Proyectos de prácticas

Redes de Ordenadores
2º Ingeniería Biomédica

Curso 2022-23

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

©2022
Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
Algunos derechos reservados
Este trabajo se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de
Creative Commons disponible en
http://creativecommons.org/licenses/by-sa/4.0/deed.es

Contenido

- Práctica 1: Ethernet, Hub, Switch
- Práctica 2: Formato del datagrama IP y configuración de direcciones IP
- Práctica 3: Tablas de encaminamiento, IP, ARP, ICMP
- Práctica 4: traceroute
- Práctica 5: UDP, TCP
- Práctica 6: Domain Name System (DNS)
- Práctica 7: HTTP

Prácticas con NetGUI

Práctica 1: Ethernet, Hub, Switch

Redes de Ordenadores
2º Ingeniería Biomédica

Curso 2022-23

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

©2022
Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
Algunos derechos reservados
Este trabajo se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de
Creative Commons disponible en
http://creativecommons.org/licenses/by-sa/4.0/deed.es

Prácticas con NetGUI

Práctica 1: Ethernet, Hub, Switch

Resumen

En esta práctica se mostrará el encapsulamiento entre unidades de datos de diferentes protocolos dentro de la arquitectura TCP/IP. Se dedicará especial atención al funcionamiento de Ethernet. Además se aprenderá a realizar capturas de tráfico con la herramienta `tcpdump`, y a analizarlas con la herramienta `wireshark`. Para la creación de una red Ethernet se usarán hubs y switches, mostrando las diferencias en el comportamiento de cada uno de estos dispositivos.

IMPORTANTE: Toma nota de todo lo que hagas en un **cuaderno de laboratorio** en formato electrónico. En él debería constar lo que vas aprendiendo en cada apartado de la práctica, los pasos que has tenido que ir dando para obtener los resultados pedidos, los comandos que has empleado, las respuestas a las preguntas que se realizan en el enunciado, y cualquier otra información que consideres oportuna. Este cuaderno de laboratorio te será muy útil para repasar lo aprendido y tendrás que entregarlo en el plazo establecido.

1. Análisis de ficheros de captura de tráfico

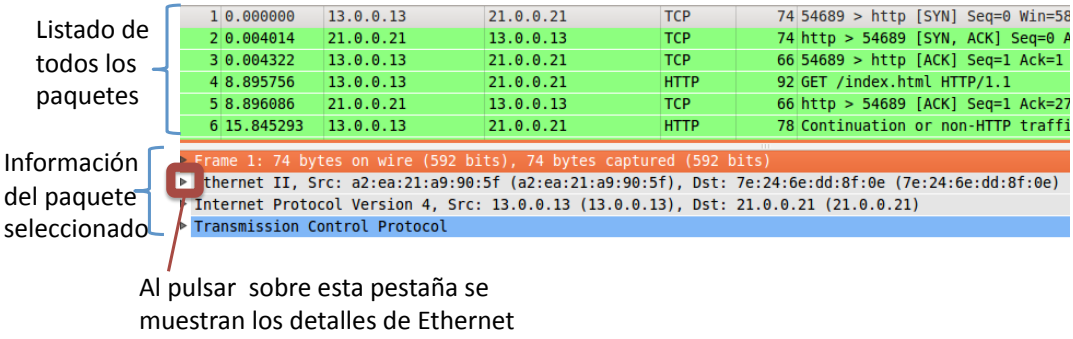
Abre el fichero de captura `cap1.cap` con `wireshark` y responde a las siguientes preguntas:

1. Selecciona el primer paquete que aparece en la captura, pulsando sobre la primera línea del Panel 1 (lista de paquetes). Éste quedará marcado en un color diferente:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	13.0.0.13	21.0.0.21	TCP	74	54689 > http [SYN] Seq=0 Win=5840 Len=0 MSS=146
2	0.004014	21.0.0.21	13.0.0.13	TCP	74	http > 54689 [SYN, ACK] Seq=0 Ack=1 Win=5792 Le
3	0.004322	13.0.0.13	21.0.0.21	TCP	66	54689 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0 T
4	8.895756	13.0.0.13	21.0.0.21	HTTP	92	GET /index.html HTTP/1.1
5	8.896086	21.0.0.21	13.0.0.13	TCP	66	http > 54689 [ACK] Seq=1 Ack=27 Win=5792 Len=0
6	15.845293	13.0.0.13	21.0.0.21	HTTP	78	Continuation or non-HTTP traffic

2. En el Panel 1 (lista de paquetes), para cada paquete se muestra:
   - Su número de orden dentro de la captura (columna `No.`). El número 1 es el primer paquete capturado.
   - Tiempo en segundos que ha pasado desde que se capturó el primer paquete (columna `Time`). El primer paquete marca el origen de tiempos, por lo que el valor de tiempo es 0.000000 segundos. El segundo paquete muestra 0.004014 segundos lo que significa que el segundo paquete se capturó transcurridos 0.004014 segundos desde que se capturó el primer paquete. Y así sucesivamente.
   - Dirección de origen del paquete (columna `Source`). En este caso muestra la dirección origen de nivel de red (dirección IP).
   - Dirección destino del paquete (columna `Destination`). En este caso muestra la dirección destino de nivel de red (dirección IP).
   - Protocolo de más alto nivel reconocido dentro del paquete (columna `Protocol`).
   - Longitud total de la trama capturada en bytes (columna `Length`), sin contar el campo CRC (4 bytes).
   - Resumen de la información más importantes contenida en los protocolos reconocidos en el paquete (columna `Info`).

   Con el primer paquete seleccionado, observa observa en el Panel 2 de `wireshark` los detalles de los protocolos para ese paquete. Indica qué protocolos se usan en ese primer paquete y a qué nivel de la arquitectura TCP/IP corresponden dichos protocolos.
3. Teniendo seleccionado el primer paquete de la captura, en la primera pestaña (`Frame`) del Panel 2 se muestra información estadísitica relativa a la captura de ese paquete. Es la única pestaña que no tiene información de ningún protocolo contenido en el paquete, y en general no necesitaremos consultar dicha pestaña.
4. El resto de pestañas del Panel 2 contiene las cabeceras de los protocolos reconocidos en el paquete, empezando por `Ethernet` y siguiendo con los protocolos de niveles superiores.
5. Teniendo seleccionado el primer paquete de la captura, despliega la pestaña que se corresponde con el protocolo Ethernet. Indica qué campos observas en la cabecera de Ethernet, comprueba que la longitud de estos campos se corresponde con lo que hemos visto en la parte de teoría. Apunta los valores de estos campos, para ello, selecciona la pestaña Ethernet dentro de Wireshark y con el botón derecho del ratón selecciona la entrada 'Copy all visible selected tree items' para que los campos de la cabecera Ethernet se queden copiados en el portapapeles y puedas pegarlos en el documento de la memoria.



6. Pulsa sobre el campo `Type` de la cabecera `Ethernet` y observa cómo en la zona del Panel 3 que muestra el contenido del paquete en hexadecimal, se colorea dicho valor. Observa que `wireshark` interpreta el valor de `Type` 0x0800 como el código asociado al protocolo IP. ¿Qué significa?
7. Observa que en las capturas no aparecen los bytes ni el preámbulo, ni el comienzo de trama . El hardware de la tarjeta Ethernet elimina estos campos, pues no forman parte propiamente de la trama Etherente. Observa que tampoco aparece el CRC: el hardware de la tarjeta Ethernet comprueba que es correcto y lo elimina también de la trama. Si no fuera correcto descartaría la trama y no aparecería en la captura.
8. Selecciona el segundo paquete y observa en el Panel 2 de `wireshark` los detalles de los protocolos para ese paquete. Indica qué protocolos se usan en ese segundo paquete y a qué nivel de la arquitectura TCP/IP corresponden dichos protocolos.
9. Con el segundo paquete seleccionado, despliega la pestaña que se corresponde con el protocolo Ethernet. Indica qué campos observas en la cabecera de Ethernet. A la vista de los valores de estos campos indica si crees que este segundo paquete lo envía la misma máquina que envía el primer paquete.
10. Fíjate en la longitud del primer paquete que aparece en su columna `Length` del Panel 1. Dicha longitud hace referencia a la longitud de toda la trama Ethernet sin el CRC. Para calcular la longitud de toda la trama Ethernet habría que sumar a la columna `Length` de una trama los 4 bytes del CRC que no aparecen en la trama capturada. ¿Crees que la primera trama lleva bits de relleno en Ethernet?
11. Si la columna `Length` de la trama tuviera un valor igual a 60 bytes (longitud total de la trama igual a 64 bytes: 60 más 4 bytes del CRC) ¿podrías decir si dicha trama tiene o no relleno?
12. Observa el paquete número 18. Indica qué protocolos se usan en ese paquete y a qué nivel de la arquitectura TCP/IP corresponden dichos protocolos.
13. Observa el campo longitud de la trama Ethernet asociada al paquete número 18. Si la máquina que está enviando esa información hubiese tenido más datos para enviar dentro de la trama 18, explica si hubiera podido incluirlos también en el campo de datos de dicha trama.

Cierra el fichero de captura `cap1.cap` y abre el fichero de captura `cap2.cap` con `wireshark` y responde a las siguientes preguntas:

1. Teniendo seleccionado el primer paquete de la captura, despliega la pestaña que se corresponde con el protocolo Ethernet. Indica qué campos observas en Ethernet. Apunta los valores de estos campos.
2. Fíjate en el campo `Type`. El valor es diferente al que viste en el fichero de captura anterior. ¿A qué protocolo se refiere este valor?
3. ¿Qué significa el valor del campo dirección destino Ethernet que aparece en ese primer paquete?
4. Fíjate en el campo longitud de la primera trama. ¿Cuánto es la longitud total de la trama contando el CRC?
5. En este caso, el paquete es un mensaje del protocolo ARP que va encapsulado dentro de Ethernet. Todos los mensajes del protocolo ARP tienen la misma longitud, 28 bytes. La cabecera de Ethernet ocupa 14 bytes y el CRC 4 bytes. Por tanto la longitud total de la trama sería 46 bytes y será necesario introducir relleno para alcanzar la longitud de trama mínima en Ethernet (64 bytes). El relleno debería ser 18 bytes.
6. Observa para este paquete el campo `Padding`. ¿Qué longitud tiene? ¿Qué crees que significa este campo?

2. Generación de tráfico Ethernet y análisis de la captura de tráfico

Descárgate de la siguiente página todos los escenarios de red, uno de estos ficheros será utilizado en este apartado. El resto de ficheros los utilizarás en apartados posteriores. Ten en cuenta que deberás introducir tu DNI para acceder a dicho escenario porque cada alumno partirá de una configuración diferente:

`http://mobiquo.gsyc.urjc.es/practicas/ro/p1.html`

Descomprime el fichero `p1-ethernet.tgz` usando el botón derecho (opción Extraer a).

Desde un terminal ejecuta: `netgui.sh`

Carga el escenario que acabas de descomprir: `File` → `Open` y navega por la ventana hasta la carpeta donde hayas dejado el escenario de NeGUI, de forma que en la casilla `Folder name` debe aparecer todo el trayecto hasta dicha carpeta y terminar con la palabra `p1-ethernet` que es la carpeta que contiene la configuración del escenario. Al pulsar sobre el botón `Open` deberías ver un dibujo similar al mostrado en la figura 1:

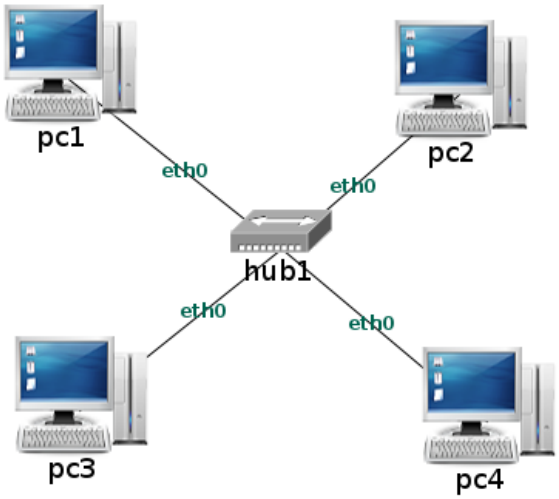


Figura 1: Escenario p1-ethernet.tgz

Arranca cada uno de los PCs, de uno en uno, esperando que termine de arrancar una máquina para arrancar la siguiente. Observarás que el icono de las máquinas aparece ahora con dos triángulos azules, que indican que las máquinas están ejecutándose. Al arrancar las máquinas se configuran con una dirección de nivel de red, una dirección IP. El protocolo IP será objeto de estudio del tema siguiente.

1. Consulta las direcciones Ethernet que hay configuradas en cada una de las interfaces de las máquinas, para ello ejecuta por ejemplo en `pc1`:

   ```
   pc1:~# ifconfig eth0
   ```

   Apunta las direcciones Ethernet de cada interfaz de todos los pcs.

2. Inicia una captura de tráfico en `pc3`. Para ello ejecuta los siguientes comandos.

   En `pc3`:

   ```
   pc3:~# tcpdump -i eth0 -s 0 -w /hosthome/p1-ethernet.cap
   ```

   Ahora vas a generar tráfico de la siguiente forma: `pc1` va a enviar una trama Ethernet a `pc2` y `pc2` va a responder. Para ello ejecuta en `pc1`, susituyendo previamente donde dice `<direcciónEthernetPc2>` por la dirección Ethernet de la máquina `pc2`:

   ```
   pc1:~# arping -c 1 <direcciónEthernetPc2>
   ```

   Donde:

   - La dirección Ethernet que estamos utilizando es la dirección Ethernet destinataria de las tramas, en este caso la de `pc2`.

- La opción `-c 1` hace que `arping` envíe un único paquete a la máquina `pc2` y que ésta le responda.

Interrumpe la captura pulsando `Ctrl+C` en la ventanas de `pc3`.

Analiza las tramas Ethernet que aparecen en la captura. Para cada paquete indica:

(a) Dirección Ethernet origen.
(b) Dirección Ethernet destino.
(c) ¿Qué crees que se hubiera capturado en las interfaces de `pc1(eth0)`, `pc2(eth0)`, `pc4(eth0)` si hubiéramos arrancado también `tcpdump` en dichas interfaces? ¿Por qué?
(d) Indica qué máquinas reciben la primera trama capturada y qué máquinas la procesan y se la entregan al protocolo de nivel superior.
(e) Indica qué máquinas reciben la segunda trama capturada y qué máquinas la procesan y se la entregan al protocolo de nivel superior.
(f) Si la primera trama llevara como dirección destino ff:ff:ff:ff:ff:ff indica qué máquinas recibirían dicha trama y qué máquinas se la entregarían al protocolo de nivel superior.

3. Dispositivos de interconexión: hub y switch

3.1. Diferencias en el comportamiento entre un hub y un switch

Descomprime el fichero `p1-hub-switch.tgz` y carga el escenario en NetGUI, similar al que se muestra en la figura 2. Arranca los pcs de uno en uno. Cuando todos los pcs estén arrancados, inicia el switch.

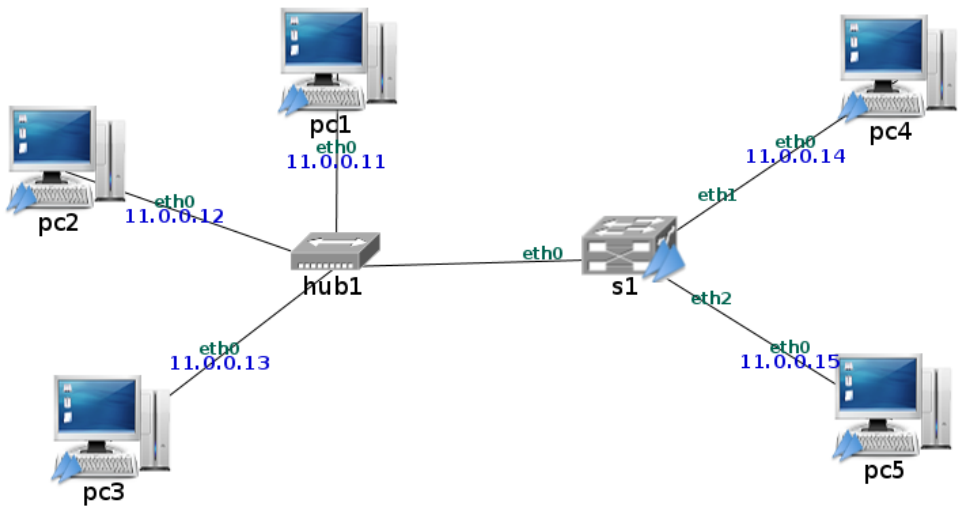


Figura 2: Escenario hub-switch.tgz

1. Indica cuál es la dirección Ethernet de `pc3`.
2. Comprueba que la tabla de direcciones Ethernet aprendidas en `s1` está vacía (sólo tiene información de las interfaces locales del switch).
3. ¿Qué crees que ocurrirá en el `hub1` cuando se ejecuta el comando `arping` desde `pc1` a `pc3`?
4. ¿Qué crees que ocurrirá en `s1` cuando se ejecuta el comando `arping` desde `pc1` a `pc3`?
5. Inicia una captura de tráfico en `pc2` y otra en `pc4` y guarda los paquetes capturados en 2 ficheros diferentes (`p1-switch1.cap` y `p1-switch2.cap` respectivamente). Ejecuta el comando `arping -c 3` desde `pc1` a `pc3` (la opción -c 3 hace que se envíen 3 paquetes de `pc1` a `pc3` y se reciba una respuesta por cada envío). Observa la tabla de direcciones Ethernet aprendidas en `s1` y explica su contenido.
6. Interrumpe las capturas de tráfico y explica justificadamente los mensajes capturados en `pc2` y `pc4`. Relaciona la información de las capturas con el contenido de la tabla de direcciones aprendidas por `s1`.
7. Borra la tabla de direcciones aprendidas en `s1`. ¿Qué crees que ocurrirá en `s1` cuando se ejecuta el comando `arping` desde `pc1` a `pc5`?

8. Comprueba tu suposición previa ejecutando ejecutando el comando `arping -c 3` en la máquina `pc1` dirigido a `pc5` y realizando las capturas de tráfico en `pc2` (en el fichero `p1-switch3.cap`), `pc4` (en el fichero `p1-switch4.cap`) y `pc5` (en el fichero `p1-switch5.cap`) que muestren el tráfico intercambiado. Explica el contenido de la tabla de direcciones aprendidas, interrumpe las capturas y explica las direcciones aprendidas en relación con el tráfico capturado.

9. Para ver cómo varía la tabla de direcciones aprendidas en un switch ejecuta el siguiente comando en `s1`:

   ```
   watch -n 1 brctl showmacs s1
   ```

   Este comando `watch` ejecuta cada segundo (`-n 1`) el comando `brctl showmacs s1`. Ejecuta `arping -c 5` desde `pc1` a `pc5`. Explica qué es lo que ves en la tabla de direcciones aprendidas. Indica en qué momento desaparecen las direcciones Ethernet de `pc1` y `pc5`. Cuando hayas terminado este apartado, puedes interrumpir el comando `watch` pulsando `Ctrl+c`.

