ETSIIT
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

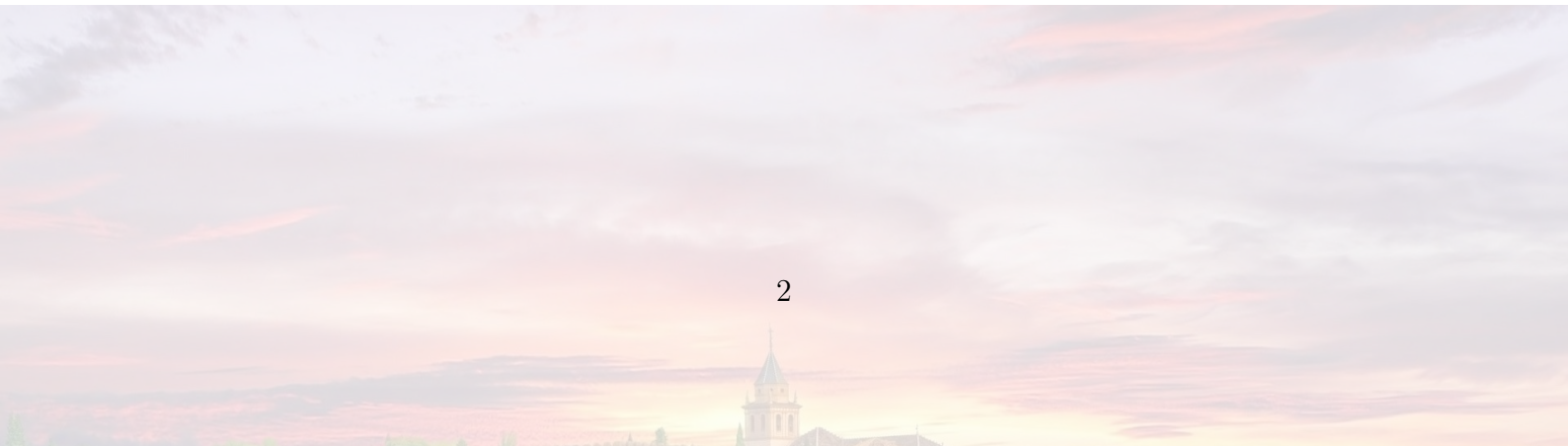
Ingeniería Informática + ADE

Universidad de Granada (UGR)

Autor: Ismael Sallami Moreno

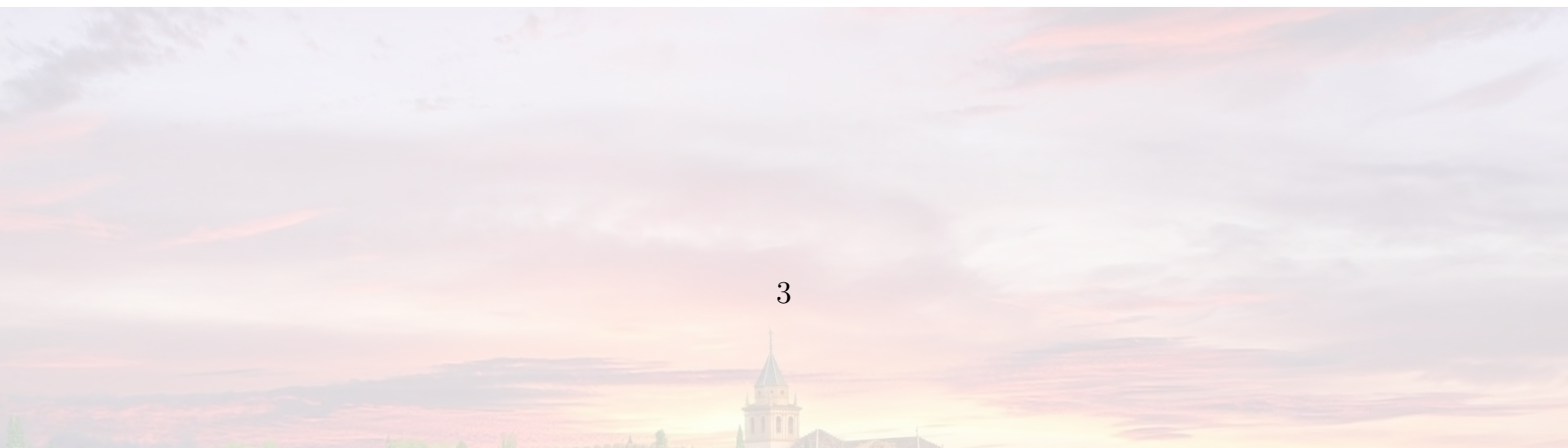
Asignatura: Fundamentos de Redes

Índice



1 Prácticas

1.1. Laboratorio



Práctica 0

Descripción del Laboratorio de Redes

Aula 3.7 de la E.T.S. Ing. Informática y de Telecomunicación

1. Objetivo de la práctica

El objetivo de la presente práctica es realizar una toma de contacto inicial de los alumnos con el Laboratorio de Redes (Aula 3.7 de la ETSIIT), conocer la estructura del mismo y presentar los equipos existentes, que serán utilizados en los distintos programas prácticos correspondientes a las asignaturas impartidas por el área de Ingeniería Telemática de la Universidad de Granada.

Así, se pretende que el alumno adquiera las nociones básicas sobre:

- Estructura general del laboratorio
- Estructura física, equipamiento y cableado
- Estructura lógica de la red del laboratorio

2. Estructura general del laboratorio

El Laboratorio de Redes (véase la Figura 1) dispone de 26 puestos de usuario y varios equipos de comunicaciones e interconexión de redes. Estos sistemas se articulan en base a unos bloques de equipos, denominados ***islas***, que pueden funcionar de forma independiente entre sí (véase la Figura 3). En total se dispone en la actualidad de 4 de estas islas, numeradas de 1 a 4, cada una de las cuales tiene asociados 6 (ó 7) de los puestos de trabajo y un conjunto idéntico de equipos de comunicaciones, de forma que todas las islas son equivalentes en cuanto a arquitectura, equipos incluidos y funciones que es posible realizar.

Figura 1. Detalle del Laboratorio de Redes (Aula 3.7 de la ETSIIT)

Físicamente, los equipos de comunicaciones que componen cada una de las islas se encuentran ubicados en un único armario de 2 metros de altura (véase la Figura 2), estando las estaciones de trabajo asociadas distribuidas en el laboratorio como se muetra en la Figura 3.

Figura 2. Vista delantera y trasera del armario que contiene los equipos de interconexión

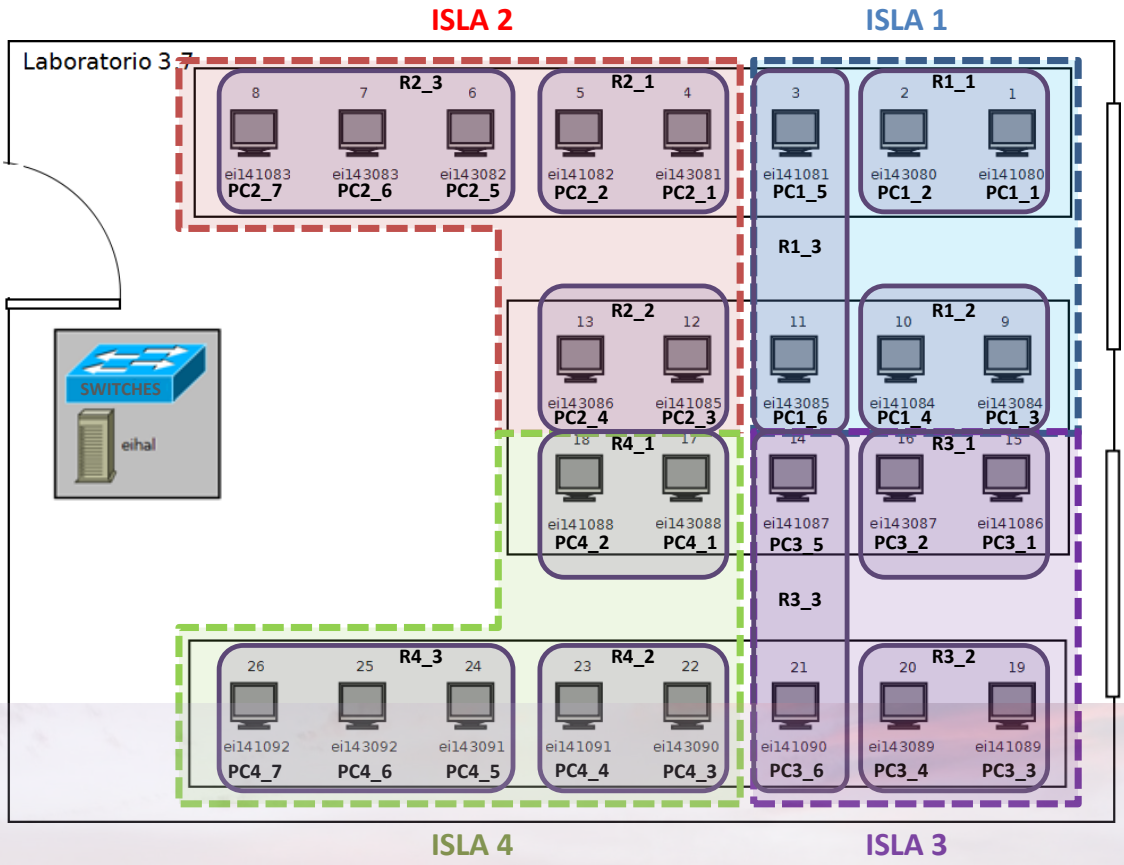


Figura 3. Disposición física y lógica (en islas) de los equipos en el Laboratorio de Redes

Los puestos de trabajo asociados a cada isla se encuentran etiquetados, en la CPU, de acuerdo a la nomenclatura ei*AAABBB*, donde *AAA* e *BBB* representan el tercer y cuarto byte, respectivamente, de sus direcciones IP en la red *interna* (del tipo 172.18.*AAA*.*BBB*/24). Además, también se etiquetan (véase la Figura 3) mediante la nomenclatura PC*X*_*Y*, siendo *X* el número de isla (de 1 a 4) e *Y* el número de equipo dentro de dicha isla (de 1 a 6 ó de 1 a 7, dependiendo de la isla).

Cada puesto de trabajo está asociado a una isla, de forma que dicho puesto de trabajo utilizará, para la realización de las prácticas, los equipos que contiene la isla en cuestión.

Es **importante**, en consecuencia, **elegir adecuadamente los equipos** a utilizar por cada grupo de alumnos

3. Equipamiento y cableado

Cada una de las islas contiene el equipamiento neceseario para realizar las diferentes prácticas. Los equipos que están contenidos en una isla *X* son los siguientes (véase la Figura 4y la Figura 7):

- 6 *routers* Mikrotik: implementan la *red de datos* (véase el siguiente apartado)

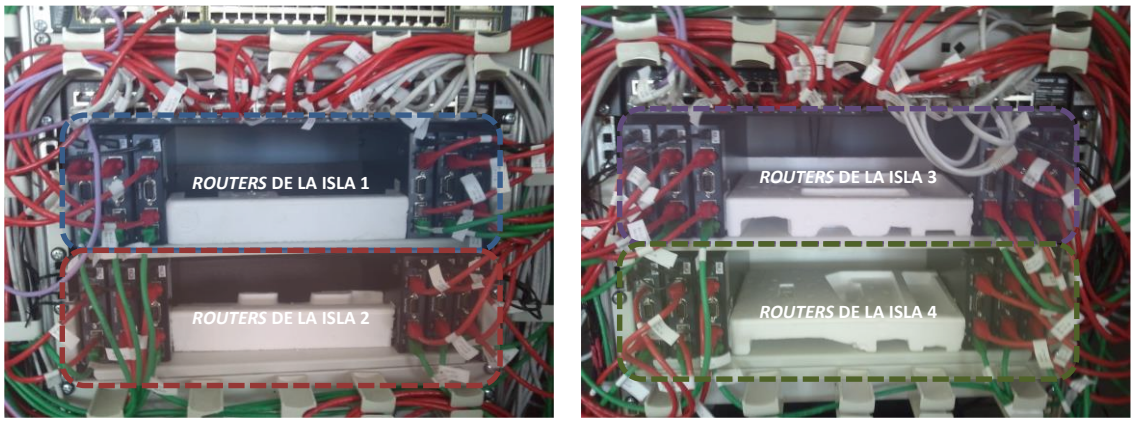


Figura 4. *Routers* Mikrotik de todas las islas

Además, estas islas se interconectan a través de 5 conmutadores:

- Conmutador **SW_BC**: compatible con el esquema anterior del Laboratorio 2.8, permite que los PCs se interconecten a través de la *red interna* (véase el siguiente apartado)
- Conmutador **SW_G**: permite conectar todos los equipos (PCs, *routers* Mikrotik y conmutadores) a través de la *red de gestión*, de forma que todos los equipos sean configurables desde cualquier PC del laboratorio
- Conmutador **SW_12**: interconecta los equipos (PCs y *routers* Mikrotik) correspondientes a las islas 1 y 2
- Conmutador **SW_34**: interconecta los equipos (PCs y *routers* Mikrotik) correspondientes a las islas 3 y 4
- Conmutador **SW_I**: para interconexión entre conmutadores

Todos los puestos de trabajo del área está equipados con 4 tarjetas de red:

- Una tarjeta de red externa. Esta tarjeta permite el acceso del equipo a la red de la ETSIIT y al exterior (Internet), siendo idéntica a la de los restantes laboratorios. Esta tarjeta estará deshabilitada cuando se arranque el equipo en alguna de las configuraciones específicas para la realización de las prácticas correspondientes a las asignaturas adscritas al Área de Ingeniería Telemática.
- Una tarjeta de *red interna* (interfaz de red *interna* en las imágenes Linux; véase el siguiente apartado para más detalles de esta red)
- Una tarjeta de *red de datos* (interfaz de red *datos* en las imágenes Linux; véase el siguiente apartado para más detalles de esta red)
- Una tarjeta de *red de gestión* (interfaz de red *gestión* en las imágenes Linux; véase el siguiente apartado para más detalles de esta red)

Figura 5. Detalle de los conmutadores y del cableado (parte trasera)

El laboratorio puede operar como cualquier otro laboratorio de la Escuela utilizando la tarjeta de red externa.

Sin embargo, la operación simultánea de la red externa y la red interna está deshabilitada. Por tanto, no habrá acceso al exterior del laboratorio y, por tanto, a las cuentas de usuario de la ETSIIT ni a Internet cuando se arranca en modo de red interna.

En general, deberá **trabajar en modo interno** para poder acceder a los equipos de su isla. Recuerde que, en este modo, no puede enviar ni recibir información del exterior, de forma que la **copia personal de información** deberá realizarla **a través de memorias USB**.

4. Estructura lógica del laboratorio

Cada una de las islas que componen el laboratorio es independiente del resto y está preparada para que los alumnos trabajen en grupos de 6 personas como máximo. Dos de las islas tienen un séptimo equipo

que podrá utilizarse en caso de que alguno de los otros equipos se encuentre averiado temporalmente.

Además, cada isla tiene tres redes lógicamente separadas, como se explica a continuación.

Por un lado, la *red interna* conecta todos los equipos en dos redes con direccionamiento privado, según la etiqueta en el la CPU. Estas dos redes son 172.18.141.0/24 y 172.18.143.0/24.

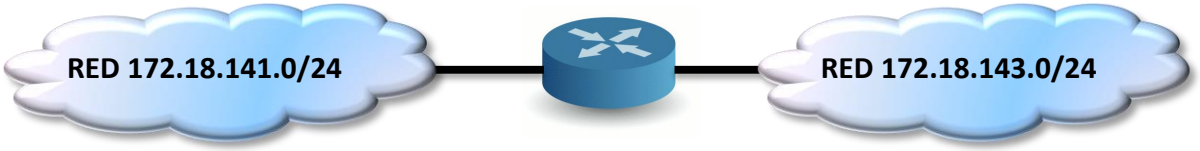


Figura 6. Esquema de la *red interna* del Laboratorio de Redes

Por otro lado, la *red de datos* permite conectar los PCs a los *routers* Mikrotik según el esquema, suponiendo la isla *X* (*X* = 1 … 4), mostrado en la Figura 7. Como se observa, dos PCs se conectan a un mismo *router* a través de la red 33.*X*.*Y*.0/24, siendo *X* el número de isla e *Y* el grupo de PCs dentro de la isla (*Y* = 1 … 3). Estos primeros *routers* (R*X*_1, R*X*_2 y R*X*_3) se conectan a otros dos *routers* (R*X*_4 y R*X*_5) a través de la red 172.16.*X*.0/24. Finalmente, estos dos *routers* se conectan a un *router* (R*X*_6), a través de la red 172.17.*X*.0/24. Este último *router* está conectado a sus homólogos de otras islas, a través de la red 220.10.10.0/24, permitiendo así la interconexión entre islas en caso de que fuese necesario para la práctica llevada a cabo.

