Universidad Politécnica de Cartagena

Escuela Técnica Superior de Ingeniería de Telecomunicación

PROYECTO FINAL DE CARRERA: MEDIDAS Y MODELADO DEL TRÁFICO DE RED

AUTOR: Ignacio Cano Serrano
DIRECTOR: Alejandro Santos Martínez Sala

Septiembre / 2013

Autor	Ignacio Cano Serrano
E-mail del autor	Ignacio.cano.serrano@gmail.com
Director	Alejandro Santos Martínez Sala
E-mail del director	alejandros.martinez@upct.es

Titulo del PFC	Medidas y modelado del tráfico de red.

Resumen

El crecimiento cada vez mayor de la Web y su evolución progresiva ha hecho de Internet, y sus muchas aplicaciones, uno de los instrumentos más poderosos y populares para la transmisión de información, lo que hace necesario hacer la navegación accesible, rápida y mejor. El protocolo http ha sido hasta hace bien poco, una herramienta fuerte y poderosa, pero la necesidad de mejorarlo se ha hecho imprescindible. Las limitaciones que tiene el protocolo Http se han visto solventadas por la nueva tecnología Ajax "Asynchronous Javascript y XML" que, al ser nueva, existen aún modelos, estadísticas y probabilidades por estudiar.

El objetivo del proyecto es realizar pruebas para analizar el tráfico con esta nueva forma de navegar, comparándola con la estándar. Se centrará en pruebas de conexiones en las que se haga vigente la navegación Web2.0 como es el caso de Facebook, obteniendo los parámetros necesarios para su posterior análisis y modelaje y, utilizando después algunos programas basados en hacer análisis estadísticos, e incluso concretar parámetros para crear una cadena de Markov fiable.

Titulación	Ingeniero Técnico de Telecomunicaciones
Departamento	Tecnologías de la información y las comunicaciones
Fecha de Presentación	Septiembre - 2013

INDICE GENERAL

ÍNDICE DE ILUSTRACIONES

Capitulo 1: Introducción

Ajax no es una tecnología. Es el conjunto de más tecnologías independientes que en conjunto dan lugar a perspectivas de gran alcance.

De "*Ajax: A new approach to web applications* " J.J. Garrett

El crecimiento cada vez mayor de la Web y su evolución progresiva ha hecho de Internet, y sus muchas aplicaciones, uno de los instrumentos más poderosos y populares para la transmisión de información de todo tipo. El creciente uso de Internet, y más general de la red, traen como consecuencia la creciente necesidad de hacer la navegación accesible, rápida y sencillamente mejor. Y como consecuencia, su continuo estudio para la adaptación y progresión de nuevas tendencias tecnológicas. Internet es un mundo que nunca para de crecer.

1.1 Antecedentes

Con el surgimiento de Internet, el mundo de la computación ha ido obteniendo significantes avances, entre los cuales se pueden destacar: el desarrollo del lenguaje de marcado de hipertexto HTML, el desarrollo del primer navegador Web comercial, el desarrollo del protocolo para transferencia de hipertexto (HTTP), el lenguaje de marcado extensible (XML), y el lenguaje de *scripting* JavaScript. Pero además de estos avances tecnológicos, Internet crea un impacto dentro de la sociedad, ya que éste pasa a ser uno de los principales medios de comunicación e información masiva gracias a la disponibilidad y ubicuidad que posee.

El rápido crecimiento de Internet ha conducido a que se desarrolle un nuevo enfoque dentro de la creación de aplicaciones, donde el objetivo principal es tener una mayor riqueza en elementos de interfaz y de interacción. Estos cambios han generado la búsqueda de una experiencia mejor para el usuario de aplicaciones Web, así como un acercamiento al enfoque interactivo que prestan las nuevas aplicaciones.

Con el propósito de cubrir estos requerimientos, han surgido una serie de tecnologías y enfoques que permiten el desarrollo de aplicaciones en la Web con mayor riqueza en cuanto a interfaz e interacción se refiere, tal es el caso del conjunto de tecnologías AJAX, el cual revoluciona Internet gracias a sus características y prestaciones, entre las cuales destaca las peticiones asíncronas y modificación dinámica del contenido de las páginas.

El punto de partida del presente proyecto fin de carrera son los estudios realizados sobre la navegación con la Web 1.0; múltiples universidades y otras fuentes de investigación centraron sus energías en la elaboración de estudios relacionados con el modelado y caracterización de pruebas realizadas en base al protocolo HTTP.

1.2 Formalización del problema y objetivos generales del proyecto

Han hecho falta la estimulación de investigadores, programadores de software y especialistas en redes para participar en la construcción de una nueva tecnología que permite una mejor navegación Web en muchos aspectos, un mayor rendimiento, optimización de recursos y, principalmente, una mejora más que notable para el usuario. En términos específicos, dicha nueva y revolucionaria tecnología se denomina "*Asynchronous Javascript and XML*", o simplemente Ajax, que, citando a Garrett, es "un conjunto de tecnologías independientes" cuya unión ha permitido que navegar por la Web rinda mejor, sea más veloz y fácil. Que con Ajax se realice un mejor y ligero rendimiento es un hecho, pero lo que no es tan sencillo de explicar, desde el punto de vista del tráfico, es la forma en la que Ajax actúa sobre la transmisión de datos entre clientes y servidores. Ajax, de hecho es una contradicción a la navegación estándar basada en el uso simple del protocolo HTTP. Las diferencias entre Ajax y la navegación clásica son bastante notables, así como los modelos estadísticos y probabilísticas que caracterizan a la Web 1.0, pero para entender realmente las diferencias, hay que tener en cuenta los modelos anteriores y partiendo de las consideraciones adecuadas hasta poder compararlos con los de la navegación Ajax.

El problema fundamental es que, hasta la fecha, no existe un modelo detallado y completo que permita modelar las variables características de una sesión cualquiera basada en Ajax, y por tanto, que permita una comparación desde el punto de vista estadístico y de probabilidad los modelos que derivan de la navegación clásica y la de Ajax. Por esta razón, el trabajo presentado en este proyecto final de carrera tiene como objetivo la creación de un modelo estadístico y probabilístico del tráfico de ahora ya conocida categoría Web 2.0. Esto, por lo tanto tiene como objetivo desarrollar un modelo que permita buscar, en primer lugar, parámetros clave que distingan la navegación basada en Ajax de la normal, y, a continuación, poder estudiar la evolución de estos parámetros hasta asimilarlos de forma que sea posible llegar hasta una curva conocida de estadística y probabilidad que los defina. Sólo de esta manera, se podrá comparar estadísticamente las dos modalidades de navegaciones. Aunque esto solo sea una pequeña pieza del puzzle, hay mucho que indagar sobre esta nueva navegación.

Con el fin que se acaba de mencionar, se propone una sistemática válida para cualquier escenario, en este proyecto se explica la metodología de forma que se entienda por qué medimos unos parámetros en concreto, se sigue una serie de instrucciones, se extraen los datos necesarios de un modo y se trabaja con ellos. Se presenta un procedimiento que será de utilidad en cuánto a querer resaltar diferencias, medir cantidad de tráfico, crear gráficas y estadísticos, parámetros para según que modelos, entre otras cosas, de cualquiera de las dos navegaciones, aplicadas a cualquier aplicación Web.

1.3 Estructura de la memoria

La presente memoria está dividida en tres partes. La primera parte se desarrolla en el capítulo 2, subdividido éste a su vez en otros dos apartados que se centran, respectivamente, en la descripción del funcionamiento de la navegación clásica, detallando también aquellos aspectos por los cuales existía la necesidad de mejorar; y la presentación de los modelos estadísticos que describen la evolución de los parámetros que caracterizan el tráfico de red (paquetes, solicitud, respuesta, tiempo...). La segunda parte se centra en la descripción del funcionamiento del mecanismo Ajax, en particular enunciando algunos de los problemas de la navegación clásica a los que Ajax ha puesto remedio, y la descripción de algunos modelos estadísticos, a menudo incompletos, obtenidos a partir de algunos experimentos.

El segundo bloque de la memoria desarrollado en el los capítulos 3 y 4, se concentran en el desarrollo y elaboración de las pruebas sobre Facebook, qué variables y parámetros son imprescindibles y situarse en el entorno para con él llegar a las conclusiones y entendimiento final del trabajo, ambientarse y conocer las herramientas utilizadas.

En los capítulos 5 y 6, comprende el núcleo del PFC: a través de pruebas de laboratorio se presenta una campaña de medidas donde se extrae información sobre las características de la navegación basada en Ajax y, además, modelar muchos de los parámetros característicos del tráfico de red.

En el capítulo final, se encuentra un resumen de las conclusiones del proyecto. Seguido por el anexo donde se encuentran las tablas de datos que se han empleado para caracterizar el tráfico. Y terminando con la bibliografía, donde se sitúan los libros y páginas de donde se han extraído citas e información para este estudio.

2 Capitulo 2: Estado del arte de la navegación con Web 1.0 y Web 2.0

2.1 Modelo clásico http

Pese a que es bien conocido el funcionamiento del protocolo HTTP , la primera parte de este capítulo se encargará de repasar los puntos básicos de dicho protocolo para contextualizar y no olvidar conceptos importantes a la hora de entender bien que se está midiendo.

2.1.1 El protocolo http

El HTTP es el protocolo cliente-servidor más utilizado por las aplicaciones Web para la comunicación entre dos o más ordenadores. El funcionamiento del protocolo HTTP, sus especificaciones y las diferentes versiones, están ampliamente detallados en el RFC 1945 y RFC 2616. El HTTP, a diferencia de otros protocolos de la capa de aplicación como FTP, Telnet, mail (SMTP), etc., que fueron creados para determinadas aplicaciones (es decir, para la transferencia de datos de un cierto tipo), fue diseñado explícitamente para la transferencia de datos de cualquier naturaleza: el archivo que soporta el protocolo HTTP es el formato HTML, pueden ser archivos gráficos, sonoros, secuencias de animaciones, archivos generados por procesadores de texto, y en general, todos los formatos de tipo MIME. En la Fig. 2.1 se encuentra un esquema básico, pero fundamental para la comprensión de la decisión de las variables que se seleccionarán para el estudio. El mensaje HTTP (*Request* y *Reply*) está encapsulado dentro del segmento.

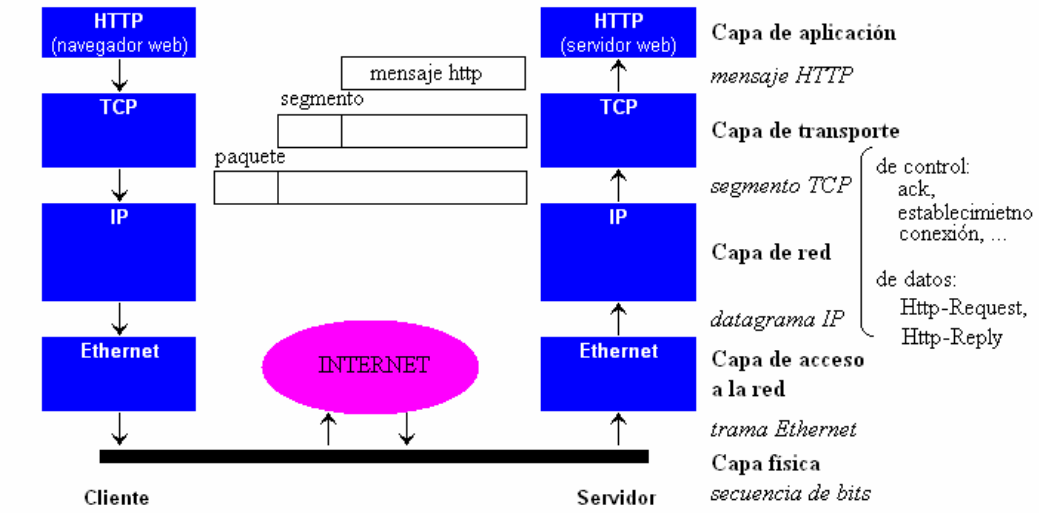


Fig. 2.1: Esquema básico http.

2.1.2 Las versiones del protocolo http

Las versiones oficiales 1.0 y 1.1 del protocolo HTTP, fueron, en realidad, precedidas por la versión 0.9, creada a finales de los 80; tal versión fue la base de World Wide Web, sin embargo, la primera versión del protocolo que está disponible, es la 1.0, que se llevó a cabo en 1991 y propuesta como RFC 1945 [1] en 1996. Posteriormente, la creciente popularidad de la World Wide Web, y el consecuente éxito del protocolo HTTP, pusieron en evidencia algunas limitaciones de la versión 1.0 del protocolo, por lo tanto, se extendió en la versión 1.1 y se presentó como RFC 2068 [2] en 1997 y posteriormente actualizado en 1999 como se describe en el RFC 2616 [3]. En la actualidad se pueden utilizar ambas versiones del protocolo, en función de las necesidades y el tipo de conexión que se establece entre el cliente y el servidor.

La versión 1.0 requiere que se establezca una nueva conexión TCP para cada objeto incorporado en una página Web. Esta característica implica tiempos de latencia medianamente consistentes y una excesiva sobrecarga del protocolo, ya que es necesario verificar tres operaciones para el reconocimiento mutuo entre el cliente y el servidor por cada objeto enviado y recibido. Mientras que la Versión 1.1, definida como *connection oriented*, ha superado los límites de las versiones anteriores. En este caso se establece una conexión TCP persistente por un determinado tiempo *t* cuando un objeto viene solicitado / enviado en una página Web; Consigue que cuando se emite una nueva solicitud dentro de los límites de tiempo *t*, se reutiliza la conexión ya existente. De este modo se reduce el tiempo de latencia.

2.1.3 La comunicación cliente-servidor

Dado que el protocolo HTTP proporciona una comunicación bilateral, hay dos tipos de mensajes HTTP: mensajes de petición o *Web Request*, y los mensajes de respuesta o *Web Reply*. Por esta razón, el tipo de protocolo HTTP es también conocido como *Request/Reply*. En la Fig. 2.2 se muestra el modelo de comunicación cliente-servidor. Estos mensajes se transmiten en un formato denomindado MIME, *Multipurpose Internet Mail Extensions*.

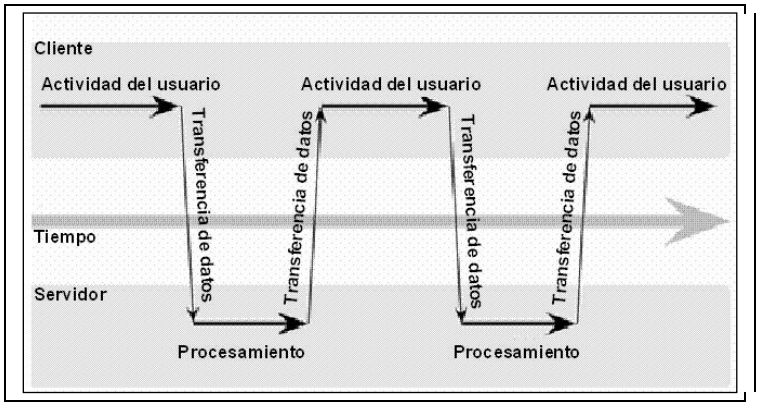


Fig. 2.2: Modelo clásico para aplicaciones Web.

Para permitir el intercambio de mensajes, el cliente y el servidor deben estar identificados por una dirección IP única. Para ello, se utiliza el método de direccionamiento del protocolo IP. Actualmente la versión en uso del protocolo IP es IPv4, descrita originalmente en el RFC 791 [5].

2.1.4 Las consideraciones sobre el protocolo HTTP

Las explicaciones dadas hasta ahora hacen pensar erróneamente que el el protocolo HTTP es la solución a todo el mecanismo de Internet y que no tiene defectos en su implementación y en su uso. Sin embargo, el Protocolo HTTP presenta dos limitaciones importantes.

El primer fallo reside en el hecho de que cada transacción Request-Reply se abre y se cierra después de una conexión TCP. El cliente envía peticiones de recursos al servidor, descartando la página desplegada actualmente. El servidor recibe la petición y la procesa, respondiendo seguidamente con una página HTML nueva, la cual es recibida y desplegada por el navegador. Este proceso genera una interrupción en la interacción entre el usuario y la aplicación en cada petición. Además, como las páginas de respuesta enviadas por el servidor posiblemente contienen pocos cambios respecto a la página anterior, puede producir una sobrecarga innecesaria entre cada petición. De modo que TCP y HTTP juntos son ineficientes.

El segundo problema es la característica del protocolo sin estado, que significa que el protocolo no tiene memoria, no existe algo referente a preguntas y respuestas pasadas. La ventaja de ser sin estado es la facilidad y la simplicidad del protocolo. De hecho, cualquiera puede escribir un servidor HTTP con una cantidad relativamente pequeña de código. Pero por otro lado, el HTTP no garantiza que haya una conexión estable durante la transacción. Para resolver este problema, se integra al protocolo HTTP el uso de las llamadas "*cookies*", Una cookie es un componente que lleva el seguimiento del cliente y más información de estado en el servidor. Pero no deja de ser un “parche” que soluciona parcialmente una limitación del protocolo.

2.1.5 El modelo de tráfico Http

A partir de la introducción y del éxito del protocolo HTTP se realizaron numerosos estudios experimentales sobre el modelado de tráfico Web.

El objetivo principal de la creación de un modelo, es describir y explicar la tendencia del objeto de estudio. En este caso, el objeto es describir el tráfico HTTP, que en realidad se compone de varios parámetros. Obviamente, es inverosímil pensar que extraer información sobre la cantidad y la calidad del tráfico durante cualquier sesión de navegación Web, puede reproducir fielmente y al detalle cualquiera de las sesiones. Por lo tanto, un modelo se limita al carácter general de aspectos comunes a todos los objetos modelados.

Para caracterizar las aplicaciones Web hay tres enfoques:

- Un primer enfoque basado en un análisis del tráfico del servidor (basados principalmente en la captura de *Request*).
- Un segundo enfoque basado en el análisis del tráfico del cliente (basados principalmente en la captura de *Reply*).
- Y un tercer tipo de enfoque basado en el análisis de los paquetes transferidos a una subred, por lo general una conexión Ethernet, que se consigue rastreando la información sobre los mensajes, cantidad en bytes, tiempos, etc…

2.1.5.1 Descripción de las pruebas

La mayoría de los resultados obtenidos del análisis experimental llevado a cabo por varios investigadores son similares. Muchas de las variables que aquí se explican no son objeto de nuestro estudio, pero se mencionan para conocer más en profundidad los parámetros de la navegación. O sea, más que sus conclusiones y datos lo que interesa de examinar otros artículos es la lista de variables que se seleccionan para su medición, para poder crear nuestro propio procedimiento y obtener nuestros propios resultados. Los principales artículos que se han usado como referencia para comprobar las variables son los siguientes:

El primer análisis experimental tratado en el "*Modeling of http traffic*" [6], que compara las diferencias entre el tráfico resultante del uso de HTTP 1.0 y del 1.1. Los autores del texto lo que quieren demostrar es que no hay diferencias relevantes en términos de cantidad de tráfico entre el uso de HTTP 1.0 y 1.1 y que existe obviamente una cierta correlación entre el comportamiento de los usuarios y el tráfico generado a nivel TCP.

"*An empirical modelo of HTTP network traffic*" [7] tiene como objetivo desarrollar un modelo completo para el tráfico HTTP, tratando de controlar el mayor número de variables posible.

Las pruebas reportadas en el artículo "*A behavioral modelo f web traffic*" [8] se basan en el análisis combinado de la cabecera de los paquetes TCP y HTTP.

Y, finalmente, "*A new traffic model for current user web Browning behavior*" [9] que es el análisis más reciente (2007), cuyo propósito es crear un modelo que reproduzca fielmente el tráfico HTTP, con el fin de lograr un generador de tráfico.

2.1.5.2 Request

La longitud de la petición en [7] se mide contando los bytes de todos los paquetes, los resultados revelaron (Fig. 2.3) que el tamaño medio de la petición en bytes es de 320 bytes. La densidad de probabilidad resultante es bimodal, con dos picos: el primer máximo de 250 bytes y el segundo a 1 KB. La explicación más probable es que el primer pico corresponde a las solicitudes para transmitir archivos simples, mientras que el segundo pico corresponde a los archivos más complejos.

