Proyecto Final de Carrera:

Herramientas y medidas del tráfico de red

Autor: Pablo Hidalgo Perales

Director: Alejandro Martínez Sala

Septiembre 2012

Agradecimientos

Gracias, solo puedo dar desde aquí las gracias a todo aquel que me haya ayudado durante todos estos años de universidad que culminan con la elaboración de este proyecto. A mis compañeros de clase y amigos que me han aguantado durante estos años de universidad, a mis amigos de siempre que se alegran de que ya sea ingeniero, a los amigos que he conocido estos años en Cartagena, a Alejandro, director de este proyecto por sus ánimos y facilidades para que saliera adelante, a mi familia que por fin tienen un hijo ingeniero y en especial a mi abuela que me ha facilitado mucho la vida durante todos estos años.

Capítulo especial merece todo lo relacionado con mi experiencia Erasmus. En primer y más importante lugar, gracias a la profesora Luisa Massari por ayudarme todo el año en la elaboración de este proyecto, por facilitarme las cosas y resolver cualquier problema que haya tenido en el camino, gracias a mis amigos erasmus que creían que nunca llegaría a terminar este proyecto, en especial a Javi Ram e Inma, los grandes amigos que me llevo este año, gracias al resto de mis amigos Erasmus que es imposible destacarlos a todos, gracias a mi profesora de italiano Maria Grazia, y gracias en general a todo el mundo que me he cruzado este año y ha ayudado a que este año sea inolvidable.

Grazie, posso solo ringraziare tutti quelli che mi hanno aiutato in questi anni di Università, che si concludono con questa tesi di laurea: a i miei compagni di corso e miei amici che mi hanno supportato durante questi anni di Università, a i miei amici di sempre che si allegrano del fatto che già sia un ingegnere, a i amici che ho conosciuto questi anni in Cartagena, al direttore di questo progetto per il suo animo e consigli, alla mia famiglia che finalmente ha un figlio ingegnere e, in special modo, a mia nonna che mi ha sostenuto tanto in questi anni di università.

Una parte speciale merita tutto quello che riguarda la mia esperienza Erasmus. In primo luogo, e più importante, va un ringraziamento alla Professoressa Luisa Massari per avermi aiutato tutto l'anno nell'elaborazione di questa tesi, avermi facilitato le cose e per aver risolto qualsiasi problema che ho incontrato nel corso del camino. Anche grazie a i miei amici Erasmus, che credevano che non sarei mai riuscito a finire questo lavoro, in speciale a Javi Ram e Inma, i grandi amici che mi porto da quest'anno, e grazie al resto dei miei amici Erasmus che sarebbe imposibile menzionare tutti. Infine, un ringraziamento alla mia profesoressa di italiano Maria Grazia e in generale a tutti quelli che hanno fatto di quest'anno un'esperienza indimenticabile.

INDICE GENERAL

INDICE DE ILUSTRACIONES

Capítulo 1: Introducción

Este proyecto nace como consecuencia del convenio Erasmus entre la Universidad Politécnica de Cartagena (UPCT) y la Universitá degli Studi di Pavia. Ha sido dirigido en Italia por la profesora Luisa Massari y en España por el profesor Alejandro Martínez Sala.

En este trabajo se aborda el análisis de la red de tráfico de la universidad de Pavia (Campus de ingeniería) y se hace un estudio sobre los distintos protocolos del modelo TCP/IP así como un estudio de las diferentes herramientas para capturar y analizar tráfico que existen en la actualidad, con sus pros y contras y sus principales características.

1.1 Objetivos del proyecto

Ante tal demanda de internet en un gran complejo, como es una universidad, se requiere obtener información sobre cuál es el tráfico generado.

Para ello este proyecto ofrece una visión global del tráfico generado en varias fases del año, y en unos determinados momentos del día.

Antes de empezar a realizar los experimentos, se necesita tener unos conocimientos teóricos previos sobre dicho proyecto. En primer lugar se realizará un estudio teórico sobre los diferentes protocolos que intervienen en TCP/IP, destacando con especial interés los protocolos que nos encontramos en mayor cantidad en las pruebas realizadas.

Tras este estudio sobre los protocolos, se propone hacer un estudio sobre las herramientas open source disponibles en el mercado para capturar y analizar tráfico. De estas herramientas se pide un estudio sobre sus principales características y funciones. Se ofrecen 5 herramientas y tras ver cual ofrece mejores condiciones para realizar este proyecto, se opta por una.

Una vez acabado el estudio teórico, se realizan los experimentos necesarios desde el laboratorio di evalutazione delle prestazione (laboratorio de evaluación de las prestaciones) de la universidad de Pavia. En un primer momento se deben obtener las capturas en unas determinadas condiciones que serán debatidas posteriormente y después se realizaran los análisis necesarios para realizar las estadísticas sobre cantidad de tráfico acumulado, cantidad de bytes transmitidos, cantidad de paquetes… y otras estadísticas que se explicarán en el capítulo de análisis más detalladamente.

1.2 Estructura de la memoria

Esta memoria está enfocada en dos bloques claramente diferenciados, el primero de ellos se centra en el trabajo teórico realizado, y el segundo, en la captura de tráfico y su posterior análisis. Además, hay que añadirle unos apartados finales donde se obtienen las conclusiones del proyecto, un anexo en el que se encuentran todos los datos empleados y una bibliografía donde se indican las fuentes donde se ha obtenido la información aportada.

El primer bloque se subdivide a la vez en dos grandes estudios. El primero de ellos pertenece al capítulo 3 e indica los protocolos existentes en el modelo TCP/IP. Cada protocolo está dividido a la vez en una idea general del protocolo, el formato de sus mensajes y sus características principales. El capítulo siguiente ofrece una variedad de herramientas para captura y análisis de tráfico de red, con un resumen de sus principales características, su funcionamiento y la interfaz aportada por dichas herramientas.

El segundo bloque comprende los capítulos 4 y 5. En el capítulo 4 se muestra el trabajo realizado para poder obtener las pruebas en la captura de tráfico. Se ofrece de forma detallada los métodos seguidos en primer lugar para obtener las capturas indicando paso por paso como se han realizado y posteriormente se ofrece un método para realizar gráficas, tablas y esquemas a partir de los datos obtenidos.

En el capítulo 5 se exponen los resultados siguiendo los pasos de los métodos del anterior capítulo, mostrándose el escenario en el que se ha trabajado y posteriormente los resultados obtenidos con las explicaciones oportunas para cada dato aportado, además de realizar una comparación entre los distintos datos obtenidos en cada semana de capturas.

Capítulo 2: Estudio de los diferentes protocolos que intervienen en TCP/IP.

Antes de comenzar a definir cada protocolo es de utilidad entender con un pequeño resumen como es el modelo TCP/IP donde se encuentran estos protocolos y las diferencias con el modelo OSI.

El modelo TCP/IP fue creado por el departamento de defensa de los Estados Unidos, su nombre viene de dos de los principales protocolos de red, TCP e IP, y es la arquitectura más adoptada para la interconexión de sistemas.

Este modelo estaba basado en 4 capas que incluyen todos los protocolos necesarios para el correcto funcionamiento de una red de comunicaciones.

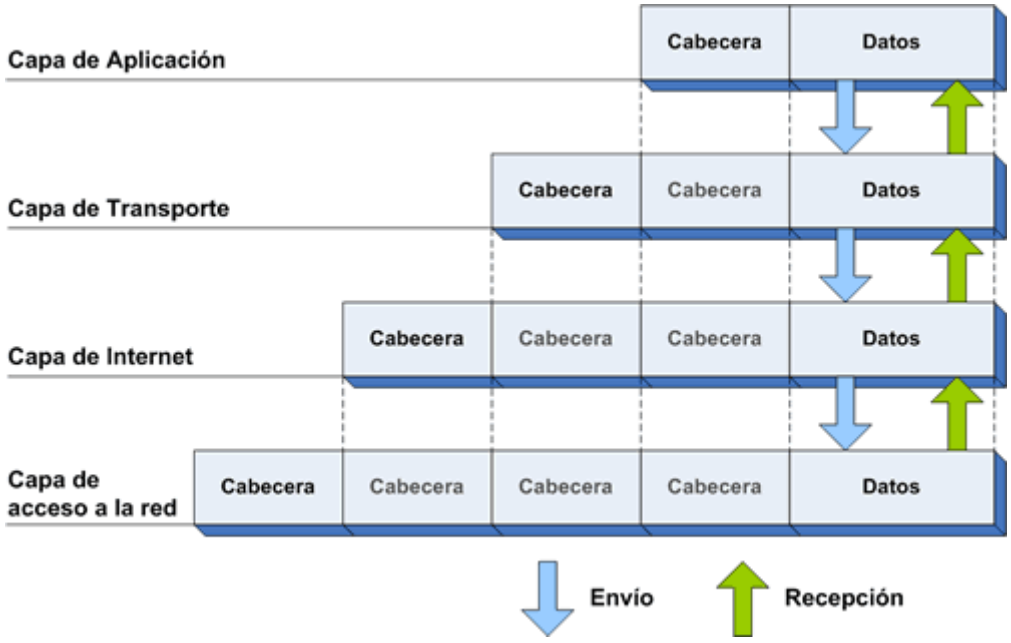


Figura 1: Modelo TCP/IP

En el nivel más bajo se encuentra la capa de acceso a red, donde se especifica información detallada de cómo se envían físicamente los datos a través de la red.

A continuación está la capa de internet (o capa de red), la encargada de empaquetar los datos en datagramas IP y seleccionar la mejor ruta para enviar los paquetes a la red.

La capa de transporte proporciona servicios de transporte desde el host origen hacia el host destino. Los dos protocolos más comunes son TCP y UDP.

Por último, la capa de aplicación maneja protocolos de alto nivel, aspectos de representación, codificación y control de diálogo.

Además de este modelo, existe el modelo OSI, que es el modelo de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones, creado en el año 1984 por la Organización Internacional para la Estandarización (ISO).

OSI es un modelo de 7 capas, similar a TCP/IP, donde las dos primeras capas (física y acceso a red) se englobarían en la capa de acceso a red de TCP/IP y las capas de aplicación, presentación y sesión (OSI) en la capa de aplicación de TCP/IP.

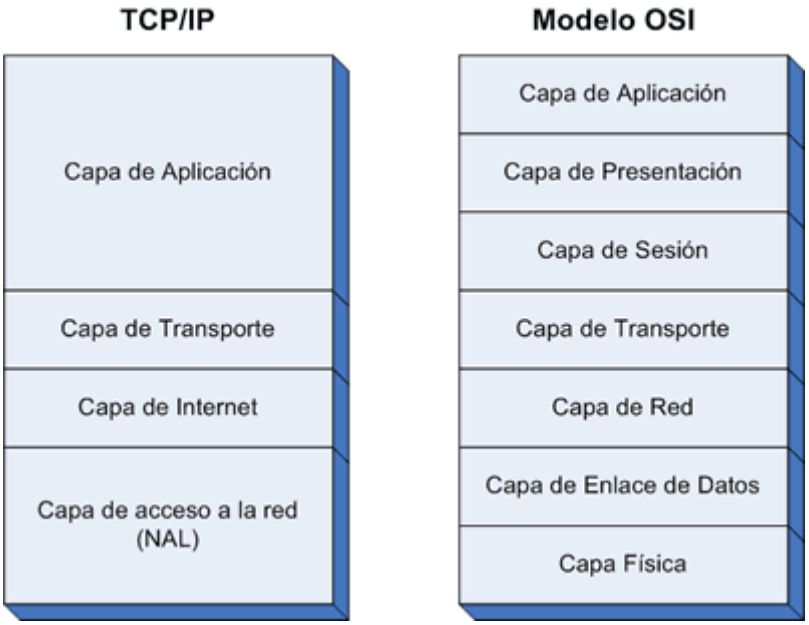


Figura 2: Modelo OSI vs Modelo TCP/IP

A continuación se explican las principales funciones de cada capa del modelo OSI.

La función de la capa física es la de codificar en señales los dígitos binarios que representan las tramas de la capa superior, además de transmitir y recibir estas señales a través de los medios físicos que conectan los dispositivos de la red.

La capa de enlace de datos se ocupa del direccionamiento físico, de la topología de la red, del acceso al medio, de la detección de errores, de la distribución ordenada de tramas y del control del flujo.

La capa de red tiene las mismas funciones que la capa de internet de TCP/IP.

Al igual que la capa anterior, la capa de transporte también tiene las mismas funciones que su capa homóloga en TCP/IP.

La capa de sesión se encarga de mantener y controlar el enlace establecido entre dos computadores que están transmitiendo datos

La capa de presentación estandariza la forma en que se presentan los datos a las aplicaciones.

La capa de aplicación ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos.

Para terminar está introducción se muestran las principales diferencias de ambos modelos.

- La principal diferencia es el número de capas, 7 capas del modelo OSI por las 4 capas del modelo TCP/IP
- OSI distingue claramente los servicios, las interfaces y los protocolos.
- TCP/IP es más simple al disponer de menos capas.
- Los protocolos TCP/IP son los estándares en torno a los cuales se desarrolló Internet, de modo que la credibilidad del modelo TCP/IP se debe en gran parte a sus protocolos. En comparación, las redes típicas no se desarrollan normalmente a partir del protocolo OSI, aunque el modelo OSI se usa como guía.

2.1 TCP

Este protocolo fue creado por Vint Cerf y Robert Kahn durante los años 1973-1974. Es el protocolo que garantiza que los datos lleguen correctamente hasta su destino y en el mismo orden que se transmitieron, por lo que es orientado a conexión. Además, proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

TCP se encuentra en el nivel de transporte del modelo TCP/IP, entre las capas de red y de aplicación, y es un protocolo orientado a conexión y fiable.

2.1.1 Características

- TCP permite colocar los datagramas nuevamente en orden cuando vienen del protocolo IP.
- TCP permite que el monitoreo del flujo de los datos y así evita la saturación de la red.
- TCP permite que los datos se formen en segmentos de longitud variada para "entregarlos" al protocolo IP.
- TCP permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, aplicaciones) en la misma línea pueda circular simultáneamente.
- Por último, TCP permite comenzar y finalizar la comunicación amablemente.

2.1.2 Formato

El formato de un segmento de TCP es el siguiente:

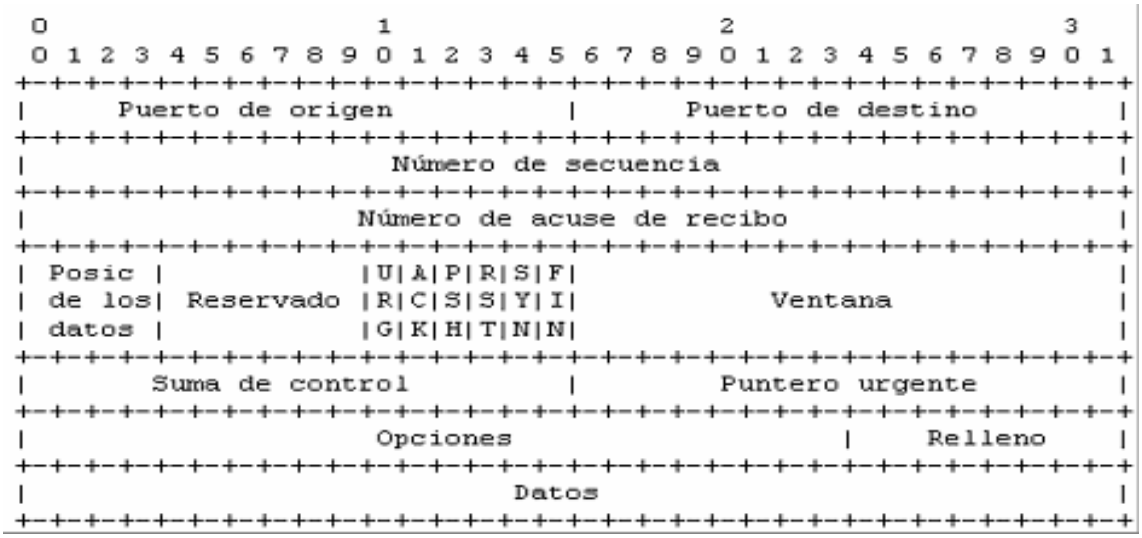


Figura 3: Formato TCP

- Puerto de origen (16 bits): Indica el puerto de la maquina origen.
- Puerto de destino (16 bits): Indica el puerto de la maquina destino
- Numero de secuencia (32 bits): El número de secuencia del primer octeto de datos de este segmento (excepto cuando el indicador SYN esté puesto a uno). Si SYN está puesto a uno es el número de secuencia original y, entonces, el primer octeto de datos es ISN+1 (ISN: Initial Sequence Number).
- Número de acuse de recibo (32 bits): Si el bit de control ACK está puesto a uno, este campo contiene el valor del siguiente número de secuencia que el emisor del segmento espera recibir. Una vez que la conexión queda establecida, este número se envía siempre.
- Posición de los datos (4 bits): El número de palabras de 32 bits que ocupa la cabecera de TCP. Este número indica donde comienzan los datos.
- Reservado (6 bits): Reservado para usos futuros.
- Bits de control (6 bits): Indican información adicional.

    - URG: Si está a uno, el paquete debe procesar de forma urgente.
    - ACK: Si está a uno, el paquete es un acuse de recibo.
    - PSH: Si está a uno, el paquete opera de acuerdo con el método push.
    - RST: Si está a uno, se restablece la conexión.
    - SYN: Indica un pedido para establecer una conexión.
    - FIN: Si está a uno, se interrumpe la conexión.

- Ventana (16 bits): Campo que permite saber la cantidad de bytes que el receptor desea recibir sin acuse de recibo.

- Suma de control (16 bits): La suma de control se realiza tomando la suma del campo de datos del encabezado para poder verificar la integridad del encabezado. Es un protocolo de detección de errores.
- Puntero urgente (16 bits): Indica el número de secuencia después del cual la información es urgente.
- Opciones (tamaño variable): Diversas opciones del protocolo.
- Relleno: Espacio restante después de que las opciones se rellenan con ceros para tener una longitud que sea múltiplo de 32 bits.

2.1.3 Funcionamiento

En TCP se puede establecer en tres etapas la conexión: Establecimiento de la conexión, transferencia de datos y cierre de la conexión.

Establecimiento de la conexión: Este proceso se suele realizar con un mecanismo comúnmente llamado negociación en tres pasos. En el primer paso la maquina origen transmite un segmento con el indicador SYN a uno (para indicar que es un segmento de sincronización) con un número de secuencia N (número de secuencia inicial). En el segundo paso la maquina receptora recibe el segmento y le envía un acuse de recibo (con ACK a uno y SYN también a uno indicando sincronización). Por último (paso 3), la maquina origen transmite un acuse de recibo con ACK a uno y SYN a cero (ya no es de sincronización). Su número de secuencia es incrementado y el acuse de recibo representa el número de secuencia inicial del servidor incrementado en uno. Tras estos pasos las maquinas ya están conectadas y listas para recibir datos.

Transferencia de datos: En esta etapa se utiliza el método de las ventanas para enviar los datos. Este método limita la cantidad de acuses de recibo fijando un número de secuencia después del cual se requiere un acuse de recibo. Este número se guarda en el campo ventana del segmento. Además, el tamaño de esta ventana no es fijo. De hecho, el servidor puede incluir el tamaño de la ventana que considera más apropiado en sus acuses de recibo guardándolo en el campo ventana. De este modo, cuando el acuse de recibo indica un pedido para aumentar la ventana, el cliente se desplazará al borde derecho de la ventana. Por el contrario, en el caso de una reducción, el cliente no desplazará el borde derecho de la ventana hacia la izquierda sino que esperará que avance el borde izquierdo (al llegar los acuses de recibo).

Cierre de la conexión: Puede cerrar la conexión tanto la máquina origen como la máquina receptora. Para cerrar la conexión una de las máquinas envía un segmento con el indicador FIN fijado en 1, y la aplicación se autocoloca en estado de espera, es decir, que deja de recibir el segmento actual e ignora los siguientes. Después de recibir este segmento, la otra máquina envía un acuse de recibo con el indicador FIN fijado en 1 y sigue enviando los segmentos en curso. Después de esto, la máquina informa a la aplicación que se ha recibido un segmento FIN y luego envía un segmento FIN a la otra

máquina, que cierra la conexión. Este intercambio de mensajes se conoce como negociación a cuatro pasos.

2.2 UDP

UDP es el otro protocolo más común en la capa de transporte además de TCP. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión. Este protocolo es mucho más simple que TCP ya que no es orientado a conexión ni tiene ningún tipo de detección de errores.

2.2.1 Características

- Proporciona un mecanismo para distinguir múltiples aplicaciones fuente ó destino en mismo host: los puertos.
- Permite identificar la aplicación destino por su dirección IP y puerto UDP.
- Las aplicaciones que lo usan son las responsables de control de errores, control de flujo, números de secuencia…
- Entrega no confiable.
- Indicado para eventos transmitidos en directo (streaming).

2.2.2 Formato

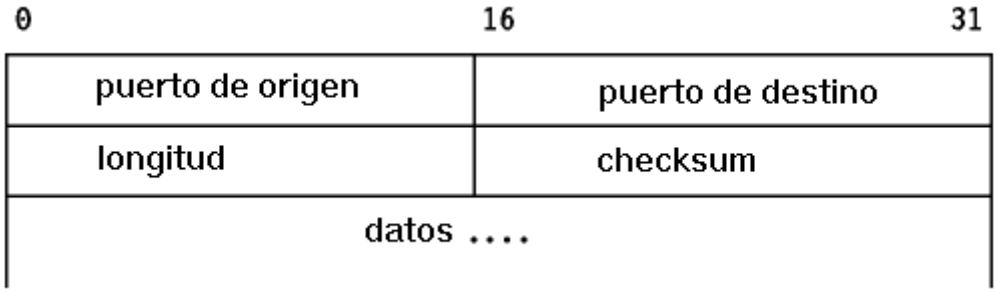


Figura 4: Formato UDP

- Puerto de origen (16 bits): Especifica el puerto de la aplicación que genera el mensaje. Este valdrá normalmente cero, salvo que la aplicación solicite una respuesta.
- Puerto de destino (16 bits): Especifica el puerto de la aplicación a la que va dirigido el mensaje.
- Longitud (16 bits): Especifica la longitud total del segmento, con el encabezado incluido.

- Checksum (16 bits): Opcional en IPv4 y obligatoria en IPv6. Se calcula a partir de una pseudo-cabecera IP (con las IP origen y destino, el protocolo y la longitud del paquete UDP), la cabecera UDP y los datos.

2.2.3 Puertos

Los puertos UDP proporcionan una ubicación para enviar y recibir mensajes UDP. Un puerto UDP funciona como una única cola de mensajes que recibe todos los datagramas destinados al programa especificado mediante cada número de puerto del protocolo.

Hay un rango que va de 0 a 65535 para asignación de puertos.

- Los puertos 1 a 1023 se llaman puertos "bien conocidos".
- Los puertos 1024 a 49.151 son puertos registrados.
- Los puertos 49.152 a 65.535 son puertos efímeros y son utilizados como puertos temporales.

Estos son algunos de los principales puertos UDP:

Puerto UDP	Descripción
53	Consultas de nombre DNS
69	Protocolo de transferencia de archivos TFTP
137	Servicio de nombres NetBIOS
138	Servicio de datagramas NetBIOS
161	Protocolo simple de administración de redes SNMP
520	Protocolo de información de enrutamiento RIP

2.2.4 Comparativa entre TCP y UDP

UDP	TCP
Servicio no orientado a conexión	Servicio orientado a conexión
No garantiza ni confirma la entrega de datos	Garantiza la entrega mediante confirmaciones
Requisitos de carga pequeños	Requisitos de carga mayores
Admite comunicación punto a punto y punto a varios puntos	Solo admite comunicación punto a punto
Rápido	Más lento

2.3 IP

El protocolo IP es el protocolo de red de referencia de internet. Fue creado (al igual que TCP y otros muchos) por el departamento de defensa de los Estados Unidos y, como su nombre indica, pertenece a la capa de internet del modelo TCP/IP.

IP es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

2.3.1 Características

- No detecta ni corrige errores.
- No fiable, es decir, no garantiza la entrega, duplicidad y orden de los paquetes.
- No orientado a conexión.
- Define la unidad de transferencia denominada datagrama o paquete IP.
- Los paquetes se enrutan independientemente.
- Fragmenta los paquetes en caso de ser necesario.
- Direccionamiento de 32 bits jerárquico.
- En estos momentos existen 2 versiones: IPv4 e IPv6.

Durante la explicación se detallará lo relativo al protocolo IPv4, en un apartado posterior se hablará de IPv6.

2.3.2 Formato

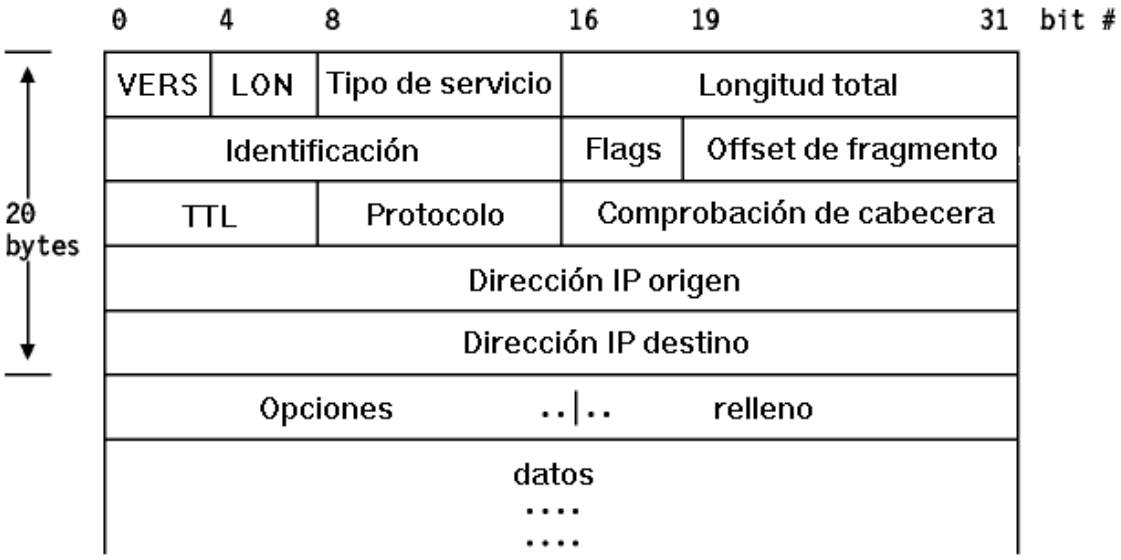


Figura 5: Formato IP

- VERS (4 bits): Numero binario de la versión del protocolo IP del datagrama.
- LON (4 bits): Longitud de la cabecera en palabras de 32 bits.
- Tipo de servicio (8 bits): Tipo de servicio, indica la forma en la que se debe procesar el datagrama.
- Longitud total (16 bits): Tamaño total del datagrama en bytes

Los tres campos siguientes (Identificación, Flags, Offset de fragmento) son específicos de paquetes fragmentados y serán explicados en el apartado 2.3.3.1 Fragmentación.

- TTL (8 bits): Número máximo de router que puede atravesar el datagrama (cantidad de saltos).
- Protocolo (8 bits): Especifica el protocolo de nivel superior que transporta el datagrama.
- Comprobación de cabecera (16 bits): Método de cálculo para comprobar que no se ha modificado el datagrama, se recalcula en cada salto.
- Dirección IP origen (32 bits): Dirección fuente del datagrama (en formato de red).
- Dirección IP destino (32 bits): Dirección destino del datagrama (en formato de red).
- Opciones (Longitud variable): Distintos tipos de opciones que se pueden añadir.
- Relleno (Longitud variable): Espacio a rellenar para ser múltiplo de 32 bits.
- Datos (Longitud variable): Datos propios del datagrama.

2.3.3 Funcionamiento

Los datos en una red basada en el protocolo IP son enviados en bloques conocidos como paquetes IP o datagramas, estos paquetes pueden llegar o no a su destino, por lo que el servicio es no fiable, si se necesita fiabilidad, ésta debe ser proporcionada por los protocolos de transporte.

Si los datos a transmitir supera el tamaño máximo "negociado" en el tramo de red por donde va a circular, podrá ser dividido el paquete en fragmentos más pequeños, y luego ser reensamblados cuando sea necesario. Hay que decir que fragmentos de un mismo paquete pueden llegar por distintos caminos dependiendo de cómo estén de congestionadas las rutas en cada momento.

Para un mayor entendimiento del funcionamiento del protocolo, a continuación se resaltarán los aspectos de fragmentación, direccionamiento y enrutamiento.

2.3.3.1 Fragmentación

El tamaño máximo del datagrama es de 65536 bytes, sin embargo las redes no suelen tener una capacidad para transmitir tamaños de paquetes tan grandes, además, al utilizarse diferentes tecnologías en internet el tamaño máximo de un datagrama puede variar según el tipo de red.

Por todo ello es necesario fragmentar los paquetes. Al tamaño máximo de la trama se le denomina MTU (Unidad de Transmisión Máxima), así que si el datagrama es más grande que este valor se tendrá que fragmentar.

Para poder realizar la fragmentación hay que modificar 4 campos de la cabecera:

- Identificación (16 bits): Identificador del datagrama, se utilizará para reconocer los fragmentos de un datagrama para su posterior reensamblado.
- Flags (3 bits): Está compuesto por 3 bits.

- El primer bit está reservado, no se utiliza.
- El segundo bit (DF: No Fragmentable) indica si se puede fragmentar el datagrama o no. Si está a uno se puede fragmentar.
- El tercer bit (MF: Mas fragmentos) indica si es el último fragmento. Si está a uno es el último fragmento y si está a cero es un fragmento intermedio.

- Offset de fragmento (13 bits): Indica la posición, en unidades de 8 bytes, que ocupa el fragmento actual en el datagrama. Si es el primer fragmento tiene valor cero.
- Longitud total (16 bits): Se vuelve a calcular tras cada fragmento.

