

INFORME DE PRACTICAS DE PROTOCOLOS EN REDES

Este documento corresponde al informe de las prácticas 1,2, 3 y 4 de la asignatura de Redes y Sistemas Distribuidos para la ESIT (Escuela Superior de Ingeniería y Tecnología)



Informe de Practicas realizadas por:

- Eric Dürr Sierra (alu0101027005@ull.edu.es)
- Noah Sanchez Geurts (alu0101134956@ull.edu.es)
- Puede revisar el [repositorio github](#) donde se desarrolla.

Resumen

El siguiente documento expone varios aspectos recopilados a lo largo del desarrollo de las prácticas de protocolos. Tales aspectos conllevan desde una breve introducción a los aspectos característicos de cada uno de los protocolos hasta las conclusiones del análisis realizado por medio de la aplicación *Wireshark*.

Cabe destacar que es un estudio básico de los aspectos de los protocolos. No se pretende profundizar en gran medida en muchos de ellos.

También se expone una corta conclusión acerca de la ética del uso de la herramienta Wireshark.

Índice de Practicas

1. [Ethernet y ARP](#)
2. [IP y ICMP](#)
3. [UDP y TCP](#)
4. [HTTP](#)
5. [Ética de Wireshark](#)
6. [Respecto al trabajo grupal](#)
7. [Referencias](#)

Protocolo Ethernet y ARP

En este apartado se hablará sobre los protocolos Ethernet y ARP. Se entrará en detalles sobre sus cabeceras y funcionamiento.

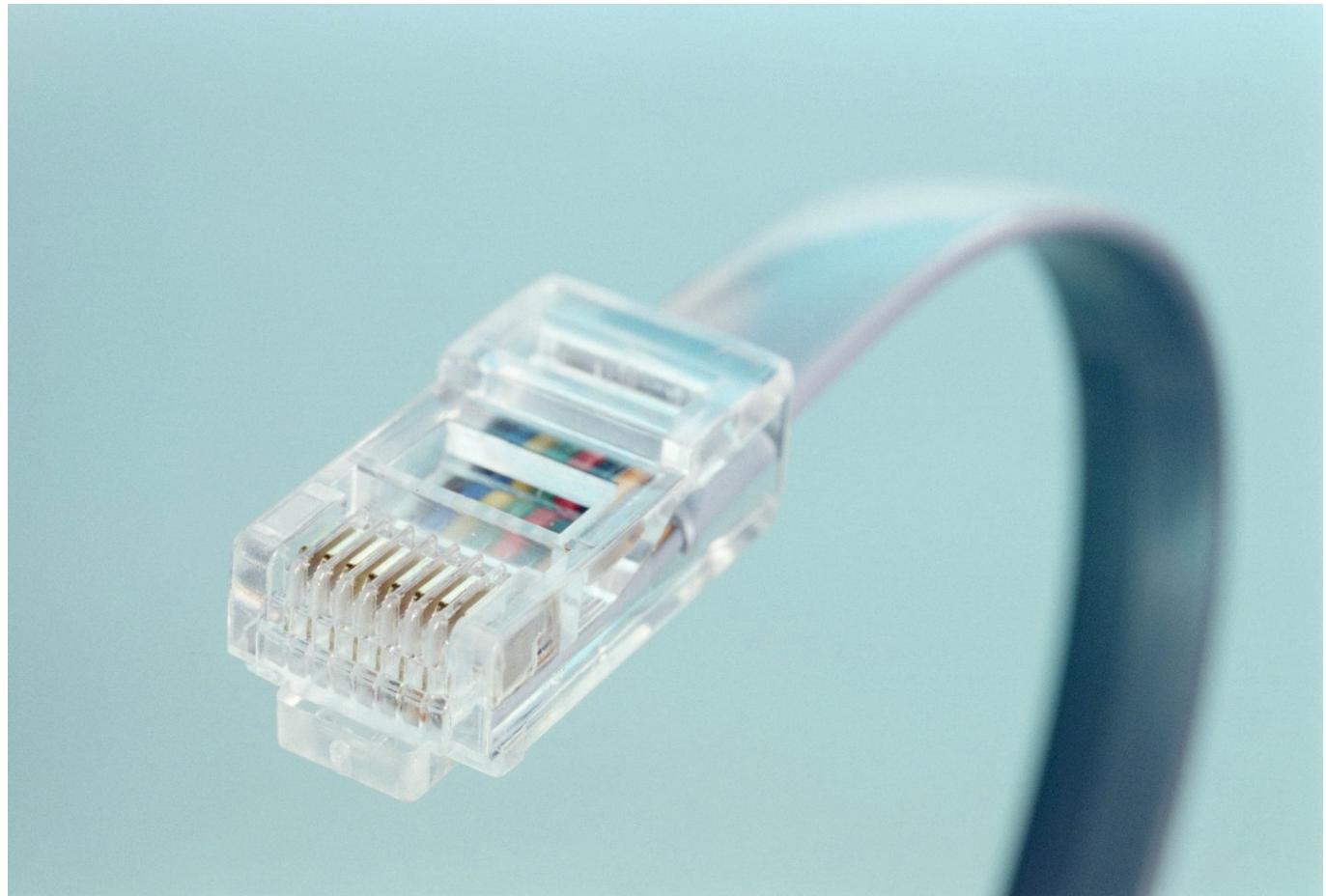


Imagen del cable RJ45, icono de este protocolo

PROTOCOLO ETHERNET

Antecedentes:

El protocolo Ethernet tiene como precursor al protocolo ALOHA siendo Ethernet una mejora basada en muchos de los conceptos de este.

Este protocolo es el más usado para áreas locales, cableadas, a nivel de enlace. Permite el acceso múltiple y consta de un esquema que le permite evitar colisiones.

Al estar a nivel de enlace se habla de una "trama" cuando nos referimos al paquete enviado. Esta es una traducción directa del mensaje en bits de la capa física y una encapsulación del datagrama de la capa de red, que es su superior.

Estructura del marco Ethernet

La trama Ethernet se compone por una carga útil; que es el datagrama, ya encapsulado, de la capa superior y por una cabecera Ethernet; que contiene información relativa al direccionamiento, control, uso y comprobación de la comunicación.

Por lo general se estructura en varias subcapas.

El estudio de este protocolo durante la práctica no ha profundizado en cuestiones demasiado concretas. Los aspectos que se presentan son en su mayoría básicos.

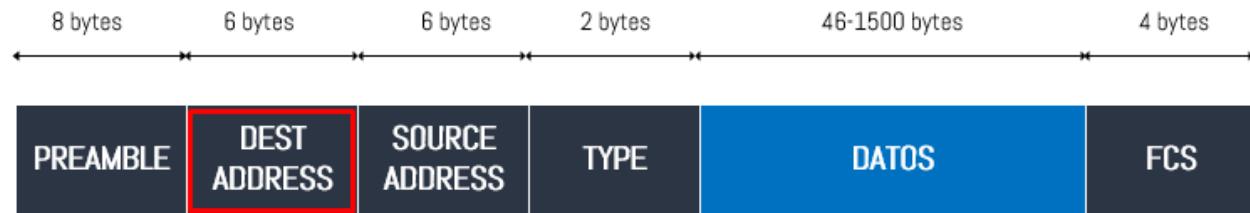


Imagen que expresa la trama Ethernet.