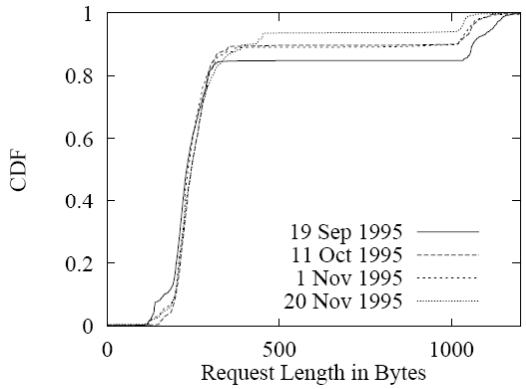


Fig. 2.3: Tamaño de las solicitudes (byte).

El análisis " *A new traffic model for current user web Browning behavior* " [9] sobre el tamaño de la solicitud reveló una media de 318.59 bytes.

En "*An empirical modelo of HTTP network traffic*" [7] también se investigó la diferencia entre solicitud primaria y secundaria (solicitud primaria es la primera solicitud hecha por el cliente al servidor en relación a una página Web, y solicitud secundaria todas las solicitudes posteriores a la primera, todas a la misma página). El análisis reveló que en la primera petición Web de una página muchos archivos tienden a ser más largos que los posteriores. La media de la primaria es de 240 bytes, mientras que la media de la secundaria es de 230 bytes. El gráfico de la Fig. 2.4 muestra la diferencia entre la curva de la función de distribución en el primario y el secundario del primer análisis propuesto en [7].

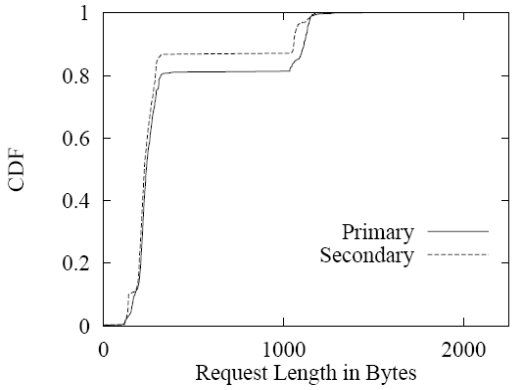


Fig. 2.4: FDP del tamaño de las solicitudes primarias y secundarias.

2.1.5.3 Reply

Los artículos [7] y [8] han estudiado también los Reply. En general, afirman que las repuestas tienden a ser más grandes que las solicitudes Web. Normalmente las respuestas incluyen código Html, datos multimedia o mensajes de error. Los archivos multimedia son notablemente más grandes que los archivos de texto. Se obvian algunas de las gráficas, puesto que se ha visto en el apartado anterior la forma y en capítulos posteriores se harán con nuestras pruebas.

La densidad de probabilidad de la Reply ha mostrado una media de 8 a 10 KB. Existe alguna respuesta con un alto número de bytes (como máximo 1MB) lo cual indica que se trataba de mensajes de error. Estas características revelan una función densidad de probabilidad Paretiana. El gráfico de la Fig. 2.5 modela la evolución de la longitud de la respuesta Web [7].

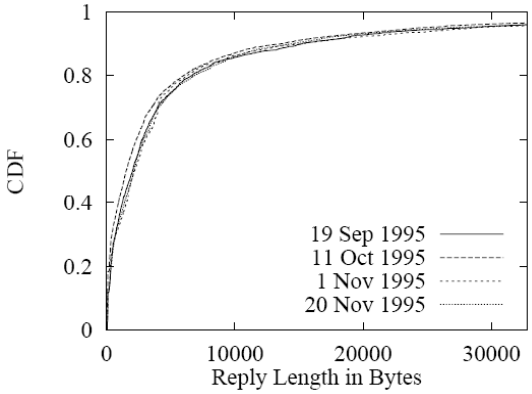


Fig. 2.5: FDP de la dimensión de la respuesta

2.1.5.4 Las conexiones

Dentro del análisis de [6], tratado por completo en una navegación simple Web con la versión 1.1, se observa, en primer lugar, que en el servidor durante la navegación se incrementa el uso de conexiones persistentes. Según el tiempo T se distinguen tres áreas principales:

- La primera en la que el tiempo T es muy bajo, lo que probablemente corresponde a que el usuario no espera a que la página esté completamente cargada antes de hacer clic a otra página. En este caso, el uso de conexiones persistentes es bajo y el número de conexiones por página aumenta.

- La segunda área donde el tiempo T es medio, probablemente corresponde a la situación en la que el usuario visita muy rápidamente una página antes de solicitar otra. En este caso, el tráfico HTTP se compone de pocas conexiones TCP pero de larga duración.

- Por último, la tercera zona, donde T es el tiempo muy alto, que corresponde a la situación en la que el usuario visita una página y se centra en una lectura detallada antes de entrar en una segunda. En este caso, obviamente, el tiempo de espera termina y se cierra la conexión, luego requiere actualizar la página. Aumenta el número de conexiones mientras que la carga principal de cada conexión disminuye.

En general, las tres áreas mencionadas están presentes en todos los análisis sobre el tráfico HTTP. El gráfico de la Fig. 2.6 presenta en el eje *x* el tiempo entre la llegada de dos páginas consecutivas, y en el eje *y* se ven representados el número medio de conexiones TCP por página (izquierda) y el tamaño medio en KBytes (derecha).

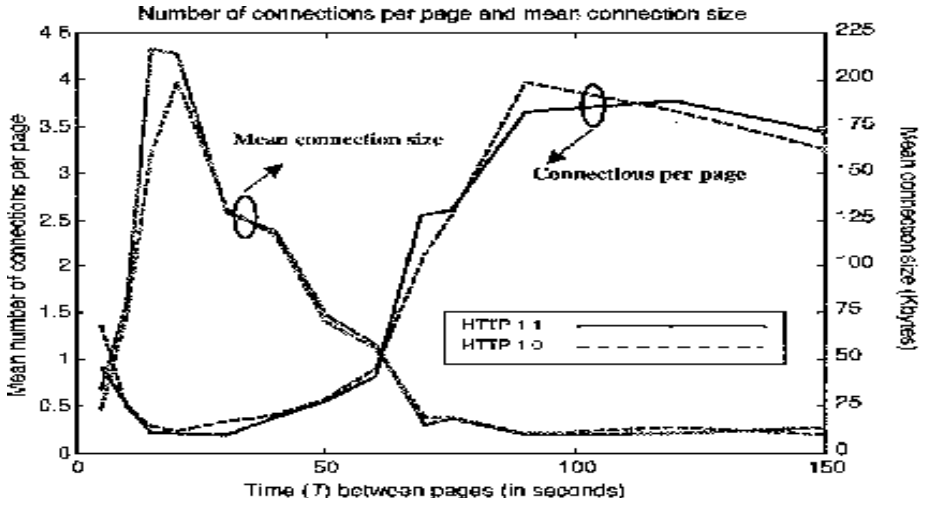


Fig. 2.6: Conexiones entre dos páginas consecutivas.

La conclusión que sigue es que, en términos de análisis de las conexiones, no hay diferencias sustanciales entre el uso de HTTP 1.0 y 1.1.

2.1.5.5 Los tiempos

En un análisis, desempeña un papel importante el análisis temporal de las sesiones de navegación. En particular, se modelan dos valores de tiempo: el tiempo de reflexión del usuario y el análisis de tiempo.

Tiempo de reflexión del usuario.

En [7], el plazo de Tiempo de reflexión, es el tiempo que transcurre entre dos solicitudes consecutivas. Depende del usuario, ya que es éste quien decide cuánto tiempo se quiere permanecer en una página.

Este tiempo también se ve influido en parte por el *Resfresh*, que es la actualización de una página de forma automática sin la intervención del usuario. La razón de su periocidad es la necesidad de actualizar los contenidos de la página, para que la información no se quede obsoleta.

Durante la prueba, el tiempo de reflexión del usuario es relativamente corto: la media es pequeña, de unos 15 segundos. Se muestra en el gráfico de la Fig. 2.7 las cuatro pruebas propuestas en [7].

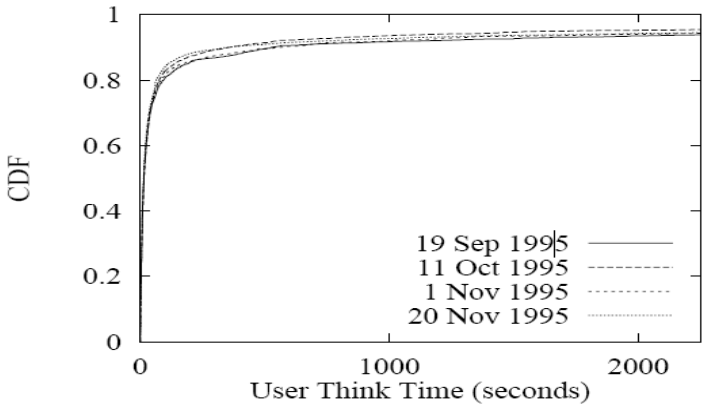


Fig. 2.7: FDP del tiempo de reflexión del usuario.

Tiempo de análisis

El tiempo de análisis es el tiempo que transcurre entre la llegada de un HTML y la llegada del siguiente objeto incorporado en la misma página Web por lo tanto significa que es el tiempo entre la llegada de la respuesta primaria y la primera respuesta secundaria. Los datos de [9] ponen de manifiesto un tiempo medio de 3,12 segundos.

2.1.5.6 Selección del servidor

Un usuario que abre una página Web, pregunta a un servidor específico por la página que quiere de vuelta.

Sucede a menudo que la navegación de un usuario se basa en el mismo servidor para más de una página, por poner un ejemplo, es muy probable que preguntando por la página principal de un sitio genérico, la segunda solicitud estará en una sección de la misma, por lo tanto, residen en el mismo servidor.

El número de documentos consecutivos presentados por un único servidor tiende a ser pequeño, aproximadamente el 80% de los servidores hacia delante harán uso de menos

de seis documentos: en este análisis se encontró que los usuarios piden un promedio de cuatro páginas consecutivas para cada servidor.

La función de distribución fue de aproximadamente la misma para los cuatro análisis propuestos en [7]. En la Fig. 2.8 se representan los resultados citados.

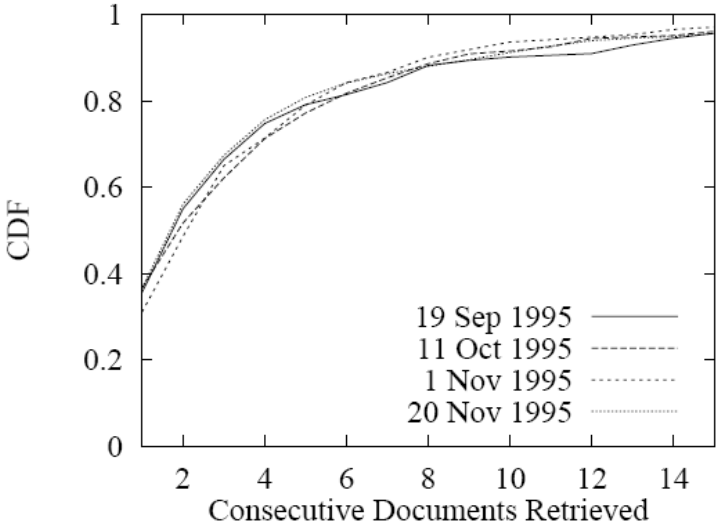


Fig. 2.8: FDP de los documentos consecutivos de un servidor.

2.2 AJAX

La rápida evolución de la Web en los últimos años nos ha llevado a la introducción de nuevas tecnologías para mejorar la experiencia de navegación por parte del usuario. Las innovaciones han sido abrumadoras hasta el punto de introducir una nueva categoría importante dentro de Internet: la Web 2.0. El propósito de la segunda parte de este capítulo es dar una explicación detallada del enfoque Ajax.

2.2.1 De la Web 1.0 a la Web 2.0

El término Web 1.0 fue introducido para identificar como puede Internet vincular documentos de hipertexto creados con HTML, sin la posibilidad de interacción con el usuario, a excepción de la navegación normal entre las páginas y el uso de motores de búsqueda.

El término Web 2.0 fue acuñado en 2004 por T. O'Reilly [10], lo que indica, en general, un estado de evolución de Internet y, en particular de la World Wide Web. La Web indica cómo la Web 2.0, es el conjunto de todas las aplicaciones en línea que permiten un alto nivel de interacción entre localización-usuario.

Aplicaciones Web 2.0 incluyen blogs, foros, sistemas de chat tales como Wikipedia, *Youtube*, *Facebook*, *MySpace*, *Twitter*, *Gmail*, *WordPress*, *Maps* y muchos otros.

La Web 2.0 es un estallido revolucionario en nuestra forma de pensar y utilizar Internet. Se dice de la Web2.0 que es un "nuevo enfoque filosófico de la red", que tiene entre sus características innovadoras el alcance social de la Web y compartir múltiples recursos. A primera vista propiedades que pueden parecer redundantes, ya que incluso la Web 1.0 permite el intercambio de recursos, un ejemplo del aspecto innovador de la

Web 2.0 puede ser el famoso *blog*, una página que se crea en medida de sus propios recursos. Para crear una página Web personal, la Web 1.0 precisa experiencia informática en términos de lenguajes de programación, lenguajes de marcas, un conocimiento de la red, conocimiento de métodos de bases de datos, y mucho más. Con la Web 2.0, cualquiera puede crear su propio blog personal con la ayuda de los instrumentos preconfeccionados para los menos experimentados.

La innovación desde el punto de vista del usuario, puede hacer pensar en un trastorno de la estructura del protocolo, pero en realidad tanto la Web 1.0 como la Web 2.0 usa los protocolos TCP / IP y HTTP, y la comunicación basada en hipertexto. La diferencia que determina la clasificación de las Webs es el hecho de que los usuarios interactúan con la Web de manera completamente innovadora: si con la Web 1.0, un usuario sólo podía ver una página, ahora con la Web 2.0 puede enriquecer el contenido de una página, como la expresión de opiniones o la introducción de información nueva.

¿Qué es lo que hace que la Web 2.0 sea el núcleo de todas estas innovaciones? son una serie de tecnologías de programación particulares, nunca antes exprimidas. La más conocida y utilizada es Ajax.

2.2.2 Ajax. Un conjunto de tecnologías independientes

El término Ajax se presentó por primera vez en 2005 en un artículo de Jesse James Garrett, titulado "Ajax: Un Nuevo Enfoque para Aplicaciones Web" [11]. Ajax significa *Asynchronous JavaScript And XML* (JavaScript y XML Asíncronos). Ajax es un enfoque de desarrollo basado en un conjunto de tecnologías independientes y ya existentes, agrupadas para presentar información e interactuar dinámicamente, de manera asíncrona, con un servidor Web. Con AJAX se busca entonces proveer de una mayor riqueza de interacción entre el usuario y la aplicación mediante tecnologías que se utilizan para el desarrollo de la construcción y aplicación de la Web interactiva RIA (*Rich Internet Application*).

Como ya se ha dicho, Ajax es un acercamiento a las aplicaciones Web utilizando una combinación de otras tecnologías. Más específicamente, estas tecnologías son: la lengua de *Markup* XHTML, el lenguaje de transformación XSLT, una serie de hojas de estilo en cascada, el estándar W3C DOM, el XMLHTTPRequest y JavaScript. En este apartado se explicará las diferentes tecnologías que se han mencionado y cómo cooperan para dar lugar a Ajax.

- XHTML, XSLT y CSS: para la presentación, estructuración y formato del contenido. XHTML permite utilizar las propiedades de XML para estructurar una página HTML; haciendo esto se garantiza la uniformidad de la estructura de las páginas Web. Esta característica es de enorme importancia porque puede adaptarse a diversas aplicaciones. XSLT transforma un documento XML en un documento con otro formato o de otro lenguaje de programación. El uso de XSLT se asocia con el de CSS, Hojas de estilo en cascada que permiten formar el contenido de la página Web, anteriormente estructurada con XHTML. Esto le da a una página Web mayor claridad y legibilidad para los usuarios.
- DOM (*Document Object Model*): Con el modelo de objetos del documento se logra obtener la estructura del documento HTML. Utilizando esta estructura se pueden agregar, eliminar y modificar elementos de la página de manera dinámica sin acceder o modificar el documento entero, todo esto mediante el

uso de la tecnología JavaScript.

- JavaScript: Mediante esta tecnología del lado del cliente se realizan las peticiones de manera asíncrona y, junto con el manejo del DOM, se logra la interacción dinámica con el usuario.
- XML: Para el intercambio de datos entre el cliente (navegador Web) y el servidor.

2.2.3 Por qué Ajax

A diferencia de las aplicaciones Web clásicas, AJAX cambia la forma en que interactúan los usuarios con las aplicaciones Web, brindando mayor continuidad y dinamismo en la interacción, y asímismo reduciendo los tiempos de espera que se presentan en las aplicaciones Web tradicionales.

AJAX se maneja en la capa de presentación de las aplicaciones Web, y los beneficios que éste brinda van dirigidos a esta capa, sin intervenir en el manejo y resguardo de los datos de la aplicación.

AJAX permite un uso más eficiente del ancho de banda, ya que sólo se transmite del servidor al cliente la información necesaria y no páginas completas con información que ya reside en el cliente, como encabezados y pie de páginas, gráficos e imágenes, etc.

Las aplicaciones AJAX pueden enviar peticiones al servidor sin interrumpir la interacción, manteniendo la página actual en el navegador de tal manera que le permita al usuario seguir interactuando con la aplicación. Esto es posible gracias al uso de peticiones en segundo plano (peticiones asíncronas).

Además de estos motivos explicados con anterioridad, el gran avance en la navegación y la revolución creada con su nuevo enfoque, no hay que olvidar que AJAX fue diseñada principalmente para superar las importantes limitaciones de las aplicaciones Web 1.0.

2.2.4 El mecanismo de Ajax

El mecanismo de Ajax es muy diferente al modelo clásico de funcionamiento de la Web, aunque brevemente, se puede tener un concepto del contraste entre ellas con la Fig. 2.9.

En el modelo Ajax viene introducido un motor Ajax, *o Ajax engine* [12], entre el cliente y el servidor. En lugar de cargar una página Web al inicio de la sesión, el navegador carga un motor AJAX, basado en JavaScript, que se encarga de la gestión de la página y de comunicarse con la Web servidor para el usuario, y este enfoque permite la interacción asincrónica entre servidores y aplicaciones, de modo que los usuarios no perciben los momentos de estancamiento de la navegación.

En particular, el motor AJAX permite superar el concepto de carga síncrona. La carga síncrona, establece que cuando una página Web se carga, todos sus elementos se cargan de forma secuencial, uno tras otro, a menudo sucede que uno o más elementos requieren un proceso de carga más largo que otros.

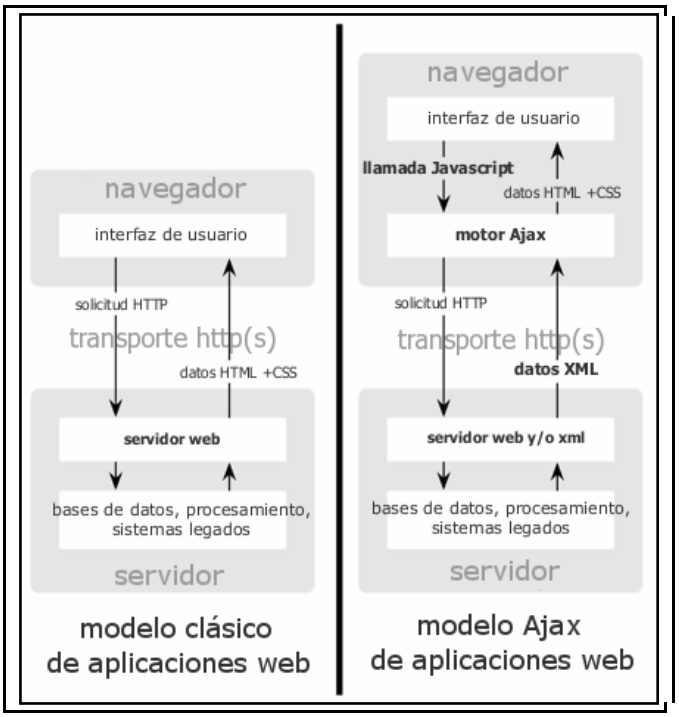


Fig.2.9: Comparación: Modelo clásico vs. Modelo AJAX.