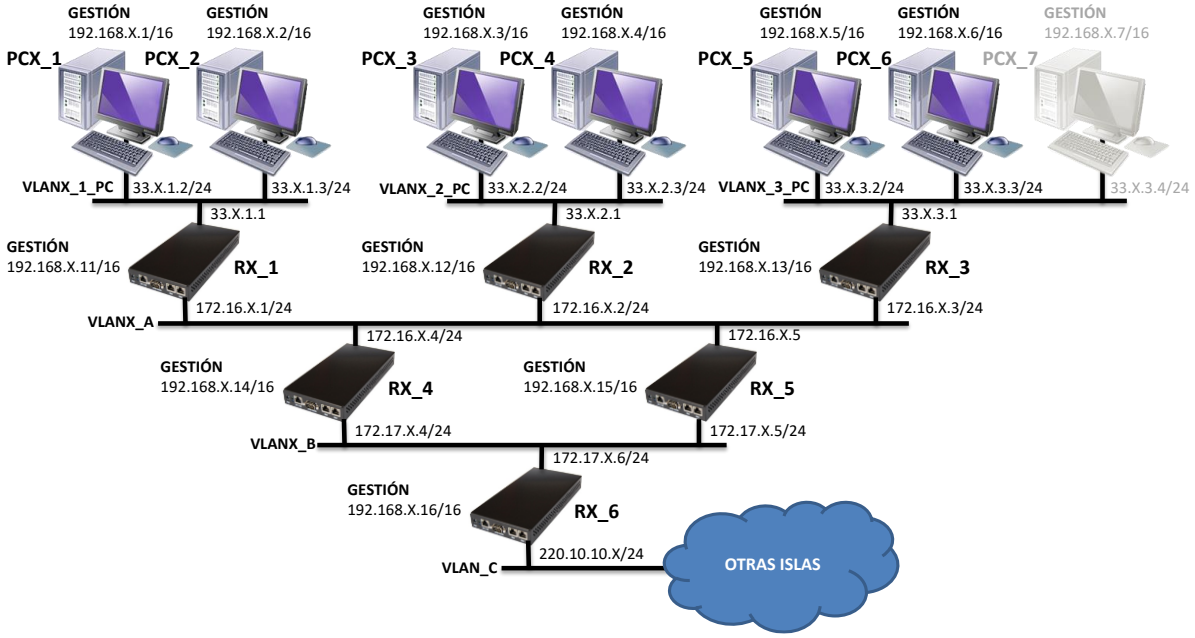


Figura 7. Esquema de una isla en Laboratorio de Redes

Por último, la *red de gestión* permite gestionar todos estos equipos (PCs, *routers* Mikrotik y conmutadores de interconexión) a través de la red 192.168.0.0/16. Las direcciones IP de PCs y *routers* en la red de gestión son las indicadas en la Figura 7.

1.2. Práctica 1

Práctica 1 – Configuración de Red I

1.1 Introducción

El objetivo de esta práctica aborda la configuración de rutas de encaminamiento tanto estáticas, de forma manual, como dinámicas, a través del uso de algoritmos de encaminamiento como por ejemplo RIP (Routing Information Protocol).

1.2 Información básica para la realización de la práctica

En esta sección se ofrece la información básica y las referencias necesarias para llevar a cabo las tareas que se proponen en la práctica.

1.2.1 Acceso al puesto de usuario y elección de sistema operativo

Para la realización de esta práctica, es necesario formar parejas. Después arrancar su puesto de usuario con la opción "Redes"→"Ubuntu 20.04".

Una vez que se haya identificado como "**administrador**"/"**finisterre**", puede pasar a modo *superusuario* mediante el siguiente comando, y utilizando la contraseña "**finisterre**"

```
# sudo su
```

1.2.2 Estructura de la Red Interna del Laboratorio

El Laboratorio de Redes (aula 3.7 de la ETSIIT) dispone de 26 puestos de usuario y varios equipos de comunicaciones e interconexión de redes, como puede observarse en la Fig. 1. El laboratorio está organizado en 4 conjuntos de equipos, denominados islas, que pueden funcionar de forma independiente entre sí. Para un mejor aprovechamiento y comprensión de las prácticas se recomienda encarecidamente leer el documento Práctica 0 - Introducción y descripción del laboratorio 3.7 disponible en el espacio de la asignatura en Prado.

Para configurar cualquier dispositivo de la isla, siempre se podrá acceder a ella utilizando su dirección en la red de *gestión*.

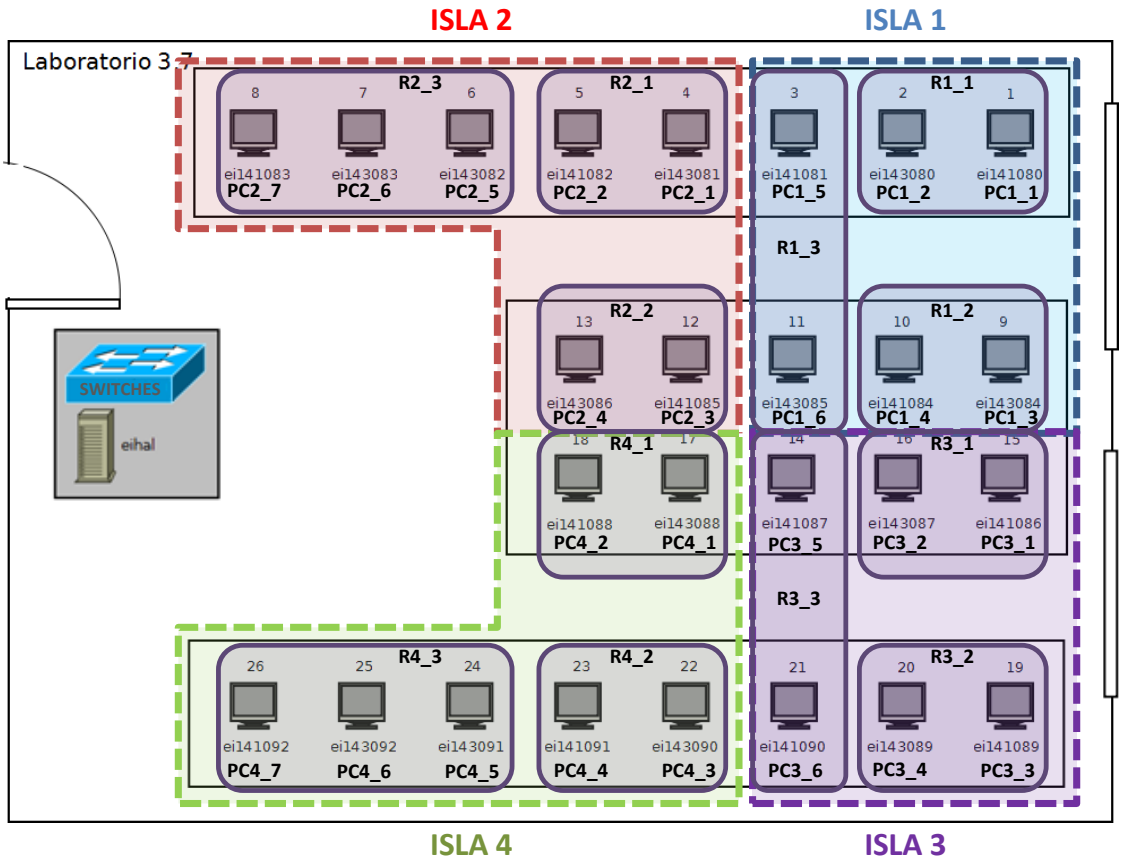


Figura 1: Disposición física y lógica (en islas) de los equipos en el Laboratorio de Redes.

1.2.3 Routers RouterBOARD

Cada isla del laboratorio 3.7 está equipada con varios *RouterBoards* de la marca *Mikrotik.* Los *RouterBoard* son pequeños *routers* integrados que ejecutan el sistema operativo *RouterOS*. Dichos *routers* están interconectados para conformar distintas topologías de red. La topología de red de cada isla viene dada por el esquema presentado en la Fig. 2.

Cada *router* y cada equipo dispone de varias interfaces, cada una conectada a una red diferente, por ejemplo, la topología mostrada en la Fig. 2 corresponde a la *red de datos,* que, a su vez, de divide en subredes diferentes. Hay además una *red de gestión* que está diseñada y configurada para poder acceder a todos los dispositivos de red y poder administrarlos. A la red de gestión se conectan los dispositivos a través de las interfaces con direcciones 192.168.X.Y, donde X corresponde al número de la isla e Y identifica al dispositivo dentro de esa red.

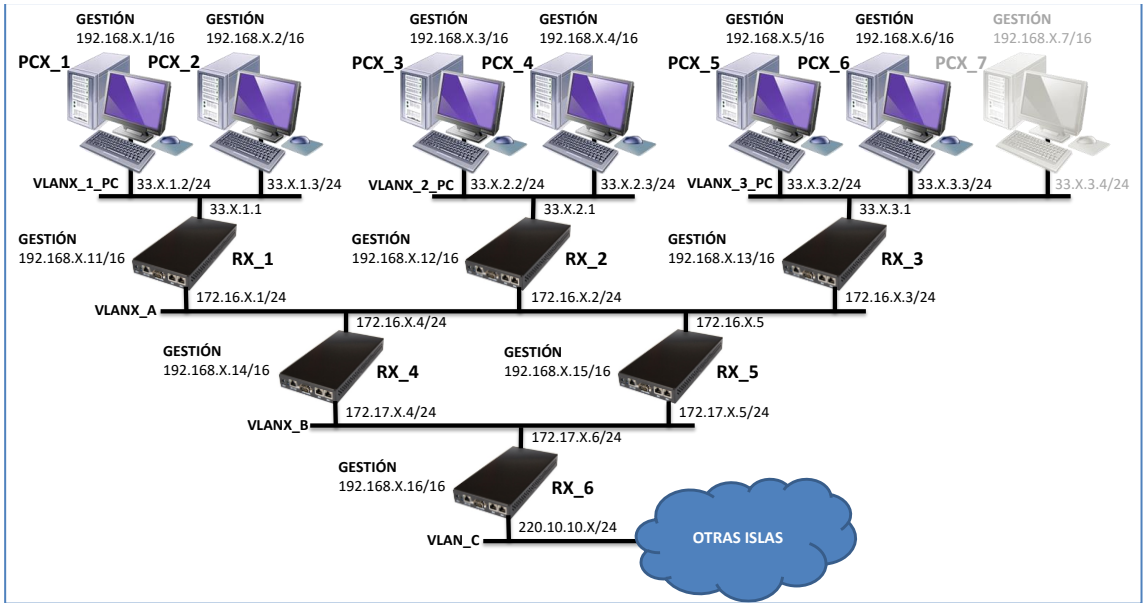


Figura 2: Esquema de una isla en Laboratorio de Redes.

1.2.3.1 Configuración de los routers MikroTik

El acceso a los *routers* para su configuración puede hacerse por diferentes vías. En primer lugar, se puede acceder a un *RouterBoard* a través de su **interfaz de línea de comandos (CLI)** utilizando una aplicación de acceso remoto (ej. TELNET o SSH) desde un puesto de usuario con el que haya conectividad con dicho *router*.

```
$ telnet <dirección IP de gestión del router>
```

El nombre de usuario es `admin` y no tiene contraseña (pulsar *ENTER*).

La interfaz CLI es similar en aspecto al terminal de LINUX de nuestro puesto. Pulsando la tecla TAB vemos los directorios y los comandos admitidos en el directorio actual. Escribiendo el nombre de un directorio y pulsando *ENTER* entramos en dicho directorio. Escribiendo el nombre de un comando y pulsando *ENTER* ejecutamos dicho comando y se nos pregunta por el resto de las opciones del comando. Para volver al directorio anterior hay que escribir dos puntos seguidos ("..") y pulsar *ENTER*.

Alternativamente el acceso a un router puede realizarse a través de la interfaz web del router (**WebFig**), con la ayuda de un navegador web y escribiendo la dirección IP de dicho router en el espacio reservado para escribir la URL.

Por último, el acceso al *router* puede hacerse mediante la aplicación **WinBox,** ejecutando desde cualquier PC el siguiente comando.

```
$ wine winbox_ubuntu1204.exe
```

A continuación, en la ventana principal que aparece, en el campo *"Connect To:"* introducir la IP de la red de gestión a la que pertenece el *router* al que se quiere acceder. Se recomienda utilizar WinBox para la gestión y configuración de los *routers* del laboratorio.

1.2.4 Encaminamiento en Redes TCP/IP

El encaminamiento en redes TCP/IP (tanto en dispositivos finales como intermedios) se realiza en base a tablas de enrutamiento donde se especifican, mediante diferentes entradas, la interfaz de red o el *router* que hay que utilizar (pasarela o *gateway*) para alcanzar un determinado destino. En la Fig. 3, se muestra un ejemplo de interconexión entre dos dispositivos finales, en este caso de la isla 1 del laboratorio, en dónde se observan, tanto el contenido de las tablas de encaminamiento necesarias, como los flujos de información IP (datagramas) tanto de petición (en color verde) como de respuesta (en color azul).

En ese mismo ejemplo, si queremos que PC_1 y PC_3 puedan alcanzarse, la tabla de enrutamiento de PC_1 debería incluir una entrada que viene a decir que, para alcanzar la red 33.1.2.0/24 (incluyendo la dirección IP 33.1.2.2 de PC_3), el datagrama IP debe reenviarse a la dirección IP 33.1.1.1, que es la dirección de la pasarela (*router*) o punto de salida de los paquetes que no van dirigidos a la subred a la que el PC_1 está conectado y que está en la misma red que dicho PC. Del mismo modo, la tabla de enrutamiento de PC_3 debe incluir una entrada diciendo que para alcanzar la red 33.1.1.0/24 (o la dirección IP 33.1.1.1), se debe reenviar el datagrama a la dirección IP 33.1.2.1 (dirección de la interfaz de la pasarela (*router*) que está en la misma red que PC_3).

En el ejemplo y para reducir el número de entradas en la tabla de encaminamiento tanto en PC_1 como en PC_3 se utiliza una entrada especial llamada *pasarela por defecto*. Esta se compone de la dirección de destino 0.0.0.0/0, dirección especial de red que agrupa todas las direcciones IP posibles en el espacio de direcciones IPv4, y una pasarela o *gateway* que es la dirección IP del *router* que está conectado a la subred correspondiente.

Normalmente, al asignar una dirección IP a una interfaz de red, suele añadirse automáticamente una entrada en la tabla de enrutamiento del dispositivo en cuestión. Este tipo de entradas se identifican, normalmente, por no tener una *gateway* definida ya que dicha interfaz está directamente conectada a la subred a la que pertenece el dispositivo. Esto permite que se pueda alcanzar cualquier dispositivo situado en la subred donde se ha asignado dicha dirección IP. En el ejemplo de la Fig. 3, en la tabla de encaminamiento del PC_1 dicha entrada se identifica con la dirección de destino 33.1.1.0/24 (subred a la que pertenece el PC_1) y la pasarela "-" indicando que dicha red se alcanza directamente, sin saltos, o lo que es lo mismo, PC_1 está directamente conectado a la subred 33.1.1.0/24.

En el ejemplo de la Fig. 3, PC_1 envía un paquete con dirección destino a PC_3. Como la dirección de destino de PC_3 (33.1.2.2) no pertenece a la subred de PC_1 (33.1.1.0/24) el paquete se reenvía por la pasarela por defecto de PC_1, es decir, la dirección IP de R1_1 (33.1.1.1) de la subred a la que ambos pertenecen. Una vez allí, el paquete ha de ser reenviado hacia su destino final, la subred 33.1.2.0/24. Para ello es necesario que el R1_1 sepa cómo hacerlo y para ello añade una nueva entrada en su tabla de encaminamiento tal que para llegar a la subred destino del paquete, necesariamente dicho paquete ha de reenviarse por el siguiente salto que es el R1_2. Concretamente la IP de la interfaz de red del R1_2 (172.16.1.2) de la subred 172.16.1.0/24 a la que también pertenece el R1_1. Una vez en el R1_2, el paquete llegará a su destino porque en dicho *router* existe una entrada en la tabla de encaminamiento, aquella entrada que indica que R1_2 se conecta directamente, también, a la subred 33.1.2.0/24

Todo lo anterior es aplicable para el flujo de ida de la información. De forma similar se considerará para el flujo de respuesta, pero teniendo en cuenta que, ahora, la dirección IP de

destino y origen se intercambian en los paquetes IP de respuesta con respecto a aquellos de petición.

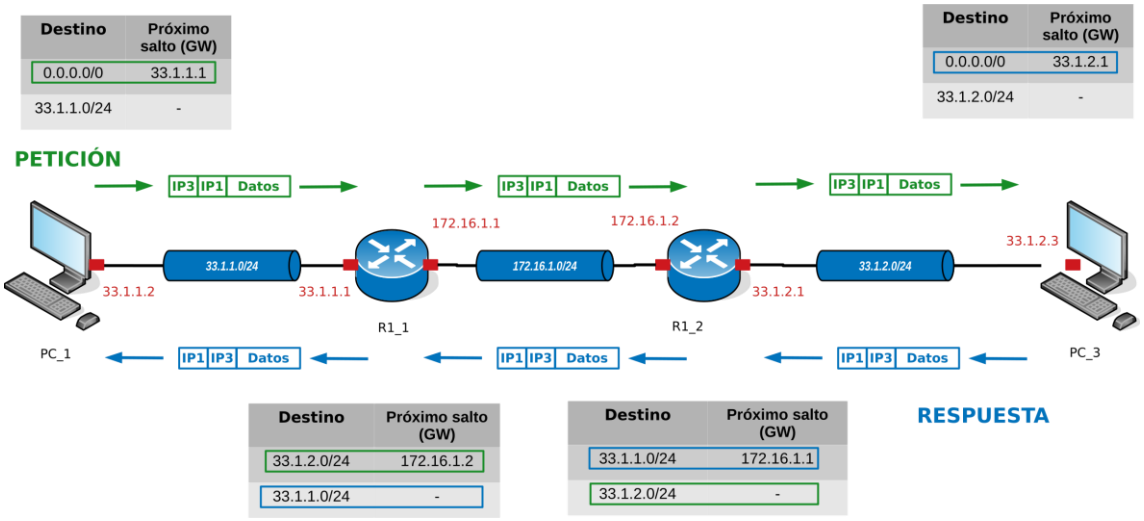


Figura 3: Flujos de conexión y tablas de encaminamiento para conectar PC_1 y PC_3 a través de los *routers* R1_1 y R1_2.

