

Monitorización del ancho de banda de redes usando wavelets

Alejandro Acedo Fajardo

Máster en Ingeniería de Telecomunicación



MÁSTERES
DE LA UAM
2017 - 2018

Escuela Politécnica Superior

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE MÁSTER

Monitorización del ancho de banda de redes usando wavelets

Máster Universitario en
Ingeniería de Telecomunicación

Autor: ACEDO FAJARDO, Alejandro

Tutor: GARCÍA DORADO, José Luis

Ponente: LÓPEZ DE VERGARA MÉNDEZ, Jorge E.

Dpto. de Tecnología Electrónica de las Comunicaciones

FEBRERO 2018

Monitorización del ancho de banda de redes usando wavelets

Autor: ACEDO FAJARDO, Alejandro

Tutor: GARCÍA DORADO, José Luis

Ponente: LÓPEZ DE VERGARA MÉNDEZ, Jorge E.

Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Febrero de 2018

Monitorización del ancho de banda de redes usando wavelets

Alejandro Acedo Fajardo

Palabras clave: Monitorización de redes, medidas de red, análisis de tráfico, compresión de datos de monitorización, Transformada Wavelet, MRTG.

Resumen

En este proyecto se presenta una implementación mejorada de la herramienta de monitorización de tráfico MRTG - *Multi Router Traffic Grapher*. Esta popular herramienta utiliza un mecanismo de promediado simple para agregar los datos en distintas escalas temporales para no saturar las bases de datos donde almacena la información. Sin embargo, la precisión obtenida al comparar las señales reconstruidas y las originales es generalmente baja por lo que puede perderse información de monitorización valiosa. Por otro lado, las medidas de red presentan un comportamiento claramente diferenciado, siendo el tráfico parecido entre los días laborales, aumentando por las mañanas, decreciendo a la hora de comer y siendo mínimo por las noches. Además, esta periodicidad se refleja también a nivel semanal, mensual e incluso anual de acuerdo con los patrones de tráfico de los usuarios. Por tanto, para aprovechar este carácter periódico se ha decidido utilizar la Transformada Wavelet en sustitución de la función promediadora de MRTG. Concretamente, se han generado tres modelos diferentes con distintas configuraciones en función del uso de los coeficientes de detalle y aproximación de la Transformada Wavelet. Estos modelos se han evaluado mediante un gran conjunto de medidas reales recogidas de RedIRIS, que es la Red Española para la Interconexión de los Recursos InformáticoS de las universidades y centros de investigación. Para cada modelo, se ha analizado la familia *wavelet* que mejores resultados ofrece, el nivel de compresión alcanzado y la mejora de la precisión con respecto a la obtenida por MRTG. Cabe destacar que la precisión se ha medido en base a dos medidas de rendimiento correspondientes al MSE y el MAPE. Los resultados muestran que el tercer modelo es el que ha conseguido un mejor equilibrio entre la información de monitorización perdida y el nivel de compresión utilizado, por lo que es el que finalmente se ha implementado en Python. Con respecto a esta implementación, cabe decir que se ha seguido el mismo formato que MRTG, generando de forma dinámica un informe en HTML con gráficas en PNG que representan la carga de tráfico de los enlaces de red a lo largo del tiempo.

Network bandwidth monitoring using wavelets

Ajendro Acedo Fajardo

Keywords: Network monitoring, network measurements, traffic analysis, monitoring data compression, Wavelet Transform, MRTG.

Abstract

This project carried out an improved implementation of the traffic monitoring tool MRTG — Multi Router Traffic Grapher. This popular tool uses a simple averaging mechanism to compact the data at different time scales to not overflow databases where the information is stored. Nevertheless, the resulting precision is generally low when comparing the reconstructed and the original signals so valuable monitoring information can be lost. Interestingly, we note that network measurements have a clearly periodic behavior, presenting similar traffic between working days, increasing usage at mornings, decreasing at lunch times and dipping to the minimum values at nighttimes. Furthermore, such periodicity is showed across weeks, months and even years according to users' patterns. Therefore, to take advantage of this periodic behavior, we have decided to apply the Wavelet Transform in substitution of the MRTG's averaging function. More specifically, we have developed three models with different settings depending on the use of approximation and details' Wavelet Transform coefficients. These models have been evaluated through a significant set of real measurements collected from RedIRIS, the Spanish National Research Network. In each model, we have analyzed the wavelet families that attain the best results, the level of compression achieved and the accuracy improvement compared with the MRTG counterpart. It is worth remarking that the precision has been obtained through two performance metrics such as MSE and MAPE. The results show that the third model achieves a better balance between the loss of monitoring information and the level of compression, hence it has been translated into a final monitoring tool. With regard to this implementation, carried out in Python, we have followed the same format and operation flow that MRTG follows, i.e., generating dynamically an HTML file containing different PNG figures that represent the traffic burden of the network interfaces over time at different time scales.

Agradecimientos

En primer lugar, quería agradecer a mis padres por todo lo que me han dado e inculcado en la vida. Gracias por haberme hecho llegar hasta aquí y por haberme apoyado en cada momento. También agradecer enormemente a mi hermano Daniel, por haberme ayudado en todo lo que he necesitado. Sin él, este camino hubiera sido mucho más difícil. En general, agradecer enormemente a toda mi familia.

Por supuesto, agradecer a mi gran apoyo durante toda carrera y el máster, la persona que ha compartido conmigo momentos únicos, siendo mi compañera en las buenas y más importante aún, en las malas. Muchas gracias a ti, Sara, por todo esto y mucho más.

Agradecer también al profesor José Luis García Dorado por haberme guiado y ayudado en todo el transcurso de este Trabajo Fin de Máster. De la misma forma, agradecer también a mi ponente Jorge E. López de Vergara Méndez. Ha sido un verdadero placer haber trabajado con grandes profesionales como vosotros.

Por último, acordarme de todos los compañeros y amigos que me llevo del Máster, por los buenos ratos vividos, los momentos de estudio en la biblioteca y un sinfín de anécdotas. Gracias a todos ellos por haber hecho disfrutar tanto esta etapa de mi vida.

Índice general

Agradecimientos	VII
Lista de figuras	X
1. Introducción y objetivos	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Organización de la memoria	3
2. Estado del arte	5
2.1. MRTG- <i>Multi Router Traffic Grapher</i>	5
2.1.1. Funcionamiento de la herramienta MRTG-2	6
2.1.2. MRTG-3 y RRDtool	8
2.2. Transformada Wavelet	10
2.2.1. Origen de la Transformada Wavelet	10
2.2.2. Aplicaciones de la Transformada Wavelet	13
2.2.3. Transformada Wavelet Continua (CWT)	14
2.2.4. Transformada Wavelet Discreta (DWT)	16
2.2.5. Análisis Multi-Resolución	17
2.2.6. Familias de Wavelets	19
2.3. Trabajos previos	21
3. Entorno de trabajo	23
3.1. Escenario de aplicación	23
3.2. Herramientas utilizadas	27
3.3. Medidas de precisión	28
4. Evaluación de la propuesta	31
4.1. Precisión de MRTG	31
4.2. Arquitectura del sistema	35
4.3. Desarrollo y evaluación de la propuesta	37
4.3.1. Primer modelo: Wavelet en desventaja respecto a MRTG	37
4.3.2. Segundo modelo: Wavelet en ventaja respecto a MRTG	44
4.3.3. Tercer modelo: Wavelet y MRTG en igualdad	47
5. Implementación de la herramienta <i>WaveTG</i>	53

ÍNDICE GENERAL

6. Conclusiones y vías futuras	59
6.1. Conclusiones	59
6.2. Vías futuras	61
Referencias bibliográficas	63
A. Tablas de resultados para los tres modelos	65

Índice de figuras

2.1. Ejemplo de gráfica obtenida en MRTG. Tráfico entrante (azul), tráfico saliente (verde).	6
2.2. Interacción de los distintos módulos de MRTG.	7
2.3. Ejemplo de promediado semanal en MRTG.	9
2.4. Transformada de Fourier.	10
2.5. Comparación de la Transformada de Fourier, la STFT y la Transformada Wavelet en el dominio de la frecuencia y del tiempo.	12
2.6. Ejemplo de <i>wavelet</i>	13
2.7. Comportamiento del factor de escala sobre la <i>wavelet</i>	14
2.8. Desplazamiento de la <i>wavelet</i> a lo largo de la señal.	15
2.9. Análisis Multi-Resolución. Diagrama de la Transformada Wavelet Directa.	18
2.10. Análisis Multi-Resolución. Diagrama de la Transformada Wavelet Inversa.	19
2.11. Distintas formas de onda para la familia Daubechies [9].	20
3.1. Arquitectura de RedIRIS.	24
3.2. Ejemplo de patrón de tráfico diario.	25
3.3. Ejemplo de patrón de tráfico mensual.	26
3.4. Función <i>wavedec()</i> de Matlab.	28
4.1. Comparativa entre un fragmento de la señal IXP1 y la reconstrucción con MRTG.	32
4.2. Comparativa entre un fragmento de la señal IXP2 y la reconstrucción con MRTG.	33
4.3. Medidas de precisión obtenidas tras aplicar MRTG a las señales de la Base de Datos.	34
4.4. Diagrama de bloques del sistema.	35
4.5. Primer modelo, reconstrucción con detalles a cero.	37
4.6. Reconstrucción de la señal original según el nivel de descomposición utilizado.	38
4.7. BBDD de MRTG.	39
4.8. BBDD del primer modelo Wavelet.	40
4.9. Medidas de precisión obtenidas tras aplicar MRTG a las señales de la Base de Datos.	41
4.10. Mejora relativa del MSE % para el primer modelo.	42
4.11. Segundo modelo, reconstrucción con coeficientes de detalle.	44
4.12. BBDD del segundo modelo Wavelet.	45

ÍNDICE DE FIGURAS

4.13. . Estudio del mejor tipo de <i>wavelet</i> a nivel semanal, mensual y anual para el segundo modelo.	45
4.14. Mejora relativa del MSE % para el segundo modelo.	46
4.15. BBDD del tercer modelo Wavelet.	47
4.16. Estudio del mejor tipo de <i>wavelet</i> a nivel semanal, mensual y anual para el tercer modelo.	48
4.17. Mejora relativa del MSE % para el tercer modelo.	48
4.18. MAPE obtenido para MRTG y el tercer modelo Wavelet.	50
4.19. Comparativa de MRTG y Wavelet con respecto a la señal original.	50
5.1. Tráfico real entrante y saliente representado cada 5 minutos en <i>WaveTG</i>	54
5.2. Ejemplo de la página web creada por <i>WaveTG</i>	55
5.3. Directorio de ficheros subidos a GitHub.	57

Capítulo 1

Introducción y objetivos

EN este primer capítulo se tratarán aspectos introductorios acerca del presente proyecto. Primeramente, se expondrá la motivación detrás de la realización de este estudio, remarcando el impacto que puede suponer en la actualidad.

Seguidamente, se detallarán los objetivos marcados al comienzo del proyecto. De esta forma, al final de la memoria se podrá realizar una valoración en base a dichos objetivos. Por último, se incluirá un breve resumen sobre la organización de la memoria.

1.1. Motivación

Hoy en día las telecomunicaciones forman un papel imprescindible en la vida de los seres humanos. A diario se utilizan millones de dispositivos electrónicos a nivel mundial con distintos fines. Estos dispositivos se encuentran conectados entre sí por redes de comunicaciones, entre la que destaca principalmente Internet. Estas redes evolucionan a pasos agigantados y se convierten en cada vez más complejas. Por tanto, su gestión y monitorización juega un papel muy importante para lograr conseguir un alto nivel de productividad y eficiencia.

En la actualidad existen numerosas herramientas de monitorización de redes. A través de éstas, los administradores de red son capaces de detectar problemas y tomar aquellas acciones correctivas y/o preventivas para solucionarlos, minimizando así el impacto negativo asociado. Algunos de estos problemas están ligados a caídas de la red, colapsos, cuellos de botella, ataques de denegación de servicio y todo tipo de anomalías. Además, la monitorización está involucrada en tareas relacionadas con la facturación, el aprovisionamiento, dimensionado, e incluso la oferta de servicios [1]. Por ejemplo, los proveedores de servicios de Internet (ISPs – *Internet Service Providers*) utilizan la información de monitorización para asegurar la calidad de servicio (QoS – *Quality of Service*) a sus clientes. Además, el continuo crecimiento de las demandas de los usuarios y la gran variedad requisitos de las aplicaciones utilizadas están forzando a los ISPs a desarrollar planes muy precisos en el dimensionado de la capacidad de las redes, no solo para mantener los niveles de QoS, sino también para reducir la inversión económica [2]. Lo que es evidentes es que

la monitorización de redes implica la transferencia y el almacenamiento de una enorme cantidad de datos. Surge así la necesidad de aplicar técnicas de reducción a estos datos para lograr una cantidad relativamente manejable.

Una de las herramientas de monitorización más utilizadas por los proveedores de servicios a lo largo de las últimas décadas ha sido MRTG (*Multi Router Traffic Grapher*) [3]. MRTG es la herramienta más popular para visualizar la carga de una red, típicamente el ancho de banda a lo largo del tiempo. Por defecto, esta herramienta representa una medida cada 5 minutos aunque existen varias granularidades temporales, que van desde el día hasta varios años, pasando por semanas y meses. Los datos más recientes se almacenan a escala temporal original, mientras que los más antiguos se agregan en escalas de tiempo más grandes. Para no desbordar las bases de datos donde se almacena la información, MRTG agrega los datos utilizando un proceso promediador simple. Sin embargo, este proceso es arbitrario y eventualmente puede provocar pérdidas valiosas de información de monitorización.

Por otro lado, las medidas de red presentan un comportamiento periódico claramente diferenciado. Por ejemplo, el tráfico es parecido entre los días laborales, aumenta por las mañanas, decae a la hora de comer y es mínimo por las noches [4]. En definitiva, las demandas tienden a seguir patrones horarios, diarios, mensuales e incluso anuales de acuerdo a los patrones de los usuarios. Por ello, estas demandas tienen una componente en frecuencia muy significativa.

De este modo, planteamos la idea de aplicar alguna herramienta matemática en MRTG que aproveche la componente frecuencial presente en las medidas de red y que minimice las pérdidas de información. Más en concreto, estudiamos la utilización de la Transformada Wavelet como sustitución de la función promedidora simple de MRTG.

1.2. Objetivos

Una vez analizados los factores que motivan la realización de este estudio, a continuación se detallarán los objetivos iniciales del mismo. En primer lugar, cabe destacar que el objetivo principal de este proyecto es la implementación de una versión mejorada de la herramienta de monitorización *Multi Router Traffic Grapher* (MRTG). Se pretende sustituir el proceso promediador simple que esta herramienta utiliza para agregar los datos, por un proceso en el que se aplique la Transformada Wavelet, convenientemente parametrizada, y que aproveche el carácter periódico que presentan las medidas de red. En definitiva, la finalidad es la de reducir la enorme cantidad de datos de entrada, perdiendo la mínima información posible y mejorando la precisión de MRTG.

Concretamente, se pretende desarrollar una aplicación en Python que simule el comportamiento de MRTG, leyendo los datos de entrada y generando informes en formato HTML con gráficas que representen el ancho de banda, tanto de entrada como de salida, de cualquier interfaz de red a lo largo del tiempo. Para lograr alcanzar este objetivo principal, se plantean una serie de objetivos específicos:

- Análisis previo de las funcionalidades y características de la herramienta MRTG y estudio en profundidad de la transformada Wavelet, prestando especial interés en su desarrollo matemático, su utilidad y sus aplicaciones en el ámbito del procesado de señal.
- Análisis y pre-procesado de las señales de entrada, correspondientes a un conjunto de series temporales que representan el ancho de banda de distintas interfaces de RedIRIS (Red Española para la Interconexión de los Recursos InformáticoS de las universidades y centros de investigación).
- Desarrollo de scripts en Matlab aplicando la Transformada Wavelet a las distintas señales de entrada. Obtención de varios modelos en base al uso de los coeficientes de aproximación y detalle de dicha transformada. Evaluación de los modelos desarrollados utilizando medidas de precisión capaces de comparar el grado de similitud entre las señales reconstruidas y las originales.
- Diseño e implementación de la aplicación final en el lenguaje de programación Python, de forma que sea lo más parecida posible a la herramienta MRTG. Descripción de sus características, obtención de estadísticas y representación gráfica de los resultados finales.

1.3. Organización de la memoria

La presente memoria está formada por 6 capítulos:

Capítulo 1. Introducción y objetivos. En este capítulo se ha definido la motivación del estudio, los objetivos marcados y la organización de la memoria.

Capítulo 2. Estado del arte. Se presentan los principales fundamentos teóricos relacionados con la herramienta de monitorización de tráfico MRTG y con aquellos relativos a la Transformada Wavelet. En ambos casos, el estudio se centra principalmente en su origen, evolución, utilidades, funcionamiento y aplicaciones.

Capítulo 3. Entorno de trabajo. Se describe el escenario de aplicación del proyecto, analizando detenidamente las señales disponibles. Además, se describen las herramientas usadas para alcanzar los objetivos del proyecto y se estudian las medidas de precisión utilizadas para evaluar la propuesta desarrollada.

Capítulo 4. Evaluación de la propuesta. En este capítulo se analiza la propuesta desarrollada utilizando la herramienta Matlab. En primer lugar, se simula el comportamiento de MRTG para obtener una base en la que poder comparar los resultados obtenidos tras posteriormente utilizar la Transformada Wavelet. También se describe la arquitectura del sistema desarrollado mediante un diagrama de bloques y se evalúa la precisión de tres modelos diferentes, de cara a obtener los parámetros de configuración apropiados para el modelo final. Para ello, se tiene en cuenta el equilibrio existente entre la compresión de los datos y la pérdida de información asociada.

Capítulo 5. Implementación de la herramienta WaveTG en Python. Se describe el proceso seguido en la implementación del modelo definitivo usando el lenguaje de programación Python y se muestran los resultados finales.

Capítulo 6. Conclusiones y vías futuras. Se extraen conclusiones del proyecto y se plantean algunas vías futuras de investigación.

Apéndices.

Capítulo 2

Estado del arte

EN este capítulo se expondrán los principales fundamentos teóricos que serán necesarios para comprender el desarrollo del proyecto. En primer lugar, se analizará en detalle la herramienta de monitorización Multi Router Traffic Grapher - MRTG, explicando las bases de su funcionamiento y su evolución temporal.

Por otro lado, se estudiarán los principios teóricos de la Transformada Wavelet, haciendo un repaso a su origen, los fundamentos matemáticos de la Transformada Continua y Discreta, y el Análisis Multi-Resolución. También se comentarán sus aplicaciones más importantes y las familias wavelets más conocidas.

Por último, se recogerán los trabajos previos más relevantes relacionados con la aplicación de las wavelets a señales de monitorización de tráfico, analizando sus ventajas y resultados obtenidos.

2.1. MRTG- *Multi Router Traffic Grapher*

MRTG es una herramienta desarrollada por Tobias Oetiker y Dave Rand, utilizada para monitorizar la carga de tráfico de interfaces de red. Para ello, MRTG hace uso de la información procedente del protocolo SNMP (*Simple Network Management Protocol*) relativa a la carga de tráfico de los distintos dispositivos de red. En concreto, calcula el tráfico generado en un intervalo de tiempo como la resta de peticiones SNMP consecutivas, que preguntan a los *routers* cuanto tráfico en total han reenviado desde que están en funcionamiento. Dicha información es procesada adecuadamente para la generación de informes en lenguaje HTML (*HyperText Markup Language*) que contienen gráficas dinámicas en formato PNG (*Portable Network Graphics*), representando visualmente la evolución del tráfico a lo largo del tiempo a escala diaria, semanal, mensual y anual. En la Figura 2.1, se puede observar un ejemplo de representación, concretamente correspondiente al ancho de banda entrante y saliente (medido en bits/segundo) de una interfaz de red a lo largo de tiempo. Cabe destacar que este software está desarrollado en lenguaje de programación Perl y es público bajo la licencia GNU GPL (*General Public License*).

Además, MRTG puede utilizarse también para monitorizar cualquier variable medible (carga de la CPU, disponibilidad de disco, temperatura, etc.) y no solamente funciona en dispositivos que disponen de un agente SNMP, sino que permite ejecutar cualquier tipo de aplicación mediante scripts de usuario.

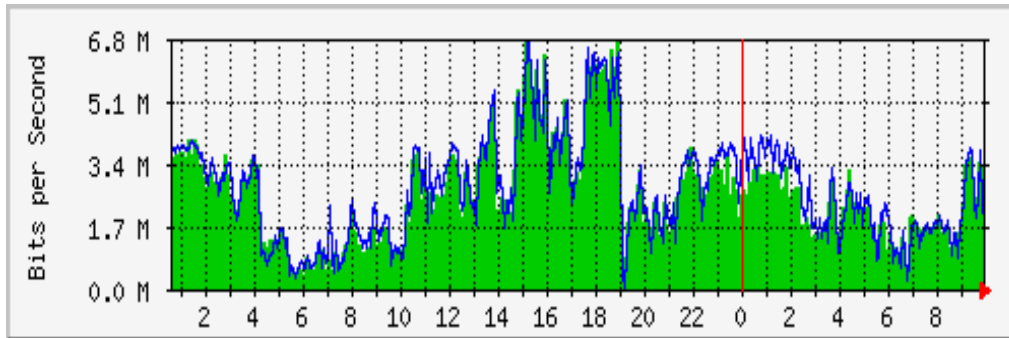


Figura 2.1: Ejemplo de gráfica obtenida en MRTG. Tráfico entrante (azul), tráfico saliente (verde).

Inicialmente, MRTG funcionaba correctamente pues el número de interfaces monitorizadas era pequeño y la herramienta no tenía problemas en la creación de las gráficas. Sin embargo, a medida que fue pasando el tiempo, surgieron problemas de escalabilidad y portabilidad [3]. Fue entonces cuando Dave Rand contribuyó con un pequeño programa escrito en C, que resolvía los problemas de rendimiento de esta herramienta, implementando aquellas tareas que más carga computacional requerían y eliminándolas del script en Perl. De esta forma, nació la herramienta MRTG-2, que no solamente era más rápida que la original sino que también era más fácil de usar de cara al usuario. En el siguiente punto se explicará en detalle el funcionamiento de esta herramienta.

2.1.1. Funcionamiento de la herramienta MRTG-2

La primera novedad de MRTG-2 con respecto a la versión original es la posibilidad de auto-configurar inicialmente la herramienta de una forma rápida y sencilla. MRTG-2 hace uso de un fichero de configuración, normalmente con extensión *.cfg*, para saber qué variables de la MIB (*Management Information Base*) debe representar. Para facilitar la elaboración de este fichero, se creó el módulo *cfgmaker*, que explora el dispositivo a monitorizar haciendo uso del protocolo SNMP para saber qué interfaces tiene y cuáles de ellas están operativas. A partir de esta información, se genera automáticamente el fichero de configuración. Esto permite al usuario utilizar la herramienta sin tener que dominar el protocolo SNMP.

Una vez se tiene preparado el fichero de configuración, MRTG genera la información correspondiente a las gráficas en formato PNG y crea las páginas HTML que las contienen, estructurándola en varios ficheros. Por otro lado, dado que el número de interfaces de los distintos dispositivos monitorizados puede ser elevado, es recomendable la creación de una

página índice que recoja el acceso a las páginas webs de las distintas interfaces. Al igual que *cfgmaker*, existe otra utilidad denominada *indexmaker* que permite realizar esta operación. Para ello, *indexmaker* hace uso de la información presente en el fichero de configuración (*.cfg*). En la Figura 2.2, se muestra la interacción entre estas tres herramientas.