2.3.3.2 Direccionamiento

El direccionamiento se refiere a la forma como se asigna una dirección IP y como se dividen y se agrupan subredes de equipos. En primer lugar se explicarán las distintas clases de direcciones.

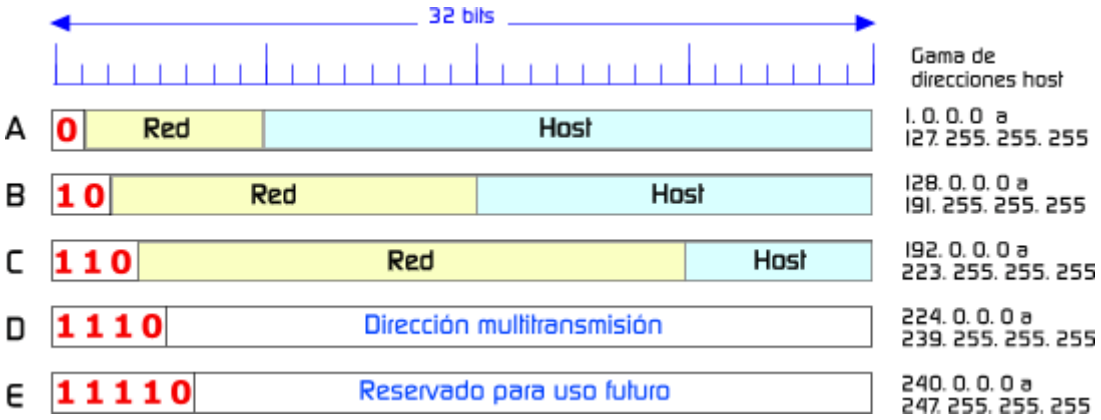


Figura 6: Clases de direccionamiento

El objetivo de dividir las direcciones IP en clases es facilitar la búsqueda de un equipo en la red. De hecho, con esta notación es posible buscar primero la red a la que uno desea tener acceso y luego buscar el equipo dentro de esta red. Por lo tanto, la asignación de una dirección de IP se realiza de acuerdo al tamaño de la red.

En las direcciones de clase A el primer byte representa la red, donde el primer bit a cero indica que es de esta clase y los otros 7 bits codifican la subred. Los 3 bytes (24 bits) restantes indican los $2^{24} - 2$ hosts distintos.

En las direcciones de clase B los dos primeros bytes representan la red, los 2 primeros bits a 10 indican que es de esta clase y los otros 14 bits codifican la subred. Los 2 bytes (16 bits) restantes indican los $2^{16} - 2$ hosts distintos.

En las direcciones de clase C los tres primeros bytes representan la red, los 3 primeros bits a 110 indican que es de esta clase y los otros 21 bits codifican la subred. El byte (8 bits) restantes indican los $2^{8} - 2$ hosts distintos.

Las direcciones de clase D (Multicast) comienzan por la secuencia 1110 y es una dirección especial donde el destinatario no es único.

Las direcciones de clase E (Usos futuros) comienzas por la secuencia 1111 y se reservan para protocolos especiales.

A su vez, las direcciones IP se pueden clasificar en públicas y privadas, y estáticas o dinámicas.

- Pública: Es la dirección IP con la que nos identificamos al conectarnos a otras redes (Internet). Esta IP la asigna el proveedor ISP (Proveedor de servicios de internet), y no tenemos control sobre ella.
- Privada: Es la dirección IP de cada equipo de nuestra red. Al contrario de lo que ocurre con la IP pública, la IP privada sí que la asignamos nosotros, aunque se puede asignar de forma automática.
- Estática: Asigna una dirección IP fija al host por lo que nos conectaremos siempre con una misma IP. Lo utilizan los servidores de Internet y no suele tener uso doméstico.
- Dinámica: Son direcciones que las asigna el proveedor ISP de las que tiene disponible en ese momento y cambia cada vez que nos desconectamos de Internet y nos volvemos a conectar. Es la utiliza habitualmente.

2.3.3.3 Enrutamiento

Enrutar es el proceso de selección de un camino para el envío de paquetes. Es realizado por los routers, que son los encargados de recibir y enviar paquetes por diferentes interfaces de red, así como proporcionar opciones de seguridad, redundancia de caminos y eficiencia en la utilización de los recursos.

Para llevar a cabo esta función es necesario lo llamado *Tabla de enrutamiento IP*. Estas tablas se encuentran en cada host y contiene el conjunto de correspondencias entre direcciones IP destino y el nodo a través del cual el router debe enviar el mensaje.

Se pueden encontrar tres tipos de enrutamiento:

- Rutas directas: Para redes conectadas localmente.
- Rutas indirectas: Para redes alcanzables tras pasar uno o más router.
- Ruta por defecto: Ruta a elegir cuando la IP destino no se encuentra entre las rutas directas e indirectas.

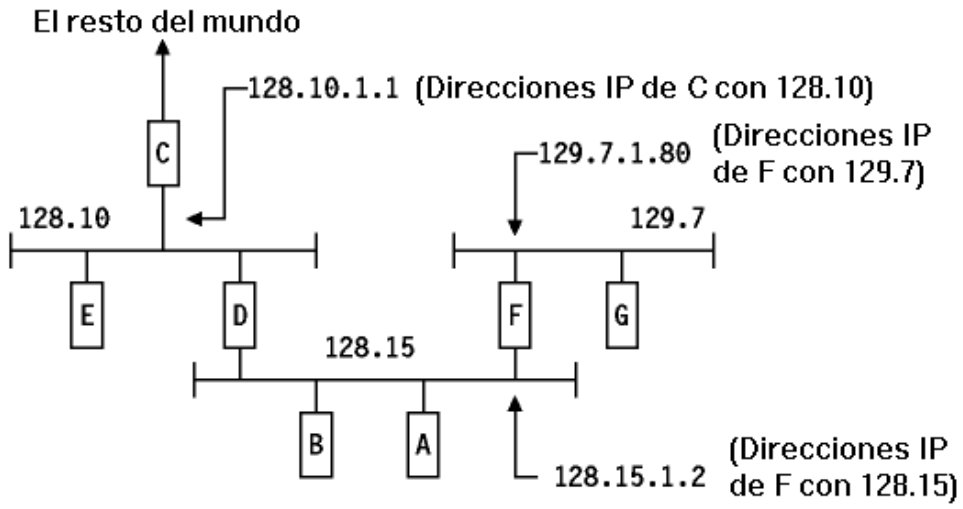


Figura 7: Ejemplo de red de enrutamiento

Para la red de la figura 2.7, la tabla del host D sería la siguiente:

Destino	Ruta
128.10	Conexión directa
128.15	Conexión directa
129.7	128.15.1.2
Por defecto	128.10.1.1

Figura 8: Tabla host D

Con la tabla y figura anterior podemos mandar desde el host D hasta cualquier otro host los datagramas. Para llegar a los hosts A, B, C, E y F existe conexión directa, para llegar al host G mira el identificador de IP y todas las direcciones que contengan 129.7 debe mandarlos a la dirección 128.15.1.2, es decir, al host F. Para otra dirección que no se encuentre en la tabla utiliza el destino por defecto y lo envía a la dirección 128.10.1.1, es decir, al host C.

Es necesario algún mecanismo para elegir el camino que tomará el datagrama, hay multitud de protocolos para ellos, pero a continuación se explicarán los dos principales.

2.3.3.3.1 Protocolo RIP

Es un protocolo de vector de distancias, es decir, que cada router le comunica al resto de los routers la distancia que los separa (la cantidad de saltos que los separa). Cuando un router recibe uno de estos mensajes incrementa esta distancia en 1, y envía el mensaje a routers directamente accesibles. De esta manera, los routers pueden mantener la ruta óptima de un mensaje, al almacenar la dirección del router siguiente en la tabla de enrutamiento de manera tal que la cantidad de saltos para alcanzar una red se mantenga al mínimo. Sin embargo, este protocolo sólo tiene en cuenta la distancia entre equipos

en cuanto a saltos y no considera el estado de la conexión para seleccionar el mejor ancho de banda.

2.3.3.3.2 Protocolo OSPF

Es más eficaz que RIP. Es un protocolo de estado de enlace. Esto significa que este protocolo no envía la cantidad de saltos que los separa de los routers cercanos, sino el estado de la conexión que los separa. De esta manera, cada router puede enviar una tarjeta del estado de la red y, como consecuencia, puede elegir la ruta más apropiada para un determinado mensaje en cualquier momento.

2.3.4 IPv6

Actualmente los dispositivo con acceso a internet (móviles, tabletas, PDA´s) son muy numerosos y sumado al crecimiento de usuarios de internet en el continente asiático el numero de direcciones IP se están agotando. Por esta razón la versión IPv4 da paso a la IPv6, con un mayor rango de direcciones asignables.

En estos momentos conviven ambos protocolos, ya que IPv6 comenzó oficialmente el día 6 de Junio de 2012, pero no son compatibles entre ellos, por el contrario, IPv6 es compatible con la mayoría de los demás protocolos como TCP, UDP, ICMP, DNS…

IPv6 utiliza 128 bits para asignar direcciones (IPv4 “solo” eran 32 bits) lo que posibilita un total de 3,4.10^38 direcciones IP. Además de esta mejora forzosa, el protocolo IPv6 aporta otras, como mayor seguridad o mejor procesamiento de los paquetes por parte de los routers debido al cambio de la cabecera, ya que esta deja de tener campos obligatorios y pasan a ser opcionales. Dicha cabecera es más simple y solo tiene 7 campos por los 14 de la cabecera de IPv6.

2.3.4.1 Formato

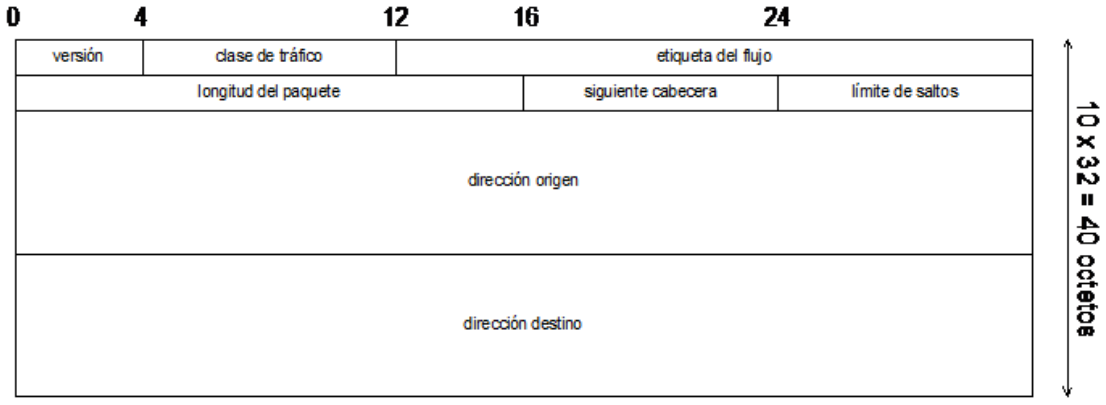


Figura 9: Formato IPv6

- Versión (4 bits): Indica la versión del protocolo en binario. En este caso 0110.
- Clase de tráfico (8 bits): Lo utiliza el emisor y el router para saber si existe algún tipo de prioridad entre paquetes.

- Etiqueta de flujo (20 bits): Utilizado para etiquetar secuencias de paquetes para los cuales se solicita un trato especial por parte de los routers.
- Longitud de paquete (16 bits): Número entero que identifica en octetos el tamaño de todo lo que está fuera de la cabecera.
- Siguiente cabecera (8 bits): Identifica la siguiente cabecera a procesar en el desencapsulamiento del paquete.
- Limite de saltos (8 bits): Entero que decrementa en 1 cada vez que pasa por un router, al llegar a 0 se descartará dicho paquete (antiguo TTL de IPv4).
- Dirección origen (128 bits): Dirección origen del paquete.
- Dirección destino (128 bits): Dirección destino del paquete.

2.3.4.2 Objetivos

- Admitir miles de millones de equipos.
- Reducir el tamaño de las tablas de enrutamiento.
- Simplificar el protocolo para permitir que los routers enruten datagramas de manera más rápida.
- Brindar mejor seguridad (autenticación y confidencialidad).
- Prestar más atención al tipo de servicio y, particularmente, a los servicios asociados con el tráfico en tiempo real.
- Facilitar la difusión a destinos múltiples, permitiendo especificar el tamaño.
- Permitir la movilidad de un equipo sin cambiar su dirección.
- Permitir el futuro desarrollo del protocolo.
- Posibilitar la coexistencia pacífica del protocolo antiguo con el nuevo.

2.4 ICMP

ICMP (Protocolo de Control de Mensajes en Internet) es un protocolo de control y notificación de errores del protocolo IP dado que éste no tiene ningún mecanismo para detectar fallos.

Este protocolo funciona por encima del protocolo IP, pero no se considera que pertenezca al nivel de transporte como UDP o TCP, sino como un complemento obligatorio que tiene que implementar obligatoriamente IP.

2.4.1 Características

- Permite a los routers enviar mensajes de control a otros routers.
- Permite saber, por ejemplo, porque no se ha entregado un datagrama.
- No corrige el problema, solo informa.

- Los mensajes ICMP viajan en el campo de datos de los datagramas IP.
- Si existe un error en un datagrama ICMP no envía ningún mensaje para evitar el efecto “bola de nieve”.

2.4.2 Formato

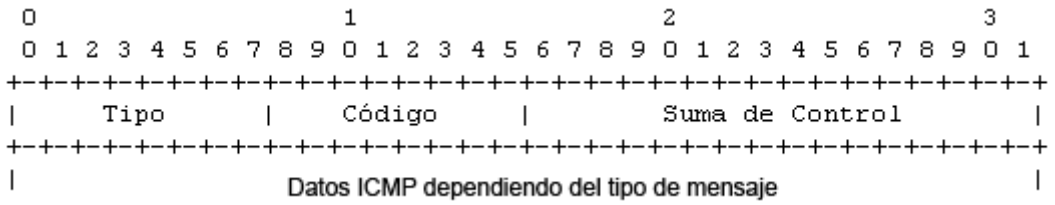


Figura 10: Formato ICMP

- Tipo (8 bits): Tipo de mensaje.
- Código (8 bits): Más información del tipo de mensaje.
- Suma de control o checksum (16 bits): Campo de comprobación de integridad para el total del mensaje ICMP.

Algunos mensajes incluyen información adicional, donde incluyen la cabecera y los primeros 64 bits de datos del datagrama que causó el problema.

Los principales valores que puede tener el campo tipo son los siguientes:

Tipos de mensajes ICMP	
0	Respuesta de eco
3	Destino inalcanzable
4	Disminución de velocidad en origen
5	Redireccionar/cambiar petición
8	Petición de eco
9	Publicación del router
10	Selección del router
11	Tiempo superado
12	Problema de parámetros
13	Petición de marca horaria
14	Respuesta de marca horaria
15	Petición de información
16	Respuesta de información
17	Petición de máscara de dirección
18	Respuesta de máscara de dirección

Figura 11: Campo tipo ICMP

A continuación se explicarán los aspectos más importantes de cada uno.

2.4.2.1 Echo reply (0) y echo request (8)

Se utiliza para detectar si otro host está activo en la red. El emisor inicializa el identificador y número de secuencia (que se utiliza en caso de múltiples peticiones de eco), añade algunos datos al campo de datos y envía el eco ICMP al host de destino. La cabecera ICMP campo de código es cero. El receptor cambia el tipo de respuesta de eco y devuelve el datagrama al remitente. Este mecanismo es utilizado por el comando ping para determinar si un host de destino es alcanzable (si exista ruta hasta él). Se conoce comúnmente como PING de ICMP.

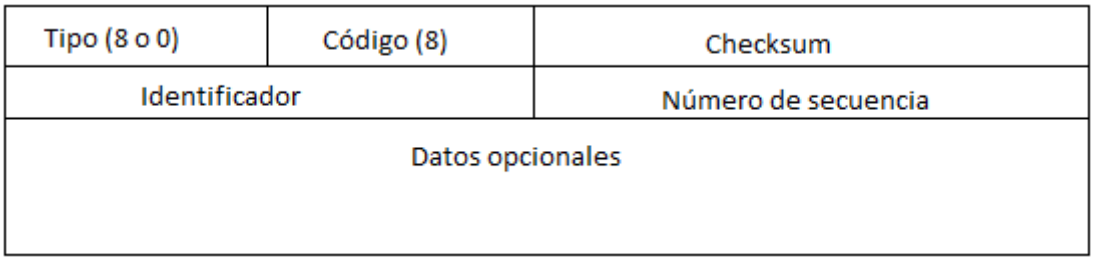


Figura 12: Formato echo request/reply

2.4.2.2 Destino inalcanzable (3)

Este mensaje se genera cuando un router no puede enviar un datagrama IP. Si este mensaje se recibe de un router intermedio, significa que el router se refiere a la dirección IP de destino como inalcanzable. Si este mensaje se recibe desde el host de destino, significa que el protocolo especificado en el campo número de protocolo del datagrama original no está activo, o que el protocolo no está activo en este servidor o si el puerto especificado está inactivo.

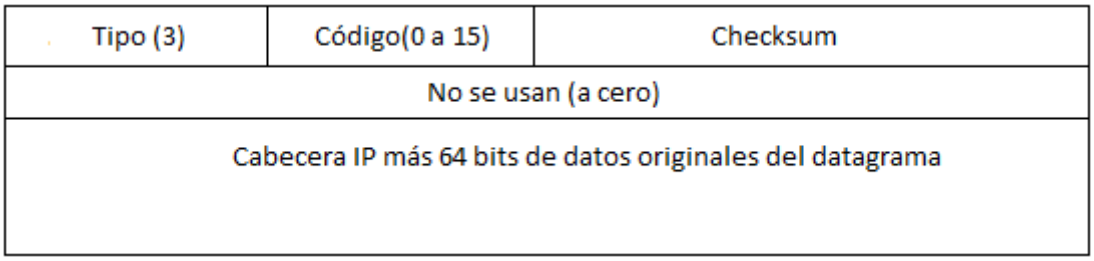


Figura 13: Formato destino inalcanzable

Según el problema que haya tenido se genera un valor distinto en el campo código. Esta es la tabla resumida de la información según el valor.

Valor	Comentario
0	red inalcanzable
1	host inalcanzable
2	protocolo inalcanzable
3	puerto inalcanzable
4	fragmentación necesaria pero el bit *No Fragmentar* estaba activo
5	ruta de origen fallida
6	red de destino desconocida
7	host de destino desconocido
8	host de origen aislado (obsoleto)
9	red de destino administrativamente prohibido
10	host de destino administrativamente prohibido
11	red inalcanzable para este tipo de servicio
12	host inalcanzable para este tipo de servicio
13	comunicación administrativamente prohibido por filtrado
14	violación con anterioridad de host
15	corte con anterioridad

Figura 14: Código destino inalcanzable

2.4.2.3 Disminución de velocidad en origen (4)

Si este mensaje se recibe de un router intermedio, significa que el router no tiene espacio de búfer necesario para poner en la cola los datagramas de salida a la siguiente red. Si este mensaje se recibe del host de destino, significa que los datagramas entrantes están llegando demasiado rápido para ser procesados. El campo del código de cabecera ICMP es siempre cero.

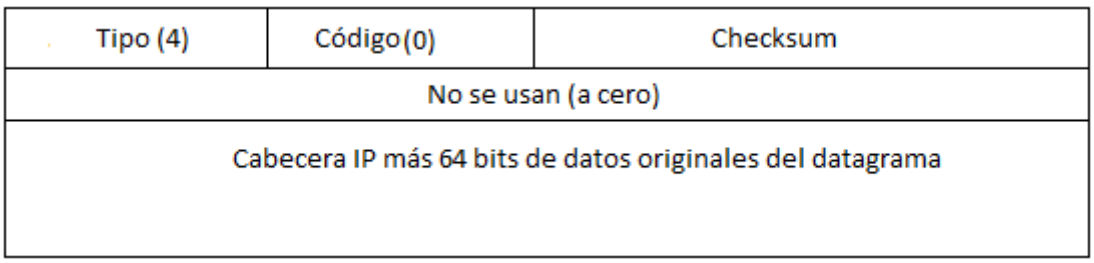


Figura 15: Formato disminución de velocidad en origen

2.4.2.4 Redireccionar (5) o solicitud de cambio de ruta.

Un router intermedio generará un mensaje de redirección ICMP cuando determina que una ruta que se solicita puede alcanzarse ya sea localmente o a través de una mejor.

Tipo(5)	Código (0 a 3)	Checksum
IP router mejor ruta		
Cabecera IP más 64 bits de datos originales del datagrama		

Figura 16: Formato redireccionar

Según el código hace lo siguiente:

0	Redirigir datagramas debido a la red
1	Redirigir datagramas debido al host
2	Redirigir datagramas debido al tipo de servicio y la red
3	Redirigir datagramas debido al tipo de servicio y el host

2.4.2.5 Tiempo excedido (11)

Si este mensaje se recibe de un router intermedio, significa que el campo TTL de un datagrama IP ha caducado (código 0). Si este mensaje se recibe desde el host de destino, significa que el fragmento IP del temporizador TTL ha expirado (código 1), mientras que el anfitrión está a la espera de un fragmento del datagrama.

Tipo(11)	Código (0 o 1)	Checksum
No se usan (a cero)		
Cabecera IP más 64 bits de datos originales del datagrama		

Figura 17: Formato tiempo excedido

2.4.2.6 Problema de parámetros (12)

Indica que se ha encontrado un problema durante el procesamiento de los parámetros de la cabecera IP. El campo puntero apunta a la posición en el datagrama original donde se ha producido el error (código 0). Si es error por falta de parámetro (código 1).

Tipo (12)	Código (0 o 1)	Checksum
Puntero	No se usan (a cero)	
Cabecera IP más 64 bits de datos originales del datagrama		

Figura 18: Formato problema de parámetros

2.4.2.7 Petición de marcha horaria (13) y respuesta de marcha horaria (14) / Sincronización de relojes

Método rudimentario para sincronizar la hora que se mantienen en los distintos dispositivos. Este método para la sincronización de tiempo es complejo y poco confiable. Por lo tanto, no se utiliza intensivamente.

Tipo (13 o 14)	Código (0 o 1)	Checksum
Identificador		Número de secuencia
Originate Timestamp (tiempo tx request)		
Originate Timestamp (tiempo rx request)		
Originate Timestamp (tiempo tx reply)		

Figura 19: Formato sincronización de relojes

2.4.2.8 Petición de información (15) y respuesta de información (16)

Estos tipos ICMP se diseñaron originalmente para permitir que un host de inicio para descubrir una dirección IP. Este método está obsoleto y ya no se utiliza.

Tipo (15 o 16)	Código (0)	Checksum
Identificador		Número de secuencia
Datos opcionales		

Figura 20: Formato de información

2.4.2.9 Petición de mascara de dirección (17) y respuesta de mascara de dirección (18)

Este campo sirve para que su un host conozca su máscara de red a través del mensaje de tipo 17 y le responda con la máscara correspondiente con un mensaje de tipo 18

Tipo (17 o 18)	Código (0)	Checksum
Identificador		Número de secuencia
Máscara de red		

Figura 21: Formato de máscara de dirección

2.4.3 ICMPv6

La anterior versión de ICMP trabajaba con la versión 4 de IP, por lo que al aparecer una nueva versión del protocolo IP (IPv6) también tiene que adaptarse ICMP.

ICMPv6 tiene el mismo formato que la anterior versión y realiza las mismas funciones, además de otras añadidas de otros protocolos como IGMP o ARP para simplificar el proceso de control de errores.

Como diferencia fundamental en los tipos de mensajes que envía ICMPv6 es la forma de tratar los mensajes de error, utilizando los bits menores de 128, y los mensajes de control, utilizando los bits mayores de 128.

2.5 IGMP

El protocolo IGMP funciona como una extensión del protocolo IP. Se emplea para realizar IP multicast, es decir, cuando el envío de datos a una dirección IP puede alcanzar múltiples servidores de una red y/o a todas las máquinas de una subred.

Los mensajes IGMP van encapsulados dentro de datagramas IP, con número de protocolo IP = 2, TTL = 1 y con la opción IP Router Alert en la cabecera IP.

Hay tres versiones:

2.5.1 IGMPv1

2.5.1.1 Formato

00 01 02 03	04 05 06 07	08 09 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Versión	Tipo	No usado	Checksum
Dirección de Grupo			

Figura 22: Formato IGMPv1

- Versión (4 bits): Indica la versión de protocolo. En este caso 0001.

- Tipo (4 bits): Si es 1 es consulta de router (Membership query), y si es 0 es respuesta de host (Membership report).
- No usado (8 bits): Espacio reservado.
- Checksum (16 bits): Método de control del paquete.
- Dirección de grupo (32 bits): Cero cuando el mensaje es tipo 1 (consulta) y dirección del grupo multicast cuando el mensaje es de tipo 0 (respuesta).

2.5.1.2 Funciones

- Unirse a un grupo: Permite unirse al host a un grupo multicast mediante el envío de un mensaje de tipo 0 con la dirección del grupo.
- Preguntas/Respuestas: Permite a los routers multicast saber qué grupos están activos en la subred enviando mensajes de tipo 1 (consulta) a los routers y éstos le contestarán con un mensaje de tipo 0 (respuesta) si están en un grupo
- Abandonar grupo: Los hosts abandonan los grupos sin avisar, dejando de mandar mensajes de respuesta cuando llega un mensaje de pregunta del router.

2.5.2 IGMPv2

2.5.2.1 Formato

Figura 23: Formato IGMPv2

- Tipo (8 bits): Hay 4 tipos diferentes:
    - Membership query (0x11): Dos tipos, general query y group-specific query.
    - Membership report versión 1 (0x12)
    - Membership report versión 2 (0x16)
    - Membership leave group (0x17)

- Tiempo máximo de respuesta (8 bits): Especifica el valor, en décimas de segundo, que un host debe esperar como máximo para contestar a un Membership query.
- Checksum (16 bits): Igual que la versión 1.
- Dirección de grupo (32 bits):
    - 0 para mensajes tipo general query.
    - Dirección del grupo multicast para mensajes de tipo group-specific query, membership report y membership leave group.

2.5.2.2 Funciones

- Unirse a un grupo: Igual que la versión 1.
- Pregunta/respuesta general: Igual que la versión 1.
- Pregunta/respuesta específica: El router pregunta por la existencia en concreto de un grupo. Los hosts responden igual que a una pregunta general.
- Abandonar grupo: El host abandona un grupo mediante el envío de un mensaje leave group.
- Elección del router multicast: Inicialmente el router asume que es el responsable del grupo, hasta que le llegue un general query de un router con un IP menor. El responsable del grupo será el que tenga la IP menor.

2.5.3 IGMPv3

2.5.3.1 Formato

Hay dos tipos de mensajes en la versión 3, query y report.

Tipo query

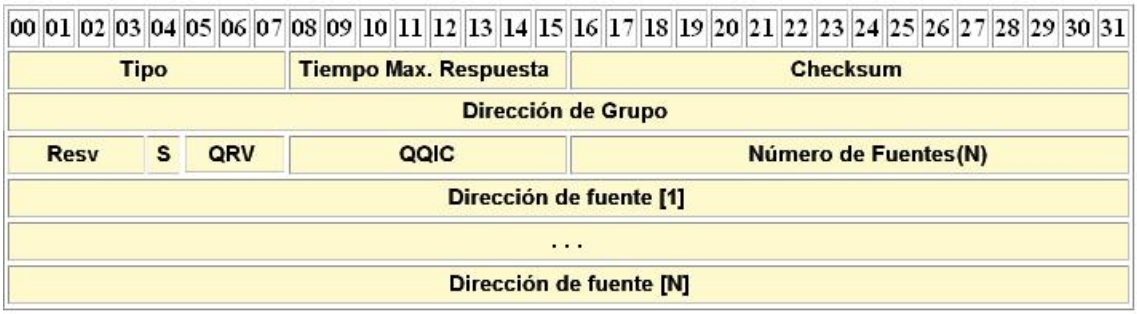


Figura 24: Formato IGMPv3 query

- Tipo (8 bits): Hay 5 tipos diferentes.
  - Membership query (0x11): 3 tipos, general query, group-specific query y group-and-source-specific query.
  - Membership report version 3 (0x22)
  - Membership report version 1 (0x12)
  - Membership report version 2 (0x16)
  - Membership report group version 2 (0x17)
- Tiempo máximo de respuesta (8 bits): Tiempo máximo antes de enviar un report.
- Checksum (16 bits): Igual que la versión 1.
- Dirección de grupo (32 bits): 0 para general queries y la dirección de grupo multicast para otras queries.
- Resv (4 bits): Campo reservado.

- S (1 bit): Si es1, indica a los routers multicast que tienen que suprimir las actualizaciones de los temporizadores cuando escuchen una query, y si es 0 elimina la elección del router multicast.
- QRV (3 bits): Robustez de la variable del consultor (Querier Robustness Variable).
- QQIC (8 bits): Intervalo del router para mandar queries.
- Numero de fuente (16 bits, N): Numero de fuentes presentes en el mensaje query.
- Dirección de fuente (32xN bits): Vector de N direcciones IP unicast, indicando las fuentes.

Tipo report

Figura 25: Formato IGMPv3 report

- Tipo (8 bits): 0x22
- Reservado (8 bits): Espacio reservado.
- Checksum (16 bits): Igual que el anterior.
- Reservado (16 bits): Espacio reservado.
- Número de registros de grupo (16 bits, N): Número de registros que contiene el report.
- Registro de grupo: Campo más amplio. Figura siguiente.

