- c('country', 'year','month','quarter', 'datetime', 'type')
BH_Mobile_Traffic_Actual <-
rbind(Traffic_intensity_MD7_DL_UL_todos,Traffic_intensity_MD7_DL_real,Traffic_intensity_MD7_UL_real)
```

Figure 61: R code to obtain the actual Busy Hour mobile traffic.

5.6.   Mobile traffic forecast

In order to make the mobile traffic prediction, the actual mobile data traffic volume that appear in the QFC and the previous months is going to be needed in order to know in the following years what mobile traffic intensity the company is supposed to have.

The mobile data traffic volume is the sum of the downlink and uplink cell level traffic volumes registered in all the Node B during the month.

The 2G technology can be obtained from the values stored in the database.

```
Total_Traffic_2G <- subset(mobile, code == 'MD-8-2gD' | code == 'MD-8-2gU')
```

Figure 62: R code to obtain the total 2G volume traffic.

For the 3G and 4G technology, the traffic average per small screen and per big screen is going to be obtained in order to have the total data traffic for this technology.

The calculation of the small screen and big screen traffic average of 3G is as follows:

$$Traffic\_per\_BS/SS\_3G_{(FC+1+2+3)} = Traffic\_per\_BS/SS\_3G_{previousYear} * (1 + SS\_3G_{client_{growth}})$$

The code for this calculation for the forecast of the following year is shown because the rest of the predictions of the +2nd year and 3rd year would be very similar.

```
# 3G = TrafficLastYear * (1 + SS_3G_client_growth)
Traffic_per_SS_3G_POS_AVG_FC_1 <- subset(Traffic_per_SS_3G_POS_AVG_PdC, year == PdC - 1)
Traffic_per_SS_3G_POS_AVG_FC_1$value <- Traffic_per_SS_3G_POS_AVG_FC_1$value*(1 + Traffic_per_SS_3G_client_growth)
Traffic_per_SS_3G_POS_AVG_FC_1$year <- PdC + 1
```

Figure 63: R code to estimate the traffic per small screen in 3G.

The calculation of the small screen and big screen traffic average of 4G is as follows:

$$Traffic\_per\_BS/SS\_4G_{AVG_{(FC+1+2+3)}} = Traffic\_per\_BS/SS\_3G_{AVG} * Coverage_{factor} * HU_{factor})$$

The code for the forecast of all years is as follows.

```
# 4G = 3G * coverage_factor * HU_factor
Traffic_per_SS_4G_POS_AVG <- Traffic_per_SS_3G_POS_AVG
Traffic_per_SS_4G_POS_AVG <- Traffic_per_SS_4G_POS_AVG$value*F1_coverage_factor*F2_HU_factor
```

Figure 64: R code to estimate the traffic per small screen in 4G.

In order to obtain the big screen clients and small screen clients traffic for 3G and 4G, first the big screen and small screen clients are estimated, and then the

traffic they would generate in the following years using the average traffic estimated before.

$$BS/SS\_4G_{(FC+1+2+3)} = BS/SS\_Clients_{(FC+1+2+3)} * \%LTE\_clients$$

$$BS/SS\_3G_{(FC+1+2+3)} = BS/SS\_Clients_{(FC+1+2+3)} - BS/SS\_4G_{(FC+1+2+3)}$$

This is translated into R code as follows. The code is similar for small screen.

```
# 4G = BS_clients * %LTE_clients
BS_4G_POS <- BS_Clients_PdC_FC
BS_4G_POS$value <- BS_4G_POS$value*Percentage_LTE_clients

# 3G = BS_clients - 4G
BS_3G_POS <- ddply(rbind(BS_Clients_PdC_FC, BS_4G_POS), groupBy, numcolwise(diff))
```

Figure 65: R code to estimate big screen clients for 3G and 4G.

Then, the traffic for 4G and 3G big screen clients and small screen clients for each country is estimated.

$$TotalTraffic\_BS/SS\_4G/3G_{(FC+1+2+3)}$$

$$= BS/SS\_3G/4G_{(FC+1+2+3)} * Traffic\_per\_BS/SS\_4G_{AVG_{(FC+1+2+3)}}$$

The code is the following, taking into account that for 3G is very similar.

```
# 4G_SS_POS
Traffic_4G_SS_POS <- rbind(SS_4G_POS,Traffic_per_SS_4G_POS_AVG)
Total_Traffic_4G_SS_POS <- ddply(Traffic_4G_SS_POS, groupBy, numcolwise(prod)) # SS_4G_POS * Traffic_per_SS_4G_POS_AVG

# 4G_BS_POS
Traffic_4G_BS_POS <- rbind(BS_4G_POS,Traffic_per_BS_4G_POS_AVG)
Total_Traffic_4G_BS_POS <- ddply(Traffic_4G_BS_POS, groupBy, numcolwise(prod)) # BS_4G_POS * Traffic_per_BS_4G_POS_AVG
```

Figure 66: R code to estimate the total traffic of big/small screen clients for 4G.

The total traffic of 2G technology is going to be a fixed value since December 2015 due to its minimum differences between months lately. Due to the fact that the small and big screen clients for the 2G have not been taken into account, the total 3G traffic value estimated before is going to be reduced a little.

$$TotalTraffic\_BS/SS\_3G_{(FC+1+2+3)} = TotalTraffic\_BS/SS\_3G_{(FC+1+2+3)} - TotalTraffic\_2G$$

Then, the monthly total traffic of these forecasts would be the sum of the three technologies.

$$MonthlyTotalTraffic_{(FC+1+2+3)} = TotalTraffic_{4GBS/SS} + TotalTraffic_{3GBS/SS} + TotalTraffic_{2G}$$

The code is the following.

```
# We join into a dataframe MD8_2g, MD8_3g y MD8_4g
Total_Traffic_2G3G4G_PdC_FC <- rbind(Total_Traffic_2G_PdC_FC,Total_Traffic_3G_PdC_FC,Total_Traffic_4G_PdC_FC)

# We sum 2g, 3g y 4g values
Monthly_Traffic_Total_PdC_FC <- ddply(Total_Traffic_2G3G4G_PdC_FC, groupBy, numcolwise(sum))
```

Figure 67: R code to estimate the monthly total traffic.

The forecast of the Busy Hour mobile traffic in the following years will be estimated with the monthly traffic obtained before, as follows.

$$BH\_MobileTraffic_{(FC+1+2+3)} = \frac{BH\_Mobile\_Traffic_{yearBefore}}{MonthlyTotalTraffic_{yearBefore}} * MonthlyTotalTraffic_{FC+1+2+3}$$

This equation in R has been implemented with the next code, the forecast for the following year (+1) is shown but it is the same with the rest of years.

```
# We obtain the forecast value for the year of FC +1
# BH_Mobile_Traffic_PdC/MD8_PdC
BH_Mobile_Traffic_FC_1 <- BH_Mobile_Traffic_PdC
BH_Mobile_Traffic_FC_1 <- rbind.match.columns.numdenom(BH_Mobile_Traffic_FC_1,Monthly_Traffic_Total_PdC)
BH_Mobile_Traffic_FC_1 <- ddply(BH_Mobile_Traffic_FC_1, groupBy, plyr::summarize, value = (value[role == "numerator"] /
value[role == "denominator"]))
# (BH_Mobile_Traffic_PdC/MD8_PdC)*MD8_FC_1
BH_Mobile_Traffic_FC_1 <- ddply(rbind(BH_Mobile_Traffic_FC_1,Monthly_Traffic_Total_FC_1), groupBy, numcolwise(prod))
BH_Mobile_Traffic_FC_1$year <- PdC + 1
```

Figure 68: R code to estimate the total mobile traffic for the Strategic Plan.

5.7. Handle and storage of the obtained data

Now that all the actual and forecast values of the carried traffic for the fixed and mobile networks of all countries of the telecommunications company have been obtained, all of them are going to be combined and stored in a database.

5.7.1. Handling the data

First of all, all the data is going to be managed, for example, all 'NaN' or very low values are going to be deleted because they are not of interest, then two columns are going to be added: network and code, in order to differentiate the type of network (fixed or mobile) and the type of value (actual or forecast).