Dependiendo del estandar de Ethernet empleado el marco puede variar. Durante el curso aplicaremos el caso del estandar IEEE 802.3. Algunos elementos más generales pueden ser similares entre varios de estos. Por ejemplo en *Wireshark* se muestran los iguentes:

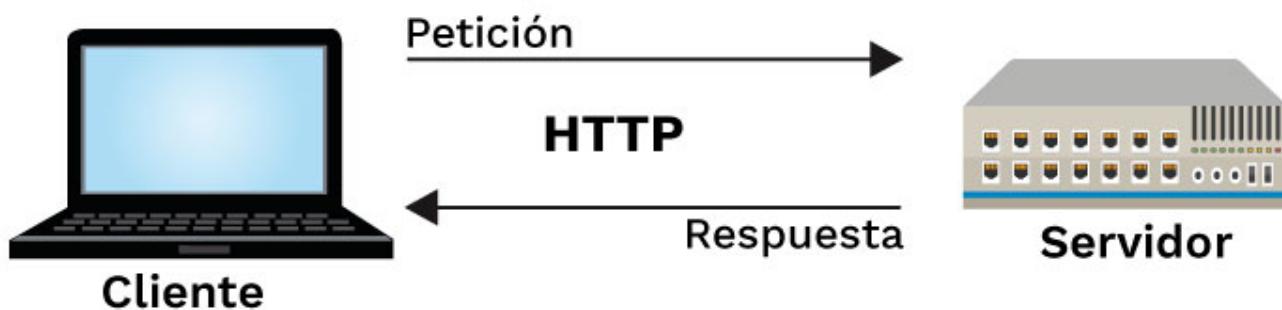
- Destino
- Origen
- tipo (*Frame type*)

Estos datos permiten conocer las direcciones MAC de los dispositivos interconectados en la misma red cableada. De esta manera se conoce el **origen** y el **destino** en una comunicación. Expresados bajo los campos del marco que indican esto.

Un ejemplo de este caso se produce en las **peticiones GET** a una página web. Donde el dispositivo origen solicita una serie de datos al dispositivo destino, que es el servidor donde se encuentra alojada.

Eventualmente el servidor contestará con una respuesta, es en este punto donde se intercambian los roles en cuanto a destinatario y origen. Este efecto se puede estudiar y visualizar mediante el marco del protocolo Ethernet.

Esta serie de valores y características son capturados por *Wireshark* en formato hexadecimal para expresar la cadena de bits que constituye la trama.



Esquema de una comunicación cliente-servidor.

El campo *frame type* indica el tipo de frame del paquete que se ha capturado o emitido en la comunicación.

Los valores que este campo puede adoptar son:

- Data Frame.
- Management Frame.
- Control Frame.

En función de la acción que esté resolviendo el protocolo adoptará un tipo u otro.

Sin embargo hay **campos que no podemos capturar** como el **CRC (cyclic redundancy check)** que está presente en la trama para realizar las comprobaciones pertinentes en cuanto a la detección de errores en la comunicación.

Este campo no es visible porque permanece en la tarjeta de red en lugar de ser transferido a la aplicación de *Wireshark*.

La popularidad del campo **CRC** recae en su sencillez y gran efectividad en cuanto a detección y corrección de errores.

Funcionamiento: El emisor y el receptor deben estar de acuerdo en un patrón de $r+1$ bits, que se conoce como generador (G). El bit más significativo de G debe ser "1".

Para una determinada secuencia de datos D , el emisor añadirá r bits adicionales R , de modo que los $r+d$ bits resultantes sean exactamente divisibles por G .*

En resumen los aspectos generales es el siguiente:

- Tecnología LAN más utilizada.
- Funciona en la capa de enlace de datos y en la capa física.
- Familia de tecnologías de redes que se define en los estándares IEEE 802.2 y 802.3.

PROTOCOLO ARP (Address Resolution Protocol)

El protocolo ARP suele ser vinculado con la capa de red ya que este ocupa la labor de reolución de direcciones. Se encarga de localizar la dirección del hardware relacionado con una dirección IP concreta.

Esta dirección del hardware se conoce como *Ethernet MAC*. Dicha dirección está expresada por un identificador de 48 bits y que es único desde su fabricación.

El protocolo ARP toma la responsabilidad de coordinar las direcciones físicas y lógicas en la comunicación mediante la red. Este protocolo se ayuda por la creación de una tabla de direcciones que emplea para la búsqueda.

Esta tabla se ubica en una memoria caché.

La tabla es recurrida por los equipos para comunicarse. De no existir el vínculo es ARP quien arbitra la solicitud. Todos los equipos dentro de la misma red comparan esta dirección fallida con la suya propia y emiten la respuesta al coincidir. Finalmente el par de direcciones es almacenado.

Se puede **relacionar ARP** con cuatro casos que se dan en la comunicación de dos hosts

- Hay dos *hosts* en una **misma red** y se quieren comunicar
- Dos *hosts* quieren comunicarse a través de un *router* **desde redes diferentes**.
- Cuando un *router* emite un paquete a **través de un router** a un *host*.
- **Dentro de la misma red**, un *router* quiere emitir un paquete a un *host*.

Se **puede caracterizar un paquete ARP** por una serie de campos que emplea para atender sus solicitudes de enlace:

- Tipo de hardware (*Hardware address space*)
- Tipo de protocolo (*Protocol address space*)
- Longitud de ID hardware (*Hardware address length*)
- Longitud de ID protocolo (*Protocol address length*)
- Operación (*Operation code*)
- direcciones del hardware (*Source/target hardware address*)
- direcciones del protocolo (*Source/target protocol address*)

Las longitudes se dan en bytes.

Los campos de direcciones de hardware y protocolo se expanden en otros dos.

Paquete solicitud/respuesta ARP	
Tipo de hardware	2 bytes
Tipo de protocolo	2 bytes
Longitud dirección de hardware en bytes (x)	2 bytes
Longitud dirección de protocolo en bytes (y)	
Código de operación	2 bytes
Dirección hardware del emisor	x bytes
Dirección IP del emisor	y bytes
Dirección hardware del receptor	x bytes
Dirección IP del receptor	y bytes

Apuntesdenetworking.blogspot.com

Se muestra un diagrama de la estructura de un paquete ARP

Cada uno de los campos son empleados para la correcta conexión entre dos puntos

Los primeros 4 bytes son ocupados por los tipos de hardware y protocolo. Estos son usados a fin de identificar el hardware y protocolo adecuados para establecer la comunicación.

Los siguientes 2 bytes son compartidos para representar la longitud de las direcciones tanto hardware como de protocolo en términos de bytes y refiriéndose al datagrama.

Por ejemplo para el protocolo IP expresaría 4.

Los consecutivos 2 bytes al último campo son utilizados para introducir el código de operación. Es en este apartado donde se indica que el datagrama es una solicitud o una respuesta.

En el caso de ser un 1 se habla de una operación de petición y de ser un 2 una respuesta sin embargo en la página [Network Sorcery](#) se habla más en profundidad de sus valores.

A partir del último campo lo que se incluye son, primero, las direcciones hardware e IP del emisor. A este le siguen los mismos campos pero para el receptor.