10. Imagina que en un momento dado, la tabla de direcciones aprendidas del switch es similar a la siguiente (no se muestran las direcciones locales del propio switch):

    ```
    s1:~# brctl showmacs s1
    port no mac addr                is local?       ageing timer
      1     <dir_Ethernet_pc2>       no                225.60
      3     <dir_Ethernet_pc5>       no                225.60
    ```

    a) Indica qué tramas Ethernet habrá reenviado s1 para que esa tabla sea posible.

    b) ¿Qué ocurriría si el switch recibiera una trama Ethernet de `pc5` dirigida a `pc3`?

    c) ¿Qué ocurriría si el switch recibiera una trama Ethernet de `pc4` dirigida a `pc5`?

    d) ¿Cuánto tiempo falta para que el switch elimine de la tabla de direcciones aprendidas las direcciones de `pc2` y `pc5`?

11. Supón que `s1` recibe la siguiente trama Ethernet:

Dir Eth. Destino	Dir Eth. Origen	Protocolo	Contenido
ff:ff:ff:ff:ff:ff	`<dir_Ethernet_pc1>`	ARP	...

    Indica cómo se comportaría el switch en las siguientes situaciones:

    a) El switch `s1` tiene la tabla de direcciones aprendidas vacías.

    b) El switch `s1` tiene aprendida la dirección Ethernet de `pc1` en el puerto 1.

3.2. Influencia de los cambios de conexión física en la tabla de direcciones aprendidas de un switch

Descomprime el fichero `p1-hub-switch2.tgz` y carga el escenario en NetGUI, similar al que se muestra en la figura 3. Arranca las máquinas y el switch.

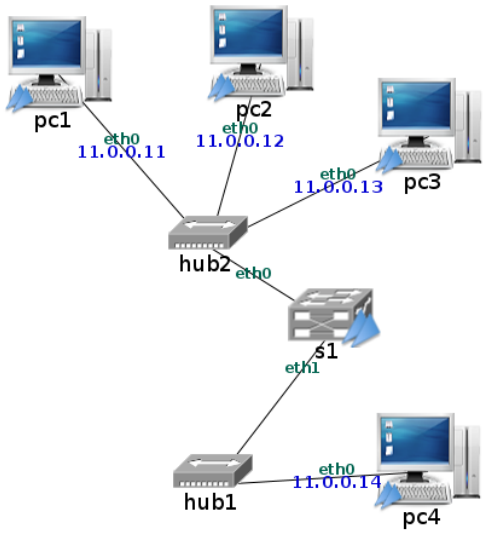


Figura 3: Escenario hub-switch2.tgz

1. Ejecuta el comando `arping` en `pc1` para que envíe un mensaje Ethernet `pc1` a `pc4`.
2. Consulta la tabla de direcciones aprendidas en `s1` y fíjate bien en los puertos donde `s1` ha aprendido.

3. Ejecuta el comando que te muestra la tabla de direcciones aprendidas cada segundo, para ver cómo va a variar su contenido.
4. Inicia una captura en `pc4` guarda su contenido en el fichero `p1-switch6.cap`
5. Antes de que caduquen las entradas de la tabla de direcciones aprendidas en `s1` (caducan a los 5 minutos), apaga `pc1` y borra el cable que le une al `hub2`. Crea un nuevo cable que una `pc1` y `hub1` y arranca `pc1`. Al arrancar `pc1` se generan automáticamente unos mensajes de autoconfiguración que provocan que el switch `s1` aprenda el nuevo sitio de `pc1`. Comprueba como se han modificado las entradas en la tabla de direcciones aprendidas en `s1` y observa el tiempo que muestra la tabla para la entrada de `pc1`.
6. Interrumpe la captura y observa cómo la máquina `pc1` ha generado mensajes al arrancar que han provocado la actualización de la tabla de direcciones aprendidas del switch.
7. ¿Qué crees que pasaría si `pc1` no hubiera generado ningún trafico automático al arrancar y en `pc3` se ejecutara `arping` hacia `pc1`?

4. Entrega de la práctica

Para entregar la práctica sube a aula virtual 2 ficheros:

- El fichero con la memoria de la práctica en formato pdf: las respuestas a las preguntas de la práctica.
- Las capturas `p1-ethernet.cap` y desde `p1-switch1.cap` a `p1-switch6.cap` dentro de un fichero comprimido: `p1.zip`. Para ello, primero crea una carpeta `p1` y mete dentro de esa carpeta todas los ficheros de captura. Desde el navegador de archivos pulsa con el botón derecho del ratón sobre el nombre de la carpeta y selecciona `'Comprimir'`, nombre del archivador `'p1'` y extensión `'.zip'`.

Práctica 2: Formato del datagrama IP y configuración de direcciones IP

Redes de Ordenadores
2º Ingeniería Biomédica

Curso 2022-23

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

©2022
Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
Algunos derechos reservados
Este trabajo se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de
Creative Commons disponible en
http://creativecommons.org/licenses/by-sa/4.0/deed.es

Práctica 2: Formato del datagrama IP y configuración de direcciones IP

Resumen

En esta práctica se aprende a configurar las interfaces de red de *hosts* y *routers* utilizando dos métodos distintos: interactivamente mediante el uso de los mandatos `ifconfig` o `ip`, y estáticamente utilizando ficheros de configuración. Además se estudiarán con detalle los campos de la cabecera IP.

IMPORTANTE: En el apartado 2 de la práctica se mencionan direcciones IP con una X entre medias (ej.: 151.X.0.1). Cada alumno debe utilizar como valor de X el que aparezca al introducir su DNI en el enlace:

`http://mobiquo.gsyc.es/practicas/ro/p2.html`

1. Campos de la cabecera IP

Carga en wireshark el fichero `cap1.cap`.

Selecciona el primer y único paquete y despliega los campos de la cabecera IP, en la zona donde se muestran los detalles de los protocolos para el paquete que está seleccionado.

Responde a las siguientes preguntas:

1. ¿Cuál es la dirección IP origen y la dirección IP destino del paquete?
2. ¿Crees que las máquinas que se están comunicando son vecinas y se están comunicando directamente o crees que lo hacen a través de uno o más *routers*?
3. Indica el valor del campo TTL.
4. Sabiendo que la captura de tráfico se ha realizado en la máquina destinataria del paquete y que inicialmente el paquete lo envío la máquina origen con TTL=64, indica cuántos *routers* intermedios ha atravesado dicho paquete.

2. Configuración de direcciones IP

2.1. El comando `ifconfig/ip`

- Arranca NetGUI. En las aulas de prácticas, la forma de arrancarlo es ejecutando en una ventana de terminal la orden `netgui.sh`.
- Crea una red como la de la figura 1 donde `pc1`, `pc2` y `pc3` son tres ordenadores y `r1` es un *router*.
  Coloca un *hub* para conectar `pc1`, `pc2` y `r1` y otro *hub* para conectar `pc3` y `r1`.
  Es importante que tengas en cuenta que cada vez que dibujas un cable desde un PC o un *router* hacia un *hub* se crea una interfaz Ethernet `ethX`, siendo X un número. Estas interfaces se numeran siguiendo el orden en el que se hayan dibujado sus cables, comenzando por `eth0`. Por eso, observa que para reproducir el mismo diagrama de la figura 1 deberás dibujar primero el cable desde `r1` al *hub1* para que esta interfaz se genere con el primer identificador `eth0`.
- Guarda la configuración de la red con Archivo → Guardar. Elige como nombre `p2-ifconfig`, **sin espacios**.
- Arranca los ordenadores y el encaminador de uno en uno. **Espera a que una máquina termine completamente de arrancar antes de arrancar la siguiente**. Los *hubs* en NetGUI son elementos pasivos que no hay que arrancar.

1. Comprueba la configuración de la red en cada una de las máquinas y en el encaminador mediante el comando `ifconfig`. ¿Qué interfaces de red tienen configuradas cada una de ellas, y qué dirección IP tiene configurada cada interfaz?
2. Utilizando la orden `ifconfig` o la orden `ip`, asigna las direcciones IP a las interfaces de red de las máquinas y el router de la siguiente forma:
   - Como `netmask` usa en todos los casos `255.255.255.0`.
   - A todas las interfaces conectadas al `hub1` asígnales una dirección que empiece por `151.X.0...`[1]
   - A todas las interfaces conectadas al `hub2` asígnales una dirección que empiece por `152.X.0...`

   NOTA: Ten en cuenta que `r1`, al estar conectado a dos *hubs*, tendrá dos direcciones IP, una para cada interfaz (`eth0` y `eth1`).

[1] Recuerda que debes sustituir la X por el número que aparezca para ti en `http://mobiquo.gsyc.es/practicas/ro/p2.html`

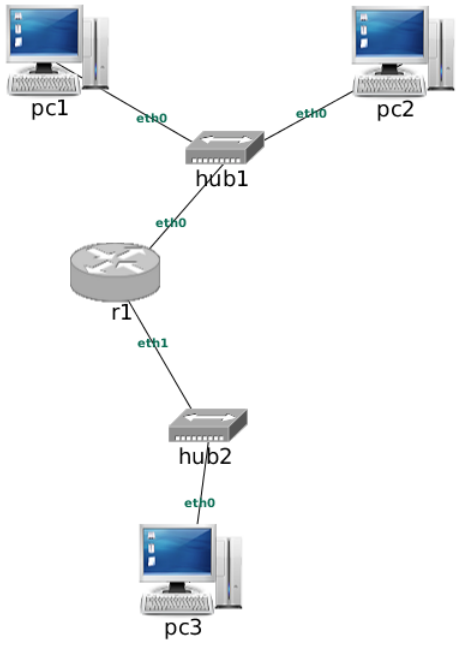


Figura 1: Red formada por tres PCs y un *router*.

3. Observa que las direcciones IP que has configurado se muestran en la interfaz de NetGUI. Comprueba en cada máquina virtual las direcciones de sus interfaces mediante `ifconfig` o `ip`. Incluye en la memoria de la práctica una imagen del escenario de NetGUI que muestre las direcciones IP que has configurado.
4. Inicia una captura de tráfico en `pc2`. Para ello ejecuta en `pc2`:

   ```
   pc2:~# tcpdump -i eth0 -s 0 -w /hosthome/p2-1.cap
   ```

   Ahora vas a generar tráfico de la siguiente forma: `pc1` va a enviar paquetes a `pc2` y `pc2` va a responder. Para ello ejecuta en `pc1`:

   ```
   pc1:~# ping -c 1 151.X.0.Y
   ```

   Donde:

   - La dirección IP que tienes que utilizar es la dirección IP destinataria de los paquetes, en este caso la de `pc2` (escribe en lugar de la X y la Y los valores de la dirección IP de la máquina `pc2` de tu escenario).
   - La opción `-c 1` hace que `ping` envíe un único paquete a la máquina `pc2` y que ésta le responda.

   Interrumpe la captura pulsando `Ctrl+C` en la ventana de `pc2`.

   Analiza los paquetes que aparecen en la captura. Para cada paquete indica:

   - Dirección Ethernet origen.
   - Dirección Ethernet destino.
   - Tipo de protocolo encapsulado (campo `Type`). Si el tipo de protocolo es IP, indica también:
     - Dirección IP origen
     - Dirección IP destino
5. Apaga el router `r1` y una vez apagado vuelve a arrancarlo. Comprueba que ha desparecido su configuración de direcciones IP.

2.2. El fichero /etc/network/interfaces

- Arranca NetGUI y construye una red como la de la figura 2. **Ten cuidado con el orden en que dibujas los cables de red de los *routers* a los *hubs***. Recuerda que para que las interfaces se ordenen en tu dibujo de la misma forma que en la figura, en los *routers* tienes que dibujar primero el cable que en la figura aparece etiquetado como `eth0`, después el que aparece etiquetado como `eth1`, y así sucesivamente.
- Guarda la configuración de la red con Archivo → Guardar. Elige como nombre `p2-interfaces`, **sin espacios**.

1. ¿Cuántas redes distintas (grupos de interfaces que son vecinas o adyacentes entre sí) crees que hay en la figura?
2. Arranca las máquinas de una en una. Comprueba que sus interfaces de red no están configuradas ejecutando `ifconfig`.

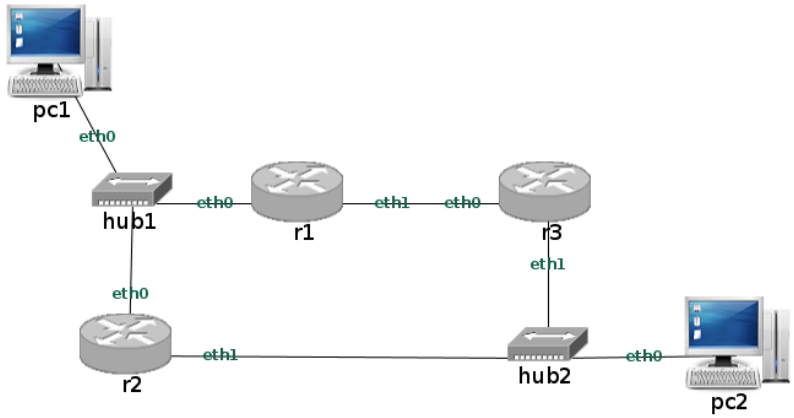


Figura 2: Red formada por 2 pcs y 3 *routers*

3. Edita el fichero `/etc/network/interfaces` de cada máquina y añade direcciones IP de la siguiente forma:
   - Como `netmask` usa en todos los casos `255.255.255.0`.
   - A todas las interfaces conectadas a una de las redes asígnales una dirección que empiece por `201.X.0` ...
   - A todas las interfaces conectadas a otra de las redes asígnales una dirección que empiece por `202.X.0` ... dirección que empiece por `203.X.0` ...
4. Ejecuta en cada una de las máquinas la orden necesaria para que se configuren las interfaces de red según lo que has escrito en el fichero de configuración. Comprueba que las interfaces están configuradas, utilizando para ello `ifconfig`. Observa que las direcciones IP que has configurado se muestran también en la interfaz de NetGUI. Incluye en la memoria de la práctica una imagen del escenario de NetGUI que muestre las direcciones IP que has configurado.
5. Ejecuta en `r1` la orden necesaria para desactivar la configuración de la red. Comprueba con `ifconfig` cómo se ha perdido la configuración de las interfaces de red en `r1`.
6. Vuelve a ejecutar la orden necesaria para activar la configuración la red y que se configuren las interfaces de red en función de lo especificado en `/etc/network/interfaces`.
7. Modifica la dirección IP de `r3(eth1)` en el fichero `/etc/network/interfaces` de `r3` para asignarle otra dirección IP diferente a la que ya habías asignado, teniendo en cuenta que debería pertenecer a la misma subred que antes. No olvides ejecutar el comando para reactivar la configuración de la red cada vez que modifiques el fichero `/etc/network/interfaces`.
8. Los cambios que has hecho en el fichero `/etc/network/interfaces` permanecerán si rearrancas las máquinas. Compruébalo apagando `r1` y volviendo a arrancarlo. Ejecuta `ifconfig` una vez que haya rearrancado y comprueba cómo las dos interfaces de `r1` están configuradas.
9. Inicia una captura de tráfico en `r2`, interfaz `eth0`. Para ello ejecuta en `r2`:

   ```
   r2:~# tcpdump -i eth0 -s 0 -w /hosthome/p2-2.cap
   ```

   Ahora vas a generar tráfico de la siguiente forma: `pc1` va a enviar paquetes a `r1` y `r1` va a responder. Para ello ejecuta en `pc1`:

   ```
   pc1:~# ping -c 2 A.B.C.D
   ```

   Donde:
   - La dirección IP que tienes que utilizar es la dirección IP destinataria de los paquetes, en este caso la de la interfaz `eth0` de `r1` (escribe en A.B.C.D los valores de la dirección IP de `r1-eth0` de tu escenario).
   - La opción `-c 2` hace que `ping` envíe 2 paquetes a la máquina `r1` y que ésta le responda a cada uno de ellos.

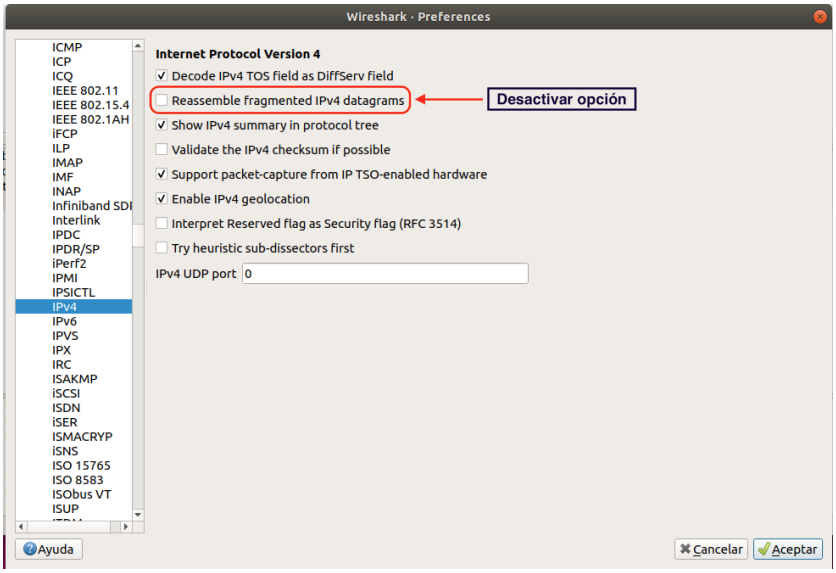
   Interrumpe la captura pulsando `Ctrl+C` en la ventana de `r2`.

   Analiza los paquetes que aparecen en la captura. Para cada paquete indica:
   - Dirección Ethernet origen.
   - Dirección Ethernet destino.
   - Tipo de protocolo encapsulado (campo `Type`). Si el tipo de protocolo es IP, indica también:
     - Dirección IP origen
     - Dirección IP destino
10. Prueba ahora (sin capturar el tráfico) a realizar el ping desde `pc1` a la dirección IP de la interfaz `eth1` de `r1`. ¿Qué ocurre? ¿A qué crees que se puede deber?

3. Fragmentación IP

Carga en wireshark el fichero `cap2.cap`.

Abre el menú `Edit` → `Preferences`, despliega la sección `Protocols` y busca el protocolo `IPv4`. Desactiva la opción señalada en la figura:



La captura muestra 3 paquetes que son 3 fragmentos de un datagrama IP original. Responde a las siguientes preguntas:

1. ¿Cómo se puede saber que los 3 paquetes pertenecen al mismo datagrama original?
2. Indica cuántos datos IP (cantidad de bytes de datos del campo de datos del datagrama IP original) viajan en cada uno de los datagramas en los que se ha fragmentado el datagrama original. ¿El primer y segundo datagrama IP podrían llevar más datos IP? ¿Por qué?
3. Indica cuántos datos IP formarían el datagrama IP original sin fragmentar.
4. Dado que los datagramas IP podrían desordenarse en el camino, indica cómo podría el destino reordenar los fragmentos y reconstruir el datagrama original.
5. Explica cómo puede saberse que el primer paquete de la captura es el primer fragmento de un datagrama fragmentado, en vez de ser un datagrama normal sin fragmentar. Observa que wireshark no muestra en este primer paquete nada que haga pensar que es un fragmento (a diferencia de lo que ocurre en los otros).
6. Explica cómo puede saberse que el último paquete de la captura es el último fragmento de un datagrama fragmentado, en vez de ser un datagrama normal sin fragmentar.
7. Activa ahora la opción de wireshark que desactivaste antes. Observa cómo cambia la información mostrada para los paquetes:
   - Ahora wireshark identifica al primer paquete como fragmento, y también el segundo, pero no lo marca en el último
   - wireshark señala en el primer y segundo paquete que ha reensamblado los 3 fragmentos en el tercer paquete
   - En el tercer paquete se mantiene la cabecera tal y como es, pero se detalla (al final de la cabecera IP) los campos de los 3 fragmentos: `[3 IPv4 Fragments (4008 bytes): #1(1480), #2(1480), #3(1048)]`
   - En el tercer paquete se incluye en la parte de datos los datos agregados de los 3 fragmentos.