```
# We remove the 0 values or very low (<2) since they will be wrong and
we do not want them to be shown in the graphs
BH_Fixed_Traffic_Actual$value[BH_Fixed_Traffic_Actual$value<2]<-NaN
BH_Mobile_Traffic_Actual$value[BH_Mobile_Traffic_Actual$value<2]<-
NaN
BH_Fixed_Traffic_PdC_FC$value[BH_Fixed_Traffic_PdC_FC$value<2]<-
NaN
BH_Mobile_Traffic_PdC_FC$value[BH_Mobile_Traffic_PdC_FC$value<2]<
- NaN
# Erase NaN
BH_Fixed_Traffic_Actual <-
BH_Fixed_Traffic_Actual[complete.cases(BH_Fixed_Traffic_Actual),]
```

```
# We add network == TOTAL in order to have
Fixed + Mobile for each country:
groupBy <- c('year','month','quarter', 'country',
'datetime', 'code', 'type')
BH_Traffic_Actual_Network_Total_DLUL <-
subset(BH_Traffic_Actual_PdC_FC, type ==
'DL+UL')
BH_Traffic_Actual_Network_Total_DLUL <-
ddply(BH_Traffic_Actual_Network_Total_DLUL,
groupBy, numcolwise(sum))
BH_Traffic_Actual_Network_Total_DLUL$networ
k <- 'Total'
```

```
BH_Mobile_Traffic_Actual <-                          BH_Traffic_Actual_Network_Total_DLUL$type <-
BH_Mobile_Traffic_Actual[complete.cases(BH_Mobile_Traffic_Actual),]   'DL+UL'
BH_Fixed_Traffic_PdC_FC <-
BH_Fixed_Traffic_PdC_FC[complete.cases(BH_Fixed_Traffic_PdC_FC),]   # We add network == TOTAL in order to have
BH_Mobile_Traffic_PdC_FC <-                          Fixed + Mobile for each country/type:
BH_Mobile_Traffic_PdC_FC[complete.cases(BH_Mobile_Traffic_PdC_FC),   BH_Traffic_Actual_Network_Total_DL <-
]                                                    subset(BH_Traffic_Actual_PdC_FC, type == 'DL')
                                                     BH_Traffic_Actual_Network_Total_DL <-
# We add two identification columns to join in a single dataframe all the   ddply(BH_Traffic_Actual_Network_Total_DL,
data                                                 groupBy, numcolwise(sum))
BH_Fixed_Traffic_Actual$network <- 'Fixed'           BH_Traffic_Actual_Network_Total_DL$network
BH_Fixed_Traffic_PdC_FC$network <- 'Fixed'           <- 'Total'
BH_Mobile_Traffic_Actual$network <- 'Mobile'         BH_Traffic_Actual_Network_Total_DL$type <-
BH_Mobile_Traffic_PdC_FC$network <- 'Mobile'         'DL'

BH_Fixed_Traffic_Actual$code <- 'Real'               BH_Traffic_Actual_Network_Total_UL <-
BH_Mobile_Traffic_Actual$code <- 'Real'              subset(BH_Traffic_Actual_PdC_FC, type == 'UL')
BH_Fixed_Traffic_PdC_FC$code <- 'PdC_FC'             BH_Traffic_Actual_Network_Total_UL <-
BH_Mobile_Traffic_PdC_FC$code <- 'PdC_FC'            ddply(BH_Traffic_Actual_Network_Total_UL,
                                                     groupBy, numcolwise(sum))
# We put all the data in a dataframe to insert it into the database   BH_Traffic_Actual_Network_Total_UL$network
BH_Traffic_Actual_PdC_FC <- rbind(BH_Fixed_Traffic_Actual,   <- 'Total'
BH_Mobile_Traffic_Actual, BH_Fixed_Traffic_PdC_FC,   BH_Traffic_Actual_Network_Total_UL$type    <-
BH_Mobile_Traffic_PdC_FC)                            'UL'
```

Figure 69: R code to treat all the mobile and fixed traffic data.

Moreover, some extra rows are going to be added in order to have the total carried traffic of the company, not only by country as it is right now.

```
# We join all the data + totals
BH_Traffic_Actual_PdC_FC <- rbind(BH_Traffic_Actual_PdC_FC, BH_Traffic_Actual_Network_Total_UL,
BH_Traffic_Actual_Network_Total_DL, BH_Traffic_Actual_Network_Total_DLUL)

# Add country == TOTAL to show the whole traffic of Telefónica
groupBy <- c('year','month','quarter', 'datetime', 'network', 'code', 'type')
BH_Traffic_Actual_PdC_FC_Total <- ddply(BH_Traffic_Actual_PdC_FC, groupBy, numcolwise(sum))
BH_Traffic_Actual_PdC_FC_Total$country <- 'TOTAL'

# We join all the data + totals
BH_Traffic_Actual_PdC_FC <- rbind(BH_Traffic_Actual_PdC_FC, BH_Traffic_Actual_PdC_FC_Total)
```

Figure 70: R code to add the total traffic of all the company.

5.7.2. Storing in a database

All these values have to been stored in a database in order to use them in the Shiny web application.

A connection is made to the MariaDB database, all the values stored in the table "Traffic_Forecast" are erased and then all the values that are in the data frame are stored into that table. It can be seen how this is done with the subsequent code.

```
# Open the connection to the database
odbcDataSources("all")
WTACv2 <- odbcConnect(dsn = "WTAC", uid = "user", pwd = "pass")

# Empty the corresponding table
sqlQuery(WTACv2, "TRUNCATE TABLE Traffic_Forecast;")

# We sort the dataframe according to what the database will need
```

```
ordered_columns_leftside <- c('country','year','month', 'quarter', 'datetime', 'value', 'network', 'code', 'type')
BH_Traffic_Tempbulk <- BH_Traffic_Actual_PdC_FC[c(ordered_columns_leftside,
setdiff(names(BH_Traffic_Actual_PdC_FC),ordered_columns_leftside))]

# We store the data in the database
write.table(BH_Traffic_Tempbulk,"/data/WTAC/R/DPI/Web/GlobalReport_Jorge/tempbulk.txt",quote = FALSE,sep =
"\t",row.names = FALSE,col.names = FALSE,append = FALSE,na = '\\N')
Sys.chmod("/data/WTAC/R/DPI/Web/GlobalReport_Jorge/tempbulk.txt",mode = "0770")
sqlQuery(WTACv2,"LOAD DATA LOCAL INFILE '/data/WTAC/R/DPI/Web/GlobalReport_Jorge/tempbulk.txt' INTO TABLE
Traffic_Forecast")