1.2.4.1 Configuración de tablas de enrutamiento en routers

La configuración de la tabla de enrutamiento en RouterOS mediante WinBox se puede llevar a cabo desde el menú *IP -> Routes* del *router*. En la Fig. 4 se muestra un ejemplo de configuración vía WinBox de una entrada en la tabla de encaminamiento del router R1_1 que indica que todos los paquetes que vayan dirigidos a la subred `33.1.2.0/24` se encaminen por el *router* (pasarela) correspondiente, en este caso, el R1_2, con IP 172.16.1.2

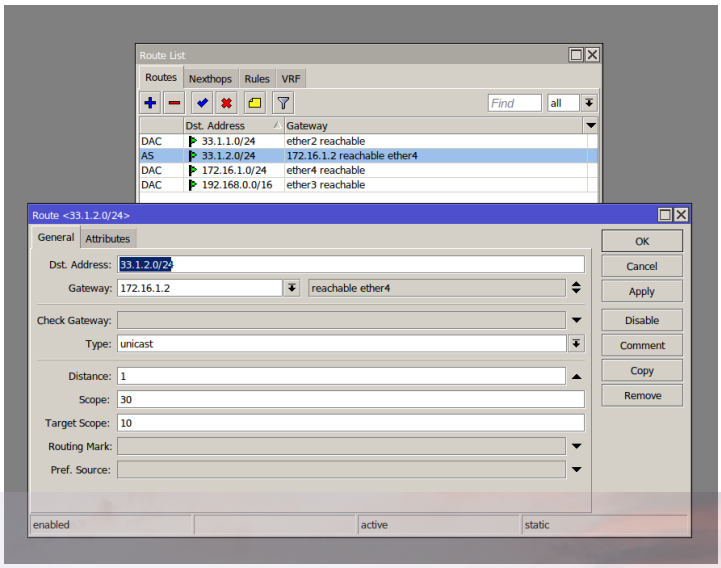


Figura 4: Cómo se incluye una nueva ruta en la tabla de encaminamiento de R1_1.

En `https://wiki.mikrotik.com/Manual:Simple_Static_Routing` puede consultar otro ejemplo de interconexión de redes diferente al de la Fig. 4 y las correspondientes entradas de la tabla de enrutamiento, con destino explícito y otras veces utilizando "gateways por defecto", utilizando la interfaz CLI, en lugar de WinBox.

1.2.4.2 Configuración de tablas de enrutamiento en puestos de usuario

Desde un terminal LINUX, la introducción de entradas en la tabla de enrutamiento puede realizarse de dos maneras:

a) Mediante el comando `route`. En este caso la configuración no se mantiene al reiniciar el sistema siendo este el método que utilizaremos en la práctica.

   Ej. 1: Para añadir una entrada en la tabla de encaminamiento que indique que para llegar a cualquier IP que pertenezca a la subred 192.168.128/25, hay que reenviar el datagrama a la pasarela 192.168.1.2, la sintaxis de *route* es:

```
route add -net 192.168.1.128 netmask 255.255.255.128 gw 192.168.1.2
```

   Ej. 2: Para añadir 192.168.1.200 como pasarela por defecto:

```
route add default gw 192.168.1.200
```

En cualquier caso, el contenido de la tabla de enrutamiento puede consultarse mediante el comando `route -n`.

1.2.5 Encaminamiento dinámico en Redes TCP/IP

Si bien la configuración de las tablas de encaminamiento de diferentes dispositivos puede llevarse a cabo de forma manual como se ha visto anteriormente, esta tarea se vuelve tediosa e incluso difícil de abordar si el número de dispositivos a configurar es elevado.

Es por este motivo por el que surgen los algoritmos y protocolos de encaminamiento dinámico que, de forma automática, establecen las correspondientes entradas en las tablas de encaminamiento para establecer rutas que permitan el intercambio de información siguiendo el camino más corto en función de una métrica o coste asociado. Estos protocolos de encaminamiento se llevan a cabo en dispositivos de nivel de red o *routers.*

Algunos ejemplos de dichos algoritmos son OSPF (Open Shortest Path First) y/o RIP (Routing Information Protocol). En esta práctica nos centraremos en RIP.

1.2.5.1 RIP: Routing Information Protocol

RIP (RFC1058) es un protocolo de encaminamiento basado en saltos o vector distancia como métrica para computar el coste del camino más corto. Es decir, el camino más corto entre un origen y un destino es aquel que menos saltos conlleva. El número total de saltos de una ruta se corresponde con el número total de *routers* (*hops)* que es necesario atravesar para llegar al destino.

En el ejemplo de la Fig. 3, para llegar desde el PC_1 al PC_3 es necesario dar dos saltos que se corresponden con los dos *routers* por donde tendría que pasar un paquete IP.

1.2.5.2 Configuración RIP en MikroTik

Para configurar un *router* de manera que utilice RIP y añadir entradas a su tabla de encaminamiento de forma dinámica, hemos de configurar a qué redes se conecta directamente dicho *router.* Para ello, con Winbox accederemos al menú *Routing -> RIP -> Networks.* y procederemos tal y como se indica en la Fig. 5

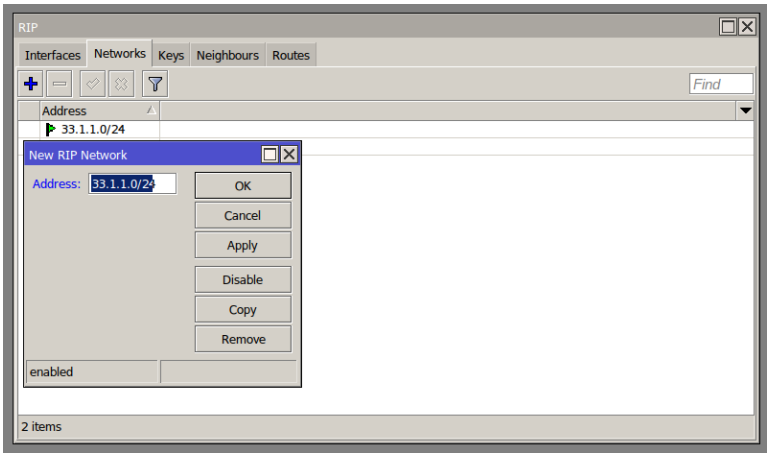


Figura 5: Configuración RIP de una de las subredes a la que se conecta directamente el R1_1.

En el siguiente enlace se puede consultar más información acerca de los diferentes comandos e información sobre la configuración de RIP en MikroTik mediante acceso CLI: `https://wiki.mikrotik.com/Manual:Routing/RIP`

1.3 Realización práctica

1.3.1 Encaminamiento estático

1) Compruebe el número de isla y puesto en el que se encuentra e identifique a sus compañeros en la isla. Compruebe las direcciones IP que tienen asignadas las diferentes interfaces de red de su puesto mediante el comando `ifconfig`, ¿cómo se llaman dichas interfaces? ¿Qué direcciones de red hay definidas? ¿Qué direcciones tiene el *router* al que se conecta el equipo que está usando?

2) Introduzca las entradas de encaminamiento necesarias para comunicar todos los puestos de usuario primero de su isla y luego de todo el laboratorio por la red de datos. Compruebe la configuración con la utilidad `ping -R` y anote los resultados.

CHECKPOINT: Avise al profesor cuando termine esta tarea.

1.3.2 Encaminamiento dinámico: RIP

Elimine/deshabilite todas las entradas de las tablas de encaminamiento en los *routers* derivadas de la realización de la anterior sección: encaminamiento estático.

3) Configure RIP en todos y cada uno de los *routers.*

Compruebe la tabla de encaminamiento tanto en el menú correspondiente en RIP como en el menú *IP->Routes.* ¿Tiene sentido lo que observa? Corrobórelo mediante la comprobación de la conectividad y saltos entre los PC de su isla con la utilidades `ping` y `ping -R` y anote los resultados.

CHECKPOINT: Avise al profesor cuando termine esta tarea.

4) Deshabilite la interfaz de uno de los routers RX_4 o RX_5 que conecta con la red 172.17.X.0/24.

Compruebe si se han producido modificaciones en las tablas de encaminamiento de los *routers* RX_1, RX_2 y RX_3 ¿Qué cambios se han producido? Apóyese de las herramientas `ping` y `ping -R` y anote los resultados.

CHECKPOINT: Avise al profesor cuando termine esta tarea.

1.4 Bibliografía

[1] Manual de MikroTik. http://wiki.mikrotik.com/wiki/Manual:TOC
[2] RFC 1058 – RIP. https://tools.ietf.org/html/rfc1058

Práctica 1 – Configuración de Red II

1.1 Introducción

Un cortafuegos (*firewall*) en una red de computadores permite establecer una pasarela o barrera entre dos subredes tal que el administrador pueda filtrar y/o permitir el tráfico cursado de una forma controlada. Además, ofrece otras funciones como por ejemplo la monitorización o la contabilidad (*accounting*) del tráfico. Los cortafuegos nos permiten tener, por tanto, un control de los servicios a los que se accede y de las comunicaciones que se llevan a cabo en una red.

En la Fig. 1, se observa un ejemplo típico de la ubicación de varios cortafuegos dentro de una organización. En dicha figura se observa un *router* de acceso, que conecta varios departamentos (típicamente con direcciones privadas) protegidos por sus correspondientes cortafuegos, además, una DMZ (Demilitarized Zone) en dónde, usualmente, se exponen públicamente diferentes servicios de red.

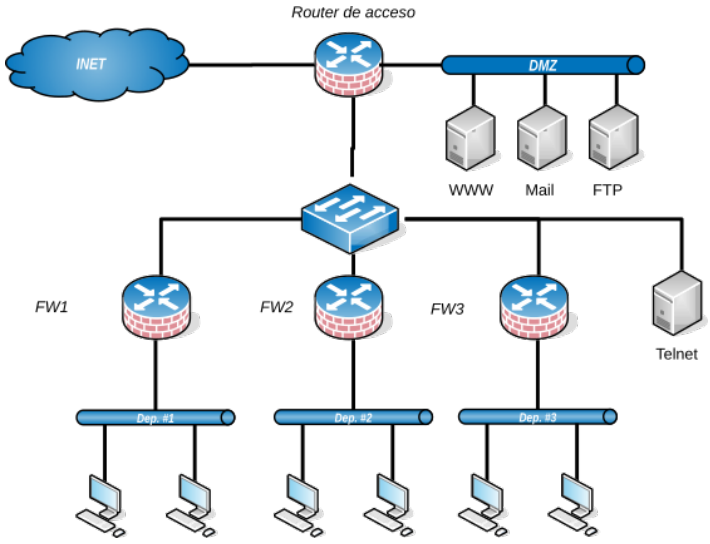


Figura 1: Ubicación típica de cortafuegos dentro de una ogranización.

Existen dos tipos principales de cortafuegos: de filtrado (*packet filters*) o de aplicación (*proxy*). Los primeros fundamentalmente ofrecen de una serie de filtros, definidos mediante un conjunto de reglas, que permiten controlar el acceso a determinados servicios, *hosts*, etc. Dichos filtros se pueden establecer teniendo en cuenta: la IP origen o destino, el campo protocolo del datagrama, el puerto origen o destino del segmento, la interfaz (dirección MAC), u otros campos de cualquiera de los protocolos implicados en capa de transporte e inferiores.

Los cortafuegos tipo *proxy* operan a nivel de aplicación y, a diferencia de los cortafuegos de filtrado, actúan como intermediarios entre los clientes (internos) y el servidor (externo). Esto es, de cara al exterior todas las peticiones provienen del *proxy* de manera que los clientes quedan ocultos.

En esta práctica configuraremos un cortafuegos de filtrado.

1.1.1 Reglas

La definición del comportamiento de un *firewall* de filtrado se hace mediante reglas. Estas, como su propio nombre indica, definen la política de acceso y control sobre el tráfico cursado, mediante unos criterios para seleccionar o no los paquetes. Además, cada regla define la acción a realizar sobre ese tráfico seleccionado. Las reglas de filtrado tienen, por tanto, dos partes:

1. El **criterio de selección** de los paquetes a los que aplicar la regla. Por ejemplo: el puerto de destino debe ser el 80.
2. La **acción** a llevar a cabo sobre los paquetes seleccionados por el criterio de selección. Por ejemplo: descartar (*drop*) el reenvío de los paquetes que cumplan con el criterio de selección.

Los criterios básicos de selección de paquetes se suelen basar en campos de los paquetes tales como: la dirección IP de destino u origen, el puerto destino u origen, el tipo de protocolo de transporte (UDP o TCP), etc. Existen otros atributos tales como el estado de las conexiones TCP, o el tipo de segmento TCP (Syn, Fin, Ack, etc.).

Tras definir el criterio de selección se ha de indicar la acción a realizar. Existen varias acciones predefinidas, siendo las más habituales:

- **accept**: acepta los paquetes que cumplen el criterio de selección, y sigue procesándolos normalmente.
- **drop**: descarta el paquete seleccionado.
- **reject**: además de descartar el paquete seleccionado, el *router* envía al origen un mensaje ICMP del tipo que se especifique.

1.1.2 Cadenas

Las reglas se asocian según un criterio de selección previo que depende del tipo de paquetes a las que se aplican, formando lo que se conoce como cadenas (*chains*). Así en el *firewall* de los *routers* Mikrotik, tal y como muestra la Fig. 2, las cadenas predefinidas son:

- **INPUT**: incluye las reglas que se aplican a paquetes que tienen como dirección destino alguna de las IP del *router*. Es decir, aquellos paquetes que van dirigidos al propio *router*.
- **OUTPUT**: incluye las reglas que se aplican a paquetes generados por el propio *router*. Es decir, aquellos paquetes que tienen como IP origen alguna de las del *router* (de sus interfaces).
- **FORWARD**: incluye las reglas que se aplican a paquetes que reenvía el *router*, es decir, los paquetes que ni se han generado ni van dirigidos al propio dispositivo. Por lo tanto, dichos paquetes no tienen ni IP origen ni destino que se correspondan con alguna de las del *router*. Sería tráfico que "lo atraviesa".

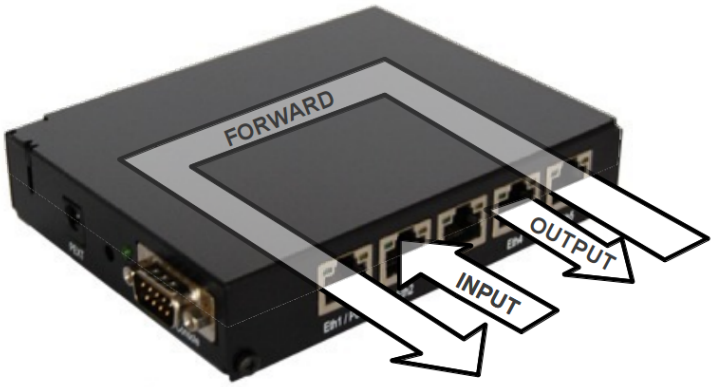


Figura 2: Cadenas de reglas de filtrado básicas.

1.2 Información básica para la realización de la práctica

En esta sección se ofrece información básica y las referencias necesarias para llevar a cabo las tareas que se proponen en la práctica.

1.2.1 Acceso al puesto de usuario y elección de sistema operativo

Para la realización de esta práctica, es necesario formar parejas. Después arrancar su puesto de usuario con la opción "Redes"→"Ubuntu 20.04".

Una vez que se haya identificado como "**administrador**"/"**finisterre**", puede pasar a modo *superusuario* mediante el siguiente comando, y utilizando la contraseña "**finisterre**"

```
# sudo su
```

1.2.2 Escenario de trabajo y dispositivos implicados

En la Fig. 3 se observa el escenario de trabajo y los dispositivos implicados para la realización de la sesión de prácticas. El direccionamiento IP de los elementos que aparecen en la figura, se corresponde con aquellas direcciones que se encontrarían en la isla 1. Como cada pareja configurará el *router* al que tiene acceso directo desde su subred, será necesario dialogar con las demás parejas de la isla para realizar y probar las tareas que se exponen al final del presente guion.

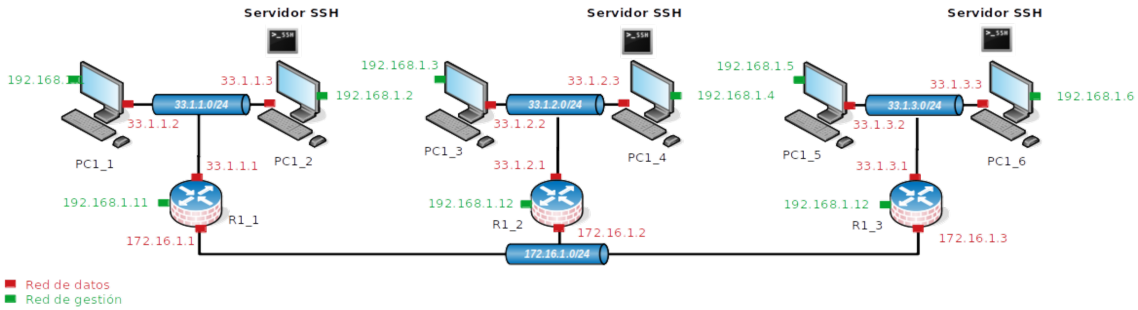


Figura 3: Escenario de trabajo y dispositivos implicados. Ejemplo para los dispositivos de la isla 1.

1.2.3 Configuración de reglas de filtrado

Para configurar el cortafuegos, acceder al menú *IP->Firewall* en WinBox. Para añadir una nueva regla, desde la pestaña de "Filter Rules", añadir las reglas requeridas.

El orden en el que aparezcan las reglas de filtrado es muy importante. Por ejemplo, si se añade al principio una regla genérica con acción *drop* para descartar todo el tráfico, las siguientes reglas de la cadena no tendrán efecto, por tanto, esta debería ir en última

ATENCIÓN: no definan reglas **drop** sobre las cadenas INPUT u OUTPUT. Esto puede hacer que el *router* quede inaccesible y no se pueda administrar.

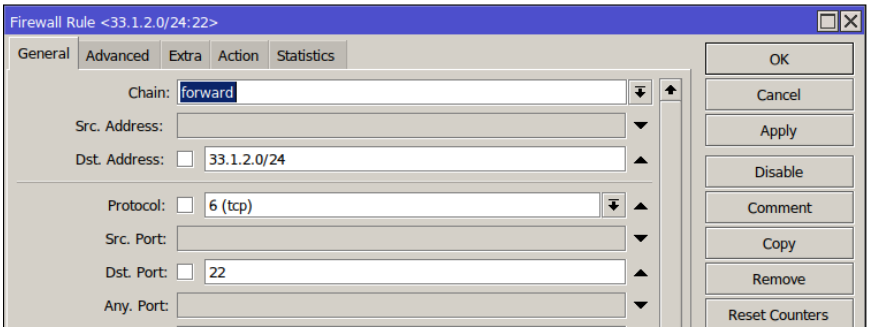


Figura 4: Configuración de una regla de filtrado desde WinBox.