4. Entrega de la práctica

Guarda los ficheros de captura en una carpeta que se llame `p2` que contenga `p2-1.cap` y `p2-2.cap`. Sube al enlace que encontrarás en Aula Virtual, y antes de que termine el plazo de entrega, un fichero de nombre `p2.zip` resultado de comprimir la carpeta `p2` y otro fichero diferente con la memoria en formato pdf:

- Memoria
- `p2.zip`: carpeta p2 y dentro los ficheros de captura

Práctica 3: Tablas de encaminamiento IP, ARP, ICMP

Redes de Ordenadores
2º Ingeniería Biomédica

Curso 2022-23

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

©2022
Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
Algunos derechos reservados
Este trabajo se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de
Creative Commons disponible en
http://creativecommons.org/licenses/by-sa/4.0/deed.es

Práctica 3: Tablas de encaminamiento IP, ARP, ICMP

Resumen

En esta práctica se aprende a configurar las las tablas de encaminamiento de las máquinas utilizando dos métodos distintos: interactivamente mediante el uso del mandato `route` y estáticamente utilizando ficheros de configuración.

IMPORTANTE: En las figuras se muestra un escenario de red en que aparecen direcciones IP con una X entre medias (ej.: 200.X.0.40). Cada alumno tendrá en su escenario unas IPs con un valor concreto de X.

Introducción

Descarga tu escenario introduciendo tu DNI en el enlace:

`http://mobiquo.gsyc.es/practicas/ro/p3.html`

Obtendrás un fichero llamado `p3-lab.tgz`, que contiene un escenario de red. Si al pulsar sobre el enlace aparece una ventana de diálogo, elige "Guardar archivo". Guárdalo, por ejemplo, en la carpeta de Descargas. Desde el navegador de archivos pulsa con el botón derecho del ratón sobre el fichero y selecciona la opción "Extraer aquí". Esta acción creará una carpeta con el nombre `p3-lab`, en la cuál estarán los ficheros de configuración de tu escenario.

Entre los ficheros del escenario se incluye el *script* `reset-lab`, que devuelve el escenario a su estado inicial cuando se ejecuta. Esto puede resultar útil durante la realización de la práctica. Para ejecutar el *script* hay que abrir un terminal de la máquina real y estar dentro de la carpeta que contiene el escenario, desde allí escribir:

```
./reset-lab
```

Si se desea simplemente devolver algunas máquinas a su estado inicial, pero no todas, es decir, si por ejemplo se desea devolver al estado inicial solo pc1 y r1, se escribirá:

```
./reset-lab pc1 r1
```

Tras descomprimir el escenario éste se encuentra en su estado inicial, por lo que no es necesario ejecutar `reset-lab` al principio.

NOTA: Para realizar esta práctica tendrás que consultar la documentación adicional sobre los comandos para modificar la tabla de *routing*, y sobre los comandos `arp`, `ping` y `traceroute`.

1. Configuración de tablas de encaminamiento con `route`

Lanza ahora NetGUI. En el menú, elige File → Open y selecciona la carpeta `p3-lab` en la que está el escenario. Verás aparecer la red de la figura 1.

Arranca únicamente las siguientes máquinas: `pc1`, `pc4`, `r3` y `pc2`.

Este escenario aplica una configuración asignando direcciones IP a todas las interfaces de las máquinas, excepto a `pc1`. Esta configuración inicial está almacenada en el fichero `/etc/network/interfaces` de cada una de las máquinas, tal y como se ha visto en la práctica anterior.

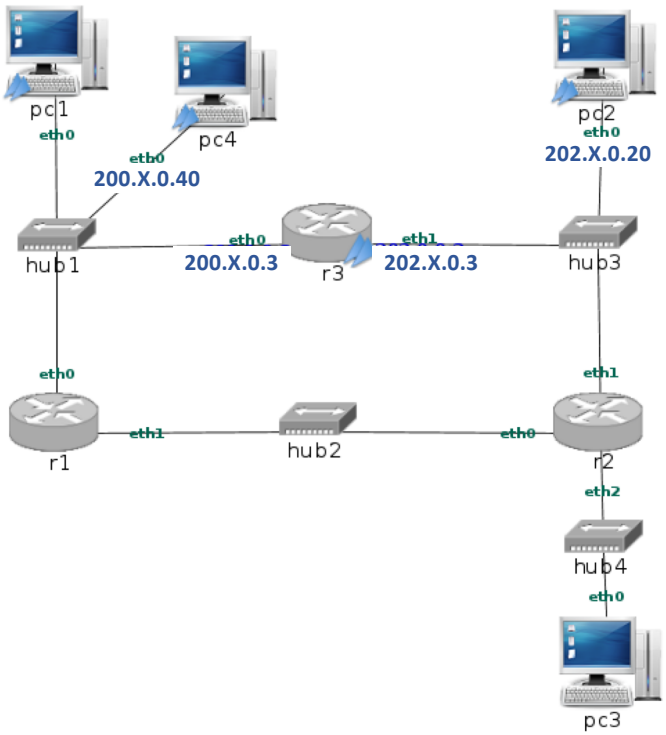


Figura 1: Sólo se arrancan: pc1, pc2, pc4 y r3

Teniendo en cuenta que sólo están estas máquinas arrancadas, responde a las siguientes cuestiones:

1. Escribe en `pc1` la orden `ping 127.0.0.1`. ¿Por qué se obtiene respuesta pese a no tener configurada `pc1` una dirección IP? Utilizando `route` comprueba el contenido actual de la tabla de encaminamiento (*routing*) de `pc1`.
2. Modifica el fichero `/etc/network/interfaces` de `pc1` para que tenga una dirección IP acorde a la de la subred a la que está conectado. (Nota: Deberás consultar previamente la máscara de la subred que tienen las otras máquinas conectadas a la misma subred que `pc1`). Reinicia la red en `pc1` para que se aplique la configuración que has escrito en el fichero `/etc/network/interfaces`.
3. Comprueba con `route` el contenido actual de su table de encaminamiento. Verás cómo, tras asignar la dirección IP a su interfaz de red, se ha añadido automáticamente una entrada en la tabla. Con esta tabla actual `pc1`, ¿a qué otras direcciones IP crees que `pc1` podrá enviar datagramas IP?
4. Dado que el resto de las máquinas encendidas tienen ya configurada una dirección IP, podrás suponer fácilmente cuál es el contenido de su tabla de encaminamiento. Mira la tabla de encaminamiento de `pc2` y `pc4`. ¿A que otras direcciones IP crees que esas máquinas podrán enviar datagramas IP?
5. Intenta deducir cuál crees que será la tabla de encaminamiento de `r3`,dado que tiene dos interfaces con IP asignada. Compruébalo consultando su tabla. Con esta tabla de encaminamiento, ¿crees que `r3` puede enviar datagramas IP a `pc1` y `pc4`? ¿Y a `pc2`.
6. Haz `ping` desde `pc1` a `pc4` y haz `ping` desde `pc1` a la dirección `r3(eth0)`. Ten en cuenta que no puedes utilizar los nombres `pc1`, `pc4`, etc. en el `ping`, sino que debes usar las direcciones IP correspondientes. ¿Funcionan estos `ping`? ¿Qué entradas de las tablas de encaminamiento se consultan en cada caso?
7. Haz un `ping` de `pc1` a `pc2` y haz un `ping` de `pc1` a la dirección `r3(eth1)`. ¿Funcionan estos `ping`? ¿Por qué?
8. Añade una ruta con el comando `route` en `pc1` para que los datagramas IP que no sean para su propia subred los envíe a través del *router* `r3`.

9. Haz ahora `ping` desde `pc1` a `r3(eth1)`. ¿Funciona este `ping`? ¿Qué entradas de las tablas de encaminamiento se consultan?
10. Haz un `ping` de `pc1` a `pc2`. ¿Por qué crees que no funciona este `ping`?
11. Añade las rutas necesarias utilizando el comado `route` para que funcione un `ping` de `pc1` a `pc2` y un `ping` de `pc4` a `pc2`. Ten en cuenta que podrás utilizar, rutas de máquina, rutas de subred o ruta por defecto.
12. ¿Crees que con la configuración que has realizado funcionará un `ping` de `pc2` a `pc1` y un `ping` `pc2` a `pc4`. Compruébalo.
13. Antes de continuar espera unos 10 minutos después de haber ejecutado el último `ping` del apartado anterior. Consulta el estado de las cachés de ARP en los pcs y en el *router* hasta que estén vacías.

    Arranca en `pc4` un `tcpdump` para capturar tráfico en su interfaz `eth0`, guardando la captura en el fichero `p3-a-01.cap`.

    Ejecuta en `pc1` un `ping` a `pc4` que envíe sólo 1 paquete (`ping -c 1 <máquinaDestino>`).

    Interrumpe la captura en `pc4` (`Ctrl+C`).

    Comprueba el estado de las cachés de ARP en `pc1`, `pc4`, `pc2` y `r3` y explica su contenido.
14. Analiza la captura con `wireshark`. Observa los siguientes campos en los mensajes:
    - Mensaje de solicitud de ARP que envía `pc1` a `pc4`.
      - Dirección Ethernet destino
      - Dirección Ethernet origen
      - Tipo en la cabecera Ethernet
      - Contenido del mensaje de solicitud de ARP: localiza el campo que contiene la dirección IP de la máquina sobre la que se está preguntando su dirección Ethernet.
    - Mensaje de respuesta de ARP que envía `pc4` a `pc1`.
      - Dirección Ethernet destino
      - Dirección Ethernet origen
      - Tipo en la cabecera Ethernet
      - Contenido del mensaje de respuesta de ARP: localiza el campo que contiene la dirección Ethernet solicitada.
    - Datagrama IP que envía `pc1` a `pc4`.
      - Dirección Ethernet destino
      - Dirección Ethernet origen
      - Tipo en la cabecera Ethernet
      - Dirección IP origen
      - Dirección IP destino
      - Campo TTL
    - Datagrama IP que envía `pc4` a `pc1`.
      - Dirección Ethernet destino
      - Dirección Ethernet origen
      - Tipo en la cabecera Ethernet
      - Dirección IP origen
      - Dirección IP destino
      - Campo TTL

    NOTA: Si tu captura ha durado más de 5 segundos, podrás ver que, 5 segundos después de la solictud de ARP de `pc1` preguntado por `pc4`, `pc4` hace una solicitud de ARP preguntando por `pc1`, pero esta solicitud de ARP no tiene dirección de destino broadcast, sino la dirección Ethernet de `pc1`. Esto demuestra que `pc4` ya tenía en su caché de ARP la dirección Ethernet de `pc1` (la aprendió de la solicitud recibida de `pc1`), y este ARP es simplemente de confirmación. No todas las implementaciones de TCP/IP realizan esta confirmación, y en esta asignatura nunca le prestaremos atención ni preguntaremos sobre ella.
15. Espera a que la caché de ARP de `pc1` esté vacía. Ahora vamos a analizar el tráfico desde `pc1` a `pc2`. ¿Cuántas capturas de tráfico crees que son necesarias para ver todos los paquetes que se generan en el escenario cuando se comunican `pc1` y `pc2`?

    Arranca un `tcpdump` en `r3(eth0)` guardando la captura en el fichero `p3-a-02.cap`.

    Arranca otro `tcpdump` en `pc2` guardando la captura en el fichero `p3-a-03.cap`.

    Ejecuta en `pc1` un `ping` a `pc2` que envíe sólo 1 paquete (`ping -c 1 <máquinaDestino>`).

    Interrumpe las capturas (`Ctrl+C`).

    Comprueba el estado de las cachés de ARP en `pc1`, `pc2`, `pc4` y `r3`. Explica su contenido.
16. Analiza las capturas realizadas con `wireshark`. Observa en los mensajes los mismo campos que analizaste el apartado anterior. Presta especial atención a los datagramas IP. Identifica los mensajes en las dos capturas que contienen el mismo datagrama IP. ¿Qué direcciones Ethernet tiene la trama que contiene esos datagramas en cada captura? ¿Qué valor tiene el campo TTL de la cabecera IP en cada uno?

2. Configuración de tablas de encaminamiento mediante ficheros de configuración

Arranca el resto de las máquinas y routers de la figura y obtendrás un diagrama similar a la figura 2. Ten en cuenta que en `pc1` ya tendrás configurada una dirección IP, mantén esta configuración.

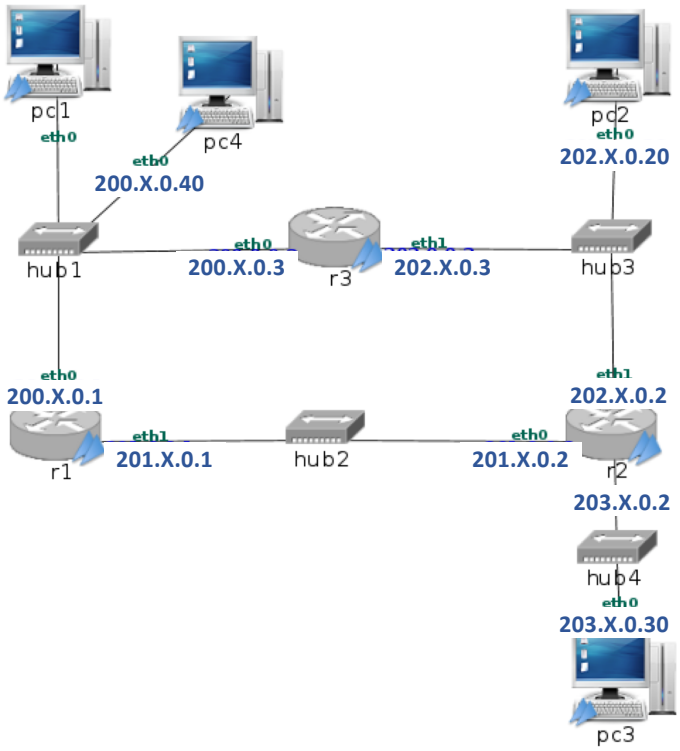


Figura 2: Todas las máquinas arrancadas

1. ¿Cuántas subredes observas en la figura? Escribe la dirección de cada una de estas subredes junto con su máscara.
2. Reinicia las máquinas `pc1`, `pc2` y `pc4`.

   Consulta las tablas de encaminamiento en todas las máquinas y *routers*, comprobarás que las rutas que configuraste en el apartado anterior han desaparecido. Los pcs y *routers* sólo tienen ruta hacia las subredes a las que están directamente conectados. Por tanto sólo podrán enviar paquetes a sus máquinas vecinas.
3. Añade rutas en las máquinas adecuadas modificando su fichero `/etc/network/interfaces` de forma que funcionen las siguientes rutas:

   a) Conectividad entre `pc1` y `pc2` en los dos sentidos, a través de las siguientes rutas:
   - pc1⇒r3⇒pc2
   - pc2⇒r3⇒pc1

   NOTA: Puedes emplear rutas de máquina, rutas de red o rutas por defecto.

   Ejecuta en `pc1` la orden `ping -c 3 <dirIPpc2>` para comprobar si hay conectividad entre las máquinas. Para verificar que el camino es el descrito previamente, deberás ejecutar `route` en `pc1` y observar qué entrada de la tabla de encaminamiento se utiliza cuando se desea alcanzar `pc2`. Esta entrada debería indicar que el siguiente salto es `r3`. A continuación en `r3` deberás ejecutar `route` y observar qué entrada de la tabla de encaminamiento se utiliza cuando se desea alcanzar `pc2`. Esta entrada debería indicar que `r3` no necesita ningún router adicional para alcanzar `pc2`.

   De forma análoga deberás consultar las tablas de encaminamiento en el sentido pc2⇒r3⇒pc1.

   b) Conectividad entre `pc2` y `pc3` en los dos sentidos, a través de la siguientes rutas:
   - pc2⇒r2⇒pc3
   - pc3⇒r2⇒pc2

   Ejecuta en `pc2` la orden `ping -c 3 <dirIPpc3>` para comprobar si hay conectividad entre las máquinas. Para verificar que el camino es el descrito previamente, deberás ejecutar `route` en `pc2` y observar qué entrada de la tabla de encaminamiento se utiliza cuando se desea alcanzar `pc3`. Esta entrada debería

indicar que el siguiente salto es `r2`. A continuación en `r2` deberás ejecutar `route` y observar qué entrada de la tabla de encaminamiento se utiliza cuando se desea alcanzar `pc3`. Esta entrada debería indicar que `r2` no necesita ningún router adicional para alcanzar `pc3`.

De forma análoga deberás consultar las tablas de encaminamiento en el sentido pc3⇒r2⇒pc2.

c) Conectividad entre `pc1` y `pc3` en los dos sentidos, a través de las siguientes rutas:
- pc1⇒r1⇒r2⇒pc3
- pc3⇒r2⇒r3⇒pc1

Ejecuta en `pc1` la orden `ping -c 3 <dirIPpc3>` para comprobar si hay conectividad entre las máquinas. Para verificar que el camino es el descrito previamente, deberás ejecutar `route` en `pc1` y observar qué entrada de la tabla de encaminamiento se utiliza cuando se desea alcanzar `pc3`. Esta entrada debería indicar que el siguiente salto es `r1`. Después, deberás ejecutar `route` en `r1` y observar qué entrada de la tabla de encaminamiento se utiliza cuando se desea alcanzar `pc3`. Esta entrada debería indicar que el siguiente salto es `r2`. A continuación en `r2` deberás ejecutar `route` y observar qué entrada de la tabla de encaminamiento se utiliza cuando se desea alcanzar `pc3`. Esta entrada debería indicar que `r2` no necesita ningún router adicional para alcanzar `pc3`.

De forma análoga deberás consultar las tablas de encaminamiento en el sentido pc3⇒r2⇒r3⇒pc1.

4. Ejecuta en `pc1` un `traceroute` hacia `pc2` y en `pc2` uno hacia `pc1` para comprobar que las rutas son las especificadas.
5. Ejecuta en `pc2` un `traceroute` hacia `pc3` y en `pc3` uno hacia `pc2` para comprobar que las rutas son las especificadas.
6. Ejecuta en `pc1` un `traceroute` hacia `pc3` y en `pc3` uno hacia `pc1` para comprobar que las rutas son las especificadas. En este último `traceroute` observarás que aparecen unos `*`, en el final del tema 4 de teoría veremos a qué se deben. En la siguiente práctica se estudiarán más en detalle estas situaciones.
7. Antes de continuar asegúrate de que las rutas entre las máquinas son las especificadas en el apartado anterior.

   Lanza con `tcpdump` capturas en las siguientes máquinas, guardando los ficheros con el nombre que se indica:
   - captura en `r1(eth0)` con nombre de fichero `p3-b-01.cap`
   - captura en `r2(eth0)` con nombre de fichero `p3-b-02.cap`
   - captura en `r3(eth1)` con nombre de fichero `p3-b-03.cap`
   - captura en `pc3(eth0)` con nombre de fichero `p3-b-04.cap`

   Ejecuta en `pc1` un `ping` a `pc3` que envíe sólo 2 paquetes (`ping -c 2 <máquinaDestino>`).

   Interrumpe las 4 capturas (`Ctrl+C`).