# We show that it has gone correctly
cat("OK – Added ", nrow(BH_Traffic_Tempbulk), "fields into the database")

# Close the connection with the database
odbcClose(WTACv2)
```

Figure 71: R code to store all the values in a database.

5.8.   Representation in the web application

To access all the data in order to make analysis and comparisons of the carried traffic of a telecommunications company, we are going to click in the "Fixed & Mobile Traffic" tab of the Shiny web application as can be seen.
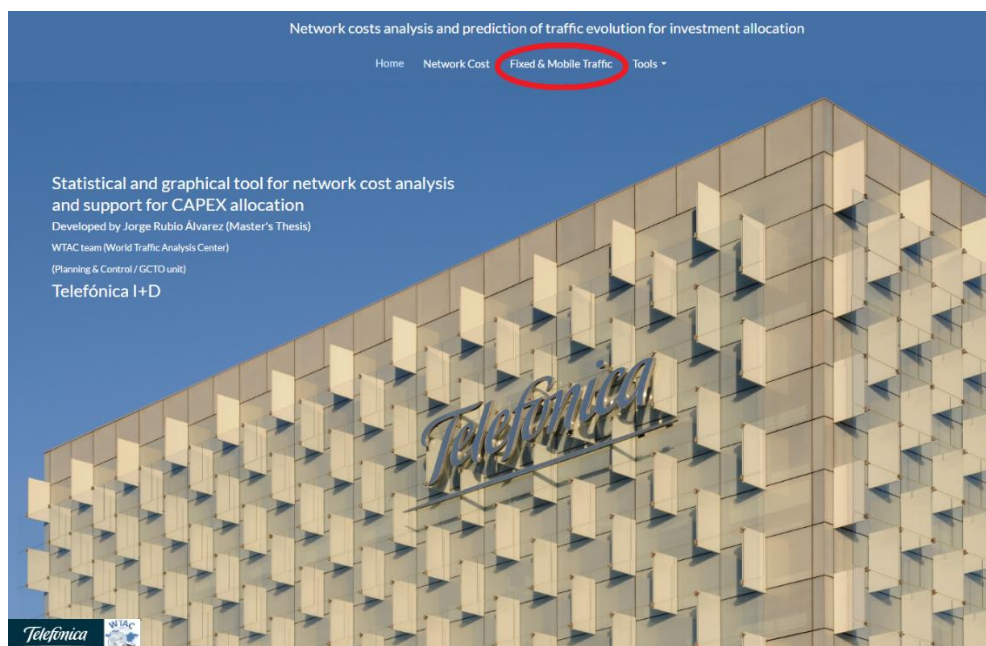

Figure 72: "Fixed & Mobile Traffic" tab view from the index webpage.

This fixed and mobile traffic section is divided into two subsections: traffic forecast which shows all the information in a table and graphic traffic forecast which shows a chart of the carried traffic past, present and future.

It can be seen how all these sections look like in the following picture.
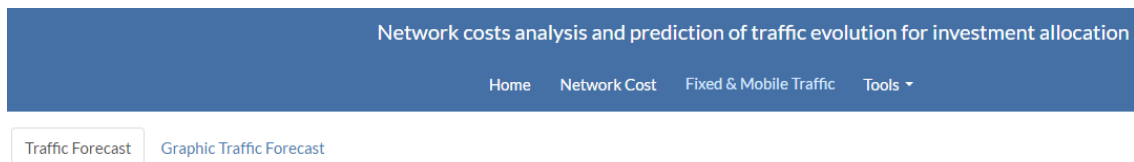
Figure 73: Subsections of fixed and mobile traffic.

5.8.1. Segmenting for traffic interpretation

The carried traffic data can be filtered by the sum of fixed and mobile networks, only fixed or only mobile network; downlink, uplink or the sum of both downlink and uplink; by countries or the values for the whole telecommunications company and on the graphic representation, values of the last month of each year or all months can be shown.

It can be seen each type of data filtering in the following picture.

Figure 74: Carried traffic filtering inside the web application.

5.8.2. Accessible and filterable traffic data

The first developed subsection of carried traffic is a table containing all the information of actual and forecast values. It can be seen all the types of networks, each carried traffic in Gbps units, to which country it belongs and more. This useful

67

representation of the carried traffic data has been achieved by writing in R the following code. The table is filtered by the last month of the year because it is the most representative value.
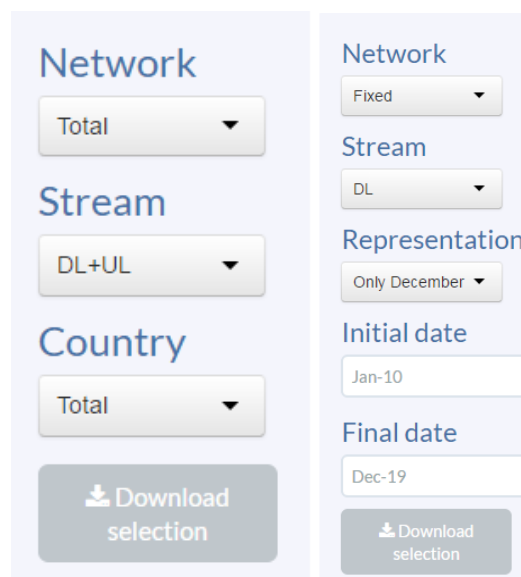
Inside the ui.R, which helps to divide the sections of the web application and to add all the selectable inputs to filter the data, this code has been written.

```
tabPanel(
"Fixed & Mobile Traffic",
includeHTML("www/FondoGlobalReport.html"),
tabsetPanel(
id = "Fixed & Mobile Traffic",
tabPanel("Traffic Forecast",
fluidPage(
h2("Traffic Forecast"),
h3("by network and country"),
#Instruction so that the values in the table are aligned to the
right
tags$head(tags$style(".table .alignRight {text-align:right;}")),
sidebarLayout(position = "right",
sidebarPanel(
width = SIDEBARPANELWIDTH,
selectInput(
 inputId = "networkGlobalTrafficForecast",
 label = h4("Network"),
 choices = c('Total','Fixed', 'Mobile')),
```

```
selectInput(
 inputId = "streamGlobalTrafficForecast",
 label = h4("Stream"),
 choices = c('DL+UL','DL', 'UL')),
selectInput(
 inputId = "countryGlobalTrafficForecast",
 label = h4("Country"),
 choices =
c('Total','C1','C2','C3','C4','C5','C6','C7','C8','C9','C10',
'C11','C12','C13','C14','C15','C16','C17','C18','C19')),
downloadButton('downloadGlobalTrafficForecast', 'Download
selection')),
sidebarPanel(
width = MAINPANELWIDTH,
h3(textOutput("myTitleTrafficForecast"), align = "center"),
dataTableOutput("myTableTrafficForecast"))
)
)),
```

Figure 75: R code of the user interface to enable the carried traffic table section.

Inside the server.R, which allow to filter the data, this code has been written.

```
output$myTableTrafficForecast <- renderDataTable({

TrafficForecast <- subset(BH_Traffic, month == 12)

if (input$networkGlobalTrafficForecast == 'Total'){
heading <- "Fixed+Mobile Traffic Forecast"
} else if (input$networkGlobalTrafficForecast == 'Fixed'){
heading <- "Fixed Traffic Forecast"
} else if (input$networkGlobalTrafficForecast == 'Mobile'){
heading <- "Mobile Traffic Forecast"
}
# We show the data only according to the chosen network
(Fixed / Mobile / Total)
TrafficForecast <- subset(TrafficForecast, network ==
input$networkGlobalTrafficForecast)
# We show data only according to the chosen stream (DL + UL
/ DL / UL)
TrafficForecast <- subset(TrafficForecast, type ==
input$streamGlobalTrafficForecast)
# We show the data only according to the chosen country
(TOTAL=telefónica)
if (input$countryGlobalTrafficForecast == 'Total'){
TrafficForecast <- subset(TrafficForecast, country == 'TOTAL')
```

```
} else {
TrafficForecast <- subset(TrafficForecast, country ==
input$countryGlobalTrafficForecast)
}
# We select the columns we want to display on the web (we
remove datatime y quarter)
TrafficForecast <- subset(TrafficForecast, select = c('country',
'year', 'month', 'network', 'code', 'value'))

colnames(TrafficForecast) <- c('Country', 'Year', 'Month',
'Network', 'Real/PdC/FC', 'Value (Gpbs)')

output$downloadGlobalTrafficForecast <- downloadHandler(
filename = paste('TrafficForecast-
',input$countryGlobalTrafficForecast,'-
',input$networkGlobalTrafficForecast,'.csv')
,content = function(file) {
 write.csv2(TrafficForecast, file,row.names = FALSE)}
)