El objeto `XMLHTTPRequest` es el responsable que permite realizar las peticiones en segundo plano, para luego recibir las respuestas del servidor sin alterar el curso de la interacción entre el usuario y la aplicación. En la Fig. 2.10 se muestra como estas peticiones asíncronas en el enfoque AJAX modifican el esquema tradicional petición-respuesta de las aplicaciones Web.

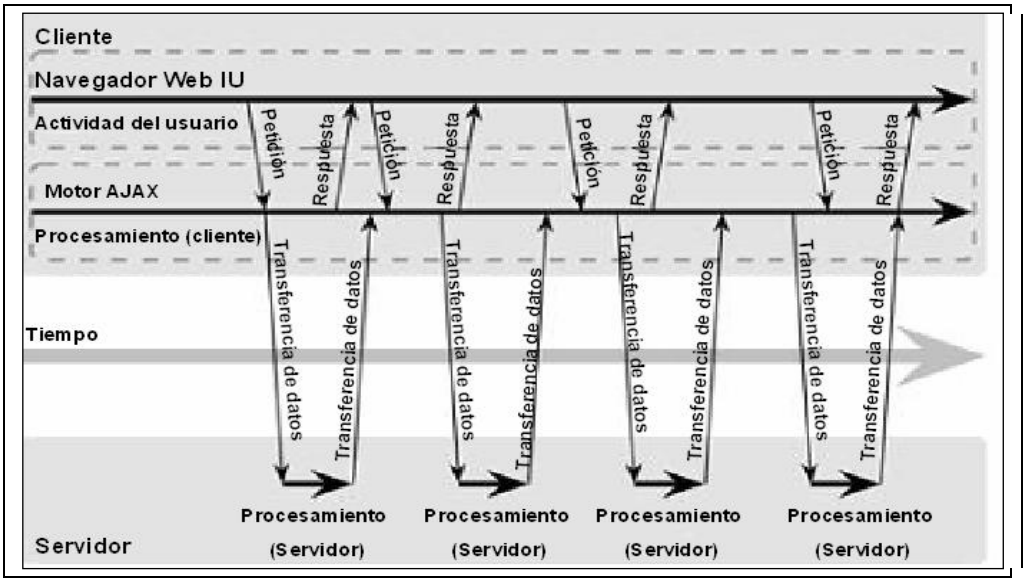


Fig. 2.10: Modelo asíncrono de aplicaciones AJAX.

2.2.5 Los enfoques de conexión

Se supone el caso de requerir una respuesta desde el servidor, es el motor el que se encarga del diálogo que permite al usuario seguir interactuando con la aplicación

mientras está esperando una respuesta. La operación denominada *pre-fetching* o pre-carga es la clave de Ajax. Tal operación requiere que se cargue en la caché del lado del cliente, los datos que tienen más probabilidad de ser respondidos. Para entender qué datos necesitarán en un futuro inmediato, se basa en tres puntos principales.

- La primera información recae sobre los hábitos del usuario, de cómo si éste siempre visita la misma página en cada sesión, será oportuno cargar en la memoria caché el contenido de ésta.

- La segunda información útil es el análisis de la actividad actual de lo que el usuario está visitando. Por ejemplo, si se está en la página número 1 de un artículo, es altamente probable que la próxima página que se necesite sea la 2.

- La tercera pieza de información se basa en la actividad de otros usuarios; y si estadísticamente se sabe que de una página le sigue una segunda, es muy probable que nuestro usuario haga lo mismo.

Los enfoques de conexión pueden ser de modalidades muy diferentes entre sí, tanto en términos de acciones como en el rendimiento final. Aquí se nombran algunos y se explicarán rápidamente su definición:

- El enfoque Tire: Ajax se comunican con el servidor a intervalos de tiempo específicos, definidos por el usuario, tales intervalos se conocen como tiempo para refrescarse o TTR (*Time to Refresh*).

- El enfoque de *streaming* más conocida como la transmisión HTTP. Este enfoque entiende el método por el cual una corriente de datos de servidor al cliente, usa una conexión HTTP persistente. A través de algunas técnicas, un script del lado del servidor mantiene la conexión abierta con el cliente, y es capaz de detectar cambios de estado de las aplicaciones del cliente, para cada cambio de estado, el servidor utiliza la conexión HTTP previamente establecida para enviar nuevos datos para la aplicación que se necesita. La ventaja de usar este método reside en la posibilidad de ahorrar tiempo.

- El enfoque *Comet*: utiliza el protocolo denominado *Bayeux*. El cliente establece una conexión con el servidor y proporciona información al servidor de las conexiones de apoyo, más tarde, el cliente le dice al servidor el tipo de conexión que se prefiere utilizar, y finalmente se obtienen las actualizaciones que se necesitan. Una vez que la conexión se ha abierto, se sostiene que para un período de tiempo predefinido, el servidor comprueba la presencia de cualquier evento en el cliente, cuando salta el *time out* el servidor pedirá al cliente la posibilidad de volver a conectarse de forma asíncrona.

- Por último hay que mencionar la diferencia entre el enfoque push y pull (tira y afloja) en relación al tiempo medio de publicación: el uso del enfoque push proporciona una respuesta mucho más rápida que la de pull. También cabe mencionar que con el método push, el cliente recibe muchos mensajes inútiles y no tiene la certeza de que los mensajes llegan correctamente, esto implica un mayor movimiento en la CPU y una posible saturación, en según que casos sería preferible utilizar el método pull.

2.2.6 Ajax hacia la normalización

Aunque Ajax ha hecho mejoras importantes, hasta la fecha, aún no se ha reconocido como un estándar oficial. Sin embargo, todas las tecnologías que conforman el Ajax ya están estandarizados y, por tanto ampliamente detalladas en la correspondiente RFC.

Para Ajax actualmente están en curso varias propuestas, que quieren estandarizar un mecanismo tal que sean factibles los aspectos fundamentales de esta nueva modalidad de navegación. Una de las propuestas, por ejemplo, utiliza Bayeux, un protocolo que envía mensajes de forma asincrónica, es de tipo Server-push (servidor de empuje), que junto a una aplicación AJAX, permite crear Comet, como ya se ha mencionado antes.

2.2.7 Los méritos y defectos de Ajax

Ajax garantiza la facilidad de uso nunca antes experimentada. En cuanto a los aspectos positivos del uso de Ajax hay que mencionar sin duda que genera la menor cantidad de tráfico, en comparación al uso de estándares del protocolo HTTP, de hecho, dada una parte de la página a recargar, sólo las partes involucradas se actualizan.

En primer lugar, con la llegada de Ajax, es una creciente complejidad en el desarrollo de aplicaciones, debido al hecho de que todavía no existe un estándar para el uso de estas tecnologías. Además, aún es difícil asegurar que todos los usuarios sean capaces de utilizar JavaScript debido a que utilicen por ejemplo un navegador incompatible, o que dependa de la calidad del sitio,

Otra desventaja es que el Ajax puede hacer que sea difícil indexar un sitio por los motores de búsqueda, si la tecnología no se implementa correctamente. Ajax determina un problema ligado a los motores de búsqueda: es necesario que los robots de los buscadores puedan acceder a los contenidos del sitio y puedan ver y seguir su navegación. Con JavaScript esto no es siempre posible. Sin embargo, hay proyectos en curso para los motores más famosos de investigación, como Google y Yahoo, para superar este problema.

2.2.8 El modelo de tráfico de Ajax

Al igual que en el caso del tráfico HTTP en sitios Web 1.0, también en el caso del uso de Ajax se han propuesto varios modelos y análisis. Dada la reciente introducción de Ajax, los modelos disponibles son mucho menos numerosos que los patrones de tráfico HTTP estándar, y por esta razón, se proponen a continuación los resultados de los artículos más detallados y completos y que sobre todo, interesados en los objetivos fijados para este trabajo y entender mejor las tendencias de Ajax.

Los artículos que aquí se ofrecen, son cuatro: "*The impact of Ajax on network performance*" [13], "*The new Web: Characterizing Ajax traffic*" [14], "*A comparison of push and pull techniques for AJAX*" [15], y por último "*Web 2.0 traffic measurement: analysis on online map applications*" [16].

2.2.8.1 Descripción de las pruebas

En el artículo [13] no especifica los autores, ni el tipo de red de seguimiento, ni tráfico generado y ni el tiempo de captura. Sin embargo, dados los resultados obtenidos y las observaciones indican que la navegación se ha hecho en un solo servidor y un único host.

El análisis [14], sin embargo, se llevó a cabo en Mónaco. La red de Mónaco tenía una capacidad de: 10 Gbps, aproximadamente 55.000 hosts y una transferencia diaria de alrededor de 3.6 TB.

El propósito del artículo [15] fue probar la reacción del servidor en diferentes condiciones, para ello fue necesario crear dos aplicaciones idénticas, una con un enfoque de tracción y la otra con un enfoque de empuje; Posteriormente, se creó una aplicación que generaba una cantidad precisa de tráfico a intervalos predeterminados.

Los autores del articulo [16] se centran en el análisis de sitios Web 2.0. relacionados a la visualización de mapas como por ejemplo: *Google Maps, Yahoo Maps, Baidu mapas, Mapas Sogou* y otros. El fin de este análisis es el estudio de la popularidad de aplicaciones de visualización de mapas basadas en la tecnología Ajax por la rapidez en la descarga continua de imágenes.

2.2.8.2 La cantidad de tráfico

En [13] se ha estudiado el tiempo para recargar la página. La razón que llevó a los autores para reflexionar sobre este aspecto es que la innovación de Ajax, debe reflejarse, principalmente, en los tiempos de recarga más frecuentes, pero con una ocupación del ancho de banda menor. La Fig. 2.11 refleja que la diferencia entre los dos histogramas es inmediatamente evidente.

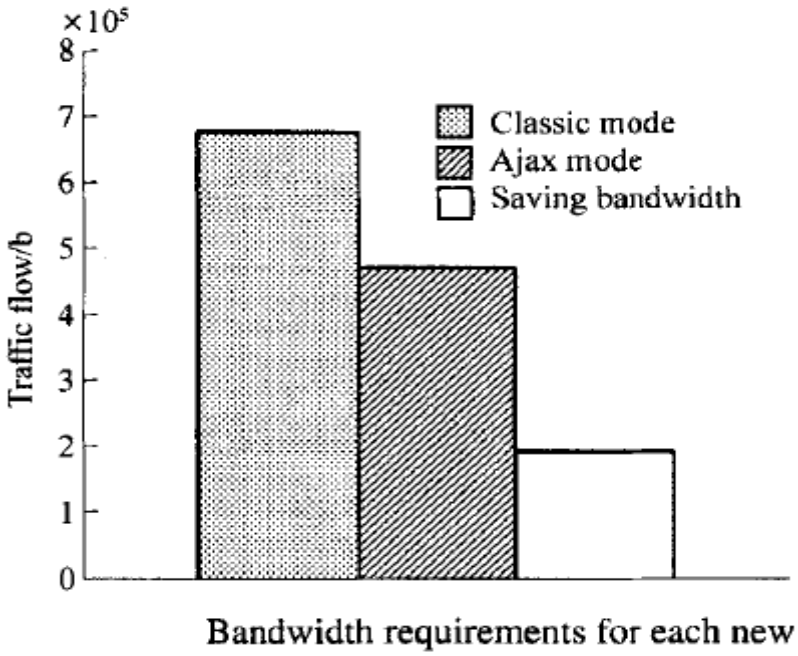


Fig. 2.11: Ocupación del ancho de banda con el modo clásico y con Ajax.

2.2.8.3 Request

A continuación se muestran las diferencias entre las curvas de densidad de probabilidad entre all-http y las otras cuatro curvas, en la Fig. 2.12 de [14].

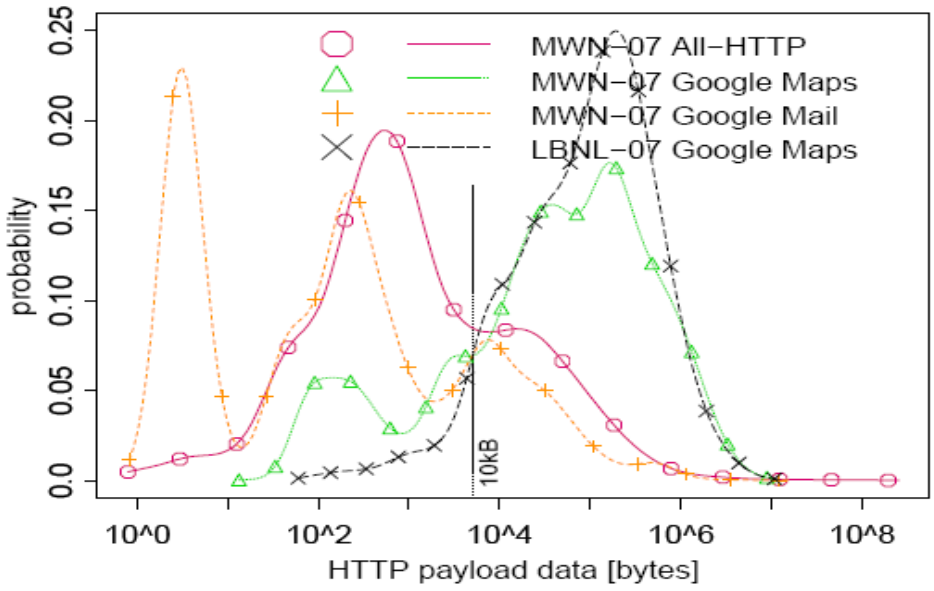


Fig. 2.12: FDP de bytes intercambiados en cada conexión.

Una vez más, la curva en referencia a *Google Maps* frente a todo HTTP muestra una tendencia ligeramente diferente, se observa que el número de solicitudes es mayor que en las otras páginas. La explicación que los autores del artículo proponen es relativa al hecho de que *Google Maps* es la aplicación que más que cualquier otra se beneficia del pre-fetching. Por lo tanto, es posible que el número mas elevado de solicitudes se deba a las solicitudes de datos próximos a aquellos utilizados en tiempo real del usuario.

Una vez más, se encontró que, usando Ajax, se presenta una transferencia más alta de solicitudes y por consiguiente de respuestas.

2.2.8.4 Las conexiones

Es interesante observar también la FDP relativa a la duración de varias conexiones en la Fig. 2.13. Las conexiones que tienden a ser más largas son sobre *Google Maps*, mientras que las otras tienden a ser más variable, con picos alrededor de 0.1 y 10 segundos.

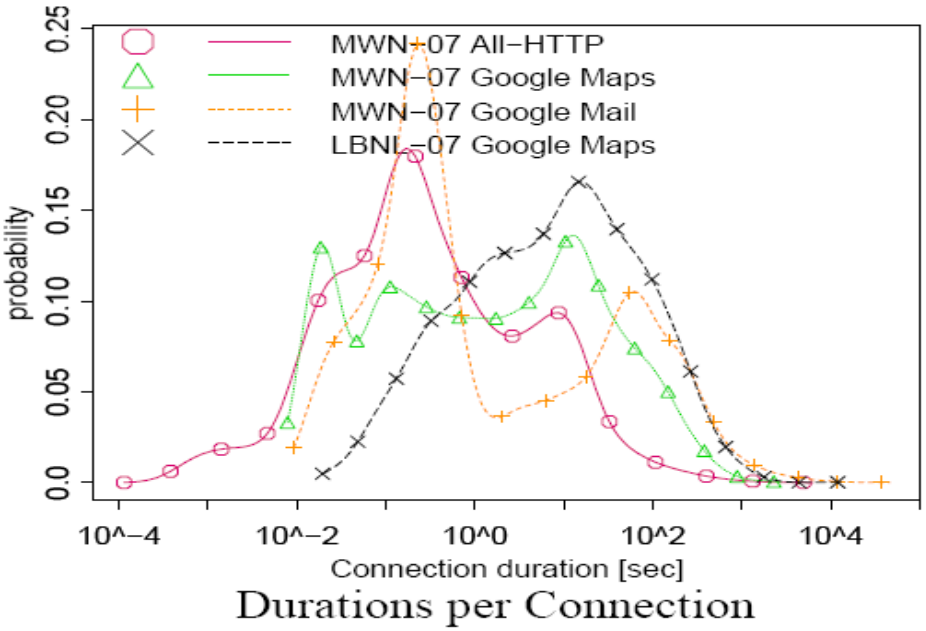


Fig. 2.13: FDP de la duración de la conexión.

2.2.8.5 El tiempo entre llegadas Request

En [14] También se ha estudiado el tiempo entre llegadas citado en el apartado sobre los modelos de tráfico de HTTP estándar. La evolución de la función de densidad de probabilidad del Inter-Request Time está representada en la Fig. 2.14:

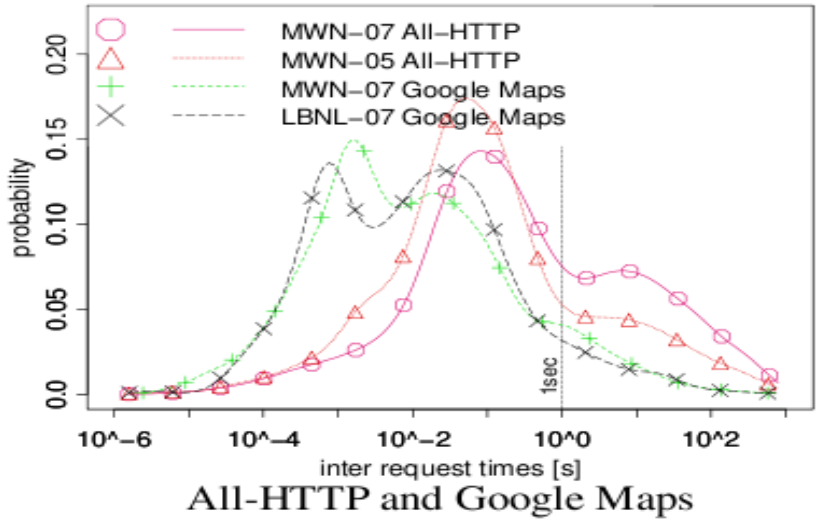


Fig. 2.14: FDP de Inter-Hora de la solicitud.

La mayor parte de las solicitudes, se generan automáticamente al cargar, se puede ver la línea vertical en las proximidades de 1 segundo, que se refiere, por supuesto, al tiempo de recarga. Sólo la curva por encima de esta línea se refiere a las solicitudes generadas por el usuario.

2.2.8.6 Selección del servidor

Por primera vez, los autores de los análisis en [16] muestran un histograma de la composición del tráfico Ajax, durante una semana. El gráfico muestra en la Fig. 2.15

que un 85% del tráfico generado es debido a la presencia de grandes imágenes, mientras el 15% restante aproximadamente es subdividida entre las páginas Html, Xml, CSS, JavaScript y otros componentes.

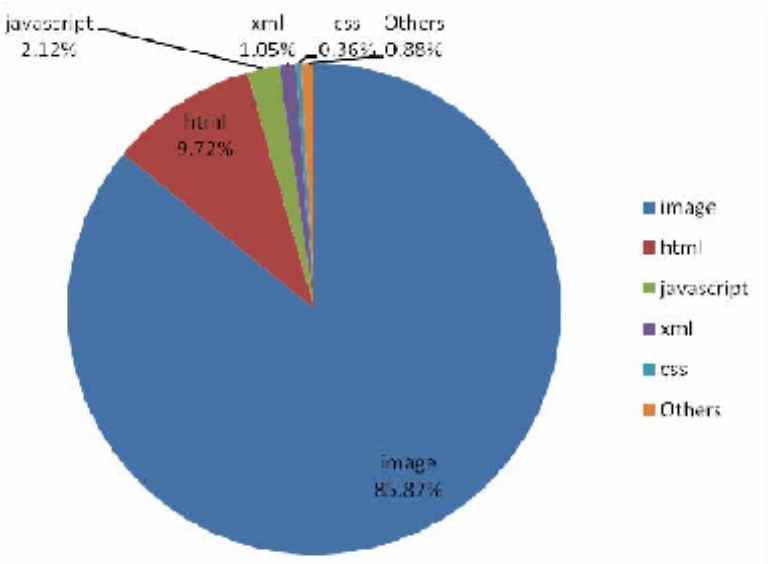


Fig. 2.15: Composición del tráfico utilizado Ajax.