   Analiza con `wireshark` las 4 capturas. Observa en ellas cómo los datagramas IP que se envían y reciben con la orden `ping` contienen un mensaje de ICMP. Comprueba en estos datagramas:
   - Dirección IP origen
   - Dirección IP destino
   - TTL en la cabecera IP
   - Campo Protocolo (tipo de protocolo) en la cabecera IP
   - Campos Tipo y Código en la cabecera ICMP.

   Consultando las capturas, responde a las siguientes cuestiones:

   a) ¿En qué se distinguen los mensajes "de ida" del `ping` de los mensajes "de vuelta"?

   b) ¿En qué capturas se pueden ver los mensajes "de ida" del `ping`? ¿Y los mensajes de vuelta? ¿Por qué?

   c) Comprueba los valores del campo TTL de la cabecera IP de todos los datagramas de todas las capturas y explica dichos valores.
8. Arranca de nuevo `tcpdump` en las mismas máquinas e interfaces que lo has hecho anteriormente pero guardando las capturas en otros ficheros diferentes:
   - captura en `r1(eth0)` con nombre de fichero `p3-b-05.cap`
   - captura en `r2(eth0)` con nombre de fichero `p3-b-06.cap`
   - captura en `r3(eth1)` con nombre de fichero `p3-b-07.cap`
   - captura en `pc3(eth0)` con nombre de fichero `p3-b-08.cap`

   Ejecuta en `pc1` la orden `traceroute` a `pc3`.

   Cuando la orden anterior haya terminado completament, interrumpe las capturas (`Ctrl+C`).

   A la vista del resultado que se muesta en `pc1`: ¿por qué *router* intermedios ha pasado un paquete para llegar de `pc1` a `pc3`?
9. Abre con `wireshark` los ficheros de captura que has obtenido. Identifica en los ficheros de capturas los siguientes paquetes:

- Los 3 mensajes enviados por `pc1` con TTL=1
- Los 3 ICMP de TTL excedido enviados por `r1`
- Los 3 mensajes enviados por `pc1` con TTL=2
- Los 3 ICMP de TTL excedido enviados por `r2`
- Los 3 mensajes enviados por `pc1` con TTL=3
- Los 3 ICMP de puerto inalcanzable enviados por `pc3`

10. Consultando las capturas, responde a las siguientes cuestiones:
    a) ¿Por qué ruta van viajando los mensajes enviados por `pc1` con TTL creciente?
    b) ¿Por qué ruta viajan los ICMP enviados por `r1`? ¿Qué dirección IP usa `r1` como IP de origen el enviar esos ICMP?
    c) ¿Por qué ruta viajan los ICMP enviados por `r2`? ¿Qué dirección IP usa `r2` como IP de origen el enviar esos ICMP?
    d) ¿Por qué ruta viajan los ICMP enviados por `pc3`?

3. Entrega de la práctica

Sube al enlace que encontrarás en Aula Virtual, y antes de que termine el plazo de entrega, los siguientes ficheros:

- Memoria
- Un único fichero p3.zip que contenga la carpeta `p3` con los ficheros de captura:
  - `p3-a-01.cap`, `p3-a-02.cap`, `p3-a-03.cap`
  - `p3-b-01.cap`, `p3-b-02.cap`, `p3-b-03.cap`, `p3-b-04.cap`, `p3-b-05.cap`, `p3-b-06.cap`, `p3-b-07.cap`, `p3-b-08.cap`

Práctica 4: `traceroute`

Redes de Ordenadores
2º Ingeniería Biomédica

Curso 2022-23

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

©2022
Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
Algunos derechos reservados
Este trabajo se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de
Creative Commons disponible en
http://creativecommons.org/licenses/by-sa/4.0/deed.es

Práctica 4: `traceroute`

Resumen

En esta práctica se aprende a:

- trabajar con escenarios de red preconfigurados
- estudiar las direcciones IP y las tablas de encaminamiento existentes en máquinas preconfiguradas
- realizar cambios a tablas de encaminamiento para cumplir las condiciones pedidas
- elegir direcciones IP y máscaras de subred apropiadas
- estudiar la salida del comando `traceroute` para deducir las rutas entre máquinas
- estudiar ficheros de captura para deducir las rutas entre máquinas

IMPORTANTE: En las figuras de los escenarios y en el texto de la práctica aparecen direcciones IP con una X entre medias (ej.: 12.X.0.100). Cada alumno tendrá en su escenario unas IPs con un valor concreto de X.

1. Escenario A

Descarga, introduciendo tu DNI en el enlace:

`http://mobiquo.gsyc.es/practicas/ro/p4.html`

el fichero `p4-lab-a.tgz`, que contiene un escenario de red. Descomprímelo de la misma manera que hiciste en la práctica anterior.

Lanza ahora NetGUI. En el menú, elige File → Open y selecciona la carpeta `p4-lab-a` en la que está el escenario. Verás aparecer la red de la figura 1.

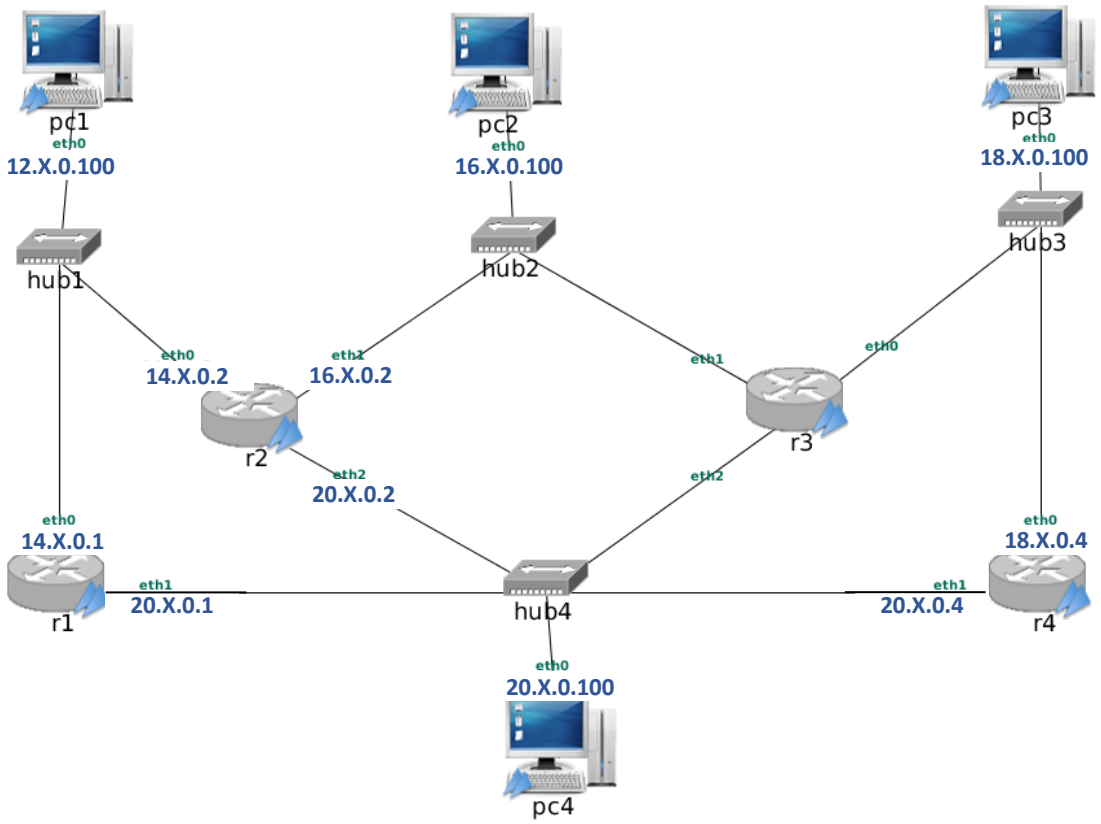


Figura 1: Escenario A

Arranca todas las máquinas de una en una, esperando que una máquina haya terminado su arranque antes de arrancar la siguiente.

1. Observa las direcciones IP que aparecen configuradas en el escenario de red. Comprobarás que todas las máquinas excepto `r3` tienen ya configurada su dirección IP. Mira el contenido de la tabla de *routing* de todas las máquinas (excepto `r3`)

2. Comprueba que en `pc1` no funciona un `ping` a la dirección `14.X.0.2`. ¿Por qué? Mira atentamente las direcciones IP de `pc1`, `r1-eth0` y `r2-eth0`. ¿Ves ya lo que pasa? Realiza los cambios necesarios **en la configuración de** `pc1` para que dicho `ping` funcione. Realiza los cambios de forma que `pc1` mantenga su nueva configuración aunque se apague y vuelva a encenderse.

   Haz una captura de tráfico en el fichero `p4-a-01.cap` que contenga paquetes que demuestren que dicho `ping` ya funciona adecuadamente.

3. La máquina `r3` no tiene configurada la dirección IP en sus interfaces de red. Configura direcciones IP adecuadas para sus interfaces `eth0`, `eth1` y `eth2`, de forma que dicha configuración se mantenga después de apagar y volver a encender `r3`. Elige todas las direcciones IP de `r3` de forma que terminen en `.3`.

4. Realiza los cambios necesarios en las tablas de *routing* adecuadas para que `pc2` y `pc3` puedan intercambiar datagramas IP y lo hagan por las siguientes rutas:

   - Desde `pc2` a `pc3`: `pc2 => r3 => pc3`
   - Desde `pc3` a `pc2`: `pc3 => r4 => r1 => r2 => pc2`

   Intenta realizar los mínimos cambios posibles sobre las tablas que ya existen.

   Ejecuta en `pc2` el comando `traceroute 18.X.0.100`. Comprueba que la salida del comando se corresponde con la ruta de `pc2` a `pc3` que has configurado.

   Repite el comando capturando el tráfico que se produce:

   - captura en el `hub2` lanzando tcpdump en `r3-eth1` con nombre de fichero `p4-a-02.cap`
   - captura en el `hub3` lanzando tcpdump en `r4-eth0` con nombre de fichero `p4-a-03.cap`
   - ejecuta en `pc2` el comando `traceroute 18.X.0.100`
   - interrumpe las capturas

   Analiza el tráfico capturado en ambos ficheros de captura y reconoce en ellos el comportamiento del comando `traceroute` tal y como se explica en las transparencias de teoría.

5. Intenta comprobar ahora la ruta de `pc3` a `pc2` ejecutando en `pc3` el comando `traceroute 16.X.0.100`. Espera a que termine completamente la ejecución del comando. Posiblemente aparecerán `*` en alguno de sus pasos. Recordando el comportamiento del traceroute, piensa a qué se deben dichos `*`. Para que no aparezcan, necesitarás configurar rutas para poder volver desde `r1` a `pc3` y desde `r2` a `pc3`. Introduce esas rutas y repite el comando hasta que funcione completamente sin mostrar ningún `*`.

6. Realiza los cambios necesarios en las tablas de *routing* para que `pc4` pueda intercambiar datagramas IP con `pc1`, `pc2` y `pc3`, independientemente de la ruta por la que lo haga. Intenta realizar los mínimos cambios posibles.

7. Localiza qué máquinas de entre `pc1`, `pc2`, `pc3` y `pc4` aún no pueden intercambiar datagramas entre sí. Realiza los cambios necesarios para que puedan. Intenta realizar los mínimos cambios posibles.

2. Escenario B

Descarga, introduciendo tu DNI en el enlace:

`http://mobiquo.gsyc.es/practicas/ro/p4.html`

el fichero `p4-lab-b.tgz`, que contiene el escenario de red para realizar los siguientes apartados. Descomprime el escenario de red `p4-lab-b` y abre dicho escenario dentro de NetGUI.

Arranca todas las máquinas de dicho escenario, de una en una, esperando que una máquina haya terminado su arranque antes de arrancar la siguiente. Obtendrás un escenario como el que se muestra en la figura 2.

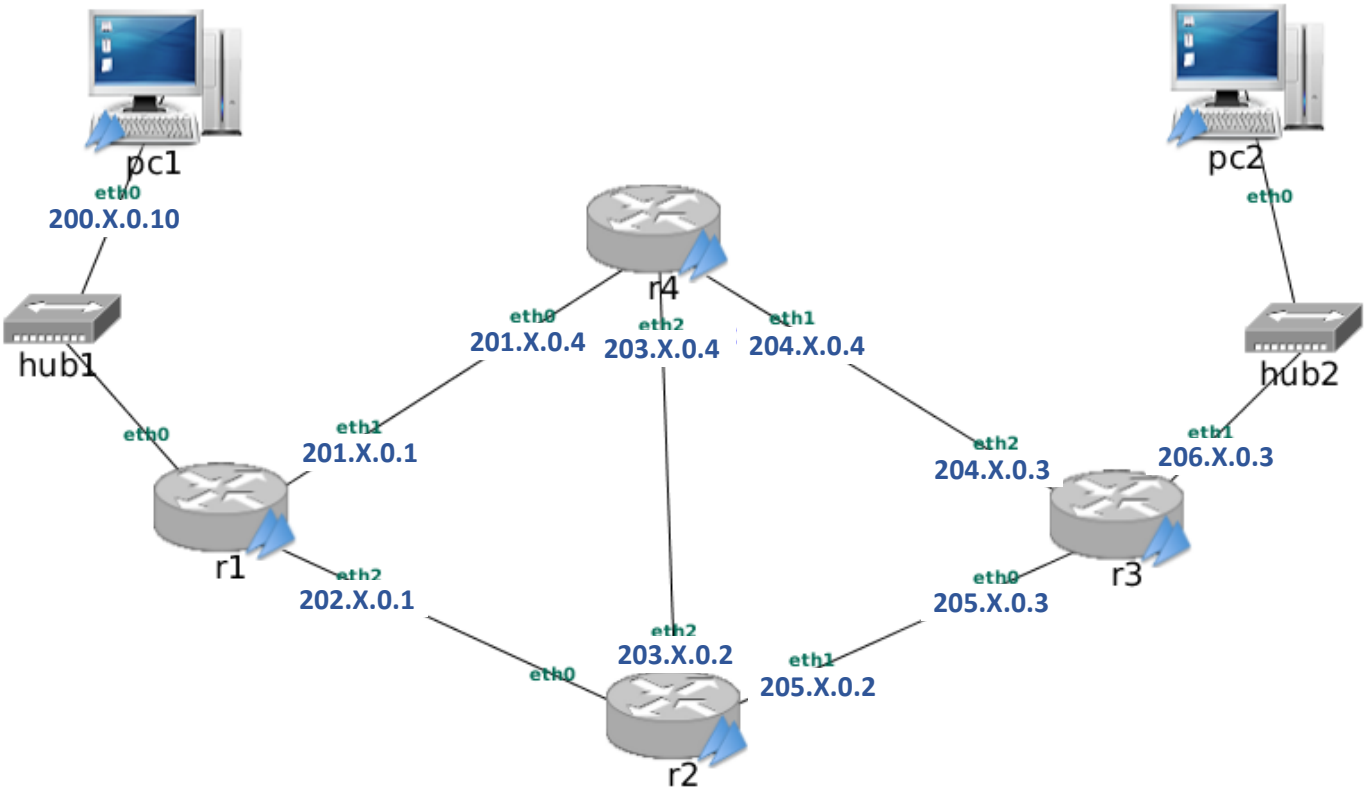


Figura 2: Escenario B

En algunas máquinas falta por configurar alguna dirección IP. Tendrás que configurarla más adelante según las condiciones que establezca el enunciado.

2.1. Traceroute desde pc1 a r4 y viceversa

Supón que se ejecutan los siguientes comandos en `pc1` y en `r4`, obteniéndose los resultados que se muestran:

- En `pc1` se ejecuta el siguiente comando:

```
pc1:~# traceroute 203.X.0.4
traceroute to 203.X.0.4 (203.X.0.4), 64 hops max, 40 byte packets
1 200.X.0.1
2 202.X.0.2
3 203.X.0.4
```

- En `r4` se ejecuta el siguiente comando:

```
r4:~# traceroute 200.X.0.10
traceroute to 200.X.0.10 (200.X.0.10), 64 hops max, 40 byte packets
1 203.X.0.2
2 202.X.0.1
3 200.X.0.10
```

1. ¿Cuáles son los *routers* que se atraviesan para ir desde `pc1` a la dirección `203.0.0.4`, en función de la salida mostrada?
2. ¿Cuáles son los *routers* que se atraviesan para ir desde `r4` a `pc1`, en función de la salida mostrada?
3. Realiza los cambios de configuración en las tablas de *routing* para que el resultado anterior se produzca en tu escenario. Efectúa sólo los cambios imprescindibles: no modifiques las rutas ni las direcciones IP que ya están configuradas en el escenario, **sólo puedes añadir direcciones IP y rutas**, cumpliendo además las siguientes restricciones:

   - En las tablas de *routing* de las máquinas (`pc1` y `pc2`) sólo puedes añadir rutas por defecto.
   - En las tablas de *routing* de los *routers* NO puedes añadir rutas por defecto.

   Utiliza `traceroute` sobre el escenario modificado para comprobar que su salida es la misma que la mostrada al principio de este apartado.

   Captura ahora el tráfico que se produce en el `traceroute` de `pc1` a `r4`:

   - lanza tcpdump en `r1-eth0` con nombre de fichero `p4-b-01.cap`
   - lanza tcpdump en `r2-eth0` con nombre de fichero `p4-b-02.cap`
   - lanza tcpdump en `r4-eth2` con nombre de fichero `p4-b-03.cap`
   - ejecuta en `pc1` el comando `traceroute 203.X.0.4`
   - interrumpe las capturas

   Analiza el tráfico capturado en ambos ficheros de captura y reconoce en ellos el comportamiento del comando `traceroute` tal y como se explica en las transparencias de teoría.

2.2. Traceroute desde pc1 a pc2

Supón que se ejecuta en `pc1` un `traceroute` a `pc2` y se obtiene el siguiente resultado:

```
pc1:~# traceroute 206.X.0.10
traceroute to 206.X.0.10 (206.X.0.10), 64 hops max, 40 byte packets
1 200.X.0.1
2 202.X.0.2
3 204.X.0.3
4 206.X.0.10
```

1. ¿Cuáles son los *routers* que se atraviesan para ir desde `pc1` a `pc2`, en función de la salida producida por `traceroute`?
2. ¿Por qué en el resultado de `traceroute` la dirección IP del tercer salto es 204.X.0.3 en vez de 205.X.0.3?
3. Realiza los cambios de configuración en las tablas de *routing* para que el resultado anterior se produzca en tu escenario. Efectúa sólo los cambios imprescindibles: no modifiques las rutas ni las direcciones IP que ya están configuradas en el escenario, **sólo puedes añadir direcciones IP y rutas**, cumpliendo además las siguientes restricciones:

   - En las tablas de *routing* de las máquinas (`pc1` y `pc2`) sólo puedes añadir rutas por defecto.
   - En las tablas de *routing* de los *routers* NO puedes añadir rutas por defecto.

   Utiliza `traceroute` sobre el escenario modificado para comprobar que su salida es la misma que la mostrada al principio de este apartado.

   Captura ahora el tráfico que se produce:

   - lanza tcpdump en `r1-eth0` con nombre de fichero `p4-b-04.cap`
   - lanza tcpdump en `r2-eth0` con nombre de fichero `p4-b-05.cap`
   - lanza tcpdump en `r3-eth0` con nombre de fichero `p4-b-06.cap`
   - lanza tcpdump en `pc2-eth0` con nombre de fichero `p4-b-07.cap`
   - lanza tcpdump en `r4-eth1` con nombre de fichero `p4-b-08.cap`
   - ejecuta en `pc1` el comando `traceroute 206.X.0.10`
   - interrumpe las capturas

   Analiza el tráfico capturado en ambos ficheros de captura y reconoce en ellos el comportamiento del comando `traceroute` tal y como se explica en las transparencias de teoría.