Nótese que la longitud variará en función de las longitudes registradas para cada parte de las direcciones.

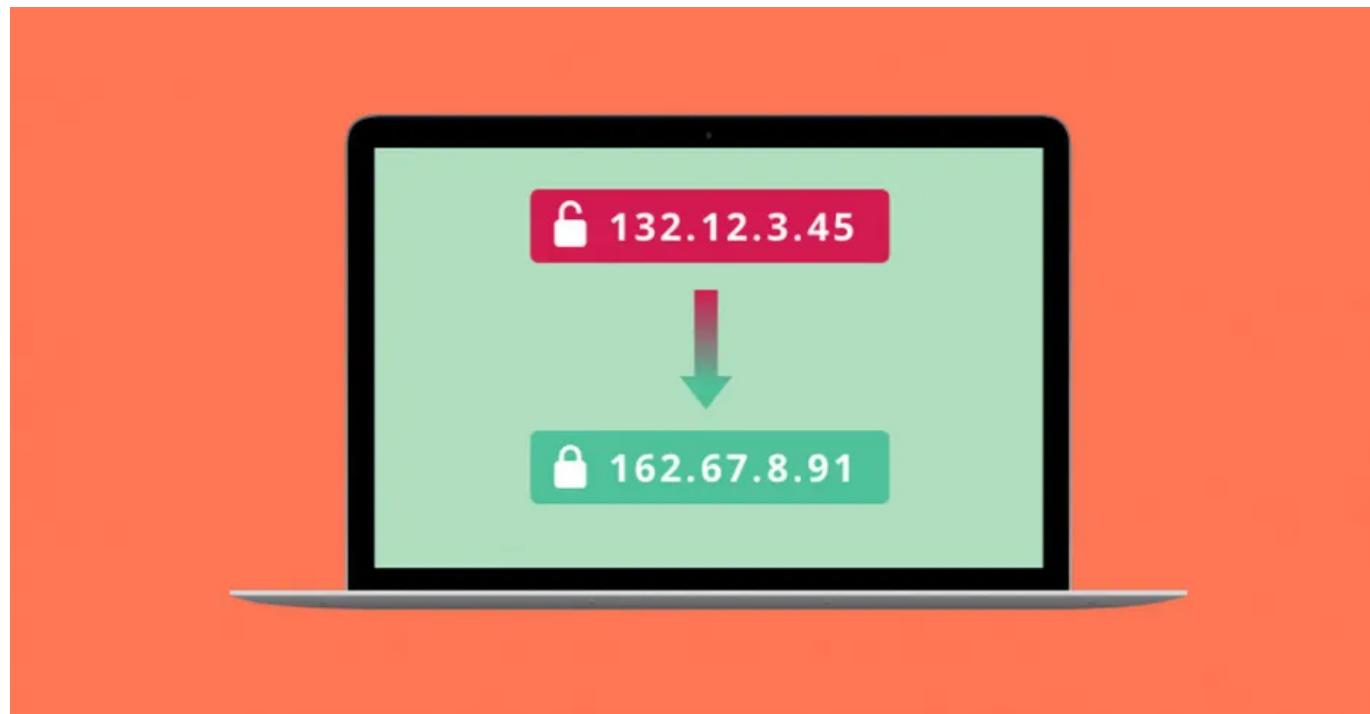
Protocolo IP e ICMP

En este apartado se hablará sobre los protocolos IP e ICMP.

Se hablará sobre la estructura de paquetes IP y sus campos de cabecera, así como la fragmentación.

IPv4, utilidad y necesidad de ICMP, comandos ping traceroute.

Ambos protocolos se encuentran en la capa de red.



Portada del Protocolo IP

Protocolo IP (Internet Protocol)

Introducción

El protocolo IP es la base fundamental de Internet, se encarga de mover datagramas a través de un conjunto de redes interconectadas. Como mencionamos antes, se encuentra en la capa de red, aunque en esta hay más componentes que hacen posible la funcionalidad de dicha capa:

- Protocolo de enrutamiento.
- Protocolo ICMP.

Estructura

Cada datagrama IP contiene:

- Cabecera.
- Datos a transmitir.

Funciones

- Mover datagramas entre un conjunto de redes interconectadas, pasándolos de un módulo a otro, hasta el destino.
- Los módulos residen en hosts y pasarelas en internet.
- Se encaminan a través de redes individuales mediante la interpretación de una dirección internet.
- En el enrutamiento entre módulos los datagramas pueden necesitar atravesar una red de menor tamaño.
- La cabecera contiene toda la información necesaria para que hosts y routers puedan encaminarlos a sus destinos y fragmentarlos cuando sea necesario.
- Fragmentacion: Mecanismo para aligerar los paquetes que superen el tamaño maximo permitido, se parte el paquete en trozos mas pequeños

Cabecera

0		10		20		30		
VERS	HLEN	Tipo de servicio		Longitud Total				
Identificacion			Banderas	Desplazamiento				
TTL	Protocolo		CRC Cabecera					
Direccion IP Origen			Direccion IP Destino					
Opciones IP (Si las hay)				Relleno				

Cabecera de un datagrama del protocolo

La cabecera tiene 20 bytes de longitud, es decir, 5 palabras, donde se encuentran diferentes campos, como:

- VERS: Indica la version del protocolo (IPv4 o IPv6)
- HLEN: Longitud de la cabecera
- Tipo de Servicio: Es el servicio solicitado por el datagrama IP
- Longitud Total: Longitud total del datagrama (Datos y Cabecera)
- Identificacion: Indica a que datagrama pertenece el fragmento para ayudar a reunir los fragmentos del datagrama anteriormente fragmentado
- Banderas o flags: Sirven para el control de la fragmentacion
- Desplazamiento de Fragmento: Se usa en datagramas fragmentados para ayudar al reensamblado del datagrama completo
- TTL: Es un valor incluido para que los datagramas no esten en bucles de enrutamiento infinitos. Su valor decrementa en 1 cada vez que pasa por el router, si llega a 0, la trama se descarta.
- Protocolo: Indica el numero del protocolo de alto nivel al que IP deberia entregar los datos del datagrama
- CRC: Es el checksum de cabecera
- Direccion IP origen: Contiene la direccion del emisor
- Direccion IP destino: Contiene la direccion de destino
- Opciones IP: Su longitud varia dependiendo de la funcion de la opcion que tenga

Fragmentacion

Es una solución al problema que se nos presenta cuando queremos enviar un paquete de datos de un tamaño superior al que puede enviar el protocolo de la capa de enlace. Para solucionar esto, se fragmentan los datos del datagrama en varios datagramas más pequeños, cada uno de estos datagramas más pequeños se llaman fragmentos.

Si fragmentamos el paquete, habrá que ensamblarlo en algún momento. Esto se suele hacer en el dispositivo receptor (para no sobrecargar a los routers y no complicar el protocolo). Para lograr este ensamblado, se usan los campos de la cabecera, que nos indican a qué paquete pertenecen los fragmentos, si el fragmento es el último o no y en qué posición del datagrama va el fragmento. De esta forma es factible hacer la fragmentación y el ensamblado de forma correcta y precisa.

