Proyectos de prácticas

Sistemas Telemáticos para Medios Audiovisuales
2º Grado en Ingeniería en Sistemas Audiovisuales y Multimedia

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)

Contenido

Sistemas Telemáticos para Medios Audiovisuales
Práctica 1: Dispositivos de Interconexión

GSyC
Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación

Septiembre de 2022

Para esta práctica, cada alumno tendrá escenarios diferentes en cada apartado. En particular, las direcciones IP de las máquinas tendrán asignado en el segundo byte un valor X distinto. Podrás ver qué valor X tienes asignado cuando cargues el escenario en NetGUI y observes la configuración.

Antes de comenzar a realizar la práctica, por favor, descarga tus escenarios del siguiente enlace donde deberás introducir tu número de DNI (8 dígitos) con la letra correspondiente:

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

1. Funcionamiento de hubs y switch

En el fichero `lab-hub-switch.tgz` está definida una red como la de la figura 1. Descomprime el fichero (con `tar -xvzf lab-hub-switch.tgz`), arranca NetGUI y abre el escenario.

No arranques aún s1.

Arranca el resto de las máquinas de una en una.

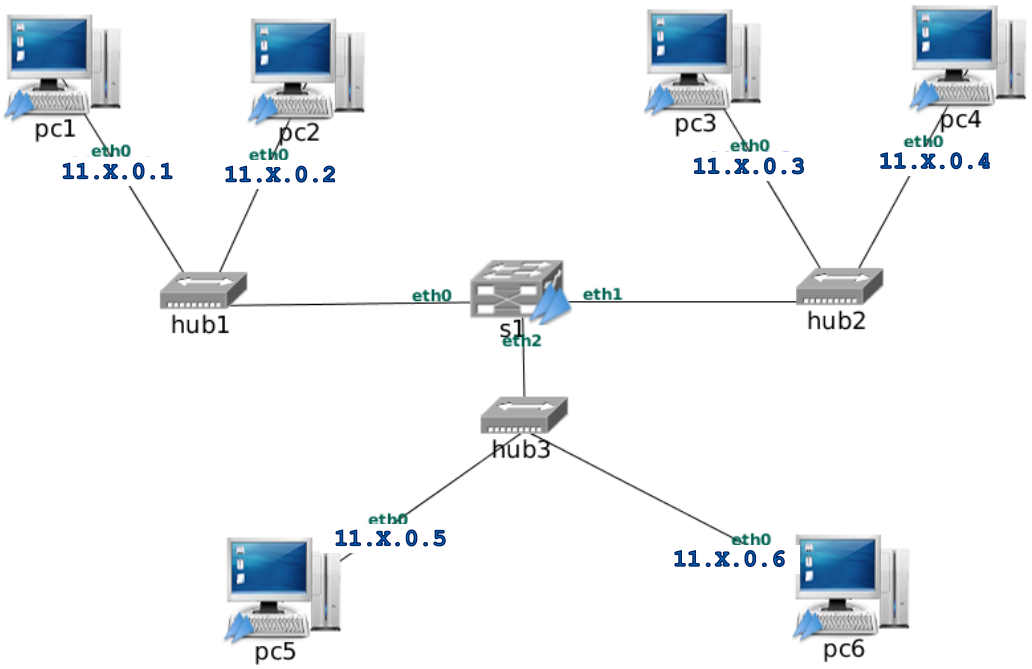

Figura 1: Escenario de hubs y switch

Deja por ahora sin arrancar el *switch* `s1`. Cada uno de los *hubs* estará aislado de los demás. Por lo tanto sólo habrá conectividad entre los ordenadores que están conectados al mismo *hub*. Las tramas Ethernet no pueden salir del *hub* en el que aparecen.

1.1. Comunicación entre máquinas con s1 apagado

NOTA: Las capturas a realizar en este apartado no es necesario redirigirlas a un fichero para estudiarlas con `wireshark`. Basta con ver la salida de `tpcdump` directamente en el terminal de cada máquina virtual, escribiendo: `tcpdump -i eth0`.

1. Piensa en qué paquetes se capturarán en `pc2`, `pc3` y en `pc5` si se hace un `ping` desde `pc1` a `pc2`.
2. Lanza `tcpdump` en `pc2`, `pc3` en `pc5`. A continuación ejecuta la siguiente orden en `pc1` para hacer un `ping` a `pc2`[1]:

   ```
   pc1:~# ping -c 3 11.X.0.2
   ```

   (`-c 3` hace que el `ping` sólo envíe 3 paquetes ICMP)

   Observa el tráfico capturado en `pc2`, `pc3` y `pc5` y comprueba si ha ocurrido lo que pensabas. Copia en la memoria lo que muestra `tcpdump` en cada una de las máquinas.
3. Comprueba que no existe conectividad (es decir, que no puede hacerse `ping`) entre máquinas que estén en diferentes *hubs*.

1.2. Comunicación entre máquinas con s1 arrancado

1. Arranca el *switch* `s1`.
2. Piensa en qué paquetes se capturarán ahora en `pc2`, `pc3` y en `pc5` repitiendo el mismo `ping`
3. Comprueba la caché de ARP en `pc1`. Si aún está en ella la dirección Ethernet de `pc2` borra esa entrada de la caché de ARP.
4. Lanza `tcpdump` en `pc2` (guarda la captura en un fichero `hub-switch-01.cap`), `pc3` (guarda la captura en un fichero `hub-switch-02.cap`) y en `pc5` (guarda la captura en un fichero `hub-switch-03.cap`). A continuación vuelve a hacer en `pc1` el `ping` a `pc2`:

   ```
   pc1:~# ping -c 3 11.X.0.2
   ```

   Interrumpe las capturas y observa el tráfico capturado en `pc2`, `pc3` y `pc5` y comprueba si ha ocurrido lo que pensabas.
5. Responde a estas preguntas:
   - ¿Por qué llega a `pc3` y a `pc5` la solicitud de ARP enviada por `pc1`?
   - ¿Por qué NO llega a `pc3` y a `pc5` la respuesta de ARP enviada por `pc2`?
   - ¿Por qué NO llega a `pc3` y a `pc5` el *ICMP echo request* enviado por `pc1`?
   - ¿Por qué NO llega a `pc3` y a `pc5` el *ICMP echo reply* enviado por `pc2`?
6. Comprueba las direcciones Ethernet que tiene cada interfaz de cada máquina de la figura (usando `ifconfig`), y apúntalas en la memoria.
7. Mira la tabla de direcciones aprendidas por el *switch* `s1` utilizando la orden `brctl showmacs s1`. Puedes utilizarla junto con la orden `watch` para observar periódicamente los cambios en las direcciones aprendidas:

   ```
   s1:~# watch brctl showmacs s1
   ```

   (`watch` repite cada 2 segundos la ejecución de la orden que se le pasa como parámetro)

   Identifica las máquinas a las que pertenece cada dirección Ethernet y explica su presencia en la tabla de direcciones aprendidas de `s1`.

   Tras 300 segundos comprobarás que el *switch* olvida las direcciones aprendidas (mira cómo va creciendo el valor de la columna *ageing timer*, contador de envejecimiento, en la salida de la orden). Comprueba también cómo el *ageing timer* de una dirección Ethernet se reinicializa cada vez que el *switch* ve una nueva trama con esa dirección Ethernet.
8. Comprueba que ahora sí existe conectividad entre todas las máquinas de la figura utilizando la orden `ping`.

[1] Fíjate en el valor que tienes asignado en tu escenario a X para ejecutar correctamente el comando

2. Redes conectadas a través de switch y router

En el fichero `lab-switch-router.tgz` está definida una red como la que aparece en la figura 2. Descomprime el fichero, lanza NetGUI y abre el escenario. Arranca todas las máquinas: pcs, routers y switches.

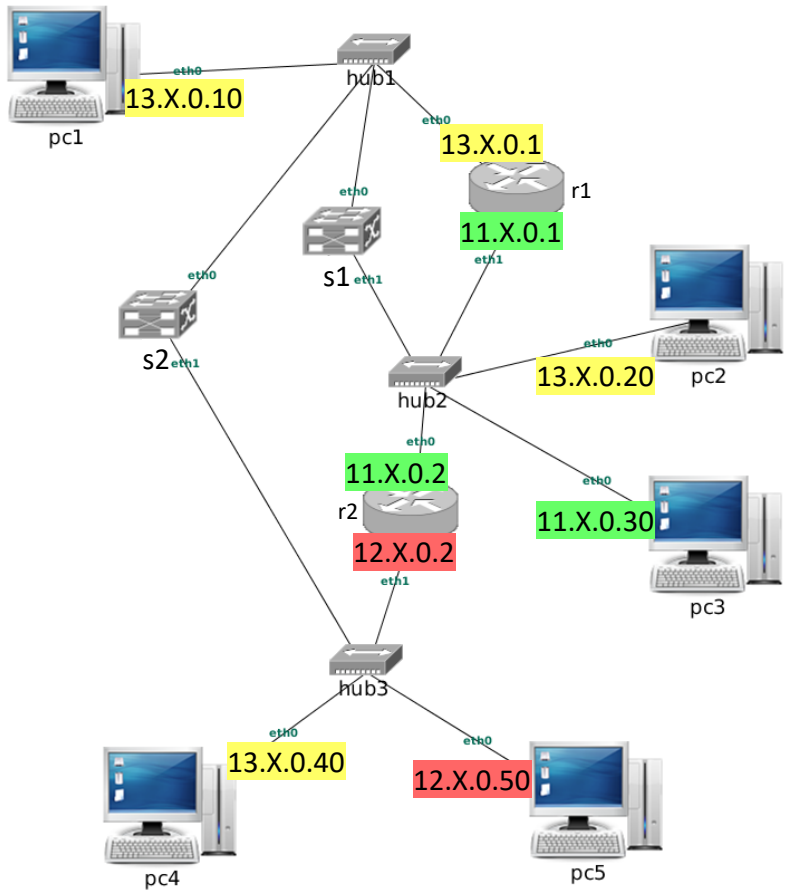

Figura 2: Escenario de redes conectadas por *switches* y *routers*

2.1. Comunicación entre pc2 y pc4

Con las cachés de ARP vacías y las tablas de direcciones aprendidas de los switches vacías se desea realizar un `ping` de `pc2` a `pc4`:

1. Observa la configuración que hay en el escenario para que `pc2` y `pc4` puedan intercambiar tráfico. ¿Cuál de los siguientes caminos crees que seguirán los mensajes ICMP echo request desde `pc2` a `pc4`?

   - `pc2` → `s1` → `s2` → `pc4`
   - `pc2` → `r1` → `s2` → `pc4`
   - `pc2` → `r2` → `pc4`

   Justifica la respuesta.

2. Indica cuántas solicitudes y respuestas de ARP serían necesarias para que dicho ping funcionase. Explica en qué pcs/routers/switches y su interfaz `eth` concreta se podrían capturar:

   - solicitud/es de ARP.
   - respuesta/s de ARP.

3. Para ver todo el tráfico generado deberás lanzar un `tcpdump` por cada `hub` de la figura. Justifica la respuesta.

4. Lanza `tcpdump` en las máquinas `pc1` (`switch-router-01.cap`), `pc3` (`switch-router-02.cap`) y `pc5` (`switch-router-03.cap`) para ayudarte a comprobar tus suposiciones [2].

5. Indica qué direcciones Ethernet habrán aprendido `s1` y `s2` después de ejecutar el `ping` y explica qué mensajes han generado dicho aprendizaje. Compruébalo.

6. ¿Crees que habrá llegado alguno de los mensajes ICMP echo request a `pc1`, `pc3` o `pc5`? Justifica la respuesta.

2.2. Comunicación entre pc1 y pc3

Con las cachés de ARP vacías y las tablas de direcciones aprendidas de los switches vacías se desea realizar un `ping` de `pc1` a `pc3`:

1. Observa la configuración que hay en el escenario para que `pc1` y `pc3` puedan intercambiar tráfico. ¿Cuál de los siguientes caminos crees que seguirán los mensajes ICMP echo request desde `pc1` a `pc3`?

   - `pc1` → `r1` → `pc3`
   - `pc1` → `s1` → `pc3`
   - `pc1` → `s2` → `r2` → `pc3`

   Justifica la respuesta.

2. Indica cuántas solicitudes y respuestas de ARP serían necesarias para que dicho ping funcionase. Explica en qué pcs/routers/switches y su interfaz `eth` concreta se podrían capturar:

   - solicitud/es de ARP.
   - respuesta/s de ARP.

3. Lanza `tcpdump` en las máquinas `r1(eth0)` (`switch-router-04.cap`), `pc2` (`switch-router-05.cap`) y `pc5` (`switch-router-06.cap`) para ayudarte a comprobar tus suposiciones.

4. Indica qué direcciones Ethernet habrán aprendido `s1` y `s2` después de ejecutar el `ping` y explica qué mensajes han generado dicho aprendizaje. Compruébalo.

5. ¿Crees que habrá llegado alguno de los mensajes ICMP echo request a `pc2`, `pc4` o `pc5`? Justifica la respuesta.

2.3. Comunicación entre pc2 y pc5

Con las cachés de ARP vacías y las tablas de direcciones aprendidas de los switches vacías se desea realizar un `ping` de `pc2` a `pc5`:

1. Observa la configuración que hay en el escenario para que `pc2` y `pc5` puedan intercambiar tráfico. ¿Cuál de los siguientes caminos crees que seguirán los mensajes ICMP echo request desde `pc2` a `pc5`?

   - `pc2` → `r2` → `pc5`
   - `pc2` → `r1` → `s2` → `pc5`
   - `pc2` → `r2` → `s1` → `s2` → `pc5`
   - `pc2` → `r2` → `r1` → `s2` → `pc5`
   - `pc2` → `s1` → `r1` → `r2` → `pc5`

   Justifica la respuesta.

2. Indica cuántas solicitudes y respuestas de ARP serían necesarias para que dicho ping funcionase. Explica en qué pcs/routers/switches y su interfaz `eth` concreta se podrían capturar:

   - solicitud/es de ARP.
   - respuesta/s de ARP.

[2] Recuerda que esta prueba necesita que las tablas de direcciones aprendidas y las cachés de ARP estén vacías. Para borrar las tablas de direcciones aprendidas de un switch puedes desactivar y activar el switch ejecutando, por ejemplo, el comando 'ifconfig s1 down' y a continuación ejecutar 'ifconfig s1 up'. Puedes comprobar cómo la tabla de direcciones aprendidas está vacía. Las cachés de ARP de las máquinas se comprueban ejecutando 'arp -a' y se pueden borrar cada una de sus entradas ejecutando 'arp -d direcciónIP'

3. Lanza `tcpdump` en las máquinas `pc1` (`switch-router-07.cap`), `pc3` (`switch-router-08.cap`) y `pc4` (`switch-router-09.cap`) para ayudarte a comprobar tus suposiciones.

4. Indica qué direcciones Ethernet habrán aprendido `s1` y `s2` después de ejecutar el `ping` y explica qué mensajes han generado dicho aprendizaje. Compruébalo.

5. ¿Crees que habrá llegado alguno de los mensajes ICMP echo request a `pc1`, `pc3` o `pc4`? Justifica la respuesta.

3. Proxy ARP

En el fichero `lab-proxyARP.tgz` está definida una red como la que aparece en la figura 3. Descomprime el fichero, lanza NetGUI y abre el escenario. Arranca las máquinas de una en una.

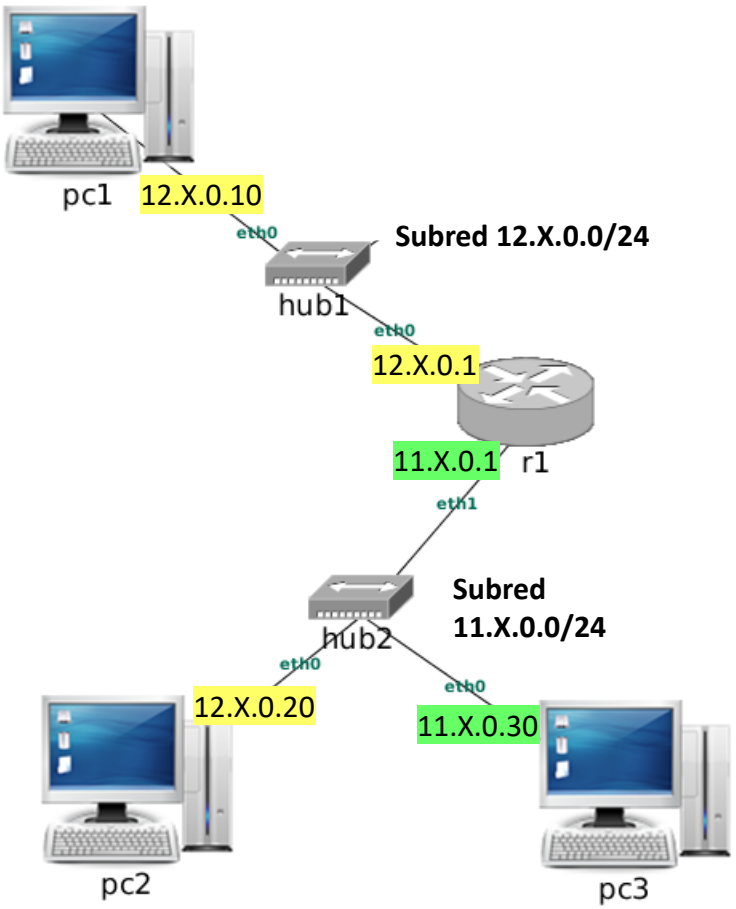

Figura 3: Escenario de Proxy ARP

Características del escenario:

- Los pcs y el router `r1` están configurados con las direcciones IP que se muestran en la figura.
- En el fichero `/etc/hosts` de cada pc están los nombres y direcciones IP de `pc1`, `pc2` y `pc3`, por lo que puedes referirte a ellos por su nombre además de por su IP en las órdenes que utilices.

1. Activa proxy ARP en la configuración del router `r1` para que las máquinas `pc1` y `pc2` tengan conectividad IP entre ellas en ambos sentidos. Explica qué modificaciones han sido necesarias y por qué.

2. Con las cachés de ARP vacías, realiza una captura en la interfaz `r1(eth0)` (`proxyARP-01.cap`) y en `pc3` (`proxyARP-02.cap`) y ejecuta un `ping` desde `pc1` a la dirección IP de `r1(eth0)`, enviando sólo 3 paquetes, y después un `ping` desde `pc1` a `pc2`, enviando sólo 3 paquetes. Interrumpe la captura y explicalas solicitudes de ARP que ves en el tráfico capturado en ambos ficheros.

3. A partir de la captura y de las direcciones IP de `r1`, ¿cómo puedes saber que `r1` está realizando proxy ARP?

4. Si se ha borrado la caché de ARP de `pc1` vuelve a ejecutar los 2 `ping` anteriores y consulta la caché de ARP de `pc1`, indica qué observas, explicando a qué máquina/s pertenece la información almacenada.

4. IP aliasing

En el fichero `lab-ipAliasing.tgz` está definida una red como la que aparece en la figura 4. Descomprime el fichero, lanza NetGUI y abre el escenario. Arranca las máquinas de una en una.