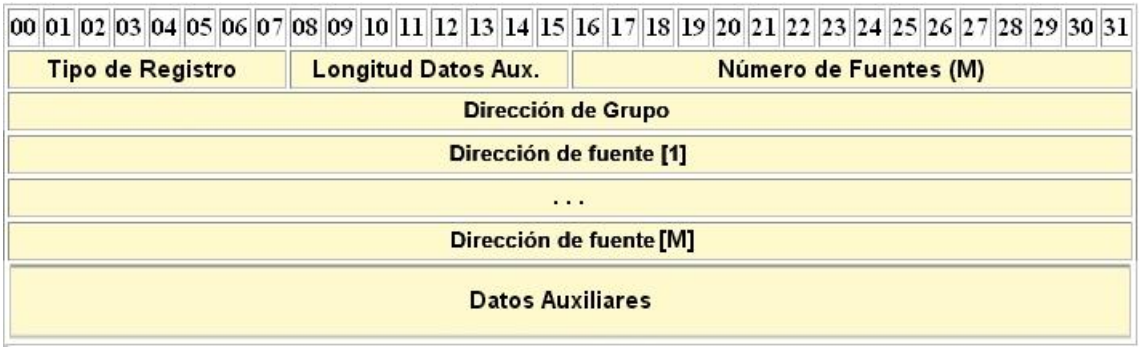


Figura 26: Registro de grupo

- Tipo de registro (8 bits): Estado actual, cambio de modo filtro y cambio de listas de fuentes.
- Longitud de datos auxiliares (8 bits): Longitud del campo datos auxiliares.

- Número de fuentes (16 bits, M): Indica número de fuentes.
- Dirección de grupo (32 bits): Dirección multicast del grupo.
- Dirección de fuente (32xM bits): Igual que en los mensajes queries.
- Datos auxiliares: Información adicional sobre el registro de grupo para versiones futuras. No se debe usar en esta versión

2.5.3.2 Funciones

- Unirse a un grupo: Como en las versiones anteriores, pero además se pueden especificar fuentes dentro de los grupos a las que unirse.
- Pregunta/Respuesta: Como en la versión 2 pero además se pueden hacer preguntas específicas sobre grupos y fuentes en ese grupo.
- Abandonar grupo: Como en la versión 2 y además se pueden dejar una o varias fuentes dentro de un mismo grupo.
- Elección del router multicast: Igual que la versión 2.
- Permite especificar pares grupo/fuente a los hosts para que puedan recibir datos destinados a esos grupos y desde esas fuentes, o desde cualquiera menos esas.

2.6 ARP

ARP es un protocolo de la capa de red que permite conocer la dirección física (MAC) que corresponde a una determinada dirección IP, de ahí su nombre, protocolo de resolución de direcciones (Address Resolution Protocol).

2.6.1 Formato

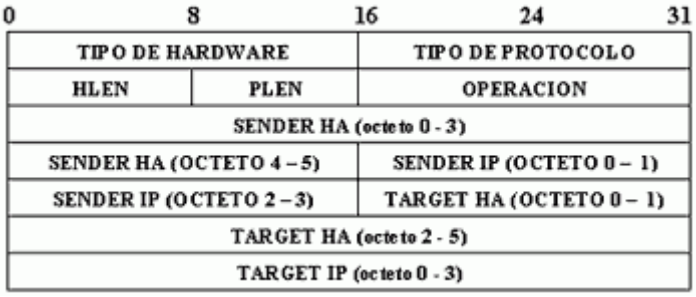


Figura 27: Formato ARP

- Tipo de hardware (16 bits): Identifica tipo de hardware que se utiliza (Ethernet, ATM, HDLC…)
- Tipo de protocolo (16 bits): Especifica tipo de protocolo (IPv4)
- HLEN (8 bits): Longitud de dirección hardware.
- PLEN (8 bits): Longitud de dirección lógica.
- Operación (16 bits): Código de la operación solicitud (1) o respuesta (2).
- Sender HA (48 bits): Dirección de origen hardware.

- Sender IP (32 bits): Dirección de origen IP.
- Target HA (48 bits): Dirección de destino hardware.
- Target IP (32 bits): Dirección de destino IP.

2.6.2 Funcionamiento

En primer lugar, se envían tramas ARP a todos los equipos de la red para averiguar sus direcciones físicas y luego crear una tabla de búsqueda entre las direcciones lógicas y físicas en una memoria caché. Esta tabla es conocida como tabla caché ARP.

Cuando un host quiere comunicarse con otro, busca en su tabla ARP la dirección. Ahora hay dos posibles soluciones para encontrar dicha dirección. Si se encuentra en la tabla ARP no hay ningún problema y envía el datagrama, ya que conoce el par de direcciones a los que mandarla, pero si no la conoce tiene que mandar una trama ARP a todos los hosts de la red. Si algún host reconoce su propia dirección IP, devuelve una trama ARP al host que la solicito con su dirección física, así como información de encaminamiento. Esta dirección se almacenará en ambos hosts para un posterior uso.

Existe otro protocolo que tiene la misma función que ARP pero actúa a la inversa: RARP.

2.6.3 RARP

Este protocolo permite conocer la dirección IP a partir de la dirección física (MAC). Es mucho menos utilizado que ARP ya que es más complejo y prácticamente solo se utiliza para estaciones de trabajo sin discos duros que desean conocer su dirección física.

2.7 DNS

El protocolo DNS es una base de datos jerárquica y distribuida que contiene asignaciones entre nombres de hosts DNS y varios tipos de datos.

Se encuentra en la capa de aplicación de TCP/IP y su función principal es traducir nombres de dominio a direcciones IP y viceversa. También se utiliza para localizar los servidores de correo electrónico.

Este protocolo fue creado por Paul Mockapetris en el año 1983 para implementar un sistema de gestión para los nombres que fuese jerárquico y fácil de administrar debido al mayor tamaño de las redes y sus interconexiones. La versión inicial creada por Paul Mockapetris ha quedado obsoleta debido a todas las mejoras dadas al protocolo DNS.

2.7.1 Terminología básica.

Antes de seguir, es necesario conocer algunos términos para un mejor entendimiento posterior.

- **Host Name:** El nombre de host es una sola palabra (formada por letras, números y guiones) como por ejemplo “www”, “teleco”, “upct”…

- **Domain Name**: El nombre de dominio es una sucesión de nombres (etiquetas) concatenados por puntos. Algunos ejemplos son “teleco.upct.es“, “upct.es”, “es”…

- **Fully Qualified Host Name** (**FQHN**): Es el nombre completo de un host. Está formado por el hostname, seguido de un punto y su correspondiente nombre de dominio. Por ejemplo, “www.teleco.upct.es”

- **Top Level Domains** (**TLD**): Los dominios de nivel superior son aquellos que no pertenecen a otro dominio. Ejemplos de este tipo son “com“, “org“, “uk” o “es“.

- **Clientes DNS:** El encargado de generar peticiones DNS a un servidor DNS.

- **Servidores DNS:** Almacenan y responden a las consultas DNS.

- **Zonas de autoridad:** Porción del espacio de nombres de dominio de la que es responsable un determinado servidor DNS.

2.7.2 Espacio de nombres

La estructura del sistema DNS se basa en una estructura en forma de árbol en donde se definen los dominios de nivel superior (TLD), esta estructura está conectada a un nodo raíz representado por un punto.

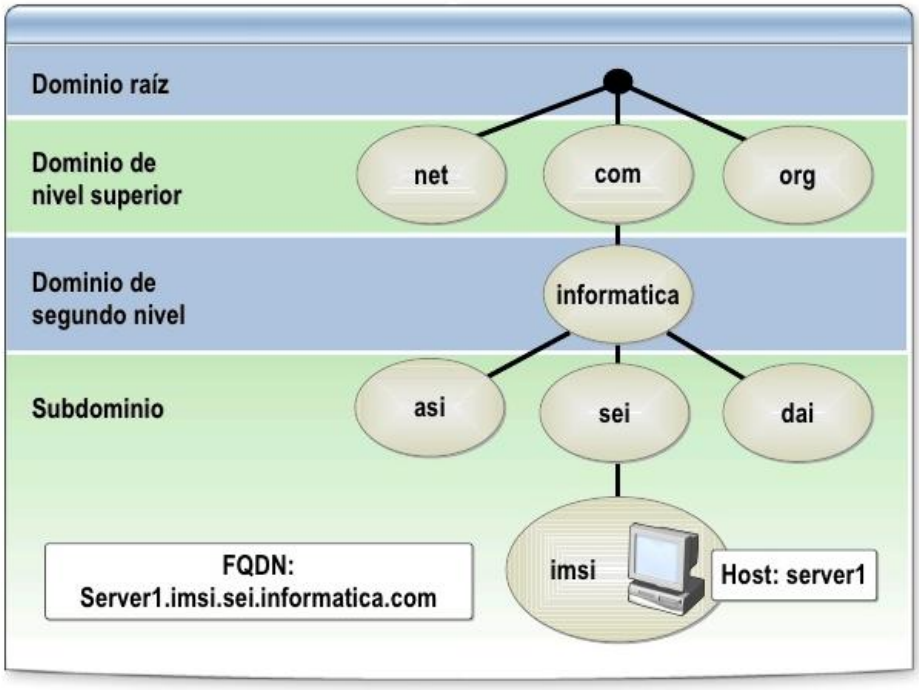


Figura 28: Estructura DNS

Cada nodo del árbol es un nombre de dominio y tiene una etiqueta con una longitud máxima de 63 caracteres.

Por lo tanto, todos los nombres de dominio conforman una estructura arbórea inversa en donde cada nodo está separado del siguiente nodo por un punto.

El extremo de la bifurcación se denomina host, y corresponde a un equipo de la red. El dominio del servidor Web por lo general lleva el nombre www.

La palabra "dominio" corresponde formalmente al sufijo de un nombre de dominio, es decir, la recopilación de las etiquetas de nodo de la estructura arbórea, con excepción del ordenador. La parte situada más a la derecha es el TDL y cada una de las partes es un subdominio de la parte que tiene a su derecha.

La profundidad máxima de una estructura arbórea es 127 niveles y la longitud máxima para un nombre FQDN es 255 caracteres. La dirección FQDN permite ubicar de manera única un equipo en la red de redes.

2.7.3 Servidores de nombres de dominio

Los equipos llamados servidores de nombres de dominio permiten establecer la relación entre los nombres de dominio y las direcciones IP de los equipos de una red. Estos tipos de servidores definen una zona, es decir, una recopilación de dominios sobre la cual tiene autoridad.

Tipos de servidores:

- **Primarios:** Almacenan la información de su zona en una base de datos local. Son responsables de mantener la información actualizada y cualquier cambio debe ser notificado a este servidor.
- **Secundarios:** Son aquellos que obtienen los datos de su zona desde otro servidor que tenga autoridad para esa zona. El proceso de copia de la información se denomina transferencia de zona.
- **Maestros:** Son los que transfieren las zonas a los servidores secundarios. Cuando un servidor secundario arranca busca un servidor maestro y realiza la transferencia de zona. Un servidor maestro para una zona puede ser a la vez un servidor primario o secundario de esa zona. Estos servidores extraen la información desde el servidor primario de la zona. Así se evita que los servidores secundarios sobrecarguen al servidor primario con transferencias de zonas.
- **Locales o caché:** No tienen autoridad sobre ningún dominio, se limitan a contactar con otros servidores para resolver las peticiones de los clientes DNS. Estos servidores mantienen una memoria caché con las últimas preguntas contestadas. Cada vez que un cliente DNS le formula una pregunta, primero consulta en su memoria caché. Si encuentra la dirección IP solicitada, se la devuelve al cliente; si no, consulta a otros servidores, apuntando la respuesta en su memoria caché y comunicando la respuesta al cliente.

2.7.4 Resolución de nombres de dominio.

La acción de resolución de nombres de dominio se conoce como “consulta”, y es una petición que se envía a un servidor DNS. Hay de dos tipos: recursivas e iterativas.

- **Recursiva:** Obliga al servidor DNS a que responda aunque tenga que consultar a otros servidores. El servidor no tiene la información en sus datos locales. Esta opción es más frecuente.

- **Iterativa:** El servidor contesta si tiene la información y si no, le remite la dirección de otro servidor capaz de resolver. De esta forma el cliente tiene mayor control sobre el proceso de búsqueda.

2.7.5 Tipos de registro

Los registros están asociados a DNS y se utilizan para almacenar distinta información. Hay muchos tipos, pero los siguientes son los más importantes.

- A (Dirección): Este registro se utiliza para traducir nombres de hosts del dominio en cuestión a direcciones IP.
- CNAME (Nombre canónico): El nombre canónico es un alias para un host determinado.
- NS (Nombre Servidor): Especifica el servidor (o servidores) de nombres para un dominio.
- MX (Intercambiador de correo): Define el servidor encargado de recibir el correo electrónico para el dominio.
- PTR (Puntero/Indicador): Especifica un registro inverso, a la inversa del registro A, permitiendo la traducción de direcciones IP a nombres.
- TXT (Texto): Permite asociar información adicional a un dominio. Esto se utiliza para otros fines, como el almacenamiento de claves de cifrado.

2.8 HTTP

El protocolo de transferencia de hipertexto, HTTP, es un protocolo que se encuentra en la capa de aplicación de TCP/IP y es usado para la transferencia de información entre sistemas, de forma clara y rápida. Este protocolo es el más utilizado en internet.

Fue creado por Tim Berners-Lee (considerado el padre de la web) en 1989 y tiene varias versiones posteriores. Está basado en el modelo cliente/servidor para intercambiar información entre los clientes web y los servidores HTTP.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor

responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan.

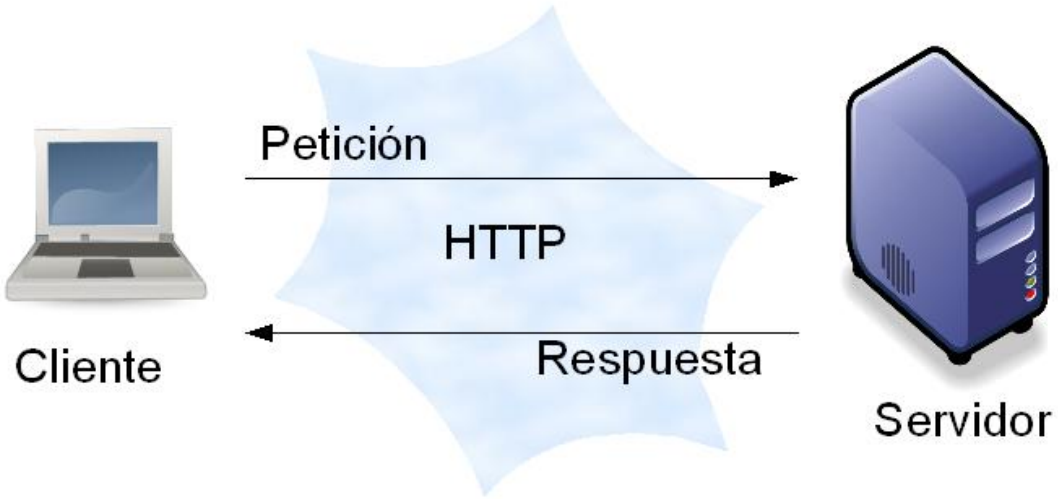


Figura 29: Comunicación HTTP

2.8.1 Versiones

- **HTTP/0.9:** Es la primera versión. Definía un protocolo sencillo a nivel de aplicación para distribución de datos a través de redes.
- **HTTP/1.0:** Permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MIME. Disponible detalladamente en el RFC 1945.
- **HTTP/1.1:** Debido a las limitaciones de la anterior versión (definida en el RFC 2616), se creó esta nueva versión que ofrece mejoras como:

  - Integración de conexiones persistentes (Keep-Alive) en el protocolo.
  - Búsqueda de incremento en el desempeño. Definición de uso de proxies HTTP.
  - Soporte a host virtuales basados en nombre. Para ello se obliga a que las peticiones HTTP/1.1 incluyan un nuevo campo de encabezado llamado Host.
  - Negociación de contenido. Clientes y servidores pueden, mediante intercambio de cabeceras, negociar características comunes. Cuando el servidor ofrece la información en diversas representaciones, el cliente puede seleccionar la que más le interesa.
  - Mantener sesiones por medio de cookies.
  - Nuevos métodos de petición (OPTIONS, TRACE, DELETE, PUT, CONNECT).

2.8.2 Características

- Toda la comunicación entre los clientes y servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original.
- Permite la transferencia de objetos multimedia. El contenido de cada objeto intercambiado está identificado por su clasificación MIME.
- Existen tres verbos básicos (hay más, pero por lo general no se utilizan) que un cliente puede utilizar para dialogar con el servidor: GET, para recoger un objeto, POST, para enviar información al servidor y HEAD, para solicitar las características de un objeto
- Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Es decir, en una operación se puede recoger un único objeto. En la actualidad se ha mejorado este procedimiento, permitiendo que una misma conexión se mantenga activa durante un cierto periodo de tiempo, de forma que sea utilizada en sucesivas transacciones. Este mecanismo, denominado HTTP Keep Alive, es empleado por la mayoría de los clientes y servidores modernos.
- No mantiene estado. Cada petición de un cliente a un servidor no es influida por las transacciones anteriores. El servidor trata cada petición como una operación totalmente independiente del resto.
- Cada objeto al que se aplican los verbos del protocolo está identificado a través de la información de situación del final de la URL.

2.8.3 Comunicación HTTP

2.8.3.1 Formato de mensajes

Hay dos tipos de mensajes, petición y respuesta, cada uno con su estructura, pero a groso modo el formato de un mensaje genérico sería el siguiente:

- **Línea de solicitud:** Línea inicial donde indica que hacer (petición) o que ha sucedido (respuesta).
- **Cabecera:** Contiene los atributos del mensaje.
- **Cuerpo:** Es opcional. Su presencia depende de la petición y del resultado.

2.8.3.2 Peticiones

Una solicitud HTTP es un conjunto de líneas que el navegador envía al servidor. Incluye:

- **Línea de solicitud:** La línea está formada por tres elementos que deben estar separados por un espacio: Método, dirección URL y versión.
- **Cabecera:** Aporta información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

- **Cuerpo:** Opcional.

Esta es la sintaxis de petición HTTP:

```
MÉTODO URL VERSIÓN<crlf>
  CABECERA: Valor<crlf>
. . . CABECERA: Valor<crlf>
  Línea en blanco <crlf>
  CUERPO DE LA SOLICITUD
```

Y este podría ser un ejemplo de petición:

```
GET http://teleco.upct.es HTTP/1.0 Accept : Text/html If-Modified-
Since : Saturday, 15-Julio-2012 14:37:11 GMT User-Agent : Mozilla/14.0
```

De entre los tres parámetros de la línea de solicitud el más importante es el método. HTTP/1.0 incorpora 8 métodos, aunque solo obliga a implementar GET y HEAD, siendo los demás opcionales. En HTTP/1.0 solo se implementan 3 métodos, GET, HEAD y POST.

2.8.3.2.1 Métodos

- **GET:** Solicita el recurso ubicado en la URL especificada. Para reducir el tráfico en la red, se crearon dos variantes de este método:
  - GET condicional: Se transmite el contenido de la respuesta solo si se cumplen unas condiciones determinadas por algunas cabeceras como las siguientes: If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, If-Range.
  - GET parcial: Se transmite solo parte del contenido pedido. Se realiza con la cabecera Range.
- **HEAD:** Es igual que el método GET, salvo que el servidor no tiene que devolver el contenido, solo las cabeceras. Se utiliza para preguntar información sobre un documento. HEAD es mucho más rápido que GET, ya que se transfiere mucha menos información. Es generalmente usada por clientes que usan caché para ver si el documento ha cambiado desde la última vez que accedieron a él. También es usado para testear la validez, accesibilidad y reciente modificación de enlaces.
- **POST:** Es usado para transferir datos del cliente al servidor. Estos datos son enviados en el cuerpo del mensaje. La URL corresponde a una página dinámica que trata los datos enviados.
- **PUT:** Almacena un documento en la URL especificada. Si la dirección de destino ya contenía un documento, se considera que se está enviando una versión actualizada del mismo. Es el camino más eficiente para subir archivos a un servidor.
- **DELETE:** Elimina el documento referenciado en la URL.
- **TRACE:** Este método se utiliza para saber si existe el receptor del mensaje, rastrear los intermediarios por los que pasa la petición y así usar la información para hacer un diagnóstico.

- **CONNECT:** Este método se reserva para uso con proxys. Permitirá que un proxy pueda dinámicamente convertirse en un túnel. Por ejemplo para comunicaciones con SSL.
- **OPTIONS:** Pide información sobre las características de comunicación proporcionadas por el servidor. Le permite al cliente saber los métodos HTTP que soporta el servidor y así negociar los parámetros de comunicación.

2.8.3.3 Respuestas

Una respuesta HTTP es un conjunto de líneas que el navegador envía al servidor. Incluye:

- **Línea de solicitud:** La línea está formada por tres elementos que deben estar separados por un espacio: Versión del protocolo, código de error y texto explicativo del código anterior.
- **Cabecera:** Mismo formato que las de petición.
- **Cuerpo:** Contiene el documento solicitado.

Así que una respuesta HTTP tiene la siguiente sintaxis:

```
VERSION CÓDIGO-ERROR EXPLICACION<crlf>
        CABECERA: Valor<crlf>
    . . . CABECERA: Valor<crlf>
      Línea en blanco <crlf>
      CUERPO DE LA SOLICITUD
```

Y este podría ser un ejemplo de respuesta HTTP:

```
HTTP/1.0 200 OK Date: Sat, 15 Jul 2012 14:37:12 GMT Server :
Microsoft-IIS/2.0 Content-Type : text/HTML Content-Length : 1245 Last-
Modified : Fri, 14 Jul 2012 08:25:13 GMT
```

El apartado más importante de las respuestas HTTP es el código de error.

2.8.3.3.1 Códigos de error

El código de error es un número de tres cifras que indica si la petición ha sido atendida satisfactoriamente o no, y en caso de no haber sido atendida, indica la causa. Se dividen en cinco clases según el primer digito:

- 1xx: Mensajes informativos.
- 2xx: Mensajes de éxito.
- 3xx: Mensajes de redirección.
- 4xx: Mensajes de error del cliente.
- 5xx: Mensajes de error del servidor.

Un ejemplo de códigos típicos de error son los siguientes:

Código	Significado
200 OK	La solicitud del cliente fue satisfactoria y el servidor ha devuelto la información solicitada.
204 No Content	El cuerpo de la respuesta no tiene contenido. Esto puede indicar, por ejemplo, un problema con un CGI que no devuelve datos.
301 Moved Permanently	El URL solicitado no está disponible en el servidor. Ha sido movido a otra ubicación. Las solicitudes futuras deberán hacerse a esa ubicación.
400 Bad Request	Hay un error de sintaxis en la solicitud del cliente. Por ejemplo mandar una solicitud indicando que el cliente soporta HTTP/1.1 y no enviar el encabezado de Host.
404 Not Found	Este es junto con el 200 OK, el código más habitual. Indica que el documento solicitado no está disponible, probablemente el URI haya sido mal escrito.
500 Internal Server Error	Este mensaje indica que algo ha ido mal en el servidor, casi siempre tiene que ver con problemas en programas CGI.

Figura 30: Tabla códigos de error HTTP

2.8.3.4 Cabeceras

Hay cuatro tipos de cabeceras en HTTP: Generales, de petición, de respuesta y de entidad.

- **Cabeceras generales:** Este tipo de cabeceras se aplican tanto a las peticiones como a las respuestas. Son las siguientes:

Encabezado	Significado
Cache-Control	Permite especificar distintas directivas para controlar la caché, tanto del cliente como de servidores proxy.
Connection	Especifica opciones de la conexión de red.
Date	Envía una fecha en la representación estándar definida en el protocolo.
Pragma	Transporta información no HTTP a un receptor que sea capaz de entenderla.

Trailer	Indica que ciertas cabeceras HTTP pueden encontrarse en el final de un mensaje con múltiples partes (multipart).
Upgrade	Información sobre protocolos adicionales soportados por el cliente.
Via	Añadidos al mensaje de proxys o gateways para indicar que pasó por ellos.
Warning	Información adicional sobre un estado o transformación de un documento que podría no estar reflejado en el cuerpo del mismo. Por ejemplo, transformaciones que se hacen en servidores caché.

Figura 31: Tabla cabecera general HTTP

- **Cabeceras de petición:** Los utiliza el cliente para enviar (en sus peticiones de servicio) información adicional al servidor.

Encabezado	**Significado**
Accept	Listado de tipos MIME que el cliente soporta. Hay otros encabezados relacionados como Accept-Charset , Accept-Encoding , Accept-Language .
Authorization	Indica las credenciales de acceso a un recurso que presenta el usuario.
Expect	Indica que comportamiento del servidor necesita el cliente.
From	Dirección de correo que controla el cliente (navegador).
Host	Nombre o IP del host desde donde se conecta el cliente.
If-Match	Un cliente que tiene recursos en cache puede verificar si están actualizados incluyendo este encabezado. Hay otros encabezados que también tienen que ver con la caché. If-Modified-Since , If-None-Match , If-Range , If-Unmodified-Since .
Max-Forwards	Cuantas veces la petición del cliente puede ser reenviada por proxys.
Proxy-Authoritation	Indica las credenciales de acceso a un proxy que presenta el usuario.
Range	Indica que porción de recurso (rango de bytes) recuperar.
Referer	Es el URI del recurso desde donde la petición se ha realizado (generalmente por provenir de un enlace HTML).
TE	Que codificaciones de transferencia está dispuesto a recibir el cliente.
User-Agent	Información sobre el agente de usuario (generalmente navegador) que origina la petición.

Figura 32: Tabla cabecera de petición HTTP

- **Cabeceras de respuesta:** Los utiliza el servidor para enviar información adicional al cliente

Encabezado	Significado
Accept-Ranges	Indica las unidades en las que el servidor acepta peticiones de rangos.
Age	Tiempo estimado por el servidor para cumplir la petición.
Etag	Valor actual de la etiqueta de entidad solicitada.
Location	Contiene un URI al que el cliente debe ser redireccionado.
Proxy-Authenticate	Indica el esquema de autentificación que acepta un servidor proxy.
Retry-After	Cuanto tiempo se espera que el servicio no esté disponible.
Server	Información del software servidor.
Vary	Indica que un recurso tiene múltiples fuentes que pueden variar de acuerdo a la lista de encabezados de petición.
WWW-Authenticate	Indica que el recurso solicitado necesita de credenciales de autorización.

Figura 33: Tabla cabecera de respuesta HTTP

- **Cabeceras de entidad:** Contiene información relacionada directamente con el recurso que se le va a proporcionar al cliente.

Encabezado	Significado
Allow	Informa al cliente de los métodos válidos asociados con el recurso.
Content-Type	Indica el tipo MIME de los contenidos. Hay otros encabezados muy relacionados como Content-Language , Content-Length , Content-Location , Content-MD5 , Content-Range o Content-Encoding .
Expires	Indica la fecha y hora en la que el recurso se considerará obsoleto.
Last-Modified	Indica la fecha y hora en la que el recurso original fue modificado por última vez.

Figura 34: Tabla cabecera de entidad HTTP

2.8.3.5 Esquema de una comunicación

La estructura de un proceso de comunicación normal sigue los siguientes pasos:

1. Un programa cliente establece la conexión con un programa servidor Web.
2. El cliente envía una petición indicando el método, URI a la que pretende acceder y versión del protocolo. Además se pueden enviar diversos encabezados HTTP. Un ejemplo simple de petición es el siguiente:

```
GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1
User-Agent: amano/1.0 [es]
```

3. El servidor responde con una línea de estado, que incluye la versión de protocolo, un código de éxito o error y el texto explicativo al código. Además se enviarán varios encabezados HTTP adicionales. Finalmente, dentro del cuerpo del mensaje aparecerán los datos solicitados. Un ejemplo simple de respuesta es el siguiente:

```
HTTP/1.1 400 Bad Request
Date: Fri, 16 Jul 2012 18:26:55 GMT
Server: Apache/1.3.20 (Unix)
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1

175
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
...
```

4. El programa servidor cierra la conexión.

2.9 SSH

SSH es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente.

Además, permite copiar datos de forma segura, gestionar claves RSA (sistema criptográfico) para no escribir claves al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro (tunelling) mediante SSH.

Este protocolo provee fuerte autenticación, por lo que reemplaza a métodos menos seguros como telnet, rsh y rcp.

2.9.1 Características

- Tras la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.
- El cliente transmite su información de autenticación al servidor usando una encriptación robusta de 128 bits.
- Todos los datos enviados y recibidos durante la sesión se transfieren por medio de encriptación de 128 bits, lo cual los hacen extremadamente difícil de descifrar y leer.
- El cliente tiene la posibilidad de reenviar aplicaciones X11 desde el servidor. Esta técnica, llamada *reenvío por X11*, proporciona un medio seguro para usar aplicaciones gráficas sobre una red.

2.9.2 Versiones

Actualmente existen dos versiones del protocolo SSH, la original y patentada versión 1, que sufre de algunos agujeros de seguridad, y la versión 2 que no es vulnerable al hueco de seguridad e implementa un algoritmo de intercambio de claves mejorado. Es altamente recomendable usar la versión 2 del protocolo, aunque la 1 sigue siendo compatible.

2.9.3 Funcionamiento

Al conectarse un cliente de SSH con el servidor, se realizan los siguientes pasos:

1. El cliente abre una conexión TCP al puerto 22 del host servidor.

2. El cliente y el servidor acuerdan la versión del protocolo a utilizar, de acuerdo a su configuración y capacidades.

3. El servidor posee un par de claves pública/privada de RSA (llamadas “claves de host”). El servidor envía al cliente su clave pública.

4. El cliente compara la clave pública de host recibida con la que tiene almacenada, para verificar su autenticidad. Si no la conociera previamente, pide confirmación al usuario para aceptarla como válida.

5. El cliente genera una clave de sesión aleatoria y selecciona un algoritmo de cifrado simétrico.

6. El cliente envía un mensaje conteniendo la clave de sesión y el algoritmo seleccionado, cifrado con la clave pública de host del servidor usando el algoritmo RSA.

7. En adelante, para el resto de la comunicación se utilizará el algoritmo de cifrado simétrico seleccionado y clave compartida de sesión.