PROTOCOLO ICMP (Internet Control Message Protocol)

Introducción

El protocolo ICMP es otro de los componentes de la capa de red. La funcionalidad de este protocolo es el intercambio de información acerca de la capa de red, generalmente mensajes de error.

Usa el soporte básico de IP como un protocolo de nivel superior (Es realmente una parte de IP). Concretamente, ICMP, se sitúa por encima de IP, el motivo de esto es que todos los mensajes ICMP son transportados dentro de datagramas IP. Es decir, el mensaje ICMP es la carga útil del datagrama.

Las cabeceras de estos mensajes son bastante sencillos, solamente tienen 3 campos:

Tipo	Código	Suma de Comprobación
Datos		

Cabecera de un mensaje ICMP

- Tipo: En este campo está situado el tipo de mensaje, como el tipo de error que ha ocurrido o qué solicitud se ha hecho.
- Código: Subtipo del primer campo Tipo, precisa el motivo.
- Suma de Comprobación: Datos de comprobación de errores

Campo de Tipo	Tipo de Mensaje ICMP
0	Respuesta de eco
3	Destino inaccesible
4	Disminucion del trafico desde el origen
5	Redireccionar
8	Solicitud de eco
11	Tiempo excedido para un datagrama
12	Problema de parametros
13	Solicitud de marca de tiempo
14	Respuesta de marca de tiempo
15	Solicitud de informacion
16	Respuesta de informacion
17	Solicitud de mascara
18	Respuesta de mascara

Campo Tipo y significados

Comandos Ping y Traceroute

- Comando Ping

- Utiliza el protocolo ICMP para envía una petición de eco a un host o router.
- Al emitir echo request se recibe un echo response, estos datagramas se componen por:
 - Cabecera IP + Cabecera ICMP + carga util: estampa de tiempo y numero de bytes de relleno
 - Con el mecanismo se pretende averiguar si el nodo destino es alcanzable, y saber su tiempo de ida y vuelta.

- Comando Traceroute:

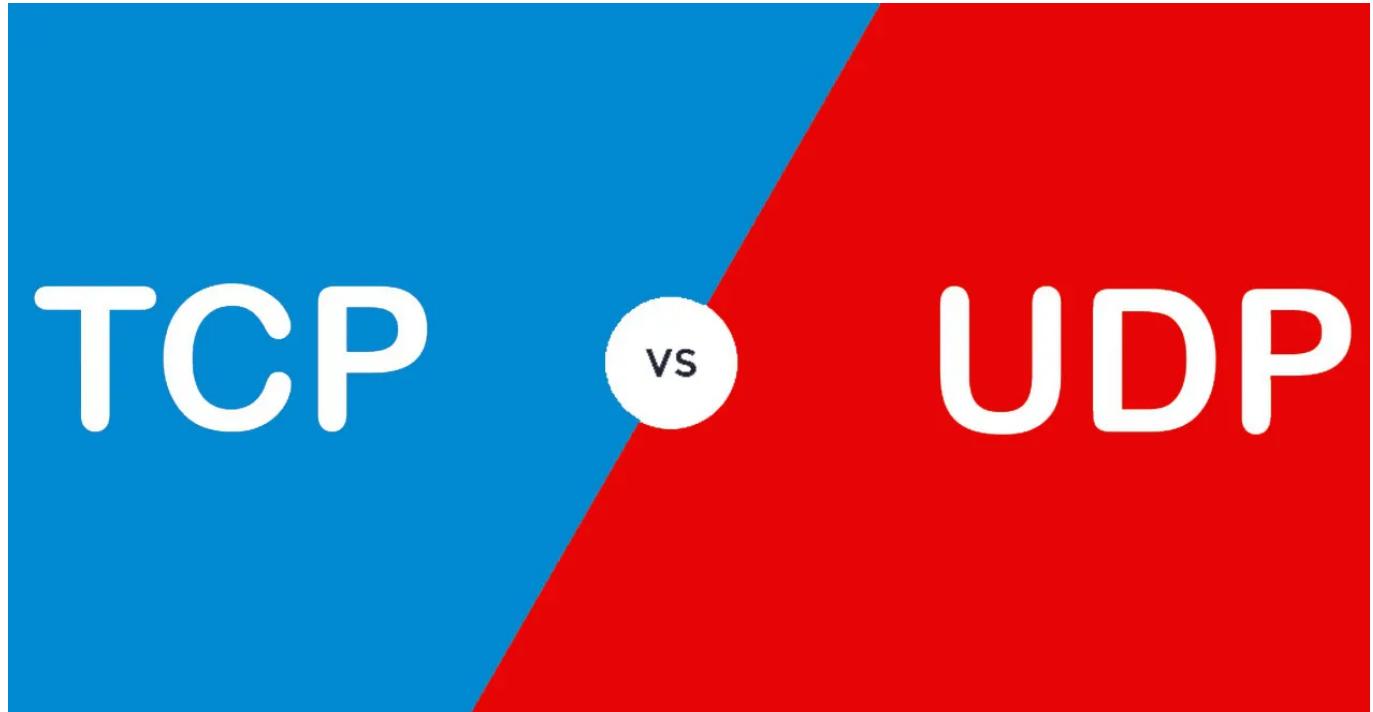
- Determinar el recorrido entre dos hosts, usa el campo TTL de la cabecera IP
- Un paquete podría quedar vagando por la red, por lo tanto debe existir un mecanismo que detecte esto.
- El campo TTL asigna un valor por defecto (64) en el emisor, dicho valor decrementa al cruzar un router.
- Si el valor llega a 0 el paquete es descartado y se envía un mensaje ICMP (TTL exceeded) al emisor.
- Traceroute construye un paquete para emitirlo entre dos hosts, TTL toma el valor 1 en este caso.
- Vuelve al emisor tras devolverlo el primer router, luego se emite con TTL valor 2 (causando otro mensaje ICMP)

- El proceso sigue hasta alcanzar el destino, al llegar toma la carga útil y la interpreta:

- Carga útil como mensaje ICMP "Echo request", se responde con Echo reply.
 - Carga útil como datagrama UDP a un puerto aleatorio, se responde con Destination unreachable (ICMP)
 - Carga útil como segmento TCP a un puerto aleatorio incluyendo un flag de sincronización activo -> Destination unreachable (ICMP)
- Cuando el host emisor recibe el mensaje se termina la traza y se finaliza el proceso

Protocolo UDP y TCP

En este apartado se hablará sobre los protocolos UDP y TCP.
Se hablará sobre la estructura y funciones generales de UDP y TCP.
Tambien se comentará el cierre TCP y el funcionamiento ACK en TCP.



Portada para ambos protocolos

Protocolo UDP (User Datagram Protocol)

Introducción

El protocolo UDP es un protocolo de la capa de transporte, por lo que trabaja con segmentos. Este protocolo se limita a proporcionar las cosas mínimas que la capa de transporte debe realizar. Su principal funcionalidad es pasar los mensajes de la capa de transporte a la de red, y los datagramas de la capa de red a la de transporte.