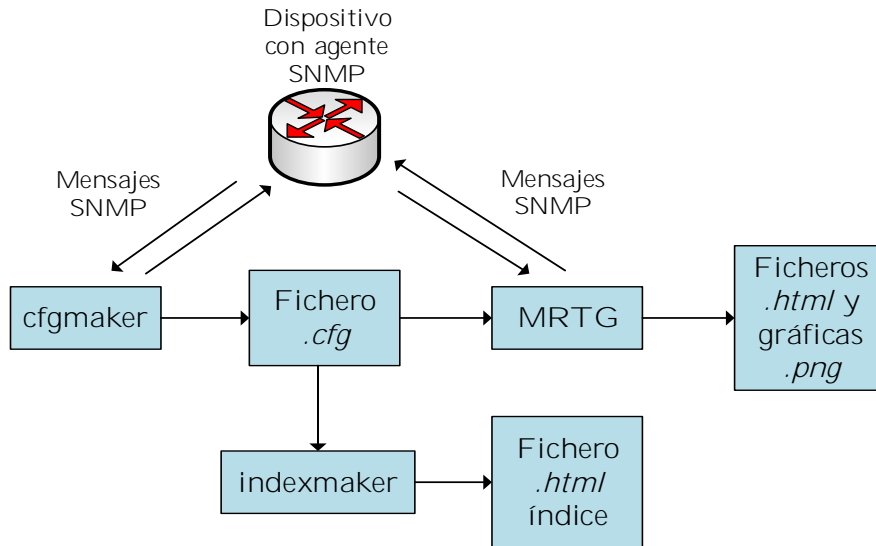


Figura 2.2: Interacción de los distintos módulos de MRTG.

En cuanto al almacenamiento de los datos, cabe decir que MRTG-2 presenta distintas resoluciones temporales. En los ficheros *.log* se almacena la información con una resolución decreciente a medida que los datos son más antiguos. En la Tabla 2.1, se muestran dichas resoluciones para niveles de agregación diarios, semanales, mensuales y anuales. Los datos que superan los dos años de antigüedad se eliminan, por lo que se puede utilizar la herramienta indefinidamente sin preocupaciones en cuanto a las limitaciones de memoria.

Intervalo temporal	Número de muestras	Resolución temporal (min)
Diario	1	5
Semanal	6	30
Mensual	24	120
Anual	288	1440

Tabla 2.1: Resolución temporal usada por MRTG-2 para distintas agregaciones.

Dada la mejora en cuanto al rendimiento de MRTG-2, esta herramienta se fue utilizando cada vez más, llegando de nuevo rápidamente al límite de su rendimiento. En este caso, este límite se situaba aproximadamente en la monitorización cada 5 minutos del ancho de banda de unas 500 interfaces de red. Además, el hecho de que se utilizase este software para tareas las cuales no fue diseñado, mediante la modificación de la configuración de

forma personalizada, provocó que se replanteara la separación en distintas herramientas según su cometido:

- Recolección de datos
- Almacenamiento
- Consolidación
- Visualización

Para ello, surgió la herramienta MRTG-3 cuyos objetivos principales son la mejora de la flexibilidad y la rapidez. A continuación, en el punto 2.1.2 se analiza esta nueva versión, describiendo sus características novedosas, entre las que se incluye el nacimiento de un nuevo método de almacenamiento de datos denominado Round Robin Database Tool (RRDtool).

2.1.2. MRTG-3 y RRDtool

La nueva versión MRTG-3 se aleja de ser una simple herramienta de monitorización de tráfico, convirtiéndose en un grupo de herramientas que monitorizan un diverso conjunto de series temporales. Tiene la capacidad de crear una amplia variedad de gráficas, usando datos recogidos de una o varias fuentes.

En primer lugar, MRTG-3 empezó con la implementación de un nuevo mecanismo de almacenamiento de datos más eficiente denominado *Round Robin Database* (RRD), implementado en un programa escrito en lenguaje C y denominado *rrdtool*. Este programa puede ser utilizado a través de la línea de comandos o bien mediante *bindings* en Perl y presenta las siguientes mejoras respecto al fichero *.log* utilizado en MRTG-2:

- El formato RRD utiliza datos de tipo doubles en lugar de enteros, eliminando los problemas de *overflow* ocurridos al monitorizar interfaces de *routers* rápidos.
- Se crea la etiqueta *unknown*, distinguiendo así las situaciones en los que los datos de entrada son cero, de las que no se ha podido obtener datos válidos.
- Los parámetros asociados al número de entradas en el fichero *.log*, la resolución y el número de fuentes de datos en paralelo son configurables. En cuanto a las fuentes de datos, pueden ser de diversa naturaleza, desde un contador de bytes hasta la temperatura de un sensor.
- Los datos se almacenan en formato binario, haciendo su acceso más eficiente pues no se requiere de ninguna conversión.

Precisamente, en cuanto al almacenamiento, cabe decir que se sigue el método *Round Robin*, guardando cientos de datos por segundo en áreas pre-asignadas denominadas

Round Robin Archives (RRAs). A su vez, cada RRA tiene una configuración única en cuanto a la resolución temporal, tamaño y método de consolidación. Cuando llegan nuevas muestras de la fuente de datos a la RRD (*Round Robin Database*), se consolidan en varios ficheros RRAs de acuerdo a sus definiciones. Al añadirse una entrada RRA, la más antigua se descarta, manteniendo de esta forma el fichero con el mismo tamaño y refrescando los datos.

Un archivo RRD generado por MRTG tiene por defecto 8 RRAs, correspondientes a datos diarios, semanales, mensuales y anuales, tanto en valores máximos como en promedio. Por ejemplo, un fichero RRA semanal presenta un paso de 6 muestras. Si se utilizan intervalos de 5 minutos, cada valor de un RRA semanal tiene una ventana temporal de media hora. Además, si se tiene configurado como factor de consolidación el valor medio, lo que realmente se almacena es el promedio de esas 6 muestras. En la Figura 2.3, se muestra este ejemplo de forma gráfica, para $n = 24$ muestras. Este proceso es similar para los intervalos mensuales y anuales.

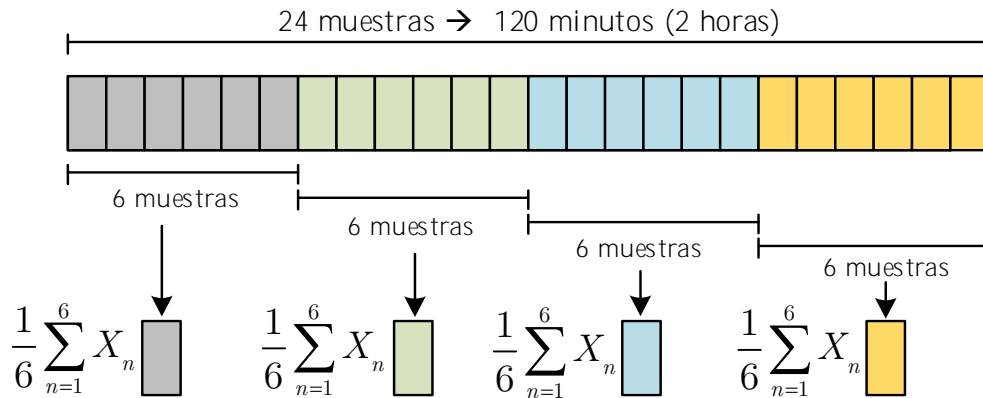


Figura 2.3: Ejemplo de promediado semanal en MRTG.

Al utilizarse este método, se crea la posibilidad de que otras herramientas, como por ejemplo Routers2, puedan acceder a la base de datos y representar la información con distintos estilos, liberando así las tareas de monitorización. Por otro lado, el módulo de almacenamiento de los datos también se encarga de convertir las medidas de red en *rates*. Por defecto, se utiliza el método *Counter*, calculando el *rate* como la diferencia entre un valor y el anterior dividido por el número de segundos transcurridos entre dichas muestras. Para datos en los que se necesita almacenar el valor tal cual se recibió, como por ejemplo puede ser el espacio de disco utilizado, se emplea otro método denominado Gauge.

Por último, cabe decir que la representación de las gráficas es tan flexible como el formato RD. De esta forma se pueden generar gráficas de cualquier tamaño e intervalo temporal y representar los datos procedentes de un conjunto de fuentes almacenadas en distintos RRDs. Además, el módulo gráfico también calcula el valor máximo, mínimo y promedio de los datos representados, incluyéndolos en el fichero HTML. Esto permite a

los gestores de red obtener aún más información en un simple vistazo. Con respecto a la generación de las páginas webs, cabe decir que MRTG-3 trabaja con plantillas haciendo su diseño más simple y flexible.

2.2. Transformada Wavelet

Dado que la Transformada Wavelet es la técnica que se implementará para lograr alcanzar el objetivo de este proyecto, a continuación se expondrá un estudio en detalle de esta transformada. Se analizará su origen y sus aplicaciones más comunes, a la vez que se repasarán las bases de su funcionamiento, centrando el estudio en el análisis Multi-Resolución pues será el que se utilizará en el desarrollo práctico.

2.2.1. Origen de la Transformada Wavelet

En la naturaleza, existen multitud de fenómenos físicos que pueden ser descritos a través de una señal en el dominio del tiempo, en el que la variable dependiente corresponde a la amplitud de dicha señal y la variable independiente corresponde al tiempo. Normalmente, estas señales presentan oscilaciones que varían lentamente, con cambios abruptos puntuales. Sin embargo, no siempre la escala temporal es relevante. En numerosos casos, lo que realmente caracteriza a las señales, y las diferencia del resto, es su comportamiento frecuencial. Por este motivo, surgió la necesidad de desarrollar técnicas de procesamiento de señal, basadas en dominio de la frecuencia y en las distribuciones Tiempo-Frecuencia.

En primer lugar, en el dominio de la frecuencia destaca principalmente la Transformada de Fourier dada su gran utilidad para el procesamiento de señales un con comportamiento frecuencial. Se caracteriza principalmente por su capacidad de representar una señal como suma de ondas sinusoidales, que oscilan de forma infinita y que no están localizadas en el tiempo o en el espacio, tal y como se puede ver en la Figura 2.4.

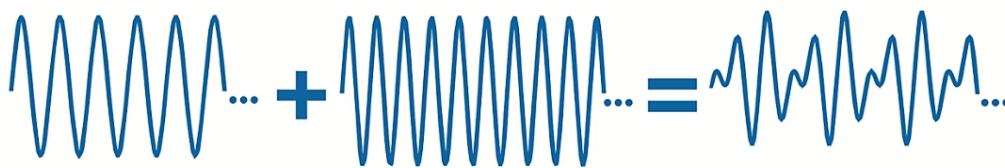


Figura 2.4: Transformada de Fourier.

Sin embargo, si se representa esta transformada sobre el eje de tiempo, no se puede indicar el instante preciso en el que tiene lugar un evento a una frecuencia concreta, de decir si se trabaja en el dominio del tiempo, es imposible obtener información acerca de la frecuencia. Lo mismo ocurre al contrario, en el dominio de la frecuencia no hay información del tiempo. Además, la Transformada de Fourier está diseñada para aplicarse a señales

estacionarias, finitas y periódicas, por lo que existe una gran multitud de señales que quedan excluidas.

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (2.1)$$

Por tanto, surgió la Transformada de Fourier de Tiempo Corto (STFT) cuya expresión queda representada en la ecuación 2.2. Consiste en utilizar una ventana deslizante $g(t)$ que va dividiendo la señal en fragmentos temporales pequeños, en los que pueda asumirse estacionariedad. Una vez se tiene separada la señal, se aplica la Transformada de Fourier a cada una de las tramas.

$$S(\omega, \tau) = \int_{-\infty}^{\infty} f(t)g^*(t - \tau)e^{-i\omega t} dt \quad (2.2)$$

$$g(t) = \frac{1}{s} e^{-\frac{\pi t^2}{s^2}} \quad (2.3)$$

La principal característica de la STFT es su capacidad de obtener una representación Tiempo-Frecuencia de la señal, lo que permite conocer el valor de las componentes armónicas, así como el instante temporal en el que se ubican. Sin embargo, existe una gran limitación y es el hecho de que únicamente se puede conocer el intervalo temporal en el que ocurre un cierto evento, dependiendo este intervalo de forma directa del tamaño de la ventana [5]. De esta forma, las ventanas temporales estrechas permiten una gran resolución en el dominio del tiempo, pero peor en el dominio de la frecuencia y viceversa. Además, la STFT no presenta un método numérico para la reconstrucción de la señal original.

Por tanto, surgió la necesidad de seguir investigando en cuanto al desarrollo de métodos matemáticos basados en el estudio de las distribuciones Tiempo-Frecuencia. Como consecuencia, a mediados de los años '80 surgió la Transformada Wavelet, la cual se caracteriza debido a que mantiene el enventanado de la STFT pero presenta la posibilidad de que las ventanas tomen anchuras variables. Con esto se consiguen diferentes resoluciones para distintas frecuencias [6].

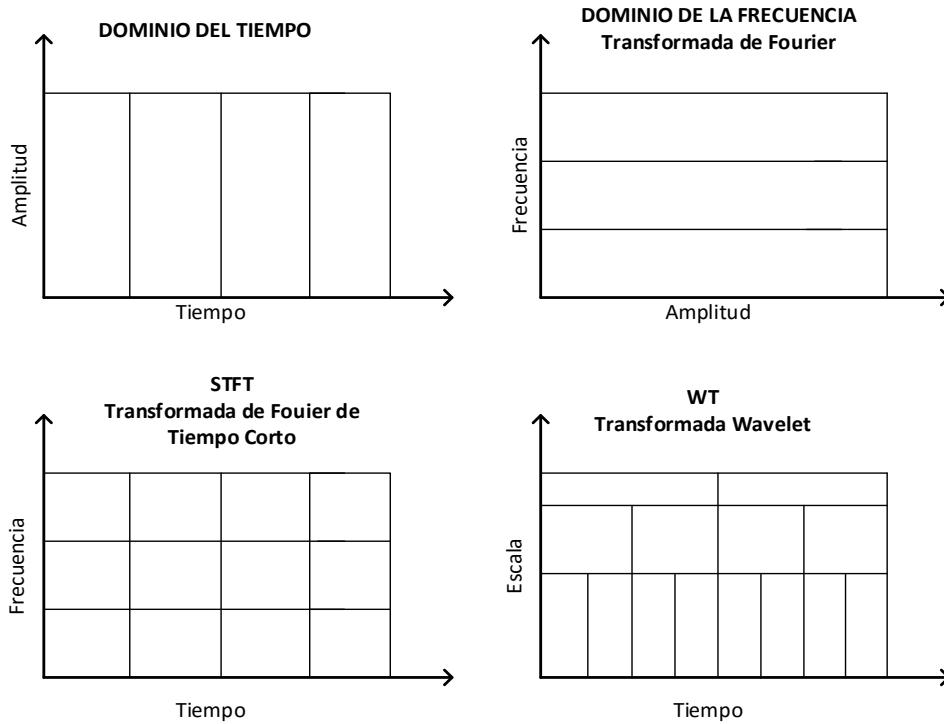


Figura 2.5: Comparación de la Transformada de Fourier, la STFT y la Transformada Wavelet en el dominio de la frecuencia y del tiempo.

Cabe mencionar, que esta Transformada se basa en el uso de wavelets, que no son más que ondas de duración finita y media cero, que están bien localizadas en ambos dominios, tiempo y frecuencia. La principal diferencia con respecto a las sinusoides de la Transformada de Fourier, es que las wavelets son irregulares y asimétricas. Además presentan distintas formas y tamaños.

Por otro lado, en cuanto su funcionamiento, a diferencia de la Transformada de Fourier que se encarga de descomponer una señal como suma de ondas sinusoidales de distinta frecuencia, la Transformada Wavelet consiste en descomponer una señal en diversas versiones escaladas (estiradas o encogidas) y desplazadas (adelantadas o atrasadas) de una *wavelet* denominada *wavelet madre*. Por tanto, una ventaja importante de esta técnica es la eficiencia que presenta en cuanto al análisis de señales de rápida transitoriedad.

Las diferencias de la Transformada Wavelet con respecto a la Transformada de Fourier y la SFTF se pueden ver en la Figura 2.5, donde se ha realizado una representación del comportamiento de cada una de estas herramientas tanto en el dominio del tiempo como el de la frecuencia. Además, en la Figura 2.6 se ha recogido un ejemplo de la forma de onda de una *wavelet*, donde se puede apreciar a simple vista que realmente presenta media cero:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (2.4)$$

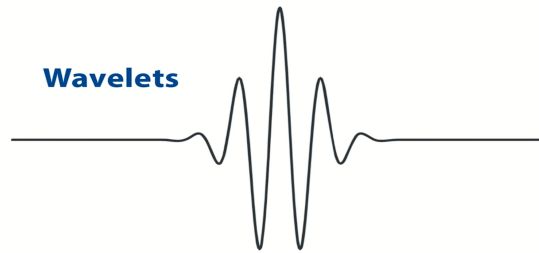


Figura 2.6: Ejemplo de *wavelet*.

2.2.2. Aplicaciones de la Transformada Wavelet

La Transformada Wavelet puede ser utilizada en una gran variedad de señales, ayudando a la resolución de problemas de diversa naturaleza. En primer lugar, un campo en el que cobra especial importancia es el de tratamiento de imágenes sobre todo en el ámbito de la biomedicina donde, por ejemplo, se ha utilizado para determinar patrones o irregularidades en electrocardiogramas y electroencefalogramas gracias a la descomposición Tiempo-Frecuencia. La clave en este ámbito es conservar la calidad del diagnóstico de los pacientes de las series temporales originales, pero reduciendo el almacenamiento necesario para no saturar los sistemas en los hospitales.

Además, esta Transformada permite también la compresión de imágenes y el reconocimiento de patrones, lo que facilita el almacenamiento y tratamiento de grandes cantidades de datos. Una de sus aplicaciones más directas es en el estándar de compresión y codificación digital JPEG2000 (*.jp2*).

Por otro lado, la Transformada Wavelet también se utiliza en ramas como la física (astrofísica, óptica, mecánica cuántica, etc.) y permite tratar, procesar y evaluar de forma eficiente y robusta una gran variedad de señales, como por ejemplo:

- Señales de voz
- Señales radar
- Señales sísmicas
- Señales biológicas

Además, debido a sus múltiples ventajas, esta herramienta matemática está siendo utilizada en series temporales de diversos tipos. Concretamente, en este proyecto se ha utilizado la Transformada Wavelet con señales correspondientes a la carga de tráfico medido en distintas interfaces de red a lo largo del tiempo.

2.2.3. Transformada Wavelet Continua (CWT)

Como ya se ha comentado, la Transformada Wavelet de una señal continua $f(t)$ consiste en la descomposición de una señal en un conjunto de funciones que forman una base y que se denominan wavelets. Como se puede ver en la ecuación 2.5, la señal se representa como una expansión de coeficientes fruto del producto entre ella misma y la *wavelet* madre $\psi(t)$ seleccionada, en su versión escalada o desplazada. Sumando todas las funciones obtenidas para cada instante de tiempo, se llega a la definición de Transformada Wavelet Continua (CWT) [7].

$$C(\tau, s) = \int_{-\infty}^{\infty} f(t)\psi_{\tau,s}(t) \quad (2.5)$$

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{|s|}}\psi\left(\frac{t-\tau}{s}\right) \quad (2.6)$$

Tal y como se puede observar de las expresiones anteriores, la transformada depende de dos variables τ y s , que corresponden respectivamente a los parámetros de traslación y escala.

- El parámetro de escala, se refiere al proceso de estiramiento (*stretching*) o reducción (*shrinking*) de la señal en el dominio del tiempo. Se caracteriza por ser siempre positivo e indicar cuánto se escala la señal en el dominio del tiempo. Como se puede ver en la Figura 2.7, valores superiores a 1 producen una dilatación de la *wavelet*. Por otro lado, si se utilizan valores entre 0 y 1, se produce una compresión que se refleja en una visión detallada de la señal en el dominio del tiempo.

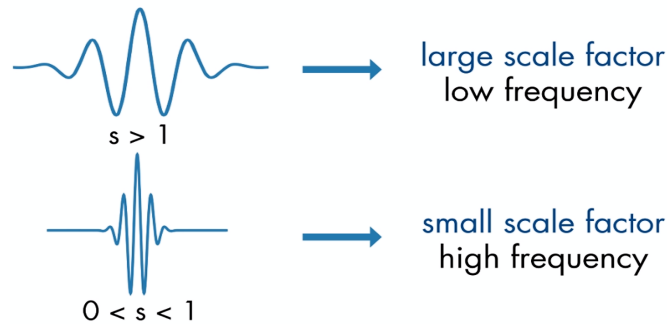


Figura 2.7: Comportamiento del factor de escala sobre la *wavelet*.

En cuanto al comportamiento en frecuencia de este parámetro, cabe decir, que el factor de escala es inversamente proporcional a la frecuencia $s = 1/f$. Por ejemplo, si se escala por 2, la señal se estira ocupando el doble de tiempo y su frecuencia se reduce a la mitad. Al contrario ocurre si se utiliza un valor $s = 0.5$, la señal se reduce a la mitad en el dominio del tiempo y su frecuencia aumenta al doble. De esta forma,

una *wavelet* más estirada, ayuda a capturar los cambios que varían lentamente en una señal, mientras que una *wavelet* más comprimida ayuda a capturar aquellos cambios más abruptos.

- Por otro lado, el término correspondiente a la traslación se relaciona con el desplazamiento de la ventana a través de la señal. Este parámetro se encarga de retrasar o adelantar el inicio de la onda a lo largo de la longitud de la señal. De esta forma, en la Figura 2.8, se puede observar que la *wavelet* centrada en k es desplazada a lo largo del tiempo para alinearla con el rasgo buscado de la señal.

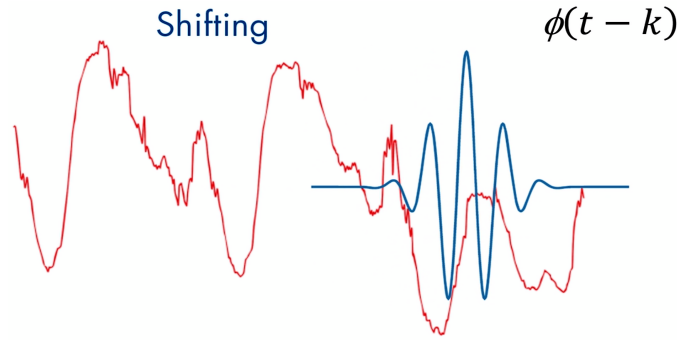


Figura 2.8: Desplazamiento de la *wavelet* a lo largo de la señal.

Una vez analizados los parámetros de la CWT a continuación se va a explicar el proceso seguido para su cálculo. En primer lugar, se debe seleccionar la *wavelet* madre que desee utilizar. Seguidamente, dados dos valores de τ y s y un tramo de señal, se calcula un coeficiente $C(\tau, s)$ que representa la correlación entre la *wavelet* y dicho tramo. Cabe decir, que a mayor valor del coeficiente, mayor es la similitud, lo que resalta la importancia en la elección del tipo de *wavelet*. Este proceso se realiza de forma iterativa, cambiando el valor de τ , es decir, desplazando la *wavelet* a través del eje temporal. Una vez cubierta toda la señal, se obtiene el conjunto de coeficientes que la representan para dicho factor de escala. Sin embargo, como se pretende conseguir resolución en frecuencia, a continuación se escala la *wavelet* y se vuelve a realizar todo el proceso. De este modo, se van obteniendo un conjunto de coeficientes que representan a la señal en los dominios del tiempo y de la frecuencia.