8. Luego se realiza la autenticación del usuario. Aquí pueden usarse distintos mecanismos.

9. Finalmente se inicia la sesión.

2.9.4 Métodos de autenticación

Existen varios métodos de autenticación en SSH. Los dos siguientes son los métodos más importantes:

- **Autenticación con contraseña:** Es el método más simple. El cliente solicita al usuario el ingreso de una contraseña (password) y la misma es enviada al servidor, el cuál validará la misma utilizando los mecanismos configurados en el sistema. Hay varias desventajas, pero las dos principales son que el cliente siempre tiene que introducir la contraseña cada vez que se conecta al servidor, y que ésta contraseña siempre va hacia al servidor, por lo que si el sistema está manipulado puede usarse con usos maliciosos.
- **Autenticación con clave pública:** En este caso, el usuario debe poseer un par de claves pública/privada, y la clave pública debe estar almacenada en el servidor. Tras establecer la conexión, el servidor genera un número aleatorio que es cifrado con la clave pública del usuario usando RSA o DSA. El texto cifrado es enviado al cliente, que debe descifrarlo con la clave privada correspondiente y devolverlo al servidor, demostrando de esta manera que el usuario es quien dice ser. Con este tipo de autenticación se solucionan las dos desventajas del método anterior, ya que no se introduce ninguna contraseña, y ninguna contraseña secreta (en este caso la privada), es enviada al servidor.

2.10 Otros

Existen otros muchos protocolos importantes, sobre todo en la capa de aplicación (FTP, DHCP, SMTP, TELNET), pero los explicados son los principales que se encuentran en el propósito de este proyecto.

Capítulo 3: Herramientas para capturar y analizar tráfico de red.

3.1 Wireshark

La primera herramienta para analizar tráfico que vamos a detallar es posiblemente la más conocida de todas: Wireshark.

Wireshark, conocido anteriormente como Ethereal, es una herramienta para analizar el tráfico de la red y sirve como una aplicación didáctica para el estudio de las comunicaciones y la resolución de problemas en red.

La funcionalidad que ofrece es similar a lo que ofrece tcpdump (explicado posteriormente), pero con la ventaja de tener una interfaz grafica que facilita todo el trabajo.

Wireshark incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

3.1.1 Características

Las características más importantes de esta herramienta son:

- Permite ver, a un nivel bajo y detallado, que está pasando en una red, además de permitir la captura en el momento desde una interfaz de red.
- Este software cuenta con una interfaz gráfica lo que facilita su utilización especialmente para usuarios no muy avanzados y que no están acostumbrados a la operación mediante líneas de comandos, pero también cuenta con una versión basada en texto llamada Tshark.
- Wireshark es software libre, y se ejecuta sobre la mayoría de sistemas operativos UNIX, incluyendo Linux, Solaris, FreeBSD, NetBSD, OpenBSD, y MacOS X, así como en Microsoft Windows.
- Muestra los paquetes con información detallada.
- Abre y guarda paquetes capturados.
- Importa y exporta paquetes capturados, así es posible compatibilizarlos con otros software de características parecidas.
- Filtrado de información de paquetes.
- Resaltado de paquetes dependiendo el filtro.
- Crear estadísticas.

- Trabaja con maquinas conectadas de forma inalámbrica o de forma cableada.
- Captura datos de la red o lee datos almacenados en un archivo de una captura previa.
- Gran capacidad de filtrado.
- Reconstrucción de sesiones TCP.
- Compatible con más de 480 protocolos.

Esta herramienta puede usarse en cualquier ámbito, ya sea en el profesional o en el académico.

Los administradores lo usan para resolver problemas en red, los ingenieros para examinar problemas de seguridad, los desarrolladores para depurar la implementación de protocolos de red y los estudiantes (como es el caso) para comprender y aprender el funcionamiento interno de la red.

3.1.2 Interfaz

En esta imagen se puede apreciar cuales son las zonas básicas de Wireshark:

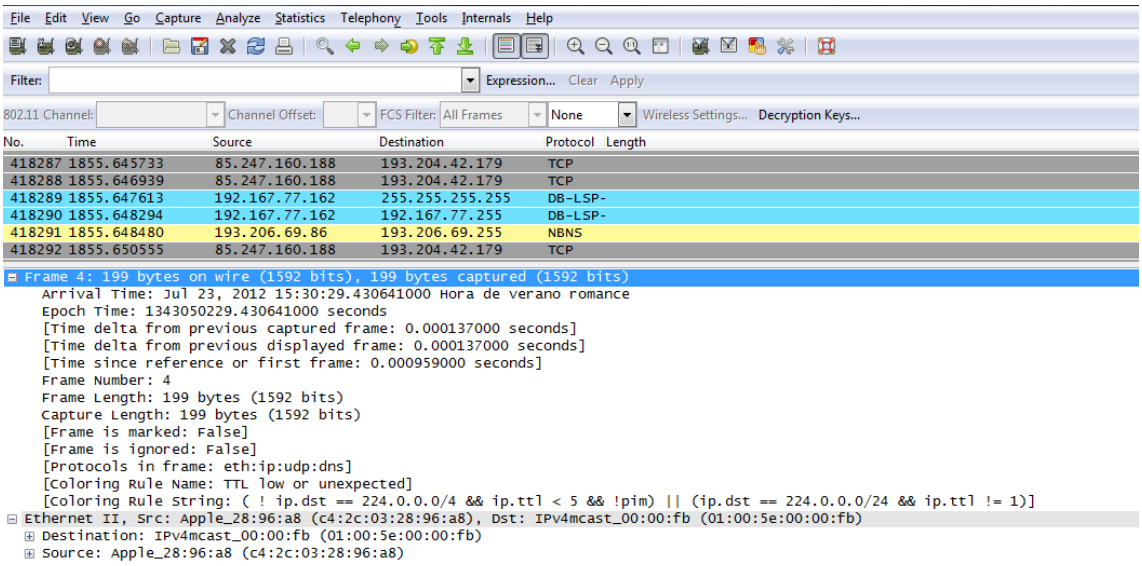


Figura 35: Interfaz Wireshark (1)

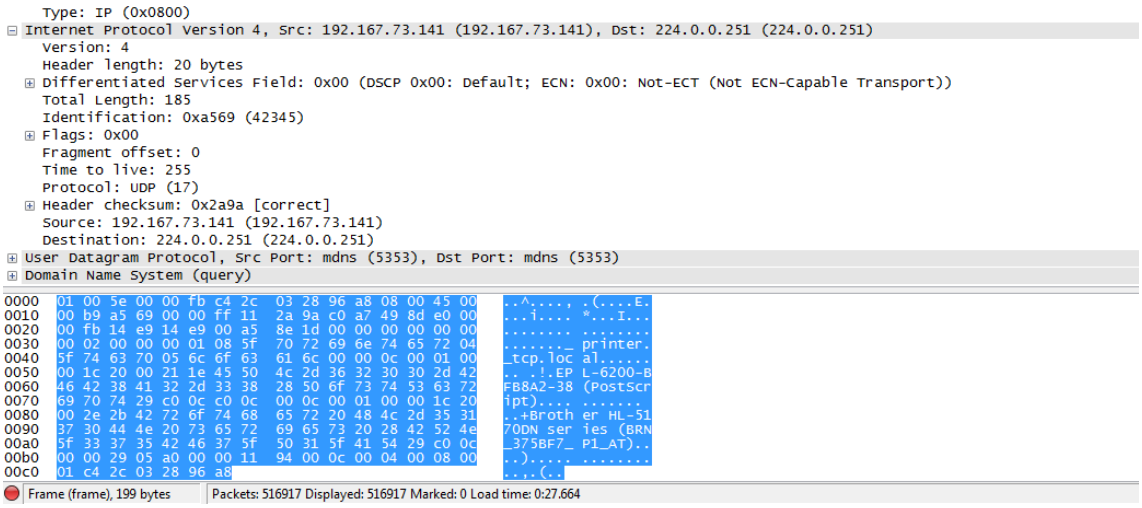


Figura 36: Interfaz Wireshark (2)

Wireshark se divide en tres zonas de datos.

- La primera es la lista de paquetes capturados con información de numero de frame, tiempo en segundos de la captura, origen, destino, protocolo e información adicional.
- La segunda zona muestras los datos del frame elegido (En este caso el 4) siguiendo las capas del modelo TCP/IP: Acceso a red, capa de red, capa de internet y capa de aplicación.

  En primer lugar muestra los datos propios del frame elegido, como hora de captura, tamaño, medidas de tiempo entre paquete y paquete capturado…

  A continuación empieza realmente lo asignado a cada capa. En el primer campo se obtiene los datos correspondientes a la capa de acceso a red con los datos de destino, origen y tipo, como direcciones MAC o los medios físicos que atraviesa.

  Después se encuentran los datos de la capa de internet, que son los más numerosos y de donde más información se puede sacar. Los datos principales de este campo son: Versión del protocolo IP, tamaño de cabecera IP, flags y métodos de corrección de errores.

  El tercer campo es el correspondiente a la capa de transporte, con los datos propios de las cabeceras UDP o TCP, según el protocolo de transporte que utilice el frame elegido.

  El último campo de esta zona son los datos de la capa de aplicación. Este campo es muy variado, y da diferentes datos según el frame que se haya elegido.

- La tercera zona indica el contenido paquete seleccionado en el panel superior en formato hexadecimal y ASCII.

A continuación se presentan varios ejemplos de capturas de wireshark:

Figura 37: Captura Wireshark

En este ejemplo, Wireshark captura todos los protocolos existentes en una comunicación. Esta captura no es muy útil, por lo que se tendría que filtrar la búsqueda para una mejor interpretación de los datos.

Ejemplo de una captura de DPU (Unidad de datos de protocolo) mediante PING:

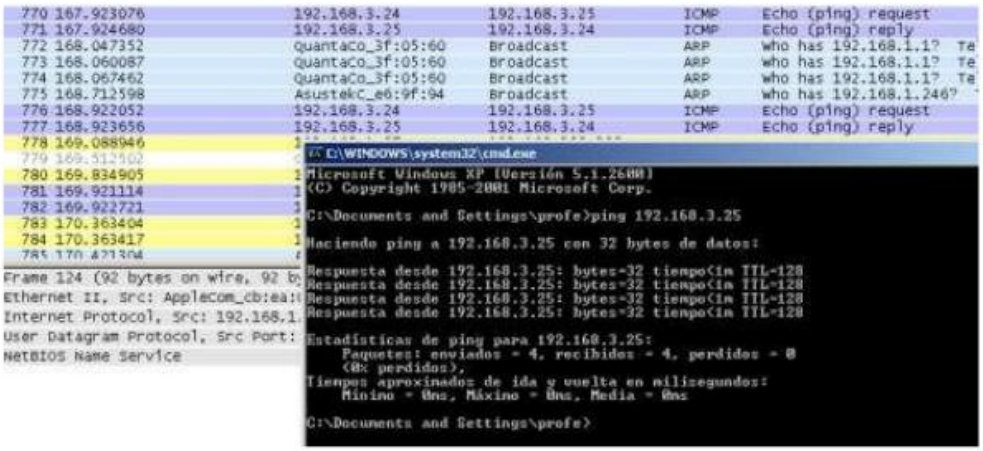


Figura 38: Captura mediante PING

En el ejemplo anterior, Wireshark captura tramas ICMP de un ping realizado desde la maquina 192.168.3.24 hacia 192.168.3.25 y un protocolo ARP de broadcast.

A continuación veamos un ejemplo de una captura de estadísticas de Wireshark.

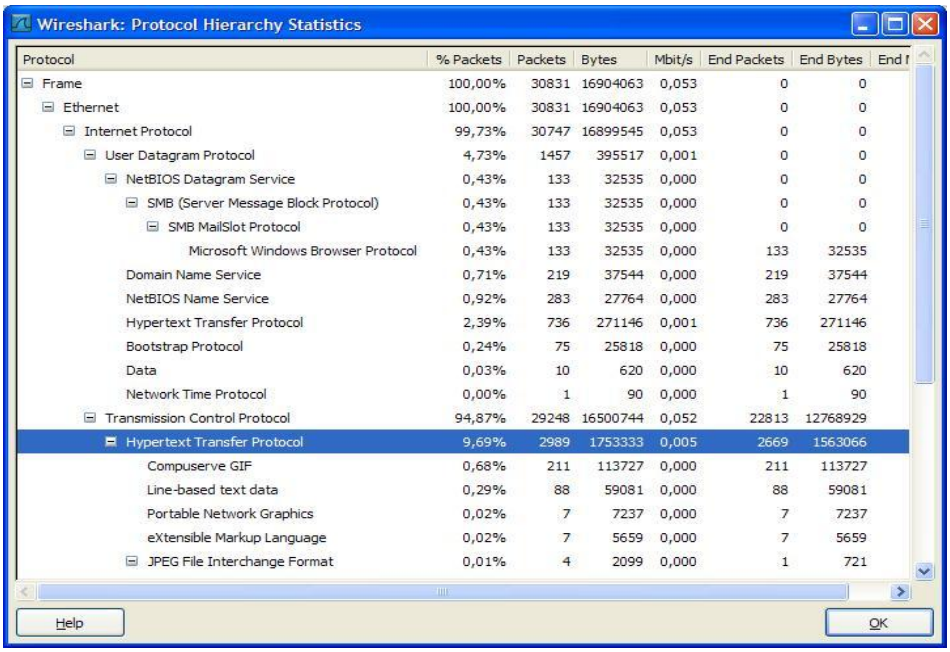


Figura 39: Estadísticas Wireshark

De aquí podemos obtener varios datos, como por ejemplo que el protocolo dominante es el IP con un 99.73%. En el siguiente nivel, el protocolo más utilizado es el TCP con el 94.87%.

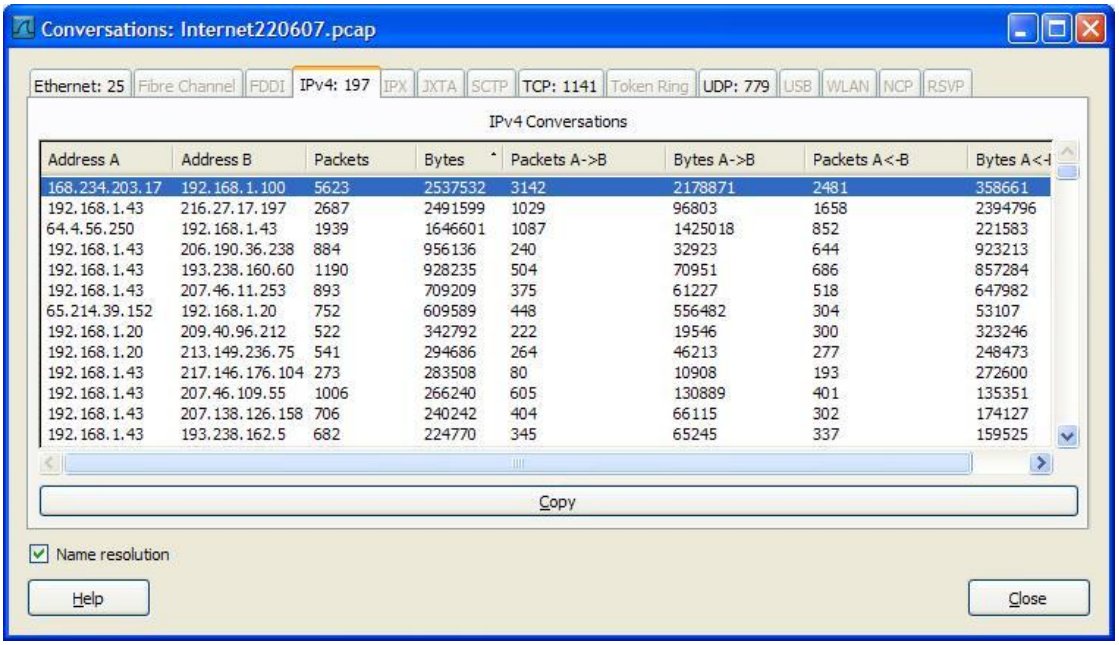


Figura 40: Conversation Wireshark

En este ejemplo vemos que hosts transmiten más cantidad de bytes.

Las opciones de Wireshark son amplísimas, por lo que se sitúa en el mercado como uno de los mejores analizadores de red.

Esta herramienta se puede conseguir tanto para sistemas operativos UNIX como WINDOWS.

3.2 NetworkMiner

NetworkMiner es una herramienta de análisis que intercepta y analiza los paquetes que viajan por la red local. A diferencia de otras herramientas similares, esta herramienta escucha pasivamente la actividad de red, sin generar tráfico.

Puede ser usado como sniffer pasivo/herramienta de captura de paquetes con el objetivo de detectar detalles específicos del host como sistemas operativos, hostname, sesiones, etc. sin generar ningún tráfico en la red.

Portable y fácil de usar, NetworkMiner es una herramienta de análisis segura y con funciones más que suficientes.

3.2.1 Características

Como características importantes se podrían destacar las siguientes:

- Identificación de cualquier sistema operativo.
- Reconstrucción de archivos de otras herramientas como wireshark.
- Extracción de imágenes (también de otras herramientas).
- Identificación de credenciales.
- Útil para el análisis de trafico malware.
- Permite importar archivos PCAP para análisis off-line.
- Búsqueda de datos esnifados.

Esta herramienta se combina perfectamente con wireshark y permite obtener más información sobre el host detectado. Por ejemplo, mientras Wireshark informa sobre la dirección IP, esta herramienta, permite ver también que tipo de sistema se está ejecutando, cual es el puerto por el que se escucha o el servidor web que se utiliza, entre otras cosas.

Para identificar el sistema operativo se basa en TCP SYN y SYN+ACK haciendo uso de la base de datos. También puede realizar fingerprinting para el escaneo de puertos UDP y TCP.

3.2.2 Interfaz

Al pulsar F5, NetworkMiner comenzará a registrar el tráfico, mientras que con F8 parará la operación de captura. Organizado en pestañas, se acumularán los datos obtenidos.

De todas las pestañas, la principal es host. En ella, NetworkMiner muestra los equipos detectados usando un arbol jerárquico desplegable.

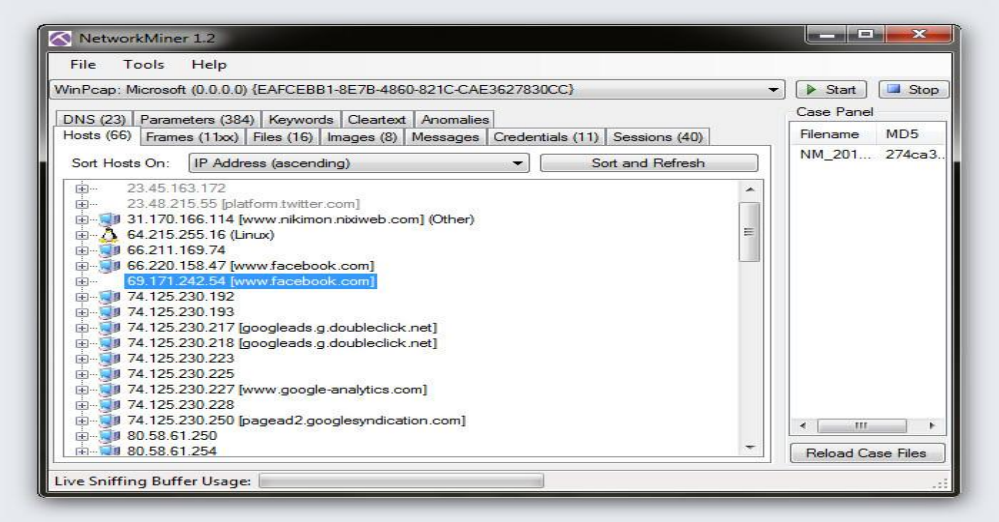


Figura 41: Interfaz NetworkMiner

Si se pulsa sobre cualquier host, sale un menu despegable con todos los datos referidos al host elegido, como se muestra en la siguiente imagen.

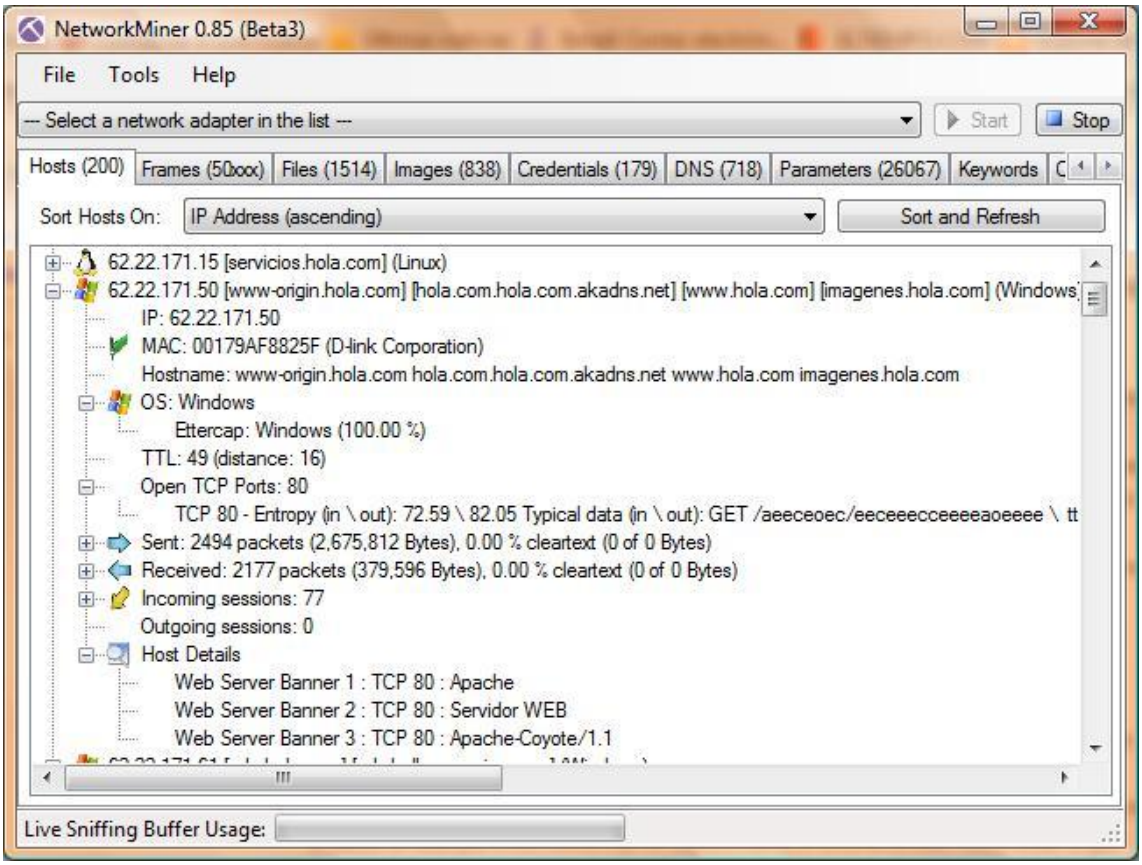


Figura 42: Ejemplo NetworkMiner

Se encuentra disponible para el sistema operativo Windows y UNIX.

3.3 Spiceworks

Spiceworks es un software gratuito que genera información útil de la LAN de una organización, sacando estadísticas de cuantos dispositivos hay en la red, cuantos PC's, servidores, impresoras, switches... Además, de cada uno saca toda la información posible, que sistema operativo tiene, que software/parches tiene instalado, su configuración, datos físicos...

Ofrece una opción interesante sobre alertas. Se pueden generar alertas y que esta herramienta avise cuando hay una, por ejemplo, que no tenga antivirus, espacio en disco, servicios caídos, etc.

De igual manera, Spiceworks incluye diversos datos sobre cada componente analizado en la red, por ejemplo brinda información sobre los SO instalados en cada PC; antivirus; datos de hardware y ajustes de configuración.

3.3.1 Características

Las características principales de esta herramienta son:

- Mapa de red, la característica más important. Spiceworks ofrece un mapa con todas las redes y diferentes dispositivos instalados en la red.
- Inventario de dispositivos.
- Auditoriar software.
- Cambia la configuración de la red.
- Monitorizar el servidor SQL.
- Clasificar y borrar dispositivos.

3.3.2 Interfaz

En la pagina principal de Spiceworks se ofrece la cantidad total de equipos conectados a la red. La siguiente imagen identifica una red con todos los equipos que hay en ella, como servidores, impresoras, PC's, switchs…

Figura 43: Interfaz SpiceWorks

Otra opción es la del mapeo de la red. Las siguientes imágenes muestran un ejemplo de ello.

Figura 44: Ejemplo SpiceWorks (1)

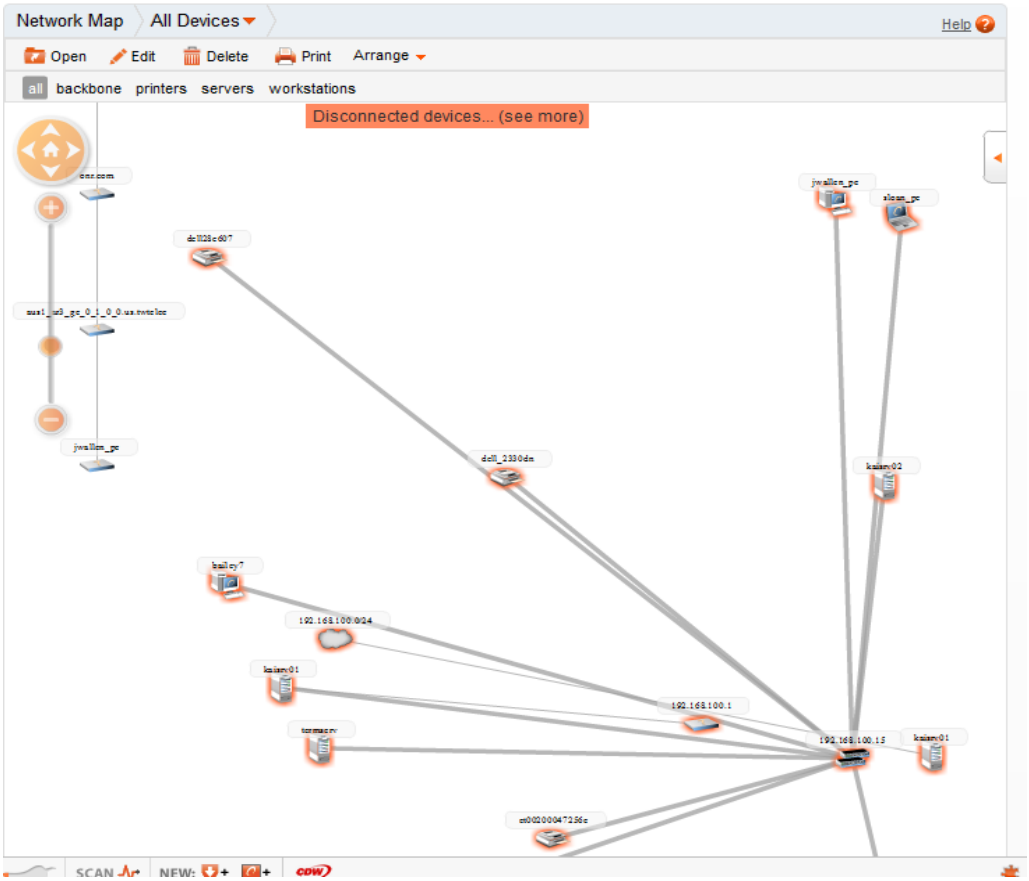


Figura 45: Ejemplo SpiceWorks (2)

Esta herramienta está enfocada, sobre todo, a la organización de una red, por lo que no es la más indicada para realizar este proyecto.

3.4 Tcpdump/Windump

Tcpdump es otra herramienta para analizar el tráfico de red open source.

En un principio fue ideada para sistemas UNIX pero tiempo después salió su correspondiente versión en Windows con el nombre de windump.

Esta herramienta no se basa en un entorno grafico sino en la línea de comandos de la terminal de UNIX.

Su principal función es analizar el tráfico que circula por la red y permitir al usuario capturar y mostrar en tiempo real los paquetes transmitidos y recibidos en nuestra red.

Ofrece la posibilidad de añadirle unos filtros para depurar la salida y así obtener directamente lo que estamos buscando, en caso de no escribir ningún filtro a continuación de la orden, tcpdump volcaría toda la información del adaptador de red seleccionado.

3.4.1 Características

Los principales usos de tcpdump son:

- Depurar las aplicaciones que utilizan la red para comunicar.
- Depurar la propia red.
- Leer y capturar los datos enviados por otros usuarios si el sistema por el que se están enviado no están cifrados, como pudiera ocurrir en telnet o en algunos casos de HTTP.

3.4.2 Parámetros

Tcpdump puede utilizar los siguientes parámetros:

- -A: Imprime cada paquete en código ASCII.
- -D: Imprime la lista de interfaces disponibles.
- -n: No convierte las direcciones de salida.
- -p: No utiliza la interfaz especificada en modo promiscuo.
- -t: No imprime la hora de captura de cada trama.
- -x: Imprime cada paquete en hexadecimal.
- -X: Imprime cada paquete en hexadecimal y código ASCII.
- -c count: Cierra el programa tras recibir 'count' paquetes.
- -C file_size.
- -E algo: secret.

- -F file
- -i interface: Escucha en la interfaz especificada.
- -M secret
- -r file
- -s snaplen
- -T type
- -w file: Guarda la salida en el archivo 'file'.
- -W filecount
- -y datalinktype
- -Z user

Además puede utilizar estos filtros para obtener adecuadamente la información que queremos:

- type [host|net|port]: Máquina en particular [host], red completa [net] o puerto concreto [port].
- dir [src|dst|src or dst|src and dst]: Especifica desde [src] o hacia dónde [dst] se dirige la información.
- proto [tcp|udp|ip|ether]: Protocolo que queremos capturar.

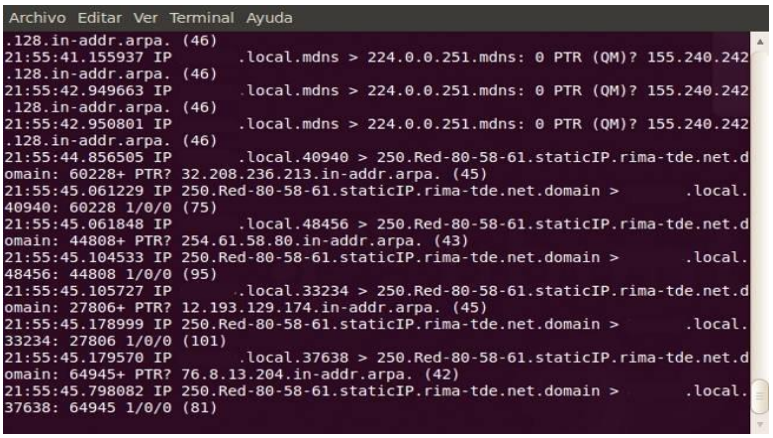


Figura 46: Ejemplo Tcpdump

3.5 Netflow Analyzer

NetFlow Analyzer es una herramienta de monitorización de ancho de banda a la vez que una completa herramienta de análisis de tráfico de red, que aprovecha las tecnologías de flujo para proporcionar visibilidad en tiempo real del rendimiento del ancho de banda de red.