Como dato curioso, se observa en la Fig. 2.16 que el tráfico monitorizado varía según los momentos del día. En particular, se basa en un estudio muy concreto de las páginas: *Google*, *baida* y *Ségou*, se ve que en las horas comprendidas entre las 1:00 y las 7:00 el número de sesiones es decisivamente más bajo respecto a otros momentos del día. El fenómeno es debido muy probablemente a las costumbres del lugar, como las horas de sueño o la hora de cenar. Si se tuviera una gráfica de un estudio semanal, por ejemplo, se vería que el viernes presenta una mayor actividad en páginas como *Google Maps* y atribuimos este aumento de peticiones a que la gente planea sus fines de semana con Internet.

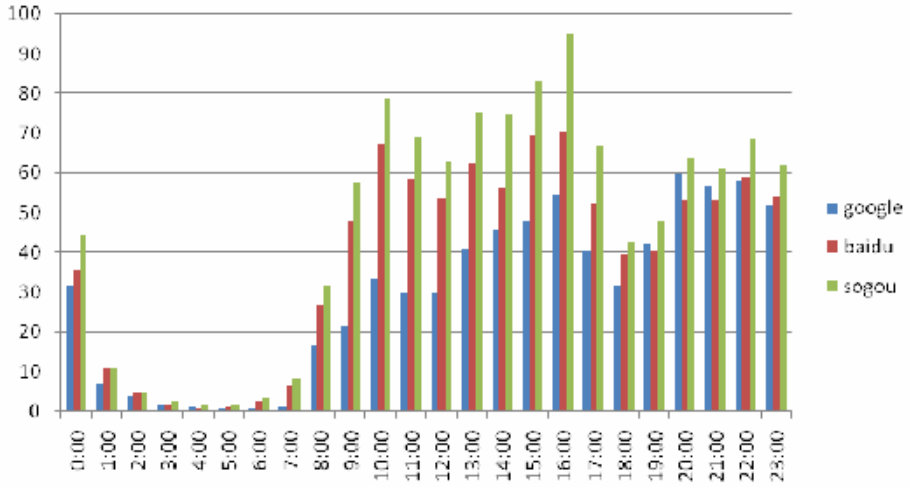


Fig. 2.16: Número medio de sesiones por cada hora.

3 Capítulo 3: Parámetros para caracterizar tráfico en la Web 2.0

3.1 Introducción

El propósito de este PFC es enriquecer los modelos propuestos en el capítulo anterior, para decidir cuál de estos enfoques se acerca más a la realidad que se observó en las pruebas llevadas a cabo y se describen a continuación.

Partimos de la navegación con HTTP, se pretende estudiar los modelos de la nueva navegación con Ajax. Por esta cuestión, era necesario analizar los modelos anteriores más importantes, y sobre todo poner atención en el contraste de cada uno de ellos. El control de las variables es fundamental para analizar el tráfico en diferentes frentes y cuestiones.

Para analizar el impacto real que Ajax tiene sobre el tráfico de la red, y sobre todo las diferencias con el funcionamiento normal de la Web 1.0, se han realizado una serie de pruebas, cada una de las cuales permitieron obtener información, estadísticas y modelos de utilidad. Las diferencias más relevantes entre las dos navegaciones son las siguientes:

En la Web 1.0 destaca: repositorio estático de la información, el rol del navegante es pasivo, la dirección de la información es unidireccional y su publicación requiere de conexión, la producción es individual y proviene de fuentes limitadas.

Mientras en el caso de la Web 2.0 es todo lo contrario: el usuario es generador de contenidos, puede editar y responder (blogs), el navegador tiene rol activo, comparte información, participa y trabaja colaborativamente, la interfaz es interactiva, posee múltiples fuentes de producción (wiki), y se forman comunidades de aprendizaje y conexión (como las Redes Sociales).

Todas las pruebas se dividieron en dos grupos dependiendo de la extensión envío o el objetivo final del análisis individual. En nuestro caso, los dos grandes bloques son con o sin interacción. Y la página seleccionada es Facebook, es una de las redes sociales mas importantes y utilizadas en el mundo, que es uno de los motivos revolucionarios de las Tecnologías que componen Ajax.

El primer grupo de pruebas consiste en una serie de capturas para medir el tráfico de red utilizada y caracterizar el tráfico con valores numéricos con interacción del usuario sobre la página Facebook. Mientras el segundo se centrará en no interactuar, bien con Ajax o sin Ajax.

A continuación resaltar las fortalezas y debilidades de los modelos introducidos en el capítulo anterior, y sugerir los nuevos avances en la distribución y en la probabilidad de cada uno de los valores estudiados.

3.2 Explicación de variables y parámetros

Para involucrar bien lo que hay que a medir, se comienza con la imagen de la Fig. 3.1, una ilustración básica que describe el funcionamiento de cualquier navegación, donde no se distingue que tipo de Web se utiliza. Puesto que las variables son las mismas, lo que realmente denota la diferencia es que la Web 1.0 es una comunicación unidireccional y la Web 2.0 se basa en la interactividad, dinamismo y peticiones asíncronas.

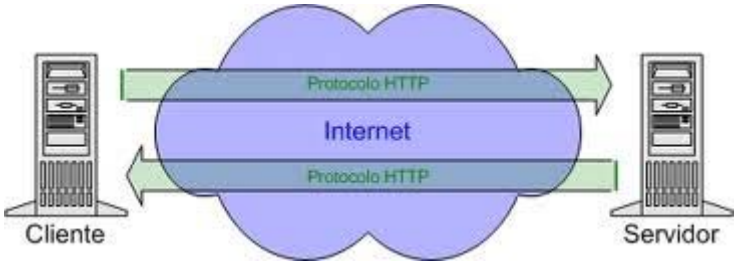


Fig. 3.1: Comunicación básica Cliente-Servidor.

La comunicación entre Cliente/Servidor se representa con un diagrama de intercambio de mensajes que se muestran a continuación. Entre estas dos piezas elementales es necesaria una comunicación continua en la que se intercambia mucha información de diversos protocolos. Para este estudio no se necesitan todas las variables de todos los protocolos. Se descartan los demás y hay que quedarse con el protocolo TCP.

Una vez establecida la conexión, el cliente enviará continuamente peticiones al Servidor a las que el servidor responderá: *Request* y *Reply* respectivamente. Estos son los segmentos interesantes para nosotros. Con la ayuda de la siguiente imagen, la Fig. 3.2, se entiende la comunicación que se acaba de describir.

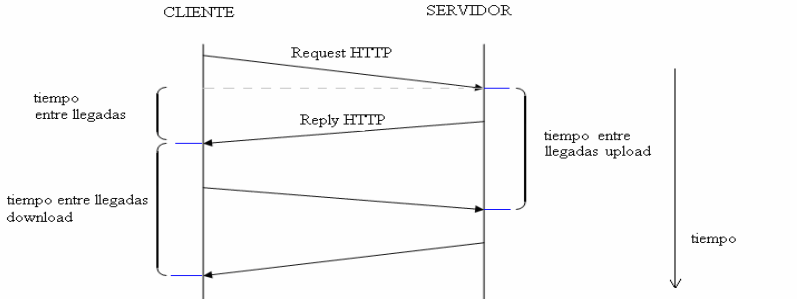


Fig. 3.2: Parámetros de las conexiones.

Si se observa la figura anterior se observa los diferentes tiempos que se pueden medir. Los instantes de tiempo en el que los mensajes se reciben pueden ser de tres tipos: tiempo entre llegadas, que mide la frecuencia con la que se intercambia información a ambos lados de la comunicación, el tiempo entre las peticiones que llegan al servidor (upload), y el tiempo entre las respuestas que llegan al cliente (download).

No puede haber dos conexiones iguales en un mismo instante en toda la Red. Aunque bien es posible que un mismo ordenador tenga dos conexiones distintas y simultáneas. El protocolo TCP utiliza el concepto de conexión para identificar las transmisiones. Tiene tres fases: establecimiento, transmisión y cierre. La duración de este proceso entre el cliente y el servidor es el tiempo de conexión.

Otra de las características que se puede obtener según la imagen es la cantidad de paquetes que se envían/reciben por conexión. A su vez, la dimensión de cada uno de los paquetes transmitidos, puesto que según los datos que aporten serán de un tamaño menor o mayor. Estas variables si se relacionan con el tiempo darán lugar a nuevas otras como el número de paquetes por segundo y bytes por segundo.

¿Por qué son importantes estos parámetros?, estos son los que nos indican, en cierto modo, la interacción que se ejecuta en una navegación entre el cliente y el servidor Web e, independientemente de la navegación esta comunicación se mide igual. Puesto que ya se sabe los principios de cada tipo de navegación: la Web 1.0 es unidireccional, mientras que la Web 2.0 proporciona interactividad y peticiones asíncronas, pero la comunicación Cliente-Servidor consta de los mismos elementos. Será importante analizarlos para ver cual de las navegaciones provoca más o menos movimiento. Nos permite tener una visión más detallada de lo que realmente ocurre durante una sesión del usuario con, en nuestro caso, la aplicación Facebook.

Una vez aclarado el funcionamiento básico entre Cliente y Servidor, será más fácil entender los parámetros principales con los que se van a trabajar para cada una de las tres opciones: upload, download y total, se enumeran a continuación:

- Paquetes por segundo.
- Desviación típica de paquetes por segundo.
- Bytes por paquete.
- Bytes por segundo.
- Número de paquetes.
- Tiempo de las conexiones.
- Desviación típica del tiempo de las conexiones.
- Diferencia de tiempo entre llegadas.
- Desviación típica de la diferencia de tiempo entre llegadas.

Y a raíz de estos parámetros se obtienen las gráficas estadísticas siguientes:

- Histograma de la dimensión de los paquetes.
- Histograma del tiempo de las conexiones.
- Histograma de la diferencia de tiempo entre llegadas.
- FDP del tiempo de las conexiones
- FDP del tamaño de los paquetes.

4 Capítulo 4: Herramientas para el entorno de pruebas y modelado

4.1 Introducción

Antes de empezar a explicar los pasos realizados para efectuar las pruebas con dicho programa, se han seguido unas instrucciones necesarias para que las circunstancias fueran propicias al margen del buen uso del Wireshark, se citan a continuación:

En un principio este trabajo no estudia la navegación según la hora del día, ni que las pruebas tengan que ser ejecutadas al mismo tiempo, por lo tanto no nos regimos a ninguna franja horaria ni a un acuerdo entre varios usuarios a la vez.

Es importante que la conexión a Internet sea directamente mediante cable y no con Wi-Fi, se necesita que se capturen todos los paquetes sin intermediarios.

Mientras las pruebas están siendo efectuadas, no se puede abrir ninguna otra pestaña o ventana con el navegador, ni por supuesto utilizar otro navegador, se capturarían paquetes dirigidos a diferentes páginas que estropearían los resultados de nuestras pruebas. Sólo se centrarán en el tráfico existente de TCP y HTTP entre nuestra máquina y la red exterior durante la navegación que según las opciones que se han seleccionado será de Ajax o navegación clásica.

Se ha elegido el navegador *Firefox Mozilla*, es un navegador Web libre y de código abierto descendiente de Mozilla Aplication Suite y desarrollado por la Fundación Mozilla. Es fácil añadir funciones a través de complementos y opciones entre los que hay una amplia selección, lo que lo convierte en el navegador más personalizable y seguro del momento. En nuestro caso bastará con seleccionar en el menú "Herramientas", después "Opciones" y dentro de ellas "Contenido", dependiendo de si se selecciona la casilla Activar JavaScript o no, se estará en una navegación Web 1.0 o Web 2.0, a modo práctico, cuando se desactiva esta opción, se demuestra que no muestra la opción de activar o desactivar el chat al que nos tiene acostumbrados Facebook.

Para adecuar las circunstancias y recomendaciones debidamente también tiene que garantizarse que ninguna otra información que no nos interese intervenga, pero normalmente, por comodidad en nuestras vidas cotidianas se tienen programas que se abren automáticamente al encender el ordenador (*Skype*, *Dropbox*, …) o la existencia de otros programas que son simplemente necesarios como sería el caso de los antivirus y sus actualizaciones, o actualizaciones de otro tipo. Para ello, hay que ir hasta el Panel de Control y deshabilitar las actualizaciones automáticas antes del comienzo de las capturas. Y obviamente, cerrar los programas mencionados con anterioridad en el caso de que se tuviera alguno que se abriera sin necesidad de arrancarlo manualmente.

Una vez llegados a este punto, se está preparado para la realización de las pruebas, no sin antes explicar brevemente los programas que se van a utilizar y su función. Se ha seleccionado como herramienta para nuestras capturas de tráfico el programa Wireshark, conocido originalmente como *Ethereal*, su principal objetivo es el análisis de tráfico además de ser una excelente aplicación didáctica para el estudio de las comunicaciones y para la resolución de problemas de red, además cuenta con todas las características estándar de un analizador de protocolos. La funcionalidad que provee es similar a la de tcpdump, pero añade una interfaz gráfica y muchas opciones de organización y filtrado de información.

Para la primera parte del análisis de los datos resultantes de las capturas, que se basa en hacer estadísticos como las medias, desviaciones típicas, histogramas y funciones, se puede utilizar cualquier versión de Matlab, pero para trabajar con las funciones relacionadas con modelos de Markov se necesitará una versión de este programa suficientemente moderna, a partir de la versión 7.1 ya contiene la librería *Statistics Toolbox.*

4.2 Procedimiento de la captura de las pruebas

El primer punto es definir el escenario de navegación, que se verá en el capítulo posterior, en este capítulo se explicará la elaboración y desarrollo de las pruebas de forma que sean válidas no sólo para este caso en particular, sino para todos los estudios relacionados con este campo. Una vez determinado el caso práctico, se procederá de la siguiente manera:

La primera de las opciones de Wireshark de la que se parte está dentro de "*Capture Options*": y deseleccionar la casilla "*Capture packets in promiscuous mode"*, opción bastante importante. Al estar seleccionada Wireshark captura todos los paquetes que la interfaz recibe/envíe. Cabe hacer una pequeña aclaración: cuando tu equipo está conectado a través de un hub, la tarjeta de red recibe todos los paquetes que transmitan/reciban los equipos conectados al mismo hub. Esto es porque, cuando el hub recibe un paquete lo reenvía a todos los puertos conectados, y es el computador quien decide que hacer con ellos (si el paquete es para él, lo recibe; si el paquete es para otro equipo, lo ignora). Esto no sucede así cuando se usa un Switch, puesto que cuando se usa una red switcheada se verifica el destinatario del paquete, antes de enviarlo, en cualquier caso no interesa seleccionar el modo promiscuo. Se aplica un filtro, que será "*not arp*" aunque al recoger los datos se utilizará un filtro bastante más estricto, de momento, para la inicialización de las pruebas bastará con este. Por último nos queda ajustar el tiempo que se desea que el programa esté capturando, se han realizado tanto pruebas de 10 minutos hasta de 2 horas, pero finalmente para este caso, las pruebas más concluyentes y con las que finalmente se han trabajado han sido de 40 minutos, aunque dependerá del escenario de navegación que se quiera estudiar. Una vez todo preparado, clickar *Start* y no hacer nada con el programa hasta la recolección de los datos en una fase posterior, se dejará que capture el tiempo definido.

4.3 Procedimiento de medidas

El software Wireshark, proporciona una serie de comandos para extraer cualquier tipo de información sobre el tráfico de red capturado durante una sesión de navegación.

Cada paquete capturado, puede ser analizado en términos de longitud, en bytes, contenido, origen, destino, el formato de mensajes, y mucho más.

Del esquema de la Fig. 4.1 se parte para entender la extracción de los datos y el desglosamiento de las diferentes gráficas y tablas, que ayudarán a entender mejor la sistemática, explicada detalladamente en los párrafos posteriores:

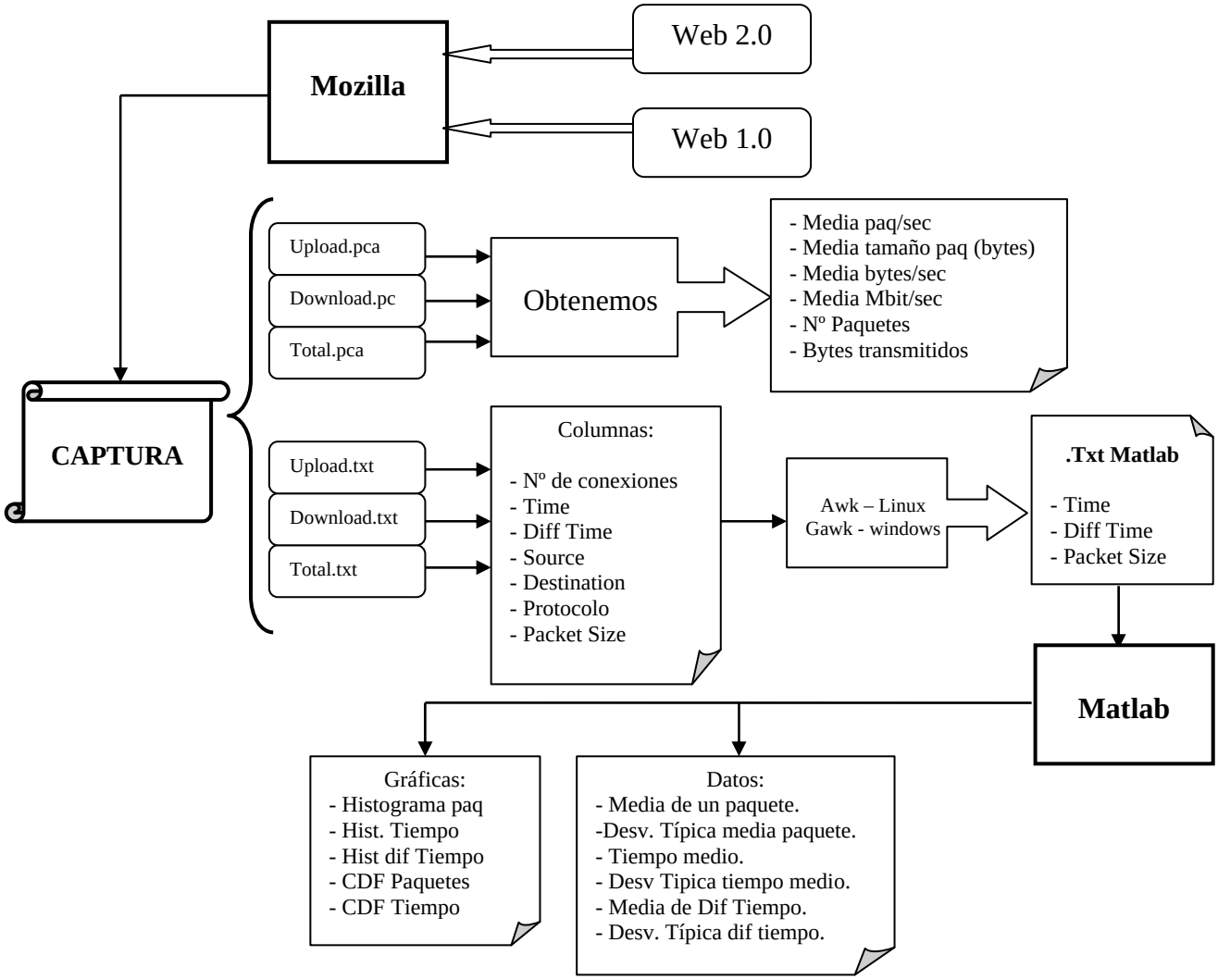


Fig. 4.1: Esquema de la obtención de datos.

Para que todo se vea mucho más claro y para una mejor compresión del trabajo realizado, se mostrarán ejemplos de los diferentes elementos mencionados en el esquema anterior. Se parte de la determinación del escenario de navegación sobre la aplicación Web que se desee. Lo primero a visualizar es la ventana de una captura ya limpia, como la de la Fig. 4.2, para llegar hasta ella, se procederá así:

Transcurrido el tiempo necesario determinado para las pruebas, hay que guardar los datos debidamente para su posterior uso, además de guardarlo como .pcap también lo se salva con el formato .txt, de este modo seguimos teniendo la captura con toda la información (.pcap) y una forma más cómoda y útil para trabajar con otros programas de análisis y caracterización (.txt). Es importante mencionar, que primero se guardarán

los .pcap por cada prueba y a partir de estos sus correspondientes .txt. Si se guardan los archivos de texto directamente de la original, es muy frecuente encontrarse que dentro del archivo la información no está organizada por columnas, que es como interesa.