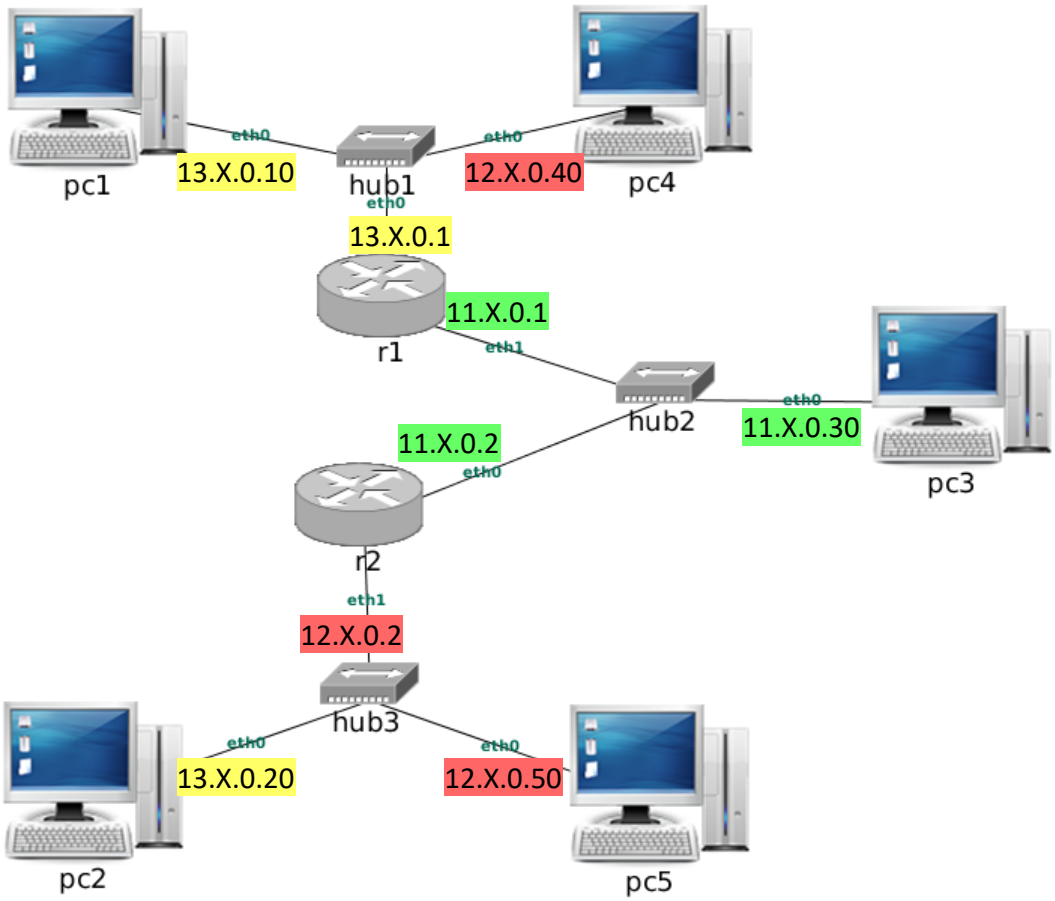

Figura 4: Escenario de IP Aliasing

Características del escenario:

- En el fichero `/etc/hosts` de cada pc están los nombres y direcciones IP de `pc1`, `pc2`, `pc3`, `pc4` y `pc5`, por lo que puedes referirte a ellos por su nombre además de por su IP en las órdenes que utilices.
- `pc1`, `pc3` y `pc5` tienen conectividad IP entre ellos. `pc2` y `pc4` no tienen conectividad IP, ya que no están conectados a sus respectivas subredes.

1. Asigna **direcciones IP adicionales** en los routers mediante *IP aliasing*, y configura las tablas de encaminamiento que sean necesarias para que `pc2` pueda hacer `ping` a `pc3`, ten en cuenta que desde `r2` se debería poder alcanzar también a `pc1` [3]. Indica por qué has configurado esas direcciones IP adicionales y en qué interfaces.

[3] Nótese que cuando añades una dirección por IP aliasing a una tabla de encaminamiento se añade automáticamente una entrada para la subred a la que pertenece, entrada que a veces es necesario borrar para que no haya en la misma tabla dos rutas diferentes a la misma subred.

2. Realiza una captura en `r2(eth1)` (`ipAliasing-01.cap`) y ejecuta un `ping` desde `pc2` a `pc3` enviando 3 paquetes y después ejecuta un `ping` desde `pc5` a la dirección IP de `r2(eth1)`. Interrumpe la captura y explica las solicitudes de ARP que observas.

3. ¿Se puede saber sólo mirando el fichero de captura que en `r2` no se ha configurado proxy ARP?

4. Con la configuración que has realizado previamente ¿pueden comunicarse `pc1` y `pc5`? ¿Por qué? Si tu respuesta es negativa, modifica la configuración para que `pc5` y `pc1` puedan intercambiar tráfico.

5. Utiliza de nuevo *IP aliasing* para que `pc4` pueda hacer `ping` a `pc1`, ten en cuenta que desde `r1` se debería poder alcanzar también a `pc5`.

6. Realiza una captura en `r1(eth0)` (`ipAliasing-02.cap`) para ver qué paquetes se intercambian cuando `pc4` hace `ping` a `pc1`. Explica los resultados en la memoria.

5. VLANs

En el fichero `lab-vlan.tgz` está definida la topología de una red como la de la figura 5 en la que aún no se han configurado las VLANs. Descomprime el fichero, arranca NetGUI y abre el escenario. Arranca las máquinas de una en una.

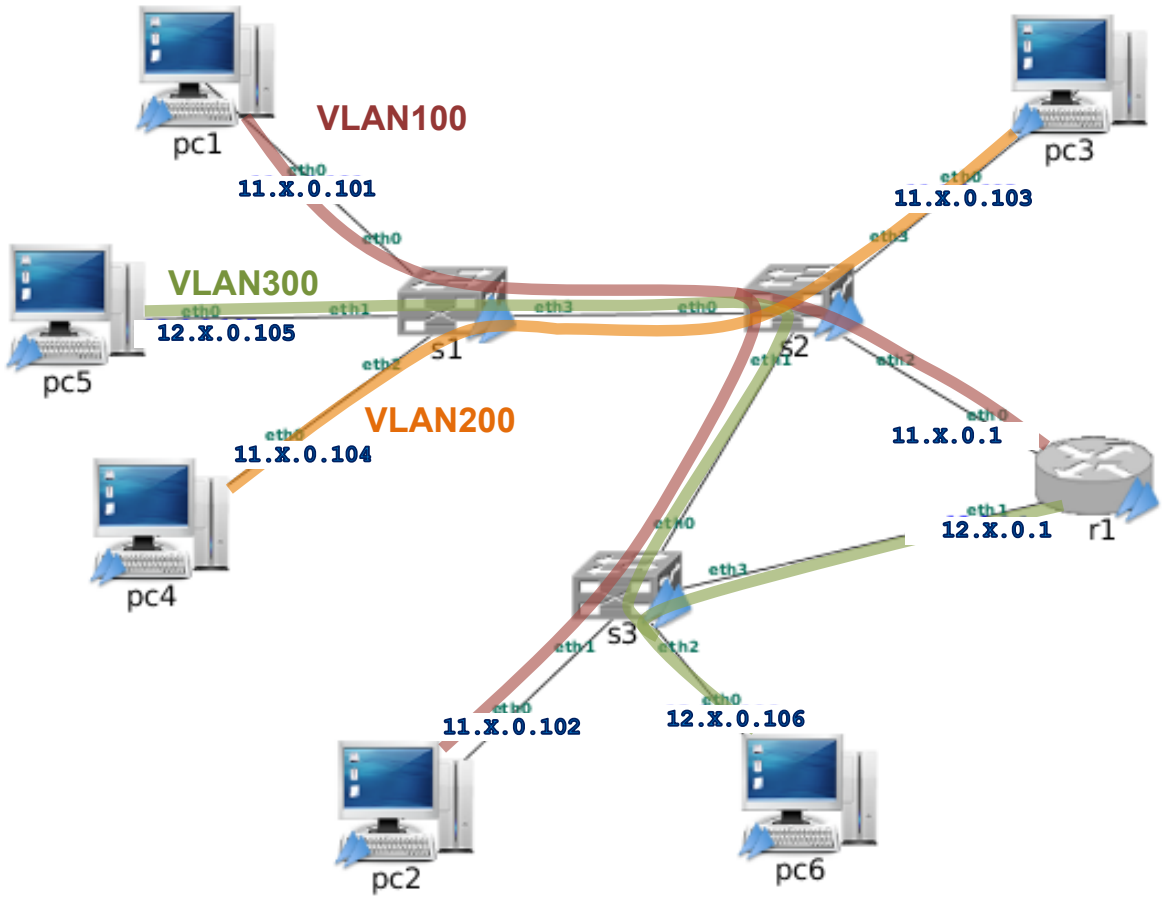

Figura 5: Escenario de VLANs

Los dispositivos de interconexión `s1`, `s2` y `s3` están configurados para que funcionen como *switches* Ethernet.

1. Explica qué máquinas se pueden comunicar entre ellas.

   Compruébalo realizando `ping`.

2. Suponiendo que la caché de ARP de pc1 está vacía, indica dónde se puede capturar un solicitud de ARP que la máquina `pc1` envía preguntando por la dirección Ethernet de la máquina `pc2`.

Compruébalo realizando capturas. Para este caso puedes utilizar `tcpdump -i <interfaz> -s 0` sin necesidad de guardar la captura en un fichero, de esta forma verás el resultado mostrado en pantalla. (Comprueba antes que en la caché de ARP de `pc1` no se encuentra la dirección Ethernet de `pc2`; si estuviera, bórrala).

5.1. Configuración de VLAN100

Para facilitar la configuración de las VLANs en cada switch se propone que esta configuración quede almacenada en un fichero de script. Un script es un fichero que contiene comandos que se ejecutarán en el intérprete de comandos, tal y como si los tecleáramos en el terminal.

La configuración de la VLAN100 está escrita en los ficheros `vlan-s1.sh`, `vlan-s2.sh` y `vlan-s3.sh`, que se encuentran en `s1`, `s2` y `s3` respectivamente.

1. Estudia estos scripts para entender qué hace cada uno de ellos. Observa que la primera línea `#!/bin/bash` indica el intérprete que va a ejecutar este script, en este caso `bash`. El resto de líneas en el fichero que comienzan por `#` son comentarios. Cada uno de los comandos que se desean ejecutar se escriben en líneas diferentes [4].

2. Ejecuta los scripts para aplicar la configuración. Debes ejecutar cada uno de esos scritps en su switch, por ejemplo en `s1`:

   ```
   s1:~# ./vlan-s1.sh
   ```

   Puedes comprobar la configuración que tiene en un *switch* escribiendo `brctl show`.

3. Haz un dibujo de cada switch que muestre las interfaces que intervienen en la VLAN100, indicando si estas interfaces llevan o no etiqueta VLAN.

4. Indica qué máquinas se pueden comunicar entre ellas.

5. Suponiendo que la caché de ARP de `pc1` está vacía, indica dónde se puede capturar un solicitud de ARP que la máquina `pc1` envía preguntando por la dirección Ethernet de la máquina `pc2`.

   Compruébalo realizando las capturas que creas necesarias, sin necesidad de guardar en fichero el tráfico capturado. (Comprueba antes que en la caché de ARP de `pc1` no se encuentra la dirección Ethernet de `pc2`, si estuviera, bórrala).

6. Indica qué ocurre cuando se hace un `ping` desde `pc1` a `pc2`, teniendo en cuenta que ambas máquinas se encuentran en la misma subred. Compruébalo realizando las capturas necesarias, sin necesidad de guardar en un fichero el tráfico capturado.

7. Asegúrate de que la caché de ARP de `pc1` está vacía, bórrala si es necesario. Arranca `tcpdump` en las siguientes interfaces: `pc1(eth0)` (`vlan-01.cap`), `s1(eth3)` (`vlan-02.cap`), `s2(eth2)` (`vlan-03.cap`), `s3(eth0)` (`vlan-04.cap`) y `pc2(eth0)` (`vlan-05.cap`), guardando esta vez el tráfico capturado en un fichero. Realiza un `ping` desde `pc1` a `pc2`.

8. Interrumpe las capturas. Observa las direcciones Ethernet aprendidas por `s1`, `s2` y `s3`.

9. Analiza las 5 capturas, indica en qué capturas se observa la etiqueta de VLAN en el tráfico y qué identificador de VLAN contiene.

   a) ¿Qué *switch* introduce dicha etiqueta?

   b) ¿Qué *switch* elimina dicha etiqueta?

   c) ¿`pc1` y `pc2` tienen alguna forma de saber si están usando una VLAN para comunicarse?

   d) ¿Por qué sólo se ve una trama Ethernet en la captura realizada en la interfaz `s2(eth2)`?

[4] Los comandos que se han escrito para desactivar y borrar el switch terminan con '2 > /dev/null' que significa que si se produce algún error al ejecutar el comando, ese error no se muestra. Esto es necesario porque si ejecutamos sucesivas veces este script, la primera vez que se ejecutó, se desactivó y eliminó el switch y en las sucesivas veces que se ejecute dicho script el switch no existirá y se mostraría un error al desactivarlo y eliminarlo.

e) ¿En qué se diferencia la solicitud de ARP que se captura en `pc1(eth0)` de la misma solicitud que se captura en `s1(eth3)`?

f) ¿En qué se diferencia el mensaje ICMP Echo request que se captura en `pc1(eth0)` del mismo mensaje que se captura en `s1(eth3)`?

10. Indica qué ocurre cuando se hace un `ping` desde `pc1` a `pc4`, teniendo en cuenta que ambas máquinas se encuentran en la misma subred y conectadas al mismo *switch*. Compruébalo realizando una captura en `pc1(eth0)` (`vlan-06.cap`) y otra en `s1(eth3)` (`vlan-07.cap`). Explica los resultados.

5.2. Configuración de VLAN200

Configura la VLAN200 en los *switches* que creas necesarios. Para ello edita los ficheros de configuración proporcionados en el apartado anterior y añade la configuración de VLAN 200.

Antes de ejecutar la nueva configuración es necesario borrar la anterior, para ello, reinicia los switches `s1` y `s2`, y a continuación ejecuta sus scripts modificados.

Puedes comprobar la configuración que tiene cada *switch* escribiendo `brctl show`.

1. Haz un dibujo de cada switch que muestre las interfaces que intervienen en la VLAN200, indicando si estas interfaces llevan o no etiqueta VLAN.

2. Indica qué máquinas se pueden comunicar entre ellas con la configuración de VLAN200.

3. Asegúrate de que la caché de ARP de `pc4` está vacía, bórrala si es necesario. Arranca `tcpdump` en las siguientes interfaces: `pc4(eth0)` (`vlan-08.cap`), `s1(eth3)` (`vlan-09.cap`), `pc3(eth0)` (`vlan-10.cap`) y `pc1(eth0)` (`vlan-11.cap`) guardando esta vez el tráfico capturado en un fichero. Realiza un `ping` desde `pc4` a `pc3`.

4. Interrumpe las capturas y observa las direcciones Ethernet aprendiadas por los switches `s1`, `s2` y `s3`. Explica el resultado.

5. Analiza las 4 capturas, indica en qué capturas se observa la etiqueta de VLAN en el tráfico y qué identificador de VLAN contiene.

6. Indica qué ocurre ahora cuando se hace un ping desde `pc1` a `pc4`, teniendo en cuenta que ambas máquinas se encuentran en la misma subred, conectadas al mismo *switch* y las interfaces de dicho *switch* tienen configurada una VLAN. Compruébalo realizando una captura en `pc1` (`vlan-12.cap`) y otra en `pc4` (`vlan-13.cap`). Explica el resultado.

5.3. Configuración de VLAN300

En este apartado se analiza el comportamiento de 2 VLANs que están conectadas a través de un router. Esta configuración se proporciona en unos scripts que ya se encuentran en el escenario: `vlan100y300-s1.sh`, `vlan100y300-s2.sh` y `vlan100y300-s3.sh`, en `s1`, `s2` y `s3` respectivamente.

Antes de ejecutar la nueva configuración es necesario borrar la anterior, para ello, reinicia todos los switches y a continuación ejecuta cada uno de los scripts anteriores.

Puedes comprobar la configuración que tiene en un *switch* escribiendo `brctl show`.

1. Haz un dibujo de cada switch que muestre las interfaces que intervienen en la VLAN300, indicando si estas interfaces llevan o no etiqueta VLAN.

2. Realiza un ping desde `pc6` a `pc1`. ¿Qué crees que está ocurriendo?

3. Realiza un ping desde `pc6` a `pc5`. ¿Qué crees que está ocurriendo?

4. Suponiendo que la caché de ARP de `pc6` está vacía, al realizar un `ping` de `pc6` a `pc1`, ¿qué solicitudes de ARP hay y en qué interfaces aparecen? ¿Cuáles de ellas tendrán etiqueta VLAN e indica qué etiqueta?

   Compruébalo realizando las capturas que creas necesarias, sin necesidad de guardar en fichero el tráfico capturado. (Comprueba antes que las cachés de ARP de `pc6` y de `r1` están vacías, bórralas si es necesario).

5. Asegúrate de que las cachés de ARP de `pc6` y `r1` están vacías, bórralas si es necesario. Arranca `tcpdump` en las siguientes interfaces: `pc6(eth0)` (`vlan-14.cap`), `s3(eth1)` (`vlan-15.cap`), `r1(eth0)` (`vlan-16.cap`), `s2(eth0)` (`vlan-17.cap`) y `pc1(eth0)` (`vlan-18.cap`), guardando esta vez el tráfico capturado en un fichero. Realiza un `ping` desde `pc6` a `pc1`.

   Supón en qué interfaces aparecerá el tráfico etiquetado y su identificador de VLAN. Comprueba tus suposiciones analizando las capturas, indica en qué capturas se observa la etiqueta de VLAN en el tráfico y qué identificador de VLAN contiene.

6. Entrega de la práctica

Sube al enlace que encontrarás en `aulavirtual` antes de que termine el plazo de entrega, los siguientes ficheros:

- Memoria en formato pdf donde se explique razonadamente la resolución de cada uno de los apartados de este enunciado.
- Fichero de nombre `p1.zip` o `p1.tgz` resultado de comprimir **una carpeta de nombre p1** que contenga en su interior todos los ficheros de captura de tráfico:
  - De `hub-switch-01.cap` a `hub-switch-03.cap`.
  - De `switch-router-01.cap` a `switch-router-09.cap`.
  - `proxyARP-01.cap` y `proxyARP-02.cap`.
  - `ipAlising-01.cap` e `ipAlising-02.cap`.
  - De `vlan-01.cap` a `vlan-18.cap`.

  Puedes crear el fichero de esta forma: primero crea una carpeta de nombre `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'`.

Sistemas Telemáticos para Medios Audiovisuales
Práctica 2: Protocolos de Encaminamiento: OSPF

GSyC
Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación

Septiembre de 2022

Resumen

En esta práctica, cada alumno tendrá escenarios diferentes en cada apartado. En particular, las direcciones IP de las máquinas tendrán asignado en el segundo byte un valor X distinto. Podrás ver qué valor X tienes asignado cuando cargues el escenario en NetGUI y observes la configuración.

Antes de comenzar a realizar la práctica, por favor, descarga tus escenarios del siguiente enlace donde deberás introducir tu número de DNI (8 dígitos) con la letra correspondiente:

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

Para la realización de estos ejercicios se utilizará el paquete de software *quagga* que permite estudiar el funcionamiento del protocolo OSPF. En la documentación adicional se explica cómo se configura el software *quagga* en Linux.

1. OSPF: todos los routers en la misma área

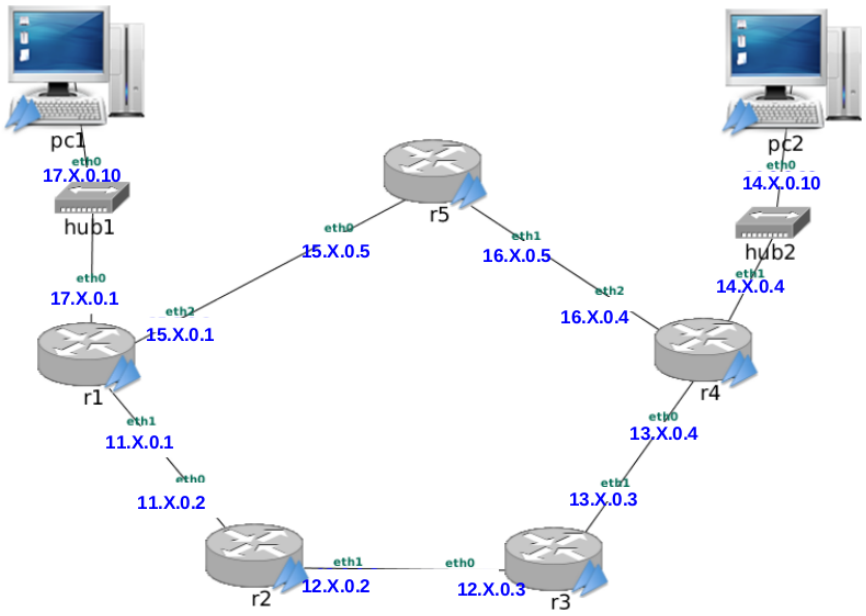

Figura 1: Diagrama de red para el protocolo OSPF

1. En el fichero `lab-OSPF.tgz` está definida una red como la que se muestra en la figura 1. Descomprime el fichero de configuración del escenario `lab-OSPF.tgz`. Al arrancar NetGUI debes abrir el escenario definido en el directorio `lab-OSPF`.

2. Arranca todas las máquinas de una en una. Las máquinas `pc1` y `pc2` tienen rutas por defecto a `r1` y `r4` respectivamente. Los *routers* no tienen configurada ninguna ruta, salvo la de las subredes a las que están directamente conectados. Compruébalo con la orden `route`.