output$myTitleTrafficForecast <- renderText({heading})
return(TrafficForecast)
})
```

Figure 76: R code of the server to enable the carried traffic table section.

It can be seen how all this R code is transformed to an attractive and functional table with all the information about carried traffic and its predictions thanks to the Shiny package.

Figure 77: Table to access and filter fixed and mobile carried traffic information.

It can be seen an example of how to filter the carried traffic data with the following picture. All the sum of fixed and mobile networks from the whole telecommunications company and only the downlink is going to be shown.
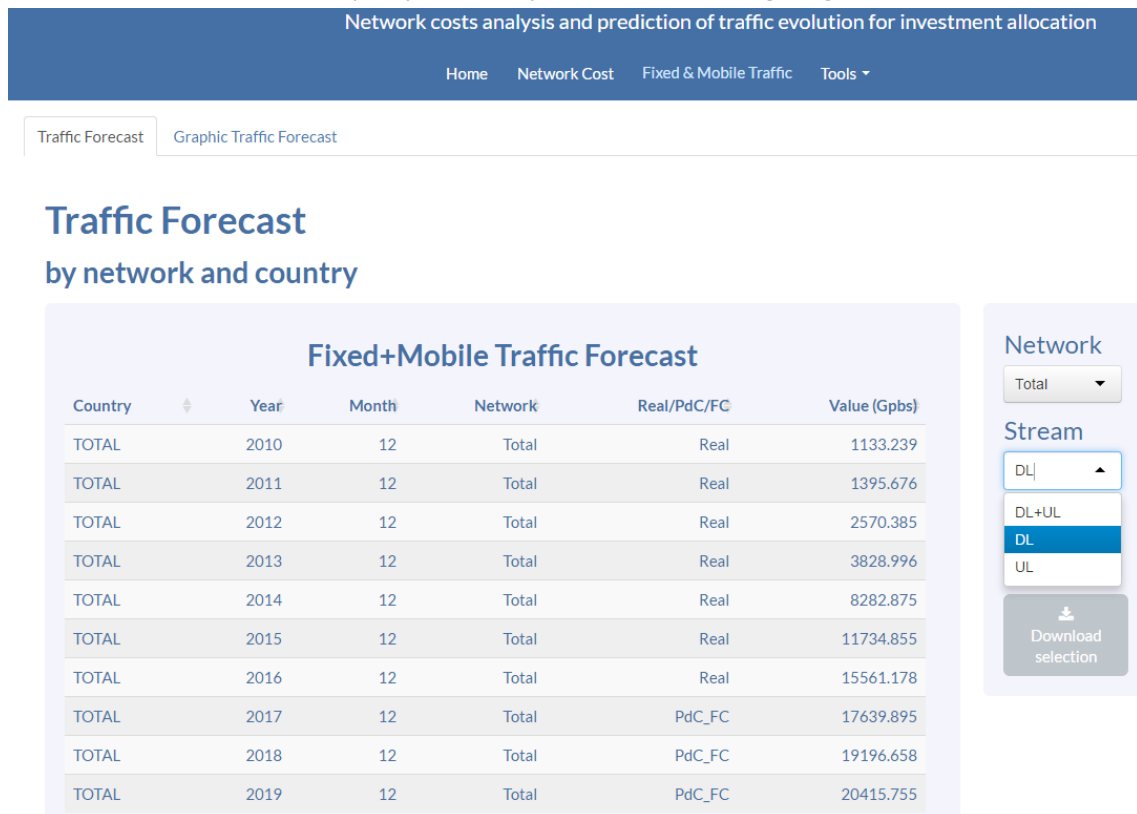

Figure 78: Example of filtering carried traffic by stream.

Another example of filtering could be by country, picking one operating business (OB) and only its fixed network and the uplink stream as can be seen in the picture.
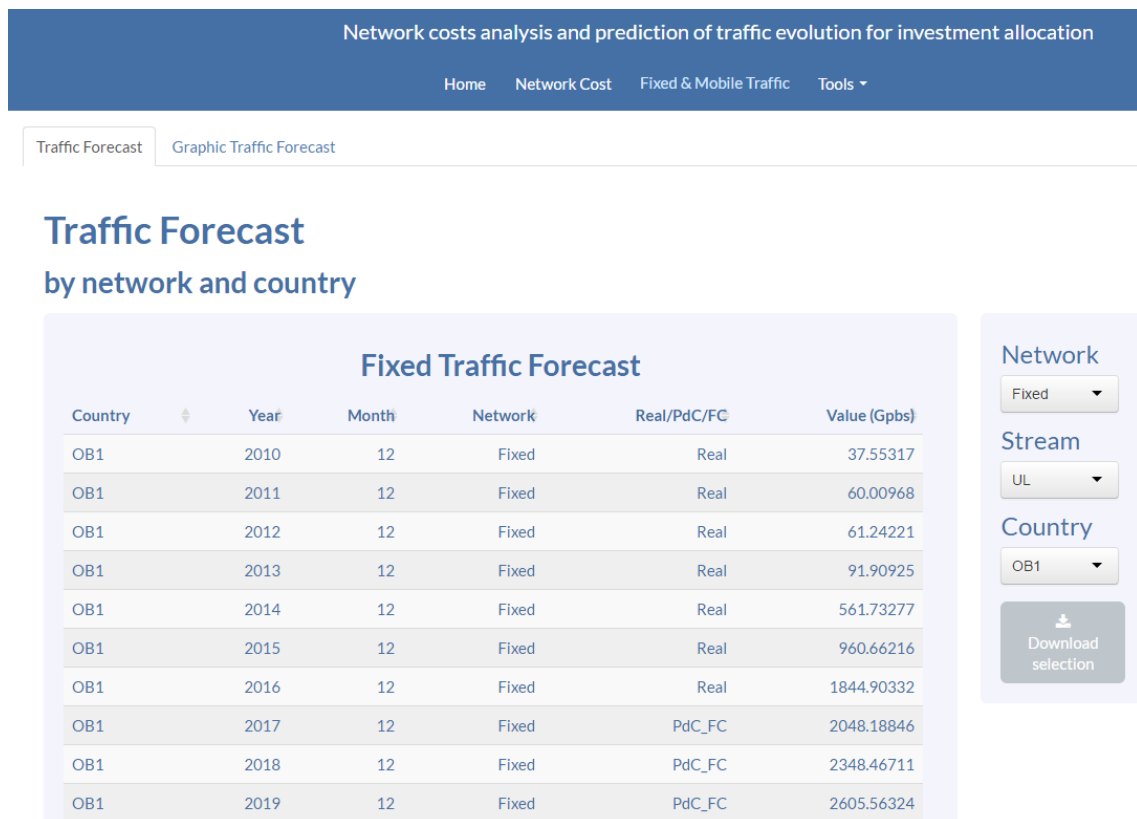
Figure 79: Example of filtering carried traffic by network, stream and country.

Furthermore, a "Download selection" has been implemented button, as it was done with the network costs tab, with the downloadHandler function in R, in order to let anyone to download all the data in a CSV file. It can be seen how this information would appear in this file in the following table.

Country	Year	Month	Network	Real/PdC/FC	Value (Gbps)
OB1	2010	12	Fixed	Actual	262.5060
OB1	2011	12	Fixed	Actual	240.0204
OB1	2012	12	Fixed	Actual	366.0033
…	…	…	…	…	…

Figure 80: CSV example of download carried traffic function.

5.8.3. Graphical illustration of traffic's evolution over time

The last subsection shows a graph of the carried traffic for fixed and mobile networks in a concrete period of time and its predictions for the following three years, which would be add to the strategic plan of the company.

This is very helpful in order to see easily which countries have more traffic in different situations: fixed or mobile network, downlink or uplink stream and how is going to be their growth. All these values have been anonymized, so they do not correspond to the actual values of Telefónica operations, but these values can be used as a way to demonstrate the powerfulness of this tool.

On the one hand, the ui.R code part is very similar to the previous section, just with some changes, because in this case a graph is going to be shown, not a table, and the possibility of viewing all the values of all months is being added, not only the most representative value which is December's values. Some lines of code from the previous section have been changed, which are the following.

```
selectInput(
 inputId = "monthsTrafficForecast",
 label = h4("Representation"),
 choices = c('Only December','All months')),
dateInput2(
 "InitialDateGraphicTrafficForecast",
 h4("Initial date"),
 value = as.Date(InitialDate)),
dateInput2(
 "FinalDateGraphicTrafficForecast",
 h4("Final date"),
 value = as.Date(FinalDate)),
sidebarPanel(
width = MAINPANELWIDTH,
h3(textOutput("myTitleGraphicTrafficForecast"), align = "center"),
showOutput("myChartGraphicTrafficForecast", "nvd3")))
))
```
Figure 81: R code of the user interface to enable the graphic carried traffic section.

On the other hand, the server.R code part is very similar to the previous section too, just with some changes to show a graph.