Para eliminar los restos de paquetes de otros protocolos y asegurarnos de que no se hayan colado paquetes TCP o HTTP de otros elementos se realiza el siguiente filtrado. Abrimos cada una de las capturas y se procederá del siguiente modo:

En la opción "*Conversations*" del menú "*Statistics*" se observa que la pestaña en la que se está sea *Ethernet*. Se selecciona la conexión que más tráfico haya cursado (porque se tratará del usuario y la aplicación en cuestión), se aplica un filtro que eliminará todo el tráfico que no concierne al usuario con la página precisa distinguiendo las MACs o IPs. A la hora de seleccionar, hay que saber que estas son las 3 opciones que se van a salvar:

A < - > B es una selección total de los paquetes transmitidos (total.pcap y total.txt).
A -> B los paquetes transmitidos por el usuario (upload.pcap y upload.txt).
A <- B los paquetes transmitidos por el servidor (download.pcap y download.txt).

Ahora sólo quedará filtrar los demás protocolos que no tienen que aparecer y entorpecen el análisis, a continuación se muestra un ejemplo de la línea de filtros a aplicar que se añade tras la aplicación del filtro anterior.

```
(eth.dst==00:11:d8:9c:bf:d6 && eth.src==00:17:9a:fa:81:d5)and not arp
and not udp and not llc and not icmp and not sna and not dns and not
snmp and not ipx
```

Si se observa la captura ahora se ve que sólo aparecen las conexiones del protocolo TCP, HTTP (como se pretende) y otros como TLSv1, que es un protocolo de seguridad: *Transport Layer Security* (TLS; seguridad de la capa de transporte) que pertenece a un grupo de protocolos que son criptográficosy que proporcionan comunicaciones seguras por una red.

Entrando en *Edit -> Preferences -> Columns*, se pueden añadir y eliminar columnas. Se añade la columna *DiffTime* y se pone como tipo de campo "*DeltaTime*" que se mide en segundos. Se quitan las columnas que no nos dejan extraer los datos de una forma cómoda como es la columna de información. Por defecto Wireshark muestra ya las columnas de número de conexiones, tiempo, fuente, destino, protocolo y tamaño de paquete.

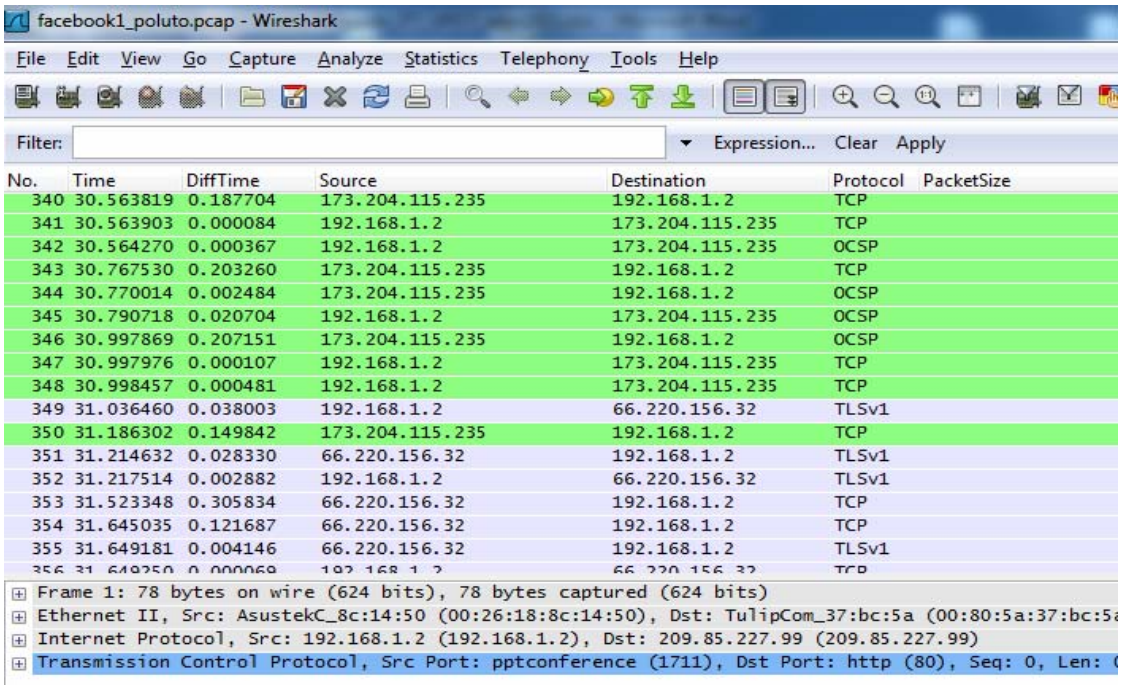


Fig. 4.2: Ventana de información del Wireshark

Directamente del Wireshark se pueden obtener valores de interés, basta con entrar en *Statistics -> Summary* y entre otras variables se encuentra:

- Número de paquetes.
- Media paquete/segundo.
- Media del tamaño de paquete (bytes).
- Media bytes/segundo.
- Media Mbit/segundo.

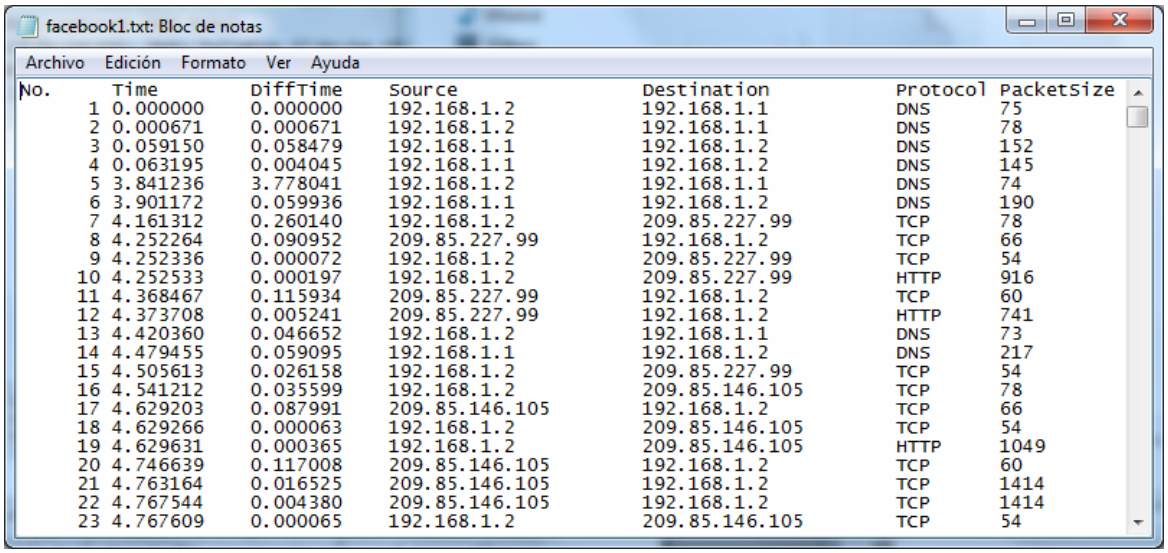
Todas estas variables las se van a ir guardando en un Excel, uno por cada prueba, en él se encontrarán las variables extraídas y la organización de las gráficas resultantes para upload, download y total, como muestra el ejemplo de la Tabla 4.1, pero en este punto, aún no está completa.

PROVA4	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	0,266	0,293	0,559
media tamaño paq (bytes)	323,461	581,962	459,12
media bytes/sec	85,944	170,748	256,675
media Mbit/sec	0,001	0,001	0,002
media de un solo paq	323,4607	581,9622	459,1198
desv. tipica de un solo paq	437,3826	587,7915	537,3073
media tiempo	634,4575	644,9791	640,0242
desv. Tipica tiempo	751,8609	761,2551	756,5354
media dif. Tiempo	3,7636	3,4083	1,7887
desv. Tipica dif. Tiempo	11,0145	10,5157	7,7831
histograma del nº de paq	623 (fig1)	688 (fig6)	1311 (fig11)
histograma del tiempo	fig2	fig7	fig12
histograma de la dif. De T	fig3	fig8	fig13
Grafica Paq	fig4	fig9	fig14

Grafica Tiempo	fig5	fig10	fig15

Tabla 4.3: Excel resumen de los datos de una prueba

De cada prueba se guarda la original (que es la que contiene toda la información, incluida la prescindible), la "limpia" tras el filtrado que será la total.pcap (para A<->B), la upload.pcap (A->B) y download.pcap (A<-B). A partir de estos, se guardan los ficheros como .txt, dentro de *File -> Export -> File* se selecciona la casilla *Displayed* y *Packet Sumary Line*, se guardan en el fichero los datos a los que se les ha aplicado el filtro, el fichero presenta la forma de la Fig. 4.3, evidentemente no se muestra entera porque tienen un número de filas alto, en este caso, por ejemplo es de 4816 filas:


facebook1.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

No.	Time	DiffTime	Source	Destination	Protocol	PacketSize
1	0.000000	0.000000	192.168.1.2	192.168.1.1	DNS	75
2	0.000671	0.000671	192.168.1.2	192.168.1.1	DNS	78
3	0.059150	0.058479	192.168.1.1	192.168.1.2	DNS	152
4	0.063195	0.004045	192.168.1.1	192.168.1.2	DNS	145
5	3.841236	3.778041	192.168.1.2	192.168.1.1	DNS	74
6	3.901172	0.059936	192.168.1.1	192.168.1.2	DNS	190
7	4.161312	0.260140	192.168.1.2	209.85.227.99	TCP	78
8	4.252264	0.090952	209.85.227.99	192.168.1.2	TCP	66
9	4.252336	0.000072	192.168.1.2	209.85.227.99	TCP	54
10	4.252533	0.000197	192.168.1.2	209.85.227.99	HTTP	916
11	4.368467	0.115934	209.85.227.99	192.168.1.2	TCP	60
12	4.373708	0.005241	209.85.227.99	192.168.1.2	HTTP	741
13	4.420360	0.046652	192.168.1.2	192.168.1.1	DNS	73
14	4.479455	0.059095	192.168.1.1	192.168.1.2	DNS	217
15	4.505613	0.026158	192.168.1.2	209.85.227.99	TCP	54
16	4.541212	0.035599	192.168.1.2	209.85.146.105	TCP	78
17	4.629203	0.087991	209.85.146.105	192.168.1.2	TCP	66
18	4.629266	0.000063	192.168.1.2	209.85.146.105	TCP	54
19	4.629631	0.000365	192.168.1.2	209.85.146.105	HTTP	1049
20	4.746639	0.117008	209.85.146.105	192.168.1.2	TCP	60
21	4.763164	0.016525	209.85.146.105	192.168.1.2	TCP	1414
22	4.767544	0.004380	209.85.146.105	192.168.1.2	TCP	1414
23	4.767609	0.000065	192.168.1.2	209.85.146.105	TCP	54

Fig. 4.4: Formato de los textos I.

En estos textos se ha guardado información útil como las IPs y protocolos para el seguimiento y confirmación de que todos los datos eran correctos, pero una vez confirmado, y para el procesamiento de los datos con la siguiente herramienta (Matlab) será conveniente desprenderse de ellos, y mantener las columnas de tiempo, diferencia de tiempo entre llegadas y tamaño de paquetes. Para ello, una de las formas posibles para ejecutar este cambio en los archivos de texto, es instalar el programa *gawk*, que es como el *awk* para *Linux*, se hace un script .bat con comandos sencillísimos que elimina las columnas innecesarias. Se muestra y se ve con claridad que lo que hace esta línea de comandos es coger del documento input.txt las columnas señaladas y las guardarlas en output.txt:

```
awk "{ print $2, $3, $7 }" input.txt > output.txt
```

Obteniendo como resultado ficheros como el que se muestra en la Fig. 4.5.

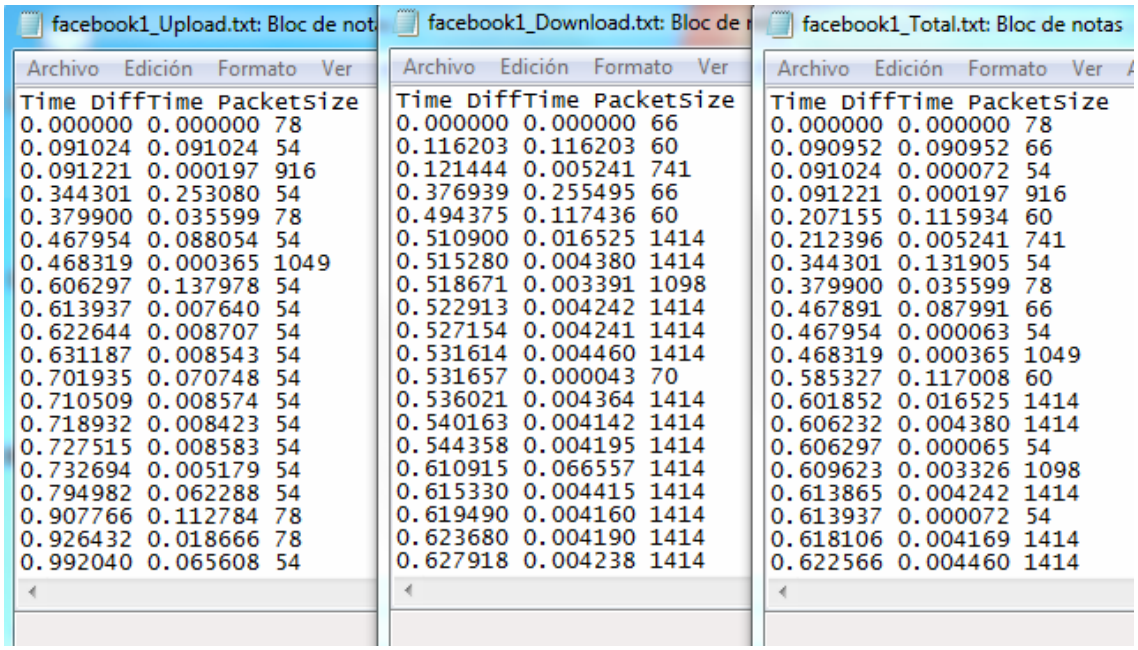


Fig. 4.5: Formato de los textos II.

Una vez hecho esto con los tres .txt de todas las pruebas se está preparado para la obtención de los demás parámetros con Matlab. El paso a seguir, es importar uno a uno los ficheros, de forma manual sería prácticamente imposible, pero seleccionando *ImportData* dentro de *File* en el menú principal de Matlab se salvan los datos en forma de matriz rápidamente, teniendo en cuenta que sólo se necesita marcar la opción “matriz Data”.

Se ha creado un .m, los ficheros con los que trabaja Matlab, para la optimización de los comandos. El script escrito, que se muestra a continuación, calcula las medias de las tres columnas, la desviación típica y después los histogramas en el siguiente orden: paquetes, tiempo y diferencia de tiempo.

```
mean(data(:,3))
std(data(:,3))
mean(data(:,1))
std(data(:,1))
mean(data(:,2))
std(data(:,2))
figure(1)
hist(data(:,3))
figure(2)
hist(data(:,1))
figure(3)
hist(data(:,2))
```

Los histogramas son gráficos que muestran la distribución de una serie de datos en función a su frecuencia. Formados por barras verticales cuya anchura corresponde con el rango del intervalo, Matlab divide el rango de los datos en 10 intervalos igualmente espaciados, y la altura representa el número de datos que están dentro de cada intervalo. Como representación de los histogramas se muestra la Fig. 4.6 de ejemplo. Corresponde a fig11, fig12, fig13 de la tabla 4.1:

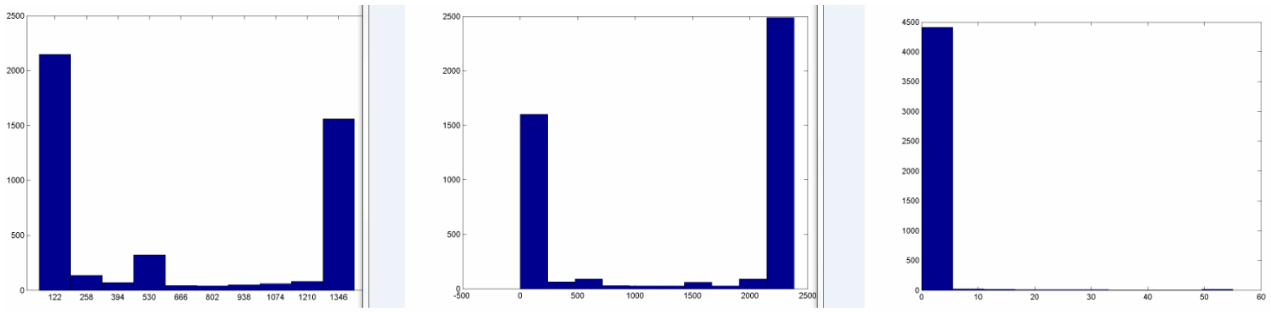


Fig. 4.6: Histograma del número de paquetes, tiempo y diferencia de tiempo Total

Matlab utiliza una función ya definida como CDF de las siglas del inglés (*Cumulative Distribution Function*), que es la función de densidad de probabilidad (FDP). Es una función matemática que caracteriza el comportamiento probable de una variable aleatoria. Si *f* en una función de densidad de probabilidad para una variable aleatoria X, la función de distribución acumulativa o CDF asociada llamada F es:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt$$

Una función CDF tiene las siguientes propiedades: El rango de la función es de 0 hasta 1 y si $y>x$, entonces el CDF de y es mayor o igual que el del CDF de x.

Cuando se ejecuten las funciones cdf.m para los paquetes y cdfb.m para la duración de las conexiones, se obtendrán las funciones gráficamente.

```
function [mean_a,stdev_a] = cdf(data)
[CDFa,xa]= ecdf(data); %Empirical (Kaplan-Meier) cumulative
distribution function.
figure(1);             %CDFa es el vector de valores de la empirical
cdf evaluado en xa
plot(xa, CDFa);
title ('Experimental CDF for the duration of the connections');
xlabel('Duration of the connections (s) ');
ylabel('CDF');
mean_a = mean(CDFa)
stdev_a = std(CDFa)
```

```
function [mean_a,stdev_a] = cdfb(data)
[CDFa,xa]= ecdf(data); %Empirical (Kaplan-Meier) cumulative
distribution function.
figure(1);             %CDFa es el vector de valores de la empirical
cdf evaluado en xa
plot(xa, CDFa);
title ('Experimental CDF for the packets size');
xlabel('Packets size (bytes)');
ylabel('CDF');
mean_a = mean(CDFa)
stdev_a = std(CDFa)
```

Se ejecuta la línea de comandos para que se representen las gráficas:

```
[mean_a,stdev_a]= cdfb(data(:,3))
[mean_a,stdev_a]= cdf(data(:,1))
```

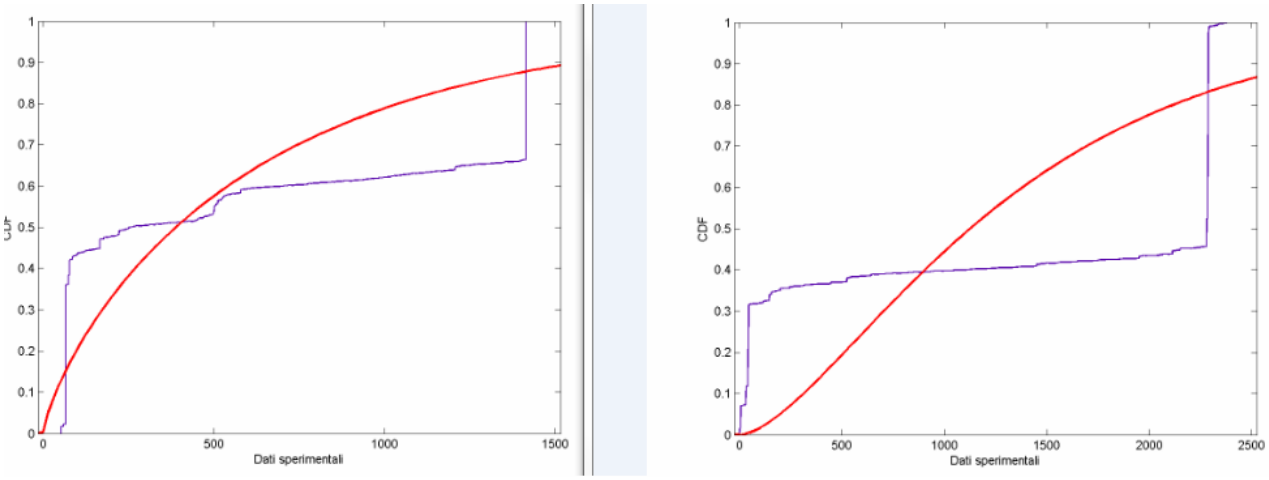


Fig. 4.7: CDF del número de paquetes y del tiempo respectivamente.