Propiedades de UDP:

- Incluye detección de errores
- Si se utiliza UDP, la aplicación se comunica casi directamente con el protocolo IP

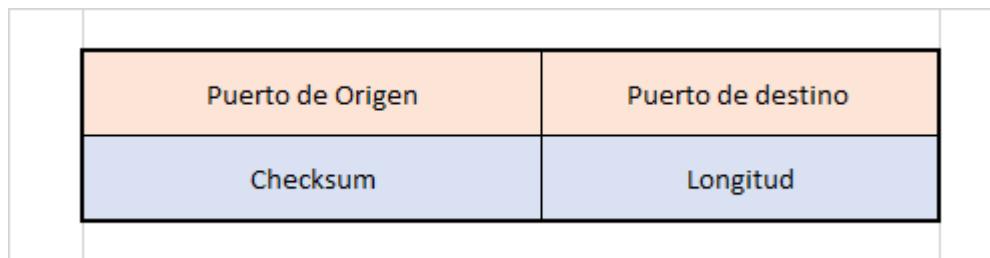
Características de UDP

- Mejor control en el nivel de aplicación: Esta se refiere al envío y cuando se realiza este. Cuando un proceso de la capa de aplicación le pasa los datos, los empaqueta (en un segmento UDP) e inmediatamente entrega dicho segmento a la capa de red, debido a que no posee un mecanismo de control que regule el flujo (como pasa con TCP). Esto afecta a las velocidades de transmisión.

- Sin establecimiento de la conexión: A diferencia de TCP, que debe realizar un proceso de la conexión antes de realizar la transferencia de datos, UDP no añade ningún retardo debido al establecimiento de conexión.
- Sin información del estado de la conexión: UDP no tiene control sobre la información del estado de la conexión y no controla ninguno de estos parámetros.
- Poca sobrecarga debida a la cabecera de los paquetes: Los segmentos UDP solo requieren 8 bytes en la cabecera, mientras los TCP, requieren 20 bytes.

Cabecera

Tiene un total de 4 campos de 2 bytes cada uno, por lo que ocupa un total de 8 bytes



Cabecera de un mensaje UDP

- Puerto de origen: Indica el puerto del proceso que envía, es el puerto que se direcciona en las respuestas.
- Puerto de destino: Especifica el puerto del proceso destino en el host de destino.
- Longitud: Es el tamaño, en bytes, de este datagrama de usuario incluyendo la cabecera.
- Checksum: Es un campo opcional, de 16 bits, que permite realizar la suma de comprobación.

Protocolo TCP (Transmission Control Protocol)

Introducción

El protocolo TCP es uno de los protocolos más usados en Internet. Forma parte de la capa de transporte. Los principales servicios que ofrece son de comunicación segura, orientado conexión, full-duplex, punto a punto y, evidentemente, multiplexación/demultiplexación.

La conexión que se establece se denomina "acuerdo en tres fases". Esto es porque para establecerla, hay que enviar un total de 3 paquetes.

Los dos primeros segmentos enviados no llevan ninguna carga útil. En este inicio de conexión también se acuerdan parámetros de la comunicación como los buffers de emisión.

Cabe resaltar que esta conexión es conocida únicamente por los nodos extremos de la comunicación. Los nodos intermedios (routers y switches) no saben de su existencia. Por el simple hecho de que estos dispositivos solo ven datagramas, no conexiones.

Cada segmento tiene una cabecera de formato variable debido al campo de opciones. Sin opciones, la cabecera ocupa un total de 20 bytes. Como se aprecia en la siguiente imagen, tiene los siguientes campos:

Puerto de Origen	Puerto de Destino		
Número de Secuencia			
Número de confirmación de recepción			
Longitud Encabezado TCP	Reservado	Flags	Tamaño de Ventana
Suma de Verificación		Apuntador Urgente	
Opciones			

Cabecera de un mensaje UDP

- Puerto origen: Identifica el número de puerto de un programa de aplicación de origen.
- Puerto destino: Identifica el número de puerto de un programa de aplicación de destino.
- Número de secuencia: Especifica el número de secuencia del primer byte de datos de este segmento.
- Número de confirmación de recepción: Contiene el valor del siguiente número de secuencia que el emisor del segmento espera recibir.
- Longitud encabezado TCP: Especifica el tamaño de la cabecera en palabras de 32 bits.
- Reservado: Se deja para uso futuro, debe estar a 0.
- Flags: Se emplean para diferentes variables como SYN, FIN, RSt, ACK, etc...
- Tamaño de ventana: Tamaño de la ventana de recepción que especifica el número máximo de bytes que pueden ser introducidos en el buffer.
- Suma de verificación: Utilizado para la comprobación de errores tanto en cabecera como en datos.
- Puntero urgente: Cantidad de bytes desde el número de secuencia que indican el lugar donde acaban los datos urgentes.
- Opciones: Se utiliza para poder añadir características no cubiertas en la cabecera.

Para poder proporcionar una transferencia de datos fiables, TCP usa los campos "Número de secuencia" y "Número de confirmación de recepción".

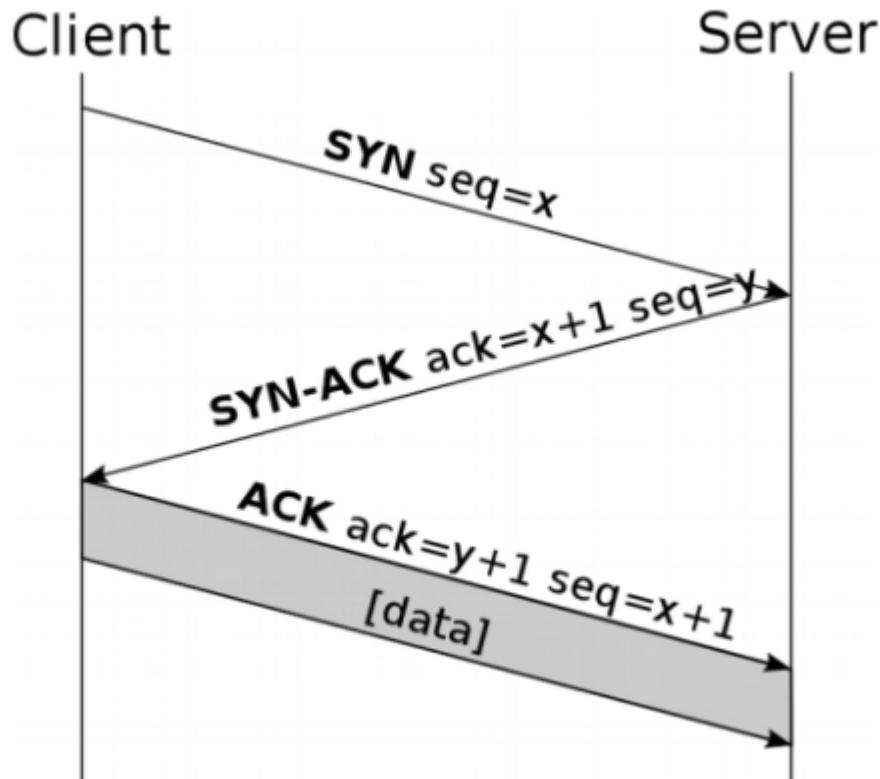
- La forma en que esto funciona es la siguiente:

Cuando el protocolo TCP inicia la conexión, este observará el flujo de datos de comunicación como un conjunto de bytes desordenados. Por lo que no va a ver la comunicación como un envío de paquetes sino como un envío de bytes.