   Los routers no tienen ningún de ellos arrancado `quagga` ni configurado OSPF. En los siguientes apartados se configurará OSPF en cada *router* de **forma incremental** dentro de la misma área (en el área 0) para que las tablas de encaminamiento permitan alcanzar cualquier punto de la red.

1.1. Activación de `r1`

Para observar los mensajes que envíe `r1` cuando se active OSPF, arranca `tcpdump` en `pc1` (`ospf-01.cap`), en `r2(eth0)` (`ospf-02.cap`) y en `r5(eth0)` (`ospf-03.cap`) utilizando la opción `-s 0` para que capture los paquetes completos y guardando la captura en un fichero con la opción `-w`.

A continuación configura OSPF en el encaminador `r1` en el área 0 para que su identificador de *router* sea la mayor de sus direcciones IP y para que exporte las rutas hacia las tres redes a las que está conectado. Ten en cuenta que en su interfaz `eth0` no habrá ningún otro router OSPF conectado y por ello configuraremos esa interfaz como **pasiva**. Para realizar la configuración edita con `mcedit` los ficheros `daemons` y `ospfd.conf` en `r1`, y después arranca *quagga*. Espera un minuto aproximadamente e interrumpe las capturas.

Analiza el comportamiento de `r1` estudiando las capturas con *wireshark* y consultando el estado de OSPF a través de su interfaz VTY y de la orden `route`:

1. Comprueba que en la captura realizada por `pc1` no se observan mensajes OSPF ya que has configurado esa interfaz pasiva.

2. Observa los mensajes `HELLO` que se envían al arrancar *quagga* en `r1` y analízalos utilizando Wireshark.

   a) ¿Cada cuánto tiempo se envían dichos mensajes? Observa si coincide con el valor del campo `Hello Interval` de los mensajes.

   b) Comprueba que el campo `Area ID` se corresponde con el identificador de área que has configurado en el fichero `ospfd.conf`.

   c) Comprueba que el identificador del *router* se corresponde con el que has configurado en el fichero mirando el campo `Source OSPF Router` de la cabecera obligatoria de OSPF en los mensajes `HELLO`.

   Comprueba que este identificador es el mismo para los mensajes enviados por las interfaces `eth1` y `eth2` de `r1`, aunque los mensajes se envíen con dirección IP origen diferente (cada mensaje llevará como dirección IP origen la de la interfaz de red de `r1` por la que se envíe).

   d) Observa el valor de los campos `DR` y `BDR` en los primeros mensajes `HELLO`. ¿Qué ocurre con dichos campos transcurridos 40 segundos después del primer mensaje `HELLO`? ¿Por qué?

3. ¿Se observan en las capturas mensajes `DB Description` o `LS Update`? ¿Por qué?

4. ¿Debería haber aprendido alguna ruta `r1`? Compruébalo consultando la tabla de encaminamiento mediante la orden `route`.

5. Consulta la información de OSPF relativa a la tabla de encaminamiento utilizando la interfaz VTY en `r1` con `show ip ospf route`.

6. Consulta la información de los vecinos que ha conocido `r1` a través de los mensajes `HELLO` recibidos mediante `show ip ospf neighbor`.

7. Consulta la información de la base de datos de *Router Link States* de `r1` con `show ip ospf database router`.

8. Consulta la información de la base de datos de *Network Link States* de `r1` con `show ip ospf database network`

1.2. Activación de `r2`

Para observar los mensajes que envíe `r2` cuando se active OSPF, y los que envíe `r1` a consecuencia de la activación de `r2`, arranca `tcpdump` en `r1(eth1)` (`ospf-04.cap`), en `r3(eth0)` (`ospf-05.cap`) y en `r5(eth0)` (`ospf-06.cap`) utilizando la opción `-s 0` para que capture los paquetes completos y guardando la captura en un fichero con la opción `-w`.

A continuación configura OSPF en el encaminador `r2` en el área 0 para que su identificador de *router* sea la mayor de sus direcciones IP y para que exporte las rutas hacia las dos redes a las que está conectado. Para ello edita los ficheros `daemons` y `ospfd.conf` en `r2`, y después arranca *quagga*.

Espera dos minutos aproximadamente e interrumpe las capturas.

Analiza el comportamiento de `r2` y `r1` estudiando las capturas con *wireshark* y consultando el estado de OSPF a través de las interfaces VTY y de la orden `route` en cada encaminador:

1. Observa la captura realizada en `r1` y responde a las siguientes cuestiones:

   a) Observa que aparecen mensajes `DB_DESCRIPTION` cuando `r1` detecta la presencia de `r2` y viceversa. ¿Cuál es su propósito? ¿Qué IP de destino llevan esos mensajes?

   b) Observa los mensajes `LS Request` que envían `r1` y `r2`. ¿Qué `LSA` pide cada uno al otro? ¿Qué IP de destino llevan estos mensajes?

   c) Observa el primer mensaje `LS Update` que envía `r1`. Comprueba que se corresponde con el `LS Request` enviado por `r2`. Comprueba cómo se corresponde su contenido con lo almacenado en la base de datos de `r1` analizada en el apartado anterior. Observa sus campos para ver si este mensaje incluye la información de que `r1` ha descubierto a `r2` como vecino. ¿Crees que la información contenida en este mensaje deberá cambiar próximamente? ¿Por qué?

   Observa el campo `LS Age` del anuncio que viaja en el mensaje, y explica su valor.

   d) Observa el primer mensaje `LS Update` que envía `r2`. Comprueba que se corresponde con el `LS Request` enviado por `r1`. Observa sus campos para ver si este mensaje incluye la información de que `r2` ha descubierto a `r1` como vecino. ¿Crees que la información contenida en este mensaje deberá cambiar próximamente? ¿Por qué?

   Observa el campo `LS Age` del anuncio que viaja en el mensaje, y explica su valor.

e) Observa el segundo y tercer mensajes `LS Update` que envía `r1`. ¿Responden a algún `LS Request` previo? ¿Por qué se envían? ¿Qué información contienen?

Observa el campo `LS Age` de los anuncios que viajan en los mensajes, y explica su valor.

f) Observa el segundo mensaje `LS Update` que envía `r2`. ¿Responde a algún `LS Request` previo? ¿Por qué se envía? ¿Qué información contiene?

Observa el campo `LS Age` del anuncio que viaja en el mensaje, y explica su valor.

g) ¿Por qué razón `r2` no envía ningún mensaje `Network-LSA`?

h) Observa los mensajes `LS Acknowledge`. Mira su contenido para comprobar a qué `LSA`s asienten.

i) Pasados 40 segundos del arranque de `r2`, ¿qué ocurre con los campos `DR` y `BDR` de los mensajes `HELLO` que intercambian?

2. Observa la captura realizada en `r5` y en `r3`. Explica por qué solo hay mensajes `HELLO`.

3. ¿Deberían haber aprendido alguna ruta `r2` y `r1`? Compruébalo consultando la tabla de encaminamiento en ambos encaminadores mediante la orden `route`.

4. Consulta la información de OSPF relativa a la tabla de encaminamiento utilizando la interfaz VTY en cada encaminador con `show ip ospf route`. Comprueba la métrica de cada ruta y a través de qué *router* se alcanza.

5. Consulta la información de los vecinos que ha conocido cada encaminador a través de los mensajes `HELLO` mediante `show ip ospf neighbor`. Analiza la información que muestra este comando en `r1` donde ya hay elegidos DR y BDR para la subred 11.X.0.0/24.

6. Consulta en cada encaminador la información de las bases de datos de *Router Link States* y de *Network Link States* mediante `show ip ospf database router` y `show ip ospf database network` respectivamente.

   Comprueba que la información mostrada coincide con el contenido de los últimos `LS Update` enviados por los encaminadores.

7. Apunta el número de secuencia de los mensajes `Router-LSA` y `Network-LSA` que ha generado `r1`, los campos `LS Age` y su contenido (recuerda que se encuentran almacenados en la base de datos de `r1` y `r2`). En un apartado posterior se hará referencia a esta información.

8. Consulta un resumen de las bases de datos en cada encaminador con `show ip ospf database`.

1.3. Activación de `r3` y `r4`

Para observar los mensajes que envíen `r3` y `r4` cuando activen OSPF, y los que envíe `r2` a consecuencia de la activación de `r3` y `r4`, arranca `tcpdump` en `r1(eth1)` (`ospf-07.cap`), en `r2(eth1)` (`ospf-08.cap`) y en `r3(eth1)` (`ospf-09.cap`) utilizando la opción `-s 0` para que capture los paquetes completos y guardando la captura en un fichero con la opción `-w`.

Configura OSPF en `r3` y en `r4` (ambos en el área 0), y **arranca quagga a la vez en ambos**. Analiza el comportamiento de los encaminadores estudiando las capturas con *wireshark* y consultando el estado de OSPF a través de las interfaces VTY y de la orden `route` en cada encaminador:

1. Trata de suponer los valores de `DR` y `BDR` en las subredes `12.X.0.0/24` y `13.X.0.0/24`. Comprueba si tus suposiciones son ciertas. Comprueba en los mensajes `HELLO` de la captura en `r3` cómo se ha producido la elección de `DR` y `BDR` al arrancar `r3` y `r4` a la vez.

2. En la captura en `r3` observa el intercambio de mensajes `LS Update` que se produce mientras arrancan `r3` y `r4`.

3. En la captura en `r2` observa el intercambio de mensajes `LS Update` que se produce mientras arrancan `r3` y `r4`.

   Observa también en dicha captura los mensajes `LS Update` que `r3` envía por inundación de los recibidos por él de `r4`. Indica cómo puedes saber si un `LS Update` lo ha originado el encaminador que lo envía o está siendo propagado por inundación (pista: mira el campo `Source OSPF Router` y el campo `Advertising router`).

4. Antes de examinar la captura en `r1` trata de suponer qué tipos de mensaje aparecerán en ella. Comprueba tus suposiciones.

5. Trata de suponer qué modificaciones se habrán realizado en las tablas de encaminamiento de cada *router*. Observa las tablas de encaminamiento utilizando la interfaz VTY con el proceso `ospfd` para verificar tus suposiciones.

6. Consulta la información de los vecinos que ha conocido cada encaminador a través de los mensajes `HELLO` mediante la interfaz VTY. Analiza el resultado del comando `show ip ospf netghbor` donde puedes ver si un vecino es el DR y el BDR de cada una de las subredes a las que está conectado un router.

7. Consulta en cada encaminador la información de las bases de datos de *Router Link States* y de *Network Link States*.

   Comprueba que la información mostrada coincide con el contenido de los últimos `LS Update` enviados por los encaminadores.

8. Por activar `r3` y `r4` la información de los mensajes Network-LSA y Router-LSA que generó `r1` (que se encuentran almacenados en todas las bases de datos) no debería haber cambiado (salvo LS Age). Compruébalo con la información que apuntaste en el apartado 1.2 (7). Fíjate en el campo número de secuencia y responde a estas preguntas:

   - Si es el mismo que tenías apuntado, fíjate en el campo LS Age e indica cuándo crees que cambiará el número de secuencia y por qué. Espera ese tiempo para comprobarlo.
   - Si es diferente, fíjate en el campo LS Age e indica cuándo ha cambiado y por qué.

9. Consulta el resumen de las bases de datos en cada encaminador.

1.4. Reconfiguración de rutas: Activación y desactivación de `r5`

1. Tras haber arrancado OSPF en los encaminadores `r1, r2, r3` y `r4`, `pc1` y `pc2` deberían tener conectividad IP. Compruébalo con las órdenes `ping` y `traceroute`.

   Interrumpe *quagga* en los encaminadores `r1`, `r2`, `r3` y `r4`. Comprueba que ya no funciona un `ping` de `pc1` a `pc2`. Deja lanzado el `ping` de `pc1` a `pc2`, y rearranca *quagga* en `r1`, `r2`, `r3`, `r4`, fijándote en los segundos (aproximadamente) que pasan desde que está arrancado *quagga* en todos los encaminadores hasta que el `ping` empieza a funcionar. Apunta este valor de tiempo.

2. Indica en la tabla de encaminamiento de `r1` que se muestra con la orden `route`. Fíjate en la métrica para la red `14.X.0.0/24`.

3. Realiza los cambios necesarios para que la ruta seguida por los datagramas IP que envía `pc1` a `pc2` vayan por la ruta `pc1 => r1 => r5 => r4 => pc2` , y para que los que envía `pc2` a `pc1` vayan por la ruta `pc2 => r4 => r5 => r1 => pc1`. Para realizar este apartado no podrás añadir o eliminar manualmente rutas en las tablas de encaminamiento. Mirando la tabla de encaminamiento de `r1`, observa y apunta el número de segundos que aproximadamente tarda en aprender `r1` la nueva ruta.

   Comprueba que se está utilizando dicha ruta a través de la orden `traceroute`. Apunta las rutas y sus métricas en las tablas de encaminamiento de cada encaminador.

   Comprueba cómo ha mejorado la métrica para la red 14.X.0.0/24 desde el *router* `r1`. ¿Qué valor tiene ahora?

4. Comprueba la ruta que están siguiendo los mensajes intercambiados entre `pc1` y `pc2` con `traceroute`.

   Deja corriendo en `pc1` un `ping` hacia `pc2`.

5. A continuación interrumpe la ejecución de *quagga* en el encaminador `r5` utilizando la orden `/etc/init.d/quagga stop`. Podrás observar con la orden `route` que ahora `r5` no conoce rutas aprendidas por OSPF. Apunta la tabla de `r5`. Tampoco exporta información de vecinos hacia otros encaminadores.

6. Observarás que el `ping` de `pc1` a `pc2` deja de funcionar durante un buen rato (fíjate en el número de secuencia `icmp_seq`, éste aumenta con cada paquete enviado cada segundo).

   Observa durante este período, en el que no está funcionando `r5`, la tabla de encaminamiento de `r1` y `r4`. Apunta el contenido.

   Observa también durante este periodo la lista de vecinos conocidos por `r1` y por `r4` (utilizando la interfaz VTY con el proceso `ospfd`). Observa la evolución de la columna `Dead Time` de las distintas entradas. ¿Qué entradas no reinician la cuenta desde los 40 segundos? ¿Por qué?

7. Espera hasta que vuelva a funcionar el `ping` (fíjate en el número `icmp_seq`). Observa y apunta el número de segundos que aproximadamente está sin funcionar el `ping` debido a que aún no se ha olvidado la ruta a través de `r5`.

   Comprueba que finalmente `r5` ha desaparecido de entre los vecinos conocidos por `r1` y `r4`.

8. Comprueba ahora las entradas de las tablas de encaminamiento de `r1` y de `r4`.

   Interrumpe el `ping` y comprueba la ruta que están siguiendo los mensajes intercambiados entre `pc1` y `pc2` con `traceroute`.

9. Por último, vuelve a arrancar de nuevo *quagga* en `r5`. Observa cómo cambian las tablas de encaminamiento en `r1` y `r4` y apenas se interrumpe el `ping`.

   Comprueba de nuevo cuál es ahora la ruta que están siguiendo los mensajes intercambiados entre `pc1` y `pc2` con `traceroute`. Observa y apunta el número de segundos que aproximadamente tarda en aprenderse de nuevo la ruta a través de `r5`, mirando continuamente la tabla de encaminamiento de `r1`. Mira también los números de secuencia de los `icmps` del `ping`, y fíjate si alguno se pierde mientras se cambia de la ruta antigua a la ruta nueva.

2. OSPF: red con varias áreas

En el fichero `lab-OSPF-Areas.tgz` está definida una red como la que se muestra en la figura 2. Descomprime el fichero de configuración del escenario `lab-OSPF-Areas.tgz`. Al arrancar NetGUI debes abrir el escenario definido en el directorio `lab-OSPF-Areas`.

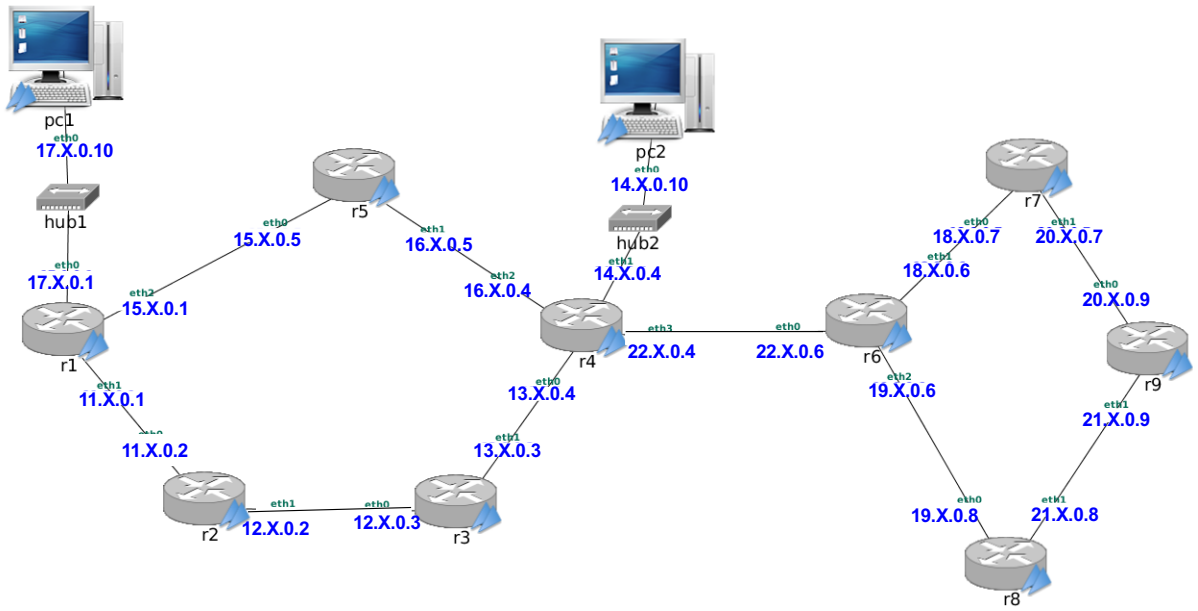

Figura 2: Diagrama de red con todos los *routers* en el área 0

- Arranca todas las máquinas de una en una. Las máquinas `pc1` y `pc2` tienen rutas por defecto a `r1` y `r4` respectivamente. Los *routers* tienen configurado OSPF, **estando todos ellos en el área 0**.
- Arranca *quagga* en todos los *routers*, y espera aproximadamente un minuto.

1. Con la orden `route` comprueba las tablas de encaminamiento en `r1`, `r4`, `r6` y `r9` e incluye su contenido en la memoria. Deberían tener ruta a todas las redes de la figura. Comprueba el coste de cada ruta.
2. Comprueba en esos mismos *routers*, a través de su interfaz VTY, los mensajes LSU *Router-LSA* y *Network-LSA* presentes en sus bases de datos. Toma nota de qué mensajes hay exactamente:
   - Para Router LSA: toma nota del campo Link State ID que representa el router descrito en ese mensaje.
   - Para Network LSA: toma nota del campo Link State ID que representa la subred descrita en ese mensaje.

- Apaga *quagga* en todos los *routers*. Configura ahora todos ellos de forma que se establezcan las áreas que se muestran en la figura 3. Para ello, edita sus ficheros `/etc/quagga/ospfd.conf` y cambia el área al que pertenece cada interfaz de cada router en las líneas `network`.
- Reinicia *quagga* en todos los *routers* **excepto `r4` y `r6`**, y espera aproximadamente un minuto.