Esta herramienta permite conocer profundamente el rendimiento de la red.

3.5.1 Características

- Los administradores de la red (usuarios) pueden ver qué aplicaciones se están usando en la red.
- Se conoce el ancho de banda consumido por cada aplicación.
- Asignación de aplicaciones.
- Agrupación de aplicaciones.
- Capacidad de asignar aplicaciones en función del puerto, del protocolo y de la dirección IP.
- La distribución de protocolos puede verse fácilmente.

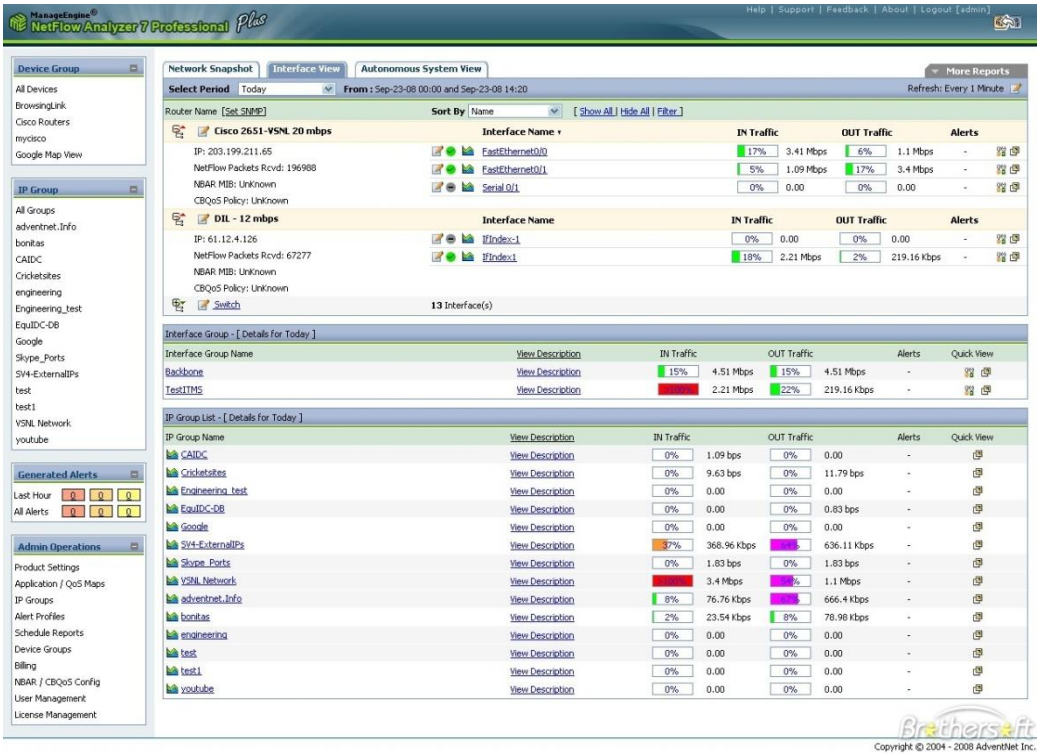


Figura 47: Ejemplo Netflow Analizer

3.6 Conclusión

Tras analizar las ofertas del mercado sobre herramientas de monitorización y análisis de tráfico y ver sus características, he elegido la herramienta Wireshark para realizar las pruebas de este proyecto, ya que sobre la que hay más información y además, dispongo de un conocimiento previo sobre Wireshark al utilizarla en varias asignaturas durante la carrera.

Capítulo 4: Capturas y pruebas

4.1 Introducción

En este capítulo se especifica la forma de captura de datos en la herramienta Wireshark.

Tras la elección de Wireshark para realizar las pruebas de captura, el siguiente objetivo es elegir la franja horaria y la duración de las capturas para un análisis correcto de las mismas.

En un primer momento se realizaron pruebas de captura a distinta horas del día y de distinta duración, desde las más básicas de 10 minutos, hasta pruebas de 3 horas. La cantidad de bytes recibidos para sesiones muy prolongadas eran altísimas y se producían dificultades para analizar el tráfico, el procesado de los datos se volvía muy tedioso y el ordenador no podía soportar tanto esfuerzo, por lo que se decide que las sesiones de captura sean de 40 minutos.

Tras hacer varias pruebas en distintos momentos del día, de 9.00 de la mañana a 12.00, y de 14.30 a 19.00 de la tarde, se decide que las pruebas se han de realizar entre las 15.00 y las 17.00 horas (aproximadamente) de cada día y con descansos entre medias de 2 semanas para ver cómo evolucionan y difieren los datos de unas semanas a otras. Estas diferencias se notan sobretodo en la última semana debido al menor consumo de internet.

4.2 Procedimiento de las capturas de las pruebas

A continuación se enumeran los pasos realizados para obtener las capturas en wireshark.

Lo primero que hay que hacer es obtener el paquete wireshark descargándolo desde la propia web de la compañía http://www.wireshark.org/download.html e instalarlo.

Una vez instalado, se abre el programa y se elige la opción *Capture Options*. Aparece una ventana que indica diferentes opciones que se pueden elegir para la captura de paquetes. En primer lugar se elige la interfaz por la que capturar los paquetes, en el caso de este proyecto se ha elegido la interfaz de red perteneciente al laboratorio en el que estaba realizando las pruebas, siempre el mismo (laboratorio di evalutazione delle prestazione de la Universidad de Pavia). También se marca la opción *capture packets in promiscuous mode* para que se capturen todos los paquetes que circulen por la red, tanto los recibidos como los enviados. En las primeras pruebas no se puso ningún filtro en el campo *capture filter*, pero tras la segunda semana se introdujo el filtro ip para capturar solo los paquetes pertenecientes a IP, que son los que posteriormente se analizarán. Para las primeras pruebas se introdujo posteriormente ese filtro en el proceso de análisis en lugar de en el de captura. Todas las demás opciones quedan marcadas para obtener la información más abundante posible menos la opción *enable network name resolution* dado que no aportaba nada remarcable al análisis. Por último se especifica el tiempo de

captura en la opción *Stop capture after…* dándole un valor de 40 minutos, que es el tiempo elegido de duración óptima de las capturas.

Tras elegir todas las opciones indicadas se pulsa el botón *Start* para dar comienzo a la captura de los paquetes.

Una vez acabados los 40 minutos se pulsa el icono guardar, o bien la opción *Save* de *File*, y se guarda la captura realizada en la carpeta elegida. En este proyecto se han guardado las capturas diferenciándolas entre días y semanas.

Este proceso se repite para realizar cualquier captura para después ser analizado correctamente y tener siempre unas capturas que coincidan tanto en duración como en igualdad de hora.

4.3 Procedimiento de medidas

Las medidas obtenidas en este proyecto son muy variadas, y son todas relacionadas con los diferentes datos que da Wireshark una vez obtenida una captura.

Las primeras medidas obtenidas son las propias de Wireshark, es decir, las que da la herramienta directamente, y a partir de ellas, se obtendrán las demás aportando gráficas y esquemas para un mejor entendimiento.

Para obtener las medidas se carga en Wireshark la captura elegida mediante la opción *File Open*. A partir de aquí se pueden elegir diferentes opciones para obtener los datos propios de la captura seleccionada. A continuación se ofrecen las opciones utilizados en este proyecto para el análisis de datos.

En primer lugar se elige la opción *Summary* del menú *Statistics*, esta opción ofrece datos como los de fecha de captura, formato, tamaño del archivo… y otros de mayor importancia referidos al tráfico de red como los siguientes:

- Número de paquetes
- Media de paquetes por segundo
- Media de bytes por segundo
- Media de megabytes por segundo
- Media del tamaño de paquetes
- Bytes totales

Traffic	Captured	Displayed	Marked
Packets	516917	516917	0
Between first and last packet	2399,282 sec		
Avg. packets/sec	215,447		
Avg. packet size	252,226 bytes		
Bytes	130379926		
Avg. bytes/sec	54341,224		
Avg. MBit/sec	0,435		

Figura 48: Summary

De estos datos tendremos especial interés en obtener la cantidad total de paquetes y la cantidad total de bytes para su posterior análisis en días sucesivos y semanas en forma gráfica.

A continuación se elige la opción *Protocol Hierarchy* del menú *Statistics*, esta opción permite obtener el número de paquetes y bytes, y el porcentaje de paquetes y bytes pertenecientes a cada protocolo. Esta opción es de gran ayuda para obtener posteriormente el gráfico sobre la cantidad total de paquetes de la capa de transporte que utilizan el protocolo TCP, UDP u otros menos reseñables.

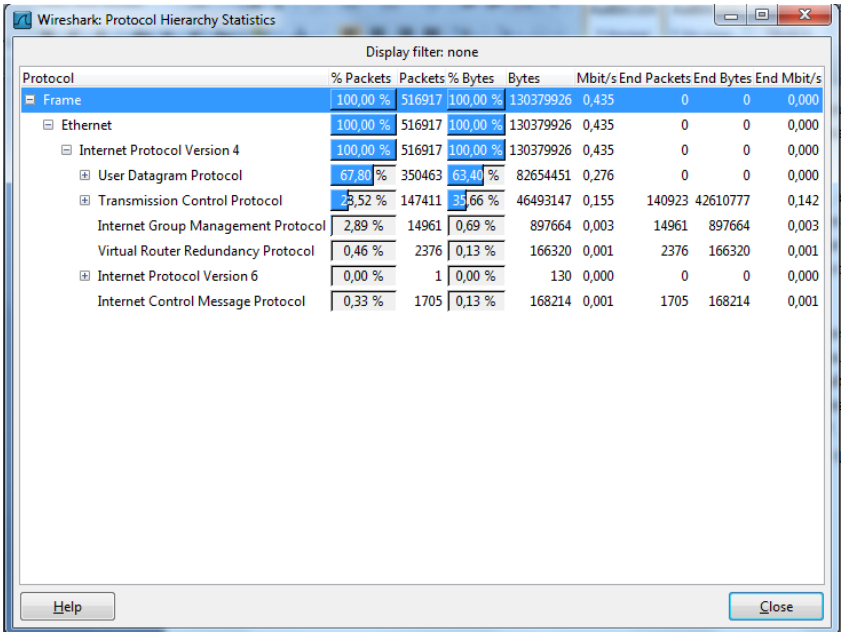


Figura 49: Protocol Hierarchy

Otra opción utilizada para obtener datos es la opción *Conversations* del menú *Statistics*. Esta es la opción que reportará mayores datos para ser analizados de forma gráfica, ya que ofrece todo tipo de información sobre las distintas comunicaciones habidas durante la captura a los diferentes niveles del modelo TCP/IP y sobre los diferentes protocolos.

De todas las pestañas de *Conversations* (Ethernet, IPv4, IPv6, TCP, UDP…) la que se analizará detalladamente en este proyecto es la pestaña IPv4, ya que es la que ofrece toda la información relacionada con las IP´s que se intercambian datos.

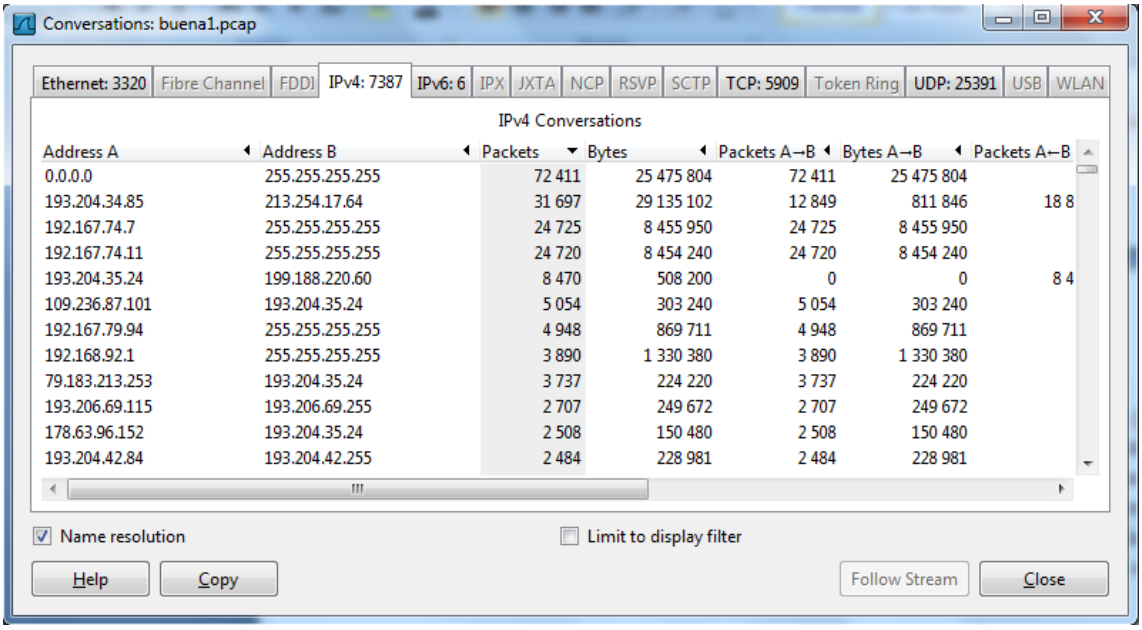


Figura 50: Conversations

Como se ve en la figura anterior esta pestaña se puede ordenar según el número de paquetes o bytes totales, desde las IP´s que más o menos datos reciben. Este dato se tendrá en cuenta en un procedimiento posterior para obtener las IP´s que más tráfico reciben, pero no se modificarán aquí las tablas, si no en un Excel que se utilizará posteriormente para que sea más fácil su modificación.

Es importante no marcar la opción *Limit to display filter* para que solo se tengan en cuenta los paquetes capturados mediante el filtro ip comentado anteriormente.

Es de gran utilidad en *Conversations* la opción *Copy*, ya que permite copiar todos los datos sobre direcciones, paquetes, bytes, destino de los datos… en un archivo .txt separados por comas y entrecomillados para poder trabajar en otros programas de procesado de datos.

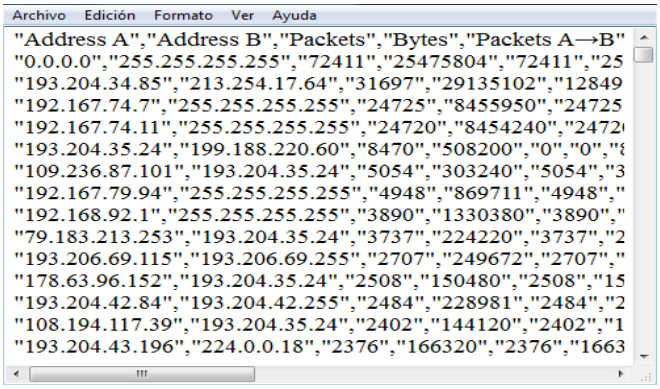


Figura 51: Ejemplo archivo.txt

Otra opción que ofrece Wireshark es *IO Graphs* del menú *Statistics*, que ofrece gráficas sobre los paquetes pudiendo aplicar todos los filtros necesarios para ver gráficas de algún protocolo en especial. Es una opción gráfica muy visual pero en este proyecto se crearán otras gráficas aparte y no será necesario utilizar esta opción.

Las 3 opciones detalladas anteriormente son de donde se obtendrán la mayoría de datos.

Para el proceso de análisis, se necesitan tener todos estos datos organizados de forma simple para poder utilizarlos para crear gráficas de control y evolución de los datos de forma diaria, semanal y total. Por ello se guardarán en distintas tablas los datos obtenidos en las opciones anteriores.

De la opción *Summary* se guardarán los datos proporcionados en una tabla para la posterior evaluación de los resultados.

De *Protocol Hierachy* se anotarán en una tabla Excel los resultados exactos de cada porcentaje de paquetes correspondientes a cada protocolo para crear una gráfica sobre la utilización de los protocolos de transportes TCP y UDP principalmente.

De *Conversations* se elegirá, como anteriormente se ha comentado, la pestaña IPv4, y de esta pestaña se copiaran los datos y se guardarán en un archivo .txt con el nombre de cada sesión. Tras copiarlos, estos datos se cargarán en una tabla Excel diferenciándolos en celdas para poder ordenar de mayor a menor el tráfico recibido y ver cuáles son las IP´s con el tráfico mayor tanto en bytes como en paquetes. También es de utilidad ver cuántas IP´s han enviado un solo paquete. Para tener ordenados estos datos seleccionamos la columna que queramos evaluar, por ejemplo, paquetes totales, y elegimos la opción ordenar de mayor a menor para obtener las IPs con más tráfico, también podríamos utilizar un filtro en el caso de ver las IPs que envían un solo paquete eligiendo la opción *Filtrar* del menú *Ordenar y filtrar* marcando solo el filtro de número 1. Está opción de filtros es muy útil también para crear la gráfica de evolución del número de paquetes que se intercambian dos IP´s utilizando también otros filtros del menú anterior denominados *mayor o igual que*, *menor o igual que* o *entre*.

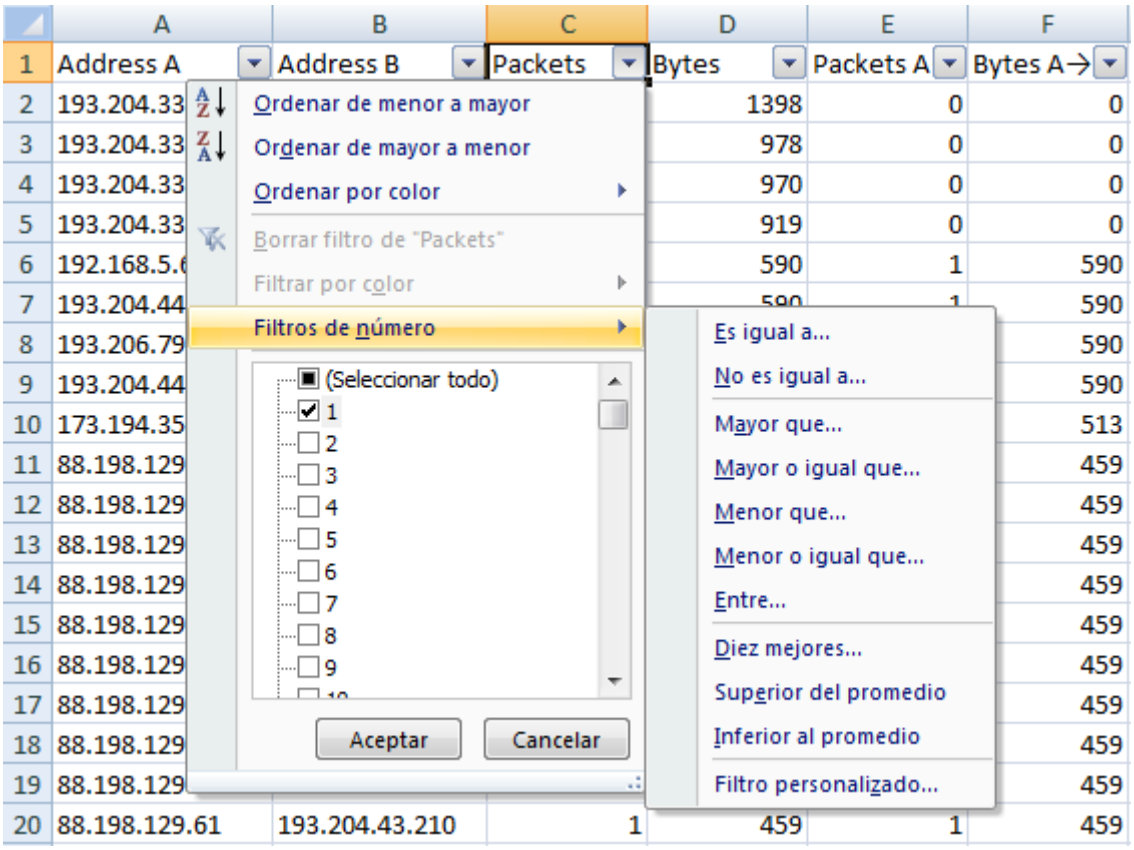


Figura 52: Ejemplo Excel

Para todas las gráficas y tablas se ha decidido usar el programa Excel, ya que ofrece las suficientes herramientas y características para poder realizar todas las gráficas planteadas en este proyecto.

Capítulo 5: Análisis de resultados

5.1 Introducción

En este capítulo se detallarán los resultados de las medidas realizadas mediante los métodos explicados en el capítulo anterior.

Se puede diferenciar los resultados en tres bloques. En el primero se estudia el tráfico global generado durante cada día, diferenciándolo entre bytes y paquetes, y ofreciendo tablas resúmenes sobre dichos datos. En el segundo bloque se obtienen gráficas sobre el tráfico generado por y para cada IP, y otras estadísticas como IP´s con un solo acceso. Por último, en el tercer bloque, se ofrecen los gráficos sobre el conjunto de datos que ofrece un mismo protocolo, con especial interés en los protocolos de la capa de transporte TCP y UDP.

Ante el abundante número de gráficas y tablas, en este capítulo se presentarán las principales y las gráficas globales para obtener la conclusión final. El resto de gráficas y tablas se encuentran en el anexo final de la memoria ordenadas por días y semanas.

5.2 Escenario

Las pruebas se realizaron en el laboratorio di evalutazione delle prestazione(laboratorio de evaluación de las prestaciones) de la universidad de Pavia.

De la red de este laboratorio se han obtenidos la totalidad de las pruebas y siempre desde y hacia el mismo host donde estaba instalado el software.

Destacar que todas las capturas se hicieron alrededor de la misma hora después de realizar multitud de pruebas en diferentes horarios. Tras analizar las diferentes capturas de prueba realizadas durante unas semanas en periodos de mañana y tarde, se elige, como se ha dicho anteriormente, la franja para capturar tráfico entre las 15.00 y las 17.00 horas para capturar tráfico. Estas capturas tienen una duración de 40 minutos y se realizaron en diferentes periodos del año 2012 entre los meses de Marzo y Julio.

5.3 Resultados

La primera medida en la cual obtenemos un resultado a destacar es la que nos ofrecía en un principio la opción *summary* de Wireshark. Este dato lo presentamos tanto por días, como por semanas.

SEMANA 1	Día 1	Día 2	Día 3	Día 4	Día 5	Total
Número de paquetes	649501	705641	638674	650461	637185	656292,4
Media Paquetes/seg	270,625417	294,017083	266,114167	271,025417	265,49375	273,4551667
Media tamaño (bytes)	794,721	141,143	479,942	150,812	599,956	433,3148
Número de bytes	383531355	413215586	373843535	388702063	356795425	383217592,8
Media Bytes/seg	159804,731	172173,161	155768,14	161959,193	148664,76	159673,997
Media Mbits/seg	1,248	1,345	1,217	1,265	1,161	1,247

SEMANA 2	Día 1	Día 2	Día 3	Día 4	Día 5	Total
Número de paquetes	670546	654128	701542	666741	632458	665083
Media Paquetes/seg	279,394167	272,553333	292,309167	277,80875	263,524167	277,1179167
Media tamaño (bytes)	564,211	357,821	754,942	203,571	502,184	476,5458
Número de bytes	400531204	390541236	425641223	390314562	362145874	393834819,8
Media Bytes/seg	166888,002	162725,515	177350,51	162631,068	150894,114	164097,8416
Media Mbits/seg	1,304	1,271	1,386	1,271	1,179	1,282

SEMANA 3	Día 1	Día 2	Día 3	Día 4	Día 5	Total
Número de paquetes	705483	685432	625426	701283	640561	671637
Media Paquetes/seg	293,95125	285,596667	260,594167	292,20125	266,900417	279,84875
Media tamaño (bytes)	354,265	726,821	405,192	451,268	236,952	434,8996
Número de bytes	415203679	394715283	350284162	405678154	365478016	386271858,8
Media Bytes/seg	173001,533	164464,701	145951,734	169032,564	152282,507	160946,6078
Media Mbits/seg	1,352	1,285	1,140	1,321	1,190	1,257

SEMANA 4	Día 1	Día 2	Día 3	Día 4	Día 5	Total
Número de paquetes	357845	306521	402354	405360	342365	362889
Media Paquetes/seg	149,102083	127,717083	167,6475	168,9	142,652083	151,20375
Media tamaño (bytes)	595,721	195,254	359,462	261,825	601,426	402,7376
Número de bytes	201367821	185437955	225418312	230001269	192038465	206852764,4
Media Bytes/seg	83903,2588	77265,8146	93924,2967	95833,8621	80016,0271	86188,65183
Media Mbits/seg	0,655	0,604	0,734	0,749	0,625	0,673

Figura 53: Tablas datos principales

Para las 3 primeras semanas vemos unos valores que podemos denominar normales, ya que estos datos son similares cada día, con la variación típica existente entre distintos días debido a que cada día no circula por la red el mismo tráfico.

La cantidad total de paquetes se mantiene constante entre los valores de 600000 y 700000 paquetes aproximadamente y la cantidad de bytes también se mueve entre los valores de 350000000 y 400000000 bytes.

También las medias de paquetes/seg y bytes/seg se mantienen constantes y se mantienen alrededor de 275 paquetes/seg y 160000 bytes/seg como se indican en los valores finales de media calculados a partir de las muestras tomadas cada día.

El valor diferencial lo encontramos en la semana número 4, ya que desciende de forma drástica los valores en todos los campos medidos, llegando a capturarse la mitad de paquetes y bytes y su consiguiente bajada de media en las velocidades. Este hecho se debe a que las últimas capturas están realizadas a finales del mes de Julio, donde el consumo de tráfico de red es mucho menor que las semanas anteriores. Este dato es de relevancia y puede servir en un futuro para adecuar la velocidad de la red para un menor consumo.

Hay que resaltar los datos obtenidos sobre la media del tamaño de los paquetes, medidas obtenidas en bytes, que parecen no tener ninguna relación entre ellas, ya que cada día hay una media diferente pudiendo ser un día hasta cuatro veces más grande esta media de tamaño que el día sucesivo. Este hecho es debido a que cada día la red recibe unos paquetes diferentes, y de distintos protocolos, por lo que es lógico que cada día se obtenga una media de tamaño distinta.

Con todos estos datos podemos obtener la siguiente estadística global donde se ofrecen los datos de las 4 semanas en una única tabla.

GLOBAL	Semana 1	Semana 2	Semana 3	Semana 4
Número de paquetes	656292,4	665083	671637	362889
Media Paquetes/seg	273,455167	277,117917	279,84875	151,20375
Media tamaño (bytes)	433,3148	476,5458	434,8996	402,7376
Número de bytes	383217593	393834820	386271859	206852764
Media Bytes/seg	159673,997	164097,842	160946,608	86188,6518
Media Mbits/seg	1,2474531	1,28201439	1,25739537	0,67334884

Figura 54: Tabla global datos principales

En esta tabla-resumen vemos que el número de paquetes va aumentando, pero es tan mínimo el número que podemos decir que se mantiene constante entre 655000 y 670000 paquetes, salvando, como se ha dicho anteriormente, la última semana debido al menor tráfico de datos durante el mes de verano.

Estos datos analizados anteriormente corresponden a un entorno global donde no se tenía en cuenta si una IP generaba mucho o poco tráfico, o el número de IP´s que absorben la mayor parte de los datos. A continuación se ofrecen unas gráficas y tablas explicando varios de estos hechos para un mejor entendimiento.

En primer lugar vamos a mostrar la cantidad de bytes que transmite cada IP. Como el número de IP´s que intervienen en todo el proceso de captura de datos es altísimo, llegando en algunas capturas a superar las 15000, se han elegido las 15 direcciones IP que más tráfico reciben para poder analizar gráficamente lo trabajado de una forma más clara, ya que el resto de IP´s podríamos decir que reciben una cantidad de datos residual respecto a las elegidas.

Exponer los resultados de todas las gráficas sería muy repetitivo, ya que se comportan prácticamente en todos los casos siguiendo un mismo patrón. Así que se mostrará un caso en concreto que sirve para entender bien el fin de estas gráficas. El resto de gráficas se encuentran en el anexo para su consulta si es necesario.

La captura elegida para explicar este punto es la de la sesión 3 de la semana 1, ya que sus resultados son similares a la media calculada, por lo que es una buena representación del resto de días.

Por temas de confidencialidad de la Universidad de Pavia, no se puede poner el nombre/número de cada dirección IP, así que lo he sustituido por el lugar que ocupa cada una de ellas con respecto a la cantidad total de bytes.