- Por tanto, cada paquete va a tener en el campo "Número de secuencia" el número del primer byte del segmento dentro de todo el flujo de bytes de la comunicación.
- Esto facilita la identificación de los diferentes paquetes.
- El otro elemento es el que se coloca en el campo "Número de confirmación de recepción".

- Este número indica el número de secuencia del siguiente byte que está esperando el transmisor. Es decir, mediante el siguiente byte que espera se dice hasta dónde ha recibido el receptor la secuencia de bytes.
- La confirmación de recepción de paquetes se hace entonces mediante el campo "Número de confirmación de recepción" y el flag ACK.
- No podemos hablar de mensajes ACK como tal porque al ser una conexión full-duplex el ACK de los mensajes recibidos se transmite conjuntamente con un nuevo paquete de información, haciendo más eficiente el protocolo.
- Se puede dar la situación de que nos encontramos con un hueco. Es decir, haber recibido el mensaje n antes que el $n-1$. Esto nos causa un problema que no se especifica en la definición del protocolo. El receptor indicará hasta dónde tiene y el emisor tendrá que enviar la secuencia correspondiente, pero si el receptor tiene una secuencia posterior puede descartarla y esperar a que le llegue de una manera ordenarla o guardarla a la espera de que lleguen los bytes faltantes.
- Para el reenvío de paquetes perdidos TCP usa un temporizador, con un tiempo estimado de ida y vuelta del paquete para saber si se ha perdido. En caso afirmativo el paquete es reenviado.

Ejemplo de comunicación TCP



Ejemplo de comunicación TCP

En este ejemplo vemos cómo se envía un mensaje con inicio de secuencia en x . El otro punto de la conexión (server en este caso) envía un nuevo paquete con una secuencia que inicia en y , añadiendo la confirmación (ACK) del mensaje anterior, indicando que espera recibir el byte siguiente (indicado como $x+1$).

Protocolo HTTP

Es en esta parte del documento es donde se expondrán varias de las características del protocolo HTTP y su análisis.



Imagen de cabecera del protocolo HTTP, suele aparecer antes de la dirección DNS para indicar que se emplea ese protocolo.

HTTP hace referencia a las siglas en inglés de Hyper Text Transfer Protocol. Es un protocolo a nivel de aplicación. Transmite sobre el protocolo TCP y es empleado para la comunicación cliente-servidor donde se transfiere la información que requiere el navegador para mostrar una página por pantalla.

Aunque el ejemplo más sencillo sea el navegador también sirve a otro tipo de aplicaciones y programas.

Por lo general se emiten dos grupos de datos. El propio contenido (que no es necesariamente hypertexto) y los metadatos contenidos en la cabecera.

Lo primero que se nos viene a la cabeza con este protocolo es la petición de documentos HTML.

Fue diseñado a principios de los 90 y se caracteriza por ser en gran medida ampliable.

El protocolo HTTP opera usando una serie de mensajes formateados que depende de unas estructuras definidas.

En lugar de ser un flujo continuo de datos, en este caso, se emplean comunicaciones mediante **petición** y **respuesta**.

Una petición hace referencia al mensaje que envía el cliente mientras que los que son emitidos por el servidor se conocen como respuestas.

Los tipos de operaciones HTTP más comunes son:

- GET: es una solicitud de datos al servidor.

Indica que solicitamos un recurso como HTML, CSS o cualquier otro tipo de archivo.

- POST: es una solicitud para enviar una entidad a un recurso en específico.

Se emplea comúnmente en los formularios. Causa cambios en el estado.

- PUT: reemplaza las representaciones del recurso con la carga útil.

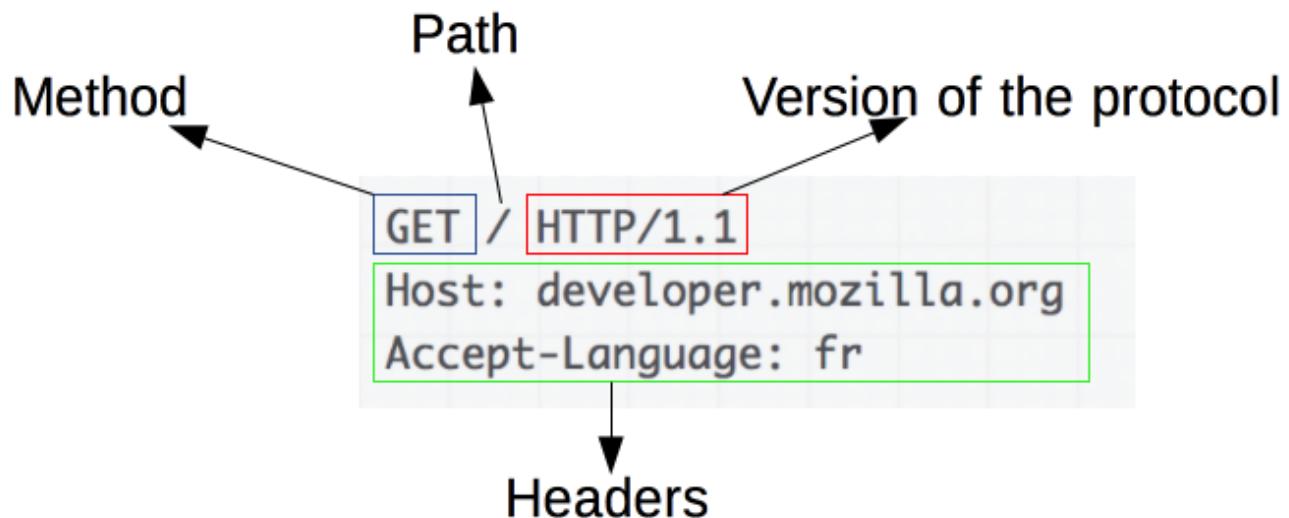
- DELETE: Borra el recurso especificado.

También se les conoce como peticiones HTTP, mensajes HTTP o métodos HTTP.

Se encuentra una lista más extensa y detallada en la [página de Mozilla](#)

Las peticiones tienen una estructura de mensaje distinta a la de las respuestas. Se componen por el tipo de petición, el path o ruta a la que se aplica, la versión del protocolo y luego la serie de cabeceras que componen los metadatos del mensaje. Esta última sección se reemplaza por el cuerpo del mensaje cuando la petición lo requiere (POST, PUT, etc.)