En resumen, Wireshark ofrece la posibilidad de exportar los resultados de la captura en un archivo genérico de texto. Esta opción bastante estándar, ha permitido la extracción de datos de interés para hacer análisis. En particular, los datos fueron agrupados adecuadamente con un sencillo script Windows y en un segundo momento procesados con el software Matlab, lo que permitió el cálculo de las variables básicas y la generación de gráficos experimentales y modelos relacionados. En el siguiente capítulo se describen los resultados de los datos experimentales con Matlab.

4.4 Procedimiento para obtener cadenas de Markov

Los procesos del mundo real, generalmente, producen salidas que se pueden caracterizar por señales discretas o continuas, estacionarias (en el caso de que las propiedades estadísticas no varíen con el tiempo) o no estacionarias, puras o perturbadas por otras señales (como por ejemplo, el ruido).

Muchas veces es útil contar con las instrumentos matemáticos que proporcionan las bases para una descripción teórica de la señal en cuestión, o incluso generar una señal similar a la original (cuando por ejemplo, el costo de la extracción de la señal real es alto). Estas herramientas matemáticas se denominan con el nombre de modelos y se dividen en modelos determinísticos y estadísticos. Los modelos determinísticos, en general, explotan algunas propiedades específicas o señales conocidas (como por ejemplo que la señal sea una onda sinusoidal, o la suma de exponenciales,..) En estos casos, la caracterización de un modelo es claro, todo lo que se requiere es estimar los valores de los parámetros del modelo (amplitud, frecuencia y fase de una onda sinusoidal,...). Los modelos estadísticos, sin embargo, tratan de caracterizar sólo las propiedades estadísticas de la señal. Ejemplos de modelos estadísticos incluyen los procesos de Gauss, Poisson, de Markov, ocultos de Markov y otros. El propósito de nuestro trabajo, se ocupará de los procesos ocultos de Markov o Hidden Markov Models (HMM).

En este tipo de modelos, el sistema produce el estado según la probabilidad adecuada. Los símbolos representan componentes observables y mediante el uso de la observación de estas componentes se puede retomar probabilísticamente y fiablemente el estado.

Los HHM, por lo tanto, representan un proceso estocástico múltiple que no puede ser directamente observado, sino a través de otros procesos estocásticos que producen la secuencia de observación. Para explicar mejor este concepto, se supone que de una observación concreta, se pueden obtener las probabilidades de cómo va a ser la secuencia, que correspondería con la matriz de transición entre los estados.

A continuación se expone un esquema en la Fig. 4.8 con el que se entenderá el proceso a seguir para obtener la cadena de Markov.

Con este fin, se utilizan dos funciones de Matlab de la librería Statistics Toolbox: hmmtrain y hmmgenerate, no sin antes haber discretizado la secuencia con valores de entre 1 a N, por exigencias de estas funciones que se explican a continuación. La matriz de probabilidad de transmisión debe ser de NxN, y la de emisión de NxM, el número de estados de la secuencia, que como ya se ha dicho será de N, hay que probar con 5, con 10 , 15 y con 20, por lo que se tendrá que discretizar a 5, 10, 15 y 20 la señal.

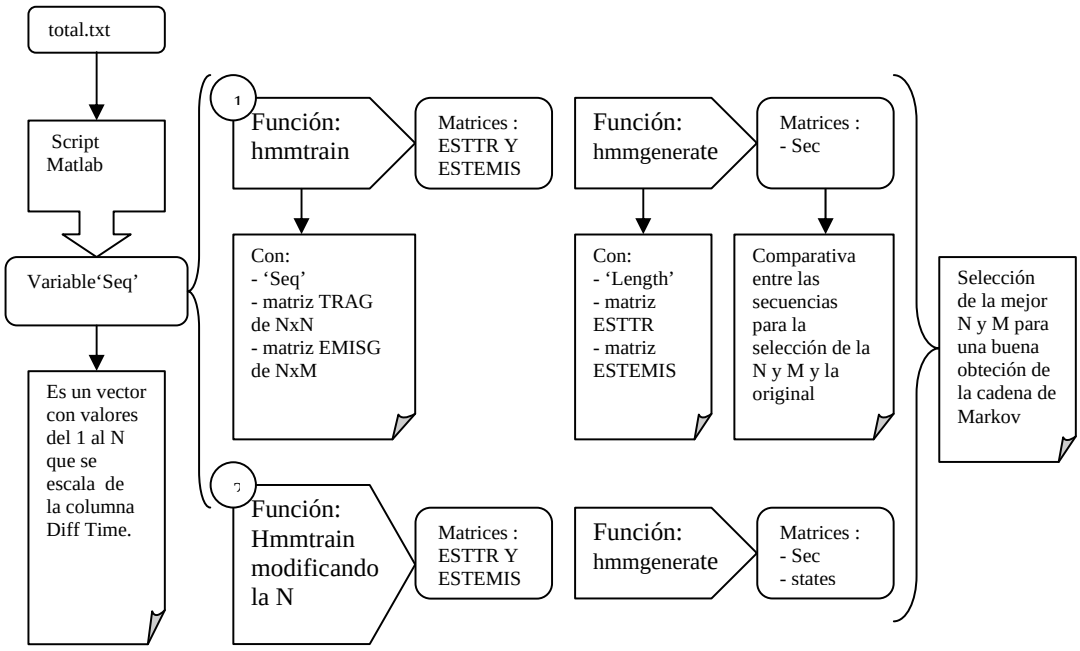


Fig. 4.8: Esquema para la obtención de las cadenas de Markov.

Para que el resultado final sea más preciso, lo que se hará es crear otro pequeño script para unir todas las columnas de DiffTime de todas las pruebas, decidiendo de antemano si el objeto de estudio es en general (total.txt) o por lo contrario algo más específico (upload.txt o download.txt), con el simple comando del que ya se hizo uso para extraer las tres columnas en secciones anteriores, da como resultado el fichero de la imagen de la Fig. 4.9 que consta de 3600 valores:

```
awk "{ print $2}" input.txt > output.txt
```

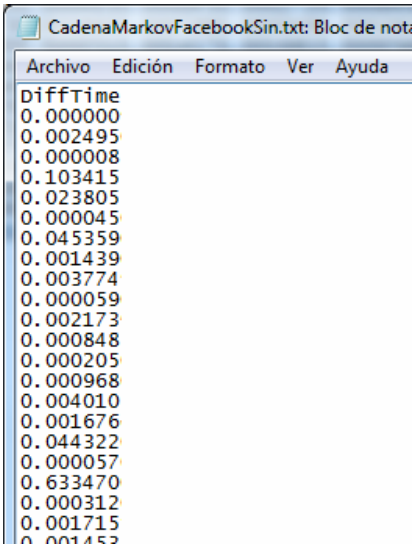


Fig. 4.9: Fichero con el tiempo entre llegadas total de las pruebas.

Una vez unificado los datos en un solo fichero, se vuelve a importar a Matlab como matriz de nuevo y ejecutar estas sencillas líneas de comando en Matlab para escalar las secuencias recogidas en el archivo de texto, siendo N = (5, 10,15, 20):

```
seq=data(:,1)
a=max(seq)/N
b=(seq./a)+1
m=max(b)
e=m/N
SecuenciaIntermedia=b/e
SecuenciaOriginal=int16(SecuenciaIntermedia)
```

Cuando se discretiza la señal lo que realmente se hace, es definir 1, 2,… hasta N, como estados de la secuencia, perteneciendo cada uno de ellos a un intervalo de tiempo (el correspondiente a la diferencia de tiempo entre llegadas). Por eso el estado 1, como se comprobará más adelante es el más probable de todos, porque el tiempo entre llegadas es más bien pequeño, pero existen esperas de hasta 25 segundos , que también hay que caracterizar.

En el caso de N=10, el mínimo y el máximo de la señal son : 0,000001 y 25.0187 respectivamente. Si la diferencia entre los valores la dividimos entre los 10 estados, se definen cada uno de los estados

Si se observa la siguiente Fig. 4.10, se ven los N estados, con las M probabilidades de cada uno:

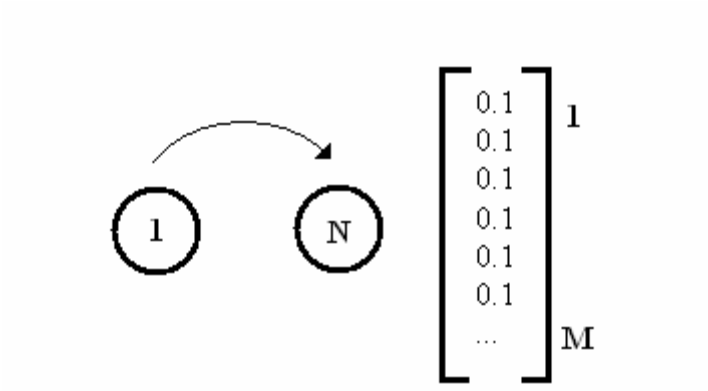


Fig. 4.10: Esquema de las probabilidades y estados.

La función [ESTTR,ESTEMIT] = hmmtrain(seq,TRGUESS,EMITGUESS): Estima las probabilidades de transición y de emisión de un modelo de Markov utilizando el algoritmo de Baum-Welch. 'Seq' es un vector columna que contiene una sola secuencia, una matriz con una fila por cada secuencia, 'TRGUESS' (i,j) es la probabilidad estimada de transición del estado de i al estado j. Y 'EMITGUESS' (i,k) es la probabilidad estimada de que el símbolo k se emite desde el estado i. Se crea TRGUESS, EMITGUESS con unas dimensiones de NxN y NxM correspondientemente.

En el caso de que N=5 : habrá 5 columnas por 5 filas con el mismo valor de 0,2 de probabilidad. Para el caso de N=10 : habrá 10 columnas por 10 filas con el mismo valor de 0,1 de probabilida. Para N=15 : habrá una matriz de 15x15 con una probabilidad de 0,066. Para n=20 : habrá una matriz de 20x20 de 0,05 de probabilidad, puesto que son estados sin memoria y que todos los eventos tienen la misma probabilidad de ocurrir, teniendo que sumar 1 tanto a las filas como las columnas.

M tiene un valor de 10 que no se va a modificar porque no es la variable que se tiene que justificar. Para la matriz de emisión NxM los valores serán todos de 0.1, como se ha señalado en la figura anterior.

La función [seq] = hmmgenerate(len,TRANS,EMIS) toma un modelo conocido de Markov, especificada por las matrices de probabilidad de transición TRANS y de emisión EMIS, se utilizan los resultados de la función anterior para generar la nueva secuencia. La longitud de seq y estados es 'len'. TRANS (i,j) es la probabilidad de transición del estado i al estado j. Y EMIS (k,l) es la probabilidad de que el símbolo l se emita desde el estado k.

De este modo, se consigue que a partir de una "muestra" de una secuencia, se pueda obtener una fiel reproducción de una secuencia parecida a la original, seleccionando finalmente el valor adecuado para la variable con la que se experimenta.

5 Capítulo 5: Análisis de los resultados para el caso práctico: Facebook

5.1 Diseño experimentos

Los modelos de diseño de experimentos son modelos estadísticos clásicos cuyo objetivo es averiguar si unos determinados factores influyen en una variable de interés y de cuantificar dicha influencia. La metodología del diseño de experimentos se basa en la experimentación. Es sabido que si se repite un experimento, en condiciones indistinguibles, los resultados presentan una cierta variabilidad. Comparar las respuestas en diferentes niveles de observación, con distintos condiciones, de las mismas variables será nuestro objetivo.

El presente capitulo se ocupa de enunciar los resultados principales obtenidos en los análisis de las pruebas anteriormente explicadas. Además, en el curso del capitulo poco a poco se indican las diferencias básicas de la navegación basada en Ajax con la navegación clásica y una comparativa exhaustiva de todos los parámetros implicados y descritos.

5.2 Escenario de navegación

El sitio Web que se ha seleccionado para navegar es Facebook. Facebook es un sitio web de redes sociales creado por Mark Zuckerberg y fundado junto a Eduardo Saverin, Chris Hughes y Dustin Moskovitz. Originalmente era un sito para estudiantes de la Universidad de harvard, pero se abrió a cualquier persona con una cuenta de correo electrónico. Facebook cuenta con más de 1000 millones de usuarios por todo el mundo y traducido a 70 idiomas. La versión principal está hecha con Ajax, sin embargo, está disponible una versión en HTML que no requiere Ajax.

La red en la que fue capturado el tráfico es la red del Laboratorio de Electrónica y Telecomunicaciones de La Universidad de Pavía.

El tráfico capturado en cada una de las pruebas es el tráfico desde y hacia el host en el que se ha instalado el software y por esta razón, todos los análisis que siguen son referentes al envío de un solo usuario.

En muchos de los análisis anteriores se presta mucha atención al día y la hora del día en que se efectuó la captura del tráfico, en este caso, en el momento que la navegación fue programada y no fue producida casualmente en términos de número de hosts, los usuarios conectados, los sitios visitados, etc. es irrelevante especificar la fecha y la hora en que se efectuaron las capturas.

Aunque en un primer momento se hicieron numerosas pruebas con diferentes especificaciones, algunas de ellas resultaron no ser necesarias, y otras no eran quizás lo suficientemente interesantes. Finalmente, los grupos de las pruebas están divididos

según su interacción. Todas las pruebas realizadas tuvieron el mismo periodo de duración, que debía ser mayor de media hora, así que las hicimos de 40 minutos cada una, indistintamente del grupo al cual pertenecieran.

Tanto del grupo de pruebas 1 como del 2, se obtiene de cada captura una carpeta con tres capturas "limpias" correspondientes a upload, download y total. Otros tres ficheros de texto que guardan los datos con los que se trabajan en Matlab, quince imágenes se dividen en las tres clasificaciones de antes upload, download y total, cinco para cada una. En las que se encuentran los histogramas de los paquetes, del tiempo y la diferencia de tiempo entre llegadas. Las dos restantes son las funciones de probabilidad del número de paquetes y del tiempo. Y por último un Excel con los valores que se desea destacar para hacer más cómoda la comparación. Todo esto para solamente una prueba, haciendo un recuento de todos los archivos con los que se han hecho las comparativas, se cuentan 6 pruebas del grupo 1 y 8 pruebas del grupo 2.

5.2.1 Las pruebas del Grupo 1

Las primeras pruebas efectuadas, corresponden al Grupo 1. El objetivo de este grupo de ensayos es el de desarrollar un modelo de tráfico más detallado a ser posible, por lo que las pruebas realizadas en este grupo son de una duración larga y de navegación más intensa y enfocada a la extracción de los datos elaborados y de los cuales se obtenga información útil. Consistió en una navegación basada en Ajax, durante la cual el usuario ha podido interactuar con la pagina y chatear con otros usuarios en línea durante un total de cuarenta minutos. El objetivo principal de estas pruebas fue determinar algunos parámetros que pueden describir aproximadamente una sesión de navegación Ajax en términos de cantidad de tráfico bajo unas circunstancias normales.

De estas pruebas como pueden ser las que más discrepancias tengan entre sí por la propia implicación del usuario, se han realizado seis de las que se sacarán conclusiones más adelante entre sí.

5.2.2 Las pruebas del grupo 2

Estas pruebas se llevaron a cabo de una manera diferente a las pruebas del Grupo 1, dentro de este grupo hay una distinción de navegación. Los cuatro ensayos de cada categoría, que se incluyeron en el análisis se basaban alternativamente en la habilitación y la deshabilitación de JavaScript y sin interactuar nada, para poder comparar las diferencias entre los dos modos de funcionamiento. La navegación consiste en no interactuar con el sitio Web. La elección de efectuar una prueba de este tipo, se ha hecho para analizar los tiempos de la recarga automática de la página si se utiliza la tecnología Ajax.

Observar la cantidad de paquetes que se envían y se reciben sin hacer ninguna petición, simplemente conectarse a la página sobre la que se desea capturar el tráfico sin tan siquiera mover el ratón. En este bloque se encuentran dos distintos tipos de capturas, cada una sobre un tipo de navegación diferente.

1. Facebook con JavaScript activado (Web 2.0)
2. Facebook con JavaScript desactivado (Web 1.0)

De estas pruebas se han hecho cuatro de cada tipo, realmente la diferencia no debe de ser muy grande entre las de la misma categoría, puesto que no interviene el grado de participación del usuario. Pero puesto que cambia el tipo de navegación varían los datos. No siempre se recibirá ni se enviará la misma información dependiendo de en qué Web se esté.

5.3 Comparativa estadística de los resultados

El uso de Wireshark, junto con algunas secuencias de comandos de Matlab para la extracción correspondiente de los datos, explicado con detalle en el capítulo anterior, permitió obtener información sobre las sesiones de navegación, en particular, para cada una de las pruebas está presente una tabla resumen en la figura 5.1 que muestra la media total de las medias de las pruebas. El resto de tablas que no se ven en esta imagen se encuentran en los anexos.

Entre las pruebas del Grupo 2 se puede analizar la diferencia real de funcionamiento de Ajax, ya que no existe interacción con el usuario, los datos no están influenciados por el comportamiento de los usuarios. La media de los datos obtenidos son los siguientes,

Facebook con Ajax	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	0,809	1,08	1,889
media tamaño paq (bytes)	200,276	961,027	635,16
media bytes/sec	162,049	1037,769	1199,778
media Mbit/sec	0,001	0,008	0,01
media de un solo paq	200,2765	961,0272	635,1602
desv. tipica de un solo paq	291,471	587,924	612,9326
media tiempo	1330,4	1427	1385,7
desv. Tipica tiempo	1068,9	1056,4	1062,7
media dif. Tiempo	1,2359	0,9261	0,5294
desv. Tipica dif. Tiempo	6,2932	5,4869	4,1442
histograma del nº de paq	1928 (fig1)	2573 (fig6)	4501 (fig11)
histograma del tiempo	fig2	fig7	fig12
histograma de la dif. De T	fig3	fig8	fig13
Grafica Paq	fig4	fig9	fig14
Grafica Tiempo	fig5	fig10	fig15

Facebook sin Ajax	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	0,495	0,69	1,0257
media tamaño paq (bytes)	183,373	785,96	526,588
media bytes/sec	95,317	585,41	758,66
media Mbit/sec	0,001	0,005	0,005
media de un solo paq	183,3733	785,9596	526,5881
desv. tipica de un solo paq	265,5031	670,035	612,3693
media tiempo	512,5292	498,66	506,66
desv. Tipica tiempo	705,4595	698,56	701,36
media dif. Tiempo	2,3422	1,5624	0,945
desv. Tipica dif. Tiempo	6,8456	5,821	4,456
histograma del nº de paq	860 (fig1)	1138 (fig6)	1998(fig11)
histograma del tiempo	fig2	fig7	fig12
histograma de la dif. De T	fig3	fig8	fig13
Grafica Paq	fig4	fig9	fig14

Grafica Tiempo	0,495	0,69	1,0257

Fig. 5.1: Media de los parámetros de las pruebas del grupo 2

En primer lugar podemos observar que el número de paquetes transmitidos, durante la prueba con Ajax,son mas del doble que sin Ajax. En la figura siguiente se muestra los dos histogramas de paquetes con Ajax y sin Ajax respectivamente:

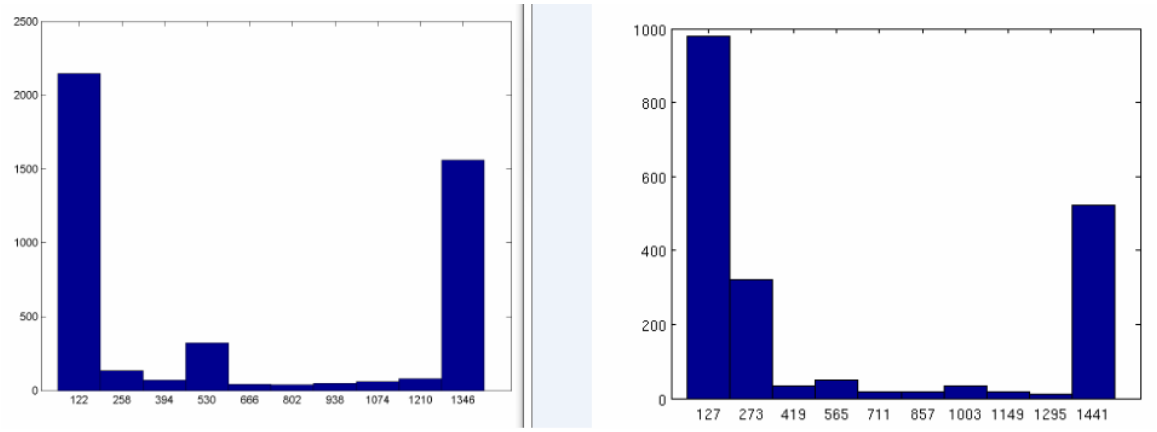


Fig. 5.2: Histogramas del Nº de paquete con Ajax y sin Ajax respectivamente

También podemos observar que en la prueba de Facebook con Ajax se transmiten 1200 bytes por segundo y de media 1,9 paquetes por segundo mientras en la prueba de Facebook sin Ajax se han transmitido unos 700 bytes por segundo y de media un paquete por segundo. Esta diferencia de casi el doble en los dos parámetros nombrados se atribuye a la activación o no de Ajax, por lo que podemos afirmar que con la navegación 2.0 se transfieren más paquetes que con una navegación 1.0. El tamaño del paquete medio de las navegaciones no presenta una gran diferencia , aunque podemos observar una mínima superioridad del tamaño de los paquetes de la navegación 2.0.