Finalmente, cabe decir que una de las ventajas más importantes de esta transformada con respecto a la Transformada de Fourier de Tiempo Corto es la existencia de una Transformada Wavelet Continua inversa (ICWT). Esta expresión matemática que se recoge a continuación, es la que se utiliza para reconstruir la señal original.

$$f(t) = \frac{1}{K_\psi} \int \int C(\tau, s) \frac{\psi(\tau, s)}{\tau^2} d\tau ds \quad (2.7)$$

donde

$$K_\psi = \int_{-\infty}^{\infty} \frac{|\psi(\omega)|^2}{\omega} d\omega < \infty \quad (2.8)$$

2.2.4. Transformada Wavelet Discreta (DWT)

La Transformada Wavelet Discreta (DWT) surge como adaptación a las señales digitales, presentes hoy en día en multitud de campos. La DWT es una herramienta idónea para la eliminación de ruido y compresión de señales e imágenes, ya que ayuda a representar un gran conjunto de datos con un menor número de coeficientes. Esta transformada se define en las siguientes expresiones:

$$\psi_{\tau,s}[k] = \frac{1}{\sqrt{|s|}} \psi\left(\frac{k-\tau}{s}\right) \quad (2.9)$$

$$c^{m,n}[k] = \sum_k f[k] \psi_{m,n}[k] \quad (2.10)$$

Siendo $s = 2^m$ y $\tau = n2^m$ por lo que las expresiones finales quedan se la siguiente forma:

$$\psi_{m,n}[k] = \frac{1}{\sqrt{2^m}} \psi\left(\frac{k-n2^m}{2^m}\right) \quad (2.11)$$

$$c^{m,n}[k] = \frac{1}{\sqrt{2^m}} \sum_k f[k] \psi\left(\frac{k-n2^m}{2^m}\right) \quad (2.12)$$

La DWT sigue una escala diádica con lo que consigue una mejor eficiencia, eliminando la redundancia entre los valores de los coeficientes, lo que supone un requerimiento menor de memoria. En cuanto a la definición de la Transformada Wavelet Discreta Inversa, su expresión es la siguiente

$$f[k] = \sum_j \sum_n C[j,n] \psi_{j,n}[k] \quad (2.13)$$

Cabe mencionar que la principal diferencia entre la Transformada Wavelet Discreta y la Transformada Wavelet Continua es la posibilidad de construir las *wavelets* mediante un Análisis Multi-Resolución. Esta idea se fundamenta en que la Transformada Wavelet se basa en un conjunto de funciones base que se generan a partir de una función *wavelet* madre. Estas funciones base, que se denominan Bases de Escala y Bases Wavelets, proporcionan los llamados coeficientes de aproximación y coeficientes de detalle, respectivamente. Si dichas bases son ortogonales, los coeficientes de detalle y de aproximación se pueden obtener mediante un Análisis Multi-Resolución, que se explica en detalle en la siguiente sección.

La expresión de los coeficientes de aproximación es la siguiente:

$$W_\phi[j_0, k] = \frac{1}{\sqrt{M}} \sum_n f[n] \phi_{j_0, k}[n] \quad (2.14)$$

La expresión de los coeficientes de detalle es la siguiente:

$$W_\psi[j, k] = \frac{1}{\sqrt{M}} \sum_n f[n] \psi_{j, k}[n] \quad j \geq j_0 \quad (2.15)$$

La siguiente expresión muestra cómo se expresa una señal en función de dichos coeficientes.

$$f[n] = \frac{1}{\sqrt{M}} \sum_k W_\phi[j_0, k] \phi_{j_0, k}[n] + \frac{1}{\sqrt{M}} \sum_k^\infty W_\psi[j, k] \psi_{j, k}[n] \quad (2.16)$$

2.2.5. Análisis Multi-Resolución

El Análisis Multi-Resolución es una técnica desarrollada en 1988 por Stephane Mallat que permite analizar una señal en múltiples bandas de frecuencia. Su objetivo es el de obtener una representación de la señal en función de los coeficientes de aproximación y de detalle [8]. En definitiva, consiste en una buena forma de implementar la Transformada Wavelet Discreta en cuanto a la eficiencia computacional se refiere.

El Análisis Multi-Resolución, como su propio nombre indica, pretende representar la señal de interés en diferentes resoluciones, desde la más baja a la más alta. Para ello, se hace uso de distintas frecuencias de corte mediante el siguiente procedimiento.

La señal de entrada $X[n]$ se pasa inicialmente a través de un filtro paso alto y un filtro paso bajo y se sub-muestra con factor 2 (quedándose con una de cada dos muestras). Como resultado, tras pasar la señal por el filtro paso alto se obtienen los coeficientes de detalle del primer nivel cD1. Por otro lado, los coeficientes de aproximación del primer nivel cA1 se obtienen tras pasar la señal por el filtro paso bajo. De este modo, se tiene un conjunto de coeficientes que representan a la señal en un rango de frecuencias determinado y a lo largo del tiempo. Si lo que se quiere es aumentar la resolución en frecuencia, se debe seguir descomponiendo la señal en un segundo nivel. En este caso, la sub-banda A1 se vuelve a filtrar paso alto y paso bajo para obtener los coeficientes de detalle y aproximación del segundo nivel (cD2 y cA2). Cabe destacar que a medida que se aumenta el número de niveles, aumenta la resolución en frecuencia pero disminuye la resolución en el tiempo a causa del sub-muestreo realizado en cada nivel. De esta forma, en cada nivel los coeficientes presentan la mitad de muestras que aquellos del nivel anterior. Este algoritmo se realiza de forma iterativa en función del número de niveles en los que se quiere descomponer la señal. En la Figura 2.9 se muestra el diagrama del Análisis Multi-Resolución de la Transformada Wavelet directa.

Cabe destacar que, tras realizar esta descomposición, será en los niveles más bajos donde se encuentren las frecuencias más altas y la mayor resolución en tiempo, y en los niveles más altos donde se encuentren las frecuencias más bajas con menor resolución temporal. Por tanto, la señal resultante en los niveles más altos representará la forma aproximada de la señal original, sin detalles asociados.

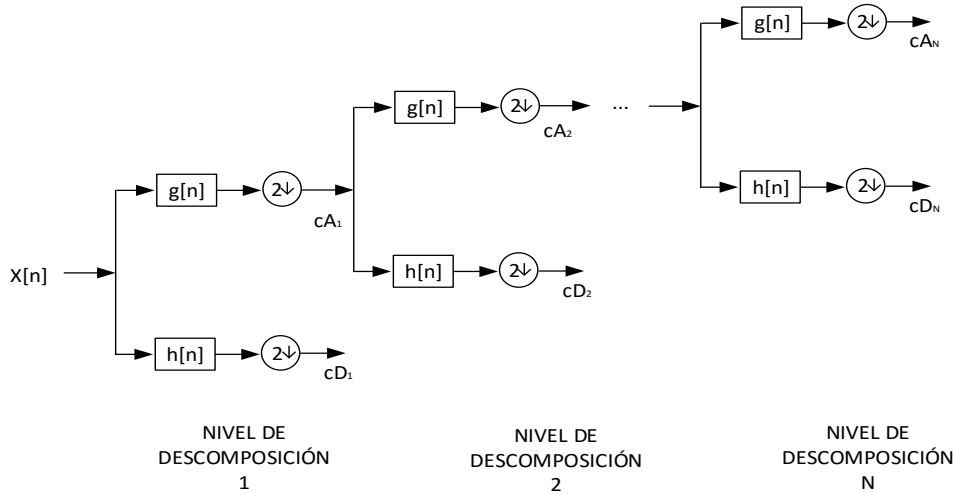


Figura 2.9: Análisis Multi-Resolución. Diagrama de la Transformada Wavelet Directa.

A continuación, una vez se han obtenido los coeficientes de detalle y aproximación, el siguiente paso será el de almacenarlos para posteriormente reconstruir la señal a partir de ellos. Para dicha reconstrucción, es necesario unir los coeficientes de forma adecuada de la siguiente forma (suponiendo N niveles):

$$C = cA_N + cD_N + \dots + cD_2 + cD_1 \quad (2.17)$$

Por ejemplo, si la señal de entrada tiene 8064 muestras y utilizamos N=4 niveles:

$$C = cA_4 + cD_4 + cD_3 + cD_2 + cD_1 \quad (2.18)$$

$$length(C) = 504 + 504 + 1008 + 2016 + 4032 \quad \text{valores} \quad (2.19)$$

En cuanto al proceso de la Transformada Wavelet Inversa, el procedimiento es el siguiente. Primeramente se realiza una interpolación de los coeficientes de detalle y aproximación del último nivel N, seguido de un filtrado paso alto y paso bajo. De esta forma, se obtienen los coeficientes de aproximación del nivel anterior N-1. Repitiendo este proceso, tal como se puede ver en la Figura 2.10, se consigue obtener la señal original.

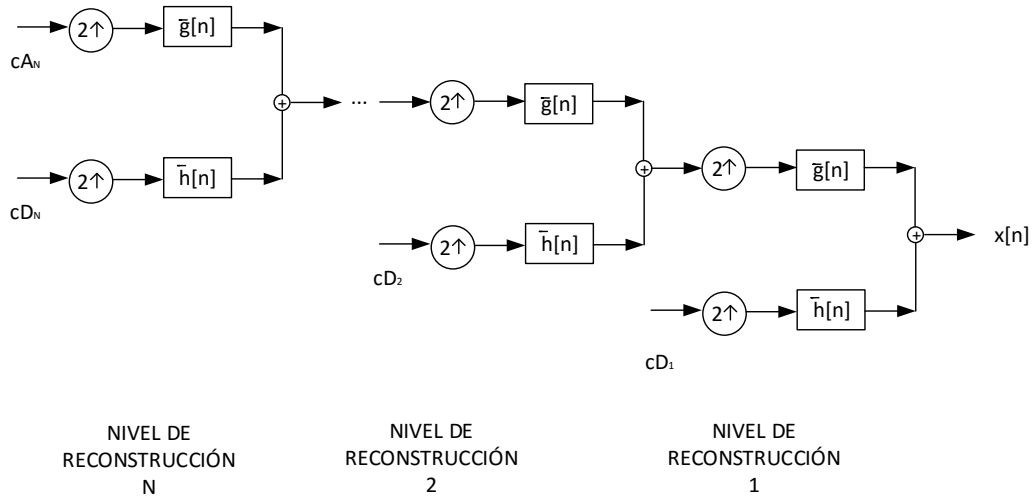


Figura 2.10: Análisis Multi-Resolución. Diagrama de la Transformada Wavelet Inversa.

Por último, cabe decir que esta técnica se puede enmarcar tanto en la compresión con pérdidas como en la compresión sin pérdidas, dependiendo de las necesidades del caso concreto:

- Por un lado, a través del Análisis Multi-Resolución Inverso es posible obtener una reconstrucción exactamente igual que la señal original. Para ello, basta con utilizar todos los coeficientes de aproximación y detalle previamente obtenidos. Sin embargo, el nivel de compresión conseguido en este caso es nulo.
- Por otro lado, si se utilizan solamente los coeficientes de aproximación del nivel de descomposición N y algunos o ninguno de detalle, la reconstrucción de la señal original no será exacta, pero a cambio se ha conseguido un nivel de compresión.

2.2.6. Familias de Wavelets

A continuación, en este punto se va a analizar el término de Familia *wavelet*. Como ya se ha comentado en el apartado 2.2.3, la Transformada Wavelet de una señal se obtiene mediante un conjunto de traslaciones y dilataciones de lo que se denomina una *wavelet* madre $\psi(t)$, para descomponer dicha señal en diferentes componentes de una frecuencia determinada. Existen una gran variedad de wavelets madre diferentes, aunque las familias más conocidas son las Daubechies, Haar, *Symlets*, Coiflets, Meyer, etc. En general, no existe una *wavelet* óptima para todos los casos, sino que dependiendo de la señal de interés algunas de ellas se ajustarán mejor que otras.

Cada familia *wavelet* presenta unas características distintas. En la Tabla 2.2 se recoge, para las familias wavelets más comunes, un conjunto de propiedades que se han seleccionado pues son las más importantes.

	Haar	Sombrero mejicano	Morlet	Daubechies	Symlets	Coiflets	Gaussiana	Biortogonal	Meyer	Meyer discreta
Orden	—	—	—	1,2,...45	1,2,...41	1,2,...5	1,2,...45	1,1,1,2,...6.8	—	—
Relatividad	No	Si	Si	Relativa	Relativa	Relativa	Si	Relativa	Si	Si
Tamaño	1	[-5,5]	[-4,4]	2N-1	2N-1	6N-1	[-5,5]	2Nd+1	[-8,8]	[-8,8]
Longitud del filtro	2	—	—	2N	2N	6N	—	—	—	—
Simetría	Si	Si	Si	No	Aprox.	Aprox.	Si	Si	Si	Si
Momentos de desvanecimiento	1	—	—	N	N	2N	—	Nr	—	—
Función de escala	Si	No	No	Si	Si	Si	No	Si	No	Si
CWT	Posible	Posible	Posible	Posible	Posible	Posible	Posible	Posible	Posible	Posible
DWT	Posible	No permite	No permite	Posible	Posible	Posible	No permite	Posible	No permite	Posible
Expresión	Si	Si	Si	No	No	No	Si	No	Si	Si

Tabla 2.2: Características de las familias de las wavelets madre.

La característica de tipo orden, sirve para diferenciar a las *wavelets* madre, de una misma familia, por la forma de onda que presentan. La Figura 2.11 muestra nueve tipos distintos de *wavelets* dentro de la familia Daubechies. Como se puede observar, existe diferencia con respecto al orden pues, por ejemplo, la *Daubechies3* (db3) tiene una forma de onda diferente a una *Daubechies9* (db9).

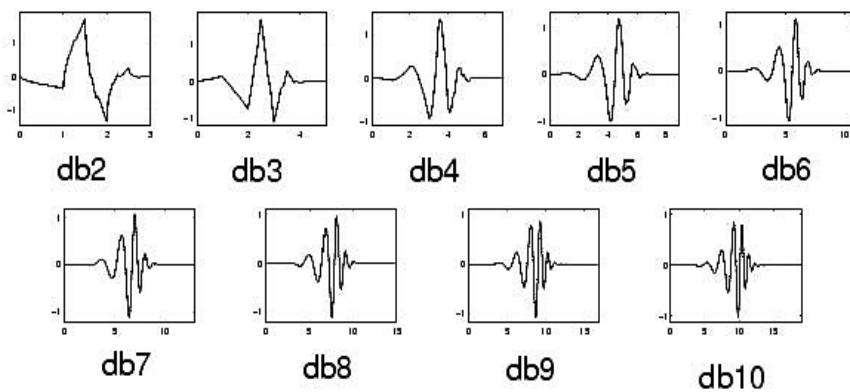


Figura 2.11: Distintas formas de onda para la familia Daubechies [9].

Por otro lado, cabe mencionar aspectos como la simetría, la función de escala y el tamaño. Si una *wavelet* es simétrica, y se mira como si de un filtro se tratase (en el Análisis Multi-Resolución), se podría decir que presenta una fase lineal, mientras que si no es simétrica se introduce distorsión en la fase. Por ejemplo, en el caso de las Symlets, se deben elegir los parámetro del filtro de forma adecuada para que las funciones *wavelets* generadas sean lo más simétricas posibles. En cuanto a la función de escala, este parámetro indica la posibilidad o no de poder utilizar la Transformada Wavelet Discreta. En este

proyecto se va a utilizar esta transformada, en particular el Análisis-Multiresolución, por lo que se usarán aquellas familias que presente esta posibilidad como son la Haar, Daubechies, Symmlets y Coiflets. Finalmente, cabe destacar que el tamaño está directamente relacionado con el número de coeficientes del filtro. La importancia de estas propiedades depende fundamentalmente del caso de aplicación concreto.

2.3. Trabajos previos

Una vez analizadas las bases teóricas de la herramienta MRTG y la transformada Wavelet, surge la necesidad de revisar los trabajos relacionados ya existentes con estas técnicas. Como ya se ha comentado en la introducción de este proyecto, la periodicidad de las señales con las que se trabaja motiva la utilización de la Transformada Wavelet.

poder ser tratada de forma eficiente, por lo que reducir su tamaño se convierte en una tarea que cada vez está teniendo más relevancia. Como es lógico, la agregación de datos reduce su volumen, pero también limita la información disponible. Existen una serie de estudios relacionados con la escala de tiempo más óptima a utilizar. Por ejemplo, el RFC 1857 [10] recomienda que el intervalo de muestreo sea de un minuto para detectar picos en el comportamiento del tráfico. Sin embargo, a esa tasa de muestreo la cantidad de datos obtenida es inmanejable, con lo que se propone agregar los datos de la siguiente forma:

- Agregación de 15 minutos para períodos diarios.
- Agregación de 1 hora para períodos mensuales.
- Agregación de 1 día para períodos anuales.

La agregación de 15 minutos se justifica dado que dicho intervalo debe ser lo suficientemente pequeño para poder capturar las variaciones en la actividad de los seres humanos. De acuerdo a este RFC, media hora es una buena estimación del tiempo en el cual las personas permanecen realizando una actividad determinada.

Por otro lado, la mayoría de ISPs (*Internet Service Providers*) recogen las estadísticas cada 5 minutos y las almacenan en sus bases de datos. Para limitar el tamaño de estas bases de datos, solamente los más recientes se almacenan a la escala de tiempo original, mientras que los datos más antiguos se almacenan en escalas de tiempo más grandes. Estas agregaciones son arbitrarias y en algunos casos se pierde información de monitorización valiosa, como ocurre en la herramienta MRTG con su proceso de promediado simple.

En [11] se propone una técnica para la agregación de los datos, basada en un equilibrio entre la eficiencia en el almacenamiento y la pérdida de información, evitando reglas arbitrarias. Se define un mecanismo para determinar un umbral objetivo para detener el proceso de sub-muestreo de una señal relativa al uso de red. El objetivo era permitir que cualquier análisis posterior sobre estas medidas fuera más simple, pues trataba con señales más cortas pero idealmente similares en términos de red. Como método de sub-muestreo

aplicaron el Análisis Multi-Resolución con *wavelets*, y como mecanismo de umbral de sub-muestreo una métrica relacionada con el impacto del rendimiento del *router*. Esto es, una serie temporal original y las series resultantes de un sub-muestreo iterativo se introducen a un *router* de cola infinita de un sólo servidor con carga media, considerando que dos señales son equivalentes en términos de red si producen un tiempo de estancia en el sistema equivalente. Esta equivalencia se establece cuando se supera un test de bondad de ajuste de la distribución de tiempo estancia en el sistema de la señal sub-muestreada, con respecto a la original. Este mecanismo se aplicó a series temporales reales obtenidas de RedIRIS y se llegó a la conclusión que es posible reducir los datos a un cuarto de su tamaño original en el tráfico generado por la mayoría de las universidades analizadas y hasta un octavo para los datos más agregados, siempre manteniendo un alto nivel de confianza.

Por otro lado, en [12] se propone un algoritmo para extraer los rasgos más significativos de las series temporales, aplicando la técnica FPCA (*Functional Principal Components Analysis*). El estudio pretende centrarse en la compresión de los datos, sin cambiar su resolución temporal. El rendimiento obtenido se ha comparado en términos de reducción del volumen de datos y precisión obtenida mediante el MAPE – *Mean Absolute Percentage Error*, con respecto a otros métodos previamente aplicados con *wavelets* a series temporales. En conclusión, se han conseguido mejores resultados para volúmenes de más de 180 días. Así mismo, FPCA representa las observaciones más típicas mejor que las *wavelets* usando un número redondeado de aproximaciones y mejorando el compromiso entre la compresión y pérdida de información. También ofrece una mejor flexibilidad en la selección de los niveles de error y compresión.

Una vez analizados los trabajos más relevantes, cabe decir que este proyecto pretende mejorar la herramienta MRTG usando el Análisis Multi-Resolución como sustitución del proceso de promediado simple usado por MRTG. De este modo, el nivel de sub-muestreo viene prefijado a tal y como lo determina MRTG y será, por tanto, un parámetro para nuestra propuesta. Esto es, la aportación es que mientras se respeta el nivel de almacenamiento que MRTG usa, vamos a evaluar cuánto podemos mejorar su precisión mediante *wavelets*, analizando múltiples señales y familias de funciones y, además, facilitamos como código libre una implementación equivalente a la de MRTG que denominaremos *WaveTG*.

Capítulo 3

Entorno de trabajo

EN este capítulo se analizará el escenario de aplicación del trabajo prestando especial atención en el patrón periódico característico de las señales disponibles para el estudio. Seguidamente, se detallarán las herramientas utilizadas para alcanzar los objetivos del proyecto. Por último, se estudiarán las medidas de precisión que ayudarán a evaluar la validez del sistema y que serán clave para el desarrollo de la aplicación final.

3.1. Escenario de aplicación

Los datos disponibles en este proyecto se corresponden a medidas del ancho de banda tomadas cada 5 minutos por la herramienta MRTG, instalada en los equipos de RedIRIS. La RedIRIS se compone de 18 puntos de presencia (PoP – *Point of Presence*) distribuidos a lo largo del territorio español, como se puede ver en la Figura 3.1 [13]. Esta red proporciona acceso a Internet a más de 350 instituciones, entre las que se incluyen principalmente universidades, centros de investigación, hospitales, etc. En total, el número de usuarios asciende a más de un millón.

Además, esta red tiene puntos de intercambio con la Red Europea de Investigación y Educación GÉANT y con otros ISPs como por ejemplo Telia, Global Crossing, etc. Del total del conjunto de datos facilitados por RedIRIS, en este trabajo se han seleccionado 14 enlaces, correspondientes a aquellos con mayor número de medidas:

- 5 enlaces de universidades a los que denominaremos U1, U2, U3, U4 y U5.
- 4 centros de investigación a los que denominaremos C1, C2, C3 y C4.
- 2 enlaces con puntos neutros españoles a los que denominaremos IXP1 e IXP2.
- 2 enlaces con ISPs externos a los que denominaremos ISP1 e ISP2.
- 1 enlace con la Red Europea de Investigación y Educación al que denominaremos E1.

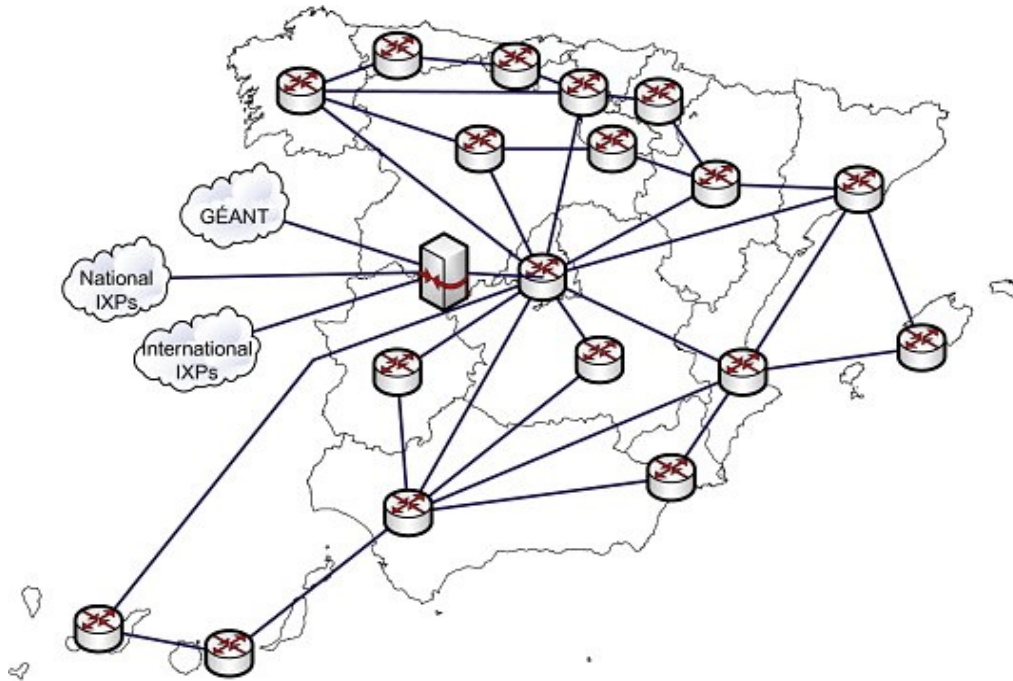


Figura 3.1: Arquitectura de RedIRIS.