Para configurar una nueva regla, seleccionar los campos y los valores que deben cumplir los paquetes en la pestaña "General". En la Fig. 4 se muestra un ejemplo de creación de una regla que permite el reenvío (*forward*) de toda aquella información con destino a la subred 33.1.2.0/24 y cuyas peticiones vayan dirigidas al puerto 22 de las máquinas destino.

La acción a realizar con esos paquetes se puede configurar en la pestaña "Action". En la Fig. 5 se selecciona la acción *accept* que significa que todos aquellos paquetes cumplan con el criterio de selección de la regla se dejarán pasar.

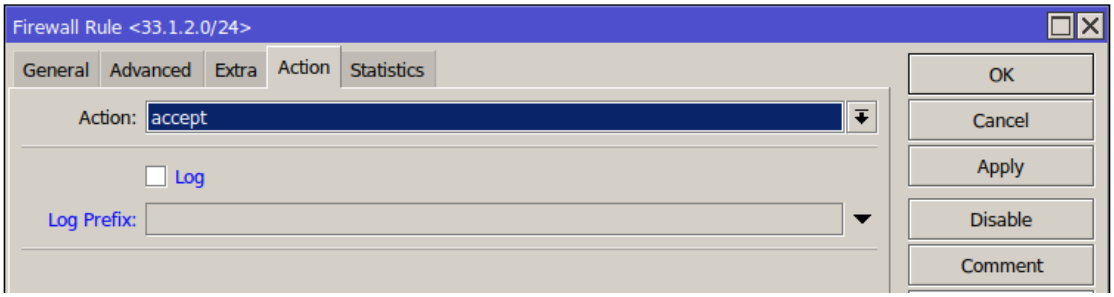


Figura 5: Configuración de la acción de una regla de filtrado.

Existen varias formas de ver si una regla se está evaluando, es decir, si la información de los paquetes que pasan por el *firewall* coincide con el criterio de selección de dicha regla. Esto nos ayudará a comprobar si la regla está bien construida. Para ello, en la ventana principal en dónde se exponen todas las reglas creadas, notar si los campos *Bytes* y *Packets* van cambiando. Esto será indicativo de que la regla se está evaluando (ver Fig. 6)

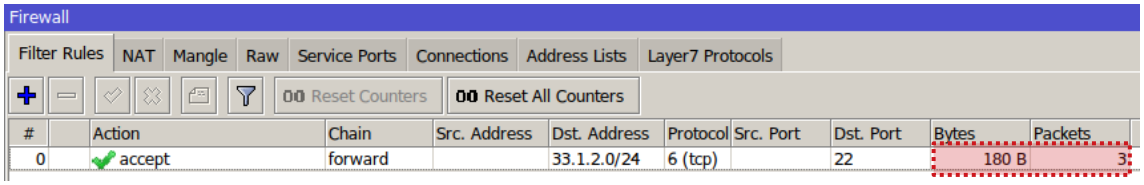


Figura 6: ¿Se está evaluando una regla? Los campos *Bytes* y *Packets* varían cuándo dicha regla se evalúa.

Otra opción muy útil y recomendable para ver cómo está funcionando el *firewall* y más específicamente si se están evaluando sus reglas es activar en el *router* desde la utilidad *System -> Logging* una nueva regla *Log Rule*. Para nuestro caso, aquella que hace *logging* de los eventos relacionados con el *firewall* y sus reglas. En la Fig. 7 se muestra un ejemplo de configuración de dicha regla de log y en la Fig. 8 se muestra cómo aparecen entradas en la ventana de *Log* (a esta ventana se acede directamente desde el menú del *router*) correspondientes, en este caso, a la evaluación de la regla de filtrado que se creó anteriormente y que se observa en la Fig. 6

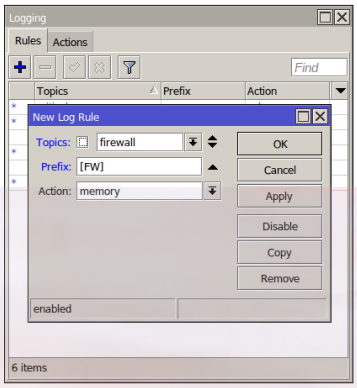


Figura 7: Configuración de la utilidad *Logging* para chequear si una regla se está evaluando.

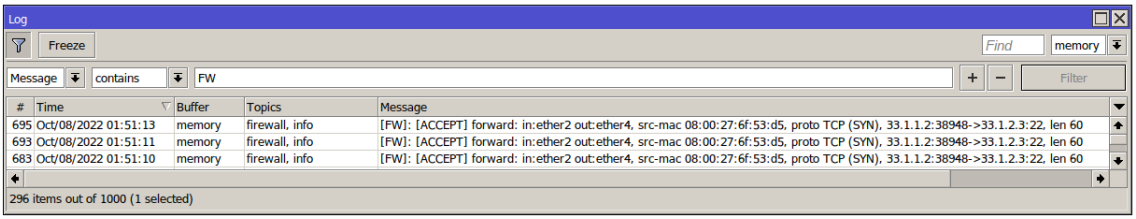


Figura 8: Entradas de *log* filtradas para el prefijo [FW] de la nueva regla de *log* creada anteriormente en la Fig. 7. Se observa como aparece una entrada por cada uno de los paquetes que se han evaluado correctamente por la regla del *firewall* creada en la Fig. 6

1.3 Realización práctica

Es necesario configurar las tablas de encaminamiento tanto en los *routers* como en los PC para que estos últimos tengan conectividad entre ellos. Se deberá comprobar que hay conectividad antes de configurar las reglas del *firewall*.

1) Configure su *router*, el que está directamente conectado a su subred, para que NO reenvíe ningún tipo de tráfico (acción "*drop*"). Habitualmente, al configurar un cortafuegos, inicialmente se deniega el reenvío de todo el tráfico, y luego se añaden reglas explícitas para el tráfico que sí se desea dejar pasar. Compruebe que ahora no es posible enviar o recibir tráfico entre los PC ubicados en diferentes subredes.

2) A continuación, configure el cortafuegos de su *router* para que permita a otros ordenadores conectarse únicamente al servidor de SSH instalado en uno de los PC de su red (ver Fig. 3).

Tenga en cuenta que el protocolo SSH transporta sus mensajes sobre TCP y utiliza el puerto 22.

Es necesario levantar el servicio SSH en el PC servidor. Para ello ejecute el siguiente comando.

```
# sudo systemctl start ssh.service
```

Para conectarse remotamente a un PC con SSH, utilizar el siguiente comando, donde `<usuario_PC_remoto>` es el usuario de la máquina remota con IP `<IP_PC_remoto>`

```
# ssh <usuario_PC_remoto>@<IP_PC_remoto>
```

CHECKPOINT: Avise al profesor cuando termine esta tarea.

3) Configure el mismo *router* para que permita hacer ping de un ordenador a otro, pero no en sentido contrario (ver Fig. 3).

 Tenga en cuenta que la herramienta `ping` envía mensajes ICMP de tipo `echo request` y recibe mensajes ICMP de tipo `echo reply`.

 CHECKPOINT: Avise al profesor cuando termine esta tarea.

1.4 Bibliografía

[1] Manual de MikroTik. http://wiki.mikrotik.com/wiki/Manual:TOC

1.3. Práctica 2

Práctica 2 : Servicios básicos de red I

1 Introducción

El objetivo de esta práctica es familiarizar al alumno con algunos de los servicios básicos de red que se utilizan habitualmente en las redes corporativas. Un ejemplo típico de red corporativa sencilla es el representado en la siguiente figura.

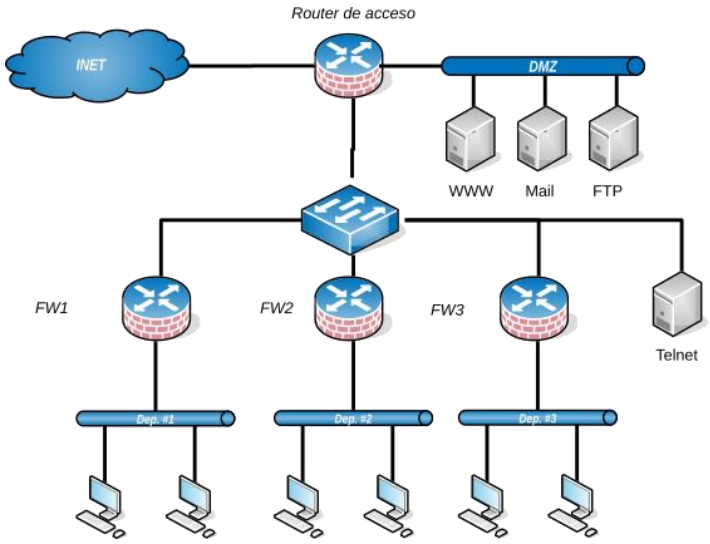


Figura 1: Ejemplo de red corporativa y sus elementos principales.

Concretamente, el servicio que se configurarán en esta sesión de prácticas es el siguiente:

- **Configuración dinámica de equipos** (**DHCP**, *dynamic host configuration protocol*): este servicio proporciona autoconfiguración de los equipos conectados a nuestra red. Normalmente, por sencillez, los servidores tendrán una configuración estática (dirección IP, pasarela, DNS), mientras que los equipos *hosts* utilizarán este protocolo para configurarse de forma automática.

  En el ejemplo de la figura, la ubicación típica para este servicio sería en los *routers* FW1, FW2 y FW3 si cada departamento gestiona sus propias direcciones IP, o en el *Router de acceso* si la gestión es centralizada. En dicho caso, los *routers* FW1, FW2 y FW3 actuarían de servidores *DHCP relay*, reenviando las peticiones DHCP al servidor DHCP centralizado.

Además de implementar y configurar este servicio, los alumnos harán uso de la herramienta *Wireshark* para monitorizar los paquetes enviados y comprobar el correcto funcionamiento del escenario desplegado.

2 Información básica para la realización de la práctica

En esta sección se ofrece información básica y las referencias necesarias para llevar a cabo las tareas que se proponen en la práctica.

2.1 Acceso al puesto de usuario y elección del sistema operativo

Para la realización de esta práctica es necesario formar parejas. Después arrancar su puesto de usuario con la opción "Redes"→"Ubuntu 20.04".

Una vez que se haya identificado como "**administrador**"/"**finisterre**", puede pasar a modo *superusuario* mediante el siguiente comando, y utilizando la contraseña "**finisterre**"

```
# sudo su
```

2.1 Asignación dinámica de direcciones IP: DHCP

El objetivo de la práctica es introducir al alumno en la asignación dinámica de direcciones IP mediante el protocolo DHCP (RFC 2131). Se estudiará como configurar un servidor DHCP alojado, usualmente, en un *router* así como los mensajes más relevantes de dicho protocolo que se intercambiarán entre clientes DHCP (PC) y servidor DHCP (*router*).

2.1.1 Escenario de trabajo y dispositivos implicados

En la Figura 2 se observa el escenario de trabajo y los dispositivos implicados para la realización de la sesión de prácticas. El direccionamiento IP de los elementos que aparecen en la figura se corresponde con aquellas direcciones que se encontrarían en la isla 1. Cada pareja tiene que identificar dichos elementos y direcciones en función de donde se ubiquen en el laboratorio. Los elementos sombreados en la figura se exponen ahí para una mejor ubicación del alumno/a dentro de la isla.

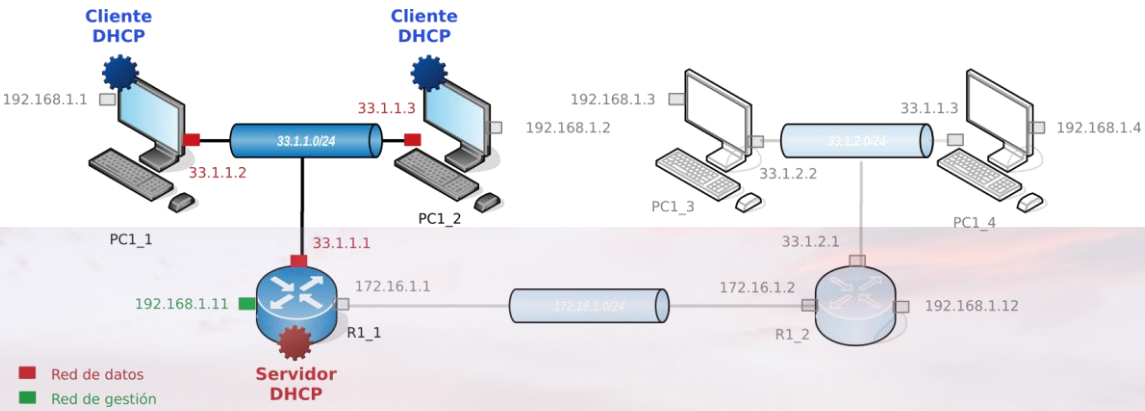


Figura 2: Escenario de trabajo y dispositivos implicados. Ejemplo para los dispositivos de la isla 1.

De acuerdo a la Figura 2, los PC finales de cada subred actuarán de clientes DHCP solicitando una IP al servidor DHCP alojado en el *router*. Este último asignará una dirección IP dentro del rango de subred al que pertenecen los PC. De forma general, dentro del rango 33.X.Y.0/24, siendo X el número de isla e Y dependiente de la subred de los PC. En el caso de la figura, para la isla 1, el servidor se encargará de asignar una dirección IP válida dentro del rango de direcciones de la subred 33.1.1.0/24.

2.1.2 Configuración de los clientes DHCP en los PC

Antes de configurar el servidor, hemos de configurar la interfaz de *datos* para que, en lugar de asignarle una IP fija de forma manual esta se le asigne automáticamente desde el servidor. Para ello utilizaremos el gestor de asignación de interfaces *netplan*. En concreto se ha de editar el fichero situado en `/etc/netplan/01-network-manager-all.yaml` para que sea la interfaz de *datos* la que se configure con DHCP.

Para guardar los cambios de configuración de interfaces con *netplan,* ejecutar:

```
$ sudo netplan apply
```

2.1.3 Configuración del servidor DHCP en *routers* MikroTik

Para llevar a cabo la configuración del servidor DHCP en los *routers* MikroTik será necesario acceder al dispositivo en cuestión a través de su IP de gestión mediante la aplicación Winbox. Acto seguido se ha de ejecutar el asistente de configuración para el servidor DHCP desde el menú *IP -> DHCP* Server pulsando sobre el botón *"DHCP Setup"*. Seguir las indicaciones y configuración de dicho asistente.

Para acceder al *router* mediante Winbox, ejecute el siguiente comando:

```
$ wine winbox_ubuntu1204.exe
```

2.1.4 Introducción al análisis de tráfico de red con Wireshark

Los analizadores de tráfico o *sniffers* son herramientas que permiten inspeccionar el flujo de tráfico que circula por alguna de las interfaces de red de un dispositivo. Existen varias aplicaciones con las que podemos realizar esta acción, siendo Wireshark la más utilizada y extendida.

Para ejecutar Wireshark, lance el siguiente comando:

```
$ sudo wireshark
```

Una vez ejecutado Wireshark, seleccione la interfaz por la que se quiere escuchar tráfico, en nuestro caso *datos* y comience la escucha haciendo doble "click" sobre la interfaz o desde el menú *Capture -> Start.* Es muy recomendable utilizar la funcionalidad de filtrado de tráfico según protocolo tal y como se observa en la Figura 3.

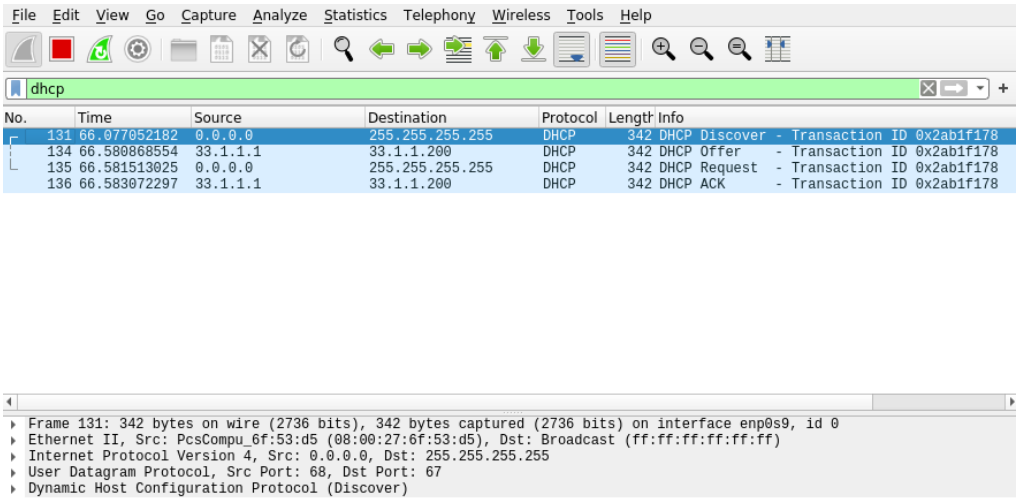


Figura 3: Ejemplo de captura de tráfico y mensajes DHCP con Wireshark.

3 Realización práctica

1) Identifique los PC, *router* y la subred a la que pertenecen. A continuación, configure las interfaces de *datos* de los PC para que, en lugar de asignarle una IP fija de forma manual tal y como actualmente se hace, se utilice el protocolo DHCP. Recuerde aplicar los cambios con *netplan.*

2) Configure adecuadamente el servidor DHCP a través del asistente para que asigne direcciones IP dentro del rango 33.X.Y.200-33.X.Y.210 a los PC de su subred. Además, establezca la dirección de DNS, por ejemplo, 33.X.Y.1, así como haga que el tiempo de asignación de IP (*lease time*) sea de 10 minutos.

3) Ejecute Wireshark para que escuche todo el tráfico DHCP que circula por la interfaz de *datos* de uno de los PCs. A continuación, solicite una IP al servidor DHCP mediante el siguiente comando.

Para que una interfaz solicite una IP al servidor DHCP, ejecutar el siguiente comando:

```
$ sudo dhclient -v datos
```

Analice tanto el contenido de las IP asignadas (*leases*) en el servidor DHCP (pestaña *Leases*) como en el PC (fichero `/var/lib/dhcp/dhclient.leases`) así como los mensajes

que aparecen tanto en la consola del PC cliente como en Wireshark. Interprete los resultados.