```
output$myChartGraphicTrafficForecast <- renderChart({
if (input$monthsTrafficForecast == 'Only December'){
BH_Traffic_FC <- subset(BH_Traffic, month == 12)
} else {
BH_Traffic_FC <- BH_Traffic
}
// rest of the code is similar to the previous section
p <- pintaKPITimeEvolByCountry(BH_Traffic_FC, columnX = 'datetime', columnY = "value", grupo = "country", graphType =
'lineChart', units = 'Traffic (Gbps)', chartName = 'myChartGraphicTrafficForecast', melt = FALSE, initial_date = initial_date,
final_date = final_date)
return(p)
```
Figure 82: R code of the server to enable the graphic carried traffic section.

It can be seen this graphical representation of the fixed and mobile carried traffic and its predictions for the following years as follows.
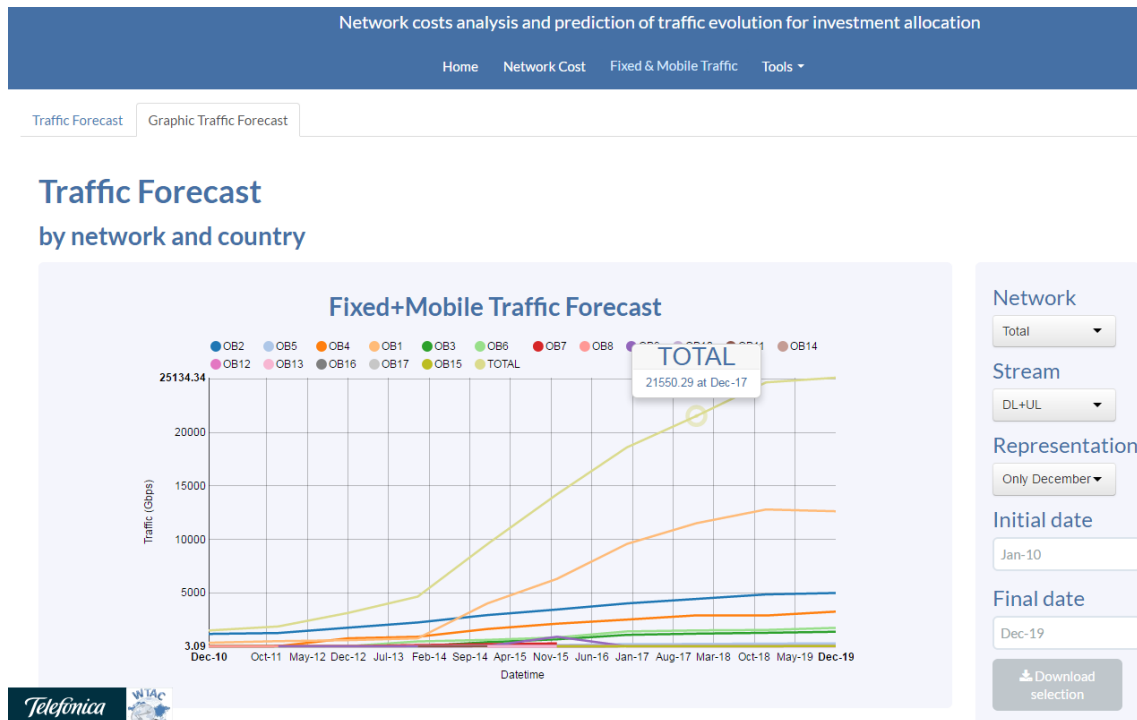
Figure 83: Graph of carried traffic evolution.

It can be seen all months, which will show some fluctuations due to diverse factors as summer when people use more data on their mobile phones and other influences.
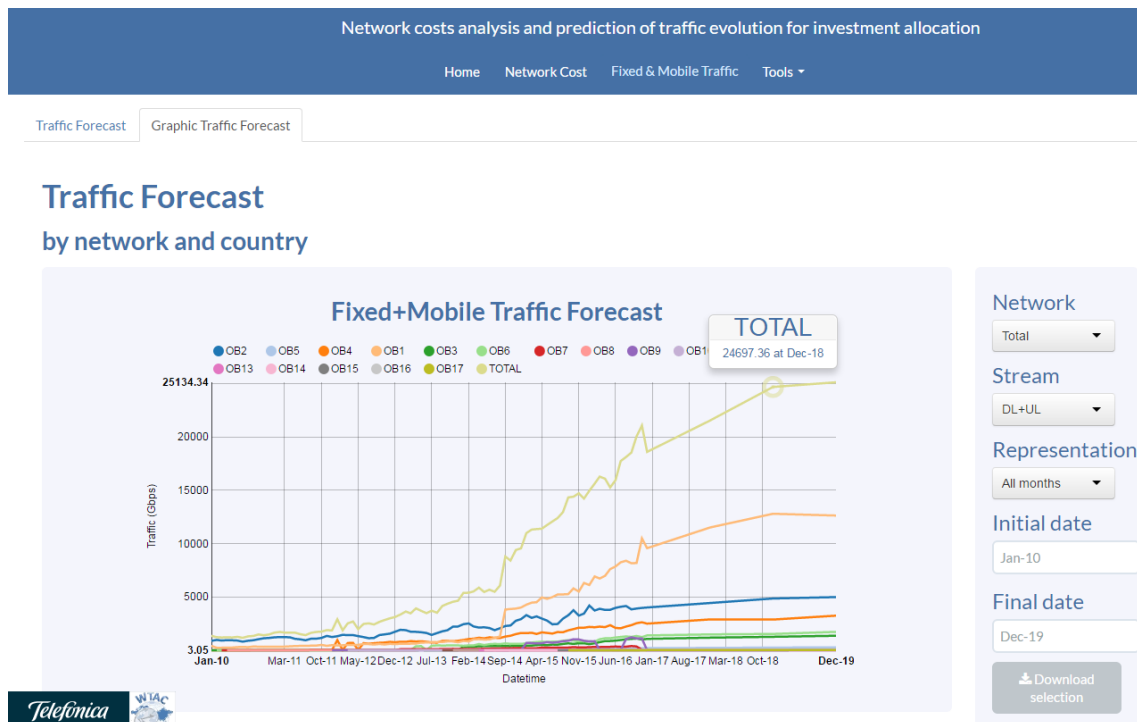
Figure 84: Example of monthly carried traffic evolution.

72

Comparisons can be made between only some countries and, for example, the downlink stream of their fixed networks as can be seen here with some operating business evolutions.
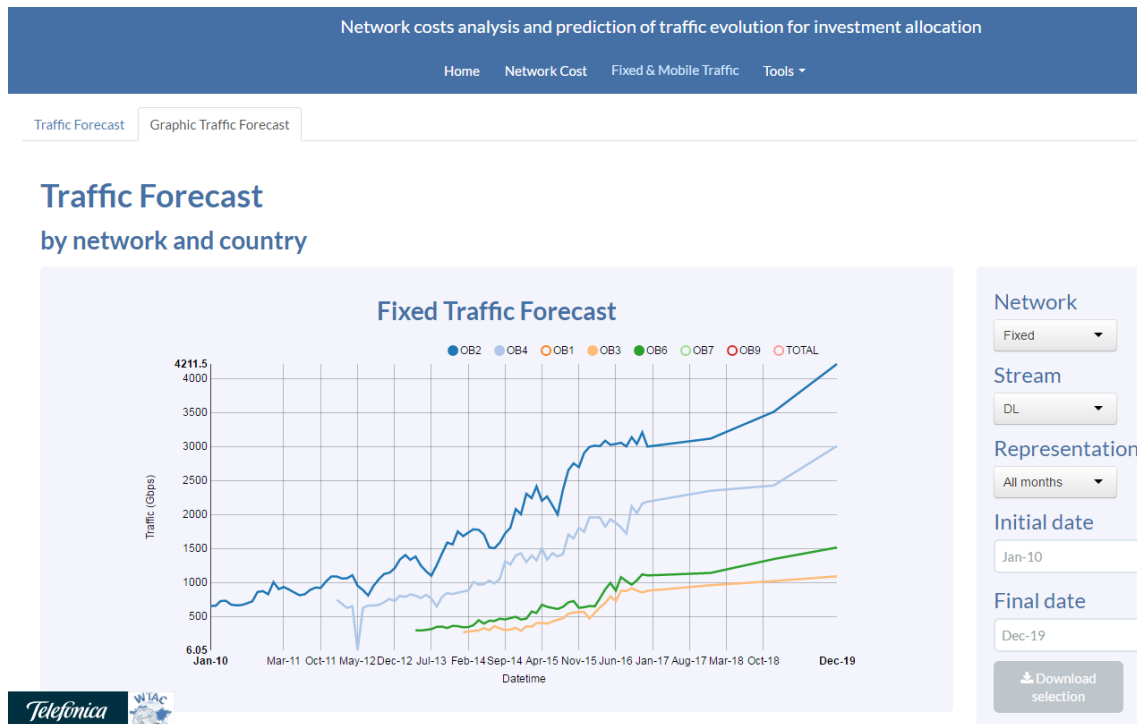
Figure 85: Example of downlink fixed traffic comparison of various countries.

As it has been said, it is better to see only the last month of every year, because this will give a much smoother evolution of the traffic, as can be seen with the following example.
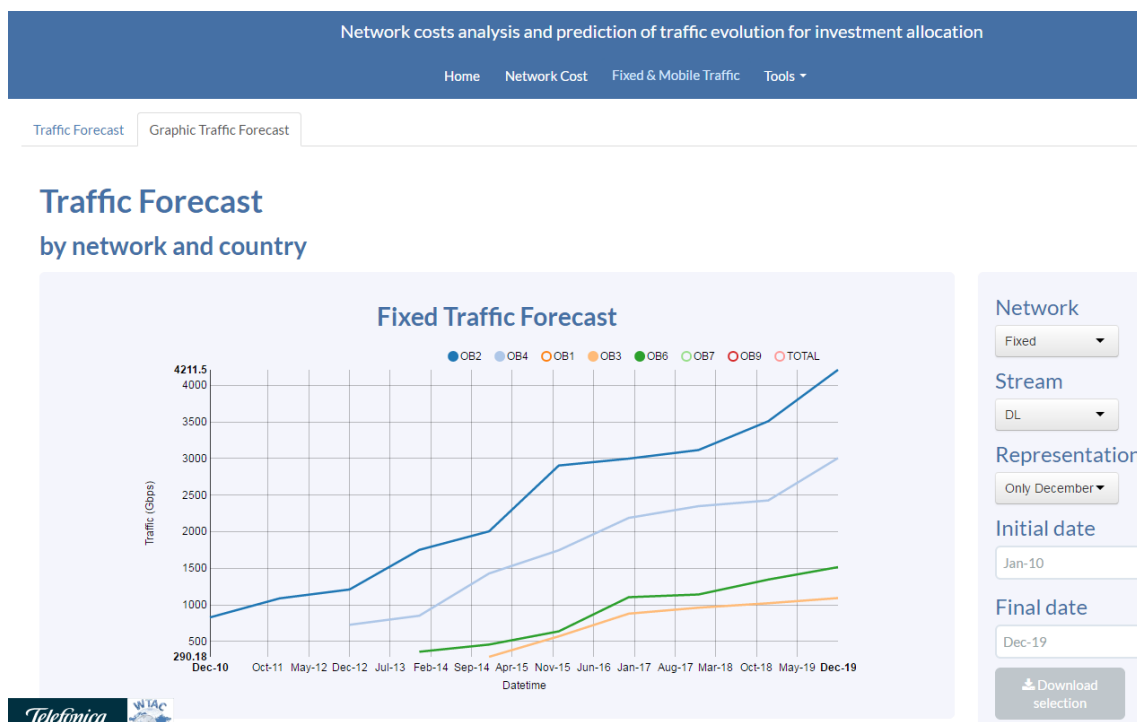
Figure 86: Example of smoother downlink fixed traffic comparison.

73

Moreover, the period of time that is wanted to be analyzed can be adjusted, for example, it can be seen the total traffic of all countries from January 2015 until the prediction for December 2019.
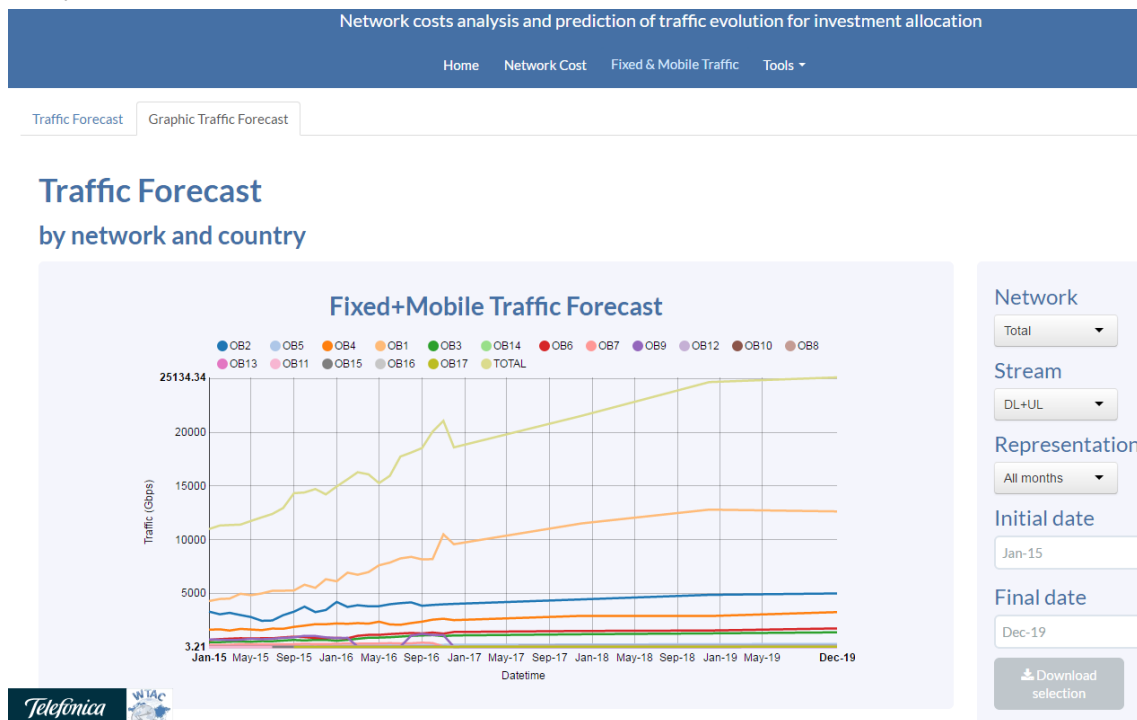
Figure 87: Example of carried traffic in a concrete period of time.

Obviously, with this filter by period of time, only the actual values of carried traffic can be analyzed by putting the final date in the final actual value, in this case would be December 2016. It can be seen it in the next picture.
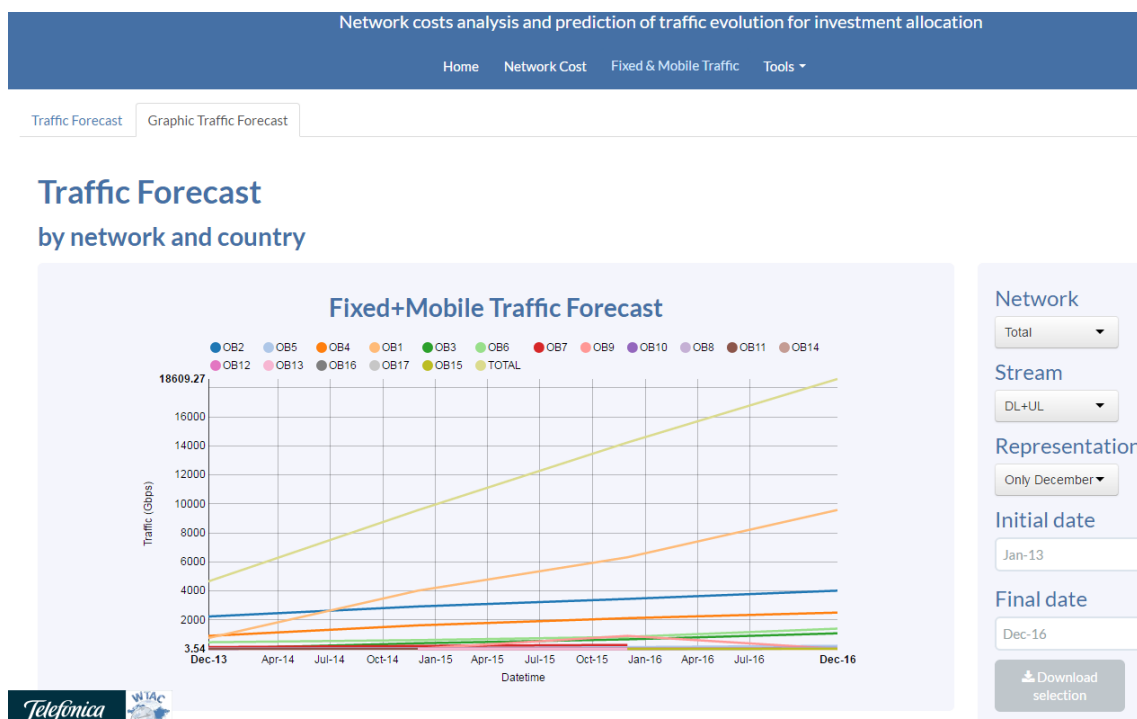
Figure 88: Example of only actual carried traffic evolution.

74

As you can see, the download functionality has been also put to extract the information that appears in the graph, so anybody can use it for his or her own purposes.

5.9.   Exploiting traffic data for CAPEX allocation

Thanks to table and chart of the web application that has been shown for the fixed and mobile carried traffic, an analysis of how all the traffic has been in the previous months and years in a company can be done and it can be seen the estimations made by this tool for the following three years, which would be included in the strategic plan (SP) of the company.

Although this data showed in the table and chart of traffic is not the true values of Telefónica, it can be seen how the predicted values is that the global carried traffic of the company is going to grow in the following years, which is expected to be like this.

Thanks to this vision of how is going to grow the traffic in each country and how it has been growing in the past years, the company can invest its capital expenses (CAPEX) in a more precise way because it would invest more in the countries that are expected to have a more growth of carried traffic, so the company can be prepared to this growth some months or years earlier rather than having to invest in the exactly moment when an operator sees that its traffic has grown so fast that has no capability to carry it out.

Besides, all this information of the table and chart can be downloaded in a CSV file in order to treat it in R or Excel or another programming language to make a more exhausted analysis of this data.

Chapter 6

Time of development, economic budget, conclusions and future guidelines

6.1.   Time of development

In this section, the phases of the project development and the planning are going to be described, identifying the important points.

A Gantt diagram is going to be employed, which contains the disaggregated tasks of the completed project. The summary of all the tasks is going to be shown along with the complete diagram.

The time used in this project has been projected taking into account that the time of accomplishment of the work is around 360 hours. This number of hours has been divided into 20 weeks of which have been assigned to the different project activities.

In all phases, both the student and the tutor in Telefónica I+D have participated, establishing a follow-up through face-to-face meetings and also with the tutor at the University Carlos III of Madrid through electronic checking. These follow-ups have been carried out throughout the project in order to follow a correct methodology of work.