Tal y como se observa en la imagen siguiente:



Esquema descriptivo de una petición HTTP

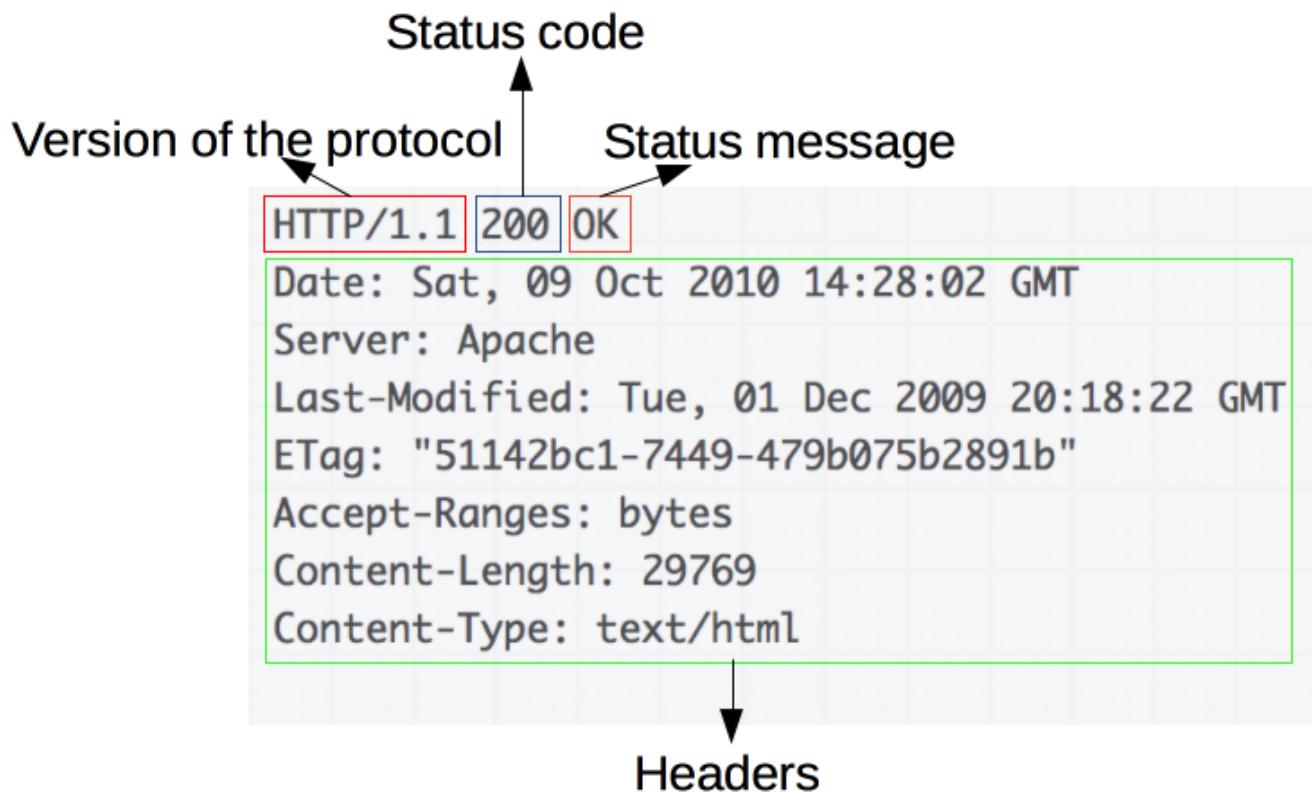
Por otro lado **las respuestas** HTTP son compuestas por la versión del protocolo, el código de estado, que es seguido por el mensaje de estado y los headers. Opcionalmente se añade el recurso que se ha perdido.

Los códigos de estado se resumen en:

- 1xx (Respuesta)
- 2xx (Éxito)
- 3xx (Redirección)
- 4xx (Error del cliente) (como el 404 -> página no encontrada)
- 5xx (Error del servidor) (como el 503 -> servicio no disponible)

todos estos códigos se especifican más aún en sus distintas variantes (101, 102, ...)

Se puede apreciar mejor la estructura de la respuesta en la siguiente imagen:



Esquema descriptivo de una respuesta HTTP

Todos estos aspectos pueden ser apreciados y analizados desde *Wireshark* a modo de paquetes. Sin embargo en el propio **navegador** también se puede hacer un **análisis de red** de la comunicación cliente-servidor mediante el protocolo HTTP.

Si hacemos hincapié en el proceso de la comunicación estudiado en la práctica, se puede apreciar que la primera petición realizada es de DNS.

Esto se da debido a que primero debemos solicitar un acceso al servidor DNS que proporciona una traducción del dominio solicitado.

La traza es la siguiente:

PC → PETICIÓN AL DOMINIO → PETICIÓN AL SERVIDOR DNS → RESPUESTA CON IP TRADUCIDA → ACCESO.

Cabe destacar que el navegador suele proporcionar una información bastante detallada. Durante la práctica se ha observado, sobre todo, en el análisis de los header de las peticiones que proporcionan información relevante a la versión HTTP empleada por el navegador o el lenguaje que usa, del mismo modo que también ilustra la dirección IP origen y la destino o el lenguaje que usa el sitio.

También la información puede indicar datos relativos a las peticiones ejecutadas así como su fecha de modificación o ejecución. Inclusive se puede ahondar en el estado recibido.

Muchas veces, también se puede apreciar como las peticiones son requeridas pero omitidas, más aún en el acceso a sitios web recientes. Esto se puede deber a que los archivos, HTML entre otros, ya han sido guardados en una caché. Esto se emplea para facilitar y agilizar la carga de datos.

Toda esta información se suele observar en esta herramienta del buscador:

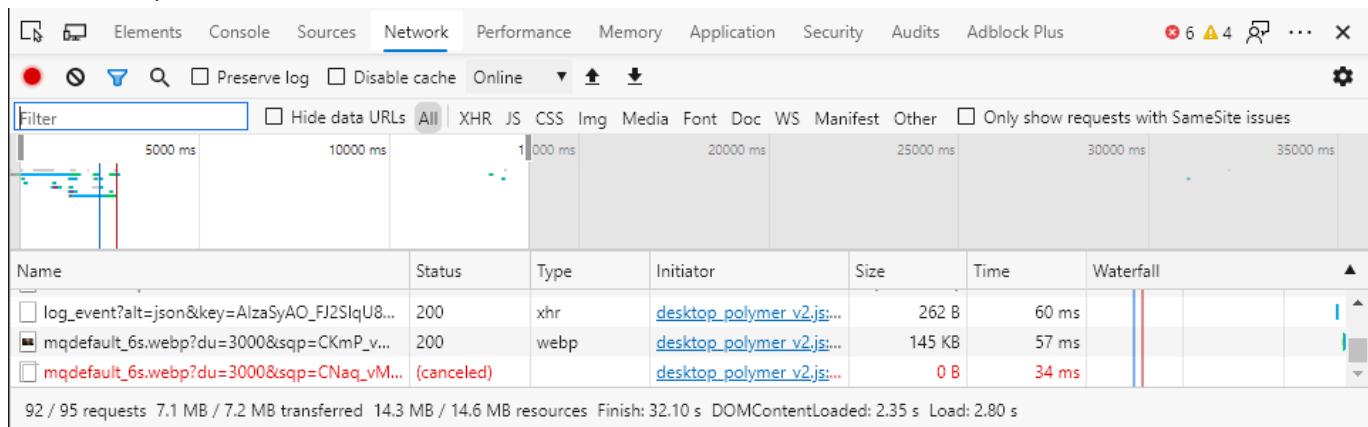


Imagen del apartado de análisis de red del navegador

Aspectos eticos de Wireshark

Este documento cuestiona las implicaciones éticas y morales del programa Wirehark, utilizado para la obtención de paquetes y el análisis de protocolos.



Logo de Wireshark

Reflexion etica

El uso de Wireshark merece una reflexión sobre su uso. A parte de su función méramente académica que nosotros realizamos, también podemos dar solución a problemas de redes, y otras funcionalidades más técnicas y precisas sobre comunicación de paquetes.