3. Escenario C

Descarga, introduciendo tu DNI en el enlace:

`http://mobiquo.gsyc.es/practicas/ro/p4.html`

el fichero `p4-lab-c.tgz`, que contiene el escenario de red para realizar los siguientes apartados.
Descomprime el escenario de red `p4-lab-c` y abre dicho escenario dentro de NetGUI.

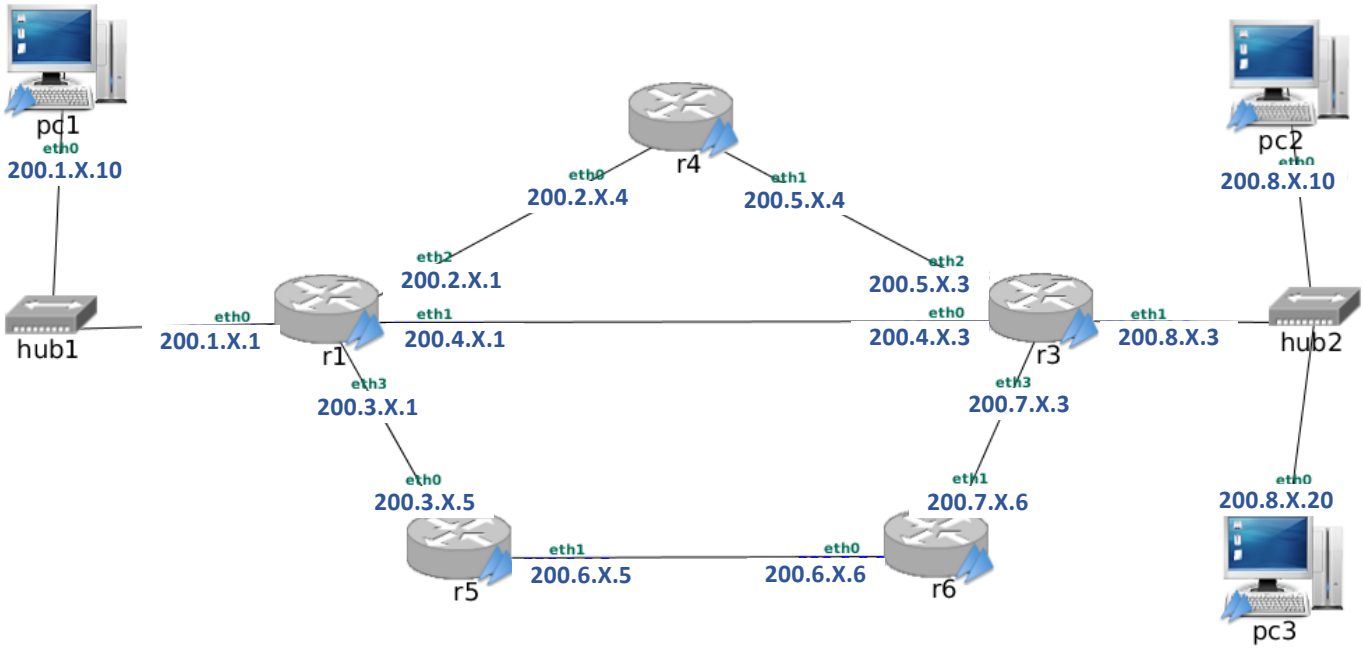


Figura 3: Escenario C

Arranca todas las máquinas de dicho escenario, de una en una, esperando que una máquina haya terminado su arranque antes de arrancar la siguiente. Obtendrás un escenario como el que se muestra en la figura 3.
El escenario no está configurado completamente. Algunas máquinas necesitan configurar rutas. Irás realizando dicha configuración a lo largo de los siguientes apartados.

3.1. Caso 1

En un escenario como el mostrado en `p4-lab-c` pero con valor de $X = 0$ se han ejecutado una o más órdenes y, mientras se ejecutaban, se obtuvieron las siguientes capturas:

- `cap1.cap`: Captura realizada en la red 200.6.X.0.
- `cap2.cap`: Captura realizada en la red 200.4.X.0.

Analiza dichas capturas simultáneamente ya que las dos juntas serán las que te permitirán responder a la siguiente cuestión: ¿Qué órdenes se ejectuaron para poder obtener esas capturas?
Modifica en el escenario las rutas necesarias para que se puedan realizar estas capturas al ejecutar dichas órdenes.
Comprueba tus respuestas ejecutando dichas órdenes sobre tu escenario mientras realizas capturas en las redes indicadas:

- lanza tcpdump en `r5-eth1` con nombre de fichero `p4-c-01.cap`
- lanza tcpdump en `r3-eth0` con nombre de fichero `p4-c-02.cap`
- ejecuta las órdenes para generar el tráfico
- interrumpe las capturas

Tus capturas deben ser iguales a las que te damos (sin tener en cuenta los posibles paquetes de ARP, y por supuesto, teniendo las IPs el valor adecuado de tu X).

3.2. Caso 2

En un escenario como el mostrado en `p4-lab-c` pero con valor de $X = 0$ se han ejecutado una o más órdenes mientras se realizaba la siguiente captura: `cap3.cap`.
Analiza la captura para poder responder las siguientes cuestiones: ¿En qué red se ha realizado dicha captura? ¿Qué órdenes han tenido que ejecutarse para poder obtener el tráfico de esta captura?
Modifica en el escenario las rutas necesarias para que se pueda realizar esta captura al ejecutar dichas órdenes.
Comprueba tus respuestas ejecutando dichas órdenes sobre tu escenario mientras realizas una captura en la red adecuada. Llama a dicho fichero de captura `p4-c-03.cap`.
Tu captura debe ser igual a la que te damos (sin tener en cuenta los posibles paquetes de ARP, y por supuesto, teniendo las IPs el valor adecuado de tu X).

4. Entrega de la práctica

Sube al enlace que encontrarás en Aula Virtual, y antes de que termine el plazo de entrega;

- Memoria en formato pdf
- Crea una carpeta `p4` con los siguientes ficheros de captura:
  - `p4-a-01.cap`, `p4-a-02.cap`, `p4-a-03.cap`
  - `p4-b-01.cap`, `p4-b-02.cap`, `p4-b-03.cap` `p4-b-04.cap`, `p4-b-05.cap`, `p4-b-06.cap`, `p4-b-07.cap`, `p4-b-08.cap`
  - `p4-c-01.cap`, `p4-c-02.cap`, `p4-c-03.cap`

  Y comprime dicha carpeta para generar un fichero `p4.zip`.

Práctica 5: UDP, TCP

Redes de Ordenadores
2º Ingeniería Biomédica

Curso 2022-23

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

Universidad Rey Juan Carlos

©2022
Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
Algunos derechos reservados
Este trabajo se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de
Creative Commons disponible en
http://creativecommons.org/licenses/by-sa/4.0/deed.es

Práctica 5: UDP, TCP

Resumen

En esta práctica se aprende el funcionamiento básico de los protocolos de nivel de transporte UDP y TCP.

Nota: Al cargar capturas en *Wireshark* es necesario ordenar los paquetes por su marca de tiempo, pulsando en la pestaña `Time`, de esta forma podremos analizar lo que ha ocurrido ordenadamente siguiendo el eje temporal.

1. Comunicación de aplicaciones usando el protocolo UDP

1.1. Análisis de captura de tráfico UDP

En la captura `udp.cap` se muestra una comunicación UDP. Contesta a las siguientes preguntas:

1. ¿Cuáles son las direcciones IP y puertos involucrados en la comunicación?
2. ¿Cuál es el número de paquetes UDP intercambiados? ¿Y número de bytes de datos intercambiados?

1.2. Estudio de UDP mediante aplicaciones cliente y servidor lanzadas con nc

Descarga de la página de la asignatura el fichero `lab-p5.tgz`, que contiene un escenario de red:

`http://mobiquo.gsyc.es/practicas/ro/p5.html`

Descomprímelo de la misma manera que hiciste en prácticas anteriores.
Lanza ahora NetGUI. En el menú, elige File → Open y selecciona la carpeta `lab-p5` en la que está el escenario. Verás aparecer la red de la figura 1.

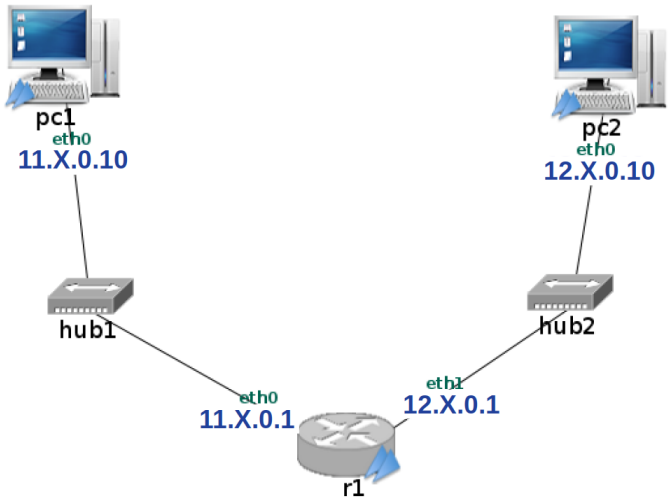


Figura 1: Escenario lab-p5

Arranca las máquinas de una en una, esperando que una máquina haya terminado su arranque antes de arrancar la siguiente.
En este apartado utilizarás la orden `nc` para observar el funcionamiento de UDP en diversas situaciones.

1.2.1. UDP es un protocolo basado en datagramas: no hay establecimiento de conexión

1. Inicia una captura en el router `r1` (fichero de captura `p5-udp-01.cap`).
2. Usando `nc` lanza una aplicación servidor UDP en la máquina `pc2` y puerto 11111: `nc -u -l -p 11111`
3. Usando `nc` lanza una aplicación cliente UDP en la máquina `pc10` para que se comunique con el servidor (no envíes datos ni desde el cliente al servidor ni desde el servidor al cliente), desde el puerto local 33333: `nc -u -p 33333 <dir_IP_pc2> 11111`
4. Interrumpe la captura.

Explica qué paquetes deberían haberse capturado. Observa la captura y comprueba tu suposición.

1.2.2. Fragmentación IP con envíos UDP

1. Inicia una nueva captura en el router `r1` (fichero de captura `p5-udp-02.cap`).
2. Escribe en el terminal donde tienes lanzado el cliente 20 líneas de texto, pulsando una letra cualquiera del teclado (con el tamaño por defecto del terminal de NetGUI, cada línea permite escribir 80 caracteres, así que estarás generando 80x20=1600 carácteres, cada uno de ellos ocupando 1 byte).
3. A continuación pulsa la tecla `INTRO` o `ENTER` (véase la figura 2).

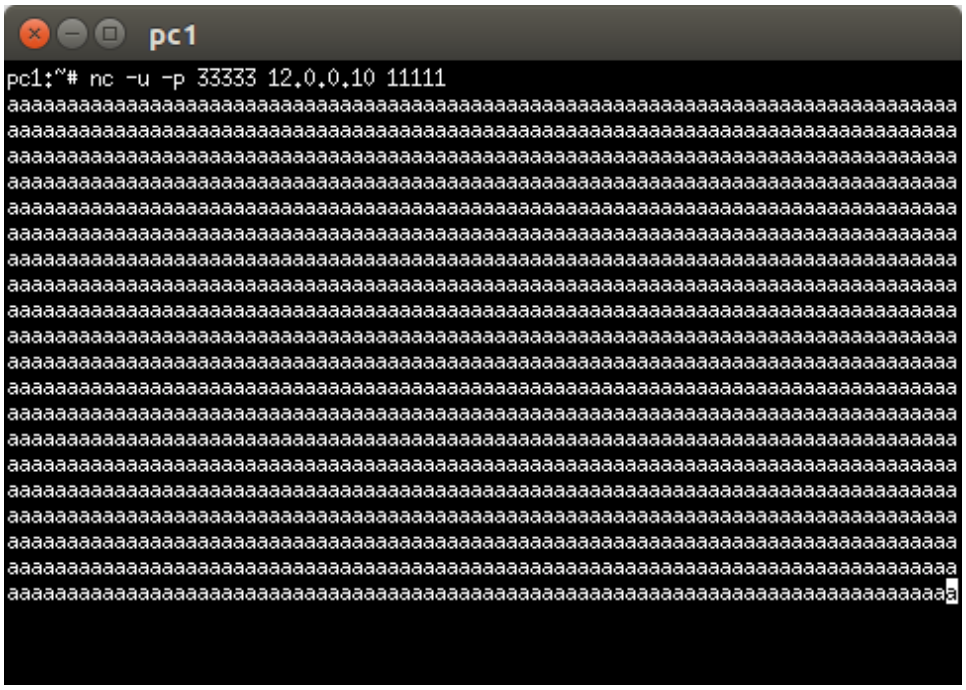


Figura 2: Tráfico UDP

Antes de observar en la captura lo que ha ocurrido responde a estas preguntas:

1. ¿cuántos datagramas UDP crees que se han enviado, y por qué?
2. ¿cuántos datagramas IP crees que se han enviado, y por qué?
3. ¿cuántos bytes de datos irán en cada datagrama UDP?

Interrumpe ahora la captura y comprueba si tus suposiciones son correctas. Explica **razonadamente** el número de datagramas UDP, el número de datagramas IP, y el número de bytes de datos que va en cada uno de los datagramas UDP.

1.2.3. UDP es un protocolo basado en datagramas: no hay cierre de conexión

1. Inicia una captura en el router `r1` (fichero de captura `p5-udp-03.cap`).
2. Interrumpe la ejecución del cliente pulsando `Ctrl+C`.
3. Interrumpe la ejecución del servidor pulsando `Ctrl+C`.

Piensa en cuántos paquetes deberían haberse capturado y por qué. Interrumpe la captura y comprueba tu suposición.

1.2.4. Buffer de recepción en UDP

Los protocolos de nivel de transporte trabajan tanto en TCP como en UDP con buffers `in/out` (recepción/envío) asociados a cada puerto (en el caso de TCP asociados a cada conexión). Cuando llega un datagrama UDP o un segmento TCP, se almacenan los bytes para la aplicación en el buffer `in` correspondiente. Cuando una aplicación envía bytes a través de un puerto UDP o de una conexión TCP, se almacenan sus bytes en el buffer `out` asociado a dicho puerto/conexión.
Para ver estos buffers vamos a utilizar la herramienta `netstat`.

1. Inicia una captura en `r1` (guarda el contenido en el fichero `p5-udp-04.cap`).

2. Lanza en segundo plano[1] una aplicación servidor utilizando nc en la máquina `pc2` que reciba mensajes UDP destinados al puerto 11111 en segundo plano:

   ```
   nc -u -l -p 11111 &
   ```

3. Observa el estado que muestra `netstat -una` en el servidor para la aplicación que acabas de arrancar y sus buffers.

4. Trae a primer plano la ejecución de la aplicación servidor arrancada con `nc`, ejecutando para ello en el terminal del servidor esta orden: `fg`

   Tras traer a primer plano una aplicación, ésta recupera la entrada estándar del teclado. Ahora por tanto lo que se teclee se le envía a la aplicación servidor.

5. Pausa con `Ctrl+Z` la ejecución de la aplicación servidor `nc`. De esta forma la aplicación en el servidor siga arrancada pero no esté ejecutándose (la CPU no ejecutará instrucciones de la aplicación servidor mientras esté suspendida su ejecución). Mientras está suspendido el servidor, si la implementación de UDP en la máquina del servidor recibe datos destinados a la aplicación servidor, estos datos se almacenarán en el correspondiente buffer `in`, y no van a ser leídos por la aplicación `nc` porque su ejecución está suspendida temporalmente.

6. Para ver cómo los datos se quedan almacenados en el buffer `in` de UDP del puerto en el que escucha el servidor, envía unos cuantos caracteres desde el cliente y pulsa la tecla `INTRO` o `ENTER`.

7. Ejecuta `netstat -una` en el servidor para ver cómo esos datos se quedan en el buffer de recepción y no los lee la aplicación. Verás que la cantidad muestra 1712 bytes y no la cantidad de caracteres que has introducido desde el cliente. Esto es debido a que en UDP se reserva este cantidad en el buffer por cada paquete recibido.

8. Ahora vamos a volver a activar el servidor, que estaba suspendido con `Ctrl+Z`, trayéndolo a primer plano. Ejecuta para ello la orden `fg`. Verás como inmediatamente los datos que habías enviado desde el cliente se muestran en la pantalla: la aplicación servidor los ha leído del buffer de recepción, que se ha vaciado, y los ha mostrado en la pantalla.

9. Una vez realizada la prueba puedes interrumpir la ejecución del cliente y el servidor con `Ctrl+C`.

10. Interrumpe la captura y trata de relacionar lo que ha ocurrido con el contenido de la captura.

1.2.5. Errores en las comunicaciones UDP

Provoca las siguientes situaciones de error:

1. Existe la máquina `pc2`, pero en ella no hay una aplicación servidor escuchando en el puerto 11111. Inicia una captura en el router `r1` (guarda el contenido en `p5-udp-05.cap`). Prueba a lanzar el cliente y envía datos desde el cliente. Interrumpe la captura. Explica razonadamente los paquetes capturados.

2. Existe la red 12.X.0.0/24 y hay ruta para llegar hasta ella, pero no existe la máquina `pc2` (para realizar este apartado apaga la máquina `pc2`). Inicia una captura en el router `r1` y guarda el contenido en `p5-udp-06.cap`. Prueba a lanzar el cliente y envía datos desde el cliente. Espera 1 minuto e interrumpe la captura. Explica su contenido.

2. Comunicación de aplicaciones usando el protocolo TCP

En este apartado utilizarás la orden `nc` para observar el funcionamiento de TCP en diversas situaciones.
Utilizaremos el mismo escenario NetGUI del apartado anterior.

[1] Añadiendo un carácter & al final de una orden introducida en la shell se ejecuta dicha orden en segundo plano, lo que significa que la entrada del teclado no quedará ligada al proceso arrancado, pudiéndose así seguir ejecutando comandos en la shell a la vez que se está ejecutando concurrentemente la orden ejecutada en segundo plano

2.1. TCP es un protocolo orientado a conexión.

2.1.1. Establecimiento de conexión

1. Inicia una captura en el router `r1` y guarda su contenido en un fichero (`p5-tcp-01.cap`).
2. Usando nc, lanza una aplicación servidor en la máquina `pc2` que atienda peticiones de conexión destinadas al puerto TCP 11111: `nc -l -p 11111`
3. Lanza una aplicación cliente con nc en la máquina `pc1` para que se establezca una conexión TCP con la aplicación servidor, usando como puerto local TCP el 33333: `nc -p 33333 <dir_IP_pc2> 11111`. Explica cuántos paquetes deberían haberse capturado y por qué.

Interrumpe la captura y comprueba si tus respuestas se corresponden con lo observado en la captura.

2.1.2. Cierre de conexión

1. Inicia una captura en el router `r1` y guarda el contenido en un fichero (`p5-tcp-02.cap`).
2. Interrumpe la ejecución del cliente pulsando `Ctrl+C` en el cliente. Explica cuántos paquetes deberían haberse capturado y por qué.

Interrumpe la captura y comprueba si tus respuestas se corresponden con lo observado en la captura.