En total, se dispone de una gran base de datos de medidas de MRTG, de unos 3 años de duración (2007, 2008 y 2009). Dada la granularidad de 5 minutos en la captura de datos, se tiene un total de 288 muestras cada día en ambos sentidos del enlace (entrante y saliente). En cuanto al formato de las medidas, cabe destacar que se compone de 5 campos:

1. UNIX *Timestamp* (1ª columna): Determina la referencia temporal de forma inequívoca. Este valor se especifica en segundos, por lo que la diferencia entre muestras consecutivas es de 300 segundos para la granularidad utilizada.
2. Tasa media de transferencia entrante en bytes/segundo (2ª columna).
3. Tasa media de transferencia saliente en bytes/segundo (3ª columna).
4. Tasa máxima de transferencia entrante en bytes/segundo (4ª columna).
5. Tasa máxima de transferencia saliente en bytes/segundo (5ª columna).

Por otro lado, dado que RedIRIS es una red principalmente académica, su patrón de tráfico se diferencia ligeramente del dado en redes residenciales, aunque el comportamiento periódico se mantiene [14] [15] [16]. De esta forma, en vez de encontrar el pico máximo de tráfico después de las 8:00 P.M., cuando los usuarios vuelven a casa después del trabajo, en RedIRIS los valores máximos se dan alrededor del mediodía tal y como se puede observar en la Figura 3.2, donde se representa el patrón de tráfico diario en un enlace de esta red. Además, se puede observar que el tráfico durante la madrugada es notablemente más bajo

que durante la mañana y la tarde. Esto también tiene sentido, pues la mayor parte de centros que componen la red permanecen cerrados por la noche.

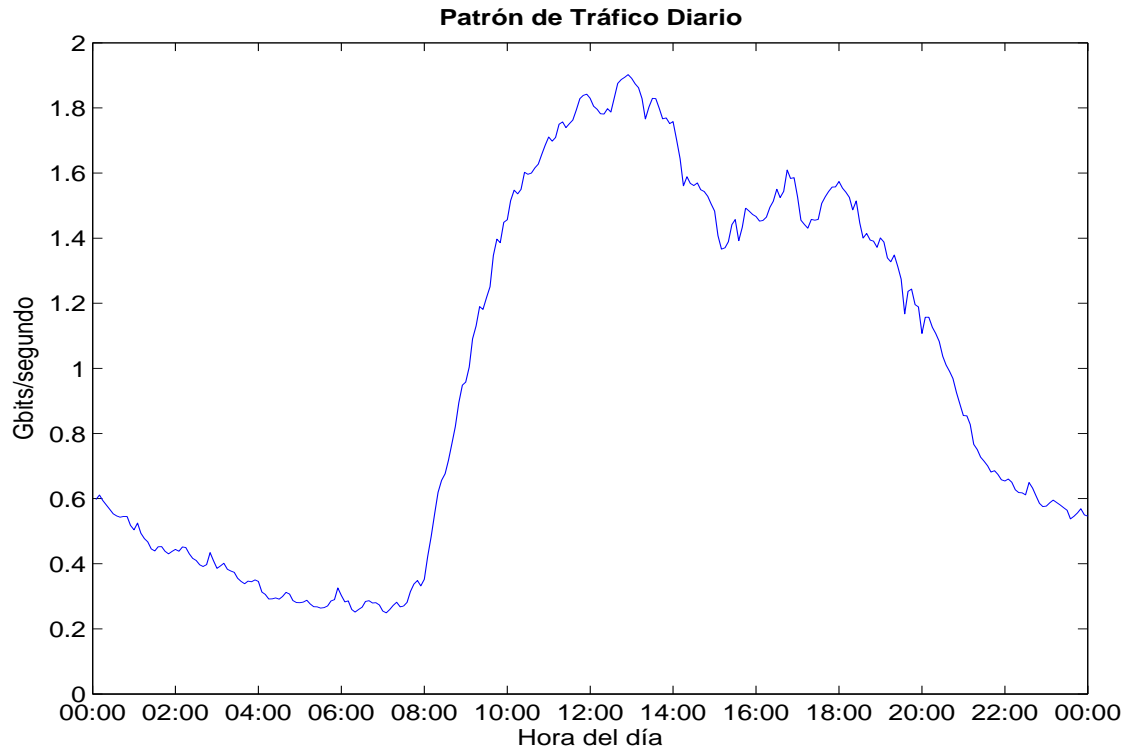


Figura 3.2: Ejemplo de patrón de tráfico diario.

Siguiendo con el estudio de la periodicidad en el patrón del tráfico, se ha analizado también los datos correspondientes a un mes consecutivo de medidas, correspondiente al intervalo de tiempo entre el 12 de Febrero de 2008 y el 21 de Febrero del mismo año (Figura 3.3). Se puede observar una clara periodicidad a nivel semanal, distinguiéndose el comportamiento del tráfico durante los días laborales del tráfico durante los fines de semana (llegando a ser hasta 4 veces menor).

Lo mismo ocurre a nivel mensual y anual, aunque el comportamiento periódico no es tan claramente diferenciado. De hecho, en los meses de Agosto, el comportamiento del tráfico rompe con la tendencia de los meses anteriores y posteriores. Por tanto, se ha decidido eliminar este mes del estudio en todas las series temporales. En definitiva, se ha trabajado con un total de 118 semanas (826 días) de medidas para cada uno de los 14 enlaces descritos anteriormente, concatenando los 3 siguientes intervalos temporales:

- Lunes 12 Febrero 2007, 00:00:00 h – Domingo 29 Julio 2007, 23:55:00 h
- Lunes 3 Septiembre 2007, 00:00:00 h – Domingo 27 Julio 2008, 23:55:00 h
- Lunes 1 Septiembre 2008, 00:00:00 h – Domingo 26 Julio 2009, 23:55:00 h

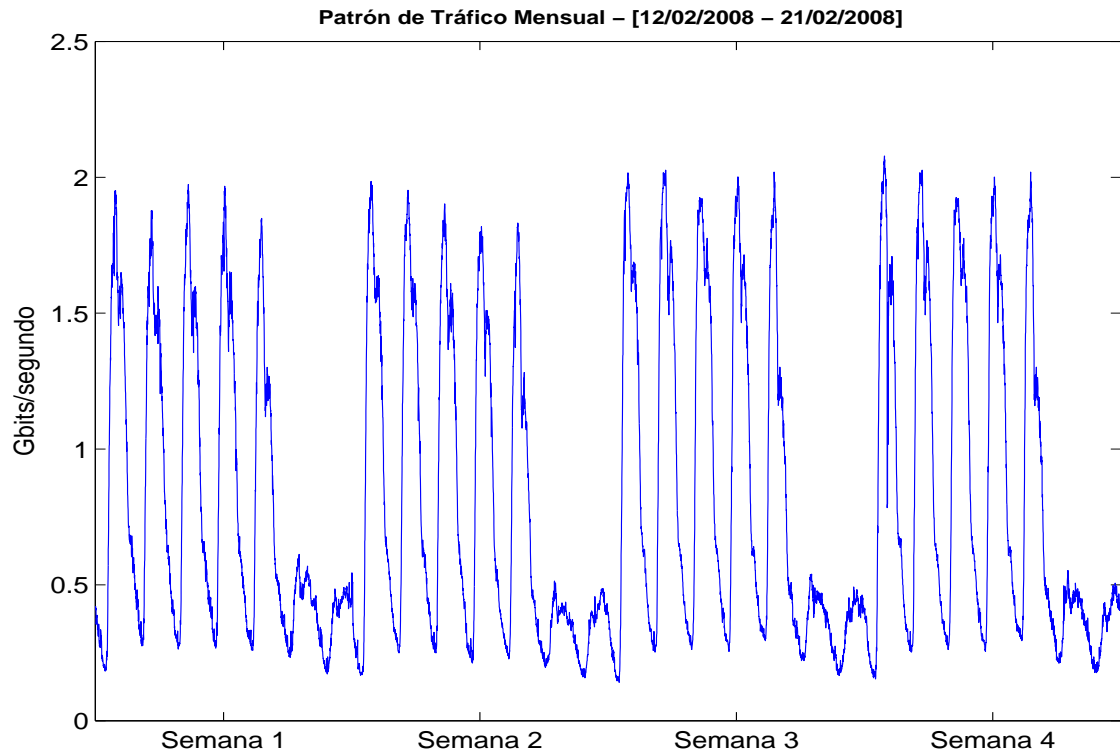


Figura 3.3: Ejemplo de patrón de tráfico mensual.

Por último cabe destacar la importancia que tiene el estudio del comportamiento del tráfico mediante la búsqueda de patrones periódicos. Actualmente, existen numerosos estudios [17] que tratan de simular el comportamiento global de Internet. Estas simulaciones, que son complementarias a los análisis, permiten explorar escenarios complejos que sería difícil o imposible analizar. Por ejemplo, se pueden simular situaciones de congestión para desarrollar mecanismos que intenten lograr una mejor eficiencia en esos casos. Sin embargo, simular el comportamiento global de Internet es un gran reto, dada la enorme heterogeneidad y rápido cambio de las redes. Por tanto, la búsqueda de invariantes como los patrones de tráfico simplifican enormemente estas simulaciones. A pesar de los claros patrones de tráfico diarios, semanales y mensuales, existen variaciones inevitables como por ejemplo:

- Diferentes protocolos presentan patrones distintos, especialmente aquellos que no se inician mediante la interacción con los seres humanos, como por ejemplo el protocolo NNTP (*Network News Transport Protocol*)
- Un mismo protocolo puede presentar patrones diferentes. Por ejemplo, el comportamiento de los protocolos relacionados con la navegación web presentan cambios en función del intervalo temporal del día.
- También existen variaciones como consecuencia de las comunicaciones que se dan entre zonas horarias diferentes.

En definitiva, las señales de trabajo en este proyecto tienen en general un comportamiento periódico aunque pueden presentar numerosas variaciones inevitables.

3.2. Herramientas utilizadas

Una vez se ha descrito el escenario de trabajo y analizado el comportamiento periódico de las señales, a continuación se detallarán las herramientas utilizadas para llevar a cabo el desarrollo y la evaluación del proyecto. Cabe destacar que un hecho importante que ha conducido a la elección de estas herramientas ha sido la posibilidad de utilizar funciones y/o librerías que implementen la Transformada Wavelet. De esta manera, se evita la implementación de estas funciones pudiendo centrar el trabajo en los objetivos marcados al comienzo del mismo.

La primera herramienta utilizada ha sido Matlab, en su versión 2013b dada su potencialidad respecto al tratamiento de señales. Mediante este programa se han implementado distintos modelos, los cuales han sido evaluados, obteniendo unos resultados en base a unas medidas de precisión que se analizarán en el punto 3.3. En función de dichos resultados, se ha decidido cómo desarrollar y parametrizar el sistema final. Es decir, esta herramienta ha sido de utilidad en la primera fase del desarrollo para obtener una primera aproximación al sistema final.

Primeramente, se ha realizado un estudio de las funciones disponibles en Matlab relacionadas con la Transformada Wavelet Discreta, y en particular con el Análisis Multi-Resolución. En particular, la función *wavedec()* es la que se encarga de realizar esta operación, recibiendo una señal de entrada, un nivel de descomposición y un tipo de *wavelet*. En cuanto a la salida, esta función devuelve dos vectores correspondientes al vector de coeficientes (aproximación y detalle) y otro que refleja el número de coeficientes de cada nivel. La Figura 3.4 muestra un ejemplo del comportamiento de esta función.

Como refleja la Figura 3.4, la función *wavedec()* devuelve un vector *C* con todos los coeficientes concatenados, por lo que se debe utilizar el segundo vector *L* para conocer qué conjunto de coeficientes corresponden a cada nivel. Para ello, existen otras dos funciones denominadas *appcoef()* y *detcoef()* las cuales reciben como entrada los vectores anteriores, el tipo de *wavelet* y el nivel de descomposición y son capaces de obtener los coeficientes de aproximación y detalle segmentados en niveles. Por otro lado, dado que la Transformada Wavelet se ha diseñado para señales de longitud infinita y en este caso se está trabajando con señales finitas, ha sido necesario cambiar el modo de funcionamiento de esta Transformada mediante el comando *dwtmode('per')*. En cuanto a la reconstrucción, para realizar la Transformada Wavelet Inversa se ha utilizado la función *waverec()*, la cual recibe como datos de entrada los vectores *C* y *L* de las mismas características que los anteriores, el nivel de descomposición y el tipo de *wavelet* madre. De forma automática, esta función obtiene la señal reconstruida, del mismo número de muestras que la original.

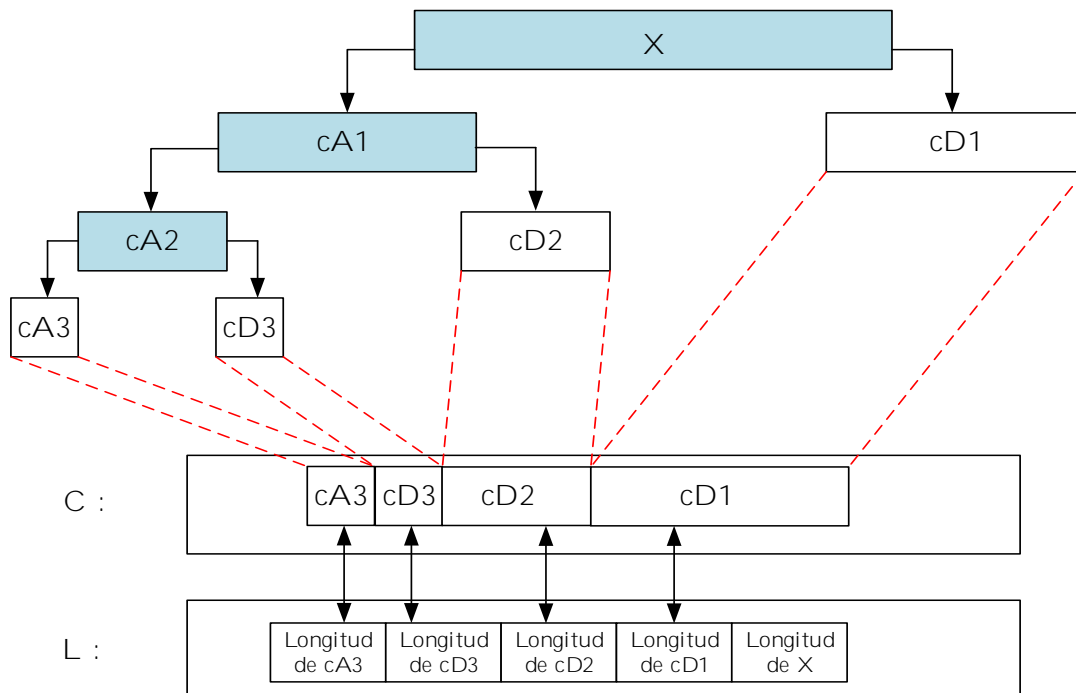


Figura 3.4: Función `wavedec()` de Matlab.

En segundo lugar, se ha utilizado el lenguaje de programación Python en su versión 2.7.6, para implementar la versión definitiva de nuestra alternativa a MRTG, la cual ha sido denominada *WaveTG*. Se ha decidido utilizar este lenguaje dado su enorme uso en la actualidad. Además, su versatilidad y la gran comunidad de usuarios que posee simplifican los problemas de programación existentes. En este caso, se ha utilizado una librería específica denominada *pywt* que contiene las mismas funciones `wavedec()` y `waverec()` que han sido utilizadas en Matlab. Finalmente, también cabe decir que se ha intentado dejar atrás el lenguaje de programación Perl utilizado por MRTG, puesto que cada vez más está en desuso.

3.3. Medidas de precisión

A continuación se detallarán las medidas que se utilizarán para comparar la precisión de la propuesta usando *wavelets* con respecto al de MRTG. Básicamente lo que se pretende es evaluar el grado de similitud entre la señal reconstruida, tras aplicar la Transformada Wavelet inversa a los coeficientes almacenados, y la señal original.

En primer lugar, la primera medida de precisión que se utilizará será el Error Cuadrático Medio (MSE – *Mean Squared Error*). Este parámetro mide el promedio de los errores al cuadrado entre los valores reales y los estimados. Su definición matemática es la siguiente se recoge en la ecuación 3.1.

$$MSE = \frac{1}{N} \sum_{n=1}^n (\widehat{X}_n - X_n)^2 \quad (3.1)$$

Dado que las señales representan la carga de tráfico por segundo en un enlace determinado, el Error Cuadrático Medio tendrá unidades de $(\frac{\text{bits}}{\text{segundo}})^2$. Esta herramienta matemática será la que se use para evaluar la precisión en cuanto a la pérdida de información de las distintas señales reconstruidas respecto a las originales.

Por otro lado, también se utilizará una segunda medida de precisión para evaluar, en este caso, al modelo que finalmente será desarrollado en Python. Esta medida es el MAPE (*Mean Absolute Percentage Error*), que devuelve una estimación de la precisión de la predicción en forma de porcentaje. Se define mediante la ecuación 3.2.

$$MAPE = \frac{100}{N} \sum_{n=1}^n \left| \frac{X_n - \widehat{X}_n}{X_n} \right| \quad (3.2)$$

siendo X_n es el valor actual y \widehat{X}_n el valor predicho. Sin embargo, el MAPE tiene algunas características las cuales se deben tener en cuenta:

- No pueden existir valores nulos, pues habría una división por cero. En general, las medidas del ancho de banda presentan valores no nulos, pero se debe tener cuidado cuando existen cortes en la red o fallos en el sistema de monitorización.
- Para valores predichos muy bajos, el MAPE puede no puede sobrepasar el 100 %, pero para valores predichos muy altos (como puede ocurrir en este caso) no existe un límite superior.

Capítulo 4

Evaluación de la propuesta

EN este capítulo se analizará la precisión que la propuesta desarrollada alcanzará para un conjunto significativo de escenarios. Para esto se utilizará como principal herramienta Matlab. En primer lugar, se estudiará la precisión del sub-muestreo simple de MRTG de cara a obtener una base como referencia a mejorar. En segundo lugar, se mostrará la arquitectura del sistema desarrollado mediante un diagrama de bloques, explicando de forma concisa cada una de sus partes. El sistema final, se elegirá tras estudiar tres modelos distintos basados en la Transformada Wavelet. Por lo tanto, a continuación se detallarán los pasos seguidos para el desarrollo de los tres modelos, así como se analizarán las características de cada uno de ellos de forma exhaustiva.

Seguidamente, se evaluarán los algoritmos propuestos basados en la Transformada Wavelet y se compararán con MRTG en un conjunto significativo de enlaces. Cabe destacar que se trabajará en todo momento con señales reales de tráfico, concretamente aquellas que se analizaron en el punto 3.1. La comparativa de la precisión de estos modelos, será la que indique qué modelo final se implementará en Python.

4.1. Precisión de MRTG

Antes de comenzar con el desarrollo y evaluación del sistema propuesto, primeramente se va a analizar la precisión de MRTG. Como ya se ha comentado en el punto 2.1, MRTG aplica un proceso de promediado simple a las señales de entrada en las distintas granularidades. De esta forma, a nivel semanal se realiza un promedio cada 6 muestras (media hora), a nivel mensual cada 24 muestras (2 horas) y a nivel anual cada 288 muestras (un día). Cabe recordar que el intervalo temporal entre muestras permanece constante e igual a 5 minutos. Por tanto, tanto una semana, como un mes y un año se representan con el mismo número de muestras, en concreto 336:

- Señal de 1 semana de duración, 2016 muestras $\rightarrow 2016 / 6 = 336$ muestras
- Señal de 1 mes de duración, 8064 muestras $\rightarrow 8064 / 24 = 336$ muestras
- Señal de 1 año de duración, 96768 muestras $\rightarrow 96768 / 288 = 336$ muestras

Estas muestras son las que se almacenan en la base de datos siguiendo una filosofía FIFO (*First Input First Output*) y se representan en forma de gráficas PNG en los archivos HTML generados.

El problema de esta herramienta es evidente cuando se requiere reconstruir la señal original de la mejor manera posible. En el caso anual, por ejemplo, únicamente se dispone de 336 valores, los cuales representan a todo el año. Si se desea obtener una señal reconstruida con el mismo número de muestras que la señal original, no hay más remedio que establecer los valores por bloques de forma que todas las muestras de un mismo día (288 en concreto) comparten el mismo valor. Este hecho hace que se premie más la compresión de los datos que el nivel de fiabilidad de la señal reconstruida. En algunos casos, la pérdida de información puede ser importante de cara a un análisis posterior de los datos.

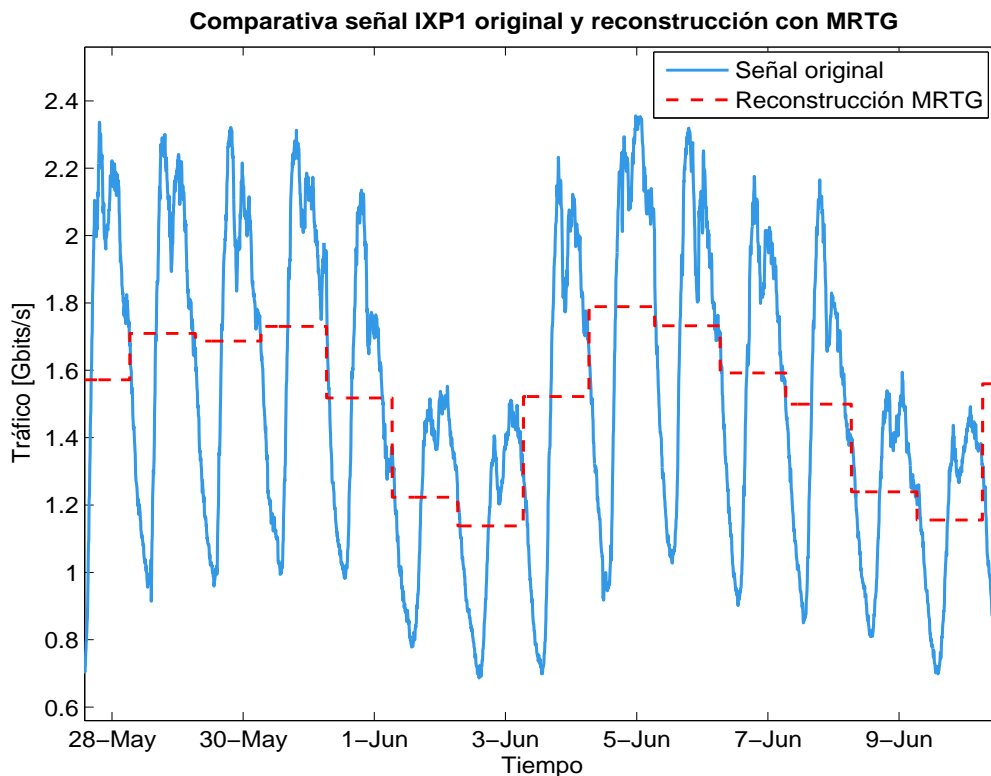


Figura 4.1: Comparativa entre un fragmento de la señal IXP1 y la reconstrucción con MRTG.

En las figuras 4.1 y 4.2 se muestran dos ejemplos de este hecho. Para ello se han utilizado dos señales con distinto volumen de datos, IXP1 e IXP2. Mientras que IXP1 presenta una gran carga de tráfico en sus enlaces, IXP2 presenta casi 100 veces menos tráfico. Para ambas señales, se ha representado el tráfico entrante en el mismo intervalo temporal, correspondiente al periodo entre el 28 de Mayo de 2017 y el 11 de Junio de este mismo año. En ambos casos se puede comprobar como la reconstrucción con MRTG dista mucho de la señal original, independientemente del volumen de tráfico total.

Como se ha dicho anteriormente, MRTG tendrá almacenadas 14 muestras, cada una de ellas correspondientes a cada día de estas dos semanas que se representan. La reconstrucción con MRTG se ha realizado fijando las 288 muestras que representan a un día con el valor de la muestra correspondiente almacenada.