The different project activities are shown in the following table with their duration in hours. The sum of each task separately cannot be done because of the fact that some tasks overlapped with others in time.

	Task description	Duration	Start	End
1	Identification of the needs and approach of the problem	20 hours	tue 06/10/16	wed 12/10/16
2	Search for bibliographic information	28 hours	mon10/10/16	thu 18/10/16
3	Research on network analysis and CAPEX allocation	60 hours	wed 12/10/16	thu 01/11/16
4	Study of the technologies to be used	32 hours	wed 26/10/16	fri 04/11/16
5	Analysis of KPIs definitions	18 hours	tue 03/11/16	wed 09/11/16
6	Installation and configuration of the R environment	12 hours	mon07/11/16	wed 09/11/16
7	Gathering all the data	44 hours	mon07/11/16	mon21/11/16
8	Data processing of all costs and traffic evolutions	140 hours	mon21/11/16	fri 06/01/17
9	Development of the Shiny web application	54 hours	sun 25/12/16	wed 11/01/17
10	Software testing	20 hours	thu 10/01/17	mon16/01/17
11	Realization of report and presentation	88 hours	thu 10/01/17	wed 08/02/17
12	Review and modification of the report	24 hours	wed 08/02/17	mon 13/02/17

Figure 89: Timetable of all the project activities.

The Gantt diagram corresponding to the temporal planning of the present project is shown in the following figure.
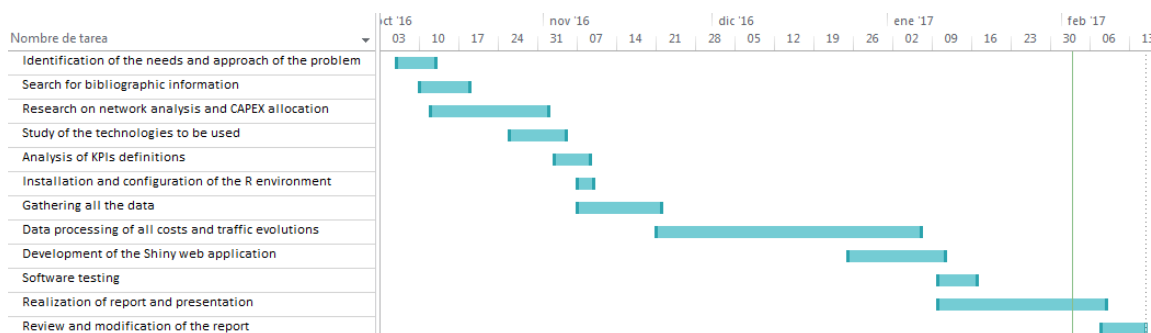
Figure 90: Gantt diagram.

The different tasks are explained in the legend that appears on the figure above so it can be observed exactly how much time each task has taken in this project along its development. As it has been said, some of the tasks have been overlapping each other; that is why two tasks can appear in the same days in the timeline.

6.2.   Economic budget

Then a study of the cost of the final project is going to be carried out, including the expenses of the personnel involved, equipment and services. All the factors that have influenced in the creation of this tool of analysis and representation of costs and traffic prediction for supporting CAPEX allocation are going to be analyzed and a final budget is going to be obtained.

The final budget has been divided into two sections: material costs as for example the hardware and software used in the development of the project and the personnel costs involved in its implementation. The budget is divided by type of costs in order to offer a better view of what the expenses have been.

As personnel costs, three people have participated in the development of the project: Adrián Maeso Martín-Carnerero and Pablo Basanta Val, both of whom are senior engineers, and Jorge Rubio Álvarez as a junior engineer.

As material costs, a series of material resources have been used in the realization of this project: computers, pen drives, servers, Windows 7 licenses or Microsoft Office Suite between others.

A person who obtains a Master in Telecommunications Engineering and that previously had obtained the Bachelor's Degree in Telecommunications Technologies Engineering will have an annual salary of about 30,000€, if this person has few or no work experience. Taking into account the working hours that sums a full year, 16.74€ per hour for a junior engineer is obtained. It can be assumed that a senior engineer will get at least 22.34€ per hour.

For the realization of this project, it has been used open and free software but also private. To calculate the cost, amortizations are going to be employed, that is, divide the cost between the years that the equipment or software is estimated to last. Besides, there are some indirect costs that must be taken into account too, such as electricity, heating, internet connection or office rental.

It can be seen in the following figure all the costs that this project has had.

CONCEPT	DESCRIPTION	COST
2 senior engineers	Approximately 120h	22.34€/h*120h*2 = 5,361.6€
1 junior engineer	Approximately 360h	16.74€/h*360h = 6,026.4€
Back-end and Front-end servers	Total cost: 2400€ Amortization: 5 months	166.67€
RStudio Shiny Server Pro	$9,995 per year Amortization:	644.58€
Microsoft Office	Total cost: 119€ Amortization: 5 months	16.52€
Computer to develop the project with Windows 7	Total cost: 600€ Amortization: 5 months	41.67€
HeidiSQL	Free	0€
Office supplies	Pens and papers	8€
Indirect costs	Electricity, Internet…	195€
	TOTAL:	12,460.44€

Figure 91: Project costs.

1. Author:

    Jorge Rubio Álvarez

2. Department:

    World Traffic Analysis Center (Telefónica I+D)

3. Description of the project:

Title:	Statistical and graphical tool for network cost analysis and support for CAPEX allocation
Duration (months):	5

4. Total budget of the project (value in Euros):

    12,460.44€

5. Budget breakdown:

CONCEPT	COST
2 senior engineers	5,361.6€