NOTA: Con `nc`, una vez que se interrumpe la conexión desde el cliente, el servidor también cierra la conexión. Con otras aplicaciones podría mantenerse abierta la conexión desde el servidor si éste tuviera más datos que enviar.

2.2. Buffer de recepción en TCP

Vamos a visualizar los bufferes in/out (recepción/emisión) en TCP. Inicia una captura en el terminal de `r1` guardando el contenido en un fichero (`p5-tcp-03.cap`).

1. Lanza una aplicación servidor utilizando `nc` en la máquina `pc2` que acepte conexiones en el puerto TCP 11111 (arráncala en segundo plano): `nc -l -p 11111 &`
2. Lanza una aplicación cliente con `nc` en la máquina `pc1` para que se conecte al servidor, desde el puerto local TCP 33333: `nc -p 33333 <dir_IP_pc2> 11111`
3. Observa el estado que muestra `netstat -tna` en el servidor y sus buffers.
4. Trae a primer plano la ejecución de `nc` ejecutando en el terminal: `fg`
5. Pausa con `Ctrl+Z` la ejecución de `nc` en el servidor, para que la aplicación del servidor siga arrancada pero no se ejecute, por tanto, si TCP en el lado servidor recibe datos, estos no se se van a leer en `nc` quedarán almacenados en la cola de entrada de la implementación de TCP.
6. Para ver cómo los datos se quedan almacenados en el servidor, envía una cadena de caracteres desde el cliente y pulsa `INTRO`.
7. Ejecuta `netstat -tna` en el servidor para ver cómo esos datos se quedan en el buffer de recepción y no los lee la aplicación.
8. Interrumpe la captura y fíjate cómo hay un asentimiento que indica que todos los datos han sido recibidos. Dado que la aplicación servidor está suspendida, los datos se encuentran almacenados en el buffer de recepción de la implementación de TCP, en el kernel del sistema operativo, pero no los ha leído aún la aplicación servidora arrancada con nc.
9. Trae a primer plano la ejecución del servidor, para ello usa `fg`. Verás como los datos que habías enviado desde el cliente se muestran en la pantalla. El servidor los ha leído del buffer de recepción y el buffer está vacío.

Una vez realizada la prueba puedes interrumpir la ejecución del cliente y el servidor.

2.3. Errores en las comunicaciones TCP

Provoca las siguientes situaciones de error:

1. Existe la máquina `pc2` pero no hay una aplicación escuchando en el puerto 11111. Realiza una captura para comprobar el tráfico generado (`p5-tcp-04.cap`). Prueba a lanzar el cliente y comprueba qué ocurre.
2. Existe la red 12.X.0.0 y hay ruta para llegar hasta ella, pero no existe la máquina `pc2`. (Para realizar este apartado apaga la máquina `pc2`). Realiza una captura para comprobar el tráfico generado (`p5-tcp-05.cap`). Prueba a lanzar el cliente y comprueba qué ocurre. Interrumpe la captura y analízala.

2.4. Análisis inicial de la captura de TCP

En la captura `tcp.cap` se muestra una comunicación TCP entre dos aplicaciones.

Contesta a las siguientes preguntas:

1. ¿Cuál es la dirección IP y el puerto del cliente TCP y la dirección IP y el puerto del servidor TCP?
2. ¿Cuántos segmentos TCP se han enviado desde el cliente al servidor?
3. ¿Cuántos segmentos TCP se han enviado desde el servidor al cliente?
4. Indica qué extremo cierra antes la conexión.

2.5. Números de secuencia

Sigue analizando la captura `tcp.cap`.

Observa que, por defecto, *Wireshark* muestra números de secuencia relativos para los dos sentidos de la conexión. Seleccionando en el menú *Edit→Preferences→Protocols→TCP*, y desactiva la opción *Relative Sequence Numbers*. De esta forma podrás observar los números de secuencia reales, en lugar de los números relativos que muestra por omisión *Wireshark*. Deja de momento desactivada esta opción.

1. ¿Cuáles son los números de secuencia reales que viajan en los paquetes que tienen activados el flag SYN o el flag FIN?

   Una vez que los hayas apuntado, vuelve a las opciones de TCP y activa otra vez la opción *Relative Sequence Numbers*. Compara ahora los números de secuencia relativos de los paquetes SYN y FIN.

   Deja ya siempre activada la opción para ver los números de secuencia relativos ya que permite analizar más cómodamente la conexión.

   NOTA: También puede resultarte conveniente desactivar la opción *Ignore TCP Timestamps in Summary*. De esta forma no aparecerán las marcas de tiempo (opción de TCP presente en los paquetes de algunas capturas) en el resumen de los paquetes, lo que te permitirá reconocer más rápidamente otros datos más importantes que figuran en ese resumen.

2. Fíjate en el campo `Len` que aparece en el resumen de los paquetes. ¿A qué se refiere?

   ¿Cuántos bytes de datos envía el cliente al servidor? Si lo has calculado sumando los `Len` de los paquetes, ten en cuenta que algunos paquetes podrían ser retransmisiones (no pasa en esta captura) y no podrías sumarlos varias veces para calcular los bytes de datos enviados.

   ¿Cómo harías el cálculo si la conexión tuviera cientos de paquetes? Fíjate en el números de secuencia relativo del FIN que envía el cliente, y qué relación tienen con la cantidad de datos enviados.

   Ten en cuenta que el paquete FIN que envía el cliente podría contener datos, aunque eso no pasa en esta conexión. En ese caso, tendrías que tenerlos en cuenta también para calcular cuántos datos se envían en la conexión. Por esta razón es más conveniente es fijarse en el ACK del FIN que envía el servidor al cliente, ¿qué número de ACK lleva? ¿Qué relación tiene con la cantidad de datos enviados?

3. ¿Cuántos bytes de datos envía el servidor al cliente? Razona la respuesta en base a lo aprendido en la pregunta anterior.

2.6. RTT

Sigue analizando la captura `tcp.cap`.

1. Para cada uno de los segmentos con datos que envía el cliente al servidor, para su paquete SYN y para su paquete FIN, indica cuál es el RTT del paquete. Observa para ello la columna `Time` del resumen de paquetes.

2.7. Retransmisión de SYN

En la captura `tcp-syn.cap` se muestra una comunicación TCP. Contesta a las siguientes preguntas:

1. ¿Cuántas veces se envía el segmento SYN del cliente al servidor?

2. Indica la secuencia de marcas de tiempo de los segmentos SYN enviados, y explica sus valores atendiendo al mecanismo *exponential backoff*. Ten en cuenta que al principio de una conexión aún no se ha podido medir el RTT del cliente al servidor. ¿Qué plazo de retransmisión inicial se ha utilizado ante la ausencia aún de un valor de RTT?

3. ¿Por qué crees que se ha retransmitido tantas veces el SYN? Observa los paquetes 8, 10, 12 y 14, ¿qué son? ¿Y los paquetes 9, 11, 13 y 15? ¿Cuál es por tanto la causa real de tantas retransmisiones del SYN?

2.8. Funcionamiento básico de la ventana anunciada

En la captura `tcp-window.cap` se muestra el principio de una comunicación TCP.

Ayudándote de la gráfica `tcptrace`, contesta a las siguientes preguntas relativas al sentido de comunicación 12.0.0.100:36185 → 13.0.0.100:37777:

1. ¿Cuál es el número de secuencia del primer byte de datos contenido en el paquete número 6?

2. ¿Cuál es el número de secuencia del último byte de datos contenido en el paquete número 6?

3. El paquete número 8, ¿cuántos bytes de datos asiente? ¿cuál es el número de secuencia del último byte asentido por este paquete?

4. Justo después de recibir el segmento número 8, y antes de recibir el segmento siguiente ¿cuál es el último valor de ventana anunciada por el receptor que ha recibido el emisor? ¿Cuántos bytes de datos podría enviar como máximo el emisor en ese momento, en uno o más segmentos, antes de que llegara otro ACK del receptor?

5. En el momento de enviar el paquete número 13, ¿en qué paquete venía la información de ventana aplicable al emisor en ese momento? ¿Cuántos bytes de datos más, además del paquete número 13, podría enviar el emisor en ese instante, antes de recibir otro ACK?

6. ¿Podría haber enviado el emisor un segmento con datos nuevos en el instante 0.321000 segundos? ¿Por qué?

7. Identifica en la gráfica `tcptrace` de este sentido de la comunicación en qué otro periodo de tiempo se produce una situación similar a la que ocurre al enviarse el paquete número 13.

Cambia ahora el sentido de la comunicación sobre la gráfica `tcptrace` para analizar el sentido de comunicación 12.0.0.100:36185 → 13.0.0.100:37777:

1. ¿Se envían bytes de datos en este sentido en los paquetes que hay en la captura? Compruébalo también sobre la lista de paquetes.

2.9. Retransmisiones y asentimientos

En la captura `tcp-timeout-probes.cap` se muestra el principio de una comunicación TCP.

Ayudándote de la gráfica `tcptrace`, contesta a las siguientes preguntas relativas al sentido de comunicación 12.0.0.100:43122 → 13.0.0.100:43123:

1. El paquete número 16 es una retransmisión, aunque *Wireshark* no lo etiqueta como tal:

   a) Mirando la gráfica `tcptrace`, ¿cómo puede identificarse que dicho paquete 16 es una retransmisión?

   b) ¿Qué paquete es la primera transmisión de los bytes de datos que viajan en el paquete 16?

   c) ¿Cuál ha sido el plazo de retransmisión aplicado a esa primera retransmisión del paquete?

   d) Mirando la gráfica `tcptrace`, ¿vuelve a ser retransmitido este paquete en la conexión? ¿Con qué plazo de retransmisión? ¿Por qué? Busca en la lista de paquetes la segunda retransmisión del paquete. Acostúmbrate a identificar paquetes en la gráfica y en la lista. Ayúdate del `Time` de los paquetes en la lista y de los valores de tiempo en el eje de las x de la gráfica.

2. Identifica en la gráfica `tcptrace` otros segmentos que son retransmisión.

3. En el momento de transmitir el paquete 18, ¿cuántos bytes están transmitidos y pendientes de que llegue su asentimiento? ¿Y cuántos paquetes (indica su número de paquete)?

4. El paquete 19 es un asentimiento. ¿Cuántos bytes asiente? ¿Y cuántos segmentos?

5. Identifica sobre la lista de paquetes otros asentimientos que asienten varios paquetes a la vez. Si utilizas la versión de `wireshark-gtk` verás que en la gráfica `tcptrace` se distingue cada paquete de asentimiento aunque lleguen varios casi a la vez. Así, con `wireshark-gtk` puede resultar más fácil identificar qué paquetes asiente cada asentimiento, cosa que no es tan fácil con la nueva versión de `wireshark`.

2.10. Sondas de ventana

Sigue analizando la captura `tcp-timeout-probes.cap`. Responde a las siguientes cuestiones.

1. Identifica en la gráfica `tcptrace` un periodo de tiempo en que el servidor está anunciando ventana 0 al cliente. Localiza en la lista de paquetes los que son enviados y recibidos en ese periodo de tiempo. ¿Por qué el cliente no envía bytes de datos en ese periodo de tiempo?

2. ¿Qué números de paquete son las sondas de ventana? ¿Cómo los etiqueta *wireshark*?

3. ¿Cuánto tiempo pasa desde que el cliente recibe el primer anuncio de ventana 0 hasta que envía la primera sonda de ventana?

4. ¿Cuánto tiempo pasa desde que el cliente recibe el segundo anuncio de ventana 0 hasta que envía la segunda sonda de ventana?

5. ¿Qué paquete es el que anuncia al cliente que la ventana vuelve a estar abierta? ¿Cuántos bytes de datos puede enviar el cliente a partir de ese momento? ¿Entre qué números de secuencia?

2.11. Factor de escala sobre la ventana anunciada

Vuelve a abrir la captura `tcp.cap`.

En los segmentos que llevan activado el flag SYN se informa de los valores de número de secuencia inicial y ventana inicial anunciada que tiene cada extremo de la comunicación TCP. La opción factor de escala de la ventana (`window scale`), cuando se utiliza, se indica en la parte de opciones de los segmentos que llevan el flag SYN.

Cuando el lado que abre la conexión quiere utilizar un factor de escala para la ventana anunciada, activará la opción en su paquete SYN. El otro lado debe activar la opción en su paquete SYN+ACK para indicar que entiende esta opción de TCP y va a utilizarla también. A partir de este momento, en los segmentos que envíen ambos lados se aplicará el factor de escala para calcular el valor real de la ventana que se está anunciando: se multiplicará 2^{factor} por el campo de ventana anunciada que viaja en el segmento. Nótese que sólo en el segmento SYN y en el SYN+ACK viaja el factor de escala, pero éste no se aplica al campo ventana anunciada de estos dos segmentos, sino que se aplica a todos los demás segmentos enviados.

1. Observa los segmentos SYN enviados por el cliente y por el servidor. En el segundo panel de *Wireshark* despliega la cabecera TCP y busca la parte de opciones. Localiza la opción *Window Scale*. Indica el factor de escala que envía el cliente y el factor de escala que envía el servidor.

   Observa que en el resumen de paquetes, para los paquetes SYN *Wireshark* anota `WS=x` siendo `x` $= 2^{factor}$.

   Mira dentro la cabecera de los dos paquetes SYN el campo de ventana anunciada. Observa que *Wireshark* muestra el valor real del campo (*Window size value*) y luego el valor tras aplicar el factor de escala *Calculated window size*. Observa que en los paquetes SYN **aún no se aplica el factor de escala**.

2. Indica cuál es el valor del campo de ventana anunciada (`Window size`) en el resto de los segmentos de la conexión (los que NO tienen el flag SYN activado) en cada uno de los dos sentidos.

   Observa que en el resumen de paquetes aparece el valor de ventana ya escalado, mientras que en la cabecera puedes comprobar el valor que viaja en el campo, y el valor tras aplicar el escalado.

   Señala para cada paquete el valor que viaja en el campo y el valor final de la ventana anunciada tras aplicar el factor de escala.

   Ten en cuenta que en esta captura los lados utilizan el mismo factor de escala, pero en otras conexiones cada lado podría usar un factor de escala diferente para sus valores de ventana.

NOTA: En muchas conexiones podrás ver que las ventanas anunciadas inicialmente son relativamente pequeñas, y que poco a poco, según avanza la conexión y el emisor va enviando datos, el receptor le aumenta el tamaño de ventana. Esto se debe a que la implementación de TCP comienza reservando un buffer de recepción no demasiado grande, porque no sabe si el emisor va a transmitir muchos o pocos datos. Según el emisor va llenando la ventana, el receptor (si dispone de más memoria) va aumentando el tamaño del buffer, y por lo tanto, de la ventana anunciada.

2.12. MSS

En la captura `tcp-mss-pmtu.cap` se muestra el principio de una comunicación TCP. Ordena en *Wireshark* los paquetes por la columna `Time`. Contesta a las siguientes preguntas:

1. Indica cuál es el valor anunciado de MSS en la parte opcional de las cabeceras de los paquetes SYN de los dos sentidos de la conexión. Dados estos dos valores, indica qué tamaño de datos crees usarán ambos sentidos de la conexión si tienen que enviar datos.

2. Mira los tamaños de las **cabeceras** de los segmentos 1 y 3 enviados por el cliente y verás que son distintos. Teniendo en cuenta que cada segmento enviado en la conexión podría tener un tamaño de cabecera diferente, ¿cómo se calculará la cantidad máxima de datos que puede llevar un segmento dado el MSS y el tamaño de la cabecera?

3. Teniendo en cuenta que en el instante en el que se envía el segmento número 4, la máquina 11.0.0.11 tenía más de 2.000 bytes de datos que enviar a la máquina 12.0.0.12, ¿por qué envía menos? ¿Cuántos bytes envía en ese segmento 4? ¿Cómo se justifica el número de bytes de datos que contiene el segmento 4 dados los valores de MSS anunciados en los segmentos SYN?

4. Explica qué tipo de paquete son los paquetes 5 y 7 de la captura. ¿Qué significan y cuál puede ser la razón de que se hayan enviado? ¿Qué campo de la cabecera IP de los paquetes 4 y 6 ha provocado que se envíen los paquetes 5 y 7?

5. Mira el paquete 8 de la captura. ¿Por qué se retransmite este segmento? Comprueba el tamaño de su campo de datos. Explica la relación de este valor con la información contenida en los paquetes 5 y 7.

Entrega de la práctica

Sube al enlace que encontrarás en Aula Virtual, y antes de que termine el plazo de entrega, los siguientes ficheros:

- Memoria en pdf
- Fichero `p5.zip` resultado de comprimir **la carpeta p5** que contenga los siguientes ficheros de captura:
  - `p5-udp-01.cap` hasta `p5-udp-06.cap`
  - `p5-tcp-01.cap` hasta `p5-tcp-05.cap`

Práctica 6: Domain Name System (DNS)

Redes de Ordenadores
2º Ingeniería Biomédica

Curso 2022-23

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

©2022
Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
Algunos derechos reservados
Este trabajo se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de
Creative Commons disponible en
http://creativecommons.org/licenses/by-sa/4.0/deed.es

Práctica 6: Domain Name System (DNS)

Resumen

En esta práctica se aprende el funcionamiento básico del DNS. Para su realización es necesario que descargues el fichero del escenario:

`http://mobiquo.gsyc.urjc.es/practicas/ro/p6.html`

NOTA: Para realizar todas las capturas de esta práctica utiliza `tcpdump` tal y como venías usándolo en otras prácticas pero además añade la opción `-n` para que la propia aplicación `tcpdump` no solicite otras resoluciones adicionales al DNS para mostrar la información de forma más amigable.

1. Introducción

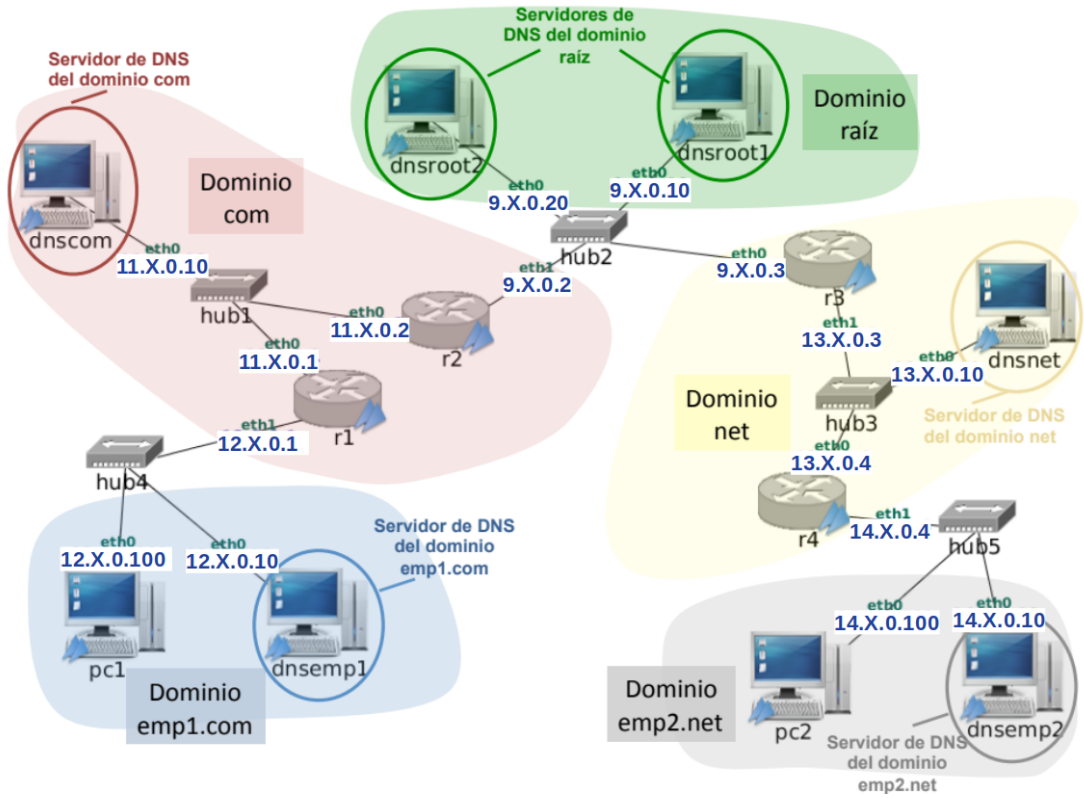


Figura 1: Árbol de dominios

1.1. Árbol de dominios

El escenario definido en `DNS-lab` está formado por 4 *routers* y 8 máquinas. Dentro de este escenario existen los siguientes dominios (véase la figura 1):

- Dominio raíz donde se encuentran las máquinas `dnsroot1` y `dnsroot2`.
- Dominio com: donde se encuentran los *routers* `r1` y `r2` y la máquina `dnscom`. Por tanto, su nombre completo es `r1.com`, `r2.com` y `dnscom.com` respectivamente.
- Dominio `emp1.com`: donde se encuentran las máquinas `pc1` y `dnsemp1`. Por tanto, su nombre completo es `pc1.emp1.com` y `dnsemp1.emp1.com` respectivamente.