Los resultados obtenidos son muy poco fiables y han motivado a la realización de este Trabajo Fin de Máster. Se pretende implementar una técnica que, mediante la Transformada Wavelet, obtenga una reconstrucción de la señal lo más similar posible a la señal original.

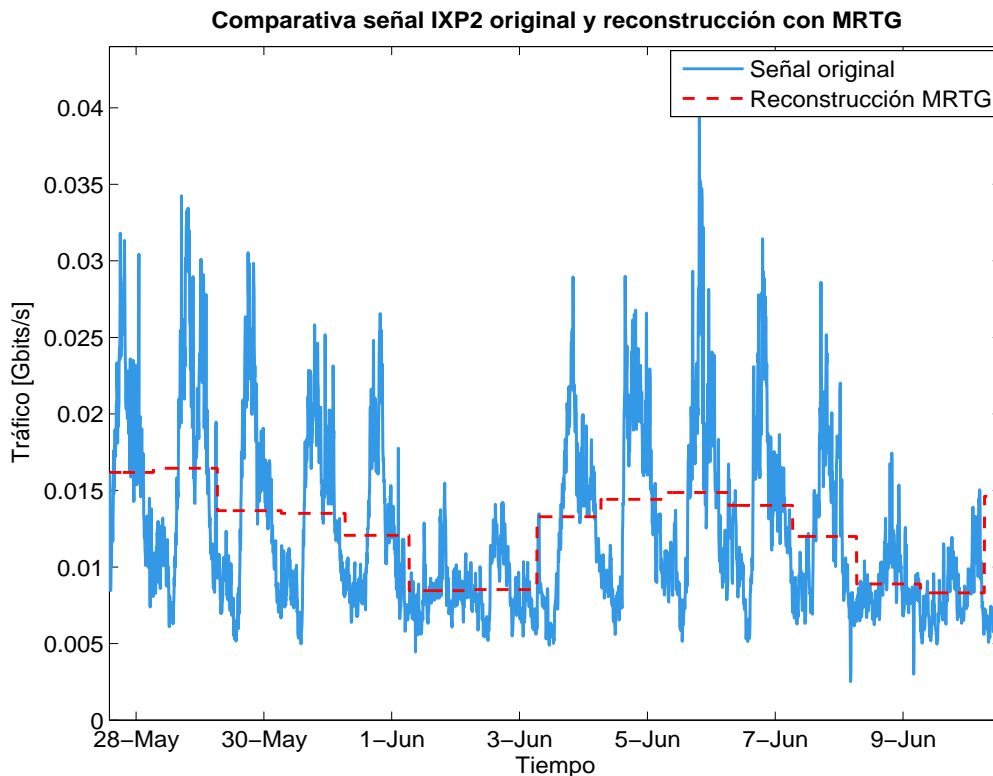


Figura 4.2: Comparativa entre un fragmento de la señal IXP2 y la reconstrucción con MRTG.

Finalmente, se van mostrar los resultados obtenidos a nivel semanal, mensual y anual tras aplicar la primera medida de la precisión, es decir, el Error Cuadrático Medio. El procedimiento seguido ha sido a nivel mensual es el siguiente. Primero, se ha dividido cada una de las señales de la Base de Datos en intervalos temporales correspondientes a una semana. Posteriormente, para cada una de las semanas se ha aplicado el proceso de promediado simple que agrega los datos en MRTG y se han reconstruido las señales. A continuación se ha calculado el MSE entre todas las señales originales y las reconstruidas y se ha realizado su media. Este mismo procedimiento se ha repetido a nivel mensual y anual.

La Figura 4.3, muestra una comparativa entre el MSE obtenido a nivel semanal, mensual y anual para cada una de las señales de la Base de Datos. Analizando esta Figura, hay un primer aspecto que llama la atención y es la notable diferencia entre el valor de MSE para la agregación anual entre las señales IXP1 y ISP1 y el resto de señales. Esto se debe a la gran cantidad de tráfico que se recoge en estos enlaces. Fijándonos en la definición de MSE, esta medida de precisión está muy relacionada con la amplitud de los datos, por lo que se obtienen valores mayores cuanto mayores son las medidas del ancho de banda. Sin embargo, estas grandes diferencias en cuanto al MSE no representan la bondad de la reconstrucción en las distintas señales. De hecho, en las figuras 4.1 y 4.2, se muestra cómo la reconstrucción obtenida es igual de mala en ambos casos a pesar de la notable diferencia en cuanto al volumen de datos entre IXP1 y ISP1. Por tanto, a pesar de que el MSE sea mayor en algunas señales que en otras debido al volumen de tráfico, se obtienen reconstrucciones poco fiables generalmente, por lo que la precisión del proceso promediador simple de MRTG es bastante mejorable.

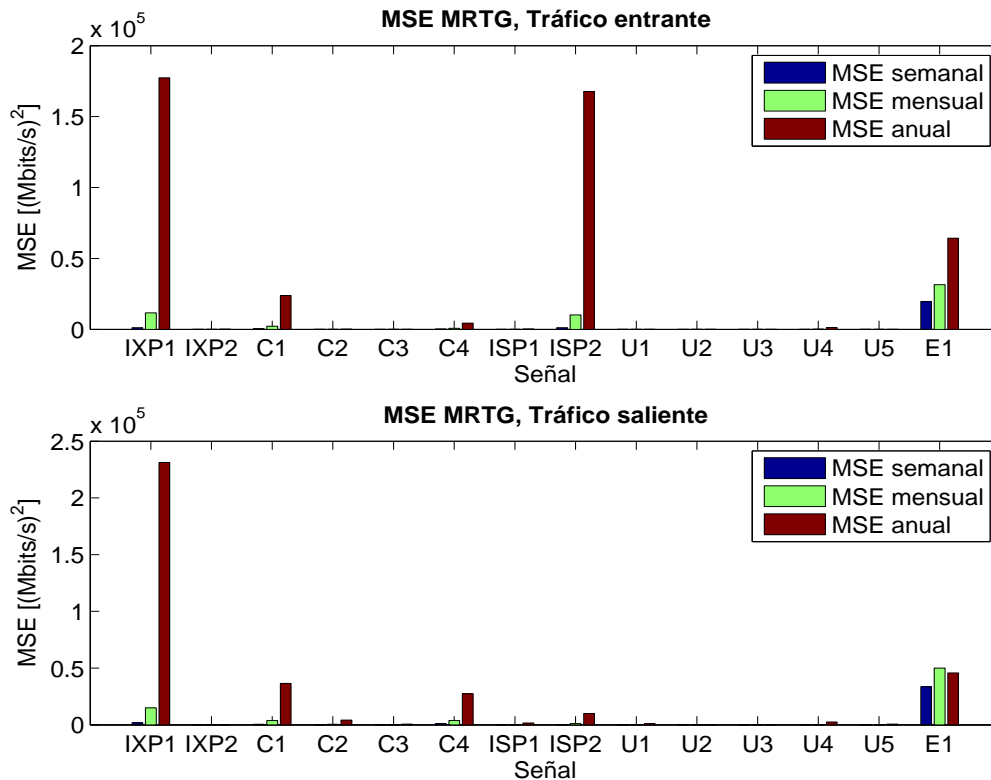


Figura 4.3: Medidas de precisión obtenidas tras aplicar MRTG a las señales de la Base de Datos.

Por tanto, surge la necesidad de encontrar otras técnicas que consigan obtener una reconstrucción más fiable de la señal original, manteniendo un buen nivel de compresión. Para ello, se ha decidido utilizar el Análisis Multi-Resolución de la Transformada Wavelet en lugar de la técnica de promediado simple utilizada en MRTG. Como ya se ha comentado con anterioridad, el carácter periódico de las señales de trabajo, analizado en el punto 3.1,

motivan la aplicación de dicha Transformada. A continuación, se detallará la arquitectura del sistema propuesta, así como los modelos desarrollados de cara a la obtención de una mejor precisión.

Además, tras las conclusiones obtenidas en este punto, a partir de ahora no se utilizará el MSE como medida de precisión en sí, sino que se comprobará la bondad de los resultados obtenidos con los algoritmos implementados como la mejora relativa del MSE. En el punto 4.3 se analizará esto con más detalle.

4.2. Arquitectura del sistema

En este punto, se va describir la arquitectura del sistema propuesto en forma de bloques, explicando brevemente cada una de sus etapas. Tal arquitectura se esquematiza en la Figura 4.4.

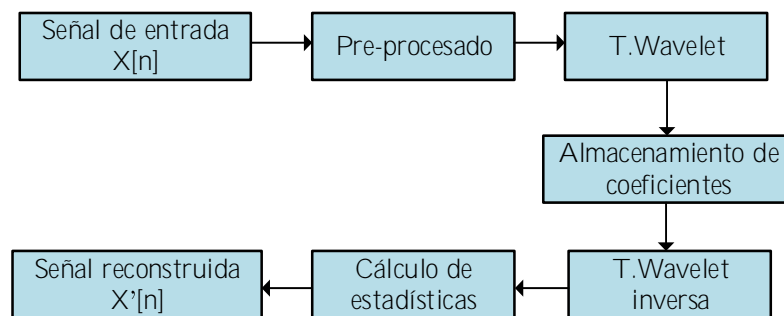


Figura 4.4: Diagrama de bloques del sistema.

- Señal de entrada:** Las señales de entrada corresponderán a las series temporales que representan el ancho de banda, tanto entrante como saliente, de los 14 enlaces descritos en el punto 3.1. Como ya se ha analizado previamente, las muestras que forman estas series temporales son capturadas cada 5 minutos haciendo uso de la herramienta MRTG instalada en los nodos de la RedIRIS. Se tomarán un total de 118 semanas de tráfico para cada enlace, eliminando el mes de Agosto en todos los casos pues las muestras de este mes rompen con la tendencia periódica de las señales. Esto se debe a que Agosto suele ser el período vacacional por excelencia en España y más aún en las instituciones (universidades, centros de investigación, etc.) que componen la RedIRIS.
- Pre-procesado de la señal de entrada:** Dado que las señales con las que se trabaja contienen datos reales, la etapa de pre-procesado es realmente importante. Estas señales están sujetas a cortes en la red y fallos en el sistema de monitorización, sobre todo cuando se trabaja con un intervalo temporal tan grande, por lo que es necesario un mecanismo de reconstrucción para mitigar dichos huecos. Por otro lado, un hecho importante a destacar es el cambio de hora realizado 2 veces al año en

España. Aunque parezca inicialmente un hecho irrelevante, los días en los que se adelanta una hora se pierden 12 muestras $\frac{60min}{5min/muestra}$, mientras que los días en los que se atrasa una hora se tienen 12 muestras más. Es realmente importante que cada día tenga el mismo número de muestras para poder aplicar con éxito el Análisis Multi-Resolución, por lo que se ha optado por la siguiente solución. En los días con una hora menos, se ha copiado la última hora de medidas correspondiente al intervalo temporal [01:00 h – 02:00 h] de dicho día. Por otro lado, en los días que tienen una hora más, se ha realizado un promedio simple de forma que en el intervalo temporal [02:00 h – 03:00 h] hay únicamente 12 muestras, en lugar de 24.

- **Transformada Wavelet:** Una vez se tienen las señales de entrada pre-procesadas, a continuación se les debe aplicar la Transformada Wavelet. Concretamente, como ya se ha comentado, se aplicará el Análisis Multi-Resolución para obtener un número fijo de coeficientes que representen a cada señal. En el punto 4.3 se explicará en detalle este proceso, analizando las distintas posibilidades planteadas. Cabe destacar que se deberá tener en cuenta el equilibrio existente entre la compresión de la señal y la pérdida de información. Cuanto mejor sea el ratio de compresión, más eficiente será el algoritmo con respecto al almacenamiento, sin embargo, el nivel de fiabilidad de la señal reconstruida también será más bajo. Al contrario ocurre lo mismo, si se almacenan demasiados coeficientes *wavelet* se podrá obtener una señal reconstruida muy similar a la señal original pero se perderá eficiencia en cuanto al almacenamiento.
- **Almacenamiento de coeficientes:** Una vez obtenidos los coeficientes *wavelet* (de aproximación y detalle) correspondientes a las señales originales, se deberán almacenar en una base de datos de forma que se pueda acceder a ellos en cualquier momento para realizar el proceso inverso y obtener las señales reconstruidas. Para no desbordar la base de datos, se eliminarán los datos de forma progresiva siguiendo un comportamiento FIFO. Por tanto, prevalecerán los datos más recientes frente a los más antiguos en las distintas agregaciones temporales.
- **Transformada Wavelet Inversa:** Para poder obtener la señal reconstruida, se debe realizar el Análisis Multi-Resolución Inverso a los coeficientes *wavelet* almacenados. Cabe destacar que las señales obtenidas tras esta reconstrucción tendrán el mismo número de muestras que las señales originales. Estas señales serán representadas en forma de serie temporal de manera que se pueda estudiar y analizar el tráfico a lo largo del tiempo en distintas agregaciones temporales.
- **Cálculo de estadísticas:** Una vez obtenidas las señales reconstruidas, se compararán con las señales originales en base a las medidas de precisión estudiadas en el punto 3.3. Concretamente, se calculará el MSE y el MAPE y además, en base a los resultados obtenidos, se podrá elegir qué familia *wavelet* ofrece la precisión más óptima. Toda esta información ayudará a elegir la configuración del sistema más óptimo, que será implementado en Python.

4.3. Desarrollo y evaluación de la propuesta

Una vez descritos los distintos bloques del sistema, a continuación se expondrá el desarrollo de tres modelos distintos dentro de la propuesta planteada. El diagrama de bloques será común en los tres casos, variando los módulos de Transformada Wavelet, Almacenamiento de Coeficientes y Transformada Wavelet Inversa. En definitiva, se pretende estudiar la precisión de estos modelos en base a la utilización de los coeficientes de aproximación y detalle del Análisis Multi-Resolución teniendo presente el nivel de fiabilidad de la señal reconstruida y el nivel de compresión utilizado.

4.3.1. Primer modelo: Wavelet en desventaja respecto a MRTG

En primer lugar, se va a explicar el funcionamiento del primer modelo desarrollado. Este modelo se basa en la aplicación del Análisis Multi-Resolución a la señal de entrada, discriminando entre las agregaciones semanales, mensuales y anuales. El objetivo es el de utilizar un nivel de compresión lo más parecido posible al usado por MRTG y obtener en la medida de lo posible una señal reconstruida más parecida a la señal original.

Por tanto, para cada una de las agregaciones (semanal, mensual y anual) se debe utilizar un nivel de descomposición diferente, pues las señales de entrada presentan distinto número de muestras. En este primer modelo, únicamente se almacenan en la base de datos los coeficientes de aproximación del último nivel, descartando todos los de detalle. De esta forma, se reduce la información a almacenar. Por tanto, en la obtención de la señal reconstruida mediante el Análisis Multi-Resolución Inverso se establecen todos los coeficientes de detalle a cero, tal y como se puede observar en la Figura 4.5.

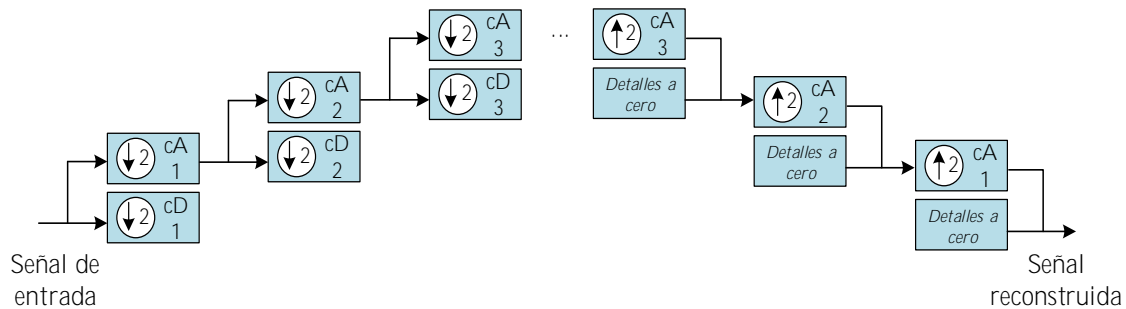


Figura 4.5: Primer modelo, reconstrucción con detalles a cero.

La elección de dicho nivel de descomposición es importante pues varía el resultado que se obtiene en la reconstrucción de la señal. Es decir, no se obtiene el mismo resultado descomponiendo la señal al nivel 1 que al nivel 9. Como se comentó en el fundamento teórico, el análisis Multi-Resolución realiza un sub-muestreo de orden 2 en cada uno de los niveles. Por lo tanto, la cantidad de coeficientes de aproximación que hay en el nivel 1 es mucho mayor que los del nivel 9. Esto significa, y como se puede ver en la Figura 4.6, que a medida que se aumenta de nivel la reconstrucción de la señal reconstruida va empeorando.

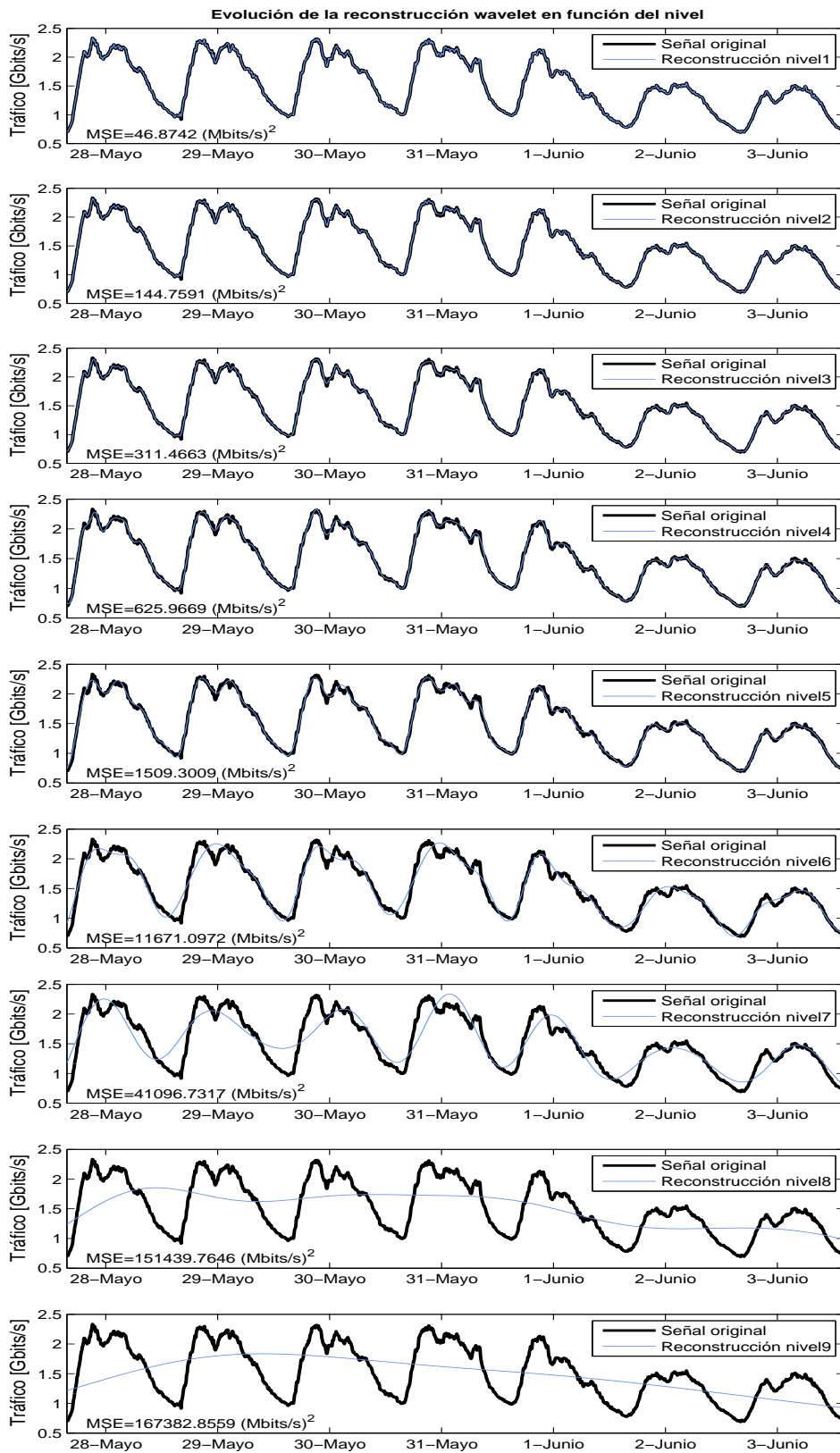


Figura 4.6: Reconstrucción de la señal original según el nivel de descomposición utilizado.

Sin embargo, cuanto menor sea el nivel de descomposición utilizado mayor será el número de coeficientes que deberán ser almacenados. Por tanto, uno de los principales retos de este proyecto es el de lograr un equilibrio óptimo entre el número de muestras almacenadas y la fiabilidad de la reconstrucción. Puesto que se tiene como base el MRTG, la elección del nivel será tal que el número de coeficientes de aproximación que se utilicen se iguale a las muestras almacenadas con esa herramienta. En otros trabajos, este límite se establece mediante un umbral, pero dado que el objetivo es implementar una herramienta mejorada lo más similar posible a MRTG, se utilizará el mismo almacenamiento.

A continuación se detalla, para las distintas agregaciones temporales, el nivel de descomposición utilizado:

- Agregación semanal: Como ya se ha comentado, una señal de una semana de duración presenta un total de 2016 muestras. La herramienta MRTG representa esta señal con 336 valores, por lo que se debe utilizar un nivel de descomposición que se ajuste a este valor. En este caso, se utiliza un nivel 3 pues resulta en 252 coeficientes de aproximación, lo cual se asemeja al valor usado por MRTG.
- Agregación mensual: Las señales mensuales presentan un total de 8064 muestras. En este caso, para ajustar el modelo a los 336 valores que almacena MRTG se ha decidido usar un nivel de descomposición 5, obteniéndose también 252 coeficientes de aproximación.
- Agregación anual: Con respecto a las señales anuales, cabe decir que tienen un total de 96768 muestras. En este caso, el nivel 8 de descomposición presenta un total de 378 coeficientes, mientras que el nivel 9 presenta un total de 189. En este caso, se ha decidido utilizar el nivel 9 de forma que se almacene un menor número de coeficientes que MRTG.

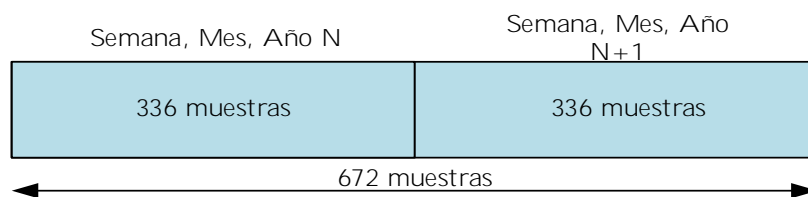


Figura 4.7: BBDD de MRTG.

Una vez que se ha elegido el nivel de descomposición a utilizar en cada caso, a continuación se estudiará cómo almacenar los datos resultantes. De cara a facilitar la evaluación de los tres modelos y la comparación con la precisión de MRTG, se ha decidido usar una base de datos diferente para cada agregación temporal. De esta forma, se tienen 3 bases de datos en las que se ha decidido almacenar como máximo 2 semanas, 2 meses y 2 años. A diferencia del comportamiento de MRTG en el que se realiza el promediado cada 6, 24 o 288 muestras, en estos modelos se necesitan 3 *buffers* de distinta capacidad.

El *buffer* semanal presenta una capacidad de 2016 muestras, el *buffer* mensual de 8064 muestras y el *buffer* anual de 96768 muestras. A medida que se completan estos buffers se aplicará el Análisis Multi-Resolución adecuado según la agregación correspondiente y se almacenarán los coeficientes de aproximación obtenidos en la base de datos siguiendo un comportamiento FIFO. En la Figura 4.7 y la Figura 4.8, se muestra un esquema de cómo se organiza la BB.DD en MRTG y en este primer modelo respectivamente.