1 junior engineer	6,026.4€
Back-end and Front-end servers	166.67€
RStudio Shiny Server Pro	644.58€
Microsoft Office	16.52€
Computer to develop the project with Windows 7	41.67€
HeidiSQL	0€
Office supplies	8€
Indirect costs	195€
Total	**12,460.44€**

The total budget for this master thesis project rises to 12,460.44€.

Leganés on February 8, 2017

The project engineer

Jorge Rubio Álvarez

6.3.   Conclusions

In this section, a series of deductions that have been obtained both during the realization of the present project, as well as to its ending, are going to be presented. In addition, the potential benefits that the project is expected to report in the long term will be outlined.

On the one hand, a review of the results achieved with the development of this tool inside a company is going to be made.

- It has been achieved the objective of the project, which had been to design and implement a tool that would allow the analysis of data from the networks' costs of a telecommunications company in order to be able to see where all the company's expenses are being spent.
- It has been achieved another objective, the analysis of the evolution of networks' traffic and its representation.
- The present project has been able to forecast the networks' traffic in the following years to support CAPEX allocation.
- It has been designed of web application in order to be able to make different comparisons of networks' costs between different types of cost such as CAPEX or OPEX, diverse countries or organizations, types of networks and the deviations from the predicted cost inside the quarterly forecast closure (QFC) and the budget (BGT).
- It has been designed of web application in order to be able to make different comparisons of carried traffic between different countries, organizations, types of networks and their evolutions amongst months and thus be able to appreciate the differences that arise between them.
- In short, it has been made a tool that will allow a telecommunications company to analyse all types of data from both the financial and the communications field, by this it is meant to costs and networks' traffic.

On the other hand, me conclusions that have been drawn on a personal level have been achieved, among which the following stand out.

- The present project has allowed to integrate all the knowledge acquired during the Degree and the Master in Telecommunications Engineering, making improvements in disciplines such as programming, communication protocols, mathematical analysis, data processing, use of databases, infrastructure and costs of a network and more.

- As a personal experience, I have been able to work and have contact with specialists from different fields such as the sales department and the world traffic analysis centre (WTAC) of Telefónica's networks.
- It has been a personal challenge to realize the integration of multiple information as distinct as traffic intensity, CAPEX costs, types of users in fixed and mobile networks and more.
- As a student, I have acquired extensive knowledge in the development of R web applications and the use of R programming language to treat data and extract relevant information from it.

The potential benefits of this project in the long term are that everyone can take this information and treat it as they want in order to analyse it in their way and obtain some conclusions about how the CAPEX or OPEX are being spent and where the company should invest in the future based on the traffic forecast of its networks.

After many hours of work, the accomplishment of the scheduled tasks, great effort and study, it can be concluded that the objectives proposed at the beginning of the realization of this document have been achieved, both the tangible objectives in terms of the development of the tool and the proposed teaching targets.

In addition, the results have been detailed in a comprehensive way with tables and graphs, making possible a continuum line for future projects.

6.4.   Future guidelines

Although the present project ends with a functional final solution, many future lines of work are opened that can be approached in upcoming works.

Some modifications could be incorporated to this tool in order to improve its appearance and include other functionalities to make its use easier. Among the most important are the following.

- Expansion of analysis functionalities for the networks' traffic in order to take relevant information of the evolution of these networks in each country.
- Increase the type of analysis of network costs in order to have a more detailed view of each type of cost or concept.
- Use of economic models to check if the way the CAPEX and OPEX are being spent is the right way or they should be spent in other ways.