- Dominio net: donde se encuentran los *routers* r3 y r4 y la máquina dnsnet. Por tanto, su nombre completo es r3.net, r4.net y dnsnet.net respectivamente.
- Dominio emp2.net: donde se encuentran las máquinas pc2 y dnsemp2. Por tanto su nombre completo es: pc2.emp2.net y dnsemp2.emp2.net respectivamente.

1.2. Servidores de DNS

En las máquinas del escenario que están configuradas como servidor de DNS se utiliza el paquete bind9. Los ficheros de configuración básica de bind9 son los siguientes (se encuentran en la carpeta /etc/bind de cada máquina virtual):

- named.conf:
  Fichero con la configuración general del servidor de DNS: lista de dominios (zonas) para las que el servidor es maestro y/o esclavo y nombres de los ficheros que contienen los mapas de esos dominios. Como ejemplo se muestra a continuación parte del contenido de este fichero en el servidor dnscom:

```
zone "com" {
        type master;
        file "/etc/bind/db.com";
};
```

  El contenido de este fichero indica que la máquina donde se encuentra dicho fichero, dnscom, es servidor maestro del dominio com (todos los nombres de máquinas que terminen en .com) también indica el fichero que almacena el mapa del dominio com, en este caso /etc/bind/db.com.

- `db.root`:

  En el caso de los propios servidores de DNS del dominio raíz este fichero es el que contiene el mapa de dicho dominio raíz (dominio "."). Para el escenario de la práctica, el contenido de `db.root` en `dnsroot1` es:

Mapa del dominio raíz (en dnsroot1)

```
TTL             1d      ; default ttl
.                       IN      SOA     ROOT-SERVER1.   root.ROOT-SERVER1.
(
                                                2009120901 ; serial
                                                8h ; refresh
                                                4h ; retry
                                                1000h ; expire
                                                20m ; negative cache ttl
                                        )
.                       IN      NS      ROOT-SERVER1.
ROOT-SERVER1.           IN      A       9.X.0.10
dnsroot1.               IN      A       9.X.0.10

.                       IN      NS      ROOT-SERVER2.
ROOT-SERVER2.           IN      A       9.X.0.20
dnsroot2.               IN      A       9.X.0.20

com.                    IN      NS      dnscom.com.
dnscom.com.             IN      A       11.X.0.10

net.                    IN      NS      dnsnet.net.
dnsnet.net.             IN      A       13.X.0.10
```

Servidores de DNS del dominio raíz

Servidor de DNS del dominio com

Servidor de DNS del dominio net

  En `dnsroot2` el fichero `db.root` tendrá un contenido similar, modificando los valores del registro SOA.

  En el caso del resto de servidores (`dnscom`, `dnsnet`, `dnsemp1` `dnsemp2`) el fichero `db.root` contiene una relación inicial de IPs de servidores del dominio raíz[1].

Fichero db.root (en los servidores que no son del dominio raíz)

```
.               518400  IN      NS      ROOT-SERVER1.
.               518400  IN      NS      ROOT-SERVER2.

ROOT-SERVER1.   518400  IN      A       9.X.0.10
ROOT-SERVER2.   518400  IN      A       9.X.0.20
```

Servidores de DNS del dominio raíz

- `db.*`:

  Los ficheros que empiezan por `db.` contienen el mapa del dominio que sirve un determinado servidor. Así, el servidor de DNS de `dnsemp1` sirve el mapa del dominio `emp1.com` y por tanto, tiene el fichero `/etc/bind/db.emp1.com` que contiene el mapa del dominio `emp1.com`:

Fichero db.emp1.com

```
$TTL                1d      ; default ttl
emp1.com.               IN      SOA     dnsemp1.emp1.com.  root.dnsemp1.emp1.com. (
                                                2009120901 ; serial
                                                8h ; refresh
                                                4h ; retry
                                                1000h ; expire
                                                20m ; negative cache ttl
                                                )

emp1.com.               IN      NS      dnsemp1.emp1.com.
dnsemp1.emp1.com.       IN      A       12.X.0.10
pc1.emp1.com.       1s  IN      A       12.X.0.100
```

Servidor de DNS del dominio emp1.com

[1] La primera vez que un servidor tenga que enviar un mensaje a un servidor raíz, le enviará otro mensaje más con una consulta preguntando la lista de servidores del dominio raíz, por si hubiera habido cambios

La siguiente tabla muestra las máquinas del escenario en las que se ha configurado `bind` para que sean servidores de DNS:

Máquina	Descripción	Ficheros de configuración
`dnsroot1`	Servidor de nombres raíz	`/etc/bind/named.conf` `/etc/bind/db.root`
`dnsroot2`	Servidor de nombres raíz	`/etc/bind/named.conf` `/etc/bind/db.root`

`dnscom`	Servidor de nombres del dominio `com`	`/etc/bind/named.conf`
		`/etc/bind/db.root`
		`/etc/bind/db.com`

`dnsemp1`	Servidor de nombres del dominio `emp1.com`	`/etc/bind/named.conf`
		`/etc/bind/db.root`
		`/etc/bind/db.emp1.com`

`dnsnet`	Servidor de nombres del dominio `net`	`/etc/bind/named.conf`
		`/etc/bind/db.root`
		`/etc/bind/db.net`

`dnsemp2`	Servidor de nombres del dominio `emp2.net`	`/etc/bind/named.conf`
		`/etc/bind/db.root`
		`/etc/bind/db.emp2.net`

Para ver el mapa de un cierto dominio puedes ejecutar la orden `less`[2] en la máquina que contiene dicho mapa:

```
less <fichero-del-mapa>
```

Así, por ejemplo, para el ver el mapa del dominio `emp2.net`, tienes que escribir en la ventana de terminal de la máquina `dnsemp2.emp2.net` la orden:

```
dnsemp2~:# less /etc/bind/db.emp2.net
```

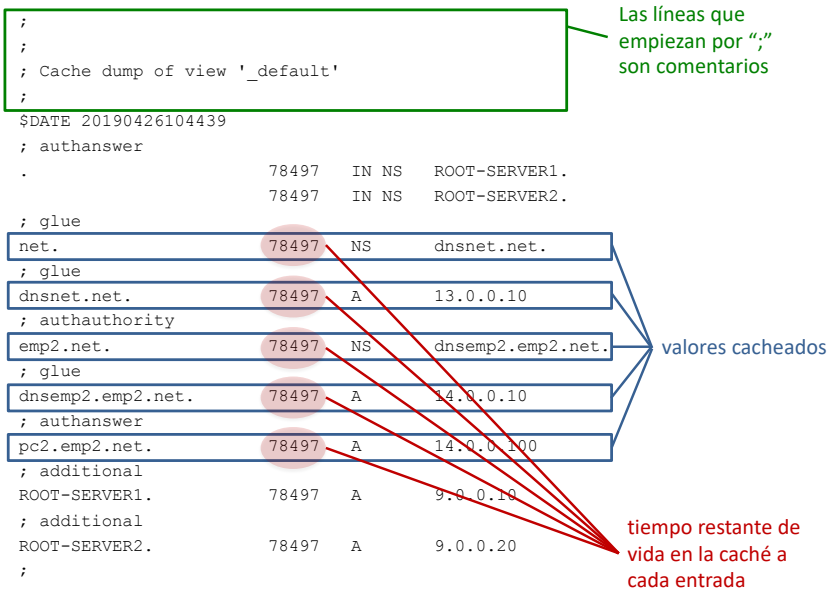
Caché de un servidor DNS

Para ver el contenido de la caché de DNS de un servidor de DNS, ejecuta en su máquina la siguientes dos órdenes, una detrás de otra:

```
rndc dumpdb -cache
less /var/cache/bind/named_dump.db
```

La primera orden vuelca el contenido de la caché de DNS del servidor en el fichero `/var/cache/bind/named_dump.db`, y la segunda te permite consultar su contenido. Un ejemplo de contenido de una caché en un momento dado sería:

[2] `less` es un visor de ficheros de texto, para salir de `less` pulsa `q`



Ten en cuenta que cada vez que quieras ver de nuevo el contenido de la caché debes ejecutar primero la orden `rndc` para actualizar el contenido del fichero, y después ejecutar la orden `less` para mostar el nuevo contenido del fichero.
Para borrar todos los contenidos de la caché de DNS de un servidor, ejecuta en su máquina la orden:

```
rndc flush
```

1.3. Configuración de resolución de nombres en las máquinas

Todas las máquinas del escenario tiene configurado su fichero `/etc/nsswitch.conf` de tal forma que cuando quieran saber la IP que se corresponde con un nombre, primero consultarán su fichero local `/etc/hosts`, y si no encuentran la respuesta, consultarán su servidor de DNS.

Cada máquina tiene configurado su servidor de DNS en su fichero `/etc/resolv.conf`, de la siguiente forma:

- Las máquinas `dnsroot1` y `dnsroot2` tienen cada una configurado como servidor de DNS a ella misma.
- Las máquinas `pc1` y `dnsemp1` tienen configurado como servidor de DNS a `dnsemp1`.
- Las máquinas `pc2` y `dnsemp2` tienen configurado como servidor de DNS a `dnsemp2`.
- La máquina `dnscom` y los *routers* `r1` y `r2` tienen configurado como servidor de DNS a `dnscom`.
- La máquina `dnsnet` y los *routers* `r3` y `r4` tienen configurado como servidor de DNS a `dnsnet`.

1.4. Programa host

Para interrogar al DNS puede utilizarse la orden `host`. Este programa es una herramienta que permite realizar consultas a un servidor de DNS, y lo usaremos este programa de la siguiente forma:

```
host <nombreDeMáquina>
```

El programa `host` devolverá la dirección IP asociada a `<nombreDeMáquina>`, como resultado de haber consultado al servidor de DNS que tenga configurado la máquina donde se ejecuta el programa.

NOTA IMPORTANTE: El programa `host` consulta directamente al DNS, sin mirar nunca el fichero `/etc/hosts`, independientemente del contenido del fichero `/etc/nsswitch.conf`. El resto de órdenes como `ping`, `traceroute`, etc, utilizan dicho fichero, y con la configuración del escenario, primero mirarán en el `/etc/hosts` y luego interrogarán al DNS.

1.5. Formato de los mensajes de DNS

El formato de mensaje de DNS tiene muchos campos. Para la realización de esta práctica consulta las transparencias 41–43 del tema de teoría que contienen resaltados los campos más importantes de los mensajes, que son los que debes intentar localizar en las capturas de `wireshark`.

2. Resolución de nombres

Arranca las máquinas del escenario definido en `DNS-lab` **de una en una** y responde a las siguientes preguntas:

1. Imagina qué ocurriría si la máquina `pc1` ejecuta `host pc2.emp2.net`. ¿Cuántos mensajes de DNS se generarían y entre qué máquinas? Es importante que consideres que es la primera consulta que se realiza en ese escenario (las cachés de los servidores de DNS están vacías).

2. Ejecuta la instrucción anterior en `pc1`, realizando previamente una captura de tráfico en `r1(eth1)` para ver todos los mensajes de DNS generados[3]. Almacena los paquetes de la captura en el fichero `p6-dns-01.cap` ejecutando el comando con la opción `-n`:
   `tcpdump -n -s 0 -i <interfaz> -w <fichero>`[4].

3. Observa en la captura cómo el mensaje de consulta que envía `pc1` tiene activado el flag *Recursion desired* para que la consulta sea recursiva y los mensajes de consulta que envía `dnsemp1` no tienen activado el flag *Recursion desired* para que la consulta se realice de forma iterativa[5].

4. Observa en la/s captura/s el valor TTL (Time To Live) de la respuesta obtenida en `pc1`. NOTA: No confundir el TTL de los mensajes de DNS de respuesta con el TTL de cabecera IP. En esta práctica siempre hablamos del TTL de los mensajes de DNS.

5. Para cada uno de los mensajes de respuesta que observes, explica qué línea/s de cada uno de los mapas de dominio (db.*) proporcionan la información que viaja en dichos mensajes (registros A o registros NS). Para ello mira el contenido de los ficheros de dichos mapas.

6. Supón que ocurriría si después de haber realizado la consulta anterior, en `pc1` se solicita de nuevo la resolución de `pc2.emp2.net`. ¿Cuántos mensajes de DNS se generarían y entre qué máquinas? ¿Por qué?

7. Ejecuta la resolución anterior en `pc1`, realizando de nuevo una captura en `r1(eth1)` y guardando su contenido en el fichero `p6-dns-02.cap` para ver todos los mensajes de DNS generados.

8. Explica el valor TTL (Time To Live) de la respuesta obtenida en `pc1`. Compáralo con el valor obtenido en el apartado 2.

9. Imagina qué mensajes de DNS se generarían y entre qué máquinas si en `pc2` se pide la resolución de `pc1.emp1.com`.

10. Ejecuta la resolución anterior en `pc2`, realizando una captura de tráfico en la interfaz `r4(eth1)` para ver todos los mensajes de DNS generados y guarda su contenido en el fichero `p6-dns-03.cap`.

11. Consulta la caché de DNS en el servidor de DNS de `pc2`, `dnsemp2`. Explica su contenido.

12. Supón que ocurriría si después de haber realizado la consulta anterior, en `pc2` se solicita de nuevo la resolución de `pc1.emp1.com`. ¿Cuántos mensajes de DNS se generarían y entre qué máquinas? ¿Por qué?

[3] Recuerda que si realizas más de una consulta a un servidor de DNS, éste almacena información en su caché. Para borrar la caché de un determinado servidor de DNS ejecuta en dicho servidor la instrucción: `rndc flush`.

[4] Utiliza la opción -n para que tcpdump no intente realizar resoluciones de DNS adicionales a las que genera el comando host.

[5] Observarás en la captura que el servidor además de consultar al servidor de DNS raíz por la resolucióng que se está realizando, como es la primera vez que el servidor envía un mensaje al servidor raíz, le enviará además otro mensaje con una consulta preguntando la lista de servidores del dominio raíz, por si ésta hubiera cambiado.

13. Ejecuta la resolución anterior en `pc2`, realizando una captura tráfico en la interfaz `r4(eth1)` para ver todos los mensajes de DNS generados y guarda su contenido en el fichero `p6-dns-04.cap`. Explica lo sucedido comparado con lo ocurrido en el apartado 7.

14. Consulta la caché de DNS en el servidor de DNS de `pc1`, `dnsemp1`. Explica su contenido.

15. Imagina qué ocurriría si después de haber realizado las consultas anteriores, en `pc1` se solicita la resolución de `r4.net`. ¿Cuántos mensajes de DNS se generarían y entre qué máquinas?

16. Ejecuta la resolución anterior en `pc1`, realizando una captura tráfico en la interfaz `r1(eth1)` para ver todos los mensajes de DNS generados y guarda su contenido en el fichero `p6-dns-05.cap`.

17. El nombre `r4.net` tiene asociadas las dos direcciones IP del *router* `r4`. Comprueba que al solicitar la resolución de `r4.net` sucesivas veces en `pc1`, el orden en el que se obtienen las direcciones IP de `r4` es aleatorio.

18. Imagina qué ocurriría en cada uno de los siguientes casos:

    a) En `pc1` se ejecuta la orden `ping pc200.emp1.com`.

    b) En `pc1` se ejecuta la orden `ping pc200.emp2.net`.

    c) En `pc1` se ejecuta la orden `ping pc20.emp2.net`.

    Para cada uno de los casos responde a las siguientes cuestiones:

    a) ¿Funcionaría el `ping`?

    b) ¿Al ejecutar el `ping` puedes ver la dirección IP asociada al nombre? ¿En qué fichero o mapa está esa asociación de nombre e IP?

    c) ¿Cuántos mensajes de DNS se generarían y entre qué máquinas?

19. Ejecuta las órdenes anteriores, realizando una captura de tráfico en cada caso:

    a) En `pc1` se ejecuta la orden `ping pc200.emp1.com` y se captura tráfico en `r1(eth1)` guardando el tráfico en el fichero `p6-dns-06.cap`.

    b) En `pc1` se ejecuta la orden `ping pc200.emp2.net` y se captura tráfico en `r1(eth1)` guardando el tráfico en el fichero `p6-dns-07.cap`.

    c) En `pc1` se ejecuta la orden `ping pc20.emp2.net` y se captura tráfico en `r1(eth1)` guardando el tráfico en el fichero `p6-dns-08.cap`.

20. Observando los ficheros de configuración de los servidores de DNS, indica qué ocurriría si en `pc1` se solicita por segunda vez la resolución de `pc20.emp2.net`.

21. Ejecuta la resolución anterior en `pc1`, realizando una captura de tráfico en `r1(eth1)` y guardando su contenido en `p6-dns-09.cap`. Indica durante cuanto tiempo se obtendría esta/s misma/s captura/s.

3. Entrega de la práctica

Para entregar la práctica sube a aulavirtual un fichero `p6.zip` que contenga la memoria y las capturas de tráfico de `p6-dns-01.cap` a `p6-dns-09.cap`.

Práctica 7: HTTP

Redes de Ordenadores
2º Ingeniería Biomédica

Curso 2022-23

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

©2022
Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
Algunos derechos reservados
Este trabajo se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de
Creative Commons disponible en
http://creativecommons.org/licenses/by-sa/4.0/deed.es

Práctica 7: HTTP

Cuando un mensaje HTTP ocupa más de un segmento TCP, Wireshark muestra el siguiente mensaje por cada uno de los segmentos TCP que son parte de dicho mensaje HTTP:
`[TCP segment of a reassembled PDU]`
Cuando Wireshark interpreta que se ha recibido todo el mensaje HTTP, como resultado de haber recibido previamente un conjunto de segmentos `TCP segment of a reassembled PDU`, Wireshark concatena todos estos segmentos para mostrar el mensaje HTTP completo.
Por ejemplo, en la figura 1 se puede ver como en el segmento 8 se muestra todo el mensaje HTTP que en realidad viajaba en 3 segmentos TCP: segmento 4, 6 y 8.