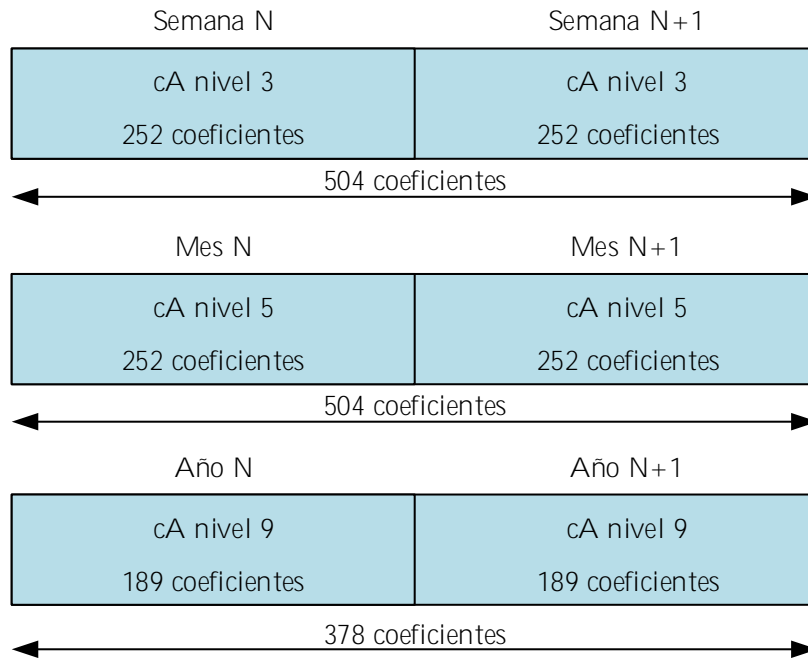


Figura 4.8: BBDD del primer modelo Wavelet.

En definitiva, en este modelo se plantea el uso de un menor número de coeficientes que los utilizados por MRTG. Los niveles de compresión a nivel porcentual para estas tres agregaciones se reflejan en la Tabla 4.1. Como se puede apreciar, el primer modelo está en desventaja con respecto a MRTG pues se está utilizando un menor número de coeficientes en todas las agregaciones temporales, sobre todo a nivel anual donde el nivel de compresión es de casi la mitad. En contraposición, se está consiguiendo un mejor *ratio* de compresión.

	Semanal (%)	Mensual (%)	Anual (%)
MRTG	16.666	4.166	0.347
Wavelet (primer modelo)	12.5	3.125	0.195

Tabla 4.1: *Ratios* de compresión para distintas agregaciones.

El siguiente paso consiste en seleccionar la familia *wavelet* que se va a utilizar para cada una de las agregaciones. En total, se ha realizado un estudio con 70 tipos de *wavelets* diferentes incluyendo todas aquellas de las familias Haar, Daubechies (db1 – db45), Symlets (sym2-sym20) y Coiflets (coif1-coif5). Este análisis, el cual se ha realizado con 5 señales originales como se puede ver en la Figura 4.9, ha consistido en calcular el Error Cuadrático Medio para cada una de las señales y para cada uno de los tipos de *wavelet*. A continuación se ha realizado el promedio de manera que se ha obtenido un MSE medio para cada familia, representado en la Figura 4.9 por una curva negra. La elección del tipo de *wavelet* a utilizar ha sido aquella con la que se ha obtenido un MSE medio menor. En particular, para este modelo se ha usado la Sym8 para agregación semanal y la Sym10 para agregación mensual y anual.

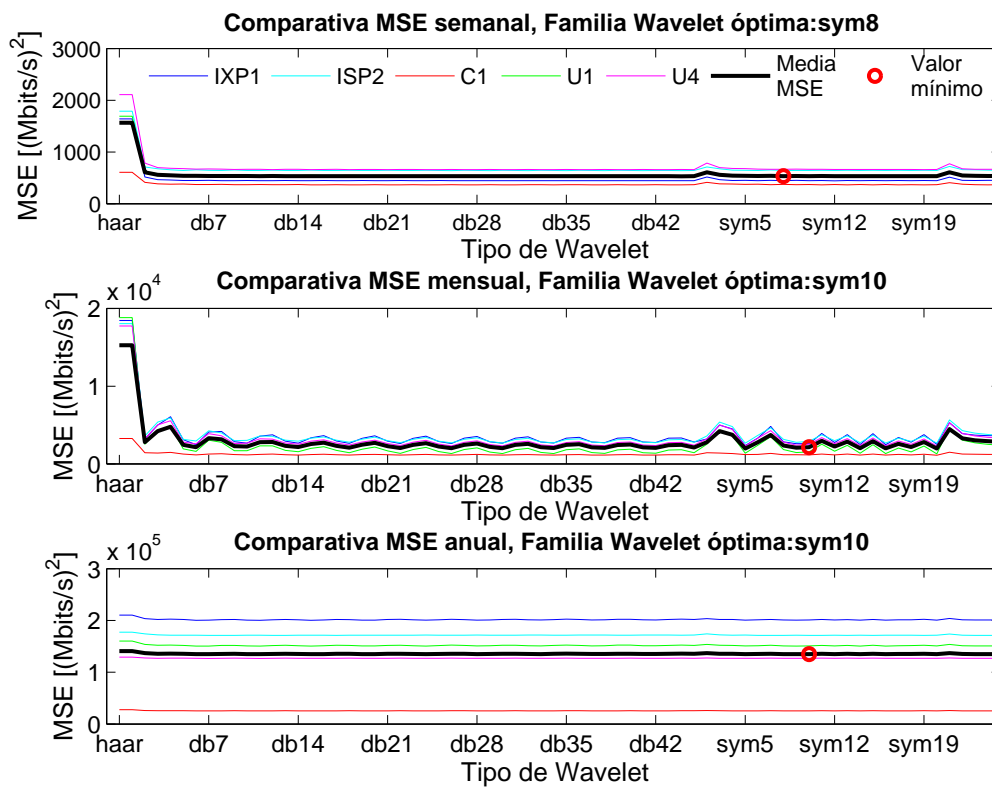


Figura 4.9: Medidas de precisión obtenidas tras aplicar MRTG a las señales de la Base de Datos.

Finalmente, se ha comprobado la eficiencia de este modelo, mediante la evaluación de las 14 señales de las que se dispone, tanto el tráfico entrante como el saliente de cada interfaz, y tomando un intervalo temporal de 118 semanas lo que equivale a 2 años 5 meses y 2 semanas de duración. A medida que se van completando los *buffers* para las distintas agregaciones, se han obtenido los coeficientes de aproximación y se han almacenado en sus bases de datos correspondientes. Además, se ha realizado la Transformada Wavelet Inversa para obtener las distintas señales reconstruidas. Por último, se ha calculado el MSE entre las distintas señales reconstruidas y las señales originales.

Para comparar el resultado obtenido, con el resultado del que se parte (correspondiente al uso de MRTG), se ha repetido este proceso simulando el comportamiento de MRTG con su proceso promediador simple. El resultado es la obtención de 118 valores de MSE a nivel semanal, 29 valores a nivel mensual y 2 valores a nivel anual. Tras realizar un promediado a estos resultados, se han recogido en forma de tabla en el Anexo I.

De cara a obtener una representación visual de los resultados, se ha calculado la mejora relativa del MSE que presenta este modelo con respecto a MRTG mediante la siguiente expresión:

$$Mejora_{relativa}(\%) = \frac{MSE_{MRTG} - MSE_{Wavelet}}{MSE_{MRTG}} \cdot 100 \quad (4.1)$$

En la Figura 4.10, se representan los resultados obtenidos tanto para el tráfico entrante como para el saliente de cada señal de entrada. A primera vista, se puede ver que el hecho de utilizar wavelets en lugar del proceso promediador simple de MRTG mejora algo el Error Cuadrático Medio en todas las señales.

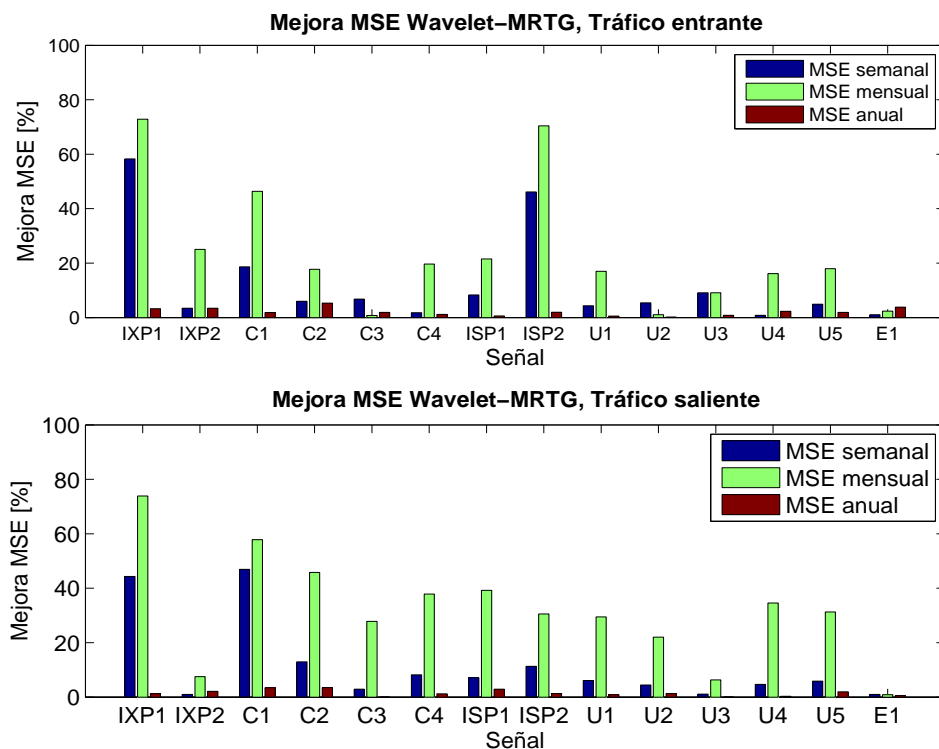


Figura 4.10: Mejora relativa del MSE % para el primer modelo.

Por otra parte, analizando los resultados obtenidos se puede ver que en la agregación mensual es la que presenta una mejora más considerable en la mayoría de los casos. Esto se debe a que el tráfico en las cuatro semanas que forman un mes presenta un comportamiento claramente diferenciado, como se observó en la Figura 3.3 del punto 3.1.

A nivel semanal, el tráfico en los fines de semana no sigue con la tendencia periódica de los días laborales. Dado que en este modelo se están utilizando únicamente los coeficientes de aproximación, la reconstrucción de la señal que se obtiene es básicamente la tendencia que sigue el tráfico. Por tanto, la mejora relativa del MSE a nivel semanal no es tan buena como la mejora relativa a nivel mensual. Por último, a nivel anual se obtiene la mejora relativa más baja en la mayoría de las señales debido a que el carácter periódico no es tan evidente entre los 11 meses del año (quitando el mes de Agosto).

Además, estudiando los resultados en cuanto al tráfico entrante, existe un conjunto de señales como es el caso de IXP1, C1 o ISP1 que presentan una mejora relativa considerablemente más grande que en el resto de señales. Para analizar este comportamiento, se ha realizado un estudio acerca de los volúmenes de tráfico medido en Tbits para las 118 semanas. Tal y como se refleja en la Tabla 4.2, estas tres señales corresponden con aquellas que presentan un mayor volumen de tráfico.

Señal	Tráfico entrante (Tbits)	Tráfico saliente (Tbits)
IXP1	184.087	463.773
IXP2	2.758	2.741
C1	175.788	75.770
C2	11.047	18.764
C3	2.620	6.028
C4	53.905	71.799
ISP1	9.456	13.216
ISP2	137.910	122.380
U1	2.794	8.299
U2	1.541	4.272
U3	2.062	5.173
U4	21.0613	16.625
U5	5.570	9.724
E1	145.714	114.330

Tabla 4.2: Volúmenes de tráfico entrante y saliente para cada señal en las 118 semanas.

Este resultado es lógico, pues cuanto mayor es el volumen de tráfico más importante será la componente frecuencial periódica presente en la señal. En relación con este hecho, algunos estudios previos han determinado que la mayoría del tráfico en Internet es generado por un pequeño porcentaje de usuarios conocidos como *heavy-hitters* [18]. Mediante el uso de aplicaciones que requieren un gran ancho de banda, estos usuarios introducen picos de tráfico en determinadas horas del día que pueden provocar alteraciones en el patrón del tráfico. En el caso concreto de este entorno de trabajo, el estudio revela que los *heavy-hitters* están homogéneamente repartidos entre el conjunto de universidades y centros de investigación de la RedIRIS [18]. Es evidente que cuanto mayor es el volumen total agregado de tráfico en un enlace, menor es el impacto de los *heavy-hitters* en la tendencia periódica del tráfico y viceversa.

Tras haber analizado los resultados de este primer modelo, se puede llegar a la conclusión que las señales reconstruidas presentan un Error Cuadrático Medio mejor que el obtenido por la herramienta MRTG, a pesar de almacenar un menor número de coeficientes. Sin embargo, en la mayoría de los casos los resultados son mejorables ya que entre otras cosas el primer modelo está en desventaja respecto al almacenamiento de coeficientes, con lo que se ha trabajado en un segundo modelo que se detalla a continuación.

4.3.2. Segundo modelo: Wavelet en ventaja respecto a MRTG

En este caso, se plantea un segundo modelo para mejorar los resultados obtenidos del punto 4.3.1. Principalmente, el cambio con respecto al primer modelo viene dado por el uso de los coeficientes de detalle. En lugar de almacenar únicamente los coeficientes de aproximación del último nivel de descomposición, en este modelo también se almacenan aquellos coeficientes de detalle que más energía presentan. Por tanto, a diferencia del primer modelo, existen ciertos coeficientes de detalle que no nulos. El esquema de descomposición del Análisis Multi-Resolución directo e inverso para este modelo puede apreciarse en la Figura 4.11.

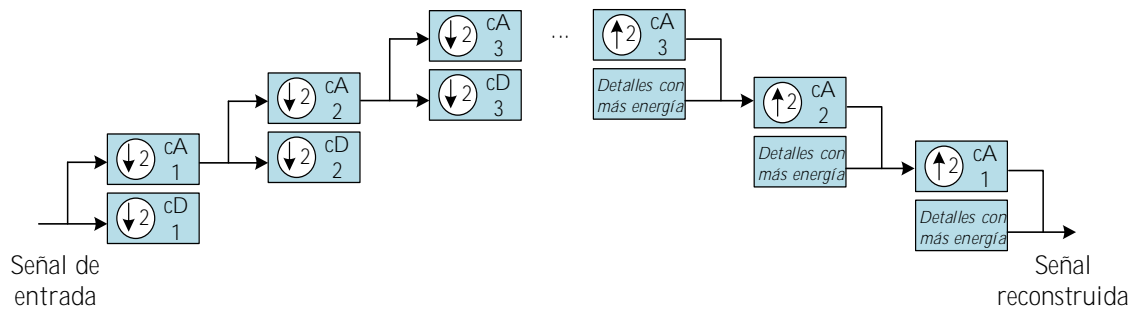


Figura 4.11: Segundo modelo, reconstrucción con coeficientes de detalle.

El proceso de selección de los coeficientes de detalle ha sido el siguiente. Dado que MRTG utiliza 336 muestras para las distintas agregaciones, se toman tantos coeficientes de detalle como sean necesarios para igualar esta cifra. Por tanto, a nivel semanal y mensual se utilizan los 84 coeficientes de detalle que más energía tienen, mientras que a nivel anual se incluyen 147 coeficientes. El esquema de la BBDD de este modelo puede apreciarse en la Figura 4.12.

Cabe destacar que en este modelo se necesitan almacenar también los índices que los coeficientes de detalle ocupan dentro del vector total de coeficientes. Esto es necesario para poder reconstruir la señal original, pues la función *waverec()* debe tener como entrada un vector ordenado de coeficientes. Por tanto, todas las posiciones del vector de coeficientes de detalle serán nulas, excepto aquellas con mayor energía. Con respecto a la BBDD, se ha decidido almacenar los coeficientes de detalle junto con los de aproximación por un lado y los índices de los coeficientes de detalle en una pequeña base de datos aparte. En realidad, como se puede comprobar, este modelo está en ventaja con respecto a MRTG pues se está utilizando un mayor almacenamiento.

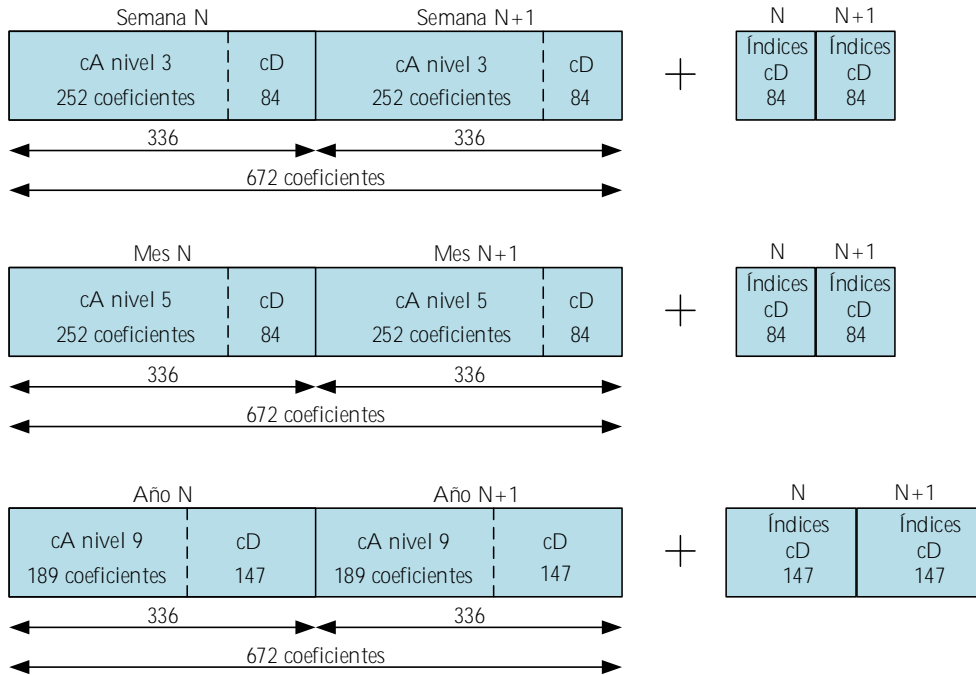


Figura 4.12: BBDD del segundo modelo Wavelet.

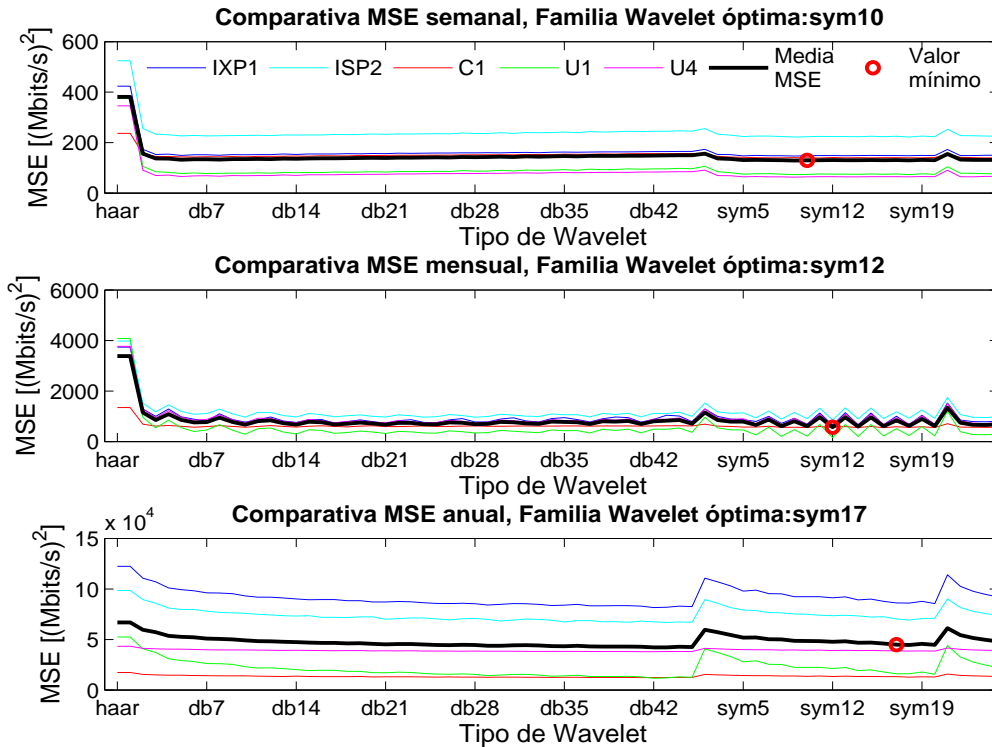


Figura 4.13: . Estudio del mejor tipo de *wavelet* a nivel semanal, mensual y anual para el segundo modelo.

En cuanto a la elección de la familia *wavelet* se ha repetido el proceso seguido en el primer modelo. En este caso, tal y como se puede apreciar en la Figura 4.13, los mejores resultados proceden también de la familia Symlet. Tras realizar el promedio del MSE en 6 señales diferentes y encontrar el valor mínimo, se ha decidido utilizar en este caso la Sym10 a nivel semanal, la Sym12 a nivel mensual y la Sym17 a nivel anual.

A continuación se recogen los resultados obtenidos para este segundo modelo. Tal y como se puede ver en la Figura 4.14, el hecho de incluir algunos coeficientes de detalle provoca que la mejora relativa con respecto a MRTG sea mucho mayor que en el primer modelo. Este hecho es lógico, pues se está dotando de más información al Análisis Multi-Resolución Inverso con lo que esta herramienta es capaz de obtener una señal reconstruida que se asemeja en mayor proporción a la señal original.

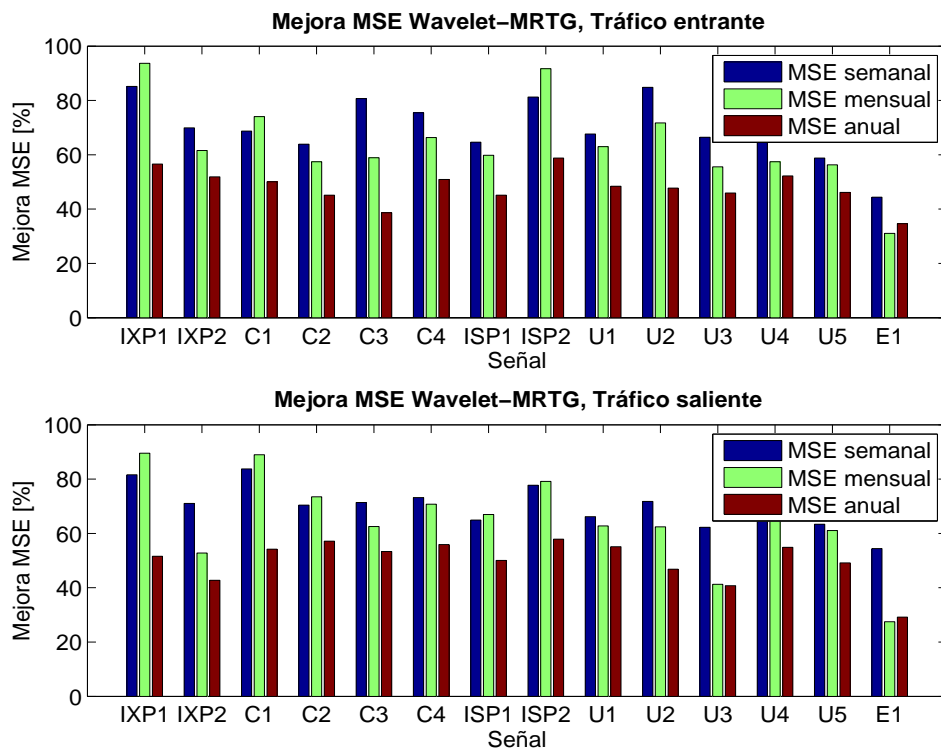


Figura 4.14: Mejora relativa del MSE % para el segundo modelo.