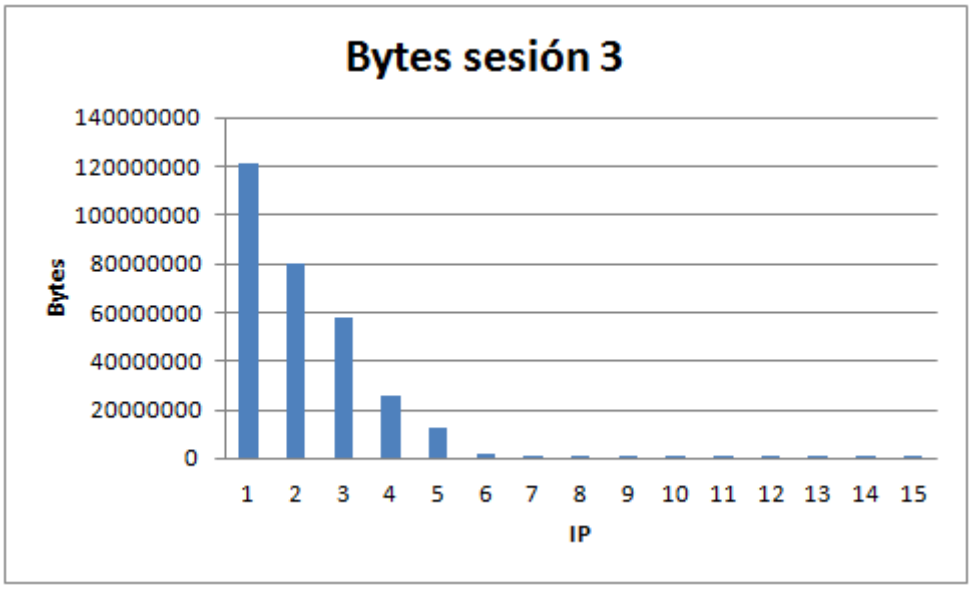


Figura 55: Bytes/IP sesión 3

Esta es la cantidad total de bytes recibidos durante el tiempo de captura para las 15 direcciones IP que más tráfico han recibido. Como se puede comprobar la mayoría del tráfico está comprendido en las primeras IP´s, siendo el tráfico de las 15 IP´s un 82.43% del total, o, en números exactos, 308156242 bytes de los 373843535 bytes totales, el resto de bytes por IP lo podemos considerar residual porque son muy pocos los bytes con respecto a las primeras IP´s.

A continuación se expone la misma relación anterior pero esta vez con paquetes en lugar de bytes.

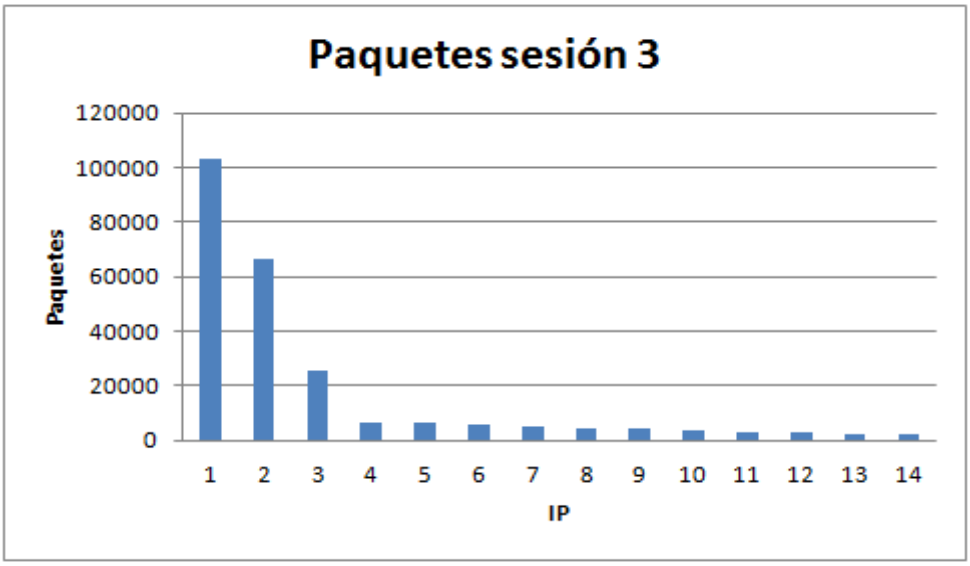


Figura 56: Paquetes/IP sesión 3

En esta gráfica observamos como también gran parte de los paquetes están comprendidos en pocas IP´s. En cambio esta vez la suma de los paquetes de las 15 primeras IP´s son solo un 38.28%, por el 82.43% de bytes, en datos numéricos, 244505 paquetes por 638674 paquetes totales. Este hecho se explica con que cada paquete puede llevar una cantidad de bytes distinta, y en una comunicación continuada entre IP´s los paquetes suelen ser de mayor tamaño debido a que llevan más información, en cambio para una comunicación corta o de un solo accesos suelen predominar los paquetes de consulta o paquetes con menor tamaño.

Como dato a resaltar, coinciden tanto las IP´s donde se reciben más bytes como las IP´s que reciben más paquetes. Este dato es lógico puesto que a mayor número de bytes, mayor número de paquetes, aunque no se cumpla siempre esta condición sí que suele darse.

En las siguientes gráficas y tablas seguimos tratando temas relacionados con el tráfico según la IP, aunque en este caso más genérico y viendo los resultados por días y semanas, y el resultado global para poder realizar comparaciones.

SEMANA 1	IP´s encontradas	IP´s un paquete
1	9622	3228
2	8719	2815
3	12562	4378
4	12078	4126
5	9375	2925
Media	10471,2	3494,4

SEMANA 2	IP´s encontradas	IP´s un paquete
1	10542	3475
2	12580	4326
3	10579	3554
4	13524	4231
5	9946	3215
Media	11434,2	3760,2

SEMANA 3	IP´s encontradas	IP´s un paquete
1	8954	3026
2	9549	3341
3	11943	3905
4	8499	2998
5	10243	3302
Media	9837,6	3314,4

SEMANA 4	IP´s encontradas	IP´s un paquete
1	6591	2115
2	6842	2345
3	6059	1994
4	7214	2451
5	6982	2314
Media	6737,6	2243,8

Figura 57: Tablas IP´s encontradas vs IP´s un paquete

En las tablas anteriores vemos el número de IP´s distintas encontradas en cada sesión y, de éstas IP´s, cuales son las que solo han recibido un paquete.

Como se puede apreciar, el número de IP´s se mantiene constante durante las tres primeras semanas alrededor de las 10000 direcciones, y de nuevo nos encontramos con que en la semana cuatro al reducirse el tráfico de red también se reducen el número de IP´s. En este caso la media baja hasta 6738 IP´s de media.

Pero el dato a destacar de esta tabla son las IP´s donde solo se recibe un paquete, es decir, donde no se recibe contestación por parte de ésta última. Se comprueba que la relación entre las IP´s con un solo acceso y las IP´s encontradas es de alrededor de un tercio, es decir, el número de IP´s encontradas es tres veces mayor aproximadamente que las IP´s con un solo paquete.

MEDIAS	IP´s encontradas	IP´s un paquete	Relación
SEMANA 1	10471,2	3494,4	2,99656593
SEMANA 2	11434,2	3760,2	3,04084889
SEMANA 3	9837,6	3314,4	2,96813903
SEMANA 4	6737,6	2243,8	3,00276317

Figura 58: Tabla medias IP´s encontradas vs IP´s un paquete

Visualmente podemos comprobar este dato en la siguiente gráfica correspondiente a las IP´s encontradas vs IP´s con un solo acceso de la primera semana:

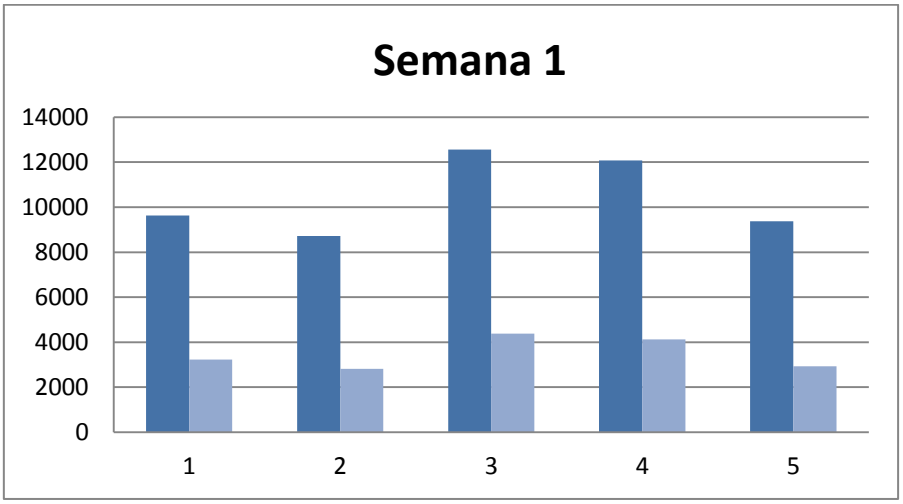


Figura 59: IP´s encontradas vs IP´s un acceso Semana 1

Esta gráfica es la perteneciente a las 4 semanas calculada a partir de sus medias (El resto de gráficas de este tipo se encuentran en el anexo):

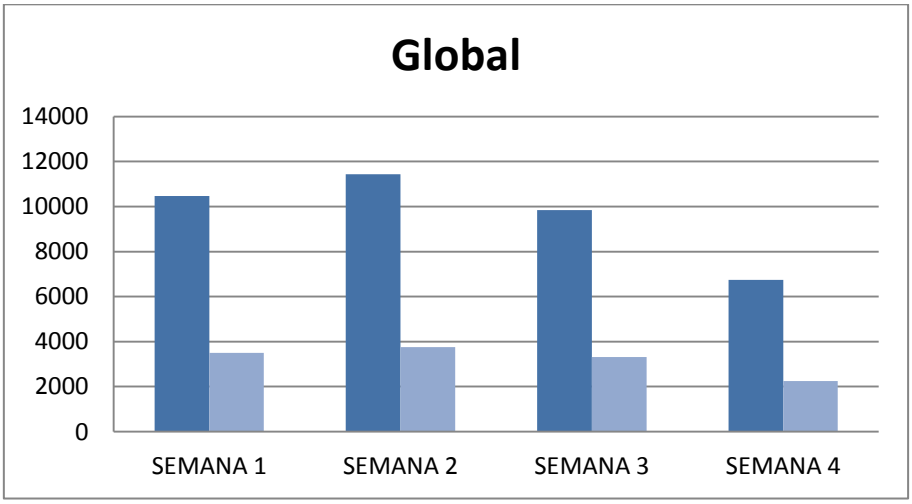


Figura 60: IP´s encontradas vs IP´s un acceso Global

Tras haber contabilizado los bytes y paquetes por IP, y las IP por paquetes, también se puede realizar una gráfica con la cantidad de IP que tienen un determinado rango de paquetes. Este gráfico se ha realizado para cada día y primero se presentará en una sesión concreta y luego en el global de las semanas.

Se vuelve a elegir el día 3 de la semana 1 para que se tenga el estudio de todas las estadísticas de un día concreto.

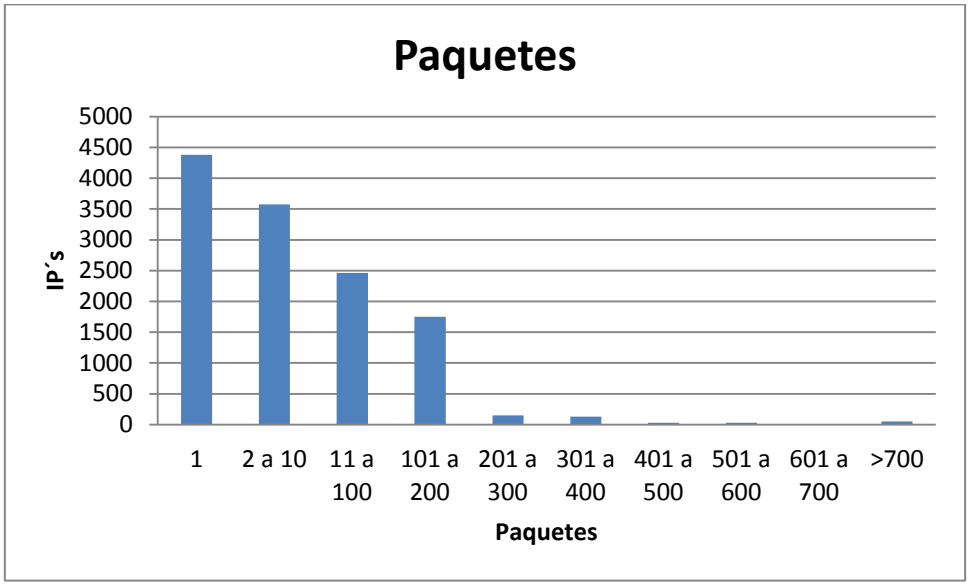


Figura 61: Rango paquetes/IP Sesión 3

Para una mejor visión del problema se ha decidido en un primer momento medir las IP´s con un solo paquetes, las que tienen de 2 a 10, las que tienen de 11 a 100 y a partir de ahí se calcula de nuevo cada 100 paquetes.

Anteriormente hemos calculado las IP´s totales y las de un solo acceso, ahora además calculamos en diferentes rangos cuantas IP´s tienen determinados paquetes.

En el gráfico anterior podemos comprobar que la mayoría de paquetes se concentran en pocas IP´s, siendo la mayoría de un solo acceso, exactamente 4378, que coincide plenamente con el número calculado anteriormente.

Este hecho reafirma lo calculado con anteriores gráficos y se vuelve a demostrar que las comunicaciones extensas están comprendidas en pocas IP´s.

En la siguiente gráfica se presenta la media de las tres primeras semanas, ya que he decidido hacerla entre éstas porque son datos similares, y dejar de lado la semana 4 porque el descenso de paquetes e IP´s es importante como se ha comentado anteriormente.

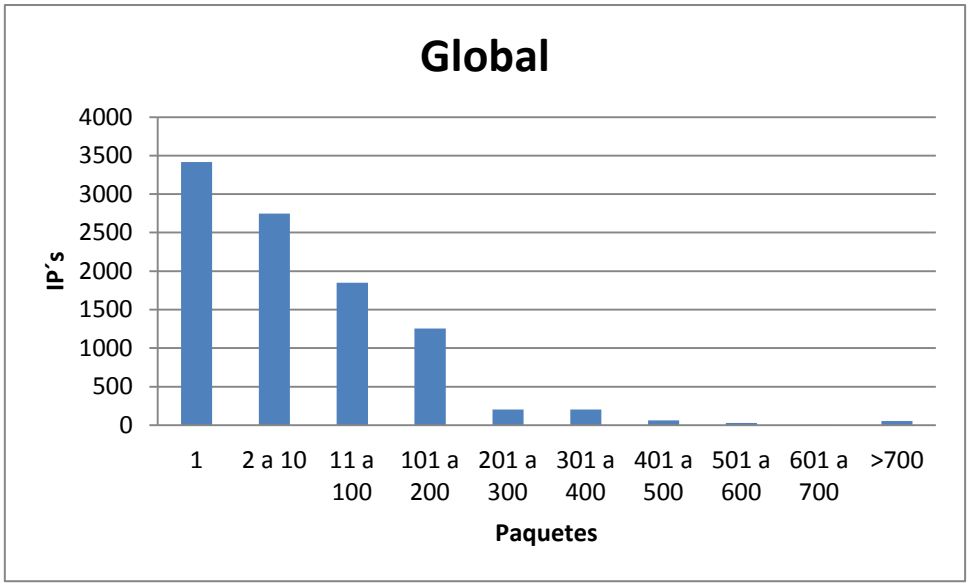


Figura 62: Rango paquetes/IP Global

Como vemos, sigue el mismo modelo que la del día 3, y además, vemos que su comportamiento es muy similar, dado que al ser la media es un resultado bastante fiable.

Para terminar con las explicaciones sobre los estudios realizados dejamos de lado las correspondientes al tráfico recibido por IP y pasamos a ver cuáles son los protocolos predominantes en los paquetes.

En este apartado se ofrece el porcentaje que ocupa en los protocolos de transporte los protocolos TCP y UDP.

Los resultados vuelven a ser presentados en un día concreto y en la media.

En primer lugar se ha elegido el día 3 de la semana 1 para el estudio particular de un día en concreto.

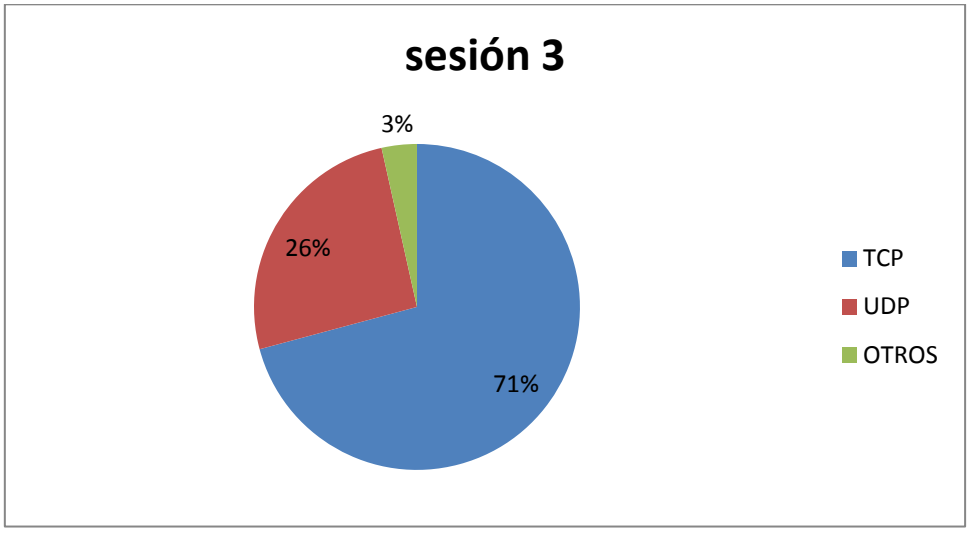


Figura 63: TCP vs UDP sesión 3

En este día la mayoría de los paquetes de transporte utilizaban el protocolo TCP, siendo un 71% de este tipo mientras el 26% son del protocolo UDP, el 3% restante corresponde a protocolos de otro tipo.

En este día se produce lo que la mayoría del resto, que el protocolo TCP se encuentra en cantidades mayores que UDP.

A continuación se presentan las tablas de cada día con respecto al tema que estamos tratando, agrupadas en semanas y calculando también sus medias para poder realizar el gráfico final que resuma la cantidad de paquetes que utilizan TCP, UDP u otro tipo de protocolo.

SEMANA 1	TCP	UDP	OTROS
Día 1	69	27,3	3,7
Día 2	62,47	32,49	5,04
Día 3	70,82	25,69	3,49
Día 4	65,24	32,47	2,29
Día 5	57,32	40,54	2,14
Media	64,97	31,698	3,332

SEMANA 2	TCP	UDP	OTROS
Día 1	75,4	22,2	2,4
Día 2	69,4	29,54	1,06
Día 3	63,41	33,49	3,1
Día 4	67,47	30,51	2,02
Día 5	61,5	36,42	2,08
Media	67,436	30,432	2,132

SEMANA 3	TCP	UDP	OTROS
Día 1	62,45	34,82	2,73
Día 2	67,33	29,1	3,57
Día 3	71,4	25,39	3,21
Día 4	62,51	33,9	3,59
Día 5	59,4	38,5	2,1
Media	64,618	32,342	3,04

SEMANA 4	TCP	UDP	OTROS
Día 1	61,5	33,4	5,1
Día 2	69,19	29,4	1,41
Día 3	68,42	28,1	3,48
Día 4	62,4	35,79	1,81
Día 5	69,82	28,47	1,71
Media	66,266	31,032	2,702

Figura 64: Tablas TCP vs UDP

Como podemos comprobar cada día tiene diferente porcentaje de cada tipo de protocolos, llegando tener un día 10 puntos menos que otro día. Una vez calculadas las medias sí que vemos que los datos son más homogéneos y se mantienen en un rango menor, diferenciándose en un máximo de 3 puntos entre semana y semana.

En este caso vemos como el menor número de paquetes recibidos en la semana 4 no modifica el porcentaje de cada protocolo y sigue comportándose como las semanas anteriores.

Vemos la gráfica de la semana 1 (las demás en el anexo):

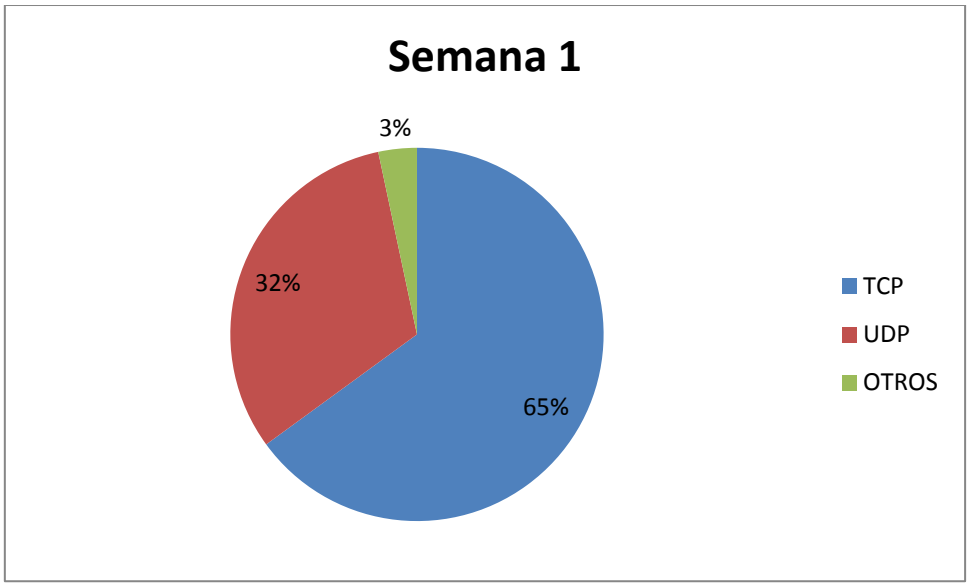


Figura 65: TCP vs UDP Semana 1

La tabla siguiente indica la media de cada semana para obtener una media total.

GLOBAL	TCP	UDP	OTROS
SEMANA 1	64,97	31,698	3,332
SEMANA 2	67,436	30,432	2,132
SEMANA 3	64,618	32,342	3,04
SEMANA 4	66,266	31,032	2,702
Media	65,8225	31,376	2,8015

Figura 66: Tabla TCP vs UDP global

Y para entenderlo todo con más claridad, se ofrece el gráfico de la media total.

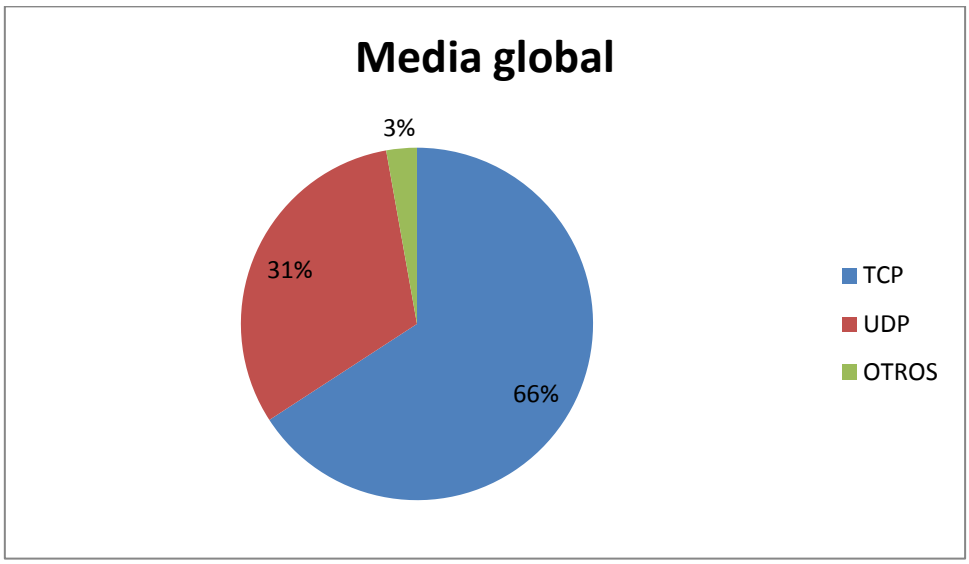


Figura 67: TCP vs UDP global

Del gráfico anterior podemos observar que los paquetes que utilizan TCP es más del doble que UDP, y cualquier otro protocolo que no sea UDP y TCP sigue siendo despreciable en comparación con estos.

Capítulo 6: Conclusiones y trabajos futuros

En este proyecto se ha realizado un estudio de diferentes medidas obtenidas con la herramienta Wireshark de la red de la Universidad de Pavía. Tras realizar el estudio teórico propuesto, se hicieron multitud de pruebas en el laboratorio di evalutazione delle prestazione, y una vez realizadas se decidió hacer pruebas cada día durante una semana (de lunes a viernes), y un total de 4 semanas, para ver como se modifican los resultados entre las distintas semanas.

Gracias a las pruebas realizadas se ha hecho un análisis que ha permitido obtener conclusiones sobre el uso de la red de la Universidad de Pavia y su funcionamiento, sobre todo desde la perspectiva del uso de cada dirección IP y la cantidad de datos que circulan por la red, además de un apartado final sobre el uso de los protocolos TCP y UDP durante el tiempo de captura

Todas las medidas realizadas en este estudio pueden servir para trabajos futuros relacionados con el uso de las diferentes estadísticas creadas, y para obtener una mejora de rendimiento en una red de gran magnitud como la de una universidad.

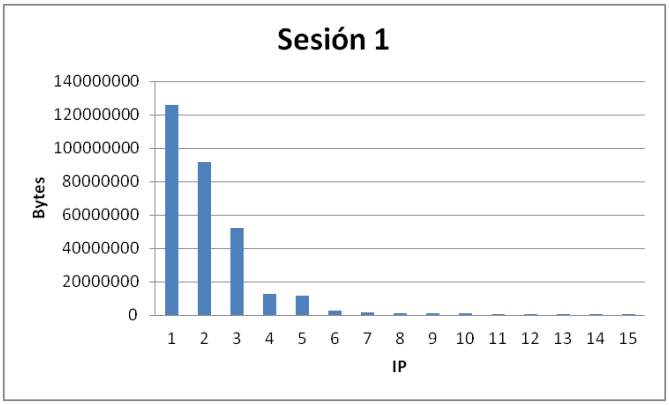
También se puede utilizar el estudio teórico previo para explicar de forma didáctica el funcionamiento de los distintos protocolos y la elección de una herramienta de captura de tráfico según las necesidades previstas.

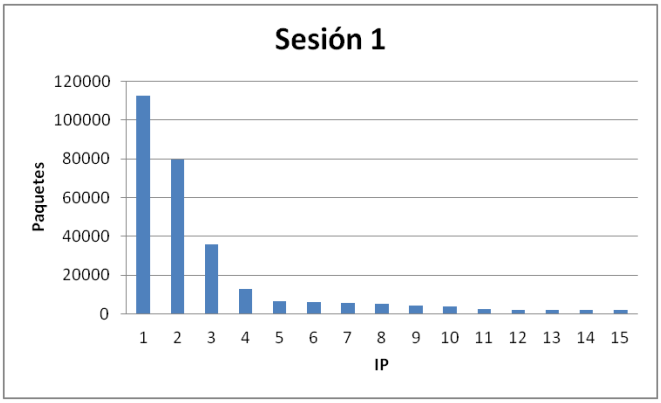
Bibliografía

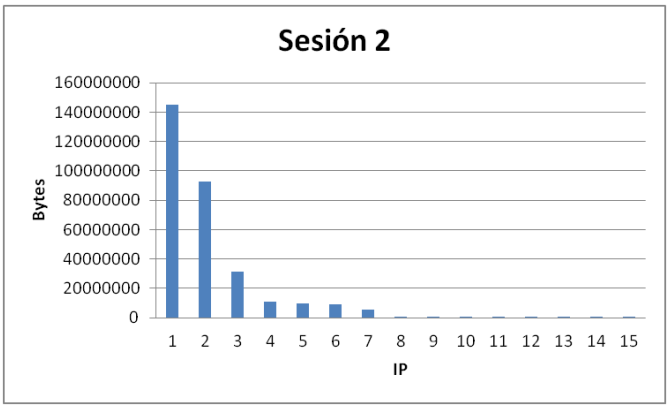
- http://www.wireshark.org Programa utilizado para analizar tráfico
- http://es.wikipedia.org/wiki/Wikipedia Consultas de todo tipo
- RFC de cada uno de los protocolos comentados
- http://dmoz.org/World/Español/Computadoras/Internet/Protocolos
  Selección de páginas web en castellano que tratan sobre los protocolos de internet
- Capítulo 5 de "Redes de Computadores: Un enfoque descendente basdado en Internet". James F. Kurose, Keith W. Ross. Addison Wesley, 2ª edición. 2003.
- Capítulos 4 y 5 de "TCP/IP Illustrated, Volume 1: The Protocols", W. Richard Stevens, Addison Wesley, 1994.