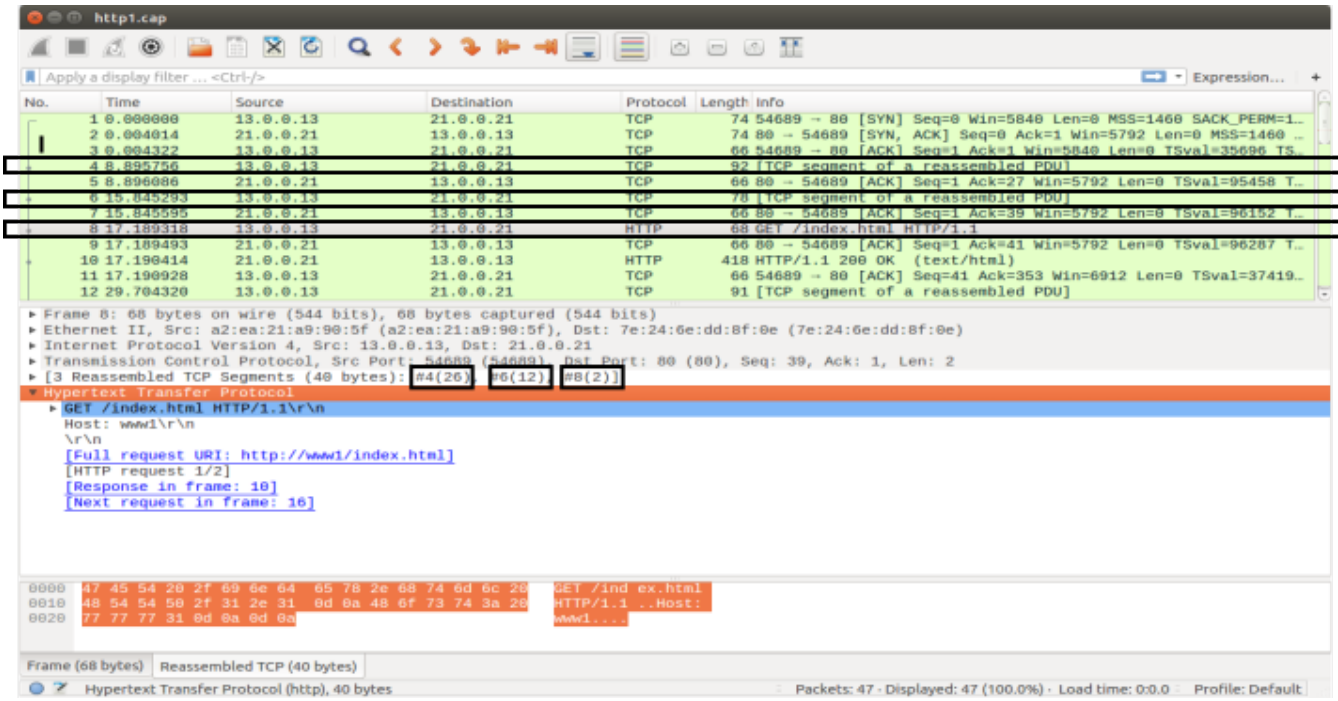


Figura 1: Mensaje HTTP compuesto de varios segmentos.

1. Comunicación cliente-servidor HTTP

Abre la captura `http1.cap` y responde a las siguientes preguntas:

1. Indica qué dirección IP es la de la máquina cliente HTTP y cuál la del servidor.
2. Indica qué versión HTTP están utilizando cliente/servidor
3. Indica el número de conexiones que se ven en el fichero de captura, y si los recursos del mismo servidor se transfieren todos por la misma conexión TCP o se usa conexión TCP diferente para cada uno.
4. ¿Cuántas peticiones GET observas desde el cliente?
5. ¿Cuántas URLs crees que ha escrito el usuario en el navegador para obtener dicha captura? ¿Cuál/es? ¿Por qué?
6. Fíjate en el contenido de la página `index.html` que se ha descargado el cliente. ¿Qué crees que ocurrirá cuando el navegador se haya descargado `index.html`?

Abre la captura `http2.cap` y responde a las siguientes preguntas:

7. Indica qué versión HTTP están utilizando cliente/servidor
8. Indica el número de conexiones que se ven en el fichero de captura, y si los recursos del mismo servidor se transfieren todos por la misma conexión TCP o se usa conexión TCP diferente para cada uno.

2. Diferentes tipos de respuestas de un servidor

Arranca el navegador Firefox. Abre una pestaña nueva y selecciona en el menú de la aplicación → Más herramientas → Herramientas para el desarrollador. La página se habrá dividido en 2 partes. La parte superior es la que muestra normalmente el navegador, la parte inferior contiene información de los mensajes HTTP intercambiados entre cliente y servidor. Selecciona la pestaña 'Red' y 'HTML', véase figura 2.

Figura 2: Herramientas para el desarrollador.

Esta vista del navegador te permitirá cargar una URL y observar todos los mensajes HTTP que se están intercambiando al cargar una página.

Selecciona la pestaña 'Todos', para ver todos los recursos descargados al solicitar una página y explica lo que ocurre al cargar las siguientes URLs:

1. Dentro de esa pestaña carga la página `http://www.google.es/prueba`. Selecciona dentro de las herramientas del desarrollador la petición GET y la pestaña Cabeceras, véase figura 3.

   Fíjate en el campo 'Estado' que indica el tipo de respuesta recibida y explica su contenido.

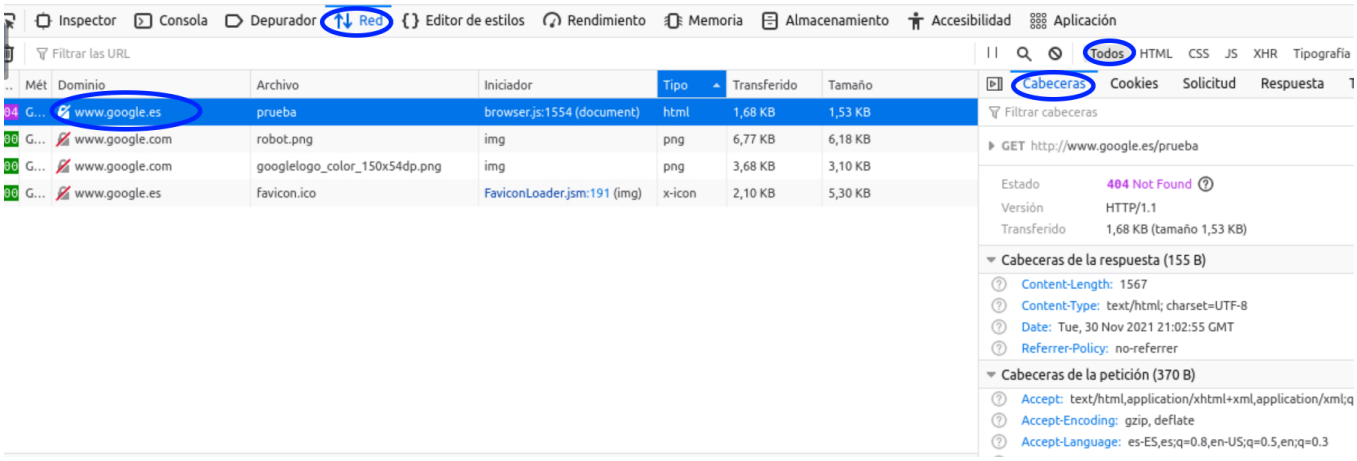


Figura 3: Cabeceras HTTP.

2. En esa misma pestaña carga la página `http://www.wikipedia.com`. Explíca qué ocurre en la primera petición GET y a partir de las líneas de cabecera que ves en la respuesta, explica la segunda petición GET. Fíjate en el lado de la derecha en el "Estado", pulsa en el deslizador "Sin procesar" para ver las cabeceras de la petición y de la respuesta tal y como han sido transmitidas.

3. Formularios en HTTP

Abre la captura `http3.cap` y responde a las siguientes preguntas:

1. Indica el número de conexiones entre cliente y servidor que aparecen en la captura.
2. Busca en la captura el segmento donde el servidor le envía al cliente un formulario. Indica los nombres de los campos del formulario que rellenará el usuario.
3. Indica si es el cliente o el servidor el que decide cómo debe enviar el cliente los datos del formulario (GET/POST) . ¿Qué método están usando en este caso? ¿Cómo lo sabes?
4. Busca en la captura el segmento donde el cliente le envía los datos del formulario al servidor y comprueba que se está realizando con el método GET
5. Fíjate cómo se llama el programa del servidor que va a recibir esos datos.
6. ¿Dónde viajan los datos que el cliente le envía al servidor? ¿Cuáles son esos datos?
7. Indica qué cabecera es la que representa el tipo de contenido del mensaje que el cliente envía al servidor con los datos del formulario.

Abre la captura `http4.cap` y responde a las siguientes preguntas:

9. Busca en la captura el segmento donde el servidor le envía al cliente un formulario. Indica los nombres de los campos del formulario que rellenará el usuario.
10. Indica si es el cliente o el servidor el que decide cómo debe enviar el cliente los datos del formulario (GET/POST) . ¿Qué método están usando en este caso? ¿Cómo lo sabes?

11. Busca en la captura el segmento donde el cliente le envía los datos del formulario al servidor y comprueba que se está realizando con el método POST.

12. Fíjate cómo se llama el programa del servidor que va a recibir esos datos.

13. Indica qué cabecera es la que representa el tipo de contenido que el cliente envía al servidor y cuál es su valor.

14. Indica en qué parte del mensaje van los datos del formulario que el cliente le envía al servidor.

15. Explica si en este caso es necesario la cabecera `Content-Length` en el mensaje HTTP que el cliente envía al servidor con los datos del formulario. ¿Por qué?

16. Observa si el servidor le manda alguna respuesta cuando recibe los datos del formulario del cliente. En caso afirmativo localiza el número de segmento y observa en las cabeceras HTTP: tipo de contenido, longitud y cuerpo del mensaje

4. Cookies

4.1. Almacén de cookies en el navegador Firefox

Para ver las Cookies en el navegador Firefox, selecciona la opción de menú de la aplicación: Editar → Ajustes. En la zona de la izquierda selecciona la pestaña "Privacidad y seguridad". Dentro de la sección "Cookies y datos del sitio" pulsa en "Administrar Datos". Podrás consultar la lista de sitios de los que tienes cookies almacenadas actualmente. Mira si tienes cookies del sitio web del Ayuntamiento de Fuenlabrada `ayto-fuenlabrada.es` (si las tienes, elimina sólo esas cookies). NOTA: En las últimas versiones de Firefox sólo puede saberse si para un sitio hay almacenadas cookies, pero no los datos de las cookies almacenadas.

Abre una pestaña nueva y selecciona en el menú de la aplicación → Más herramientas → Herramientas para el desarrollador. Selecciona la pestaña 'Red' y 'HTML', igual que se muestra en la figura 2.

Dentro de esa pestaña carga la página `http://www.ayto-fuenlabrada.es/`. Selecciona dentro de las herramientas del desarrollador la petición GET y la pestaña Cabeceras, véase la figura 4.

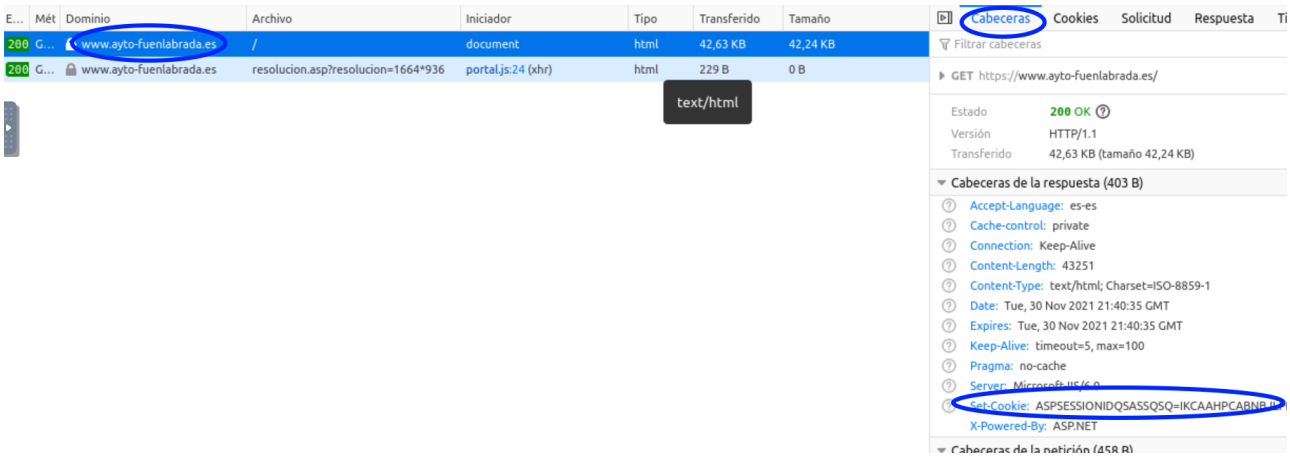


Figura 4: Cookies

1. Fíjate en la línea de cabecera que el servidor le envía al cliente con el contenido de las cookies.

2. Pulsa sobre la pestaña “Cookies” para poder ver de forma más clara el contenido de las cookies. Copia los campos importantes. Fíjate que no hay fecha de expiración, eso quiere decir que la Cookie se eliminará cuando se cierre el navegador.

3. Selecciona ahora la herramienta de desarrollador “Almacenamiento” y en el panel de la izquierda despliega “Cookies” para ver las cookies obtenidas al descargar esta página, véase la figura 5.

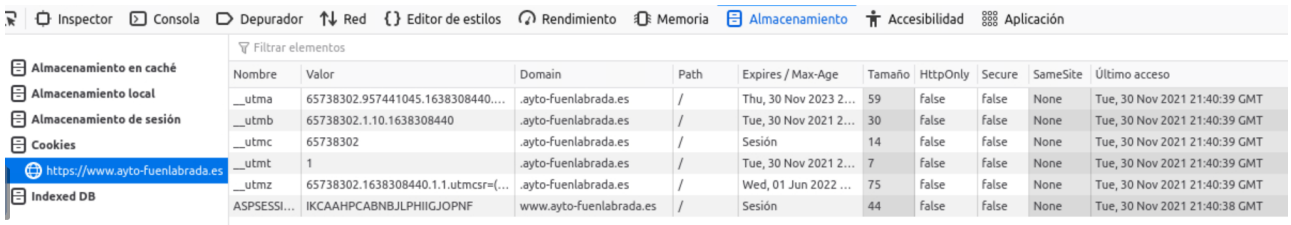


Figura 5: Almacenamiento de Cookies

Señala qué cookies has obtenido para el sitio `ayto-fuenlabrada.es`. Todas las cookies que en este caso comienzan por `__` son debidas a que el sitio web usa Google Analytics, es decir, al descargar la página del Ayuntamiento de Fuenlabrada se ha descargado también una biblioteca en javascript que ha creado estas cookies para este sitio web dentro de nuestro navegador. Estas cookies permiten medir la interacción de los usuarios con el sitio web. No te fijes en esas cookies de Google Analytics.

4. Vuelve a la herramienta de desarrollador “Red” y pulsa sobre la segunda petición GET que aparece, y observa las cookies que se envían. Usa también la pestaña Cookies para poder ver mejor los valores que se envían.

   NOTA: Sólo pueden conocerse los detalles de las cookies que se obtienen del sitio concreto observado con las herramientas del desarrollador[1].

4.2. Envío de Cookies en mensajes HTTP

Abre la captura `http5.cap` y responde a las siguientes preguntas:

1. Indica qué cookies envía el servidor al cliente:

2. Indica qué cookies enviará el cliente al servidor cuando acceda a la página con la URL: `http://elcortebritanico/tienda/index.html`

3. ¿Y si el cliente accediera en el año 2025 a dicha URL?

4. ¿Y si el cliente accediera en el año 2035 a dicha URL?

Abre la captura `http6.cap` y responde a las siguientes preguntas:

5. Indica qué cookies el cliente está enviando al servidor.

6. ¿Por qué crees que le envía dichas cookies?

[1]Si quieres consultar todos los datos de todas las cookies almacenadas en el navegador prueba a instalarte la extensión ”Cookie Quick Manager”de Firefox:
https://addons.mozilla.org/es/firefox/addon/cookie-quick-manager/

7. Escribe un ejemplo de la posibles cabeceras que le habrá enviado dicho servidor al cliente previamente.

8. A partir de la información de la captura ¿crees que si el cliente accede a otra página con la URL: `http://www2/dir1/dir2/index.html` enviaría esas cookies, más o menos?

9. A partir de la información de la captura ¿crees que si el cliente accede a otra página con la URL: `http://www2/index.html` enviaría esas cookies, más o menos?

10. A partir de la información de la captura ¿crees que si el cliente accede a otra página con la URL: `http://www/index.html` enviaría esas cookies, más o menos?

5. Comunicación a través de un Proxy HTTP

Abre la captura `http7.cap` y responde a las siguientes preguntas:

1. Indica qué dirección IP es el cliente, el proxy y el servidor final.

2. ¿Qué diferencia la petición HTTP que realiza el cliente de la petición que realiza el proxy?

3. Identifica el nombre de la máquina donde se encuentra el servidor HTTP.

4. ¿Se puede saber de la petición que realiza el proxy que dicho proxy tiene almacenada en su caché esa página?

Abre la captura `http8.cap` y responde a las siguientes preguntas:

5. Indica el número de conexiones entre cliente y servidor que aparecen en la captura.

6. Explica qué es lo que se está descargando el cliente del servidor HTTP y cuantos objetos se descarga.

7. Observa en las cabeceras HTTP el tipo de contenido de cada uno de los objetos.

8. ¿Podrías saber si los paquetes capturados se corresponde a la comunicación entre un cliente y un proxy HTTP, entre un cliente y servidor final HTTP o entre un proxy y el servidor final HTTP? ¿Por qué?

9. Sabiendo que la comunicación se ha realizado a través de un proxy HTTP, mira las cabeceras HTTP que envía dicho proxy para ver si en ellas existe alguna que muestre cuál es su nombre.

Abre la captura `http9.cap` y responde a las siguientes preguntas:

10. ¿Podrías saber si los paquetes capturados se corresponde a la comunicación entre un cliente y un proxy HTTP, entre un cliente y servidor final HTTP o entre un proxy y el servidor final HTTP? ¿Por qué?

6. Cachés en HTTP

6.1. Caché en un proxy HTTP

Estudia las capturas `http10.cap` y `http11.cap`, teniendo en cuenta que las direcciones 11.0.0.1 y 12.0.0.1 corresponden a la misma máquina. Responde a las siguientes preguntas:

1. Indica cuáles son las direcciones IP del cliente, proxy y servidor web. ¿Cómo lo sabes?
2. Explica qué es lo que ocurre en estas capturas.
3. Localiza los campos relevantes con respecto al tratamiento de caché que incluye en las líneas de cabecera el servidor. ¿Qué significan?
4. Explica qué ocurre en la segunda consulta que realiza el cliente.
5. Viendo los paquetes 14 y 16 de la captura `http10.cap` indica cómo se puede saber que el contenido proviene de una caché.
6. ¿Crees que el cliente tiene caché?

7. Entrega de la práctica

Sube al enlace que encontrarás en `aulavirtual` antes de que termine el plazo de entrega, un único fichero `p6.pdf` con la memoria de la práctica **en formato PDF**.