- It would be interesting, with the availability of more time, to try to correlate the costs and traffic data that have been used for this project in order to obtain significant results of the way an investment affects the growth of a network's traffic.

- Furthermore, due to the lack of time and resources there has been some information that has not been handled but it could help to improve this tool in order to show a more detailed view of costs and traffic.

Bibliography

[1] L. M. Walther and C. J. Skousen, Cost Analysis: Managerial and Cost Accounting, Bookboon, 2010.

[2] A. C. Elliott and W. A. Woodward, Statistical Analysis Quick Reference Guidebook: With SPSS Examples, SAGE Publications, 2006.

[3] E. J. Zajac, "FROM TRANSACTION COST TO TRANSACTIONAL VALUE ANALYSIS: IMPLICATIONS FOR THE STUDY OF INTERORGANIZATIONAL STRATEGIES," *Journal of Management Studies,* vol. 30, no. 1, p. 131–145, 1993.

[4] K. Casier, "A fair cost allocation scheme for CapEx and OpEx for a network service provider," in *Proceedings of CTTE2006, The 5th Conference on Telecommunication Techno-Economics*, 2006.

[5] M. Youssef, "Translucent network design from a CapEx/OpEx perspective," *Photonic Network Communications,* vol. 22, no. 1, p. 85–97, 2011.

[6] S. Verbrugge, "Methodology and input availability parameters for calculating OpEx and CapEx costs for realistic network scenarios," *Journal of Optical Networking,* vol. 5, no. 6, pp. 509-520, 2006.

[7] A. Rindfleisch, "Transaction Cost Analysis: Past, Present, and Future Applications," *Journal of Marketing,* vol. 61, no. 4, pp. 30-54, 1997.

[8] E.Y., "Optimize network OPEX and CAPEX while enhancing the quality of service," EYGM Limited, 2014.

[9] J. Chen, "Cost vs. reliability performance study of fiber access network architectures," *IEEE Communications Magazine,* vol. 48, no. 2, pp. 56-65, 2010.

[10] PwC, "We need to talk about Capex: Benchmarking best practice in telecom capital allocation," PricewaterhouseCoopers, 2012.

[11] T. Giles, "Cost drivers and deployment scenarios for future broadband wireless networks - key research problems and directions for research," *IEEE 59th Vehicular Technology Conference,* vol. 4, pp. 2042-2046, 2004.

[12] R. J. A. Little, Statistical Analysis with Missing Data, John Wiley & Sons, 2014.

[13] G. Keppel, Design and analysis: A researcher's handbook, Prentice-Hall, 1991.

[14] I.T.U., "Traffic intensity measurement principles," ITU-T Recommendation E.500, 1998.

[15] Telefónica I+D, "GCTO KPI Definitions," 2015.

[16] J. Hillebrand, Mastering RStudio – Develop, Communicate, and Collaborate with R, PACKT, 2015.

[17] R. D. Peng, R Programming for Data Science, Leanpub, 2015.

[18] H. G. Resnizky, Learning Shiny, PACKT, 2015.

[19] J. Verzani, Getting Started with RStudio: An Integrated Development Environment for R, O'Reilly Media, 2011.

[20] H. Halvorsen, "Structured Query Language," University College of Southeast Norway, 2016.

[21] D. Bartholomew, *MariaDB vs. MySQL,* E&B Software, 2012.

[22] P. Coletti, *Databases course book,* Free University of Bolzano Bozen, 2014.

[23] RStudio, "Shiny Official Page," [Online]. Available: shiny.rstudio.com. [Accessed 14 December 2016].

[24] R. Consortium, "RStudio Official Page," [Online]. Available: www.rstudio.com. [Accessed 10 December 2016].

[25] M. Foundation, "MariaDB Official Page," [Online]. Available: mariadb.org. [Accessed 18 December 2016].

[26] A. Becker, "HeidiSQL Official Page," [Online]. Available: www.heidisql.com. [Accessed 19 December 2016].

Attachments

Authorization for the use of Telefónica's data

Telefónica

Telefónica
Investigación y
Desarrollo

Sr. D. Jorge Rubio Álvarez (C/María Eugenia Milleret, 14, Bajo C, Ponferrada – León)

Estimado señor Jorge Rubio Álvarez:

De conformidad con lo establecido en el Convenio de Colaboración firmado entre la Universidad Carlos III de Madrid y Telefónica I+D para el Desarrollo de Programas de Prácticas de Alumnos Universitarios, mediante la presente, le autorizamos la utilización - exclusivamente en su Trabajo Fin de Máster - de la información relativa a "Statistical and graphical tool for network cost analysis and support for CAPEX allocation" a la que ha tenido acceso durante su colaboración como becario en Telefónica Investigación y Desarrollo S.A. Unipersonal.

Cualesquiera otra utilización o publicación de dicha información o de cualesquier otra a la que haya tenido acceso durante las prácticas antes mencionadas, requerirá la autorización previa y escrita de Telefónica I+D.

En prueba de conformidad con lo que antecede, le ruego firme al pie de uno de los dos originales de esta carta.

Reciba un cordial saludo

En Madrid a 14 de Septiembre de 2016.

D. Jorge Rubio Álvarez

Conforme, EL ALUMNO

D. Adrián Maeso Martín-Carnerero
Cargo en Telefónica I+D
Tutor/responsable del trabajo en TID

Telefónica Investigación y Desarrollo, S.A. Unipersonal
www.tid.es