3. Mira las bases de datos de `r1` y `r5`. ¿Hay algún mensaje LSU *Summary-LSA* en ellas? ¿Por qué?

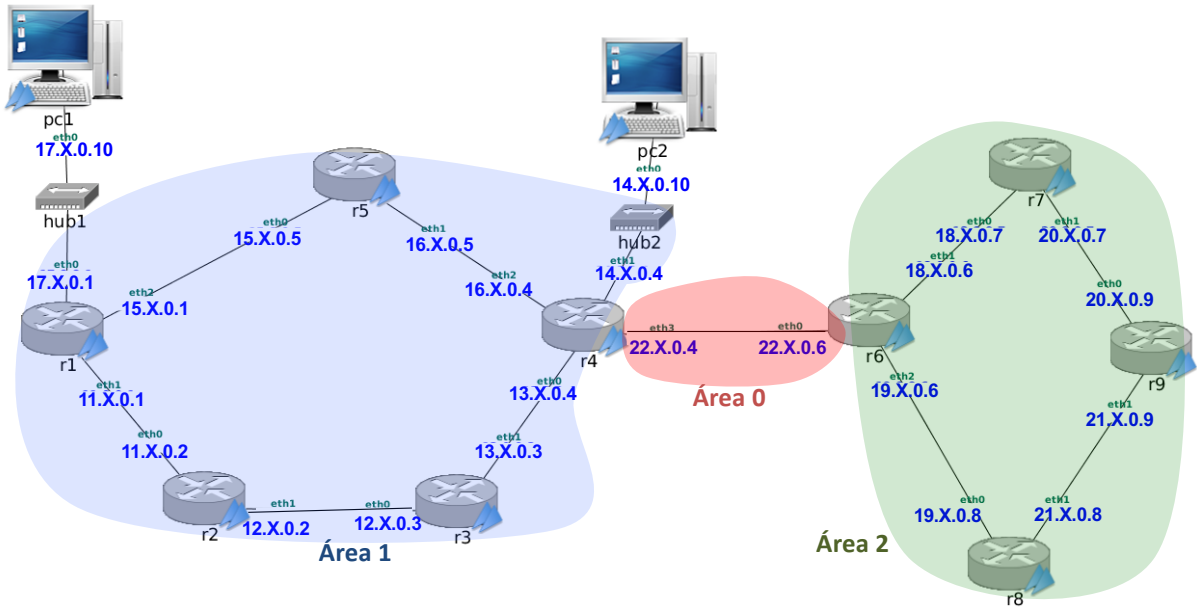

Figura 3: Diagrama de red con varias áreas

- Para observar los mensajes que envíen `r4` y `r6` cuando activen OSPF, arranca `tcpdump` en `r3(eth1)` (`ospfAreas-01.cap`), `r4(eth3)` (`ospfAreas-02.cap`) y `r7(eth0)` (`ospfAreas-03.cap`).
- Arranca ahora *quagga* en `r4` y `r6`, y espera aproximadamente un minuto. Interrumpe las capturas.

4. Localiza en la captura los mensajes `LS Update` que envía `r4` a `r3` que permiten a `r3` añadir una ruta para cada una de las siguientes redes:
   - `18.X.0.0/24`
   - `19.X.0.0/24`
   - `20.X.0.0/24`
   - `21.X.0.0/24`

   Contesta a las siguientes preguntas:

   a) ¿De qué tipo de LSAs se trata?

   b) ¿Qué router es el que está anunciando esos LSAs (`Advertising Router`)? ¿Por qué no es `r6` si las subredes son del área 2?

   c) Para cada uno de esos LSAs, indica cuál es su métrica y por qué.

   d) Busca en la tabla de encaminamiento OSPF de `r3` y relaciona el valor de la métrica del mensaje con el coste que tiene aprendido en la tabla de encaminamiento.

5. Con lo que has aprendido del apartado anterior, trata de suponer cómo serían los mensajes que `r6` le envía a `r7` para informar de las siguientes subredes:
   - `11.X.0.0/24`
   - `12.X.0.0/24`
   - `13.X.0.0/24`
   - `14.X.0.0/24`

- `15.X.0.0/24`
- `16.X.0.0/24`
- `17.X.0.0/24`

a) Para cada uno de los anuncios anteriores supón qué tipo de LSA, qué valor viaja en el campo `Advertising router`, cuál es el valor de métrica anunciado. Localiza en la captura los mensajes `LS Update` que envía `r6` a `r7` para confirmar tu suposición.

b) Supón qué habrá añadido `r7` en su tabla de encaminamiento OSPF y comprueba tus suposiciones consultando la tabla en `r7`.

6. Localiza en las tres capturas qué tipo de LSA contiene el anuncio de la existencia de la red `22.X.0.0/24` e indica su tipo y el contenido de los campos `Advertising Router` y `metric`:
   - cuando `r3` la aprende de `r4`
   - cuando `r6` la aprende de `r4`
   - cuando `r7` la aprende de `r6`
7. Localiza en las tres capturas qué tipo de LSA contiene el anuncio de la existencia de la red `14.X.0.0/24` e indica su tipo y el contenido de los campos `Advertising Router` y `metric`:
   - cuando `r3` la aprende de `r4`
   - cuando `r6` la aprende de `r4`
   - cuando `r7` la aprende de `r6`
8. Comprueba las tablas de encaminamiento en `r1`, `r4`, `r6` y `r9`. Indica el coste de cada ruta. Compara los resultados con los obtenidos en la pregunta 1.
9. Indica en esos mismos *routers*, a través de su interfaz VTY, los mensajes LSU *Router-LSA*, los *Network-LSA* y los *Summary-LSA* presentes en sus bases de datos. Compara con los resultados obtenidos en la pregunta 2.
10. Explica gracias a qué mensaje/s `r1` ha conocido las siguientes subredes y qué router ha generado dicho/s mensaje/s:
    - 12.X.0.0/24
    - 22.X.0.0/24
    - 21.X.0.0/24

3. Entrega de la práctica

Sube al enlace que encontrarás en `aulavirtual` antes de que termine el plazo de entrega, dos ficheros:

- Memoria de la práctica en formato PDF
- Un fichero de nombre `p2.zip` o `p2.tgz` que incluya todas las capturas:
  - De `ospf-01.cap` a `ospf-09.cap`.
  - De `ospfAreas-01.cap` a `ospfAreas-03.cap`.

Puedes crear el fichero de esta forma: primero crea una carpeta `p2` 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 `'p2'` y extensión `'.zip'`.

Sistemas Telemáticos para Medios Audiovisuales
Práctica 3: Protocolos de Encaminamiento: BGP

GSyC
Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación

Septiembre de 2022

Para esta práctica, cada alumno tendrá escenarios diferentes en cada apartado. En particular, las direcciones IP de las máquinas tendrán asignado en el segundo byte un valor X distinto. Podrás ver qué valor X tienes asignado cuando cargues el escenario en NetGUI y observes la configuración.

Antes de comenzar a realizar la práctica, por favor, descarga tus escenarios del siguiente enlace donde deberás introducir tu número de DNI (8 dígitos) con la letra correspondiente:

`http://mobiquo.gsyc.urjc.es/practicas/stma/p3.html`

1. Escenario de red utilizando OSPF y BGP

En el fichero `lab-BGP.tgz` se encuentran los ficheros de configuración para crear un escenario de red como el que se muestra en la figura 1. En esta figura hay tres AS (Sistemas Autónomos): AS10, AS20 y AS30. Se ha configurado OSPF dentro de AS20 y AS30 y se desea configurar BGP para conectar los 3 sistemas autónomos. En AS10 no se utilizará ningún protocolo de encaminamiento interior ya que sólo hay una red interna.

Se realizará la configuración por pasos. Supondremos para la configuración de BGP que AS10 y AS20 mantienen una relación de tránsito y que AS10 y AS30 mantienen una relación de tránsito, siendo en ambos casos AS10 el proveedor.

1.1. Configuración de los pcs y *routers* de AS20

1. Descomprime el fichero de configuración del escenario `lab-BGP.tgz`. Al arrancar NetGUI debes abrir el escenario definido en el directorio `lab-BGP`.

2. Arranca de una en una sólo las máquinas de AS20.

3. Las máquinas tienen configurada una dirección IP en cada una de sus interfaces de red. Los pcs además tienen configurada una ruta por defecto al único *router* al que están directamente conectados.

4. Los *routers* de AS20 (`as20-r1`, `as20-r2` y `as20-r3`) tienen configurado el protocolo OSPF para que intercambien información de encaminamiento dentro de AS20 (consulta los ficheros `daemons` y `ospfd.conf` de estos *routers*). Arranca `quagga` en todos los *routers* de AS20.

5. Consulta las tablas de encaminamiento utilizando la interfaz VTY con los procesos `ospfd` de cada *router* y mediante el comando `route`.

6. Comprueba utilizando `ping` que todos los pcs y *routers* tienen conectividad dentro de AS20.

7. Modifica la configuración de `quagga` en `as20-r1` para que utilice además de OSPF el protocolo BGP. Define como vecino suyo a `as10-r1`. Utiliza la redistribución de las subredes directamente conectadas (`redistribute connected`). No uses aún la redistribución de rutas entre OSPF y BGP. Inicia una captura en la interfaz `as20-r1(eth0)` y guarda su contenido en un fichero (`bgp-01.cap`). Reinicia `quagga` en `as20-r1`.

8. ¿Debería haber aprendido alguna ruta `as20-r1` por BGP? Compruébalo consultando la tabla de encaminamiento mediante el comando `route` y conectándote a la interfaz VTY del proceso `bgpd` para ver la tabla de encaminamiento BGP.

9. ¿Deberían haber aprendido alguna ruta `as20-r2` y `as20-r3`?

10. Después de al menos 3 minutos, interrumpe la captura e indica qué mensajes observas. Justifica tu respuesta.

1.2. Configuración del pc y *router* de AS10

1. Arranca `as10-pc1` y `as10-r1`. La máquina `as10-pc1` tiene configurada una dirección IP y una ruta por defecto al *router* al que está directamente conectado. El *router* `as10-r1` tiene configuradas direcciones IP, una en cada interfaz, pero no tiene configurada ninguna ruta adicional a las de las subredes a las que está directamente conectado.

2. Configura `quagga` en `as10-r1` para que utilice el protocolo BGP. Define como vecinos suyos a `as20-r1` y a `as30-r1`. Incluye la configuración en la memoria.

3. Captura el tráfico con `tcpdump` en `as20-r1(eth0)` y guarda la captura en un fichero (`bgp-02.cap`). Arranca `quagga` en `as10-r1`. Espera un minuto e interrumpe la captura.

4. Analiza la captura `bgp-02.cap` realizada:

   - Observa que el tráfico de BGP va dentro de una conexión TCP.
   - Localiza los mensajes `OPEN` que intercambian los *routers* vecinos. Observa en ambos mensajes `OPEN` los siguientes campos:
     - `My AS`
     - Identificador del *router* BGP
     - `Hold time`
     - En los parámetros opcionales, el campo `Capability` que contiene la información del número de sistema autónomo usando 4 bytes (32 bits).
   - Localiza los mensajes `KEEPALIVE` que intercambian los *routers*. Además de la cabecera obligatoria de BGP (`Marker`, `Length` y `Type`) ¿qué otra información viaja en este tipo de mensajes?
   - Localiza los mensajes `UPDATE` que intercambian los *routers*. Observa en ambos mensajes `UPDATE` los siguientes campos:
     - Rutas eliminadas
     - Rutas anunciadas
     - Atributos, en particular el valor de `NEXT_HOP` y `AS_PATH`.

- El atributo `ORIGIN` tiene como valor `INCOMPLETE` porque esas subredes se anuncian debido a la redistribución de subredes y no a través de líneas `network`. Elimina en el fichero `bgpd.conf` de `as10-r1` la línea:

  ```
  redistribute connected
  ```

  y añade las siguientes líneas (modificando el valor de X):

  ```
  network 11.X.1.0/24
  network 20.X.1.0/24
  network 20.X.2.0/24
  ```

  Interrumpe `quagga` en `as10-r1`, inicia una captura de tráfico en `as20-r1(eth0)` dirigiendo su contenido a un fichero (`bgp-03.cap`) y arranca `quagga` en `as10-r1` de nuevo. Pasado 2 minutos aproximadamente interrumpe la captura. Indica el contenido del atributo `ORIGIN` en el mensaje UPDATE que genera `as10-r1` para dichas subredes. Puedes observar como el valor de este atributo (i=IGP, e=EGP, ?=incomplete) también se observa en la tabla BGP, en la columna `Path`, junto al ASN que originó el anuncio. Copia el contenido de esta tabla en la memoria.

5. Consulta la tabla de encaminamiento utilizando la interfaz VTY con el proceso `bgpd` y con el comando `route` en los *routers* `as10-r1` y `as20-r1`. Incluye sus contenidos en la memoria y explica las diferencias.

6. Prueba a hacer un `ping` desde `as10-pc1` hacia `as20-pc2` y comprueba que no funciona. ¿Por qué? (Explica la tabla de encaminamiento que tiene `as10-r1`).

7. Modifica la configuración del fichero `bgpd.conf` de `as20-r1` para que se redistribuyan las rutas aprendidas por OSPF de AS20 a otros ASs utilizando BGP. Incluye la modificación en la memoria. Inicia una captura en `as20-r1(eth0)` y guarda su contenido en un fichero (`bgp-04.cap`). Reinicia `quagga` en `as20-r1`. Comprueba que ahora `as10-r1` tiene rutas a todas las redes de AS20. En la tabla BGP de `as10-r1` fíjate en el atributo `ORIGIN` para las subredes de AS20.

8. Interrumpe la captura y explica los anuncios de las redes internas de AS20 que ves en el tráfico capturado.

9. Prueba a hacer un `ping` desde `as10-pc1` hacia `as20-pc2` y comprueba que todavía no funciona. ¿Por qué? (Explica la tabla de encaminamiento que tiene `as20-r2`).

10. Modifica la configuración del fichero `ospfd.conf` de `as20-r1` para que se redistribuyan las rutas aprendidas por BGP a los *routers* de AS20 mediante OSPF. Incluye la modificación en la memoria. Reinicia `quagga` en `as20-r1`. Comprueba que `as20-r2` tiene ruta a la red de AS10.

11. Comprueba que ahora sí funciona el `ping` entre `as10-pc1` y `as20-pc2`.

1.3. Configuración de los pcs y *routers* de AS30

Arranca todas las máquinas de AS30 de una en una. Los *routers* tienen configurada una dirección IP por cada una de sus interfaces. Los pcs tienen configurada una dirección IP y una ruta por defecto al *router* al que están directamente conectados.

Los *routers* de AS30 (`as30-r1`, `as30-r2` y `as30-r3`) tienen configurado el protocolo OSPF para que intercambien información de encaminamiento dentro de AS30.

1. Consulta los ficheros `daemons` y `ospfd.conf` de los *routers* de AS30.

2. Arranca `quagga` en todos los *routers* de AS30. Consulta las tablas de encaminamiento utilizando la interfaz VTY con los procesos `ospfd` de cada *router* y mediante el comando `route`.

3. Comprueba utilizando `ping` que todos los pcs y *routers* tienen conectividad dentro de AS30.

4. Modifica la configuración de `quagga` en `as30-r1` para que utilice además el protocolo BGP. Define como vecino suyo a `as10-r1`. No uses aún la redistribución de rutas entre OSPF y BGP. Incluye la configuración en la memoria.

5. Captura el tráfico con `tcpdump` en `as10-r1(eth2)`, con la opción `-s` para capturar paquetes enteros y `-w` para guardar la captura en un fichero (`bgp-05.cap`). Reinicia `quagga` en `as30-r1`. Espera un minuto e interrumpe la captura.

6. Analiza la captura realizada:

   a) Observa que el tráfico de BGP va dentro de una conexión TCP.

   b) Localiza los mensajes `OPEN` que intercambian los *routers* vecinos. Observa en ambos mensajes `OPEN` los siguientes campos:

   - `My AS`
   - Identificador del *router* BGP
   - `Hold time`
   - En los parámetros opcionales, el campo `Capability` que contiene la información del número de sistema autónomo usando 4 bytes (32 bits).

   c) Localiza los mensajes `KEEPALIVE` que intercambian los *routers*. Comprueba que son similares a los que ya observaste en el apartado anterior

   d) Trata de suponer qué rutas le anunciará `as10-r1` a `as30-r1` en sus mensajes `UPDATE`. ¿Qué `AS_PATH` crees que traerán esas rutas? Como los atributos de un mensaje `UPDATE` son comunes a todas las rutas anunciadas, ¿podrá anunciar `as10-r1` todas las subredes que conoce en un solo mensaje `UPDATE`?

   e) Trata de suponer qué rutas le anunciará `as30-r1` a `as10-r1` en sus mensajes `UPDATE`.

   f) Localiza en la captura los mensajes `UPDATE` que intercambian los *routers* y confirma si tus suposiciones son ciertas. Observa el valor de los atributos `NEXT_HOP` y `AS_PATH`.

7. ¿Debería haber aprendido alguna ruta `as30-r1`? Compruébalo consultando la tabla de encaminamiento mediante el mandato `route`.

8. El resto de *routers* de AS30 ¿deberían haber aprendido alguna otra ruta? Compruébalo.

9. Prueba a hacer un `ping` desde `as10-pc1` hacia `as30-pc3` y comprueba que no funciona. ¿Por qué? (Explica la tabla de encaminamiento de `as10-r1`).

10. Modifica la configuración del fichero `bgpd.conf` de `as30-r1` para que se redistribuyan las rutas aprendidas por OSPF de AS30 a otros ASs utilizando BGP. Incluye la modificación en la memoria. Reinicia `quagga` en `as30-r1`. Comprueba que ahora `as10-r1` tiene rutas a todas las redes de AS30. En la tabla BGP de `as10-r1` fíjate en el atributo `ORIGIN` para las subredes de AS30.

11. Prueba a hacer un `ping` desde `as10-pc1` hacia `as30-pc3` y comprueba que todavía no funciona. ¿Por qué? (Explica la tabla de encaminamiento de `as30-r3`).

12. Modifica la configuración del fichero `ospfd.conf` de `as30-r1` para que se redistribuyan las rutas aprendidas por BGP a los *routers* de AS30 mediante OSPF. Incluye la modificación en la memoria. Reinicia `quagga` en `as30-r1`. Comprueba que ahora `as30-r3` tiene ruta a la red de AS10.

13. Comprueba que ahora sí funciona el `ping` entre `as10-pc1` y `as30-pc3` y realiza una captura de tráfico en `as10-pc1` guardando su contenido en el fichero `bgp-06.cap`.

14. Comprueba que hay conectividad entre todos los pcs de la figura.

2. Agregación de rutas

La configuración de BGP realizada en el apartado anterior provoca que las subredes del sistema autónomo AS20 se almacenen de forma independiente, ocupando cada una de ellas una entrada diferente en las tablas de encaminamiento de los *routers* de AS10 y AS30. De forma equivalente, cada una de las subredes de AS30 ocupan entradas diferentes en las tablas de encaminamiento de los *routers* de AS10 y AS20.

Utilizando CIDR pueden agruparse estas entradas para que los anuncios por BGP que emiten AS20 y AS30 optimicen el número de entradas en las tablas de encaminamiento en los *routers* externos a dichos sistemas autónomos.

1. Interrumpe la ejecución de `quagga` en `as20-r1` y `as30-r1`. Configura BGP en AS20 para que se optimice el número de entradas en las tablas de encaminamiento de los *routers* de AS10 y AS30. Ten en cuenta que al realizar la agregación de rutas, dicha agregación sólo puede referirse a subredes que pertenezcan a AS20. Incluye la modificación en la memoria.

2. Configura BGP en AS30 para que se optimice el número de entradas en las tablas de encaminamiento de los *routers* de AS10 y AS20. Ten en cuenta que al realizar la agregación de rutas, dicha agregación sólo puede referirse a subredes que pertenezcan a AS30. Incluye la modificación en la memoria.

3. Captura el tráfico con `tcpdump` en `as10-r1(eth2)`, con la opción `-s` para capturar paquetes enteros y `-w` para guardar la captura en un fichero `bgp-07.cap`.

4. Inicia `quagga` en `as20-r1` y `as30-r1`, espera a que todos los routers se hayan intercambiado la información de encaminamiento e interrumpe la captura. Analiza la captura realizada:

   - Trata de suponer cómo serán los nuevos mensajes `UPDATE` que intercambien los *routers* anunciando las redes de AS20 y AS30. Localízalos en la captura y confirma si tus suposiciones son ciertas.
   - Fíjate en el atributo `ORIGIN` para estas subredes en los mensajes `UPDATE` y en la tabla BGP de `as10-r1`, su valor es diferente después de realizar la agregación.

5. Consulta las tablas de encaminamiento de los *routers* de AS20 y AS30 mediante el comando `route`, para ver cómo se han agregado las rutas hacia el sistema autónomo externo. Explica el resultado.

6. Consulta la tabla BGP en `as20-r1`, observarás como las subredes de AS20 con el prefijo agregado aparecen como ruta preferida y serán las que se anuncian a otros vecinos BGP. Además, las subredes que se anunciaban previamente de forma independiente y ahora se anuncian dentro del prefijo agregado también aparecen pero marcadas con una `s` que indica que se suprimen.

Consulta esta misma información en la tabla BGP de `as30-r1` para las subredes de AS30 que agrega este router. Explica los resultados.

3. Modificación del escenario: Políticas de exportación de rutas

AS20 y AS30 se dan cuenta de que intercambian mucho tráfico entre ellos y deciden conectar directamente `as20-r1` y `as30-r1` definiendo una relación entre iguales entre los mismos.