Además, esta mejora es evidente en la totalidad de las señales de estudio independientemente del volumen total de tráfico agregado y del nivel de agregación (semanal, mensual y anual). Sin embargo, sí se puede notar que en aquellas señales con más tráfico agregado se obtienen mejores resultados. Por otro lado, la Figura 4.14 refleja también que a nivel semanal y mensual se aprovecha mejor el carácter periódico presente en las señales que a nivel anual. Como ya se ha comentado, la periodicidad entre meses no es tan fuerte como a nivel semanal o diario.

4.3.3. Tercer modelo: Wavelet y MRTG en igualdad

El tercer modelo desarrollado surge como necesidad de igualar las condiciones entre MRTG y Wavelet. Como se ha visto al analizar el desarrollo del segundo modelo en el punto 4.3.2, además de almacenar los coeficientes de detalle que más energía presentan también es necesario almacenar sus índices para poder realizar la reconstrucción de la señal. Por tanto, en este caso se plantea un modelo basado en la misma filosofía pero que almacena exactamente los mismos coeficientes que MRTG (en concreto 336) a nivel semanal y mensual y 335 coeficientes a nivel anual. El esquema de la BBDD de este tercer modelo se recoge en la Figura 4.15. En este caso, existe únicamente una base de datos por cada nivel de agregación. Por tanto, la cantidad de coeficientes de detalle que se pueden almacenar se reduce a la mitad.

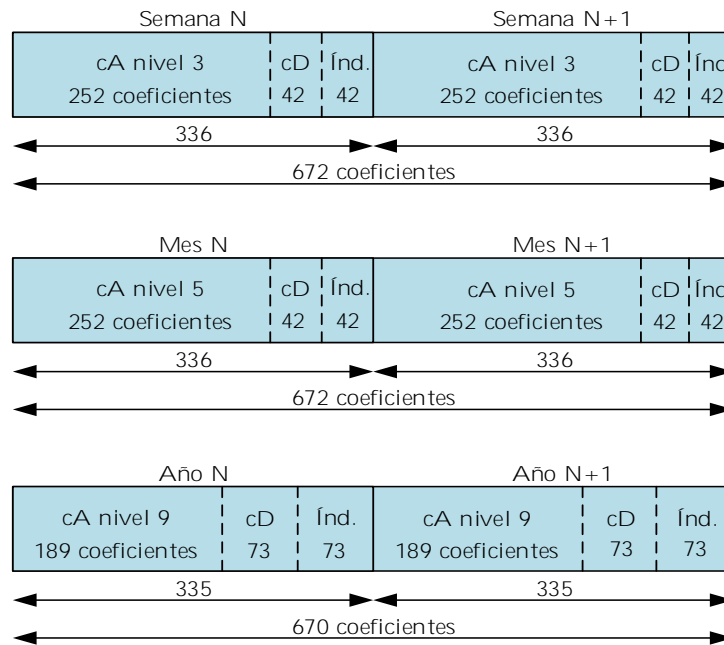


Figura 4.15: BBDD del tercer modelo Wavelet.

Antes de analizar los resultados de este modelo, cabe pensar que la mejora relativa en cuanto al MSE no va a ser tan buena como en el modelo anterior. Sin embargo, este es el modelo más justo tanto para MRTG como para Wavelet.

Al igual que en el primer y segundo modelo, se ha realizado un estudio (Figura 4.16) para determinar el tipo de *wavelet* que mejor Error Cuadrático Medio obtiene con respecto a la señal original. En este caso, se ha decidido utilizar la familia Symlet también, en concreto se ha elegido la Sym10 a nivel semanal, la Sym12 a nivel mensual y la Sym17 a nivel anual dado que son aquellas que mejor precisión han ofrecido.

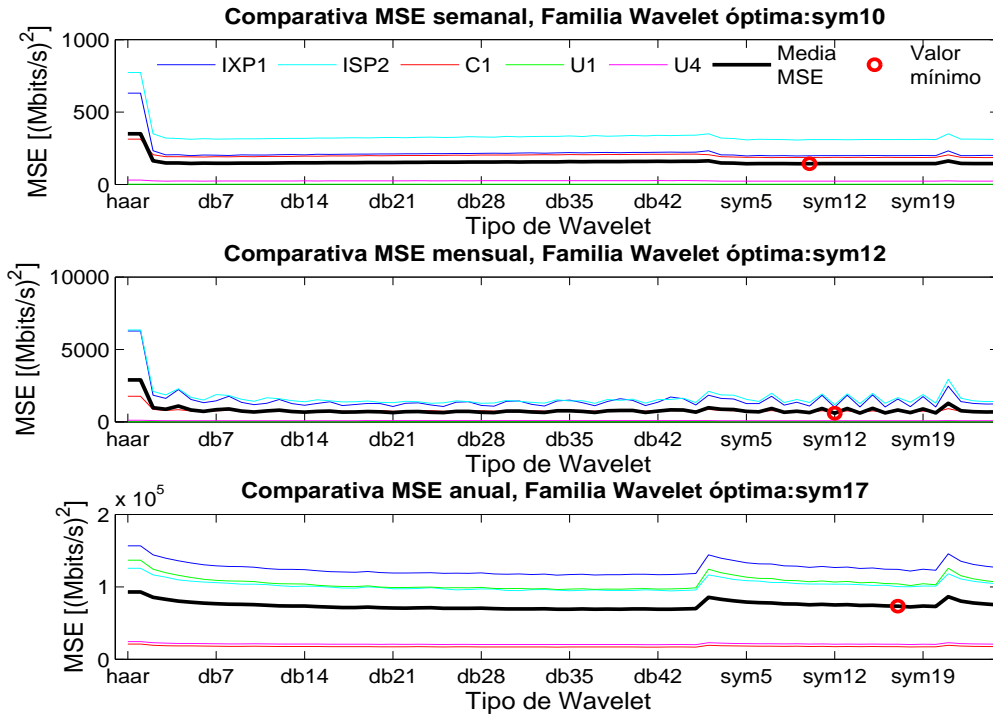


Figura 4.16: Estudio del mejor tipo de *wavelet* a nivel semanal, mensual y anual para el tercer modelo.

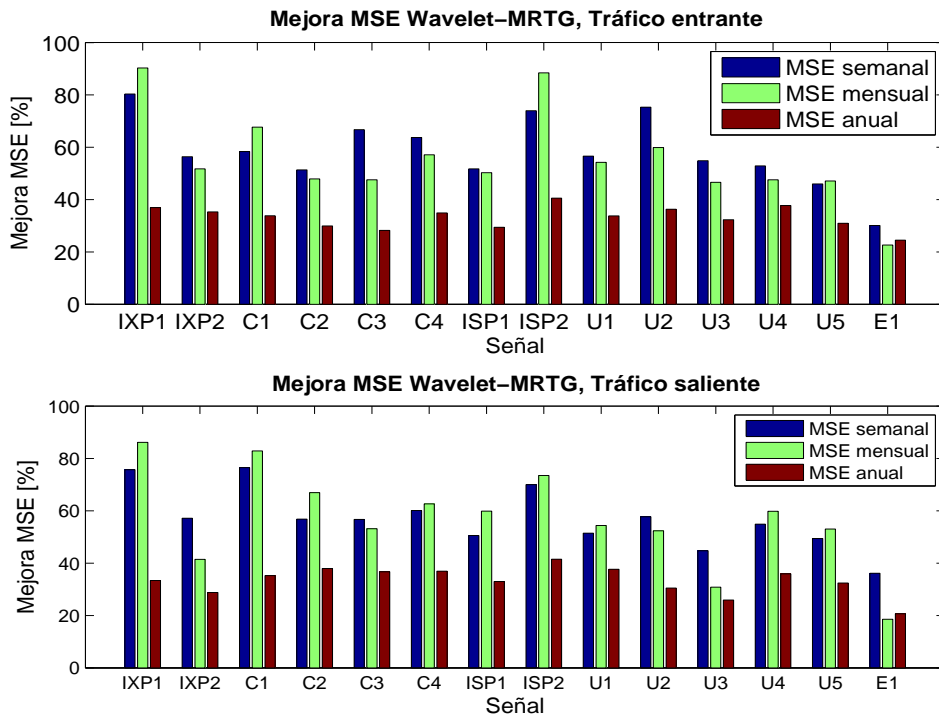


Figura 4.17: Mejora relativa del MSE % para el tercer modelo.

Analizando los resultados de la Figura 4.17, se puede ver que el MSE obtenido tras aplicar este modelo presenta una mejora relativa bastante buena con respecto al que ofrece MRTG. Al igual que ocurre en el segundo modelo, el hecho de utilizar algunos coeficientes de detalle, en concreto los que más energía tienen, hace que la señal reconstruida se ajuste más a la señal original. Además, se requiere exactamente el mismo almacenamiento que MRTG, con lo que ambos sistemas presentan el mismo nivel de compresión.

Además, dado que este es el modelo definitivo, se ha calculado también el MAPE para el tráfico entrante de los interfaces de todas las señales de trabajo. El objetivo es el de comprobar la mejora en cuanto al nivel de reconstrucción utilizando otra medida de precisión diferente al Error Cuadrático Medio.

En este caso, se ha calculado el MAPE resultante para las distintas agregaciones temporales tanto en MRTG como en este tercer modelo y se ha representado la diferencia en la Figura 4.18. Se puede ver cómo en todas las señales se produce una mejora del MAPE (valores más bajos), lo que implica directamente que el modelo propuesto presenta un mejor nivel de precisión en cuanto a la señal reconstruida utilizando el mismo almacenamiento que MRTG. Además, al igual que ocurre con el MSE, se comprueba que el MAPE es mayor cuanto mayor es el nivel de agregación. Sin embargo, en este caso se obtienen valores superiores al 100 %, sobre todo a nivel anual. Este hecho se debe a que, como ya se comentó en el punto 3.3, el MAPE no tiene un límite superior para valores predichos muy altos.

Por otro lado, de cara a observar el comportamiento de este modelo y compararlo con el de MRTG, en la Figura 4.19 se ha representado un ejemplo correspondiente a un fragmento de una de las señales reales disponibles para este estudio. Concretamente, se ha representado el tráfico real entrante a nivel mensual en Gbits/s de una interfaz de red. Además, sobre la misma gráfica se pueden ver las señales reconstruidas que obtiene la herramienta MRTG y Wavelet (utilizando la configuración de este tercer modelo).

Tal y como se puede observar en la Figura 4.19, la reconstrucción al utilizar wavelets es bastante mejor que la obtenida por MRTG. De hecho, debido al comportamiento característico de MRTG, se obtiene una señal en forma de escalón. En este caso, dado que se está trabajando con una agregación mensual, el cada escalón tiene un total de 24 muestras, o lo que es lo mismo 2 horas de duración. Por otro lado, la señal reconstruida tras aplicar el Análisis Multi-Resolución Inverso no presenta dichos escalones, sino que se ajusta de manera óptima a la forma de la señal real. Como es lógico, existen numerosos detalles de la señal real que no son reconstruidos, pero es el precio que tiene asociado la compresión de datos. Sin embargo, mediante este ejemplo se pone de manifiesto el buen equilibrio existente alcanzado entre la pérdida de información de la señal real y el nivel de compresión.

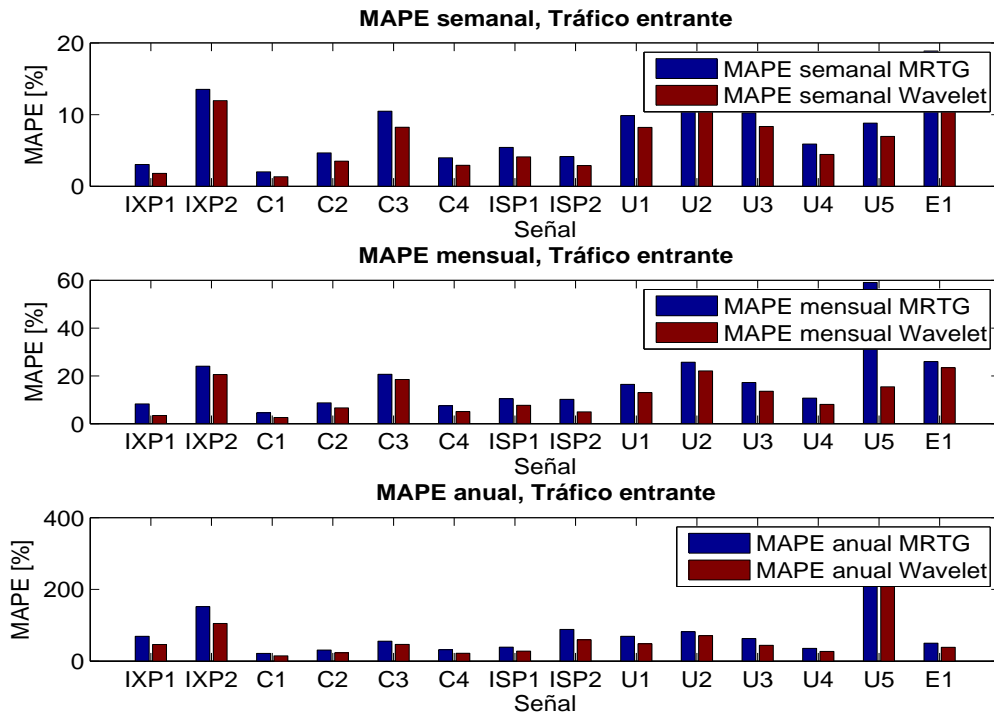


Figura 4.18: MAPE obtenido para MRTG y el tercer modelo Wavelet.

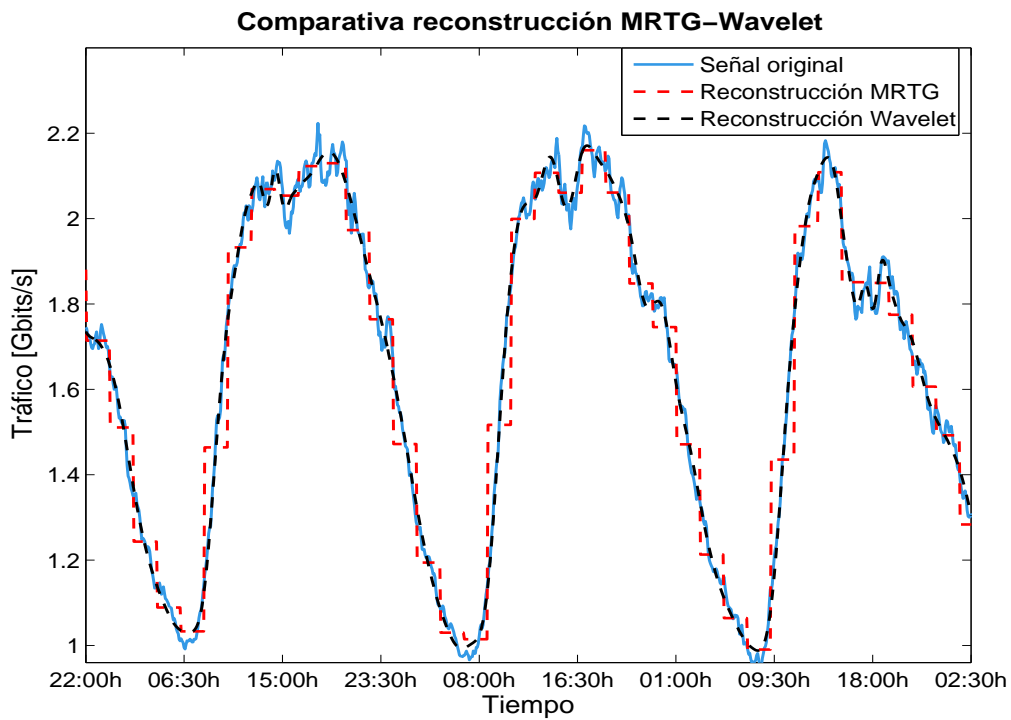


Figura 4.19: Comparativa de MRTG y Wavelet con respecto a la señal original.

Cabe destacar que si se requiriese obtener una mejor reconstrucción de la señal original bastaría con aumentar el número de coeficientes de aproximación y detalle usados. De hecho, cuanto menor sea el nivel de descomposición utilizado en el Análisis Multi-Resolución, mejor será dicha reconstrucción. Por tanto, otra de las ventajas de la Transformada Wavelet es su capacidad para adaptarse a los requisitos en cada caso de forma rápida y sencilla.

Por último, una vez evaluados los tres modelos desarrollados se ha llegado a la conclusión que este tercer modelo es el más justo con respecto a MRTG. Se ha podido comprobar que, utilizando el mismo nivel de compresión que esta herramienta, se obtienen unos resultados de mejora significativos en cuanto al nivel de reconstrucción de la señal original. Por tanto, el siguiente paso es el de implementar este modelo en forma de aplicación de forma que pueda ser utilizado por la comunidad de usuarios. En este caso, la herramienta se denominará *WaveTG* y será desarrollada en el lenguaje de programación Python. En el capítulo 5, se recogerán todos los detalles acerca de esta implementación.

Capítulo 5

Implementación de la herramienta *WaveTG*

EN este capítulo se analizarán los detalles acerca de la implementación del tercer modelo desarrollado en el capítulo 4 en el lenguaje de programación Python. Se ha decidido utilizar este lenguaje de programación dada su universalidad y enorme uso en la actualidad (primera posición en relevancia para los lectores de IEEE Spectrum [19]). Además, se pretende evitar lenguajes propietarios como Matlab y tener en cuenta que el lenguaje Perl utilizado por MRTG está, progresivamente, perdiendo relevancia.

El objetivo, una vez analizados los resultados obtenidos en Matlab, es el de crear una herramienta de representación gráfica lo más parecida posible a MRTG, pero usando la Transformada Wavalet con la configuración propia del tercer modelo. Por tanto, todas las gráficas se ajustarán al formato de visualización de esta herramienta. De esta forma, las gráficas se generan primeramente en formato PNG y posteriormente se incluyen dentro de un archivo HTML refrescado de forma dinámica. Además, al igual que ocurre en MRTG, se han calculado una serie de estadísticas asociadas a cada gráfica diferenciadas por el tráfico entrante y saliente:

- Valor máximo del tráfico en bits/segundo.
- Valor mínimo del tráfico en bits/segundo.
- Valor promedio del tráfico en bits/segundo.

Cabe destacar que además de las gráficas a nivel semanal, mensual y anual, se ha creado también una gráfica correspondiente al tráfico real en ambos sentidos, procedente directamente de los equipos de red que disponen de la herramienta MRTG instalada. En la Figura 5.1, puede verse un ejemplo de esta gráfica.

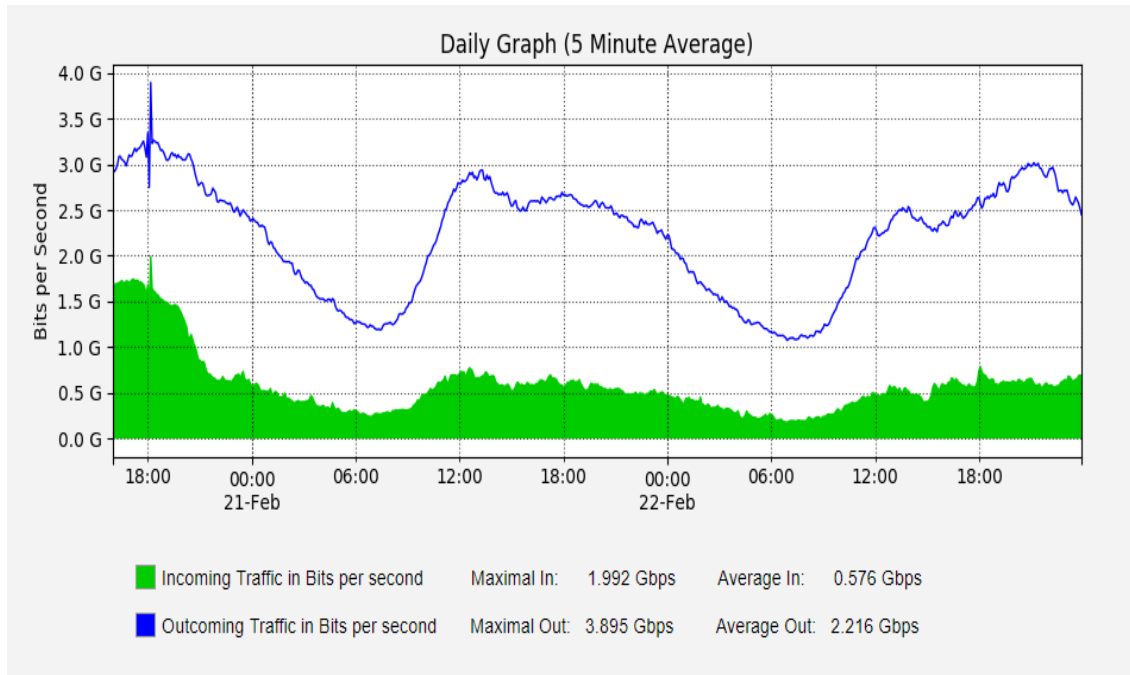


Figura 5.1: Tráfico real entrante y saliente representado cada 5 minutos en *WaveTG*.

En lo que respecta al desarrollo de esta herramienta, cabe decir que se ha hecho uso de un conjunto de librerías de Python:

- *Numpy*: Se han utilizado funciones matemáticas de alto nivel para operar con vectores y matrices.
- *Matplotlib*: Esta librería se ha usado para la generación de gráficos a partir de datos almacenados en vectores.
- *Pywt*: Mediante esta librería, se han utilizado las funciones para realizar el Análisis Multi-Resolución Directo e Inverso de la Transformada Wavelet. Concretamente, se han utilizado las funciones *wavedec()* y *wavedec()* respectivamente.
- *Pandas*: Esta librería se ha utilizado para el análisis de los datos en forma de series temporales dada su flexibilidad y la posibilidad de trabajar con los datos de forma eficiente.

Por otro lado, el funcionamiento de esta herramienta es la explicada en el tercer modelo. Los datos reales correspondientes al tráfico de entrada y salida se van almacenando en *buffers* de tamaño semanal (2016), mensual (8064) y anual (96768) de forma que cuando se completen, se aplica la Transformada Wavelet y se almacenan los coeficientes de aproximación, los coeficientes de detalles y los índices de estos últimos. Una vez se tienen estos coeficientes, se calculan las estadísticas del tráfico y se generan las gráficas correspondientes en formato PNG, refrescando de esta forma la página web. Cabe destacar que, dado que existe una gráfica con datos reales cada 5 minutos, el archivo HTML se refresca también con esta periodicidad.