Como bien sabemos, lo que hacemos es recoger paquetes de información que circulan por la red, información que en varias ocasiones, son de otros usuarios, lo que nos indica que estamos invadiendo la privacidad de los demás, quizás de una forma no tan directa, pero al fin y al cabo, estamos accediendo a su información.

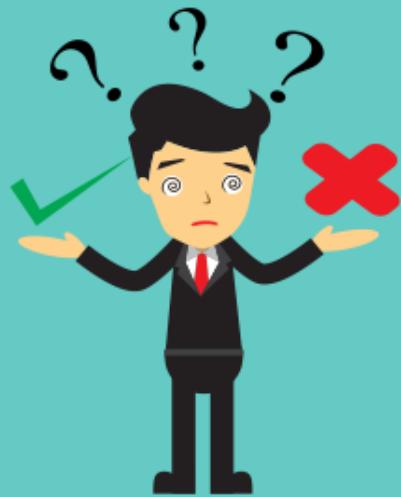
Aquí es donde surge el problema, ya que estamos accediendo a información, de la cual, no deberíamos tener permiso/acceso y que puede resultar sensible o una vulneración de la privacidad de los individuos.

- ¿Qué podemos hacer con esa información?

Está claro que con esa información se puede usar con muchos fines, algunos no tan legales como otros.

Generalmente, en internet, el uso de *Wireshark* **no es del todo lícito** en la mayoría de los casos.

Por norma general usar un programa de este estilo para algo que vaya más alla del contexto académico, cuestiones de mantenimiento, trabajo sobre las redes y el análisis de información de forma profesional es algo que puede causar muchos problemas y dificultades.



Representación de la ética

Cada persona es libre de hacer lo que crea conveniente y tendrá un criterio diferente sobre lo que es éticamente correcto hacer o no, de lo que se puede usar, escuchar, y de hasta dónde puede asomarse. Esto sin duda es inevitable, el juicio de cada uno es subjetivo, y no se comparte estrictamente con todo el mundo. Sin embargo, lo que sí debemos saber todos, es que somos completamente conscientes de lo que está ocurriendo, de las acciones que realizamos y podemos llegar a realizar con estos programas y, por su puesto, con la información que obtenemos gracias a su uso.

Evidencias y aportes del trabajo en grupo

Este documento pretende hacer evidencia de las conclusiones, situaciones y resultados que ha conllevado el trabajo en grupo a lo largo de las prácticas y el desarrollo del documento



En la imagen se muestran las herramientas empleadas para el trabajo colaborativo.

Cabe destacar que el trabajo en grupo puede ser un arma de doble filo, ya que este puede aportar tanto caos como orden; llevar menos peso de carga pero mantener la responsabilidad de coordinarse tiene sus implicaciones.

Es por ello que hemos decidido tomar la iniciativa de dar un paso de responsabilidad y semejar en la medida de lo posible nuestro modelo de trabajo colaborativo al empleado por muchas de las comunidades de desarrollo software.

El uso de **git** nos ha facilitado dividir el trabajo de manera que no nos solapemos y podamos llevar un consenso de formato y desarrollo del informe. Más aún a esto se le ha añadido el uso de GitHub para aprovecharnos de las herramientas y metodologías de trabajo colaborativo que este aporta.

Ya que empleamos esta plataforma, nos hemos decantado por el desarrollo del informe mediante el lenguaje Markdown. GitHub provee herramientas de visualización y despliegue gratuitas que creemos que podrían mejorar la calidad, la eficacia y la eficiencia del resultado final de cara a una entrega.

Al principio el desbalance entre ambas partes imponía una brecha organizativa que había que cubrir. Sin embargo, una vez formadas ambas partes del grupo nos pudimos dar el lujo de no necesitar hacer el trabajo a la vez para evitar perjuicios en el desarrollo.

La distribución de volumen de trabajo la hemos consensuado (Noah Sanchez y Eric Dürr) de manera que ninguno quedase aislado de la materia de prácticas, estudiando ambos todo lo relacionado a todos los protocolos. La manera que hemos hallado de suplir este hecho ha sido repartir primero las preguntas por cada protocolo entre ambos y a la hora de la redacción intercambiar la materia. Siendo así conscientes por ambas partes del contenido del temario.

Este volumen de trabajo se aprecia en los ".txt" que contienen las preguntas y los ".md" que contienen la redacción del informe.

Ambas partes nunca han sido vulneradas ni incomunicadas, lo cual es una ventaja que hemos podido destacar frente al caso común de los trabajos grupales. De manera constante los cambios y decisiones drásticas durante el desarrollo han sido discutidas y consensuadas antes de llevarse a cabo. Desde la elección de plataforma y método hasta los más sencillos comunicados de la finalización de un apartado.

Noah Sanchez se ha encargado de los documentos (en orden cronológico):

- eth-arp.txt

que responde las preguntas planteadas sobre Ethernet y ARP

- HTTP.txt

que responde cuestiones acerca del protocolo HTTP

- ip-icmp.md

que expone el informe de desarrollo de estos dos protocolos

- udp-tcp.md

que contiene el desarrollo del documento que redacta estos dos protocolos

- aspectos-eticos-wireshark.md

este contiene un ensayo y conclusiones acerca de los aspectos de ética de wireshark

Eric Dürr se ha encargado de los documentos (en orden cronológico):

- ip-icmp.txt

que responde las preguntas planteadas sobre IP e ICMP

- udp-tcp.txt

que responde cuestiones acerca de los protocolos UDP y TCP

- eth-erp.md

que expone el informe de desarrollo de los protocolos Ethernet y ARP

- http.md

que es el desarrollo del documento que redacta el protocolo HTTP y sus características

- evidencias-trabajo-grupo.md

que es el desarrollo de este mismo documento

Finalmente, como guiño al uso de las tecnologías de esta plataforma hemos decidido **construir un sitio web** para facilitar el acceso al informe y no necesitar de una descarga del mismo.

Por otro lado también hemos **recopilado en pdf** dicho informe tal y como dicta el caso común de elaboración de prácticas.

Esto ha aportado tres alternativas para acudir al trabajo realizado:

- [Archivos en pdf](#)
- [Un repositorio en GitHub](#)
- [Un sitio web](#)

A continuación se listan las referencias empleadas para cada protocolo

Etyhernet y ARP

- [CCNA desde cero](#)
- [Simento, Ethernet-IP](#)
- [Velius, Ethernet p1](#)
- [Ethernet en wikipedia](#)
- [CCM - protocolo ARP](#)
- [ARP en wikipedia](#)
- [UPV](#)
- [Apuntes de networking](#)
- [Network Sorcery - protocolo ARP](#)

IP e ICMP

- [Curso de Redes: Rogelio Montaña](#)
- [IP en Wikipedia](#)
- [ICMP en Wikipedia](#)
- [ICMP en Youtube](#)
- [El Procolo IP](#)

UDP y TCP

- [Curso de Redes: Rogelio Montaña](#)
- [UDP en Wikipedia](#)
- [TCP en Wikipedia](#)
- [ACK en Wikipedia](#)

HTTP

- [CCM - protocolo HTTP](#)
- [HTTP en Wikipedia](#)
- [Network Sorcery - protocolo ARP](#)
- [Generalidades del protocolo HTTP](#)