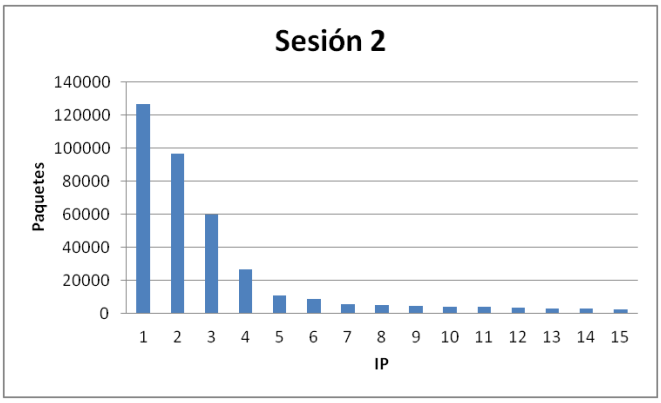
Anexo

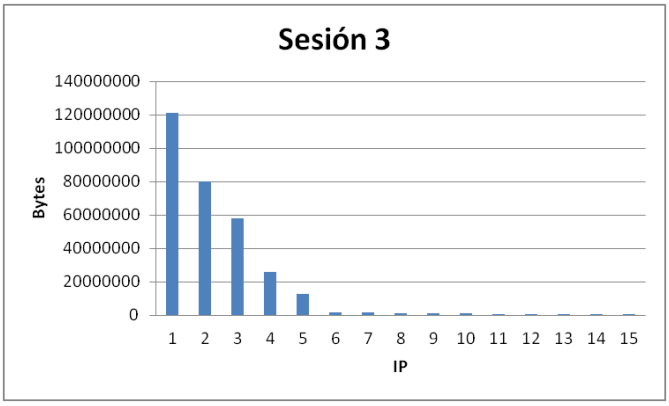
Gráficas de relación bytes/IP y paquetes/IP
SEMANA 1

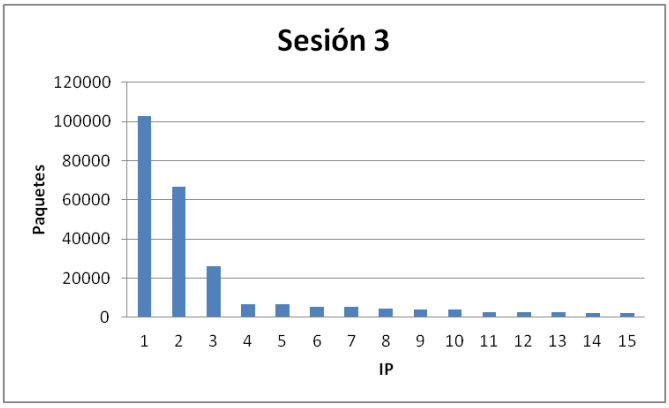


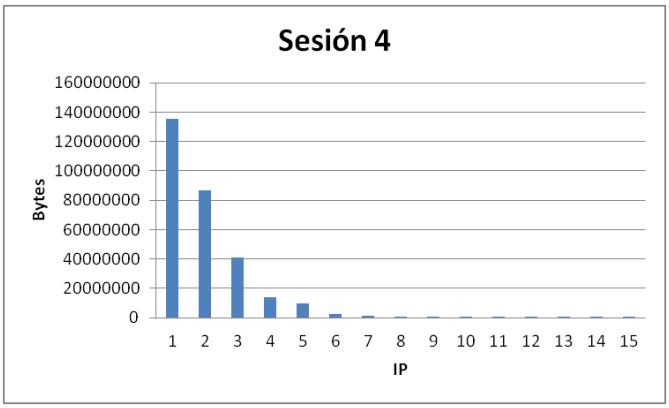


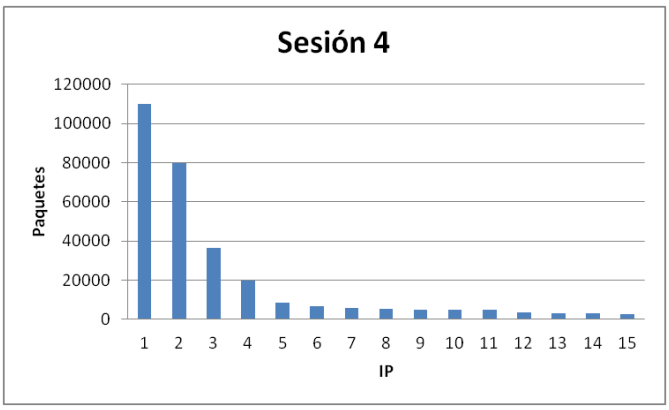


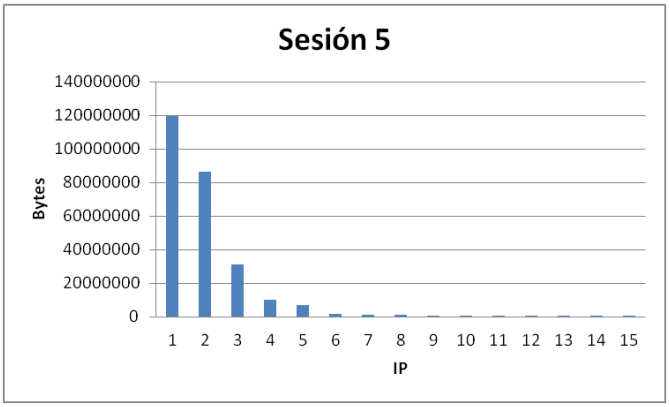


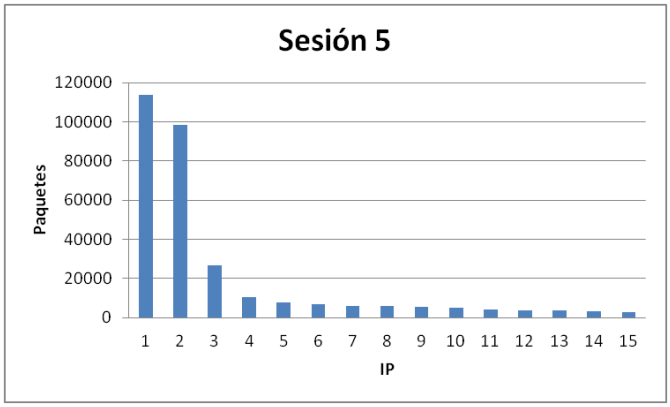




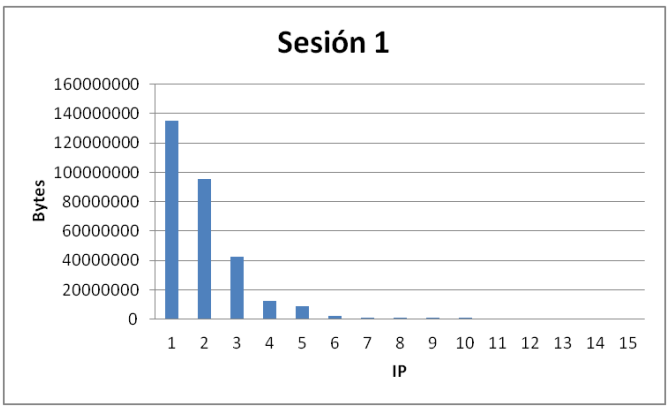

Sesión 4
Bytes
160000000
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

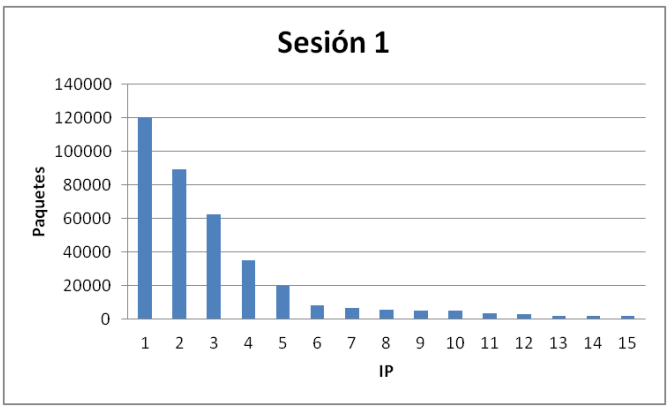

Sesión 4
Paquetes
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

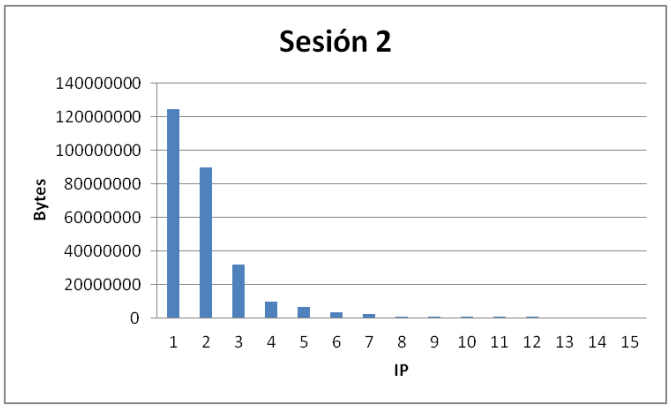

Sesión 5
Bytes
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

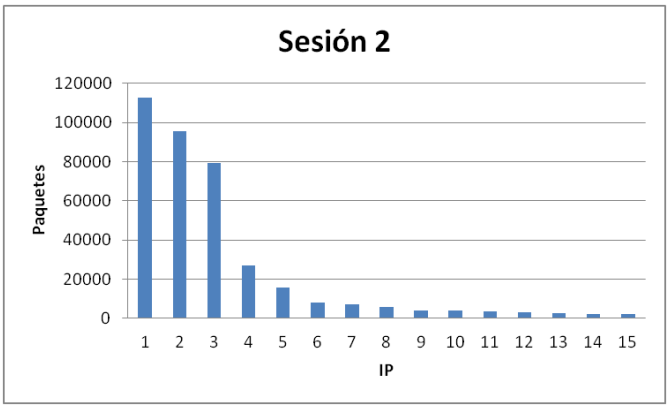

Sesión 5
Paquetes
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

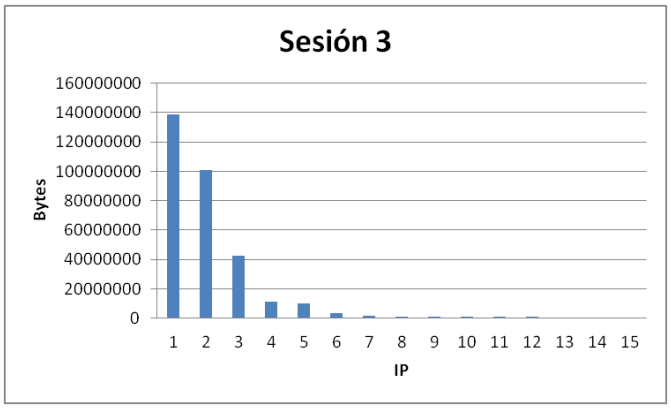
SEMANA 2

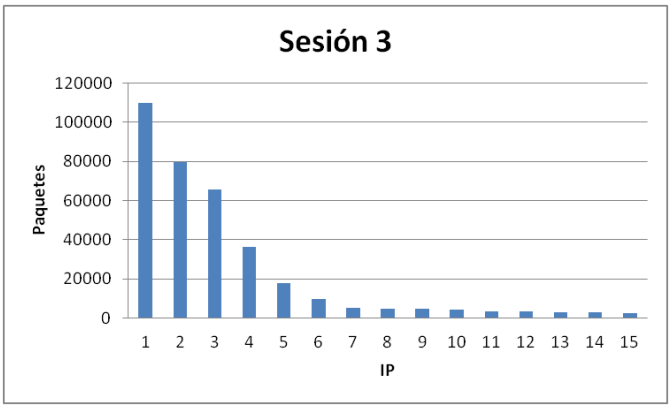

Sesión 1
Bytes
160000000
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

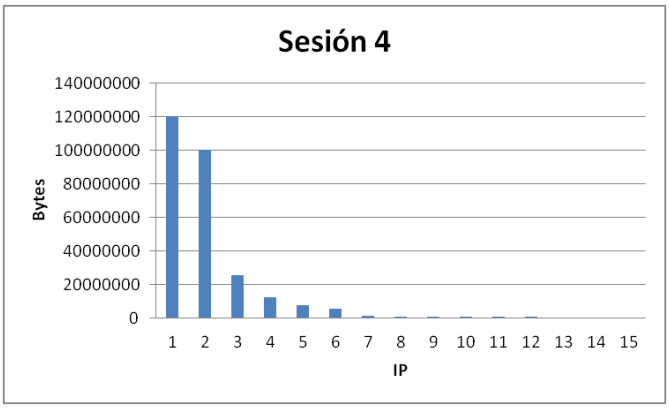

Sesión 1
Paquetes
140000
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

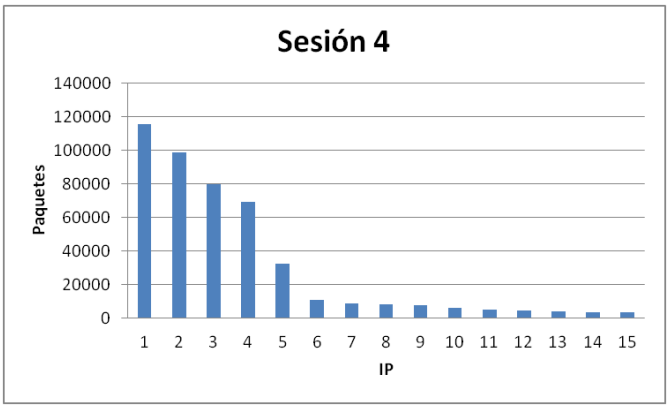

Sesión 2
Bytes
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

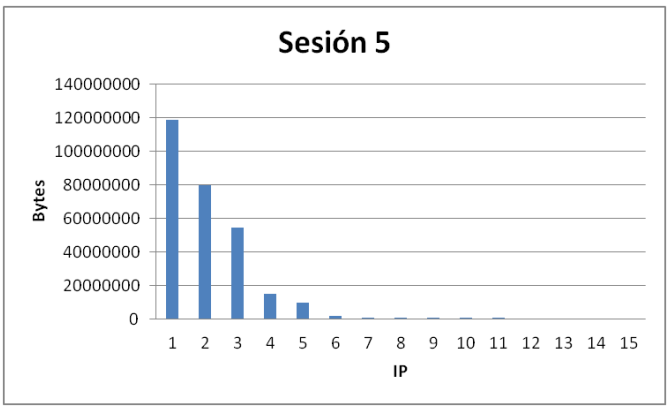

Sesión 2
Paquetes
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

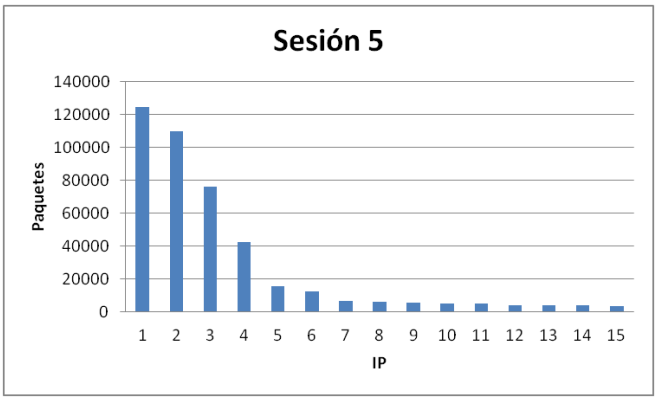

Sesión 3
Bytes
160000000
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP


Sesión 3
Paquetes
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

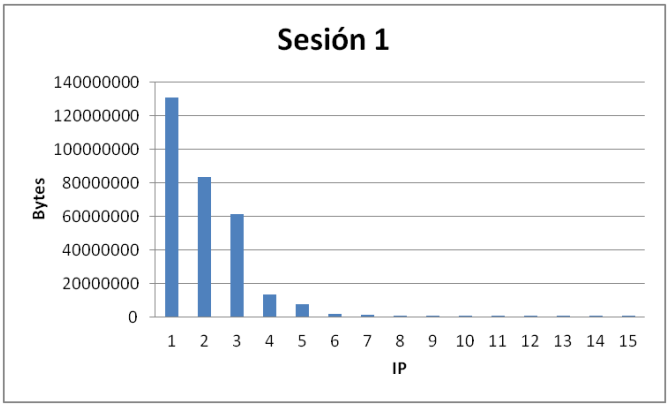

Sesión 4
Bytes
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

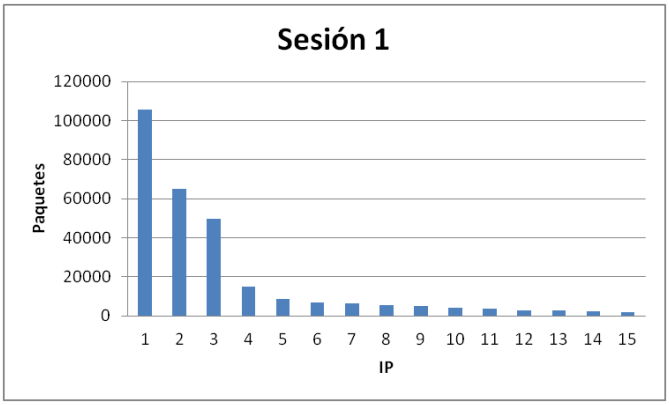

Sesión 4
Paquetes
140000
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

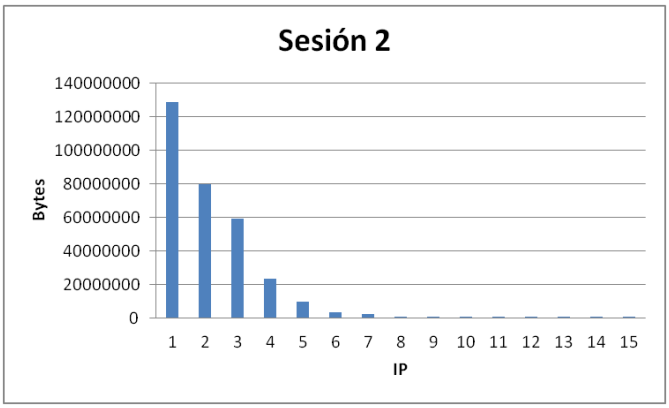

Sesión 5
Bytes
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

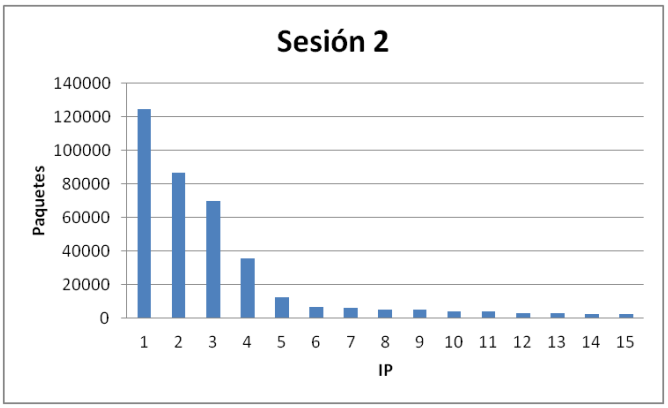

Sesión 5
Paquetes
140000
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

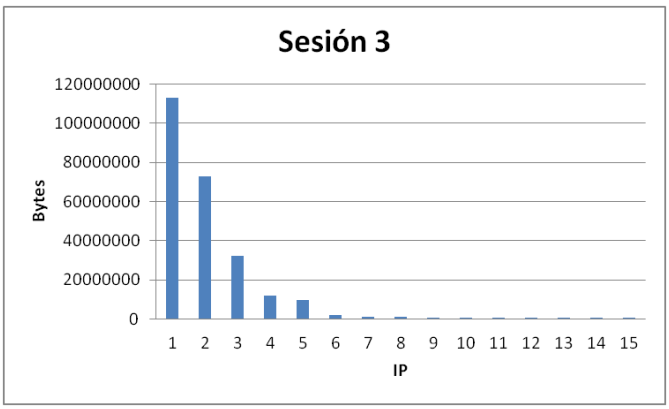
SEMANA 3

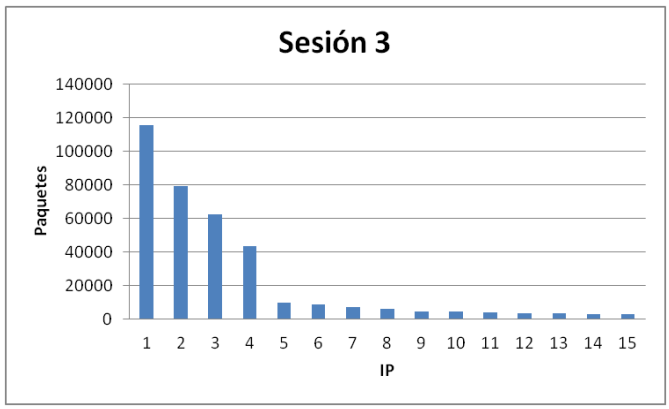

Sesión 1
Bytes
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

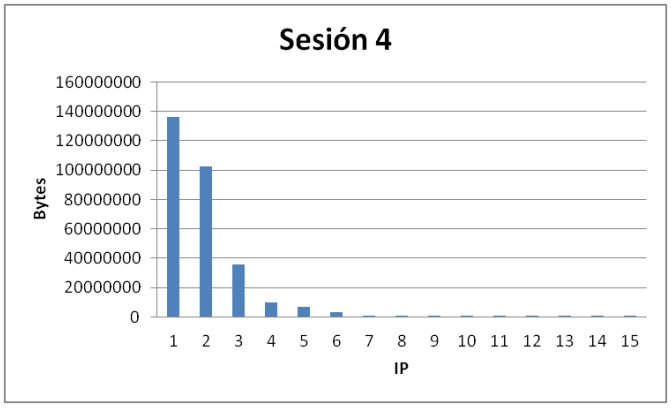

Sesión 1
Paquetes
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

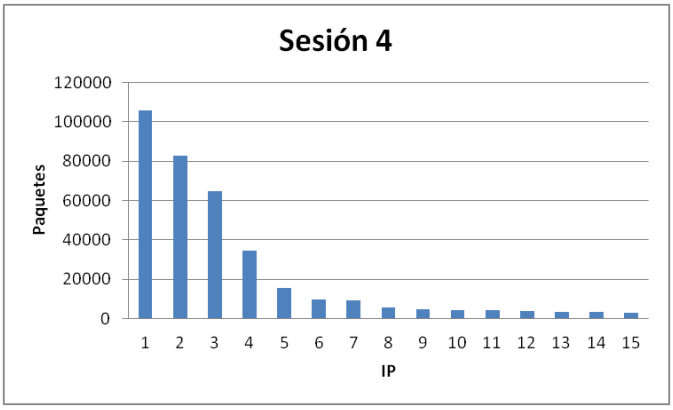

Sesión 2
Bytes
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

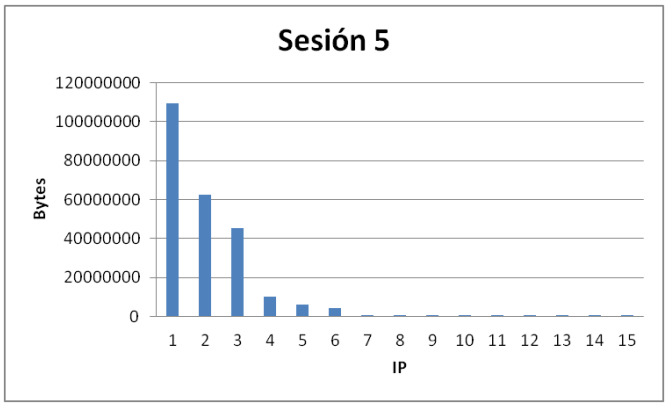

Sesión 2
Paquetes
140000
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

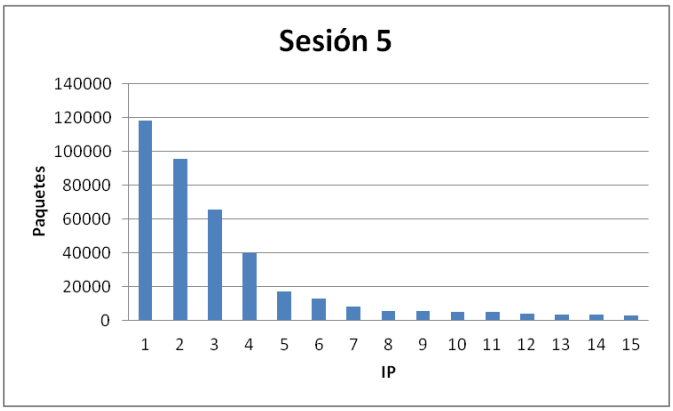

Sesión 3
Bytes
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP


Sesión 3
Paquetes
140000
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

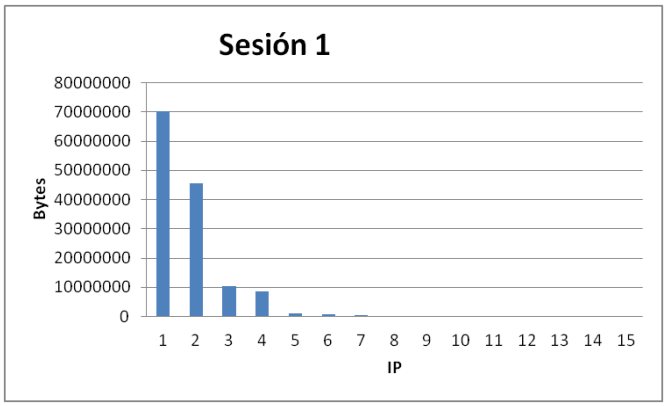

Sesión 4
Bytes
160000000
140000000
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

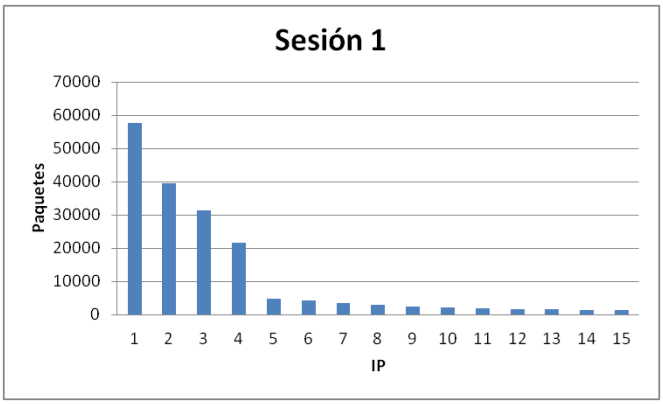

Sesión 4
Paquetes
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

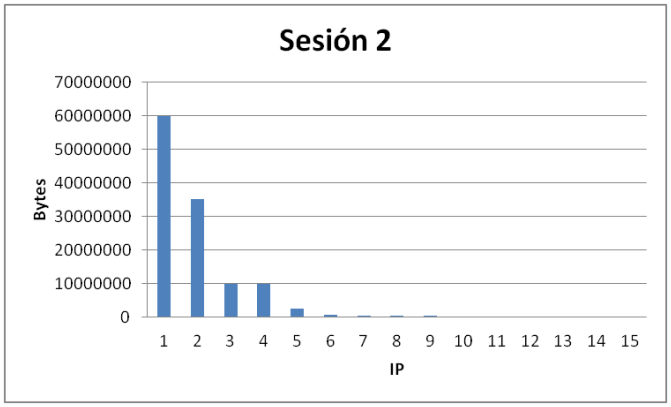

Sesión 5
Bytes
120000000
100000000
80000000
60000000
40000000
20000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

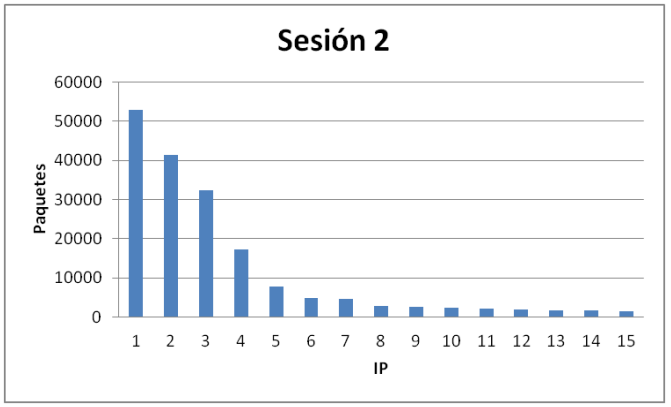

Sesión 5
Paquetes
140000
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

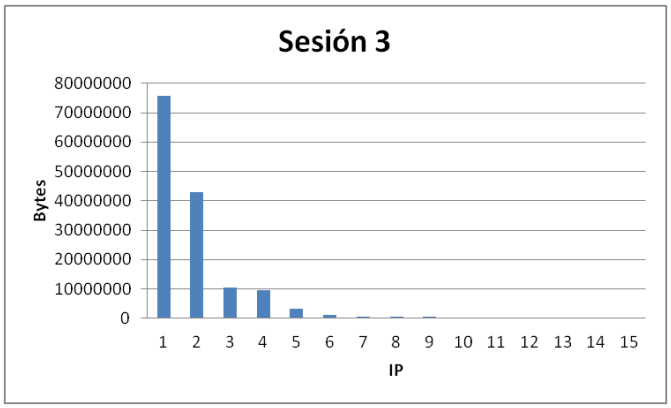
SEMANA 4

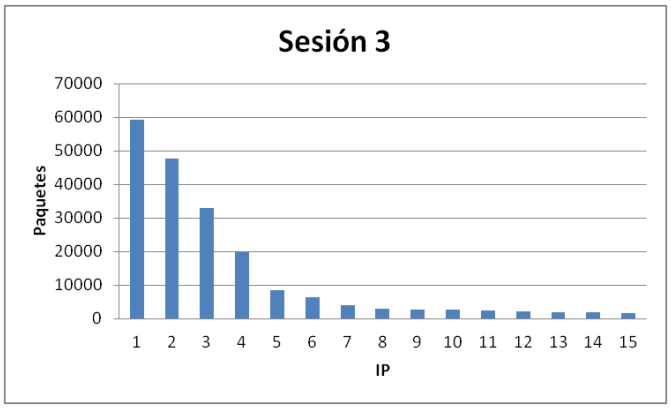

Sesión 1
Bytes
80000000
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

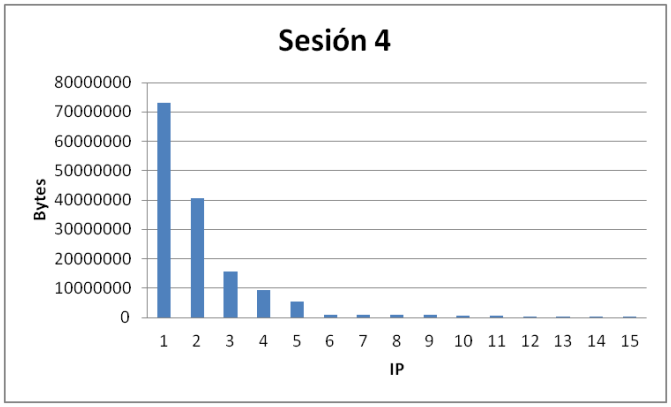

Sesión 1
Paquetes
70000
60000
50000
40000
30000
20000
10000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