CHECKPOINT: Avise al profesor cuando termine esta tarea.

4) Configuración de esquemas de retransmisión (*relay*) de peticiones DHCP. Para ello apóyese en la documentación y ejemplos de las referencias.


CHECKPOINT: Avise al profesor cuando termine esta tarea.

4 Referencias

[1] RFC 2131: https://datatracker.ietf.org/doc/html/rfc2131
[2] Configuración de DHCP con MikroTik: https://help.mikrotik.com/docs/display/ROS/DHCP#DHCP-DHCPServer

Práctica 2 – Servicios básicos de red II: NAT

1. Introducción

Network Address Translation (NAT), o traducción de dirección de red en castellano, es un servicio que permite usar en una red un conjunto de direcciones IP para las comunicaciones internas y otro conjunto de direcciones IP distinto para las comunicaciones externas. Para ello, en toda red que use NAT, debe haber una pasarela (*gateway*) o enrutador (*router*) NAT encargada de reescribir ("traducir") en la cabecera IP la dirección IP origen en los paquetes salientes y la dirección IP destino en los paquetes entrantes. La pasarela NAT utiliza una tabla de traducción NAT para mapear las direcciones IP internas en direcciones IP externas.

Este mapeo puede ser estático o dinámico. En el mapeo **estático** se define explícitamente la correspondencia uno a uno entre el conjunto (*pool* en inglés) de direcciones IP internas y el conjunto de direcciones IP externas, mientras que en el mapeo **dinámico** se define algún criterio para establecer esta correspondencia según las condiciones de la red. Así, por ejemplo, para el último caso, se podría asignar una de las direcciones IP externas disponibles (que no estén actualmente en uso por alguno de los equipos de la red) de forma aleatoria a un dispositivo de la red que inicie un periodo de actividad (genere tráfico). Del mismo modo, se podrían liberar direcciones IP de aquellos dispositivos que tengan una de ellas asignada actualmente y que hayan estado inactivos durante un cierto periodo de tiempo.

Una de las principales motivaciones de usar NAT es paliar el problema de la falta de direcciones IPv4. Por ejemplo, un conjunto de direcciones IP públicas puede compartirse con una relación 1:N (hay más interfaces IP en la red privada que direcciones públicas disponibles) usando un criterio de asignación dinámica de IP públicas como en el ejemplo descrito anteriormente. Sin embargo, en estos escenarios una misma dirección IP pública sólo puede ser usada por uno de los equipos de la red privada en un instante dado. Este problema se solventa con *Network Address Port Translation* (NAPT). NAPT va un paso más allá de NAT para soportar la traducción de identificadores de transporte tales como los puertos TCP/UDP y los identificadores de consulta ICMP. Esto le permite a un conjunto de estaciones finales compartir una misma dirección IP externa simultáneamente. Para tal fin, los identificadores de transporte de un conjunto de estaciones de la red NAT (puertos) se multiplexan en los identificadores de transporte (puertos) de una única dirección IP externa (véase la Figura 1).

En el caso de NAPT, la tabla de traducción o mapeo NAT establece una correspondencia uno a uno entre los pares *<IP interna, puerto interno>* y los pares *<IP externa, puerto externo>* (véase la Figura 1).

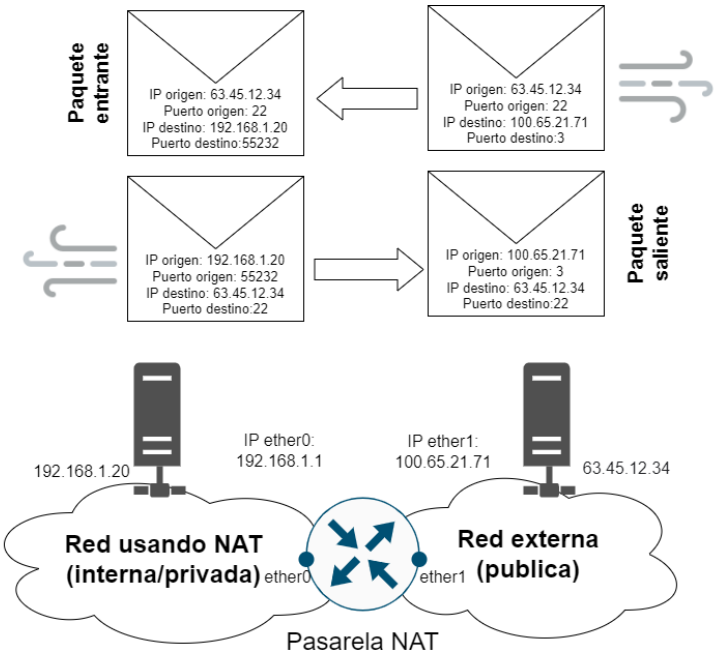


Tabla de mapeo NAT

Dirección IP local	Puerto local	Dirección IP externa	Puerto externo
192.168.1.10	55232	100.65.21.71	1
192.168.1.20	62301	100.65.21.71	2
192.168.1.20	55232	100.65.21.71	3
192.168.1.20	7888	100.65.21.71	4

Figura 1: Operación de la funcionalidad NAT.

Considerando el origen de la comunicación, podemos diferenciar dos tipos de NAT:

***Source* NAT (*srcnat*)**: Este tipo de NAT aplica a las comunicaciones originadas dentro de la propia red NAT. El *router* NAT reemplaza la dirección IP origen de los paquetes salientes originados en la red NAT. Posteriormente realizará la operación inversa para los paquetes entrantes.

***Destination* NAT (*dstnat*)**: Este tipo de NAT aplica a las comunicaciones originadas en redes externas y que van destinadas a la red NAT. En este caso, el *router* NAT deberá realizar primero una traducción de la dirección IP destino de los paquetes entrantes. Posteriormente realizará la operación inversa para los paquetes salientes.

Observe que en *srcnat* la pasarela NAT puede identificar perfectamente quienes son los equipos finales de la comunicación a partir de la información que contiene el primer paquete IP de dicha comunicación, mientras que esto no es posible en *dstnat*.

Es decir, la pasarela NAT no tendrá forma de saber cuál es el equipo de la red NAT al que va dirigido el primer paquete de una comunicación entrante (originada en una red externa) si se basa exclusivamente en la información contenida en dicho paquete. Este hecho junto a la protección requerida para mitigar las vulnerabilidades creadas por permitir las comunicaciones entrantes a la red NAT hacen que las configuraciones en los *routers* NAT asociadas a *dstnat* sean en general más complejas.

1.1 Información básica para la realización de la práctica

En esta sección se ofrece información básica y las referencias necesarias para llevar a cabo las tareas que se proponen en la práctica.

1.1.1 Acceso al puesto de usuario y elección de sistema operativo

Para la realización de esta práctica, es necesario formar grupos de tres. Después, arrancar su puesto de usuario con la opción "Redes"→"Ubuntu 20.04".

Una vez que se haya identificado como "**operador**"/"**finisterre**", podría pasar a modo *superusuario* mediante el siguiente comando, y utilizando la contraseña "**finisterre**"

```
$ sudo su
```

1.1.2 Escenario de trabajo y dispositivos implicados

En las Figuras 2 y 3 se observan los diferentes escenarios de trabajo para efectuar SRC-NAT o DST-NAT, respectivamente. El direccionamiento IP de los elementos que aparecen en las figuras, se corresponde con aquellas direcciones que se encontrarían en la isla 1. La práctica se realizará en grupos de tres, ubicándose cada miembro en cada uno de los PC que están sin sombrear en ambas figuras.

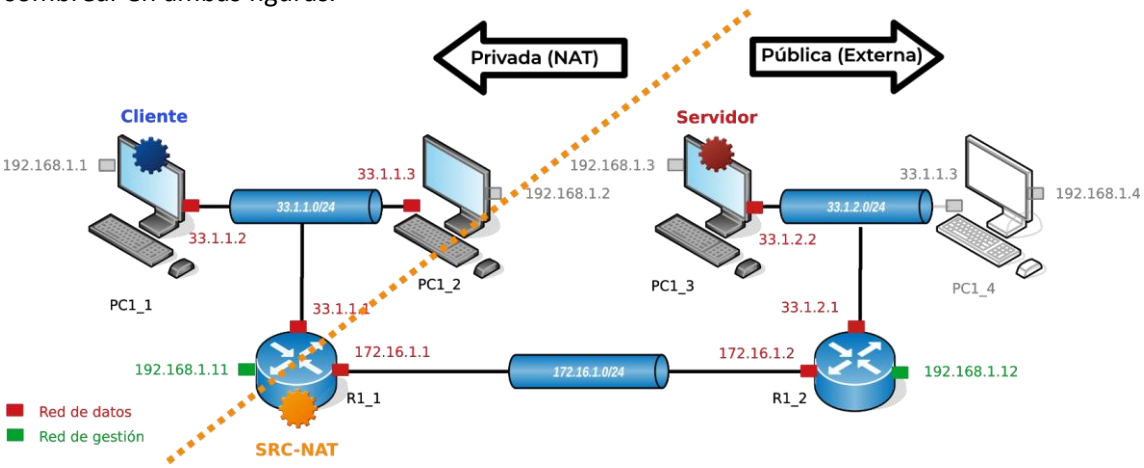


Figura 2: Escenario de trabajo y dispositivos implicados para SRC-NAT.

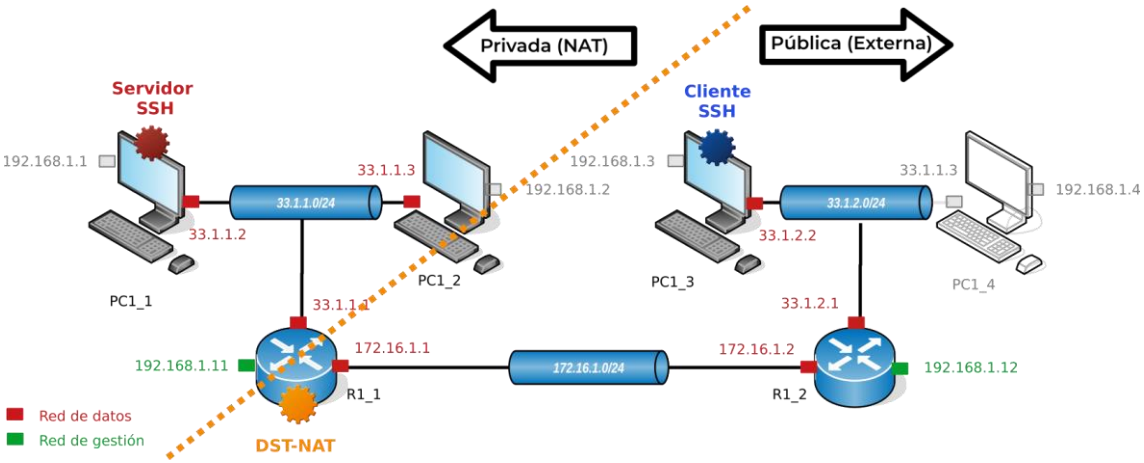


Figura 3: Escenario de trabajo y dispositivos implicados para DST-NAT.

1.1.3 Configuración de *srcnat*

Para llevar a cabo la configuración en los *routers MikroTik,* de modo que actúen como *router* NAT, será necesario acceder al dispositivo en cuestión a través de su IP de gestión mediante la aplicación Winbox. La dirección IP de la interfaz de gestión en los *routers MikroTik* tiene el formato 192.168.X.1Y, donde *X* es número de isla e *Y* es el número de subred dentro de dicha isla. La configuración NAT se lleva a cabo desde el menú *IP -> Firewall -> NAT* del *router* con la herramienta Winbox.

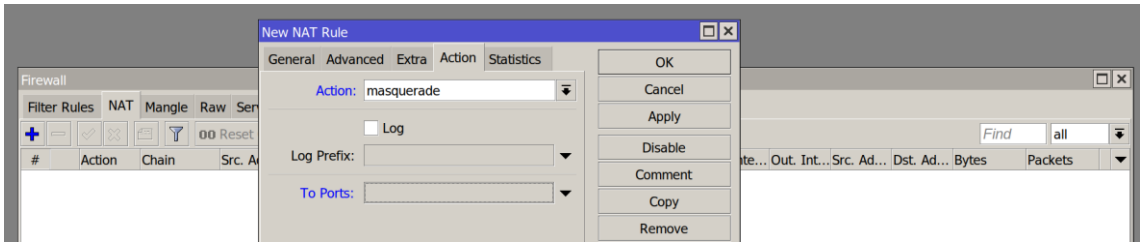


Figura 4: Configuración src-nat en Mikrotik.

La acción `masquerade` (única subversión de la acción `src-nat`) está específicamente diseñada para escenarios *srcnat* en los que la dirección IP externa es dinámica (puede cambiar). Utilice esta opción en la práctica.

1.1.4 Configuración de *dstnat*

Para dirigir todas las solicitudes a un *host* (servidor) dentro de una red interna (privada) cuando estas tienen un determinado puerto destino (acción comúnmente conocida como "*abrir un puerto*" o "*mapeo de puerto*") se puede activar la siguiente regla dentro I*P->Firewall->NAT:*

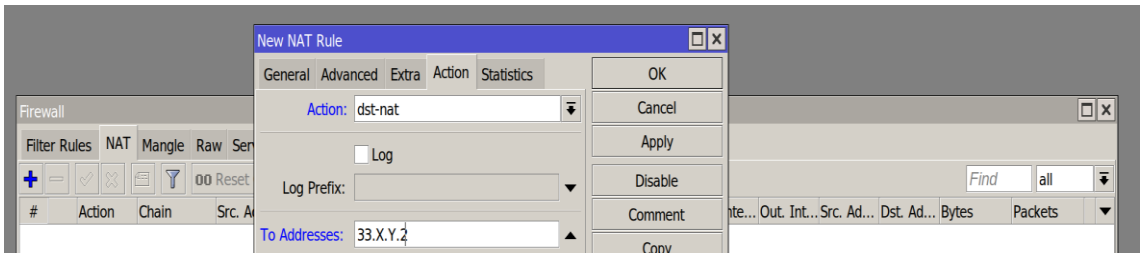


Figura 5: Configuración src-nat en Mikrotik.

Activando la regla de arriba, el router *MikroTik* redirigirá todos los paquetes al *host* con dirección IP 33.X.Y.2 dentro de la red interna (privada). En caso de que la petición fuese a un puerto en concreto, se procedería de forma similar a como se hizo a la hora de configurar reglas de filtrado en el *firewall*.

1.2 Realización práctica

Es necesario configurar las tablas de encaminamiento tanto en los PC implicados como en el *router* que hace NAT y que separa la parte pública de la privada.

1) Configure su *router*, el que está directamente conectado a su subred, para habilitar *srcnat* en dicha subred. Note que la dirección IP externa será la que tenga asignada su *router* en la subred 176.16.X.0/24 (ver Figura 2). Ejecute Wireshark en cada uno de los *hosts* implicados para ver el intercambiando mensajes ICMP sobre la interfaz `datos` y aplique el filtro "`icmp`" en ambas instancias de Wireshark. Después, use la utilidad *ping* para generar mensajes `ICMP` entre un *host* de su subred y un *host* de la subred que formada por los PC PCX_3 y PCX_4 así como el RX_2. Observe las diferencias que existen en las cabeceras de los protocolos IP e ICMP de un paquete capturado en el *host* de su subred y ese mismo paquete capturado en un *host* de una subred distinta. Analice los resultados.

CHECKPOINT: Avise al profesor cuando termine esta tarea.

2) Configure el mismo *router*, el que está directamente conectado a su subred, para habilitar *dstnat*. Concretamente, configure el *router* de modo que las conexiones SSH entrantes externas vayan a uno de los equipos de su subred (ver Figura 3). Conéctese vía SSH a dicho equipo desde cualquier equipo de otra subred. Por último, ejecute instancias de Wireshark en cada uno de los equipos (cliente SSH y servidor SSH). Capture el tráfico en la interfaz `datos`, aplique el filtro correspondiente y compare las diferencias entre el tráfico capturado en ambas partes.

Tenga en cuenta que el protocolo SSH transporta sus mensajes sobre TCP sobre el puerto 22.

CHECKPOINT: Avise al profesor cuando termine esta tarea.

1.3 Bibliografía

[1] Manual de MikroTik: NAT.
https://wiki.mikrotik.com/wiki/Manual:IP/Firewall/NAT#Summary
[2] RFC 2663: IP Network Address Translator (NAT) Terminology and Considerations.
https://datatracker.ietf.org/doc/html/rfc2663

1.4. Práctica 3

Universidad de Granada

Fundamentos de Redes

3º del Grado en Ingeniería Informática

Dept. Teoría de la Señal, Telemática y Comunicaciones

Práctica 3 – Servicios avanzados de red: HTTP y HTTPS

1.1 Objetivo

El objetivo de esta práctica es conocer y familiarizarse con las tareas habituales en la administración y configuración de los protocolos HTTP (*HyperText Transfer Protocol*) y HTTPS (HTTP *Secure*) usando el servidor Apache2 en Linux. Para ello se proponen los siguientes ejercicios:

- Instalación y configuración básica de Apache2
- Configuración de *Virtual Hosts*
- Restricción de accesos a sitios Web
- Captura y análisis del tráfico HTTP
- Generación de un certificado SSL (Secure Socket Layer) autofirmado con la utilidad `openssl` para autenticar un sitio Web
- Configuración de un sitio Web con Apache para su acceso mediante HTTPS
- Captura y análisis del tráfico TLS (*Transport Layer Security*) generado cuando se accede a un sitio web mediante HTTPS

1.2 Información básica para la realización de la práctica

En esta sección se ofrece la información básica y las referencias necesarias para llevar a cabo las tareas que se proponen en la práctica.

1.2.1 Acceso al sistema y elección de sistema operativo

 Una vez que se haya identificado, puede pasar a modo *superusuario* mediante el siguiente comando, y utilizando la contraseña "**finisterre**":

```
# sudo su
```

Para la realización de esta práctica, es necesario arrancar el equipo con la opción "Redes"→"Ubuntu 20.04". La práctica se realizará en parejas en donde uno de los equipos actuará como cliente y otro como servidor web. Será en este último en donde se configure adecuadamente Apache2.