- Interrumpe *quagga* en `as20-r1` y `as30-r1`.
- Interrumpe la ejecución de los *routers* `as20-r1` y `as30-r1`, apagando cada uno de ellos desde la interfaz gráfica de NetGUI. Dibuja un enlace directo entre ambos *routers* y arráncalos de nuevo, uno después de otro.
- Asigna la dirección 20.X.3.20 a la nueva interfaz de `as20-r1` y la dirección 20.X.3.30 a la nueva interfaz de `as30-r1`, ambas direcciones de la red 20.X.3.0/24.
- No apliques por ahora ninguna política de exportación de rutas. Modifica la configuración de BGP de `as20-r1` y `as30-r1` para añadir en cada uno al otro como nuevo vecino. Por defecto si no se configuran políticas de exportación, se anuncian todas las rutas seleccionadas como preferidas.
- Arranca `quagga` en `as20-r1` y `as30-r1`.

Incluye en la memoria las respuestas a los siguientes apartados.

1. Realiza una captura de la interfaz gráfica de NetGUI donde se vea la nueva conexión y las direcciones IP asignadas. Incluyen esa imagen en la memoria.
2. Comprueba mediante `route` en `as20-r1` la ruta hacia las redes de AS30, y en `as30-r1` la ruta hacia las redes de AS20. Utilizando la interfaz VTY en ambos *routers* observa cómo cada uno tiene dos rutas alternativas para el sistema autónomo vecino, y ha elegido una de ellas. ¿Cuál? ¿Por qué? Ten en cuenta que `LOCAL_PREF` no se ha modificado y por tanto valdrá para todas las interfaces su valor por defecto, 100.
3. Observa la tabla BGP de `as20-r1` e indica cuántas rutas alternativas existen para las subredes 20.X.1.0/24, 20.X.2.0/24 y 20.X.3.0/24. Indica cómo `as20-r1` ha aprendido estas rutas alternativas y cuál se ha seleccionado como preferida.
4. ¿Qué ruta crees que seguirán los paquetes intercambiados entre `as20-pc3` y `as30-pc2`? Compruébalo.
5. ¿Qué ruta crees que seguirán los paquetes enviados desde `as30-pc3` con destino `as10-pc1`? Compruébalo utilizando `traceroute`. Utilizando la interfaz VTY en `as30-r1` comprueba cómo tiene dos rutas alternativas para la red `11.X.1.0/24`. Observa cuál es la elegida y por qué.
6. Apaga la interfaz `eth0` de `as30-r1` con `ifconfig eth0 down`. Espera unos 3 minutos. ¿Qué habrá pasado ahora con la ruta que seguirán los paquetes enviados desde `as30-pc3` con destino `as10-pc1`? Compruébalo utilizando `traceroute`. Utilizando la interfaz VTY en `as30-r1` comprueba que ahora sólo tiene una ruta para la red `11.X.1.0/24`. Dada las relaciones entre AS10, AS20 y AS30 indica si esta situación perjudica a alguno de los AS y por qué.

7. Teniendo en cuenta las relaciones entre AS10, AS20 y AS30:

   a) ¿Qué rutas debería exportar AS20 a AS30, y qué rutas no debería exportarle?

   b) ¿Qué rutas debería exportar AS30 a AS20, y qué rutas no debería exportarle?

8. Teniendo en cuenta las rutas que deben exportarse y las que no, vuelve a configurar BGP en `as20-r1` y `as30-r1` para que se anuncien y se exporten sólo las rutas que a cada AS le interesa. Incluye las modificaciones en la memoria.

9. Vuelve a levantar la interfaz `eth0` de `as30-r1` con `ifconfig eth0 up`. Inicia una captura de tráfico en la interfaz que une `as20-r1` y `as30-r1` y guarda el contenido en el fichero `bgp-08.cap`. Reinicia `quagga` en los 3 *routers* BGP: `as10-r1`, `as20-r1` y `as30-r1`.

10. Comprueba ahora las tablas de encaminamiento en `as20-r1` y `as30-r1`, tanto con `route` como con la interfaz VTY. Explica el contenido.

11. Interrumpe la captura y explica el contenido de los mensajes UPDATE que intercambian `as20-r1` y `as30-r1`. ¿Cuáles crees que son las diferencias de los mensajes UPDATE intercambiados en el apartado 3 y ahora?

12. ¿Qué ruta crees que seguirán ahora los paquetes intercambiados entre `as20-pc3` y `as30-pc2`? Compruébalo.

13. ¿Qué ruta crees que seguirán ahora los paquetes enviados desde `as30-pc3` con destino `as10-pc1`? Compruébalo utilizando `traceroute`. Utilizando la interfaz VTY en `as30-r1` comprueba qué rutas tiene disponibles hacia la red `11.X.1.0/24`.

14. Apaga la interfaz `eth0` de `as30-r1` con `ifconfig eth0 down`. ¿Qué habrá pasado ahora con la ruta que seguirán los paquetes enviados desde `as30-pc3` con destino `as10-pc1`? Compruébalo utilizando `traceroute`. Utilizando la interfaz VTY en `as30-r1` comprueba qué rutas hay ahora para la red `11.X.1.0/24`.

4. Políticas de exportación y orden de preferencia en la selección de rutas

En el fichero `lab-BGP2.tgz` se encuentran los ficheros de configuración para crear un escenario de red como el que se muestra en la figura 2. En esta figura hay 6 AS (Sistemas Autónomos): AS10, AS20, AS30, AS40, AS50 y AS60. Se ha configurado OSPF dentro de AS20, OSPF dentro de AS30 y BGP en todos ellos para intercambiar la información de encaminamiento. Se desea que:

- AS30 y AS10 tengan una relación de tránsito, donde AS30 sea el proveedor y AS10 el cliente.
- AS30 y AS40 tengan una relación de tránsito, donde AS30 sea el proveedor y AS40 el cliente.
- AS40 y AS50 tengan una relación de tránsito, donde AS40 sea el proveedor y AS50 el cliente.
- AS40 y AS60 tengan una relación de tránsito, donde AS40 sea el proveedor y AS60 el cliente.
- AS10 y AS20 tengan una relación de tránsito, donde AS10 sea el proveedor y AS20 el cliente.
- AS20 y AS60 tengan una relación de tránsito, donde AS20 sea el proveedor y AS60 el cliente.

- AS10 y AS40 tengan una relación entre iguales.
- AS20 y AS50 tengan una relación entre iguales.

Arranca todas las máquinas de una en una. Por defecto, al arrancar las máquinas se arranca `quagga`. En el escenario se ha configurado OSPF y BGP. Sin embargo, no se han configurado las políticas de exportación ni el atributo LOCAL PREF de BGP.

1. Piensa en qué *routers* debería existir una lista de exportación de rutas e indica qué rutas deberían estar en dicha lista y a qué *router/s* se le exportaría. Interrumpe `quagga` en los *routers* en los que necesites cambiar la configuración y realiza dicha configuración. Inicia nuevamente `quagga` en dichos *routers*.
2. Comprueba en tu nueva configuración las siguientes reglas consultando cada una de las tablas BGP de los routers de la figura:
   - Un AS no anuncia a su proveedor las subredes aprendidas de otro AS proveedor o de un AS con relación entre iguales.
   - Un AS no anuncia a un AS con relación entre iguales las subredes aprendidas de un AS proveedor o de otro AS con relación entre iguales.
3. Fíjate en la tabla BGP de `as50-r1`. ¿Cuántas rutas hay en la tabla BGP para alcanzar AS60?
4. Interrumpe `quagga` en `as20-r1`. Fíjate en la tabla BGP de `as50-r1`. ¿Cuántas rutas hay ahora en la tabla BGP para alcanzar AS60?
5. ¿Cuál es la ruta que aparece en la tabla de encaminamiento de `as50-r1` para alcanzar AS60?
6. Inicia `quagga` en `as20-r1`. Espera 2 minutos aproximadamente para que `as50-r1` y `as20-r1` hayan intercambiado la información de encaminamiento BGP. ¿Cuántas rutas hay en la tabla BGP para alcanzar AS60?
7. ¿Cuál es la ruta que aparece en la tabla de encaminamiento de `as50-r1` para alcanzar AS60?
8. ¿Es consistente esta ruta con las relaciones entre ASs definidas previamente?
9. Piensa en los atributos LOCAL PREF que configurarías en `as50-r1` y realiza dicha configuración en el escenario. Incluye la configuración en la memoria. Reinicia `quagga` en `as50-r1`.
10. Interrumpe `quagga` nuevamente en `as20-r1`. Fíjate en la tabla BGP de `as50-r1`. ¿Cuántas rutas hay en la tabla BGP para alcanzar AS60?
11. Inicia `quagga` en `as20-r1`. Espera 2 minutos aproximadamente para que `as50-r1` y `as20-r1` hayan intercambiado la información de encaminamiento BGP. ¿Cuántas rutas hay en la tabla BGP para alcanzar AS60?
12. ¿Cuál es la ruta que aparece en la tabla de encaminamiento de `as50-r1` para alcanzar AS60? Ahora debería ser consistente con las relaciones entre ASs definidas previamente.
13. Modifica la configuración de `as10-r1` para definir el parámetro `LOCAL_PREF` acorde a las relaciones que tiene con sus ASs vecinos. Incluye las modificaciones en la memoria.
14. Comprueba que después de realizar la configuración, `as10-r1` tiene como ruta preferida para alcanzar las subredes internas de AS60 a través de `as20-r1` (sin la configuración de `LOCAL_PREF`, en este caso la selección de ruta dependería del orden de arranque de los routers).

5. Rutas eliminadas

Con todas las máquinas arrancadas, después de realizar la configuración del apartado anterior, piensa qué anuncios BGP con las subredes internas de AS60 se enviarían si apagaras `as60-r1` y responde a las siguientes preguntas:

1. ¿A qué otros routers enviaría `as20-r1` un mensaje BGP con rutas eliminadas? ¿Por qué?
2. ¿A qué otros routers enviaría `as40-r1` un mensaje BGP con rutas eliminadas? ¿Por qué?
3. ¿A qué otros routers enviaría `as10-r1` un mensaje BGP con rutas eliminadas? ¿Por qué?
4. ¿A qué otros routers enviaría `as50-r1` un mensaje BGP con rutas eliminadas? ¿Por qué?
5. Inicia las siguientes capturas:
   - En `as20-r1(eth2)` guardando el contenido en `bgp-09.cap`.
   - En `as20-r1(eth3)` guardando el contenido en `bgp-10.cap`.
   - En `as40-r1(eth0)` guardando el contenido en `bgp-11.cap`.
   - En `as40-r1(eth1)` guardando el contenido en `bgp-12.cap`.
   - En `as40-r1(eth2)` guardando el contenido en `bgp-13.cap`.
   - En `as30-r1(eth3)` guardando el contenido en `bgp-14.cap`.

   Interrumpe la ejecución de `quagga` en `as60-r1`. Espera a que los routers intercambien toda la información de encaminamiento (al menos 2 minutos). Explica los mensajes UPDATE que encuentras en las capturas anteriores y que contengan información de las subredes de AS60 (16.X.0.0/24), indica qué router lo envía y por qué.

6. Ruta por defecto

1. Cambia la configuración de `as40-r1` de forma que AS40 anuncie a AS50 una ruta por defecto. No elimines aún los anuncios de las subredes individuales.
2. Reinicia `quagga` en `as40-r1` y observa la tabla de encaminamiento y la tabla BGP de `as50-r1`. Comprueba que en ambas tablas hay una ruta por defecto, pero siguen estando las rutas individuales. Las rutas a las subredes individuales, al ser más específicas serán las utilizadas, sin llegar a usarse nunca la ruta por defecto.
3. Cambia la configuración en `as40-r1` para evitar que siga anunciando a AS50 las subredes individuales.
4. Reinicia `quagga` en `as40-r1` y comprueba que la tabla de encaminamiento y la tabla BGP de `as50-r1` ahora sólo tienen la ruta por defecto.

7. Entrega de la práctica

Sube al enlace que encontrarás en `aulavirtual` antes de que termine el plazo de entrega, dos ficheros:

- Memoria de la práctica en formato PDF
- Un fichero de nombre `p3.zip` o `p3.tgz` que incluya todas las capturas: Ficheros de captura de `bgp-01.cap` a `bgp-14.cap`

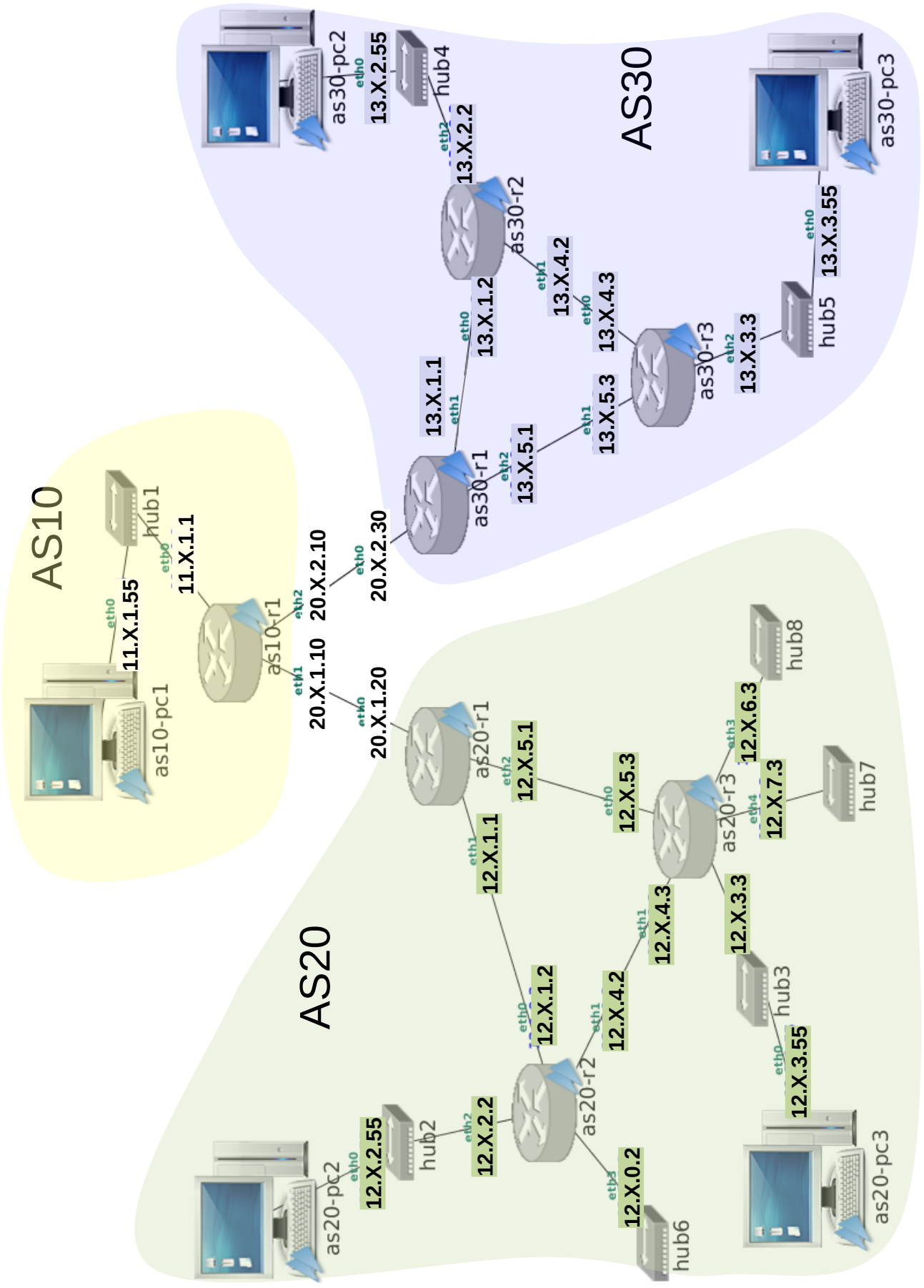

Figura 1: Escenario de red para los ejercicios de BGP

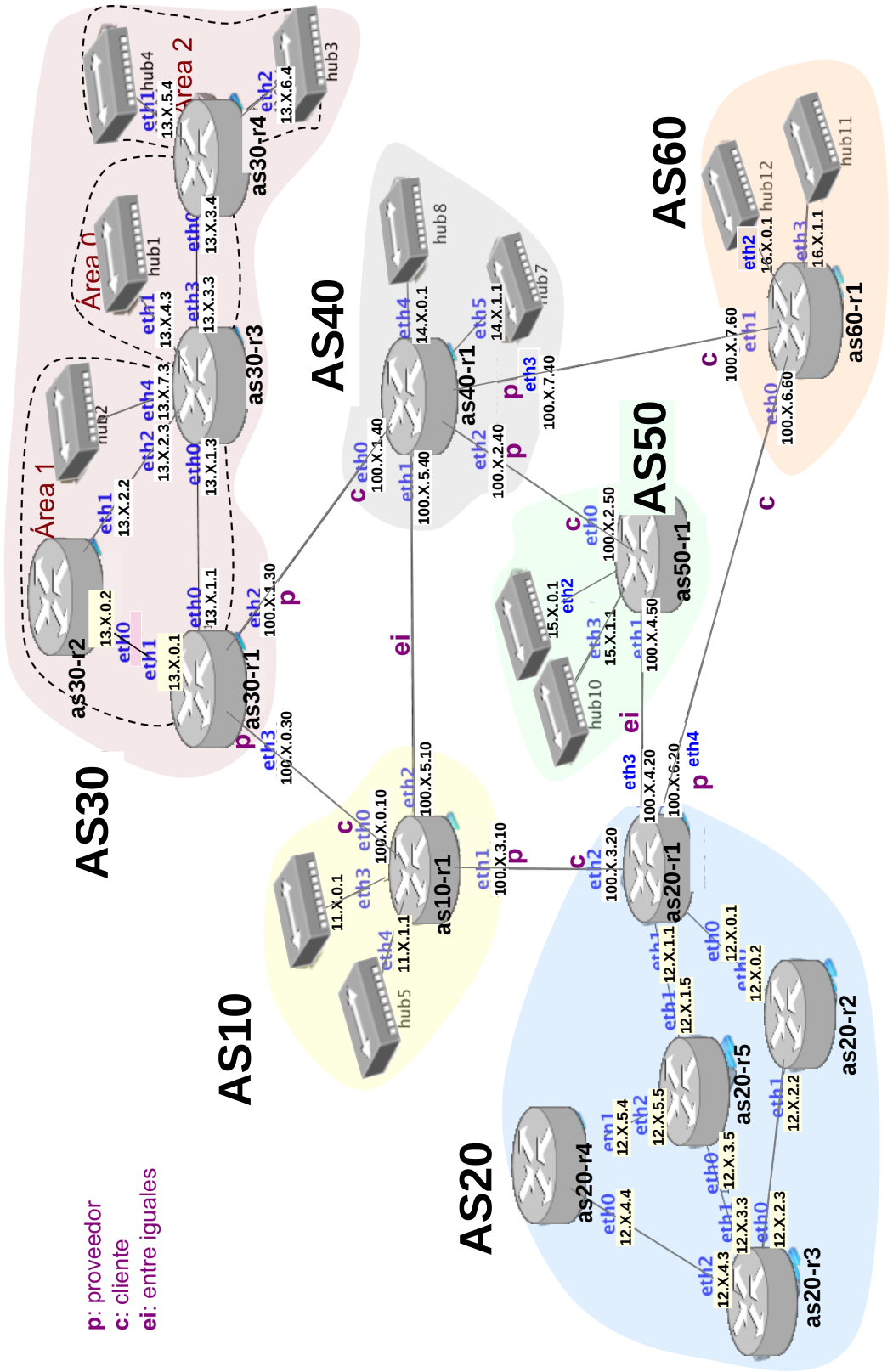

Figura 2: Escenario para listas de exportación y local pref en BGP

Sistemas Telemáticos para Medios Audiovisuales
Práctica 4
Calidad de Servicio: Control de tráfico en Linux

GSyC
Departamento de Teoría de la Señal y Comunicaciones
y Sistemas Telemáticos y Computación
URJC

Septiembre de 2022

Introducción

Descarga tu escenario de red para esta práctica del siguiente enlace:

`http://mobiquo.gsyc.urjc.es/practicas/stma/p4.html`

Descomprime el fichero que contiene el escenario de NetGUI `lab-tc.tgz`.