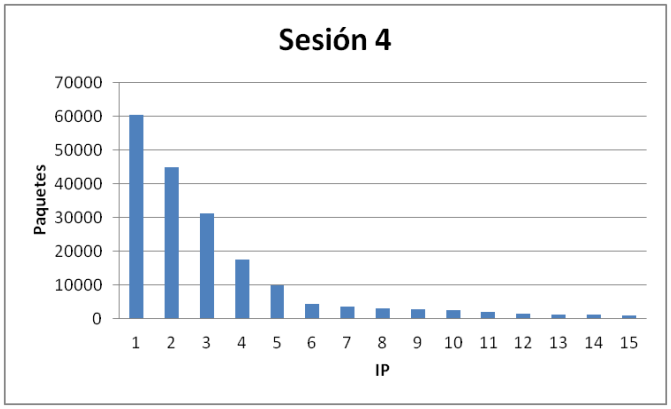

Sesión 2
Bytes
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

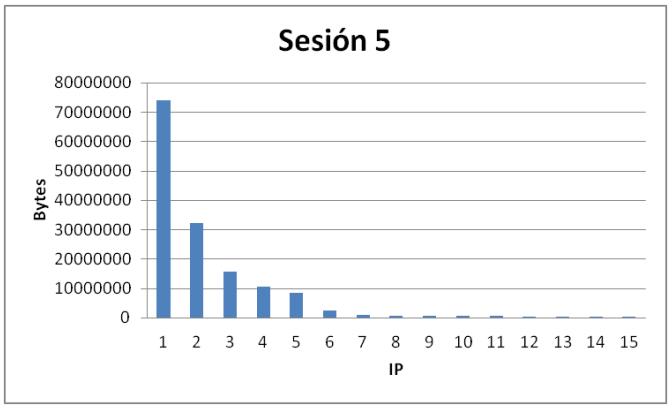

Sesión 2
Paquetes
60000
50000
40000
30000
20000
10000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

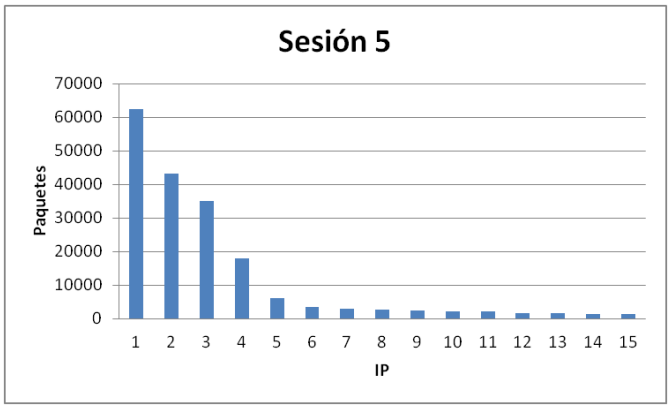

Sesión 3
Bytes
80000000
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP


Sesión 3
Paquetes
70000
60000
50000
40000
30000
20000
10000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

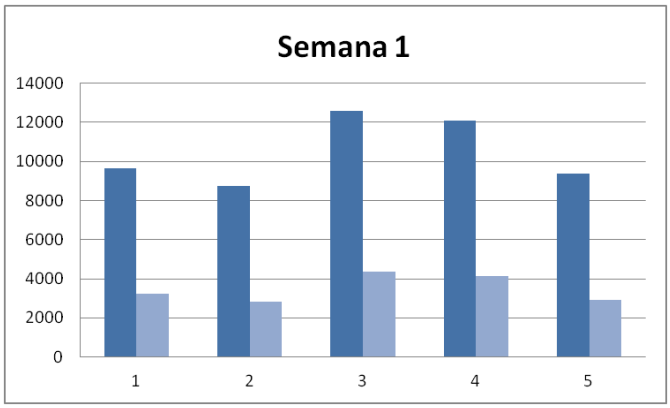

Sesión 4
Bytes
80000000
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

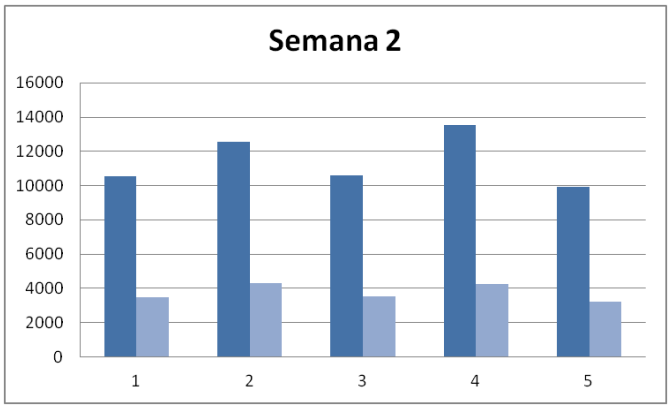

Sesión 4
Paquetes
70000
60000
50000
40000
30000
20000
10000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

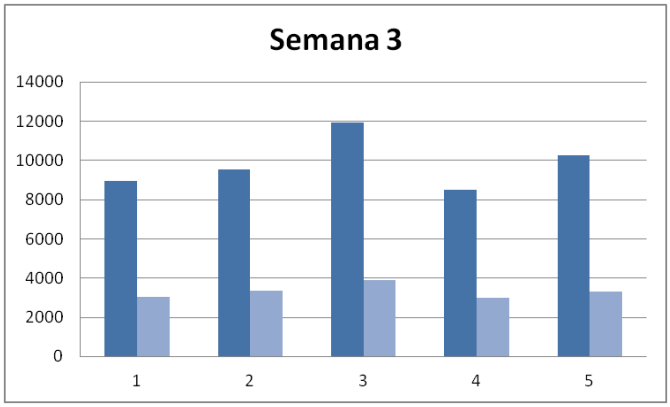

Sesión 5
Bytes
80000000
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

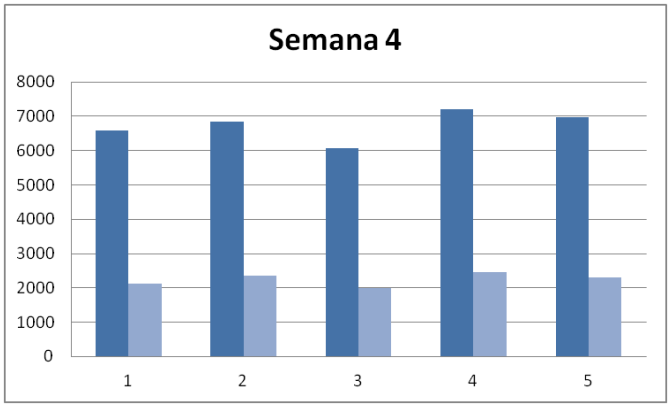

Sesión 5
Paquetes
70000
60000
50000
40000
30000
20000
10000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
IP

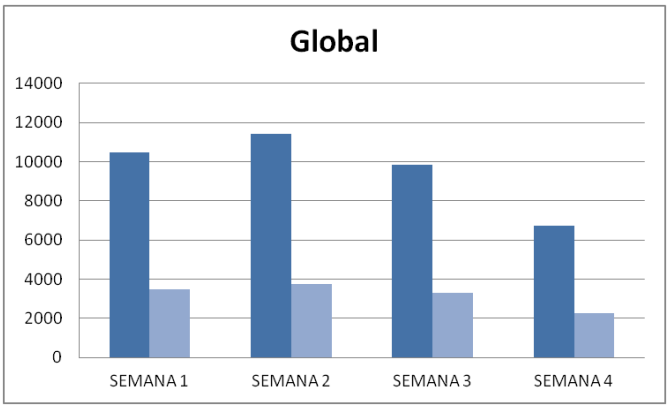
IP´s encontradas vs IP´s un solo paquete





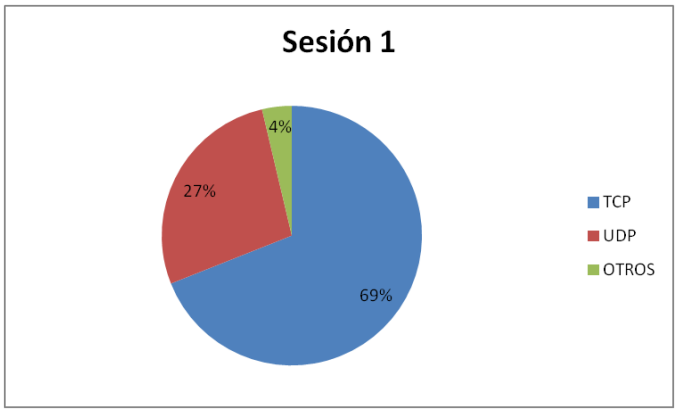


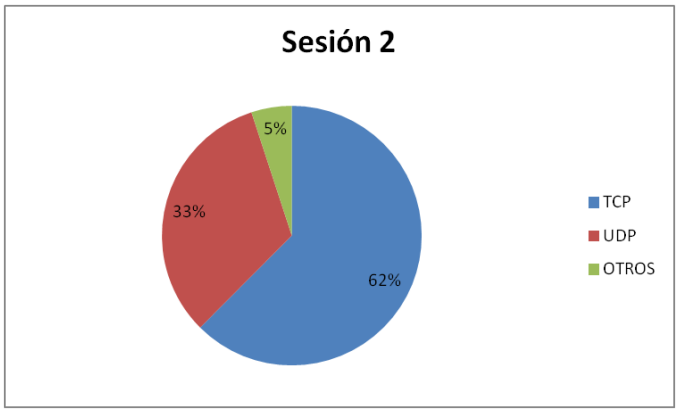


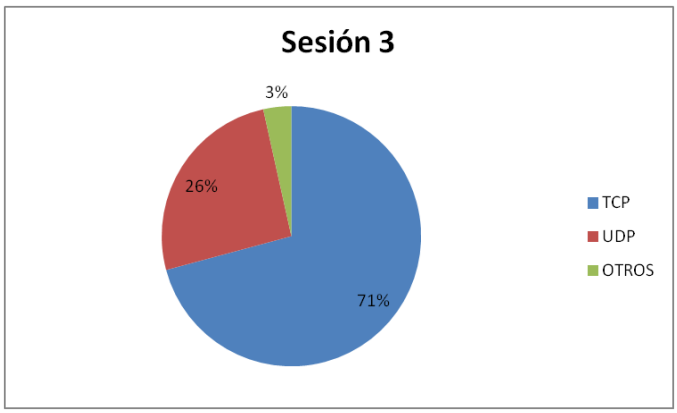


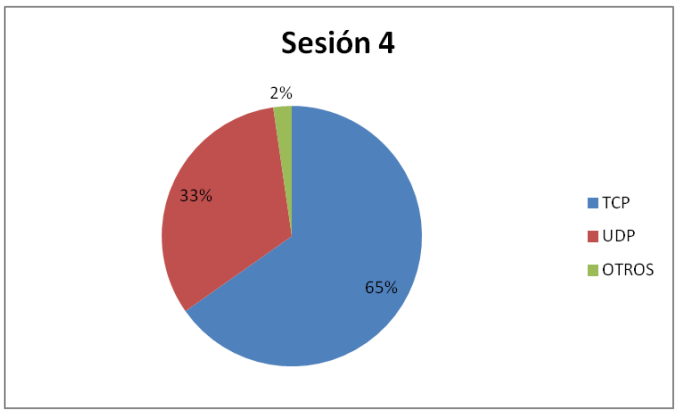
TCP vs UDP

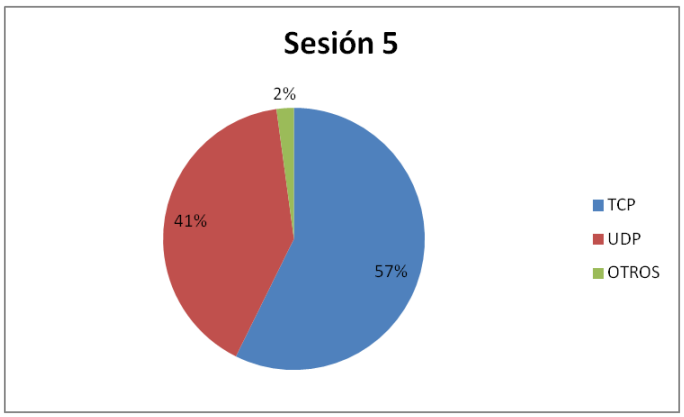
SEMANA 1

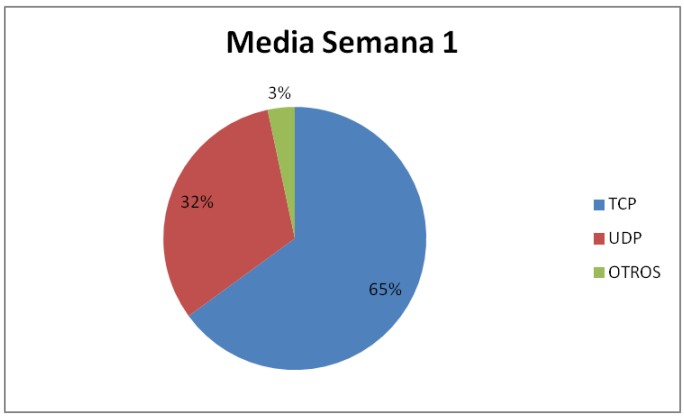




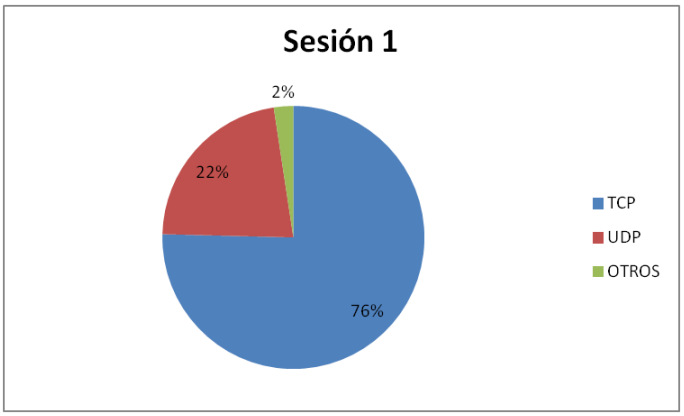


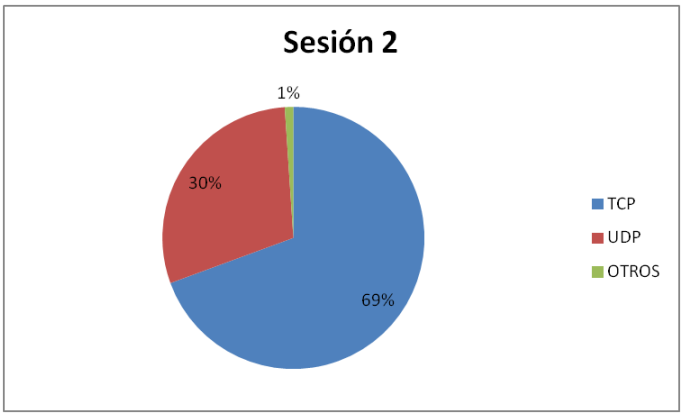


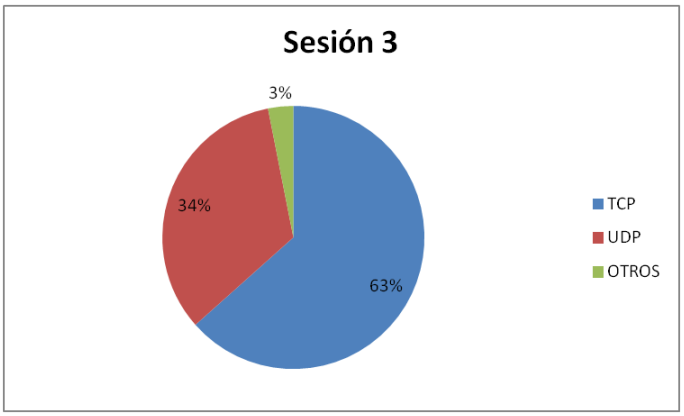


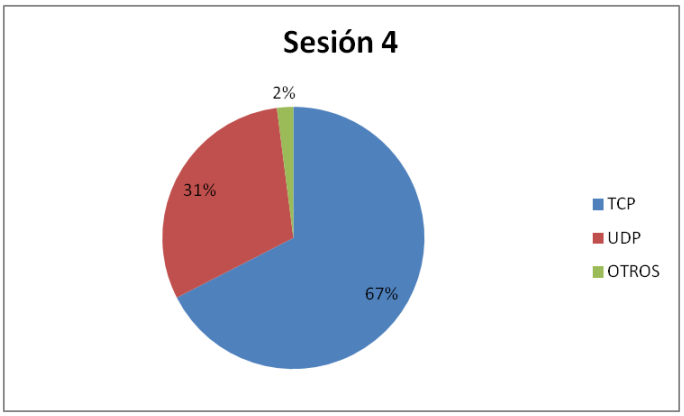


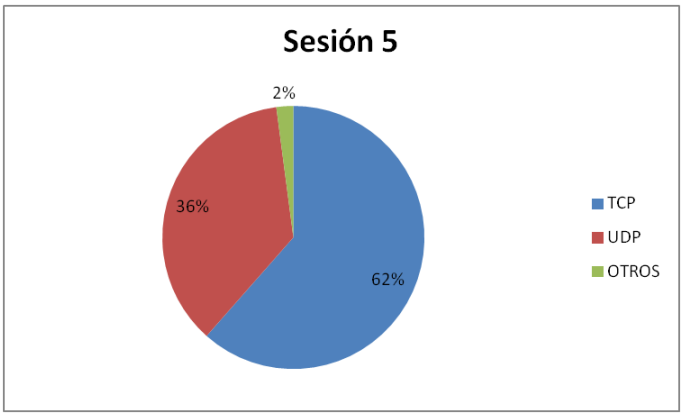
SEMANA 2

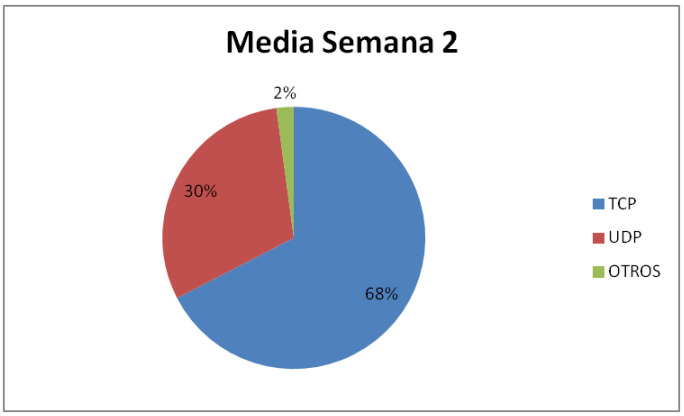

Sesión 1
2%
22%
76%
TCP
UDP
OTROS


Sesión 2
1%
30%
69%
TCP
UDP
OTROS

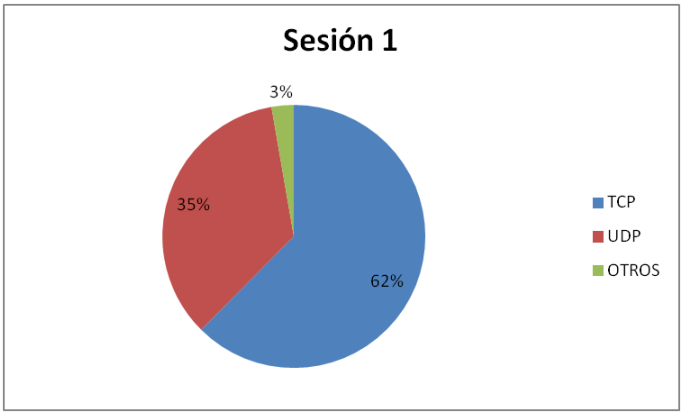

Sesión 3
3%
34%
63%
TCP
UDP
OTROS

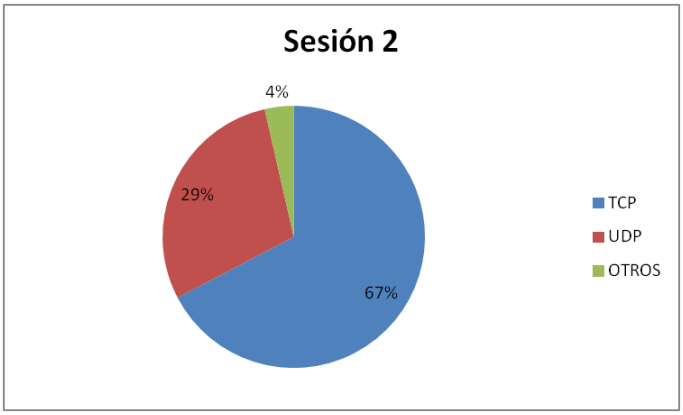

Sesión 4
2%
31%
67%
TCP
UDP
OTROS

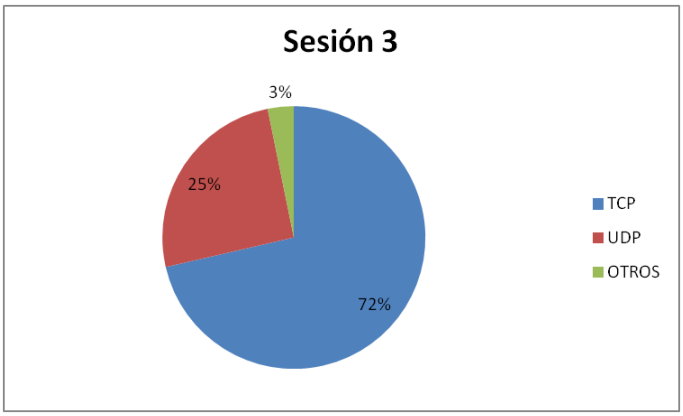

Sesión 5
2%
36%
62%
TCP
UDP
OTROS

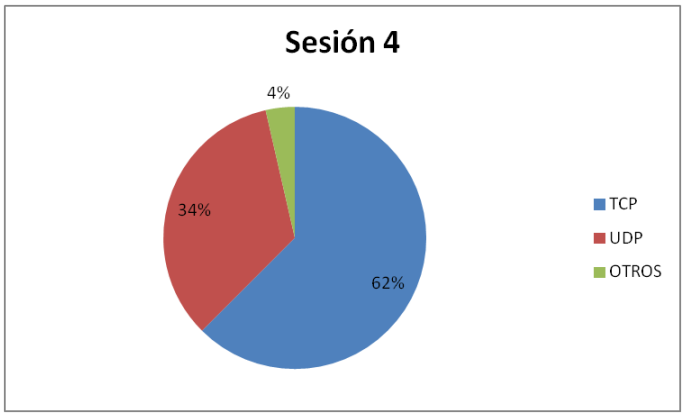

Media Semana 2
2%
30%
68%
TCP
UDP
OTROS

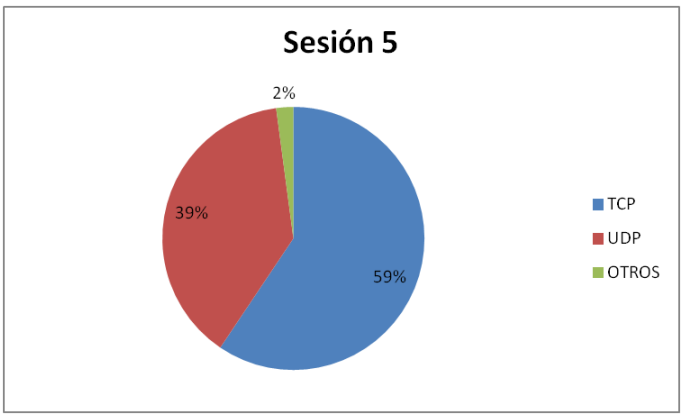
SEMANA 3

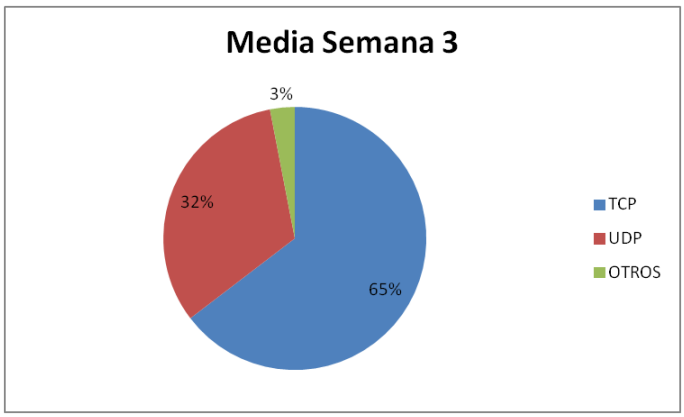

Sesión 1
3%
35%
62%
TCP
UDP
OTROS


Sesión 2
4%
29%
67%
TCP
UDP
OTROS

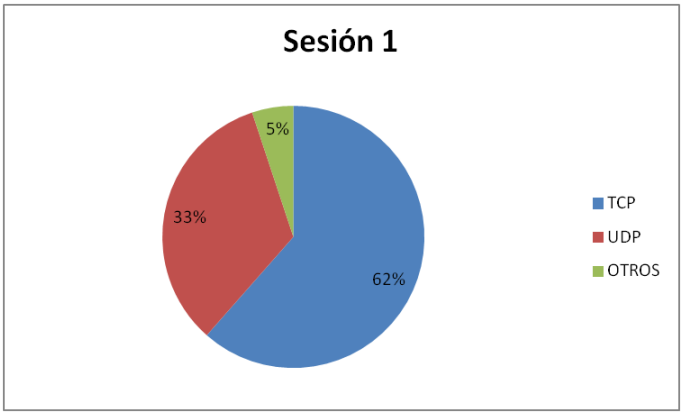

Sesión 3
3%
25%
72%
TCP
UDP
OTROS

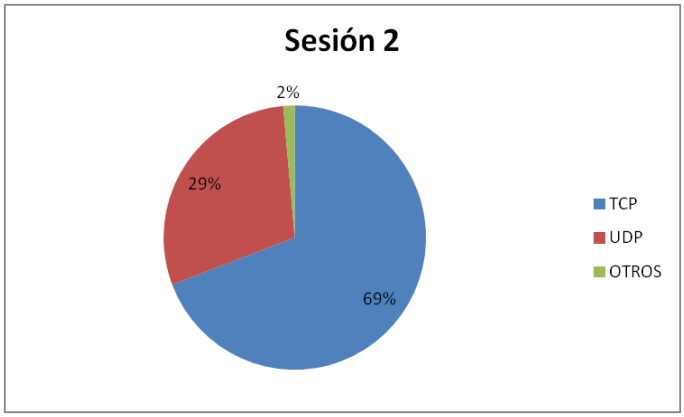

Sesión 4
4%
34%
62%
TCP
UDP
OTROS

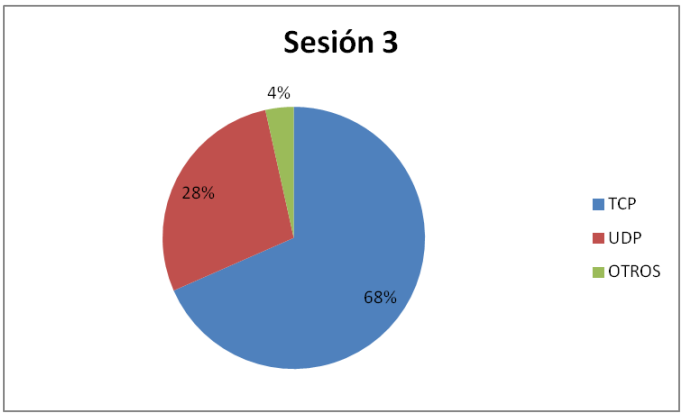

Sesión 5
2%
39%
59%
TCP
UDP
OTROS

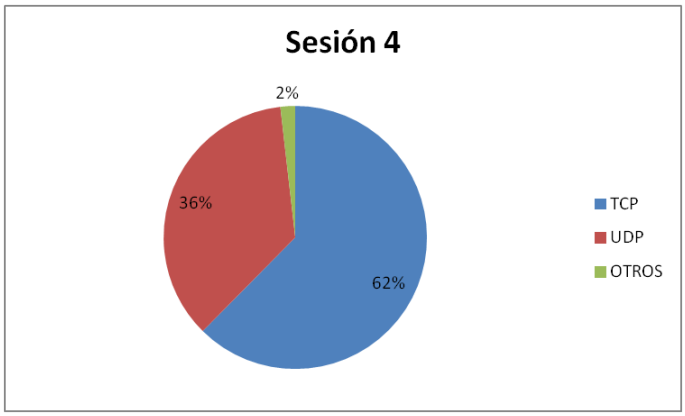

Media Semana 3
3%
32%
65%
TCP
UDP
OTROS

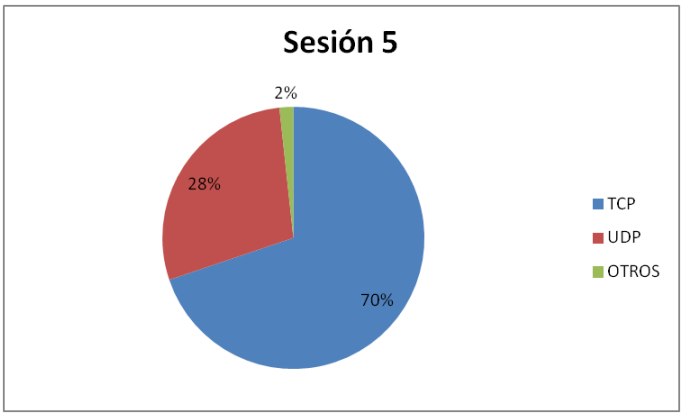
SEMANA 4

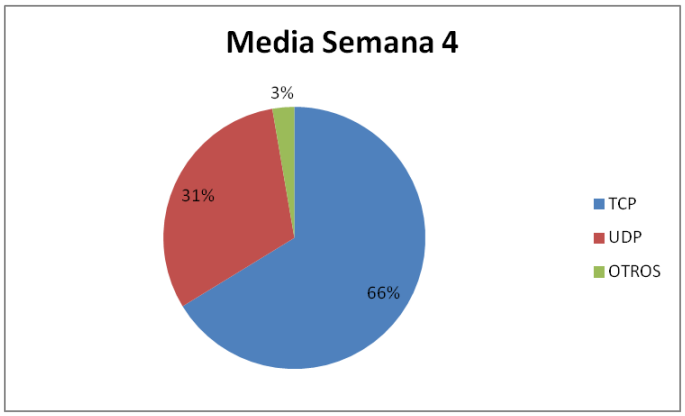

Sesión 1
5%
33%
62%
TCP
UDP
OTROS


Sesión 2
2%
29%
69%
TCP
UDP
OTROS


Sesión 3
4%
28%
68%
TCP
UDP
OTROS


Sesión 4
2%
36%
62%
TCP
UDP
OTROS


Sesión 5
2%
28%
70%
TCP
UDP
OTROS


Media Semana 4
3%
31%
66%
TCP
UDP
OTROS