1.2.2 Gestión y configuración básica de Apache2 (HTTP)

El protocolo HTTP facilita el acceso desde un cliente (que ofrece una interfaz universal o *navegador*) a recursos web (texto, imágenes, vídeo, etc) situados en un servidor identificado por su dirección IP o su nombre de dominio. El servidor Apache2, una de las implementaciones de HTTP de mayor difusión, ya está instalado en las máquinas Ubuntu 20.04 y se ejecutará en modo *standalone*. Con el siguiente comando podemos gestionar y comprobar el estado del servicio (iniciar, parar, reiniciar, re-ejecutar sin perder conexiones, deshabilitar o habilitar el inicio automático):

```
# sudo systemctl [start|stop|restart|reload|disable|enable] apache2
```

Para comprobar la correcta instalación y funcionamiento del servidor, después de iniciar este, acceder a la URL `http://localhost` o `http://direccion_ip` desde cualquier navegador usando cualquiera de las direcciones IP locales disponibles. La página web que aparecerá será la que se define por defecto durante la instalación de Apache2.

La configuración de Apache2 se lleva a cabo principalmente mediante la edición de una serie de ficheros de texto. Los más relevantes son:

- `/var/www/html`: Aquí se almacena el contenido del sitio web. Se puede modificar en los ficheros de configuración.
- `/etc/apache2/apache2.conf`: Configuración principal de Apache2. Las diferentes características y parámetros se definen mediante directivas.
- `/etc/apache2/ports.conf` : Puertos en los que Apache2 atenderá solicitudes.
- `/etc/apache2/sites-available/`: Directorio donde almacenan los *hosts* virtuales. Ver Sección 1.2.3 para más detalles.
- `/etc/apache2/sites-enabled/`: Directorio donde se almacenan los *hosts* virtuales.
- `/etc/apache2/mods-available/` y `/etc/apache2/mods-enabled/`: Contienen los módulos disponibles y habilitados, respectivamente. Por ejemplo para dar soporte a transacciones MySQL o PHP.
- `/var/log/apache2/access.log`: Fichero de trazas donde se almacenan todas las transacciones.
- `/var/log/apache2/error.log`: Fichero donde se registran los errores.

Para obtener más ayuda sobre la configuración de Apache2 y las directivas disponibles, abra un navegador web y visite la dirección:

```
http://httpd.apache.org/docs/2.4/
```

1.2.3 Configuración de Virtual Hosts en Apache2

Un servidor Apache2 puede dar servicio a varios sitios Web (con diferentes nombres de dominio) simultáneamente como si fueran funcionalmente diferentes, aunque en realidad es un único proceso el que los sirve. A cada uno de estos sitios se les denomina *host* virtual.

Cada *host* virtual puede tener su propio directorio raíz, su propia política de seguridad y en definitiva, su propia configuración.

La configuración comienza creando un directorio raíz diferente para cada uno de los sitios web o *hosts* virtuales a configurar:

```
# sudo mkdir –p /var/www/midominio1.com/
# sudo mkdir –p /var/www/midominio2.com/
```

Dentro de cada dominio debe crearse un fichero `index.html` como por ejemplo:

```
<!DOCTYPE html>
  <head>
    <meta charset="utf-8">Mi dominio 1</title>
  </head>
  <body>
  <h1>Home de mi dominio 1</h1>
  </body>
</html>
```

Es necesario cambiar el propietario de los directorios raíz y de sus ficheros al usuario de Apache2 `www-data`, por ejemplo:

```
# chown -R www-data:www-data /var/www/midominio1.com
```

En el directorio `/etc/apache2/sites-available` editamos los ficheros de configuración de los diferentes *hosts* virtuales a crear. Es habitual nombrar los ficheros con el correspondiente nombre de dominio, por ejemplo `midominio1.com.conf`

```
  <VirtualHost *:80>
    ServerName midominio1.com
    ServerAlias www.midominio1.com
    ServerAdmin webmaster@midominio1.com
    DocumentRoot /var/www/midominio1.com/

    <Directory /var/www/midominio1.com/>
        Options -Indexes +FollowSymLinks
        AllowOverride All
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/midominio1.com-error.log
    CustomLog ${APACHE_LOG_DIR}/midominio1.com-access.log combined
</VirtualHost>
```

Para obtener más ayuda sobre la configuración de Apache2 y las directivas disponibles, abra el navegador web y visite la dirección:

```
http://httpd.apache.org/docs/2.4/vhosts/
```

Una vez creados los ficheros de configuración, hay que crear enlaces simbólicos desde el directorio `sites-available` al directorio `sites-enabled`. Esto se puede hacer directamente con el comando `ln`, o bien usando un *script* especialmente preparado en la distribución de Apache2:

```
# sudo a2ensite midominio1.com
```

Comprobamos si la configuración tiene algún error:

```
# sudo apachectl configtest
```

La creación de *hosts* virtuales no implica que se creen automáticamente las entradas para que DNS sepa resolver adecuadamente los nombres de los dominios creados. Para ello, habría que configurar adecuadamente DNS para los nombres de dominios virtuales. No obstante, para hacer pruebas, esto se puede hacer localmente editando el fichero `/etc/hosts`, añadiendo los nombres de dominios virtuales creados a una de las IP locales, por ejemplo la IP de la interfaz de datos. Una vez hecho esto, reiniciamos el servicio:

```
# sudo systemctl restart apache2
```

Y finalmente, desde un navegador comprobamos que los diferentes dominios creados son servidos sin problema y de acuerdo con la configuración deseada.

1.2.4 Restricción de accesos

En muchas ocasiones es conveniente restringir el acceso ciertas zonas o directorios del sitio web con algún nivel de protección. Apache2 proporciona funcionalidad para ello. En este ejercicio aprenderemos a restringir el acceso al directorio `http://<dominio>/mi_zona_restringida`.

En primer lugar, hay que crear el directorio `/var/www/<dominio>/mi_zona_restringida`

Las directivas para restringir accesos, se pueden definir en el fichero de configuración principal del servidor (típicamente en la sección `<Directory>`

de `httpd.conf`), o en cada uno de directorios a restringir mediante ficheros `.htaccess`

Si se hace con `.htaccess`, en el servidor hay que permitir que se puedan definir directivas de autenticación en estos ficheros. Esto se hace con la directiva `AllowOverride`, la cual especifica qué directivas pueden ser definidas en el fichero de configuración del directorio asociado al sitio desplegado. Por ejemplo, si queremos permitir la configuración de directivas de autenticación con `.htaccess`, modificaremos el fichero `/etc/apache2/apache2.conf` tal y como sigue:

```
AllowOverride All
```

Además, es necesario crear un fichero de contraseñas, para ello usaremos la utilidad `htpasswd` con la siguiente sintaxis, en la que la opción `–c` es para crear por primera vez el fichero si no existe:

```
# sudo htpasswd -c /usr/local/passwords.pd miusuario
```

Una vez hecho esto, el siguiente paso se puede realizar editando el fichero `.htaccess` ubicado en `/var/www/<dominio>/mi_zona_restringida` como se muestra a continuación

```
AuthType Basic
AuthName "Directorio con control de acceso"
# (las siguientes directivas son opcionales)
AuthUserFile "/usr/local/passwords.pd"
Require user miusuario
```

`AuthType` define la metodología de autenticación. Nótese que `Basic` implica un mecanismo de identificación más que de autenticación, ya que `Basic` consiste en enviar un *login* y un *password* en texto plano desde el cliente, procedimiento este vulnerable a ataques de repetición. Para evitar esta vulnerabilidad, es recomendable usar el módulo `mod_ssl` que encripta toda la transacción.

`AuthName` define un nombre para la zona de seguridad. Una vez que nos hayamos autenticado en esta zona, el cliente reintentará automáticamente las mismas credenciales en todas las zonas protegidas con el mismo nombre en este servidor.

`AuthUserFile` define el fichero de contraseñas.
`Require` define al usuario al que se le permite el acceso. Alternativamente se puede usar `Require valid-user` lo que implicaría que se permite el acceso genérico a cualquier usuario definido en el fichero de contraseñas.

Finalmente, para que el proceso se reconfigure leyendo las nuevas directivas, tras la edición realizada es necesario re-arrancar el proceso mediante la utilidad `systemctl`.

1.2.6 Creación de un certificado SSL

Un certificado digital típicamente se expide por una Autoridad de Certificación (entidad de confianza) que vincula de forma fehaciente a una entidad con su clave pública. Se trata de una especie de tarjeta de visita en donde la Autoridad de Certificación garantiza que esa vinculación es cierta e irrevocable. El certificado está firmado con la clave privada de la Autoridad de Certificación, de tal manera que vincula datos asociados al sitio web (su identidad) y la clave pública del mismo. De este modo, es posible autenticar a un sitio web en Internet, ya que si suponemos la hipótesis de que la autoridad certificadora es de confianza, si enviamos los mensajes cifrados con la clave pública del sitio web (garantizada por la autoridad) obtenida del certificado, nadie excepto el que posea la clave privada podrá descifrar los mensajes cifrados, por lo que con este cifrado estaremos autenticando al servidor.

En esta práctica se usará la utilidad de línea de comandos `openssl` para crear el certificado SSL autofirmado (no estará expedido por una Autoridad de Certificación). En concreto usaremos el siguiente comando para crear el certificado:

```
# sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

Consulte la página de manual y compruebe las diferentes opciones y argumentos del comando anterior.

A continuación, le pedirá la siguiente información para incluirla en el certificado:

```
Country Name (2 letter code) [XX]: SP
State or Province Name (full name) []: Granada
Locality Name (eg, city) [Default City]: Granada
Organization Name (eg, company) [Default Company Ltd]: UGR
Organizational Unit Name (eg, section) []: DTSTC
Common Name (eg, your name or your server's hostname) []: frdominioseguro.com
Email Address []: webmaster@frdominioseguro.com
```

Las principales opciones empleadas para generar el certificado SSL se explican a continuación:

- `req -x509`: Selección de solicitud de firma de certificados X.509 (formato del certificado).

- `-nodes`: Indicamos que no queremos proteger el certificado con contraseña dado que Apache necesitará leerlo sin intervención del usuario.

- `-days 365`: Periodo de validez del certificado de un año.

- `-newkey rsa:2048`: Indicamos que queremos generar una clave privada para el certificado (porque no la hemos creado anteriormente). En concreto se creará una clave RSA de 2048 bits de longitud.

- `-keyout`: Para indicar el directorio en el cual queremos almacenar la clave privada creada.

- `-out`: Para indicar el directorio en el que almacenaremos el certificado a crear.

1.2.7 Configuración de Apache para habilitar el acceso a un sitio por HTTPS

Una vez disponemos de un certificado válido, es necesario configurar Apache para usar SSL. En primer lugar, es necesario habilitar el módulo Apache2 `mod_ssl`. Para ello se puede usar el *script* `a2enmod` incluido en la distribución de Apache2 como sigue:

```
# sudo a2enmod ssl
```

A continuación, Apache se debe reiniciar para activar el módulo habilitado:

```
# sudo systemctl restart apache2
```

Ahora vamos a crear un *virtual host* para que funcione con HTTPS haciendo uso del certificado SSL. En primer lugar, en el directorio `/etc/apache2/sites-available` crearemos el fichero de configuración del *virtual host* con la siguiente configuración mínima:

```
<VirtualHost *:443>
ServerName frdominioseguro.com
DocumentRoot /var/www/frdominioseguro.com

SSLEngine on
SSLProtocol -all +TLSv1.2
SSLCertificateFile          /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile     /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
```

Hay que tener en cuenta que el valor `Common Name` que configuró para el certificado ha de coincidir con el valor de la directiva `ServerName` en la configuración de arriba.

En el directorio especificado con la directiva `DocumentRoot` crearemos un archivo HTML muy simple para testeo. Primero, comenzamos creando dicho directorio raíz para el sitio:

```
# sudo mkdir –p /var/www/frdominioseguro.com
```

Ahora creamos dentro de /var/www/frdominioseguro.com el archivo HTML index.html con un editor de texto con el siguiente contenido:

```
<h1> FR DOMINIO SEGURO <\h1>
```

Es necesario cambiar el propietario de los directorios raíz y de sus ficheros al usuario de Apache2 `www-data`, por ejemplo:

```
# chown -R www-data:www-data /var/www/frdominioseguro.com
```

Una vez creados los ficheros de configuración, hay que crear enlaces simbólicos desde el directorio `sites-available` al directorio `sites-enabled`. Esto se puede hacer directamente con el comando `ln`, o bien usando un *script* especialmente preparado en la distribución de Apache2:

```
# sudo a2ensite frdominioseguro.com.conf
```

Comprobamos si la configuración tiene algún error:

```
# sudo apachectl configtest
```

Puede que le aparezca un mensaje de advertencia indicando que la directiva `ServerName` no está configurada a nivel global. Opcionalmente, para eliminar este mensaje, puede configurar `ServerName` en el nombre de dominio o la dirección IP de su servidor en `/etc/apache2/apache2.conf`. Sin embargo, este mensaje es sólo un aviso y no causará problemas si el resultado contiene `Syntax OK` (no hay errores de sintaxis en su archivo de configuración).

La creación de *hosts* virtuales no implica que se creen automáticamente las entradas para que DNS sepa resolver adecuadamente los nombres de los dominios creados. Para ello, habría que configurar adecuadamente DNS para los nombres de dominios virtuales. No obstante, para hacer pruebas, esto se puede hacer localmente editando el fichero `/etc/hosts`, añadiendo

los nombres de dominios virtuales creados a una de las IP locales, por ejemplo, la IP de la interfaz de datos. Una vez hecho esto, reiniciamos el servicio:

```
# sudo systemctl reload apache2
```

Y finalmente, desde un navegador comprobamos que el dominio creado es accesible usando `https://` al principio (`https://frdominioseguro.com`). El navegador mostrará una advertencia de seguridad porque estamos usando un certificado autofirmado y, por tanto, no está firmado por ninguna autoridad de certificación confiable. Permite el acceso al sitio haciendo click en `avanzado` y luego eligiendo la opción `aceptar riesgo y continuar`.

1.3 Realización práctica

1) Habilite el servicio HTTP en su equipo servidor. Abra un navegador web y pruebe a visitar la página de inicio desde dicho equipo (`http://localhost` o `http://127.0.0.1`). Modifique el contenido de la página de inicio, y compruebe que la dirección de su servidor es accesible.
2) Inspeccione el fichero `apache2.conf` e identifique las directivas más relevantes.
3) Cree 2 hosts virtuales con páginas de inicio diferentes y compruebe que son servidos convenientemente ante peticiones desde el cliente. Abra Wireshark e identifique los mensajes principales que se intercambian entre el cliente HTTP (navegador web) y el servidor HTTP (Apache2).
4) Cree una página de acceso restringido (es decir, que requiera usuario y contraseña antes de mostrarla) en *`http://<dominio>/restringida/`*. Utilice como credenciales de acceso el usuario *admin* y la contraseña *1234*.

 CHECKPOINT: Avise al profesor cuando termine esta tarea.

5) Cree un certificado SSL con la utilidad `openssl` para asociarlo al sitio `frpracticahttps.com`. Nombre el fichero del certificado como `frpracticahttps.crt` y el nombre del fichero de la clave privada como `frpracticahttps.key`.
6) Inspeccione los ficheros `.crt` y `frpracticahttps.key`.
7) Cree un host virtual con una página de inicio que muestre el mensaje "FR HTTPS" y configúrelo para que funcione con `HTTPS` haciendo uso del certificado creado anteriormente. Compruebe su correcto funcionamiento usando un navegador.
8) Abra Wireshark en su equipo y capture los mensajes que se generan cuando accede al sitio creado anteriormente. ¿Qué mensajes `TLS` se intercambian la aplicación cliente HTTPS (navegador web) y el servidor

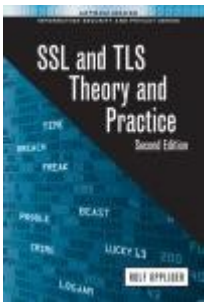
HTTPS (Apache2) durante el inicio de la conexión? ¿Qué información relevante se intercambia en esos mensajes? ¿Es posible ver los mensajes del protocolo HTTP ?

CHECKPOINT: Avise al profesor cuando termine esta tarea.

BIBLIOGRAFÍA

[1] Library: OpenSSL Cookbook, 3ed By Ivan Ristić
https://www.feistyduck.com/library/openssl-cookbook/

[2] SSL and TLS: Theory and Practice, Second Edition por Rolf Oppliger. Accesible on line desde la biblioteca de UGR.



[3] https://httpd.apache.org/docs/2.4/es/ssl/

[4] Opcionalmente para instalar LAMP (Linux + Apache + MySQL + PHP) consultar https://ubunlog.com/lamp-instala-apache-mariadb-php-ubuntu-20-04/

2 Seminarios

2.1. Seminario 1

Seminario 1: Laboratorio virtual
Guía del alumno

Introducción

El seminario 1 de FR tiene como objetivo aprender los conceptos básicos de virtualización de redes que permitan construir un laboratorio virtual que emule el funcionamiento de la red del laboratorio. El profesor le guiará paso a paso durante todo el proceso, aunque se le recomienda atienda las actividades previas siguientes.

Actividades previas al seminario

Es importante que realice las siguientes tareas con anterioridad al seminario puesto que las descargas requieren cierto tiempo.

- Descargar e instalar VirtualBox desde https://www.virtualbox.org/wiki/Downloads
- Descargar la máquina virtual de PC Linux (archivo Modelo_PC_FR.ova) desde el enlace que se le facilitará por el profesorado o desde PRADO. Importarla en VirtualBox como servicio virtualizado.
- Descargar la máquina virtual correspondiente al router utilizado en el laboratorio (archivo Modelo-MikroTik_RouterOS.ova) desde el enlace que se le facilitará por el profesorado o desde PRADO. Importarla en VirtualBox como servicio virtualizado.

Contenidos del seminario

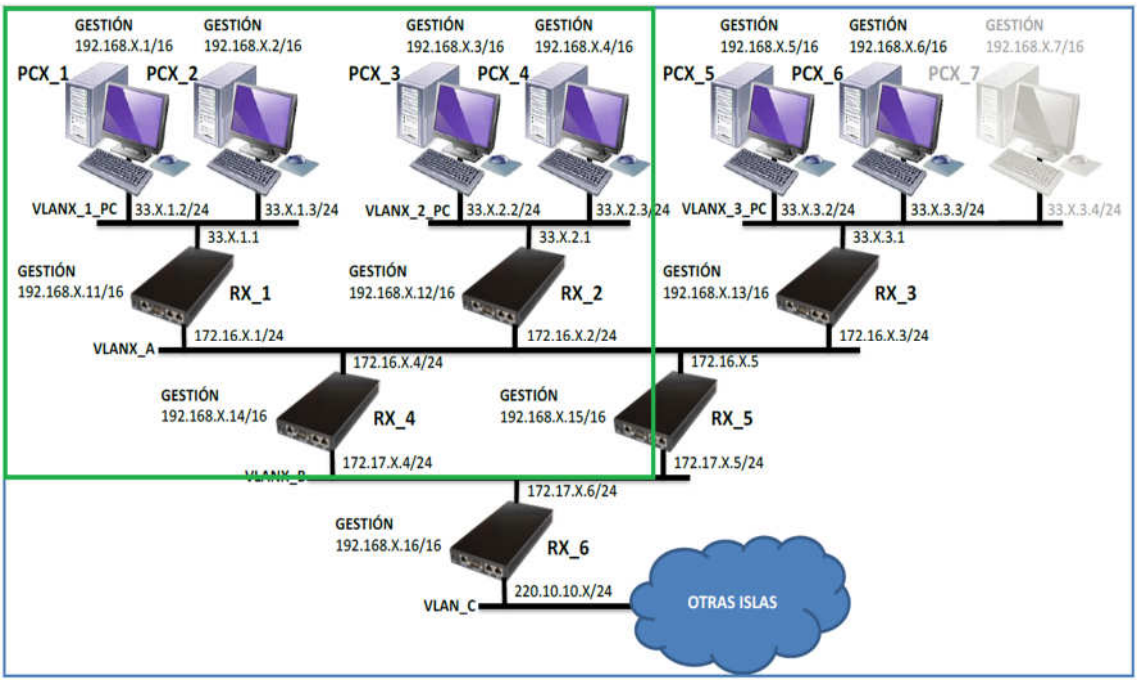
En este seminario se trabajará sobre la base de los siguientes objetivos guiados siempre por el profesor/a:

- Manejar la herramienta VirtualBox y comprender algunos conceptos básicos de virtualización
- Crear, importar y clonar máquinas virtuales en VirtualBox
- Hacer uso de máquinas virtuales de equipos de escritorio (PCs basado en Linux) y de dispositivos de interconexión de redes (routers Mikrotik)
- Configurar las interfaces de red de una máquina virtual en VirtualBox
- Modificar el nombre de los equipos y asignar direcciones IP a sus interfaces
- Comprobar la información básica de enrutamiento en los equipos
- Acceder a un router Mikrotik para configurarlo desde la interfaz de línea de comandos y desde una aplicación de forma remota (WinBox)

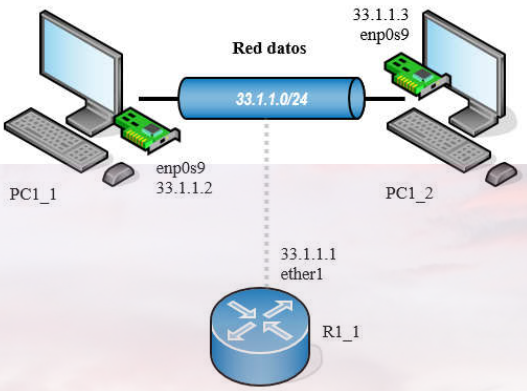
- Realizar la configuración necesaria para crear una red sencilla formada por dos PCs y un router, comprobando la conectividad mediante el uso de la herramienta ping
- Analizar capturas de tráfico en la red mediante la herramienta Wireshark y comprender algunos conceptos básicos de protocolos
- Ser capaz de extender la red sencilla formada por dos PCs y un router para construir el laboratorio virtual que emula el funcionamiento de la red del laboratorio

Desarrollo del seminario

- Queremos virtualizar la red del laboratorio, en concreto un subgrupo de PCs y routers.



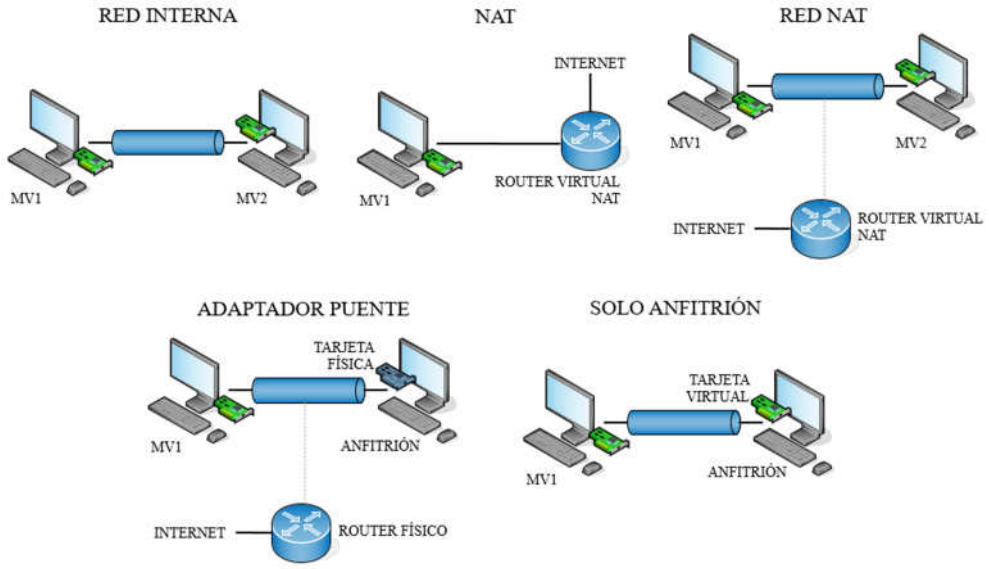
- Dado que la topología es compleja, vamos a centrarnos en una parte básica a partir de la cual se puede construir el resto (2 PCs y un router)



VirtualBox y MV de PC Linux

- Se l guiará sobre los conceptos de virtualización basados en una filosofía de capas como la de la imagen.

- Importar recurso virtualizado (máquina PC Linux descargada) y denominarlo PC1_1.
- Clonar la MV de PC Linux y renombrarla a PC1_2.
- La configuración de red en las MVs puede ser de diverso tipo:



- Nos centraremos en crear una Red Interna entre las máquinas y la denominaremos "Datos". Ya existirá y solamente habrá que habilitarla en uno de los interfaces.
- Arrancar PC1_1. Usuario: *administrador*; contraseña: *finisterre*
- Abrir una terminal y renombrar la MV:
  - Ejecutar: `sudo hostname PC11` (sin guión bajo)
  - Ejecutar: `bash` (para actualizar el prompt)

- Mostrar la configuración de red mediante el comando `ifconfig` y comprobar que el PC tiene la IP deseada (según la topología que estamos diseñando).
- Para asignar la IP de forma permanente (con ifconfig no se mantendría en esta distribución de Linux):
    - Editar fichero:
      `sudo gedit /etc/netplan/01-network-manager-all.yaml`
    - Asignar la IP correspondiente al PC1_1.
    - Actualizar estado de la red: `sudo netplan apply`
- Establecer como *gateway* por defecto la interfaz del router
    - Ejecutar: `sudo route add default gw 33.1.1.1`
- Mostrar la tabla de rutas ejecutando: `route -n`
- Repetir los pasos anteriores para PC1_2 (con la IP correspondiente).
- Una vez que tenemos las dos MVs configuradas, realizar un ping entre ellas.
    - (desde PC1_1) `ping 33.1.1.3`

MV de Router Mikrotik

- Importar el archivo OVA del router en VirtualBox (importar servicio virtualizado) y sustituir el nombre de la MV por R1_1.
- Configurar un adaptador de red en modo red interna (Datos) y deshabilitar el resto.
- Arrancar R1_1. Usuario: *admin*; contraseña: *(no tiene)*. Pulsar la tecla 'n' para no ver la licencia
- Renombrar el router a R11:
    - Ejecutar:
      `system identity set name=R11`
- Asignar una dirección IP en la interfaz ether1:
    - Ejecutar:
      `ip address add address=33.1.1.1/24 interface=ether1`
      *** *Nota: el carácter '=' corresponde a la tecla '¡', mientras que el carácter '/' corresponde a la tecla '-'* ***
    - Para eliminar una dirección IP (por si nos confundimos):
        - mostrar el # con `ip address print`
        - posteriormente borrar la dirección IP # con el comando:
        - `ip address remove numbers=#`

- Ejecutar Winbox en PC1_1 mediante el comando `wine winbox64.exe` desde el directorio principal (`/home/administrador`)
- Entrar a R1_1 desde Winbox utilizando la dirección IP 33.1.1.1 (usuario *admin* y sin contraseña)
- Abrir la ventana de direcciones IP (*IP/Adresses*) y mostrar la dirección IP que acabamos de configurar.
- Abrir la ventana de rutas (*IP/Routes*) y mostrar la red directamente conectada
- Abrir la ventana Interfaces y mostrar el nombre de la interfaz ether1, así como las estadísticas de tráfico
- Una vez configurado el router, realizar un ping entre PC1_1 y R1_1
- Usar desde Winbox la herramienta Tools/Ping para realizar un ping con origen el router
- Para apagar el router se debe ejecutar el comando: `system shutdown` en la terminal del router.

Análisis de trazas mediante software de análisis de protocolos

- Se utilizará el software wireshark.
- Ejecutar Wireshark en PC1_1: `sudo wireshark` (es necesario ser root)
- Elegir el interfaz a monitorizar.
- Analizar los paquetes transmitidos cuando se realiza un ping (entre PC1_1 y PC1_2 por ejemplo).
- Ver en la ventana principal IP origen y destino, Puertos, Protocolo, etc.
- Ver la pila de protocolos (cabeceras de la capa de enlace, red y transporte).
- Aplicar filtros de ejemplo (dirección IP de origen/destino)

Trabajo adicional

- En la topología sencilla que hemos implementado incorporar la red de gestión 192.168.1.0.
- Agregar P1_3 y R1_2 y configurar la red de datos 33.1.2.0, la red de gestión 192.168.1.0 y la red interna 172.16.1.0 (según la topología).
- Añadir rutas estáticas en las tablas de encaminamiento de los routers R1_1 y R1_2 para que haya conectividad entre PC1_1 y PC1_3. Probar un ping entre ambas MVs.

2.2. Seminario 2

Seminario 2: Diagnóstico y resolución de fallos en redes

Guía del estudiante

Introducción al entorno virtualizado

El profesor explicará al comienzo de dicho seminario el laboratorio que utilizará el estudiante y que será en el que se basará el desarrollo la actividad, identificando los recursos técnicos a utilizar, su lógica y arquitectura, distribución, configuración básica y objetivo a conseguir en su desarrollo.

Se presentará un laboratorio virtual sobre el que se tratará de identificar y solucionar una serie de errores de red típicos y presentes en cualquier red telemática de cualquier empresa.

Herramientas de diagnóstico de fallos en red

A continuación, se resumen las principales herramientas de diagnóstico de red de uso común que permiten al administrador de redes identificar y resolver problemas. El profesor abundará en información durante la realización del seminario, de modo que se obtenga el máximo de aprovechamiento.

● `ping`: es una de las herramientas más conocidas. Usa dos mensajes ICMP de tipo 8 (*Echo Request*) y tipo 0 (*Echo Reply*). El funcionamiento del `ping` es el siguiente. El origen envía un mensaje *Echo Request* al destino. Si el destino está disponible, este le envía un *Echo Reply*. Una vez el mensaje vuelva al origen, el comando ping muestra por pantalla el número de secuencia, el campo TTL y el cálculo del RTT. Al finalizar el `ping`, se muestra un resumen con estadísticas de los paquetes transmitidos, los paquetes recibidos correctamente, el porcentaje de paquetes que se han perdido y el RTT. Típicos usos:

- o `ping -R:` muestra la ruta de ida (del *Echo Request* hacia el destino) y vuelta (del *Echo Reply* hacia el origen). El RTT que calcula no es por salto, sino el del total de ida y vuelta a diferencia del `traceroute` que lo hace por saltos.
- o `ping -n:` muestra las direcciones IP en lugar de los nombres de dominio.

● `traceroute` (en sistemas operativos como GNU/Linux o Mac, o `tracert` en Windows): este comando muestra salto a salto el flujo de tráfico que hace un paquete UDP (es el paquete que se usa por defecto en `traceroute`) desde un emisor a un receptor, trazando la ruta hasta llegar al destino. De esta forma se puede conocer qué punto de la red está fallando y no deja realizar la conexión entre esos equipos. Cuando se ejecuta este comando se obtienen estadísticas del RTT o la latencia de red. Además, también indica la dirección IP de cada uno de los nodos por los que va pasando el paquete hasta llegar a su destino. El funcionamiento de `traceroute` es el siguiente (ver Figura 1). Se comienza enviando un paquete UDP al destino con TTL = 1 (el valor de TTL determina cuántos saltos puede atravesar un paquete antes de que se devuelva al origen un mensaje de tiempo excedido de ICMP) y los siguientes paquetes a enviar incrementan el campo TTL en 1 tras recibir el mensaje ICMP anterior.

- `traceroute -I`: usa ICMP para las pruebas (igual que ping).
- `traceroute -T`: usa TCP SYN para las pruebas.

● `netstat`: herramienta que muestra todos los puertos y conexiones abiertos en una máquina. Para los puertos de escucha, si la dirección de origen es 0.0.0.0, está escuchando en todas las interfaces disponibles. Si hay una dirección IP en su lugar, entonces el puerto está abierto sólo en esa interfaz específica. Al ejecutar esta herramienta en el terminal, se muestran las direcciones IP de origen y destino, así como los puertos de origen y destino. Los campos Recv-Q y Send-Q muestran el número de bytes pendientes de reconocimiento en cualquier dirección. Finalmente, el campo PID/Nombre del programa muestra el ID del proceso y el nombre del proceso responsable del puerto o conexión de escucha.

- `netstat -tln`: muestra los puertos que usan TCP en modo escucha con el puerto en formato número. La opción u (en lugar de t) lista los puertos que usan el protocolo UDP.
- `netstat -tn`: muestra los puertos que usan TCP con conexiones establecidas con el puerto en formato número.

Para mostrar la utilidad de `netstat` se puede utilizar la siguiente herramienta. de red:

- `netcat`: permite abrir puertos TCP/UDP en un host y realizar el rastreo del tráfico en esos puertos. También se puede transferir cualquier tipo de archivo. Ejemplo de uso: En el servidor:

```
# nc -l 12345
# netstat -tln
```

En el cliente (misma máquina, distinta terminal):

```
# nc localhost 12345
# netstat -tn
```

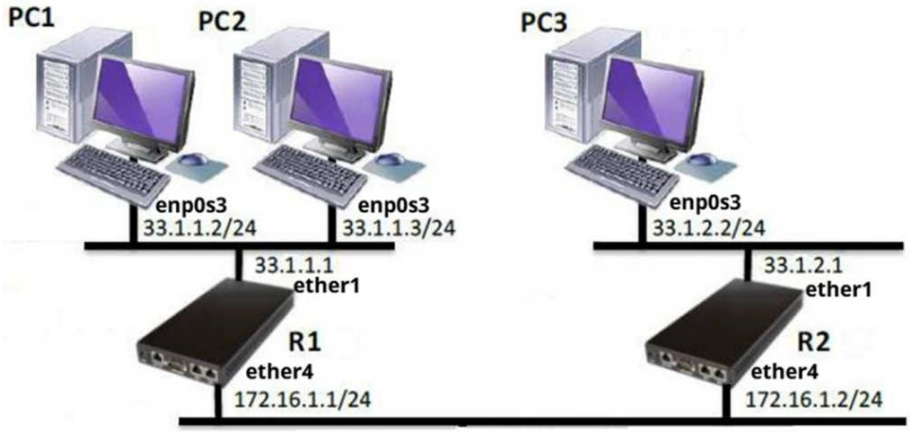
● `tcpdump`: es una herramienta de captura de paquetes que se utiliza para solucionar problemas de conectividad de red (muy parecido a `wireshark`), sólo que más liviano y se ejecuta en la línea de comandos.

- `tcpdump -D`: muestra todas las interfaces disponibles.
- `tcpdump -n -i [nombre_interfaz]`: captura paquetes IP en esa interfaz y muestra la información (direcciones IP, puertos) en formato numérico.

● `wireshark`: software *open-source* de monitorización y análisis de tráfico de red, que suele usarse como analizador de protocolos. Sirve como una herramienta didáctica para el estudio de las comunicaciones y para la resolución de problemas de red. Se pueden visualizar los campos de cada una de las cabeceras y capas que componen los paquetes monitorizados, proporcionando un gran abanico de posibilidades al administrador de redes a la hora de abordar ciertas tareas de análisis de tráfico.

Laboratorio virtual de FR para el diagnóstico y solución de problemas de red

En la siguiente imagen se muestra la red virtual sobre la que trabajaremos en el seminario. Esta red, basada en la topología que hay en el laboratorio físico del Centro por lo que se irá teniendo contacto con el mismo de cara a la prácticas de laboratorio. Estará montado en el fichero OVA que deberán descargar e importar en VirtualBox los estudiantes de la plataforma virtual de la asignatura (PRADO).



Ejecución del seminario paso a paso

Una vez tengan instalado el entorno de red virtual (compuesto por 3 PCs y 2 routers), podrán comenzar con el desarrollo del seminario. En él, con la guía y tutorización del profesor/a se irán detectando distintos problemas de conectividad entre los equipos de la red virtual, y aprenderán a corregirlos. Por lo tanto, importe el servicio de red para el Seminario y arranque solo la máquina PC1.

Fallo 1: Máquinas virtuales en operación. Realice un ping a PC2. ¿Funciona?

Fallo 2: Adaptadores de red presentes en nodos de red. Cuando estén todos los equipos de la red encendidos, ejecute el comando `ifconfig` para visualizar todas las interfaces de red que existen en los PCs. Compruebe que existan todas las interfaces necesarias denominadas `enp0s3` y `lo` (loopback). ¿Falta alguna? En caso de que falte alguna interfaz, trate de levantarla (habilitarla):