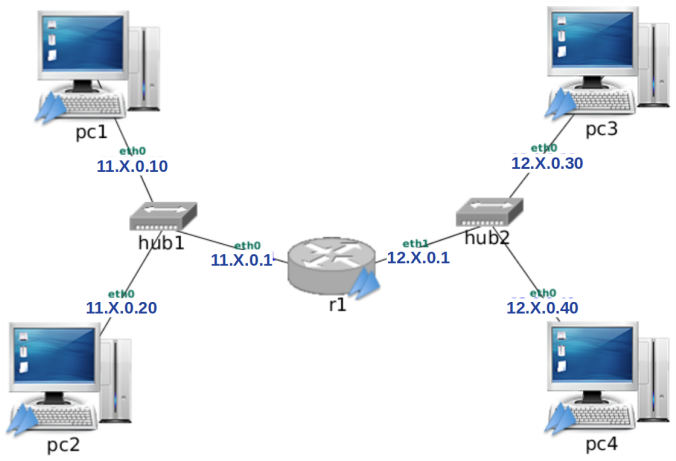

Figura 1: Escenario para control de tráfico

1. Control de Tráfico

1.1. Sin control de tráfico ni a la entrada ni a la salida

El *router* `r1` no tiene activado el control de tráfico en ninguna de sus interfaces.

1.1.1. Un flujo de datos

Inicia una captura en la interfaz `r1(eth1)` guardando el contenido en `tc-01.cap`.

Arranca `iperf` en modo servidor UDP en `pc3` y arranca `iperf` en modo cliente UDP en `pc1` para que envíe tráfico a 3M durante 10 segundos a `pc3`.

Observa en el servidor el informe que aparece al terminar de recibir el tráfico `pc1` → `pc3`.
Carga la captura en `wireshark` y muestra el flujo de forma gráfica, incluye una imagen en la memoria.

1.1.2. Dos flujos de datos

- Arranca `iperf` en modo servidor UDP en `pc4`.
- Arranca otro `iperf` en modo servidor UDP en `pc3`.
- Inicia una captura de tráfico en la interfaz `eth1` de `r1` (`tc-02.cap`).
- Escribe (todavía sin ejecutar) el comando que arranca `iperf` en modo cliente UDP en `pc1` para que envíe 3M al servidor `pc3` en el sentido `pc1` → `pc3` durante 10 segundos.
- Escribe (todavía sin ejecutar) el comando que arranca `iperf` en modo cliente UDP en `pc2` para que envíe 3M al servidor `pc4` en el sentido `pc2` → `pc4` durante 10 segundos.
- Ejecuta los dos comandos anteriores uno a continuación de otro (lo más rápidamente que puedas) para que su ejecución se realice de forma simultánea.
- Interrumpe la captura una vez que los clientes hayan terminado de ejecutar `iperf`.

A continuación analiza los resultados obtenidos:

1. Explica las estadísticas que muestran los servidores.
2. Carga la captura en `wireshark` y muestra cada uno de los flujos de forma gráfica. Incluye una imagen en la memoria que muestre los flujos de forma gráfica. Explica el ancho de banda medido para cada uno de los flujos.

1.2. Control de admisión para el tráfico de entrada

Vamos a configurar `r1` para restringir el tráfico de entrada distinguiendo 2 flujos de datos:

- Flujo 1: origen `pc1`, se quiere restringir con TBF a una velocidad de `1mbit`[1] y una cubeta de 10k bytes.
- Flujo 2: origen `pc2` se va a restringir a una velocidad de `2mbit` y una cubeta de 10k bytes.

Utiliza `tc` para definir esta configuración en la interfaz `eth0` de `r1`, que es la interfaz de entrada para los flujos 1 y 2. Haz que se aplique primero el filtro del flujo número 1 y después el del número 2. Guarda esta configuración en un fichero de *script* con el nombre `tc-ingress.sh`:

```
#!/bin/sh

# Esto es un comentario

echo "Borrando la disciplina de cola ingress en la interfaz eth0"
```

[1] En todo este enunciado, escribiremos las velocidades de transmisión con la notación que requiere la sintaxis de `tc`. Por lo tanto, escribiremos `1mbit` para referirnos a una velocidad de 1 Mbps

```
tc qdisc del ...

echo "Creando la disciplina de cola ingress en la interfaz eth0"
tc qdisc add ...
...
```

Una vez creado el *script* recuerda darle permisos de ejecución.

Si prefieres, puedes editar el script en la máquina real (Ubuntu) con un editor gráfico y luego ejecutarlo.

IMPORTANTE: Escribe el *script* de forma que sólo borre la disciplina de cola si está definida, para que no dé un error. Para ello, utiliza el comando `tc qdisc show dev eth0` y comprueba su salida. Si no devuelve nada, es que no hay ninguna qdisc definida, si devuelve algo es que hay una definida y conviene borrarla primero antes de añadirla.

Incluye dentro de la memoria el contenido del *script*, así como de los *scripts* que desarrolles en los siguientes apartados.

1. Consulta la configuración actual de las disciplinas de cola configuradas por defecto en `r1`. Indica qué resultado has obtenido para cada una de las colas.

2. Realiza una prueba de tráfico como la del apartado anterior:

   - Inicia una captura de tráfico en la interfaz `eth1` de `r1` (`tc-03.cap`)
   - Arranca dos clientes y 2 servidores tal y como lo hiciste en el apartado 1.1.2.
   - Interrumpe las capturas cuando los servidores hayan terminado de recibir todo el tráfico.

3. Explica las estadísticas que muestran los servidores.

4. Carga las capturas en `wireshark` y muestra cada uno de los flujos de forma gráfica. Incluye una imagen en la memoria que muestre los flujos de ambas capturas de forma gráfica. Explica el ancho de banda medido para cada uno de los flujos.

5. Consulta la configuración actual de las disciplina de cola configurada a la entrada en `eth0`. Indica el número de paquetes recibidos y el número de paquetes descartados.

1.3. Disciplinas de colas para el tráfico de salida

1.3.1. Token Bucket Filter (TBF)

Mantén la configuración del tráfico de entrada en `r1` que has realizado en el apartado anterior en el *script* `tc-ingress.sh`.

- Define en `r1` para su interfaz `eth1` una disciplina TBF de salida con tasa de envío de `1.5mbit`, tamaño de cubeta 10k y latencia 10 ms, y guarda la configuración en un nuevo *script* `tc-egress-tbf.sh`.

- Inicia una captura de tráfico en la interfaz `eth1` de `r1` (`tc-04.cap`).

- Arranca 2 clientes y 2 servidores tal y como lo hiciste en el apartado 1.1.2.

- Interrumpe la captura cuando los servidores hayan terminado de recibir todo el tráfico.

A continuación analiza los resultados obtenidos:

1. Explica las estadísticas que muestran los servidores.

2. Carga la captura en `wireshark` y muestra cada uno de los flujos de forma gráfica. Explica el ancho de banda medido para cada uno de los flujos.

Modifica la configuración de TBF de salida para que ahora tenga una latencia de 20 segundos (NOTA: ahora son 20 segundos en vez de 10 milisegundos) y realiza la misma prueba que antes [2].

Llama ahora a la captura `tc-05.cap`.

Interrumpe la captura cuando haya pasado tiempo suficiente para que termine de llegar todo el tráfico a los servidores (será unos 20 segundos después de que comenzó a enviarse). A continuación analiza los resultados obtenidos:

3. Explica las estadísticas que muestran los servidores.

4. Carga la captura en `wireshark` y muestra cada uno de los flujos de forma gráfica. Incluye una imagen en la memoria que muestre los flujos de forma gráfica. Explica el ancho de banda medido para cada uno de los flujos.

5. ¿Cuánto tiempo ha tardado `r1` en realizar el reenvío de todo el tráfico? Relaciona este valor con la cantidad de datos que tenía que reenviar y la tasa de envío que estaba utilizando `r1`. Del tráfico originalmente enviado por `pc1` y `pc2`, ¿cuánto ha sido descartado en la disciplina de cola asociada a `eth0` de `r1`? ¿Y cuánto ha sido descartado en la disciplina asociada a `eth1` de `r1`?

1.3.2. TBF + PRIO

Mantén la configuración del tráfico de entrada en `r1` que has realizado en el apartado anterior en el *script* `tc-ingress.sh`. Borra la disciplina de cola de salida TBF configurada en la interfaz `eth1` de `r1`.

La configuración TBF en el apartado 1.3.1 permite gestionar la tasa de envío para que no supere el valor configurado, en nuestro caso 1.5Mbit. Esta disciplina de cola es sin clases y trata a todos los paquetes por igual. Ahora vamos a querer fijar la tasa de envío de `r1` pero tratando los paquetes con diferentes prioridades.

Toma como punto de partida esta configuración para que ahora se atienda el tráfico de salida según diferentes prioridades, configurando una disciplina de cola con prioridad que sea hija de la disciplina TBF.

- Escribe un *script* en `r1`, `tc-egress-tbf-prio.sh`, para configurar TBF con los siguientes parámetros: ancho de banda `1.5mbit`, cubeta 10k y latencia 20s. Crea una disciplina de cola hija con prioridad de tal forma que se asignen las siguientes prioridades:
  - Prioridad 1 (más prioritario): tráfico de la dirección IP origen `pc1`
  - Prioridad 2 (prioridad intermedia): tráfico de la dirección IP origen `pc2`
  - Prioridad 3 (menos prioritario): sin definir, pues no lo necesitamos.

[2] Ten en cuenta que ahora el tráfico quedará en la cola de la disciplina TBF esperando a ser cursado según la tasa de envío que hemos configurado. El cliente terminará de enviar a los 10 segundos y esperará a recibir el informe del servidor. Sin embargo, el servidor no acabará de recibir (y por tanto no enviará el informe) hasta que TBF no termine de atender el tráfico de la cola de salida, que será más de 10 segundos. Al no recibir el cliente el informe del servidor, terminará imprimiendo un `Warning`. De la misma forma cuando el servidor haya terminado de recibir y envíe el informe al cliente, éste ya habrá terminado su ejecución e imprimirá un mensaje indicando que no ha podido enviar el informe al cliente: `Connection refused`.

- Inicia una captura de tráfico en la interfaz `eth1` de `r1` (`tc-06.cap`).
- Arranca 2 clientes y 2 servidores tal y como lo hiciste en el apartado 1.1.2.
- Interrumpe la captura cuando haya pasado tiempo suficiente para que termine de llegar todo el tráfico a los servidores (será unos 20 segundos después de que comenzó a enviarse).

A continuación analiza los resultados obtenidos:

1. Explica las estadísticas que muestran los servidores.
2. Carga la captura en wireshark y muestra cada uno de los flujos de forma gráfica. Incluye una imagen en la memoria que muestre los flujos de forma gráfica. Explica la evolución en el tiempo del ancho de banda medido para cada uno de los flujos.

1.3.3. Hierarchical token Bucket (HTB)

Mantén la configuración del tráfico de entrada en `r1` que has realizado en el apartado anterior en el *script* `tc-ingress.sh`. Borra la disciplina de cola de salida configurada en la interfaz `eth1` de `r1`.

- Escribe un *script* en `r1`, `tc-egress-htb.sh`, para configurar en su interfaz `eth1` una disciplina HTB de salida con ancho de banda `1.2mbit`. Reparte el ancho de banda de esta interfaz de salida de la siguiente forma:
  - `700kbit` para el tráfico con origen en `pc1`, `ceil 700kbit`.
  - `500kbit` para el tráfico con origen en `pc2`, `ceil 500kbit`.
- Inicia una captura de tráfico en la interfaz `eth1` de `r1` (`tc-07.cap`).
- Arranca 2 clientes y 2 servidores tal y como lo hiciste en el apartado 1.1.2.
- Interrumpe la captura cuando haya pasado tiempo suficiente para que termine de llegar todo el tráfico a los servidores (será unos 20 segundos después de que comenzó a enviarse). `iperf`.

A continuación analiza los resultados obtenidos:

1. Explica las estadísticas que muestran los servidores.
2. Carga la captura en `wireshark` y muestra cada uno de los flujos de forma gráfica. Explica la evolución en el tiempo del ancho de banda medido para cada uno de los flujos.

Modifica la configuración de `ceil` en cada uno de los flujos para que puedan utilizar `1.2Mbit`. Realiza la misma prueba que antes capturando de nuevo el tráfico (`tc-08.cap`) y analiza los resultados obtenidos:

3. Explica las estadísticas que muestran los servidores.
4. Carga la captura en `wireshark` y muestra cada uno de los flujos de forma gráfica. Incluye una imagen en la memoria que muestre los flujos de forma gráfica. Explica la evolución en el tiempo del ancho de banda medido para cada uno de los flujos.

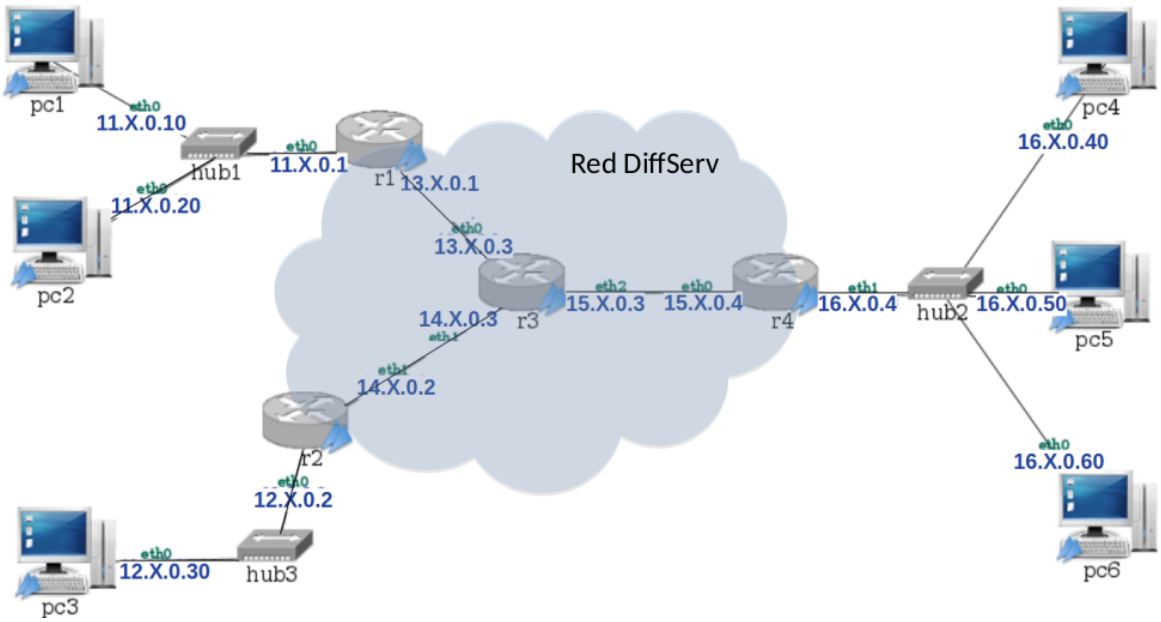

Figura 2: Escenario para DiffServ

2. Diffserv

Descarga tu escenario de red para esta práctica del siguiente enlace:

`http://mobiquo.gsyc.urjc.es/practicas/stma/p4.html`

Descomprime el fichero que contiene el escenario de NetGUI lab-DiffServ.tgz para realizar la práctica de diffServ en Linux.

En el escenario de la figura 2 se va a configurar la red para que el tráfico desde `pc1`, `pc2` y `pc3` envíen paquetes a `pc4`, `pc5` y `pc6` atravesando una red diffServ. Configura las direcciones IP en tu escenario utilizando las tus 4 subredes de la práctica 1, y elige las subredes que quieras como subdred 13.X.0.0/24 y subdred 14.X.0.0/24.

Para esta práctica se distinguirán 4 calidades diferentes, con los códigos EF, AF31, AF21 y AF11.

2.1. Configuración de función policing y marcado de tráfico en DSCP

Utiliza la herramienta `tc` para garantizar que el tráfico que entra en `r1` cumple las siguientes características:

- La red diffServ deberá garantizar a la entrada los siguientes anchos de banda para `pc1`, descartando el tráfico sobrante:
  - Flujo 1: máximo 1.2mbit con ráfaga 10k para el tráfico dirigido a `pc4`, marcado con calidad EF. Si se supera este ancho de banda, el tráfico quedará clasificado dentro del flujo 2.
  - Flujo 2: máximo de 600kbit y ráfaga 10k, marcado con calidad AF31. Si se supera este ancho de banda, el tráfico será descartado definitivamente en `r1`.
- La red diffServ deberá garantizar a la entrada los siguientes anchos de banda para `pc2`, descartando el tráfico sobrante:
  - Flujo 3: máximo 300kbit con ráfaga 10k para el tráfico dirigido a `pc5`, marcado con AF21. Si se supera este ancho de banda, el tráfico quedará clasificado dentro del flujo 4.

- Flujo 4: máximo de 400kbit y ráfaga 10k, marcado con AF11. Si se supera este ancho de banda, el tráfico será descartado definitivamente en `r1`.

Utiliza la herramienta `tc` para garantizar que el tráfico que entra en `r2` cumple las siguientes características:

- La red diffServ deberá garantizar a la entrada los siguientes anchos de banda para `pc3`, descartando el tráfico sobrante:
  - Flujo 5: máximo 400kbit con ráfaga 10k dirigido a `pc6`, marcado con AF31. Si se supera este ancho de banda, el tráfico quedará clasificado dentro del flujo 6.
  - Flujo 6: máximo 300kbit con ráfaga 10k dirigido a `pc6`, marcado con AF21. Si se supera este ancho de banda, el tráfico quedará clasificado dentro del flujo 7.
  - Flujo 7: máximo 100kbit con ráfaga 10k, marcado con AF11. Si se supera este ancho de banda, el tráfico será descartado definitivamente en `r2`.

1. Realiza *scripts* para `r1` y otro para `r2` donde se configuren estos perfiles de tráfico. Incluye dichos *scripts* en la memoria.

2. Inicia capturas: `diffServ-01.cap` en la subdred 13.X.0.0/24, `diffServ-02.cap` en la subdred 14.X.0.0/24 y `diffServ-03.cap` en la subdred 15.X.0.0/24 para que capture el tráfico que se genera en tu escenario por el envío "simultáneo" de:
   - Desde el `pc1` 2M a `pc4`
   - Desde el `pc2` 1.5M a `pc5`
   - Desde el `pc3` 1M a `pc6`

3. Interrumpe las capturas, al menos 1 minuto después de que la transmisión haya terminado. Comprueba que el resultado es el esperado:
   - El tráfico que entra en la red diffServ es el que se ha especificado en el control de admisión.
   - El tráfico está marcado según las especificaciones anteriores.

   Para ello, consulta las gráficas `IO graphs` de Wireshark aplicando los filtros sobre las marcas DSCP de tal forma que se muestre cada calidad marcada de cada una de las fuentes:
   - Tráfico de EF
   - Tráfico de AF31
     - Total
     - Con origen en `pc1`.
     - Con origen en `pc3`.
   - Tráfico de AF21
     - Total
     - Con origen en `pc2`.
     - Con origen en `pc3`.
   - Tráfico de AF11
     - Total
     - Con origen en `pc2`.
     - Con origen en `pc3`.

   Explica los resultados obtenidos e incluye todas las gráficas que consideres necesarias en la memoria.

2.2. Tratamiento de tráfico en función del marcado DSCP

Mantén la configuración realizada en `r1`, `r2`.

Se establecen los siguientes parámetros de calidad dentro del router del núcleo diffServ (`r3`) para cada una de las calidades definidas. Configura HTB con ancho de banda 2.4Mbit para compartir entre todos los flujos con el siguiente patrón:

- EF: HTB 1Mbit como mínimo y 1Mbit como máximo.
- AF31: HTB 500kbit como mínimo y 500kbit como máximo.
- AF21: HTB 400kbit como minimo y 400kbit como máximo.
- AF11: HTB 200kbit como mínimo y 200kbit como máximo.