Si nos fijamos en la diferencia de tiempo de las dos navegaciones podemos observar como con la utilización de Ajax es de 1 segundo y que con ajax es de medio segundo, con lo que podemos afirmar que la diferencia de nº de conexiones es casi el doble entre la dos navegaciones. También se observa que el tiempo medio de las conexiones aumenta con el uso de Ajax, es mas del doble.

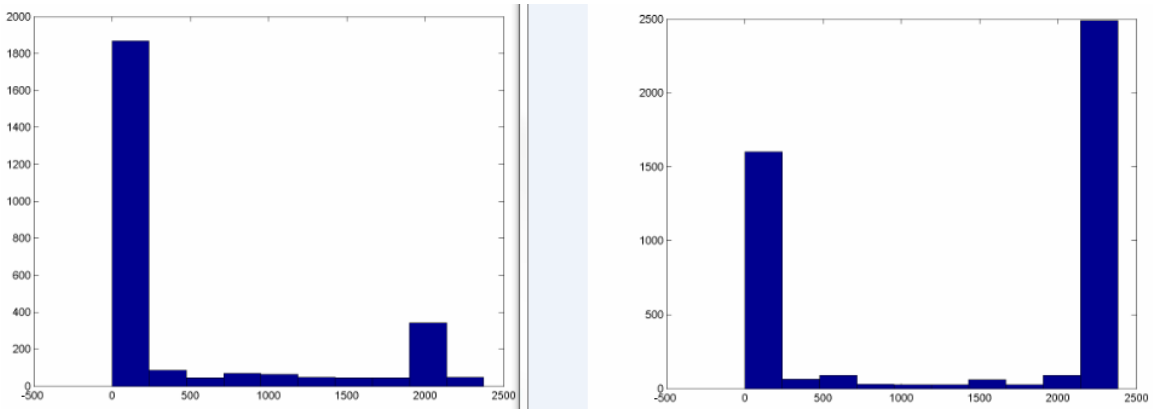


Fig. 5.3: Histograma de Tiempo sin Ajax y con Ajax respectivamente

Fijándonos en las gráficas de las modelizaciones de los paquetes para las dos navegaciones podemos observar que son muy similares, aunque sin Ajax la media de tamaño de los paquetes es un poco superior a la de Ajax.,, presentando desviaciones típicas muy similares para ambos casos.

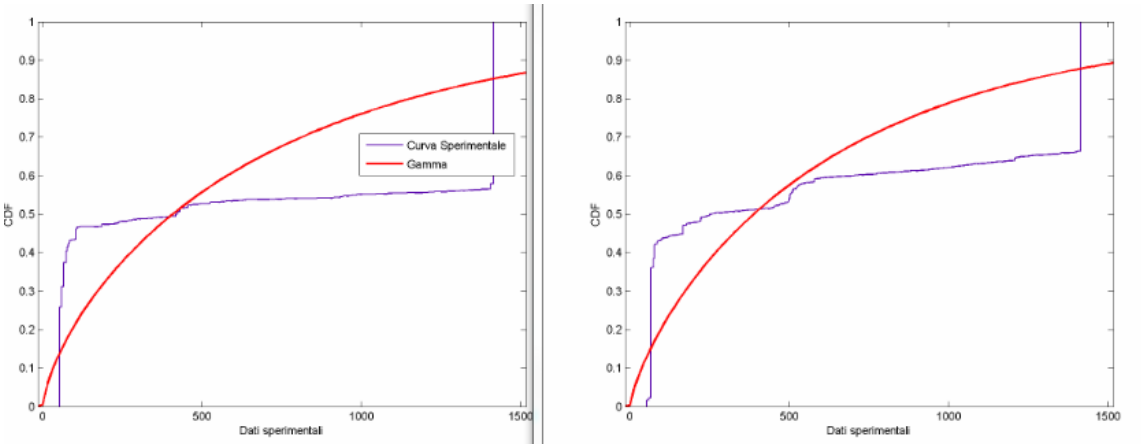


Fig. 5.4 : Modelización Paquetes con Ajax y sin Ajax respectivamente

La modelización del tiempo de las conexiones depende bastante del uso que le de el usuario a la página, se puede decir, que las conexiones con ajax presentan una media de 1385 segundos y una desviación típica de 1062 y , las conexiones sin Ajax presentan una media de 506 segundos con una desviación típica de 701 segundos. Por esta razón se puede afirmar que para la modelización del tiempo de conexión las dos navegaciones siguen un modelo parecido, pero siempre con mas actividad con el usu de Ajax, como podemos observar en las siguientes graficas:

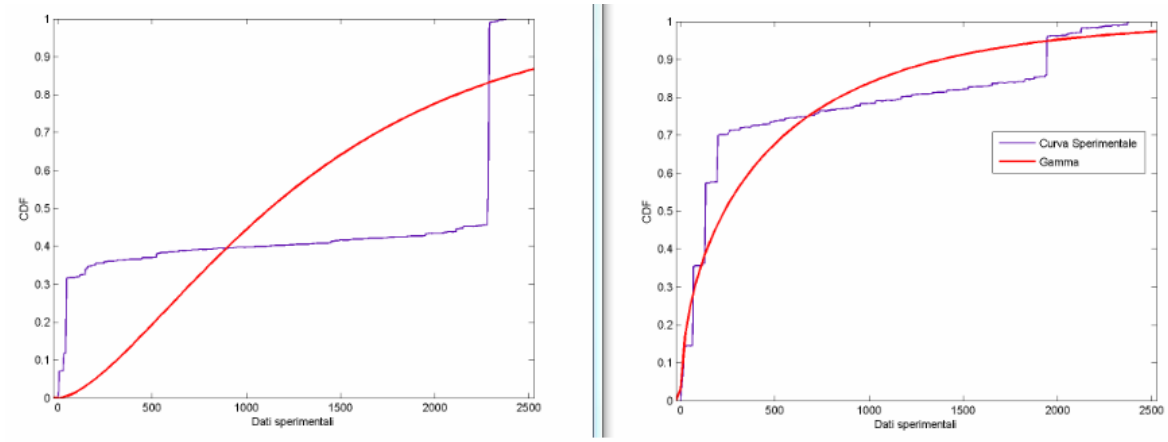


Fig. 5.5 : Modelización del Tiempo con Ajax y sin Ajax respectivamente

Con las 6 capturas del grupo 1, se puede analizar el funcionamiento de Ajax en un entorno real, los datos que se han obtenido vienen influenciados por la interacción del usuario con la página, en este caso Facebook, para así poder analizar de forma práctica la influencia del usuario en la navegación 2.0.

Facebook interac.	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	1,32116667	2,4585	3,05366667
media tamaño paq (bytes)	317,249333	1111,14667	726,416167
media bytes/sec	416,890333	3042,87133	2244,5445
media Mbit/sec	0,00333333	0,0245	0,01783333
media de un solo paq	317,249333	1037,69233	726,416167
desv. tipica de un solo paq	496,60685	595,848833	662,901967
media tiempo	966,4296	948,019567	954,823683
desv. Tipica tiempo	826,44075	850,3201	840,128967
media dif. Tiempo	0,68276667	0,52805	0,30751667
desv. Tipica dif. Tiempo	2,65776667	2,30675	1,8045
histograma del nº de paq	3108,33333	3546,33333	7194,33333

Fig. 5.6 : Media de los parámetros de las pruebas del grupo 1.

Podemos observar de los datos obtenidos que debido a la interacción del usuario, el número de paquetes capturados durante la prueba es superior que a la utilización de Ajax sin interacción. Con la interacción del usuario se han transmitido 3,1 paquetes por segundo, con una media de 716 bytes cada paquete. Comparando estos datos con los reflejados en la fig 5.1, podemos ver que se ha producido un aumento de 1,2 paquetes por segundo y un aumento de la media de paquete de unos 100 bytes con respecto a la navegación 2.0 sin interacción. Con la interacción del usuario se han transmitido 5,45 Mbytes y sin ella se han transmitido 2,85 Mbytes, siendo el nuemro medio de paquetes transmitidos durante los 40 minutos es de 7612, siendo casi el doble que los paquetes transmitidos durante la navegación 2.0 sin interacción. Todos estos resultados son Totales, sin distinguir entre Upload y Download, pero podemos ver que generalmente download presenta una media de tamaño de paquete superior a Upload, aunque la media de paquetes por segundo es muy similar entre ellas.

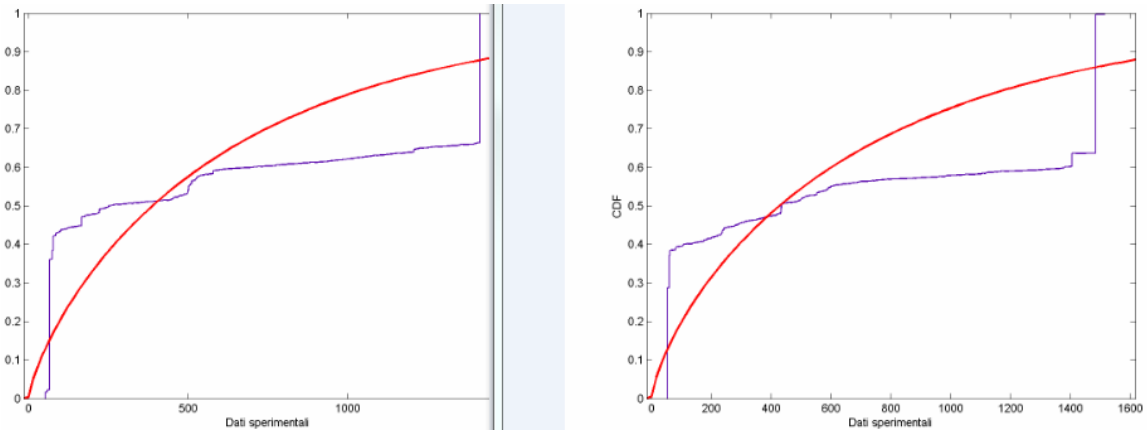


Fig. 5.7: Modelizaciones Sin y Con interacción de los paquetes.

El número de conexiones son casi el doble con la interacción del usuario y presenta una media de tiempo de la conexiones de un 25% menor que en la navegación 2.0 sin interacción.

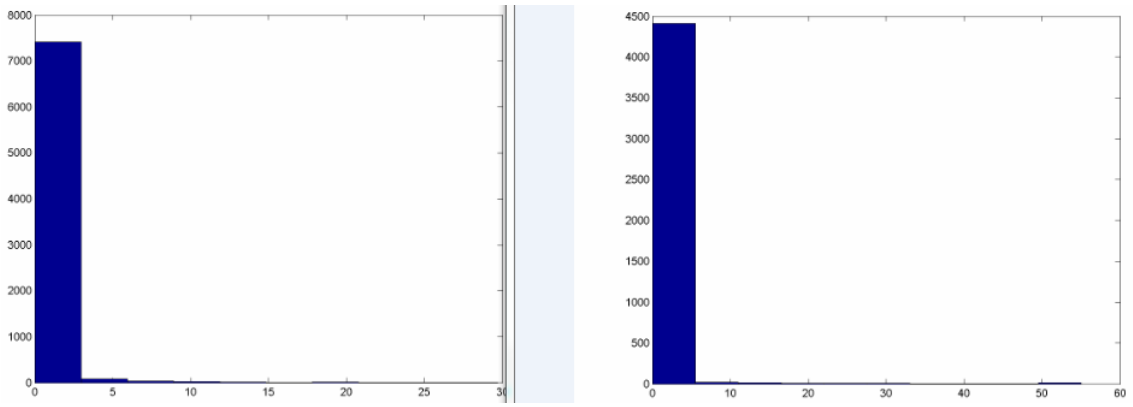


Fig. 5.8: Histogramas Dif. Tiempo Interaccionando y sin Inter. respectivamente.

5.4 Validación de la cadena de Markov

Se parte del texto obtenido con Whireshark para obtener los datos que hay que introducir al programa Statistics Toolbox de Matlab, el cual contiene una única columna con la diferencia de tiempo entre llegadas en segundos. De este momento, en adelante, la "vida" de la secuencia seguirá un procedimiento idéntico que nos dará lugar a los datos que en este apartado se ilustrarán.

El objetivo principal es crear una secuencia lo más próxima a la del ingreso, es decir, ya se han presentado los elementos que se necesitan para que Matlab trabaje con los parámetros necesarios y forme una secuencia reconstruida que represente a la original. De la secuencia efectuada ha sido revelado un número de valores con una dinámica muy variable para el tiempo entre llegadas. El número de datos es de 3500, puesto que se han agrupado las pruebas realizadas de Facebook sin Ajax, con valores muy variables entre si.

Cuando se habla de un modelo HMM se especifica, normalmente, también el número de estados N y el número de símbolos con los cuales debe ser realizado. Para reducir la complejidad y el tiempo de elaboración, la secuencia se ha discretizado de modo que los valores sean enteros y dentro del intervalo de N estados.

Para generar los parámetros de los modelos ocultos de Markov, Matlab utiliza en sus funcionalidades destinadas a este fin el algoritmo de Baum- Welch. Antes de proceder a las ilustraciones de los resultados, parece oportuno mostrar las características de las elaboraciones utilizadas para nuestro análisis a fin de justificar las limitaciones de los resultados.

El programa principal, que realiza los modelos HMM a priori, genera, como ya se ha dicho, las matrices A y B. Esto ha necesitado un ingreso, además de la secuencia original discretizada, el número de estados N y el número de símbolos M con los cuales se quiere crear el modelo. Para calcular estos parámetros, se considera que el tamaño sea apropiado, los modelos obtenidos han sido calculados con un número de estados N = 5, 10 ,15 y 20, por la recomendación de estudios de este departamento anteriores que aseguran que este es el intervalo idóneo, y lo que se pretende en este estudio es acotar

aún más la búsqueda. El valor M se fija a 10 y va variando N. Al aumentar N, de hecho, se debe ajustar el número de símbolos de la secuencia, es decir, discretizar nuevamente. Si no se hace así, el programa responderá con un error. Se muestran nuestros componentes:

- La secuencia discretizada de 3600 valores de N estados posibles:

$$\text{Seq}=\begin{bmatrix} x1 \\ x2 \\ x3 \\ \ldots \end{bmatrix}$$ *siendo x1,x2,x3... valores comprendidos entre 1 y N*

- La matriz transmisión para N=10:

$$\text{TRGUESS}_{NxN} = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$$

- La matriz emisión para N=10 y M=10:

$$\text{EMITGUESS}_{NxM} = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$$

Tras la ejecución de la función hmmtrain (seq,TRGUESS,EMITGUESS) da como resultado estas dos matrices: ESTTR y ESTEMIT que serán nuestras matrices A y B.

- La matriz de probabilidad de transmisión para N=10:

$$
\mathrm{estTR}_{\mathrm{NxN}} = \begin{bmatrix}
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1
\end{bmatrix}
$$

- La matriz de probabilidad de emisión para N=10 y M=10:

$$
\mathrm{estE}_{\mathrm{NxM}} = \begin{bmatrix}
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0 \\
0.944 & 0.0099 & 0.0072 & 0.0206 & 0.0182 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

- Se añade el parámetro que exige la siguiente función que es: len. Que se trata del tamaño del vector que dará como resultado; en este caso, el mismo número que obtuvimos de la secuencia original, para facilitar la comparación entre ambas.

Se obtiene finalmente, la primera secuencia, la cual se llamará secuenciaMarkov10 y los estados de la función hmmgenerate (len, A, B).

Hemos hecho el mismo proceso explicado anteriormente con N=5, M =10 y 5 estados, y se obtendrá la SecuenciaMarkov5. Con N=15 y M=10 y 15 estados, de la que se obtiene SecuenciaMarkov15. Y por último con N=20, M=10 y 20 estados, de la que se obtiene SecuenciaMarkov20.

Como resulta engorroso y prácticamente imposible mostrar las cuatro secuencias completas, se ilustran las comparaciones mediante ilustraciones, y se harán dichas comparaciones analizando con detalle los histogramas de las secuencias, aunque se ha de advertir ya, que la diferencia entre ellas es mínima. Se muestran el histograma de la secuencia original y las secuencias resultantes de la serie de instrucciones Matlab.

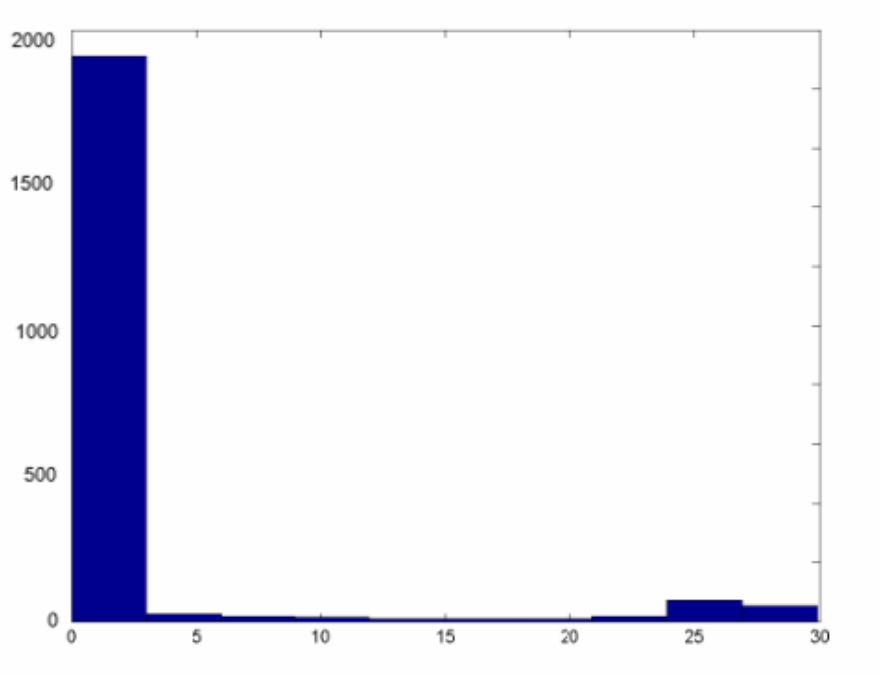


Fig. 5.9: Histograma secuencia original total.