En cuanto al análisis Multi-Resolución, en la Tabla 5.1 se recoge un resumen de las distintas configuraciones a nivel semanal, mensual y anual:

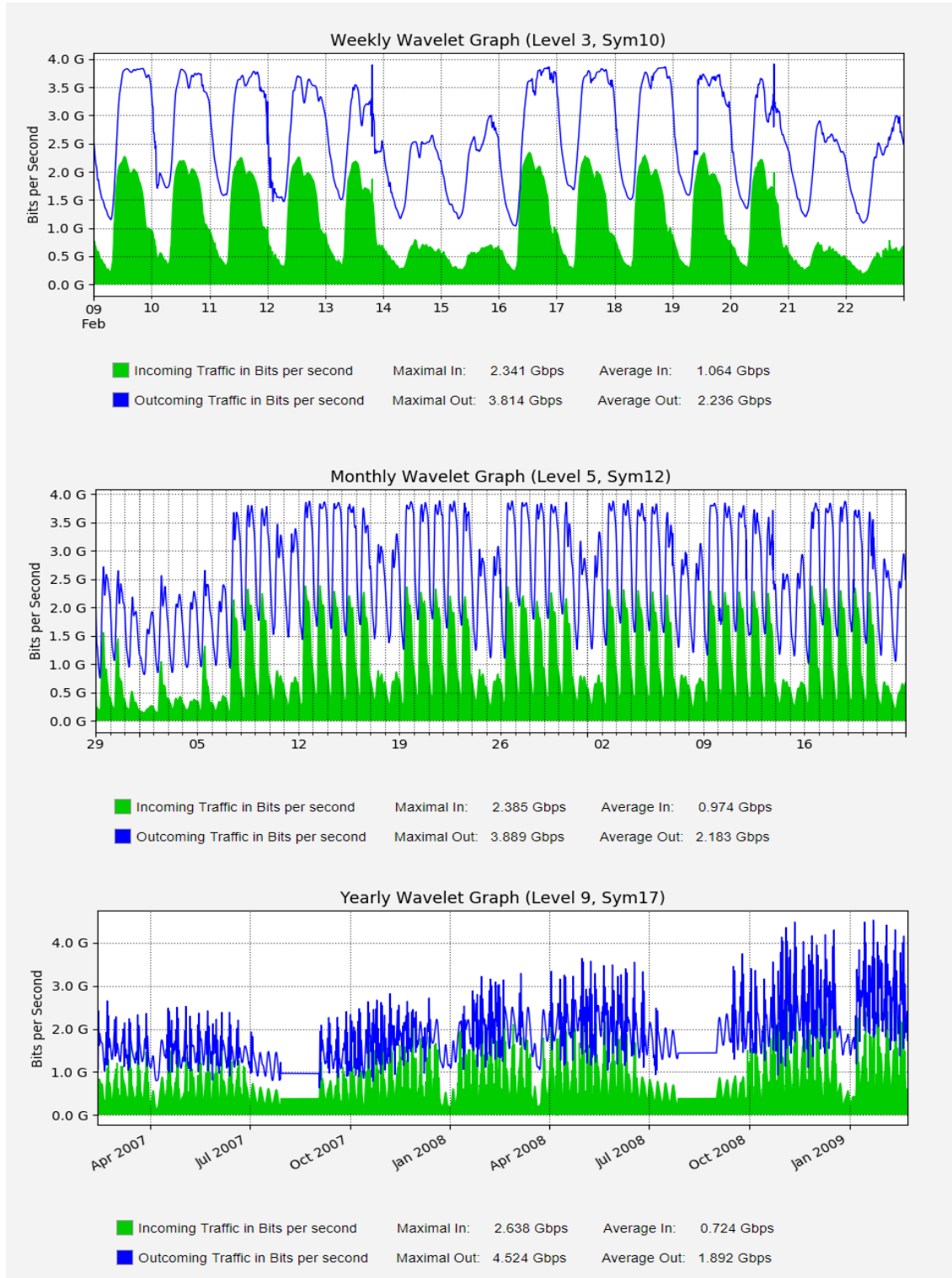


Figura 5.2: Ejemplo de la página web creada por *WaveTG*.

Agregación	Nivel de descomposición	Tipo de Wavelet	Nº coef. de aproximación	Nº coef de detalle	Índices
Semanal	3	Symlet10	252	42	42
Mensual	5	Symlet12	252	42	42
Anual	9	Symlet17	189	73	73

Tabla 5.1: Resumen de la configuración de las distintas agregaciones.

De cara a la visualización de la página web obtenida, en la Figura 5.2 se muestra un ejemplo. En este caso, se ha representado el tráfico entrante y saliente de las señales reconstruidas correspondiente a los dos últimos años, los dos últimos meses y las dos últimas semanas:

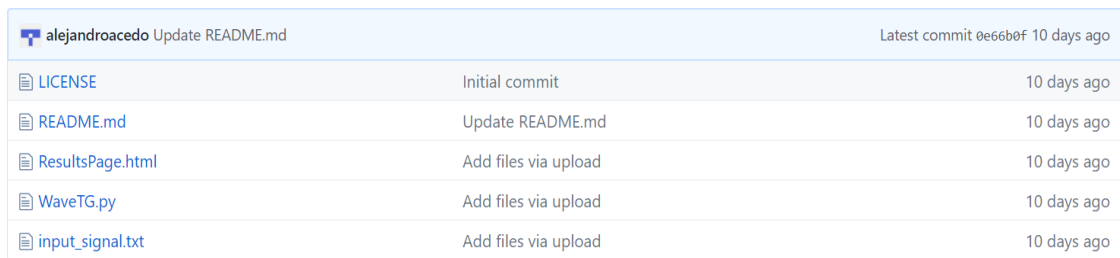
- Gráfica semanal: 09 Febrero 2009, 00:00:00 h – 22 Febrero 2009, 23:55:00 h
- Gráfica mensual: 29 Diciembre 2008, 00:00:00 h – 22 Febrero 2009, 23:55:00 h
- Gráfica anual: 26 Febrero 2007, 00:00:00 h – 22 Febrero 2009, 23:55:00 h

Como dato, añadir que en la gráfica a nivel anual, el mes de Agosto no presenta datos pues como ya se ha comentado anteriormente se han eliminado todas las muestras correspondientes a este mes de todas las señales disponibles.

Para finalizar, cabe decir que mediante *WaveTG* los gestores de red pueden supervisar el tráfico de los distintos interfaces a distintas granularidades temporales, obteniendo una mejor resolución más pequeña a medida que los datos son más recientes y viceversa. De esta forma, se pueden ver claramente las tendencias que sigue el tráfico, si ha experimentado comportamientos anómalos y ayuda a realizar una gestión eficiente en cuanto a la planificación de la capacidad de la red.

Para finalizar con este capítulo, cabe resaltar que se ha subido a la plataforma GitHub el código desarrollado bajo la licencia GNU General Public License v3.0. En dicha plataforma se indican los pasos a seguir para ejecutar el script de Python, concretamente en el fichero README.md mostrado en la Figura 5.3.

En realidad, el código subido no es más que un ejemplo en el que se ha utilizado como entrada una de las señales disponibles para este proyecto. Tras aplicar la transformada Wavelet para las distintas agregaciones (semanal, mensual y anual) se generan las gráficas en formato PNG y se actualiza la página web denominada ResultsPage.html con dichas gráficas.



The image shows a screenshot of a GitHub repository directory listing. The repository name is 'alejandroacedo' and the latest commit is '0e66b0f' from '10 days ago'. The directory listing includes the following files and their commit details:

File Name	Commit Message	Time
LICENSE	Initial commit	10 days ago
README.md	Update README.md	10 days ago
ResultsPage.html	Add files via upload	10 days ago
WaveTG.py	Add files via upload	10 days ago
input_signal.txt	Add files via upload	10 days ago

Figura 5.3: Directorio de ficheros subidos a GitHub.

El proyecto, bajo la licencia GNU GENERAL PUBLIC LICENSE v3, se puede encontrar en el siguiente enlace:

<https://github.com/alejandroacedo/WaveTG>

Capítulo 6

Conclusiones y vías futuras

EN este capítulo se expondrán las conclusiones del proyecto. En primer lugar, se analizarán los aspectos más relevantes del estudio realizado en el capítulo 4 y el desarrollo del modelo final del capítulo 5. Finalmente, en la última parte de este capítulo se pondrán algunas vías futuras de investigación.

6.1. Conclusiones

El objetivo principal del proyecto era la implementación de una versión mejorada de la herramienta de monitorización MRTG en la que se sustituyese el proceso promediador simple que esta herramienta utiliza para agregar los datos, por un proceso en el que se aplicara la Transformada Wavelet y aprovechara el carácter periódico de las señales de entrada. Además, esta nueva versión debía ser capaz de perder la mínima información de monitorización posible al reconstruir las señales originales y mantener un buen nivel de compresión, no almacenando más datos que en MRTG. Tras la realización de este proyecto se puede ver que se ha cumplido con el objetivo propuesto.

Concretamente, se ha desarrollado una aplicación en Python cuyo funcionamiento es similar al de MRTG, leyendo los datos de entrada y generando informes en formato HTML con gráficas que representan la carga de tráfico de los enlaces de red a lo largo del tiempo. Para lograr implementar esta aplicación de forma satisfactoria, previamente se han desarrollado 3 modelos distintos con distintas configuraciones. Todos los modelos comparten el mismo procedimiento, variando la forma en la que se aplica la Transformada Wavelet, en concreto el Análisis Multi-Resolución.

El primer modelo utiliza únicamente coeficientes de aproximación para representar las señales originales. El modelo está en desventaja con respecto a MRTG pues se está utilizando un menor almacenamiento. Como conclusión de este modelo, cabe decir que la mejora relativa del MSE es significativa en todas las señales. Por tanto, el hecho de utilizar wavelets ya mejora de por sí los resultados de MRTG. Además, los resultados obtenidos también reflejan que cuanto más periódico es el patrón del tráfico, mejor es la reconstrucción de la señal original. Esto suele pasar en señales que presentan un gran

volumen de tráfico, como por ejemplo IXPs, donde el tráfico tiene una tendencia periódica claramente diferenciada. En estos casos, la presencia de los denominados *heavy-hitters* tiene una menor relevancia que en señales con menor volumen de datos.

Por otro lado, el segundo modelo mejora de forma muy considerable el Error Cuadrático Medio dada la utilización de los coeficientes de detalle con más energía. En este caso, la mejora es global a todas las señales independientemente del volumen total. Sin embargo, un mejor comportamiento periódico del tráfico es directamente proporcional a un menor MSE, debido a que se obtiene una reconstrucción más fiable de la señal. En definitiva, este modelo ha presentado unos resultados muy convincentes pero se encuentra en ventaja con respecto a MRTG al almacenar los índices de los coeficientes de detalle en una base de datos aparte a la de los propios coeficientes. Por tanto, surge el tercer y último modelo.

El tercer modelo mantiene la misma filosofía del segundo modelo, pero almacena en una base de datos común los coeficientes de aproximación, de detalle y los índices de estos últimos. De esta forma, se almacenan exactamente los mismos valores en MRTG a nivel semanal, mensual y anual. Las señales reconstruidas tras realizar el Análisis Multi-Resolución Inverso se ajustan mucho más a las señales reales, con lo que el MSE obtenido es más bajo. La gran potencialidad de la Transformada Wavelet viene dada porque estos resultados tan buenos se consiguen sin disminuir el nivel de compresión utilizado por MRTG. Por tanto, además de perder la mínima información posible de la señal, se comprime de forma que no se desbordan las bases de datos. Por tanto, se puede decir que se ha conseguido obtener un buen equilibrio entre la compresión de los datos y pérdida de información. Dado que este tercer modelo ha sido el más convincente, es el que finalmente se ha implementado en Python bajo el nombre de WaveTG.

Por otro lado, cabe también destacar la buena adaptabilidad que ofrece la Transformada Wavelet a la casuística concreta, en concreto el Análisis Multi-Resolución. Según requiera la situación, esta herramienta permite obtener señales reconstruidas más o menos parecidas a las originales simplemente con el hecho de incluir un mayor/menor número de coeficientes de detalle o estableciendo un nivel de descomposición más bajo/alto, a costa claramente del nivel de compresión.

Por último, en cuanto al desarrollo de la herramienta WaveTG en Python cabe decir que se ha conseguido obtener una aproximación muy similar a MRTG en cuanto al formato de la página web se refiere. De hecho, las gráficas en PNG se generan dinámicamente siguiendo una representación muy parecida a la usada por MRTG. Esto, junto con la generación de estadísticas acompañando a las gráficas hacen de WaveTG una aplicación de código libre completa.

6.2. Vías futuras

Tras la realización de este proyecto, se han abierto una serie de líneas de investigación que pueden ser interesantes de cara a un estudio futuro:

- Validación del tercer modelo propuesto mediante un conjunto más amplio de series temporales reales relativas a medidas de red (latencia, *jitter*, etc.). Así mismo, también se plantea testear el modelo con señales periódicas de distinta naturaleza como puede ser el caso de las señales médicas.
- Repetición del estudio realizado utilizando únicamente datos relativos a días laborales, eliminando así los días festivos y fines de semana. Evaluación y comparativa de los resultados con los obtenidos en este proyecto.
- Desarrollo en lenguaje Perl de un módulo compatible directamente en la propia herramienta MRTG, haciendo uso de la Transformada Wavelet de forma similar a como se ha hecho para *WaveTG*.

Referencias bibliográficas

- [1] ACETO, GIUSEPPE, ET AL. Efficient storage and processing of high-volume network monitoring data. *IEEE Transactions on Network and Service Management*, 2013, vol. 10, no 2, p. 162-175.
- [2] GARCÍA-DORADO, JOSÉ LUIS, ET AL. On the duration and spatial characteristics of Internet traffic measurement experiments. *IEEE Communications Magazine*, 2008, vol. 46, no 11, p. 148-155.
- [3] OETIKER, TOBIAS; RAND, DAVE. MRTG: The Multi Router Traffic Grapher. En *LISA*. 1998. p. 141-148
- [4] MATA, FELIPE; GARCÍA-DORADO, JOSÉ LUIS; ARACIL, JAVIER. Detection of traffic changes in large-scale backbone networks: The case of the Spanish academic network. *Computer Networks*, 2012, vol. 56, no 2, p. 686-702.
- [5] SANTAMARÍA, FRANCISCO; CORTÉS, CAMILO A.; ROMÁN, FRANCISCO J. Uso de la transformada de ondeletas (Wavelet Transform) en la reducción de ruidos en las señales de campo eléctrico producidas por rayos. *Información tecnológica*, 2012, vol. 23, no 1, p. 65-78.
- [6] BLASCO, SILVIA. Técnicas de umbralización orientadas a la implementación de sistemas portátiles de monitorización de electrocardiogramas. 2007
- [7] DAUBECHIES, INGRID. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 1990, vol. 36, no 5, p. 961-1005.
- [8] CHUN-LIN, LIU. A tutorial of the wavelet transform. *NTUEE*, Taiwan, 2010.
- [9] http://ceres.ugr.es/~alumnos/sigpro/images/cap4_img.74.jpg
- [10] LAMBERT, MICHAEL H. A model for common operational statistics. RFC 1857. 1995.
- [11] GARCIA-DORADO, JOSE LUIS, ET AL. A queueing equivalent thresholding method for thinning traffic captures. En *IFIP/IEEE Network Operations and Management Symposium (NOMS)*, 2008, p. 176-183.
- [12] MUELAS, DAVID, ET AL. Application of functional feature extraction to the compression of network time series. En *IFIP/IEEE Integrated Network and Service Management (IM)*, 2017, p. 592-595.

REFERENCIAS BIBLIOGRÁFICAS

- [13] MATA, FELIPE; GARCÍA-DORADO, JOSÉ LUIS; ARACIL, JAVIER. Detection of traffic changes in large-scale backbone networks: The case of the Spanish academic network. *Computer Networks*, 2012, vol. 56, no 2, p. 686-702.
- [14] THOMPSON, KEVIN; MILLER, GREGORY J.; WILDER, RICK. Wide-area Internet traffic patterns and characteristics. *IEEE Network*, 1997, vol. 11, no 6, p. 10-23.
- [15] TRAMMS Consortium, TRAMMS IP Traffic report, Tech. Rep. 2, TRAMMS Project, 2008.
- [16] FUKUDA, KENSUKE; CHO, KENJIRO; ESAKI, HIROSHI. The impact of residential broadband traffic on Japanese ISP backbones. *ACM SIGCOMM Computer Communication Review*, 2005, vol. 35, no 1, p. 15-22.
- [17] FLOYD, SALLY; PAXSON, VERN. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking*, 2001, vol. 9, no 4, p. 392-403.
- [18] GARCÍA-DORADO, JOSÉ LUIS, ET AL. Characterization of the busy-hour traffic of IP networks based on their intrinsic features. *Computer Networks*, 2011, vol. 55, no 9, p. 2111-2125.
- [19] The 2017 Top Programming Languages. Stephen Cass. IEEE Spectrum Jul. 2017. <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>.

Apéndice A

Tablas de resultados para los tres modelos

Primer modelo		Tráfico entrante		Tráfico saliente	
		MRTG	Wavelet	MRTG	Wavelet
Señal	Agregación	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²
IXP1	Semanal	1167.27	486.87	1839.59	1023.77
	Mensual	11580.92	3140.98	14955.41	3903.88
	Anual	177417.17	183242.16	231190.85	228293.73
IXP2	Semanal	4.17	4.02	4.14	4.09
	Mensual	13.34	10.00	9.96	9.21
	Anual	130.82	126.35	44.60	43.67
C1	Semanal	470.68	383.02	463.59	245.94
	Mensual	2260.19	1211.97	3816.86	1609.65
	Anual	23829.77	23388.28	36566.68	37830.12
C2	Semanal	10.58	9.95	105.80	92.08
	Mensual	34.39	28.29	442.62	239.96
	Anual	190.37	180.25	4246.28	4097.98
C3	Semanal	7.71	7.18	30.85	29.94
	Mensual	19.55	19.40	102.15	73.70
	Anual	33.47	32.84	567.72	567.57
C4	Semanal	313.18	307.54	1024.02	940.40
	Mensual	780.58	627.24	3692.82	2295.45
	Anual	4402.73	4351.18	27247.20	26920.81
ISP1	Semanal	10.10	9.26	75.82	70.42
	Mensual	35.03	27.50	267.29	162.52
	Anual	252.52	251.00	1600.26	1553.07
ISP2	Semanal	1199.98	646.58	265.94	235.85
	Mensual	10198.05	3016.39	1106.33	768.16
	Anual	167751.47	164452.12	9950.58	9815.50
U1	Semanal	2.90	2.78	41.64	39.09
	Mensual	7.90	6.56	131.27	92.57
	Anual	45.68	45.43	995.33	986.56
U2	Semanal	6.07	5.74	18.85	18.01
	Mensual	14.33	14.18	53.34	41.59
	Anual	33.85	33.77	195.13	192.49
U3	Semanal	1.80	1.64	48.22	47.70
	Mensual	4.03	3.66	100.44	94.07
	Anual	14.14	14.02	245.78	245.68
U4	Semanal	47.78	47.38	72.34	68.96
	Mensual	145.25	121.78	245.45	160.58
	Anual	1312.43	1282.02	2465.06	2457.38
U5	Semanal	3.07	2.92	29.43	27.70
	Mensual	9.56	7.84	101.25	69.57
	Anual	53.21	52.19	630.01	617.54
E1	Semanal	19630.73	19429.76	33599.90	33260.94
	Mensual	31479.84	30730.36	50033.07	49589.05
	Anual	64223.01	61756.63	45802.52	45556.55

Tabla A.1: Resultados de precisión para el primer modelo evaluado sobre los 14 enlaces de red.

Segundo modelo		Tráfico entrante		Tráfico saliente	
		MRTG	Wavelet	MRTG	Wavelet
Señal	Agregación	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²
IXP1	Semanal	1167.27	173.42	1839.59	338.97
	Mensual	11580.92	735.03	14955.41	1561.40
	Anual	177417.17	77070.42	231190.85	111981.24
IXP2	Semanal	4.17	1.25	4.14	1.20
	Mensual	13.34	5.12	9.96	4.70
	Anual	130.82	62.95	44.60	25.54
C1	Semanal	470.68	147.35	463.59	75.37
	Mensual	2260.19	585.96	3816.86	421.30
	Anual	23829.77	11905.17	36566.68	16750.15
C2	Semanal	10.58	3.82	105.80	31.30
	Mensual	34.39	14.64	442.62	117.40
	Anual	190.37	104.43	4246.28	1819.40
C3	Semanal	7.71	1.49	30.85	8.82
	Mensual	19.55	8.04	102.15	38.29
	Anual	33.47	20.53	567.72	264.75
C4	Semanal	313.18	76.82	1024.02	274.62
	Mensual	780.58	262.92	3692.82	1080.35
	Anual	4402.73	2160.39	27247.20	12012.59
ISP1	Semanal	10.10	3.57	75.82	26.58
	Mensual	35.03	14.07	267.29	88.31
	Anual	252.52	138.60	1600.26	799.27
ISP2	Semanal	1199.98	225.12	265.94	59.21
	Mensual	10198.05	844.80	1106.33	230.75
	Anual	167751.47	69145.67	9950.58	4189.10
U1	Semanal	2.90	0.94	41.64	14.09
	Mensual	7.90	2.92	131.27	48.89
	Anual	45.68	23.56	995.33	447.04
U2	Semanal	6.07	0.92	18.85	5.32
	Mensual	14.33	4.05	53.34	20.05
	Anual	33.85	17.70	195.13	103.79
U3	Semanal	1.80	0.60	48.22	18.20
	Mensual	4.03	1.79	100.44	58.97
	Anual	14.14	7.65	245.78	145.66
U4	Semanal	47.78	16.06	72.34	22.64
	Mensual	145.25	61.84	245.45	79.96
	Anual	1312.43	627.24	2465.06	1112.59
U5	Semanal	3.07	1.27	29.43	10.78
	Mensual	9.56	4.18	101.25	39.40
	Anual	53.21	28.65	630.01	320.45
E1	Semanal	19630.73	10912.41	33599.90	15319.12
	Mensual	31479.84	21706.40	50033.07	36271.02
	Anual	64223.01	41982.72	45802.52	32432.84

Tabla A.2: Resultados de precisión para el segundo modelo evaluado sobre los 14 enlaces de red.

Tercer modelo		Tráfico entrante		Tráfico saliente	
		MRTG	Wavelet	MRTG	Wavelet
Señal	Agregación	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²	MSE ($\frac{Mbits}{s}$) ²
IXP1	Semanal	1167.27	229.49	1839.59	447.09
	Mensual	11580.92	1123.88	14955.41	2070.78
	Anual	177417.17	111895.21	231190.85	154052.97
IXP2	Semanal	4.17	1.82	4.14	1.77
	Mensual	13.34	6.44	9.96	5.83
	Anual	130.82	84.67	44.60	31.76
C1	Semanal	470.68	195.93	463.59	109.00
	Mensual	2260.19	730.00	3816.86	654.47
	Anual	23829.77	15771.03	36566.68	23688.08
C2	Semanal	10.58	5.15	105.80	45.72
	Mensual	34.39	17.92	442.62	146.41
	Anual	190.37	133.38	4246.28	2634.63
C3	Semanal	7.71	2.56	30.85	13.35
	Mensual	19.55	10.25	102.15	47.89
	Anual	33.47	24.02	567.72	359.44
C4	Semanal	313.18	113.56	1024.02	408.94
	Mensual	780.58	334.51	3692.82	1380.69
	Anual	4402.73	2866.94	27247.20	17191.23
ISP1	Semanal	10.10	4.87	75.82	37.49
	Mensual	35.03	17.42	267.29	107.35
	Anual	252.52	178.16	1600.26	1072.99
ISP2	Semanal	1199.98	312.71	265.94	79.92
	Mensual	10198.05	1178.64	1106.33	293.39
	Anual	167751.47	99711.49	9950.58	5821.64
U1	Semanal	2.90	1.26	41.64	20.22
	Mensual	7.90	3.62	131.27	59.86
	Anual	45.68	30.25	995.33	620.36
U2	Semanal	6.07	1.50	18.85	7.97
	Mensual	14.33	5.75	53.34	25.42
	Anual	33.85	21.55	195.13	135.64
U3	Semanal	1.80	0.81	48.22	26.66
	Mensual	4.03	2.15	100.44	69.48
	Anual	14.14	9.57	245.78	182.06
U4	Semanal	47.78	22.50	72.34	32.65
	Mensual	145.25	76.18	245.45	98.67
	Anual	1312.43	816.92	2465.06	1580.16
U5	Semanal	3.07	1.66	29.43	14.89
	Mensual	9.56	5.06	101.25	47.54
	Anual	53.21	36.73	630.01	425.87
E1	Semanal	19630.73	13719.31	33599.90	21452.32
	Mensual	31479.84	24357.17	50033.07	40751.90
	Anual	64223.01	48475.45	45802.52	36343.53

Tabla A.3: Resultados de precisión para el tercer modelo evaluado sobre los 14 enlaces de red.