1. Realiza un *script* para `r3` donde se configure esta disciplina de cola según el marcado de los paquetes e incluye dicho *script* en la memoria.
2. Inicia una captura (`diffServ-04.cap`) en la subdred 15.X.0.0/24 para que capture el tráfico que se genera en tu escenario por el envío "simultáneo" de:
   - Desde `pc1`: 2M a `pc4`
   - Desde `pc2`: 1.5M a `pc5`
   - Desde `pc3`: 1M a `pc6`

   Espera al menos 2 minutos después de que haya terminado de enviarse el tráfico de `pc1`, `pc2` y `pc3` antes de interrumpir la captura de tráfico.
3. Comprueba que el resultado es el esperado, es decir, el tráfico sigue el perfil indicado en las especificaciones anteriores. Para ello, consulta las gráficas `IO graphs` de Wireshark aplicando los filtros sobre las marcas DSCP de tal forma que se muestre cada calidad marcada de cada una de las fuentes incluyendo dichas imágenes en la memoria:
   - Tráfico de EF
   - Tráfico de AF31
   - Tráfico de AF21
   - Tráfico de AF11

   Explica los resultados obtenidos y explica si alguno de los flujos ha encolado tráfico para enviarlo posteriormente a los 10 segundos que dura la transmisión de `iperf`.
4. Modifica la configuración de HTB en `r3` para que si algún flujo no está utilizando el ancho de banda que tiene garantizado lo puedan usar el resto de flujos y vuelve a hacer una captura de tráfico (`diffServ-05.cap`) en la subdred 15.X.0.0/24. Explica qué modificaciones has tenido que hacer en el *script*.
5. Explica los resultados obtenidos e incluye las gráficas `IO graphs` que consideres necesarias.

Normas de entrega

Es necesario entregar a través del aula virtual los siguientes ficheros:

- Memoria en formato pdf donde se explique razonadamente cada uno de los apartados de este enunciado y se incluya el contenido de los scripts que hayas desarrollado.
- Fichero `p4.tgz` o `p4.zip` resultado de comprimir **una carpeta de nombre p1** que contenga en su interior los ficheros de captura de tráfico: `tc-01.cap` hasta `tc-08.cap` y desde `diffServ-01.cap` hasta `diffServ-05.cap`.

Sistemas Telemáticos para Medios Audiovisuales
Práctica 5: HTTP

GSyC
Departamento de Teoría de la Señal y Comunicaciones
y Sistemas Telemáticos y Computación

Septiembre de 2022

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.

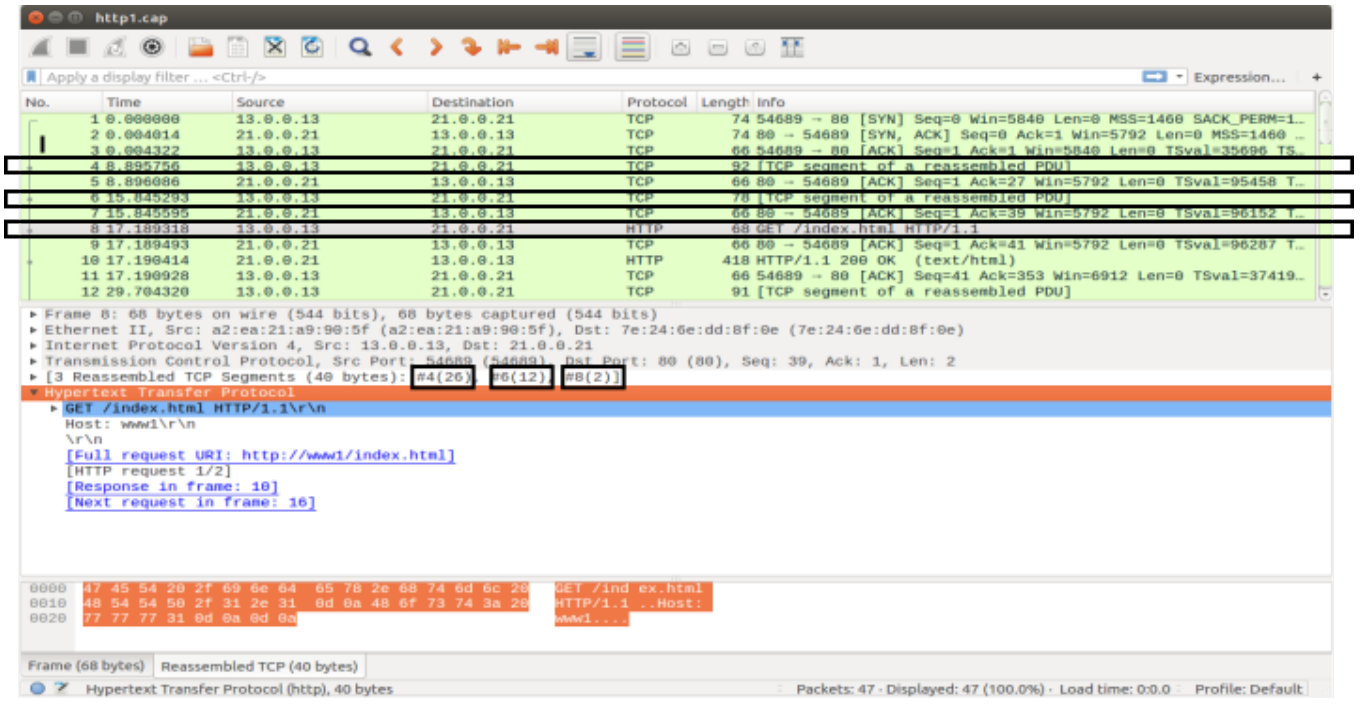

Figura 1: Mensaje HTTP compuesto de varios segmentos.

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.

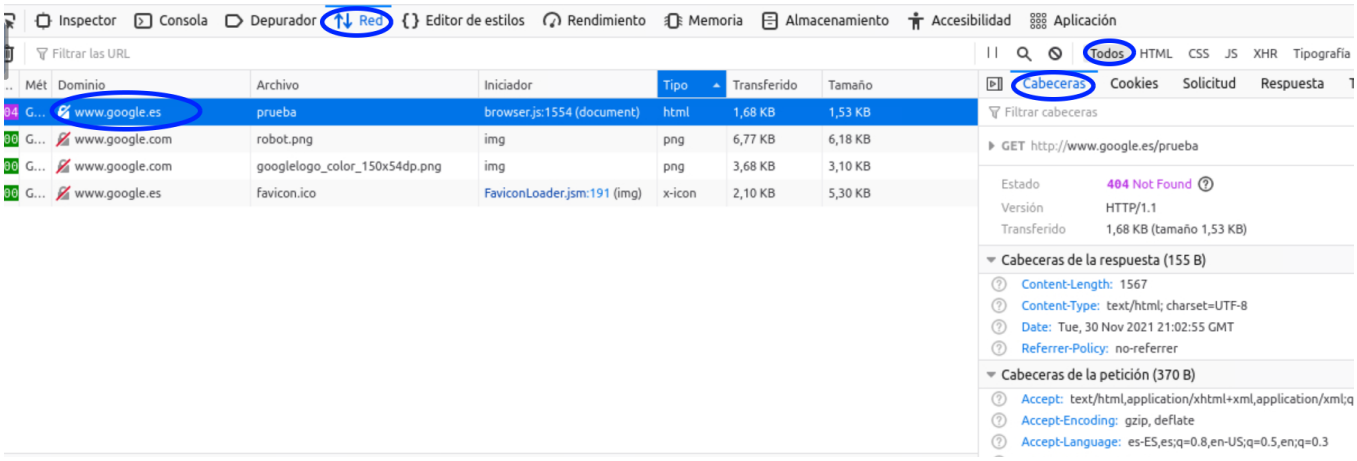

Figura 3: Cabeceras HTTP.

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.

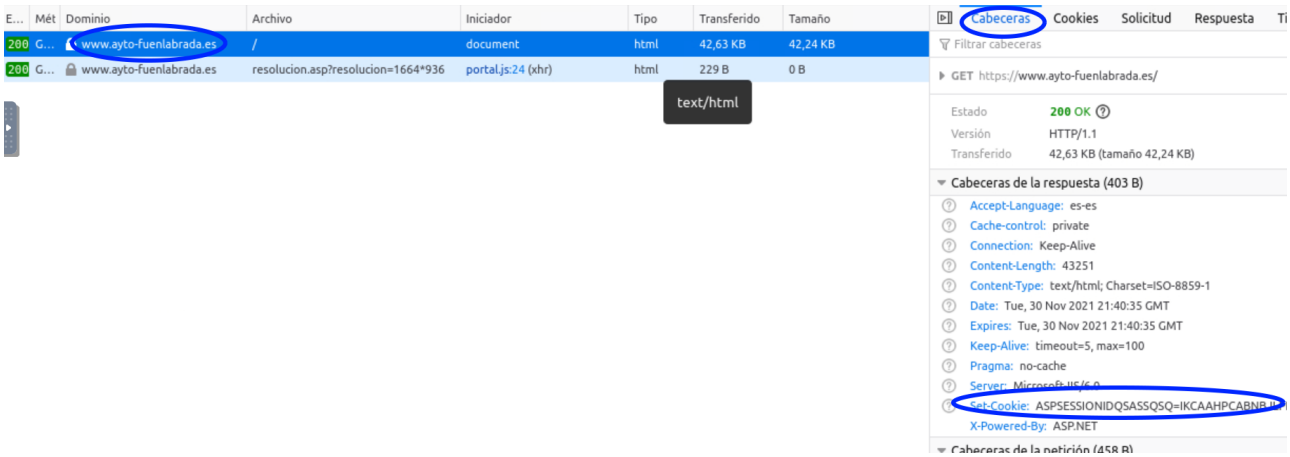

Figura 4: Cookies

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.

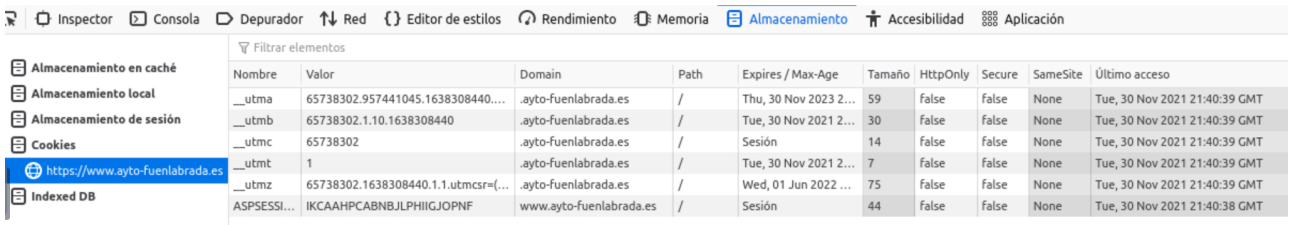

Figura 5: Almacenamiento de Cookies

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 `p5.pdf` con la memoria de la práctica **en formato PDF**.

Sistemas Telemáticos para Medios Audiovisuales
Práctica 6a: Claves

GSyC
Departamento de Teoría de la Señal y Comunicaciones
y Sistemas Telemáticos y Computación

Septiembre de 2022

1. Ejercicio 1

Se ha diseñado un sistema de comunicación que pretende que los usuarios puedan intercambiar información de manera anónima. El objetivo es dificultar que alguien que intercepte uno de los mensajes pueda conocer qué nodo envió originalmente el mensaje, ni cuál es el destinatario final del mismo, ni cuál es el contenido del mensaje.

Para conseguir este objetivo el mensaje se va enviando a través de una serie de nodos, elegidos por el nodo origen de la comunicación.

El nodo origen de una comunicación tiene que indicar en el mensaje que envía dos tipos de información:

- La secuencia de nodos que tiene que seguir el mensaje que envía
- El Contenido del Mensaje, que incluye la dirección del nodo que envía originalmente el mensaje, y el texto del mensaje.

Cuando un nodo recibe un mensaje, tiene que enviárselo al primero de los nodos especificados en la secuencia de nodos que viene en el mensaje, eliminando la primera entrada de la secuencia de nodos antes de enviar el mensaje.

Ejemplo con 5 ordenadores, X, B, C, D, Z, con direcciones IP $IP_X, IP_B, IP_C, IP_D, IP_Z$ respectivamente:

Supongamos que X quiere enviar el texto $mensajeParaZ$ a Z a través de la ruta $X \Rightarrow B \Rightarrow C \Rightarrow D \Rightarrow Z$, y que X conoce $K_B^+, K_C^+, K_D^+, K_Z^+$.

1º) X le envía a B un datagrama IP en cuyo campo de datos va la siguiente información:
   - Secuencia de nodos: $\boxed{K_B^+(IP_C) \mid K_C^+(IP_D) \mid K_D^+(IP_Z) \mid K_Z^+(IP_Z)}$
   - Contenido del Mensaje: $\boxed{K_Z^+(IP_X, mensajeParaZ)}$

2º) B descifra el primer componente de la secuencia de nodos recibida, y aprende que el siguiente salto es C. B le envía entonces a C un datagrama IP con la siguiente información en su campo de datos:
   - Secuencia de nodos: $\boxed{K_C^+(IP_D) \mid K_D^+(IP_Z) \mid K_Z^+(IP_Z)}$
   - Contenido del Mensaje: $\boxed{K_Z^+(IP_X, mensajeParaZ)}$

3º) C descifra el primer componente de la secuencia de nodos recibida, y aprende que el siguiente salto es D. C le envía a D:
   - Secuencia de nodos: $\boxed{K_D^+(IP_Z) \mid K_Z^+(IP_Z)}$
   - Contenido del Mensaje: $\boxed{K_Z^+(IP_X, mensajeParaZ)}$

4º) D descifra el primer componente de la secuencia de nodos recibida, y aprende que el siguiente salto es Z. D le envía a Z:
   - Secuencia de nodos: $\boxed{K_Z^+(IP_Z)}$
   - Contenido del Mensaje: $\boxed{K_Z^+(IP_X, mensajeParaZ)}$

5º) Z descifra el primer y único componente de la secuencia de nodos recibida, y aprende que él es el nodo destinatario. Entonces Z descifra el Contenido del Mensaje, sabiendo así que el mensaje lo ha enviado originalmente IP_X, y que el mensaje que le quería transmitir a Z era $mensajeParaZ$.

Preguntas

1. Explica si el nodo receptor del mensaje Z puede o no descifrar el mensaje para acceder a su contenido.
2. Explica si el nodo receptor del mensaje Z estar seguro de la confidencialidad del mensaje, es decir, de que ningún otro nodo ha podido descifrarlo.
3. Explica si el nodo receptor del mensaje Z puede autenticar al nodo emisor del mensaje X.
4. Explica si el nodo receptor del mensaje Z puede estar seguro de la integridad del mensaje, es decir, que ningún otro nodo ha podido alterar el contenido del mensaje.
5. Explica si un nodo cualquiera que intercepte un mensaje destinado al nodo Z puede conocer el texto del $mensajeParaZ$.
6. Explica si un nodo cualquiera que intercepte un mensaje destinado al nodo Z puede conocer el destino final del mensaje.
7. Explica si un nodo cualquiera que intercepte un mensaje destinado al nodo Z puede conocer el nodo que creó el mensaje.

2. Ejercicio 2

En una sistema existen las siguientes autoridades de certificación CA1 y CA2, ambas autoridades de certificación han incluido sus propios certificados autofirmados en las aplicaciones de comunicaciones que se usan dentro de este sistema.

Alicia tiene un certificado de su clave pública firmado por CA1 y Roberto tiene un certificado de su clave pública firmado por CA2.

Cuando Alicia se quiere comunicar con Roberto elige una clave simétrica de sesión para la comunicación que quiere establecer, K_s. Ésta es la clave que usará para convertir sus mensajes en confidenciales.

Preguntas

1. Indica cómo crees que debería enviarle la clave K_s de Alicia a Roberto.
2. Alicia no tiene la clave pública de Roberto ni Roberto la de Alicia. Indica cómo podría conseguir Alicia la K_R^+, sin quedar físicamente para intercambiarse las claves, y como puede Alicia estar segura de que esta clave se corresponde con la de Roberto.
3. Con este sistema, ¿puede estar Alicia segura de que los mensajes que envía a Roberto son confidenciales y de que en realidad se está comunicando con Roberto? En caso negativo, explica cómo conseguirías estas propiedades en los mensajes enviados desde Alicia a Roberto.
4. Con este sistema, ¿puede estar Roberto seguro de que los mensajes son confidenciales y provienen de Alicia? En caso negativo, explica cómo conseguirías estas propiedades en los mensajes enviados desde Alicia a Roberto.
5. El certificado de la clave públicad de Roberto ha caducado y ya no es válido. Roberto decide cambiar de autoridad de certificación y consigue un certificado de su clave pública emitido por la autoridad de certificación CA3. Esta autoridad de certificación CA3 no ha incluido su certificado autofirmado en las aplicaciones de comunicaciones del sistema, pero CA3 tiene un certificado de la clave pública de CA3 firmado por CA2. Indica si ahora Alicia podría enviar a Roberto mensajes confidenciales y auténticos y explica cómo lo haría.

3. Ejercicio 3

Abre el navegador Firefox y a través del menú selecciona la opción: Editar → Preferencias → Privacidad y Seguridad → Seguridad → Certificados → Ver Certificados.

En la pestaña "Autoridades" verás los certificados de las autoridades de certificación de primer nivel. Cualquier certificado que venga firmado por las autoridades de certificación que se encuentran en esta pestaña podrá ser verificado ya que se poseen de forma fiable las claves públicas de estas autoridades de certificación que permiten comprobar las firmas.

1. Escribe en la URL del navegador la siguiente dirección: `www.amazon.es`, una vez que se haya cargado la página verás junto a la URL un candado verde, pulsa sobre él y luego sobre la flecha derecha al lado de "Conexión segura" ("Mostrar detalles de la conexión"). Indica cuál es la autoridad de certificación que ha verificado esta conexión segura.

2. En esa ventana de detalles de la conexión, pulsa sobre el botón "Más información" y luego en "Ver Certificado" y en la pestaña "Detalles". Indica cuál es la jerarquía de certificados que se está utilizando para verificar a Amazon. Selecciona empezando por www.amazon.es el campo "Emisor" y ve comprobando quiénes han sido las entidades que han generado los certificados que aparecen en la jerarquía. Comprueba la cadena de todos los certificados. Señala qué certificados de la jerarquía están autofirmados.

3. Vuelve a visitar la información de certificados de las Preferencias (Editar → Preferencias → Privacidad y Seguridad → ... → Ver Certificados). Observarás que las dos entidades que aparecen en la jerarquía de certificados de Amazon tienen instalado su certificado. Una de ellas muestra su certificado como `Builtin object token`, es decir, se trata de un certificado autofirmado de una autoridad de certificación raíz que venía instalado con la aplicación Firefox. El otro certificado se muestra como `Disp. software de seguridad`, por lo que no es un certificado autofirmado y la entidad que lo ha firmado es una autoridad de certificación raíz. Indica cuál de ellos es `Builtin object token` y cuál es `Disp. software de seguridad`.

4. 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**.

Sistemas Telemáticos para Medios Audiovisuales
Práctica 6b: Cortafuegos *(firewalls)*

GSyC
Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación
URJC

Septiembre de 2022

Antes de comenzar, descarga tu escenario del siguiente enlace donde deberás introducir tu número de DNI (8 dígitos) con la letra correspondiente:

`http://mobiquo.gsyc.es/practicas/stma/p6.html`

En la figura 1 se representa un conjunto de subredes y máquinas (`pc1`, `pc2`, `pc4`, `pc5`, `r1`, `r2` y `firewall`) que pertenecen a una determinada empresa y su conexión a Internet a través de la máquina `firewall`. La empresa tiene definidas un conjunto de subredes de ámbito privado:

- 10.X.0.0/24: `r1(eth1)`, `pc1`, `pc2`
- 10.X.1.0/24: `firewall(eth0)`, `r1(eth0)`, `r2(eth0)`
- 10.X.2.0/24: `r2(eth1)`, `pc3`

Adicionalmente, la empresa tiene las máquinas `pc4` y `pc5` que se encuentran en una subred pública: 100.X.0.0/24. Estas máquinas proporcionan servicios básicos de la empresa: servidor de HTTP y servidor de fecha y hora. A este tipo configuración, donde la empresa tiene una o varias subredes públicas para ofrecer servicios a Internet se le denomina zona desmilitarizada o DMZ (DeMilitarized Zone).

Todas las máquinas de la empresa se conectan a Internet a través de la máquina `firewall` y la subred 100.X.1.0/24.

En este escenario, se considera que Internet está formado por las siguientes máquinas: `r3`, `r4`, `r5`, `pc6` y `pc7` que se encuentran conectadas a las siguientes subredes públicas:

- 100.X.1.0/24: `r3(eth0)`
- 100.X.2.0/24: `r3(eth1)`, `r5(eth2)`
- 100.X.3.0/24: `r3(eth2)`, `r4(eth2)`
- 100.X.4.0/24: `r4(eth1)`, `r5(eth0)`
- 100.X.5.0/24: `r4(eth0)`, `pc6`
- 100.X.6.0/24: `r5(eth1)`, `pc7`

Arranca de una en una todas las máquinas de la figura.

1. Introducción

A continuación se proporcionan algunos consejos para facilitar la realización de la práctica.