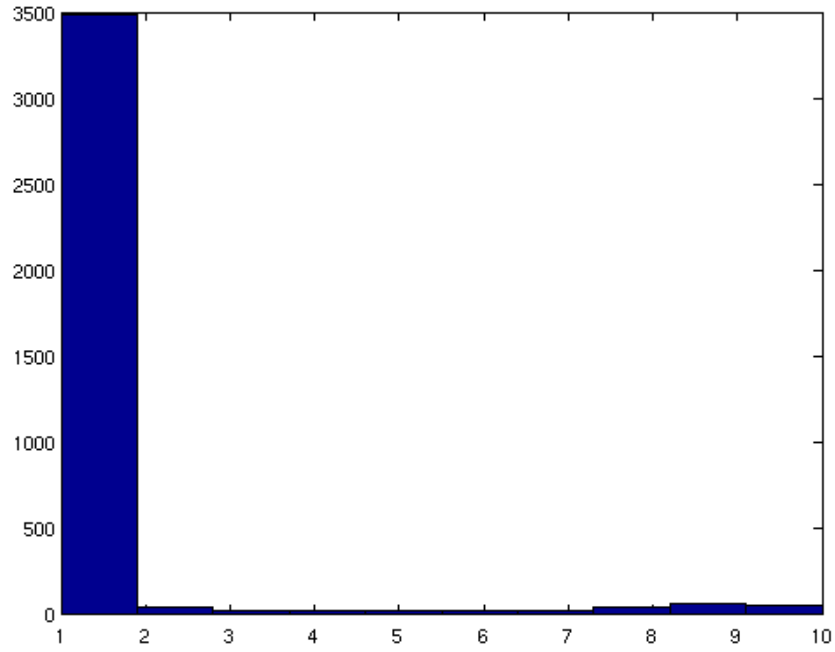


Fig. 5.10: Histograma secuencia reconstruida con Harkov N=10

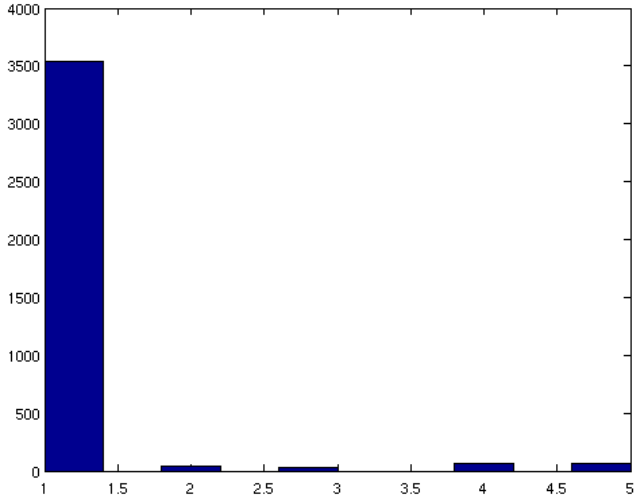


Fig. 5.11: Histograma secuencia Harkov N=5.

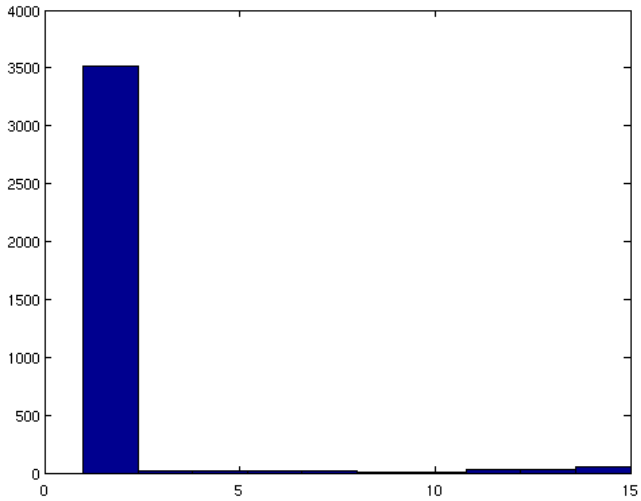


Fig. 5.12: Histograma secuencia Harkov N=15.

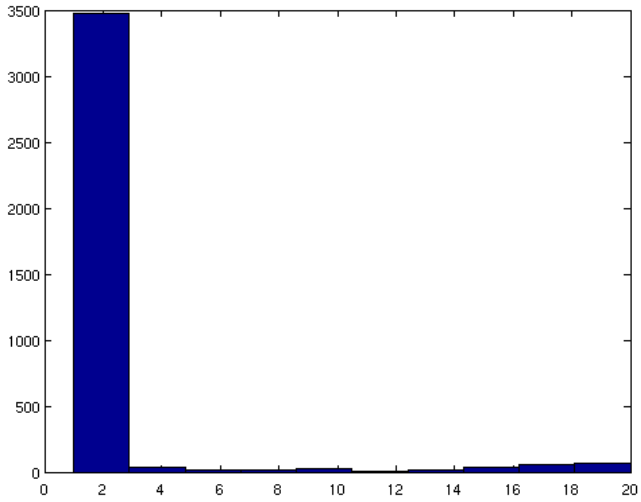


Fig. 5.13: Histograma secuencia Harkov N=20.

Se observa que con una muestra pequeña se obtiene prácticamente el mismo histograma. Puesto que son cadenas de Markov, que se caracterizan por ser “sin memoria”, es decir, que de los estados presentes se podrían deducir los futuros. Se ha seleccionado N=10, por ser el que menos discrepancias tiene aparentemente.

6 Capítulo 6: Conclusiones y trabajos futuros

La continua evolución de Internet ha dado lugar a la necesidad de introducir nuevas tecnologías para mejorar la navegación en la Web.

La concretización de estas tecnologías está representada por el Ajax. Cualquier persona puede beneficiarse de las mejoras introducidas por el presente nuevo enfoque a la navegación, sin embargo, hasta la fecha, los modelos para caracterizar las estadísticas y probabilística del tráfico son escasos e incompletos.

En este proyecto fin de carrera el objetivo principal ha sido caracterizar el tráfico de la Web basada en Ajax, llenando, en algunos puntos, la carencia de los estudios anteriores, y completando los modelos avanzando en nuevas hipótesis. Después de las pruebas de laboratorio y de los análisis llevados a cabo es posible afirmar que se han logrado la mayor parte de los objetivos.

Aunque el fin es trabajar en las conclusiones entre las navegaciones, de alguna manera durante el proceso, se ha creado un procedimiento para el desarrollo, elaboración, extracción, comparación, análisis y creador de modelos que es útil para cualquier proyecto futuro, que de una forma subliminal se ha convertido en uno de los principales logros de este PFC, reunir todos estos medios y conseguir una metodología de trabajo eficaz para posibles estudios posteriores de la misma naturaleza.

- Adecuar un escenario de navegación.

- Procedimiento a seguir para la elaboración de las pruebas

- Procedimiento para una buen análisis y comparativas con gráficas y estadísticos representativas.

- Procedimiento para desarrollar una sistemática con la intención de concretar parámetros necesarios para crear una cadena de Markov

Las pruebas experimentales han permitido extraer información útil y sacar conclusiones acerca de las características tangibles reales de Ajax, no solo desde la perspectiva de la experiencia de navegación del usuario, sino también en términos de estadística y probabilidad. En particular, los resultados obtenidos con las pruebas de laboratorio descritas en los capítulos anteriores, confirman que las sesiones de navegación basadas en Ajax son diferentes de las sesiones de navegación basadas en la navegación clásica. Además, el funcionamiento de Ajax da una nueva dinámica a la Web, que se refleja sobre todo en diferentes modelos estadísticos respecto a los ya conocidos para el funcionamiento normal de la Web. Se podría concluir, por tanto, que la Web 2.0 ha introducido no sólo un nuevo enfoque a la navegación de Internet, sino también un nuevo funcionamiento de las redes y una nueva caracterización del tráfico presente en este estudio.

No obstante, el presente estudio nos ha permitido crear un modelo estadístico probabilístico que caracteriza correctamente el tráfico de un solo usuario basado en

Ajax, en el futuro, se podría proceder por medio de pruebas con más usuarios y con múltiples sitios Web. Incluso trabajar, con la Web 3.0 ya existente, que aprovecha las ventajas de las dos versiones de las Web anteriores. Así será posible verificar los modelos encontrados que también se aplican al tráfico generado por la navegación de diversos sitios utilizados en las presentes pruebas. Además, con una navegación de múltiples usuarios también se pueden modelar los parámetros sobre los cuales sólo fue posible avanzar hipótesis debido a la falta de datos. En cualquier caso, se puede decir que el presente estudio elaborado ha abierto la puerta a un área de investigación relativa a Ajax que hasta la fecha estaba descuidada o por lo menos un tanto limitada. La contribución en estas páginas es pequeña, pero es el análisis primordial para la investigación en este campo y poder cumplir objetivos más complejos en un futuro próximo.

Bibliografía

•http://www.wireshark.org Programa analizador de protocolos de comunicaciones.

•http:/ www.maths.lth.se/matstat/staff/krys/Program/stats_tb.pdf Librería Statistics toolbox de Matlab.

•http://es.wikipedia.org/wiki/Wikipedia Consultas de todo tipo.

[1] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol HTTP/1.0. RFC 1945 (Informational), May 1996.

[2] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2068 (Proposed Standard), Jan. 1997. Obsoleted by RFC 2616.

[3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Updated by RFCs 2817, 5785.

[4] N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045 (Draft Standard), Nov. 1996. Updated by RFCs 2184, 2231, 5335.

[5] J. Postel. Internet Protocol. RFC 791 (Standard), Sept. 1981. Updated by RFC 1349.

[6] E. Casilari, F. J. Gonzblez, and F. Sandoval. Modeling of HTTP traffic. *IEEE Communications Letters*, 5:272–274, 2001.

[7] B. A. Mah. An empirical model of HTTP network traffic. In *INFOCOM '97.Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, pages 592–600, Kobe , Japan, 1997.

[8] H.-K. Choi and J. O. Limb. A behavioral model of web traffic. In ICNP '99: Proceedings of the Seventh Annual International Conference on Network Protocols, page 327, Washington, DC, USA, 1999. IEEE Computer Society.

[9] J. J. Lee and M. Gupta. A new traffic model for current user web browsing behavior. Technical report, Communications Technology Lab, Intel Corp., 2007.

[10] T. O'Reilly. What is web 2.0. design patterns and business models for the next generation of software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/whatis-web-20.html, September 2005. Stand 12.5.2009.

[11] J. J. Garrett. Ajax: A new approach to web applications. http://adaptivepath. com/ideas/essays/archives/000385.php, February 2005. [Online; Stand 18.03.2008].

[12] Samuel Carvajal. Desarrollo de aplicaciones Wev enriquecidas seguras bajo en el enfoque AJAX. Universidad central de Venezuela.Marzo 2008.

[13] Y. Jie, L. Z.W., and F. Liu. The impact of ajax on network performance. The Journal of China Universities of Posts and Telecommunications, 14(Supplement 1):32 – 34, 2007.

[14] F. Schneider, S. Agarwal, T. Alpcan, and A. Feldmann. The new web: Characterizing ajax traffic. In M. Claypool and S. Uhlig, editors, *Passive and Active Network Measurement*, volume 4979 of *Lecture Notes in Computer Science*, pages 31–40. Springer Berlin-Heidelberg, 2008.

[15] E. Bozdag, A. Mesbah, and A. van Deursen. A comparison of push and pull techniques for AJAX. In *Web Site Evolution, 2007. WSE 2007. 9th IEEE International Workshop on*, pages 15–22, Paris, 2007.

[16] S. Lin, Z. Gao, and K. Xu. Web 2.0 traffic measurement: analysis on online map applications. In *NOSSDAV '09: Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video*, pages 7–12, New York, NY, USA, 2009. ACM.

Anexo – Tablas de los parámetros

El presente documento tiene como objetivo servir de apoyo y proporcionar a los lectores las tablas reales de las pruebas, matrices y demás resultados que no se han presentado en la memoria para optimizar el proyecto.

Tablas de los parámetros pruebas Facebook interactuando:

Facebook inter1	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	0,955	1,214	2,169
media tamaño paq (bytes)	317,32	940,325	685,653
media bytes/sec	311,044	1135,789	1468,64
media Mbit/sec	0,002	0,01	0,012
media de un solo paq	317,3198	940,3249	685,649
desv. tipica de un solo paq	498,3219	616,7742	636,1446
media tiempo	985,314	1036,3	1008,8
desv. Tipica tiempo	799,982	880,0013	872,5612
media dif. Tiempo	1,0411	0,8237	0,4897
desv. Tipica dif. Tiempo	3,523	3,45	2,1624
histograma del nº de paq	2294 (fig1)	2797 (fig6)	5091 (fig11)

Facebook inter2	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	1,399	1,813	3,212
media tamaño paq (bytes)	349,24	1012,546	725,123
media bytes/sec	511,141	1769,509	2280,643
media Mbit/sec	0,005	0,013	0,018
media de un solo paq	349,2389	1012,54	725,123
desv. tipica de un solo paq	532,4181	598,9576	682,1232
media tiempo	982,568	917,1824	950,8586
desv. Tipica tiempo	785,9495	816,3455	802,5798
media dif. Tiempo	0,719	0,5792	0,3058
desv. Tipica dif. Tiempo	2,8756	2,1258	1,9038
histograma del nº de paq	3246 (fig1)	4369 (fig6)	7615 (fig11)

Facebook inter3	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	1,125	1,652	2,777
media tamaño paq (bytes)	301,379	1067,012	711,9328
media bytes/sec	3410,481	1873,296	2121,465
media Mbit/sec	0,003	0,014	0,017
media de un solo paq	301,38	1067,012	711,93
desv. tipica de un solo paq	469,3781	592,6731	652,5239
media tiempo	1041,2	1009,2	1051,6
desv. Tipica tiempo	803,4528	789,0185	872,3261
media dif. Tiempo	0,1701	0,0884	0,1203
desv. Tipica dif. Tiempo	0,7215	0,6012	0,5925
histograma del nº de paq	2856 (fig1)	4125 (fig6)	6981 (fig11)

Facebook inter4	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	1,542	2,114	3,656
media tamaño paq (bytes)	248,249	1164,317	789,93
media bytes/sec	384,135	2456,214	3014,467
media Mbit/sec	0,004	0,017	0,021
media de un solo paq	248,3487	1164,32	789,93
desv. tipica de un solo paq	429,8712	515,3581	638,3651
media tiempo	692,9218	656,4438	702,318
desv. Tipica tiempo	814,3496	809,6158	821,1524
media dif. Tiempo	0,5911	0,4256	0,2341
desv. Tipica dif. Tiempo	2,5437	2,2465	1,6234
histograma del nº de paq	3652 (fig1)	5128 (fig6)	9780 (fig11)

Facebook inter5	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	1,187	1,852	3,039
media tamaño paq (bytes)	364,608	896,019	622,35
media bytes/sec	332,264	1154,654	1667,826
media Mbit/sec	0,004	0,01	0,014
media de un solo paq	364,6078	896,0187	622,3492
desv. tipica de un solo paq	514,7487	631,2562	637,02564
media tiempo	1028,2	1004,3	1055,6
desv. Tipica tiempo	824,7569	835,9265	848,1684
media dif. Tiempo	0,9245	0,80125	0,4324
desv. Tipica dif. Tiempo	3,1464	2,8442	2,3451
histograma del nº de paq	2384 (fig1)	3124 (fig6)	5506 (fig11)

Facebook inter6	UPLOAD	DOWNLOAD	AGGREGATO
media paq/sec	1,479	2,374	3,853
media tamaño paq (bytes)	345,416	1434,39	768,424
media bytes/sec	576,889	1894,604	2634,306
media Mbit/sec	0,005	0,025	0,0255
media de un solo paq	345,4159	1434,3987	768,4238
desv. tipica de un solo paq	526,2714	630,2554	641,2871
media tiempo	928,3024	945,4	966,407
desv. Tipica tiempo	832,4587	890,2587	877,4032
media dif. Tiempo	0,5345	0,4978	0,2689
desv. Tipica dif. Tiempo	2,6524	2,3587	1,634
histograma del nº de paq	3356 (fig1)	3456 (fig6)	6812 (fig11)

Matrices resultantes de la función hmmtrain para N=5

estTR =

0.2000	0.2000	0.2000	0.2000	0.2000
0	1.0000	0	0	0
0	0	1.0000	0	0
0	0	0	1.0000	0
0	0	0	0	1.0000

estE =

0.9440	0.0099	0.0072	0.0206	0.0182	0	0	0	0	0
0.9440	0.0099	0.0072	0.0206	0.0182	0	0	0	0	0
0.9440	0.0099	0.0072	0.0206	0.0182	0	0	0	0	0
0.9440	0.0099	0.0072	0.0206	0.0182	0	0	0	0	0
0.9440	0.0099	0.0072	0.0206	0.0182	0	0	0	0	0

Con N=15

estTR =

Columns 1 through 12

0.0667	0.0667	0.0667	0.0667	0.0667	0.0667	0.0667	0.0667	0.0667	0.0667	0.0667	0.0667
0	1.0000	0	0	0	0	0	0	0	0	0	0
0	0	1.0000	0	0	0	0	0	0	0	0	0
0	0	0	1.0000	0	0	0	0	0	0	0	0
0	0	0	0	1.0000	0	0	0	0	0	0	0
0	0	0	0	0	1.0000	0	0	0	0	0	0
0	0	0	0	0	0	1.0000	0	0	0	0	0
0	0	0	0	0	0	0	1.0000	0	0	0	0
0	0	0	0	0	0	0	0	1.0000	0	0	0
0	0	0	0	0	0	0	0	0	1.0000	0	0
0	0	0	0	0	0	0	0	0	0	1.0000	0
0	0	0	0	0	0	0	0	0	0	0	1.0000
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 15

0.0667	0.0667	0.0667
0	0	0

```
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
    1.0000         0         0
         0    1.0000         0
         0         0    1.0000


estE =


  Columns 1 through 12


    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059
    0.9322    0.0078    0.0048    0.0035    0.0021    0.0029    0.0024    0.0021    0.0019    0.0021    0.0035    0.0059


  Columns 13 through 15


    0.0115    0.0107    0.0064
    0.0115    0.0107    0.0064
    0.0115    0.0107    0.0064
    0.0115    0.0107    0.0064
    0.0115    0.0107    0.0064
    0.0115    0.0107    0.0064
    0.0115    0.0107    0.0064
```

0.0115 0.0107 0.0064

0.0115 0.0107 0.0064

0.0115 0.0107 0.0064

0.0115 0.0107 0.0064

0.0115 0.0107 0.0064

0.0115 0.0107 0.0064

0.0115 0.0107 0.0064

0.0115 0.0107 0.0064

Con N=20

estTR =

Columns 1 through 12

0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500

0 1.0000 0 0 0 0 0 0 0 0 0 0

0 0 1.0000 0 0 0 0 0 0 0 0 0

0 0 0 1.0000 0 0 0 0 0 0 0 0

0 0 0 0 1.0000 0 0 0 0 0 0 0

0 0 0 0 0 1.0000 0 0 0 0 0 0

0 0 0 0 0 0 1.0000 0 0 0 0 0

0 0 0 0 0 0 0 1.0000 0 0 0 0

0 0 0 0 0 0 0 0 1.0000 0 0 0

0 0 0 0 0 0 0 0 0 1.0000 0 0

0 0 0 0 0 0 0 0 0 0 1.0000 0

0 0 0 0 0 0 0 0 0 0 0 1.0000

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

Columns 13 through 20

0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

```
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
  1.0000       0       0       0       0       0       0       0
       0  1.0000       0       0       0       0       0       0
       0       0  1.0000       0       0       0       0       0
       0       0       0  1.0000       0       0       0       0
       0       0       0       0  1.0000       0       0       0
       0       0       0       0       0  1.0000       0       0
       0       0       0       0       0       0  1.0000       0
       0       0       0       0       0       0       0  1.0000


estE =

  Columns 1 through 12

    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019
    0.8720    0.0178    0.0084    0.0056    0.0037    0.0056    0.0028    0.0028    0.0047    0.0019    0.0028    0.0019

  Columns 13 through 20
```

0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075
0.0028	0.0037	0.0028	0.0075	0.0196	0.0112	0.0150	0.0075