1.1. Edición y ejecución de *scripts*

En esta práctica se configurará la máquina `firewall` para que actúe como traductor de direcciones y como cortafuegos. Habrá que definir varias reglas utilizando `iptables`. Por este motivo, es recomendable guardar dichas reglas en un fichero *script de shell*.

Para hacer un *script de shell* crea un fichero de texto de nombre, por ejemplo, `fw.sh`, editándolo con `mcedit` en la forma:

```
mcedit fw.sh
```

La primera línea del fichero debe ser `#!/bin/sh` y las siguientes líneas serán la definición de las reglas para `iptables` tal y como se escribirían en el terminal:

```
#!/bin/sh

# Esto es un comentario

iptables -t nat -F
iptables -t nat -Z
iptables ...
...
...
```

Una vez creado el *script* debes darle permisos de ejecución con la orden:

```
chmod 755 fw.sh
```

A partir de ahora ya podrás ejecutarlo, escribiendo:

```
./fw.sh
```

Considera la posibilidad de editar y guardar el script en el sistema de ficheros de la máquina real, ejecutándolo desde dentro de la máquina virtual. Así, si tu script `fw.sh` está almacenado directamente en tu HOME de la máquina real, podrías editarlo en ella con un editor gráfico (por ejemplo, `gedit`) y luego ejecutarlo en la máquina `firewall` escribiendo dentro de esa máquina virtual:

```
/hosthome/fw.sh
```

1.2. Comprobación de la configuración del *firewall*

Durante la práctica frecuentemente tendrás que ir comprobando que el *firewall* está correctamente configurado, es decir:

- deja pasar el tráfico que debe dejar pasar
- impide el paso del tráfico que debe impedir
- realiza la traducción de direcciones IP necesaria para que no aparezcan en Internet paquetes con direcciones privadas

Para ello deberás emplear la herramienta *netcat* (`nc`) (ya utilizada en otras prácticas con anterioridad) que permite arrancar aplicaciones TCP y UDP en modo cliente o servidor.

El enunciado de la práctica te irá indicando cuándo y en qué máquinas debes lanzar un cliente o un servidor TCP o UDP para ir probando la configuración del *firewall*. Consulta la documentación adjunta para recordar la sintaxis de *netcat*.

2. Traducción de direcciones y puertos en el *firewall*: tabla `nat`

2.1. Clientes en la red privada, servidores externos

Configura un *script* `fw1.sh` en el *firewall* para que:

- se borren las reglas que hubiera configuradas previamente en la tabla `nat`
- se reinicien los contadores de la tabla `nat`
- se realice la traducción de direcciones para el tráfico saliente de las redes privadas (SNAT) y su correspondiente tráfico de respuesta.

Incluye el script en la memoria.

2.1.1. Pruebas con TCP

Ejecuta el *script* `fw1.sh` de 2.1.

1. Captura el tráfico en `r3-eth0` (`iptables-01.cap`) y en `firewall-eth0` (`iptables-02.cap`) para ver los paquetes dentro de la red de la Empresa y por Internet. Arranca las siguientes aplicaciones:

   - `nc` como servidor TCP en `pc6`, puerto 7777
   - `nc` como cliente TCP en `pc1`

   Sin escribir nada ni en el cliente ni en el servidor, consulta la información de `ip_conntrack` del `firewall` cada medio segundo. Para hacerlo automáticamente, en vez de repetir el comando utiliza `watch` de la siguiente forma:

   ```
   firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack
   ```

   Explica el número de paquetes que se han observado en cada sentido, razonando la respuesta, indicando de qué paquetes se trata (recuerda que estamos ante una conexión TCP).

2. Introduce una palabra en la entrada estándar de `pc1`, pulsa `<Enter>` y observa los cambios en `ip_conntrack`. Explica a qué se deben.

3. Realiza un Ctrl+C en el terminal de `pc1` para interrumpir la ejecución de `nc`. Observa los cambios en `ip_conntrack` y explica a qué se deben.

4. Interrumpe las capturas, y estúdialas. En particular, identifica los mismos paquetes en las 2 capturas, y observa cómo cambian las direcciones IP de los mismos paquetes según viajen dentro de la EMPRESA o por INTERNET. Explica el resultado.

5. Consulta la lista de reglas en el `firewall` con:

   ```
   firewall:~# iptables -t nat -L -v -n
   ```

Obseva qué regla(s) están cumpliendo los paquetes y cuántas veces se cumple(n).

6. Vuelve a repetir la misma prueba anterior (sin necesidad de realizar las capturas de tráfico): lanza servidor y cliente, intercambia tráfico, y termina la conexión. Vuelve a mirar qué regla(s) se están cumpliendo y cuántas veces se cumple(n).

2.1.2. Pruebas con UDP

Ejecuta el *script* `fw1.sh` de 2.1 para que se reinicien los contadores de paquetes de iptables, compruébalo consultando la lista de reglas del firewall.

1. Captura el tráfico en `r3-eth0` (`iptables-03.cap`) y en `firewall-eth0` (`iptables-04.cap`) para ver los paquetes dentro de la red de la Empresa y por Internet. Arranca las siguientes aplicaciones:

   - `nc` como servidor UDP en `pc6`, puerto 7777
   - `nc` como cliente UDP en `pc2`

   Realiza las siguientes pruebas:

   a) Sin escribir nada ni en el cliente ni en el servidor, consulta la información de `ip_conntrack` del `firewall` cada medio segundo. Recuerda que el tráfico es ahora UDP y no hay conexiones propiamente dichas. Explica el resultado.

   b) Escribe 5 líneas en el terminal de `pc2` para que se las envíe a `pc6`. Explica el número de paquetes enviados en la información que muestra `ip_conntrack`.

   c) Escribe una línea en `pc6` para que se la envíe a `pc2`. Explica nuevamente el número de paquetes en `ip_conntrack`.

   d) Observa el poco tiempo que se mantiene la "asociación" entre cliente y servidor en `ip_contrack`. Indica cuánto ha sido.

   e) Interrumpe la captura y las ejecuciones de `nc`, explica la captura y cómo ésta se relaciona con la información que has visto en `ip_conntrack`.

2. Consulta la lista de reglas en el `firewall`, e indica cuáles se están cumpliendo y cuántas veces se cumplen.

3. Interrumpe la ejecución de cliente y servidor e inicia una nueva comunicación entre un nuevo cliente y un servidor UDP e intercambia tráfico entre ellos para ver cómo evolucionan las cuentas en la lista de reglas. Explica qué reglas se están cumpliendo ahora y cuántas veces se cumplen.

4. Captura de nuevo el tráfico en `r3-eth0` (`iptables-05.cap`) y en `firewall-eth0` (`iptables-06.cap`) para ver los paquetes dentro de la red de la Empresa y por Internet cuando tienes varios clientes desde un mismo puerto origen conectándose a un mismo servidor, para ello inicia:

   - `nc` como servidor UDP en `pc7`, puerto 7777
   - `nc` como cliente UDP en `pc1`, puerto 6666
   - `nc` como cliente UDP en `pc2`, puerto 6666

   Ahora, envía una línea desde `pc1` y después una línea desde `pc2`. Ten en cuenta que `nc` no funciona como las aplicaciones servidoras que pueden atender a varios clientes a la vez. La aplicación `nc` no está preparada para que un servidor se pueda comunicar a la vez con dos clientes, por ello el envío desde `pc2` provocará que `pc7` envíe un ICMP de error a `pc2`. Pero para lo que queremos

comprobar este error no es importante, sólo queremos analizar lo que ocurre en el `firewall` con la traducción de direcciones IP y puertos.

Interrumpe las capturas y analízalas fijándote en las direcciones IP **y puertos** que se utilizan en la red de la EMPRESA y en INTERNET.

2.1.3. Pruebas con ICMP

Ejecuta el *script* `fw1.sh` de 2.1 para que se reinicien los contadores de paquetes de iptables.

1. Ejecuta el siguiente comando en `pc1` (recuerda sustituir la X por el número que te corresponde):

   ```
   pc1:~# ping -c 2 100.X.5.60
   ```

2. Consulta la información de `ip_conntrack` del `firewall`. Verás que no aparece nada. Recuerda que esto se debe a que las "conexiones" que se consideran para los paquetes ICMP es una diferente entre cada *echo request* y su correspondiente *echo reply*, asociación que se "olvida" justo después del *echo reply*.

3. Consulta la lista de reglas en el `firewall`, y mira cuáles se están cumpliendo y cuántas veces.

2.2. Servidores en la red privada, clientes externos

Aunque en una red como la que aparece en la figura, lo habitual es colocar los servidores accesibles desde el exterior en la zona DMZ, para ver cómo funciona DNAT, vamos a permitir que haya servidores accesibles desde el exterior en la red privada interna.

2.2.1. Apertura de puertos TCP

Realiza un nuevo *script* `fw2.sh` en el *firewall* para que:

- se borren las reglas que hubiera configuradas previamente en la tabla `nat`
- se reinicien los contadores de la tabla `nat`
- el tráfico de entrada al firewall destinado al puerto TCP 80 sea redirigido a `pc3`, puerto 80.

Incluye el script en la memoria. Ejecuta dicho script y arranca las siguientes aplicaciones:

- `nc` como servidor TCP en `pc3`, puerto 80
- `nc` como cliente TCP en `pc6`, de forma que su tráfico lo reciba el servidor de `pc3` (NOTA: presta especial atención a los parámetros con los que debes lanzar este cliente). Indica en la memoria el comando que has usado para lanzar el cliente y explica por qué lo has hecho así.

Explica los siguientes resultados:

1. El resultado observado en `ip_conntrack` y la traducción de direcciones IP y puertos realizada.
2. La lista de reglas en el `firewall`, indica cuáles se están cumpliendo y cuántas veces.

2.2.2. Apertura de puertos UDP

Modifica el *script* `fw2.sh` para que, adicionalmente:

- el tráfico de entrada al firewall destinado al puerto UDP 5001 sea redirigido a `pc1`, puerto 5001
- El tráfico de entrada al firewall destinado al puerto UDP 5002 sea redirigido a `pc2`, puerto 5001

Incluye el script en la memoria. Ejecuta el script que acabas de modificar y arranca las aplciaciones:

- `nc` como servidor UDP en `pc1`, puerto 5001
- `nc` como servidor UDP en `pc2`, puerto 5001
- `nc` como cliente UDP en `pc6`, de forma que su tráfico lo reciba el servidor de `pc1`. Indica el comando que has utilizado para lanzar el cliente y explica por qué.
- `nc` como cliente UDP en `pc7`, de forma que su tráfico lo reciba el servidor de `pc2`. Indica el comando que has utilizado para lanzar el cliente y explica por qué.

Explica los siguientes resultados:

1. El resultado observado en `ip_conntrack` y la traducción de direcciones IP y puertos realizada.
2. Consulta la lista de reglas en el `firewall` e indica cuáles se están cumpliendo y cuántas veces.

3. Filtrado de tráfico en el *firewall*: tabla `filter`

Crea un *script* `fw3.sh` en el `firewall` partiendo de la configuración de traducción de direcciones realizada en `fw1.sh` (clientes en la red privada, servidores externos) al que se le añada la siguiente configuración (todas en el mismo *script*). Descripción de las **especificaciones**:

1. Reiniciar la tabla `filter`: borrar su contenido y reiniciar sus contadores.
2. Fijar las políticas por defecto de las cadenas de la tabla `filter`, haciendo que por defecto se descarte todo el tráfico en el `firewall` excepto los paquetes que cree el propio `firewall` (configuración habitual en un *firewall*).
3. Permitir el tráfico de entrada dirigido a las aplicaciones que se están ejecutando en el propio `firewall` únicamente si este tráfico tiene su origen en las subredes privadas de la empresa.
4. Permitir todo el tráfico saliente desde las subredes privadas hacia Internet y el tráfico de respuesta al saliente.

   Ten en cuenta que como has partido del *script* `fw1.sh`, en dicho *script* ya tenías las reglas de la tabla `nat` para la traducción de la dirección IP de origen de los paquetes que reenvía el `firewall` y los paquetes del tráfico entrante de respuesta a éste.
5. Permitir desde Internet únicamente el tráfico entrante nuevo hacia la zona DMZ según las siguientes reglas:

   - acceso a un servidor *echo* existente en `pc4` (UDP, puerto 7). El servidor de *echo* es un servidor que al enviarle una cadena de caracteres, devuelve la misma cadena que se le ha enviado. Para comprobar el acceso a este servidor utiliza `nc` como cliente desde otra máquina.

- acceso a un servidor *daytime* existente en `pc5` (UDP, puerto 13). El servidor *daytime* es un servidor que al enviarle algo, devuelve la fecha y hora de la máquina donde está instalado. Para comprobar el acceso a este servidor utiliza `nc` como cliente desde otra máquina.

6. Permitir únicamente la comunicación entre la red privada y la zona DMZ de la siguiente forma:

   - acceso desde `pc1` a un servidor de *echo* (TCP, puerto 7) existente en `pc4`.

7. Desde la zona DMZ no se debe perimitir iniciar ninguna comunicación con la red privada ni con el propio `firewall`.

Incluye el script en la memoria.

En el escenario se encuentra lanzado en `pc4` un servidor TCP en el puerto 7. Prueba a lanzar un cliente con `nc` desde `pc1` para que se conecte con este servidor. Consulta la lista de reglas en el `firewall` e indica cuáles se están cumpliendo y cuántas veces se cumplen. Es importante que observes que las reglas de la tabla `filter`, si se cumple la condición, se aplican con cada paquete que atraviesa el `firewall` y este comportamiento es diferente a lo que ocurría con las reglas de la tabla `nat`.

3.1. OPCIONAL: Pruebas de la configuración del firewall

A continuación se dan algunas pautas para poder probar cada una de las especificaciones de `fw3.sh`. Para cada prueba, asegúrate de relanzar el *script* para que se reinicien los contadores, y comprueba qué reglas son las que se han aceptado para aceptar o rechazar el tráfico. Cada una de las siguientes especificaciones se corresponden con los puntos descritos en el apartado 2.

Especificación 3

1. Si se arranca una aplicación servidor (TCP o UDP) en la máquina `firewall` sólo podrá aceptar tráfico de un cliente que envíe mensajes desde una de las máquinas de las subredes privadas.

2. No podrá aceptar tráfico desde aplicaciones cliente lanzadas en otras subredes diferentes.

Especificación 4

1. Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de Internet y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de las subredes internas, el tráfico debe poder enviarse del cliente al servidor y del servidor al cliente, observando que el tráfico que sale del firewall con destino a la máquina de Internet no tiene como dirección IP origen la dirección de la máquina que pertenece a la subred privada, sino que lleva la dirección 100.X.1.100.

2. Si se arranca una aplicación cliente en `pc4` o `pc5` para comunicarse con el servidor que se haya arrancado en una de las máquinas de Internet, el `firewall` no debería permitir reenviar ese tráfico hacia Internet.

Especificación 5

1. Si se arranca un cliente `nc` desde una máquina de Internet se debe poder acceder al servidor de *echo* de `pc4`.

2. Si se prueba lo mismo lanzando el cliente desde `pc3`, no debería poder comunicarse.

3. Si se arranca un cliente `nc` desde una máquina de Internet se debería poder obtener la hora de `pc5`. Pulsa `<INTRO>` en el terminal de `nc` y debería obtenerse la hora que le envía `pc5`.

4. No se debe permitir otro tipo de tráfico desde Internet a DMZ. Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de DMZ y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de Internet, el tráfico no debería poder enviarse del cliente al servidor.

Especificación 6

1. Desde `pc1` se debería poder lanzar un cliente `nc` capaz de conectarse con al servidor TCP de *echo* de `pc4`.

2. Si se prueba lo mismo arrancando `nc` desde `pc2` o `pc3` no debería conectarse.

3. No se debe permitir otro tipo de tráfico desde `pc1`, `pc2` o `pc3` con `pc4` o `pc5`. Si se arranca una aplicación servidor (TCP o UDP) en una de esas máquinas y se arranca una aplicación cliente en una de las otras, el tráfico no debería poder enviarse del cliente al servidor.

Especificación 7

1. Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de las subredes privadas y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de DMZ, el tráfico no debería poder enviarse del cliente al servidor ni del servidor al cliente.

2. Si se arranca una aplicación servidor (TCP o UDP) en el `firewall` y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de DMZ, el tráfico no debería poder enviarse del cliente al servidor.

Entrega de la práctica

Es necesario entregar la siguiente documentación:

- Memoria donde se explique razonadamente el diseño y la configuración de cada uno de los apartados de este enunciado, así como las pruebas realizadas para comprobar cada característica del cortafuegos pedida.
- Capturas de tráfico desde `iptables-01.cap` a `iptables-06.cap`.

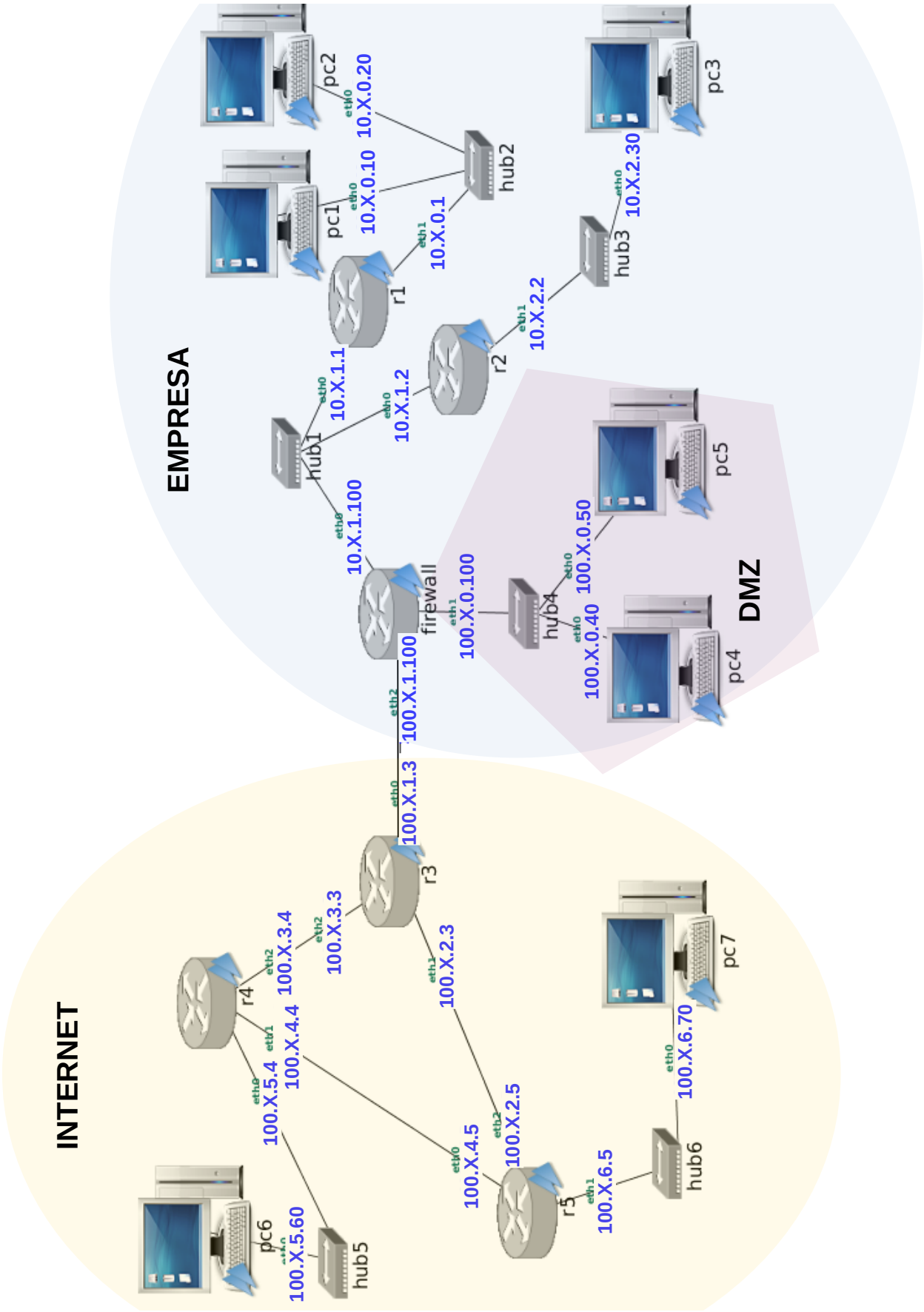

Figura 1: Escenario de red para los ejercicios de configuración de firewall