```
# sudo ifconfig enp0s3 up
```

Interprete el error que aparece: "No such device". Trate de solucionarlo con la ayuda del profesor/a. Para habilitar la tarjeta de red de un sistema virtualizado, desde VirtualBox es equivalente a instalar en un PC real (no virtual) una tarjeta de red. Tenga en cuenta que esto es distinto a habilitar/deshabilitar una tarjeta de red desde el sistema operativo. Para hacer esto último, puede usar el siguiente comando:

- Para deshabilitar la interfaz:
  ```
  # sudo ifconfig [nombre_interfaz] down
  ```
- Para habilitar la interfaz:

```
# sudo ifconfig [nombre_interfaz] up
```

Fallo 3: Conectividad de cable. Una vez presenta la tarjeta de red en los equipos del laboratorio, ejecute `ifconfig` en PC1. ¿Observa alguna diferencia entre la información que aparece sobre la interfaz `lo` y la interfaz `enp0s3`? Preste atención a las estadísticas (por ejemplo, el número de paquetes transmitidos o recibidos).

Realice un `ping` a PC2 y analice la información que aparece en pantalla. ¿Qué puede estar indicando el mensaje "red inalcanzable"?

Compruebe si hay cable conectado a la tarjeta de red en PC1. En el caso de que no esté conectado, con la ayuda de su profesor/a conéctelo.

Fallo 4: Comprobación de direccionamiento IP. En PC1, con el cable ya conectado, pruebe a realizar un `ping` a PC2. ¿Qué puede significar el error "host de destino inalcanzable"?

Compruebe que la dirección IP de la interfaz `enp0s3` en todos los PCs es correcta. Esto lo puede hacer ejecutando `ifconfig` en una terminal.
Para configurar una dirección IP puede hacerlo ejecutando lo siguiente:

```
# sudo ifconfig enp0s3 33.1.1.3 netmask 255.255.255.0
```

El comando anterior permitirá modificar la dirección IP de forma temporal. Si queremos que los cambios se mantengan al apagar y encender el PC, es necesario modificar la configuración de red en el archivo `/etc/netplan/01-network-manager-all.yaml`. En PC2, abra este archivo y edite la dirección IP para que sea correcta. Posteriormente, actualice la configuración de red en el equipo ejecutando:

```
# sudo netplan apply
```

Compruebe que puede realizar `ping` con éxito entre PC1 y PC2.
Fallo 5: Comprobación de presencia de gateway/router/encaminador en la red. Realice un `ping` desde PC1 a PC3 y analice el error que se muestra por pantalla. ¿Qué puede indicar "red inalcanzable"? Verifique con `traceroute` si los paquetes transmitidos salen de PC1.

A continuación, realice el `ping` desde PC3 a PC1. Por qué ahora el error es "red de destino inalcanzable"? Verifique con `traceroute` si los paquetes transmitidos salen de PC3 y alcanzan el destino.

Para modificar la puerta de enlace predeterminada o *default gateway* se puede realizar ejecutando el siguiente comando:

```
# sudo route add default gw 33.1.1.1
```

Puede comprobar si esta información se ha añadido como entrada en la tabla de encaminamiento del PC mediante el siguiente comando:

```
# route -n
```

Con el anterior comando es posible que la ruta introducida se almacene sólo de forma temporal. Para que este cambio sea permanente, debe modificar la configuración de red en el archivo `/etc/netplan/01-network-manager-all.yaml`. En PC1, abra este archivo y edite la línea `gateway4` para que la configuración sea correcta. Es importante que dicha línea quede bien alineada a la anterior, de lo contrario dará error. Posteriormente, actualice la configuración de red en el equipo ejecutando `netplan`.

Tras realizar el cambio, compruebe si el `ping` desde PC1 a PC3 funciona. En caso negativo, realice también el `ping` al revés, desde PC3 a PC1, y observe las diferencias en los errores mostrados en pantalla. ¿Por qué desde el PC1 el `ping` se queda "colgado", mientras que desde PC3 el error es "red de destino inalcanzable"?

Fallo 6: Comprobación de existencia de tabla de encaminamiento correcta en los routers del laboratorio. Compruebe con `traceroute` desde PC1 a PC3 si se alcanza el router R2. Analice mediante `wireshark` ejecutándose en PC3 si se capturan los paquetes ICMP de un `ping` desde PC1 a PC3. Para lanzar `wireshark`, ejecute:

```
# sudo wireshark-gtk
```

Realice también un `ping` desde PC3 a PC1 y analice los paquetes capturados con `Wireshark` ejecutándose en PC3. Trate de explicar lo que está ocurriendo.

Entre a R2 con *Winbox* desde el PC3. Para ello, desde una terminal diríjase al directorio `Desktop/Software` y ejecute:

```
# wine winbox.exe
```

En el programa, escriba la dirección IP del router (33.1.2.1) y rellene las credenciales de acceso (usuario `admin`, sin contraseña). A continuación, seleccione en el menú IP → Routes y observe la tabla de encaminamiento del router. ¿Falta alguna entrada que sea necesaria para que haya conectividad entre PC1/PC2 y PC3?

Añada la ruta que falta para encaminar tráfico hacia la subred de PC1/PC2. Tenga en cuenta que la red de destino es 33.1.1.0/24 y la dirección IP de la puerta de enlace es 172.16.1.1.

Compruebe que el `ping` entre PC1/PC2 y PC3 funciona.

Fallo 7: Control de seguridad. Tráfico de TELNET restringido en R2. Pruebe a realizar `telnet` desde PC1 y PC2 a PC3. Para ello, ejecute:

```
# telnet 33.1.2.2
```

¿Por qué se queda la terminal "colgada" sin establecer la conexión? Puede usar `wireshark` en PC3 para verificar si la petición de conexión llega a PC3. Recuerde ejecutar esta herramienta con el comando `sudo wireshark-gtk`.

Acceda a R2 con *Winbox*. A continuación, seleccione en el menú IP → Firewall y observe el contenido de la tabla. Las reglas que aparecen son las siguientes:

#0: chain: forward, dst. address: 33.1.2.0/24, protocol: icmp, action: accept
#1: chain: forward, src. address: 33.1.2.0/24, protocol: icmp, action: accept
#2: chain: forward, action: drop
#3: chain: forward, dst. address: 33.1.2.0/24, protocol: tcp, dst. port: 23, action: accept
#4: chain: forward, src. address: 33.1.2.0/24, protocol: tcp, src. port:23, action: accept

El significado de estas reglas es el siguiente:

#0: permite el tráfico tipo ICMP (`ping`) con destino cualquier IP de la subred 2 (PC3)
#1: permite el tráfico tipo ICMP (`ping`) procedente de cualquier IP de la subred 2 (PC3)
#2: descarta todo el tráfico que atraviese el router
#3: permite el tráfico `telnet` (TCP, puerto 23) con destino la subred 2 (PC3)
#4: permite el tráfico `telnet` (TCP, puerto 23) procedente de la subred 2 (PC3)

Tenga en cuenta que las reglas se comprueban de arriba abajo y sólo se "dispara" la primera que se cumpla. ¿Es necesario realizar algún cambio en las reglas para que el servicio TELNET funcione correctamente?

Mueva la regla `drop` abajo del todo arrastrándola con el ratón. Esta regla no deja pasar ningún tráfico a través del router. Se suele poner la última, de modo que todo el tráfico que no se haya dejado pasar con las reglas anteriores, se descartará.

Pruebe a realizar `telnet` desde PC1 y PC2 a PC3. ¿Funciona?

Fallo 8: Comprobación de puertos de red habilitados. En PC3 ejecute `netstat -tln` para comprobar que hay un proceso (TELNET) escuchando en el puerto 23.

A continuación, use `wireshark` para comprobar si la petición de conexión de TELNET llega a PC3. Filtre los paquetes añadiendo el siguiente filtro en la barra superior:

```
ip.addr == 33.1.1.2.
```

Analice los mensajes que aparecen en pantalla. ¿Por qué aparecen sucesivas retransmisiones? ¿Se llega a establecer la conexión TCP al puerto 23? Compruebe el estado del *firewall* de PC3. Para ello, ejecute:

```
# sudo ufw status
```

Las siglas UFW significan *Uncomplicated Firewall* y hacen referencia a una aplicación que tiene como objetivo establecer reglas en `iptables`, las tablas de *firewall* nativas en los sistemas Linux. Puesto que `iptables` tiene una sintaxis relativamente compleja, utilizar UFW para realizar su configuración es una alternativa útil. Compruebe observando el resultado si existen reglas que limitan el tráfico del servicio TELNET. Para permitir el tráfico hacia el puerto 23, ejecute lo siguiente:

```
# sudo ufw allow 23
# sudo ufw status
```

Compruebe si es posible realizar una conexión TELNET desde PC1 a PC3. ¿Funciona? Ejecute en una terminal `netstat -tn` para comprobar que se ha establecido la conexión TCP.

Fallo 9: Restricción de servicios de red en PC3. Realice un TELNET desde PC2 a PC3 y compruebe si funciona.

Para controlar el acceso a las aplicaciones, existe un mecanismo denominado *TCP Wrapper*, que consiste en una biblioteca que provee un control de acceso simple y administración de *logs* estandarizada para aplicaciones que lo soporten y reciban conexiones de red. Los *TCP Wrappers* son, por tanto, listas de control de acceso (ACL) basadas en *hosts* y utilizadas para filtrar accesos de red a los servicios locales.

Use `wireshark` en PC3 para comprobar si el paquete llega al destino. Añada el siguiente filtro: `ip.addr == 33.1.1.3`. ¿Se establece la conexión TCP? En caso afirmativo, ¿llega la petición de conexión de TELNET? ¿Qué ocurre con la respuesta?

Abra en PC3 el fichero `/etc/hosts.deny` y observe la línea:

```
in.telnetd: ALL EXCEPT 33.1.1.2
```

¿Qué significado tiene esta línea? Para permitir el tráfico desde PC2 añada su dirección IP al final de la línea, separando las direcciones IP con una coma.

Finalmente, compruebe si funciona TELNET desde PC2 a PC3. Tenga en cuenta que, para salir de TELNET, debe ejecutar el comando `exit`. Preste atención al *prompt* de la terminal para identificar la máquina en la que ejecuta los comandos.

COMPROBACIONES/HERRAMIENTAS EXTRAS. En función del desarrollo del seminario en clase, el profesor expondrá y se experimentará con herramientas de diagnóstico de red adicionales disponibles en el laboratorio virtual de estudiante y en Internet.

2.3. Seminario 3

Fundamentos de Redes

Seminario 3: Resolución de problemas del Tema 2

Curso 2023/2024

Profesor: Jesús Minguillón, minguillon@ugr.es

UNIVERSIDAD DE GRANADA

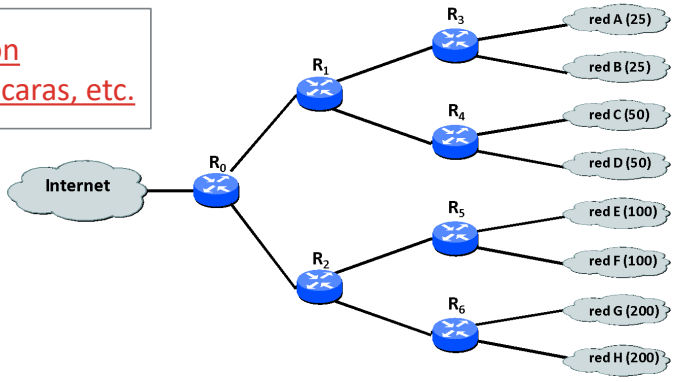
DPTO. TEORÍA DE LA SEÑAL, TELEMÁTICA Y COMUNICACIONES

Ejercicio 1

ASIGNACIÓN DE DIRECCIONAMIENTO Y ENCAMINAMIENTO IP

Se dispone de una red con la siguiente topología. Cada una de las redes finales (redes A, …, H) está compuesta por el número de hosts indicado entre paréntesis. Además, se ha contratado el rango de direcciones públicas 168.168.168.0/22.

Enlace para practicar con direcciones de red, máscaras, etc.

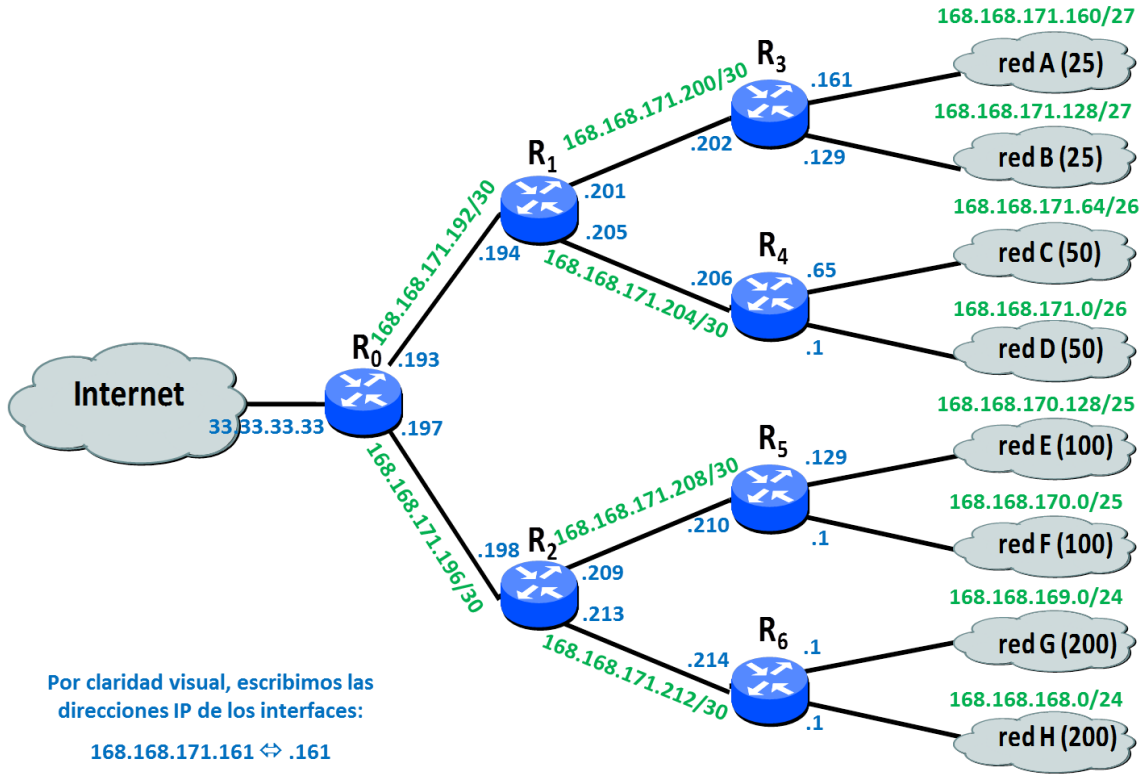


a) Proponga un esquema de asignación de direcciones (de todos los equipos) que cumpla los siguientes requisitos:

- Todos los hosts han de tener asignadas direcciones públicas.
- La asignación de direcciones ha de minimizar el tamaño de las tablas de encaminamiento.

b) Muestre las tablas de encaminamiento de todos los routers, suponiendo que se utiliza el esquema de asignación de direcciones del apartado anterior.

NOTA: El router R0 tiene una IP pública diferente en su interfaz hacia Internet, e.g. 33.33.33.33/24.



Por claridad visual, escribimos las direcciones IP de los interfaces:

168.168.171.161 ⇔ .161

R_0

Destino	Máscara	Siguiente salto	
168.168.168.0	/23	R_2 (168.168.171.198)	Hacia redes G y H
168.168.170.0	/24	R_2 (168.168.171.198)	Hacia redes E y F
168.168.171.0	/24	R_1 (168.168.171.194)	Hacia redes A, B, C y D
168.168.171.208	/29	R_2 (168.168.171.198)	Hacia subredes R_2-R_5 y R_2-R_6
168.168.171.192	/30	*	Conexión directa subred R_0-R_1
168.168.171.196	/30	*	Conexión directa subred R_0-R_2
33.33.33.0	/24	*	Conexión directa subred R_0-R_{ISP}
default	/0	IP Gateway ISP	Hacia Internet

R_1

Destino	Máscara	Siguiente salto	
168.168.171.0	/25	R_4 (168.168.171.206)	Hacia redes C y D
168.168.171.128	/26	R_3 (168.168.171.202)	Hacia redes A y B
168.168.171.192	/30	*	Conexión directa subred con R_0
168.168.171.200	/30	*	Conexión directa subred con R_3
168.168.171.204	/30	*	Conexión directa subred con R_4
default	/0	R_0 (168.168.171.193)	Hacia Internet y otras subredes

R_2

Destino	Máscara	Siguiente salto	
168.168.168.0	/23	R_6 (168.168.171.214)	Hacia redes G y H
168.168.170.0	/24	R_5 (168.168.171.210)	Hacia redes E y F
168.168.171.196	/30	*	Conexión directa subred con R_0
168.168.171.208	/30	*	Conexión directa subred con R_5
168.168.171.212	/30	*	Conexión directa subred con R_6
default	/0	R_0 (168.168.171.197)	Hacia Internet y otras subredes

R_3

Destino	Máscara	Siguiente salto	
168.168.171.160	/27	*	Conexión directa red A
168.168.171.128	/27	*	Conexión directa red B
168.168.171.200	/30	*	Conexión directa subred con R_1
default	/0	R_1 (168.168.171.201)	Hacia Internet y otras subredes

R_4

Destino	Máscara	Siguiente salto	
168.168.171.64	/26	*	Conexión directa red C
168.168.171.0	/26	*	Conexión directa red D
168.168.171.204	/30	*	Conexión directa subred con R_1
default	/0	R_1 (168.168.171.205)	Hacia Internet y otras subredes

R_5

Destino	Máscara	Siguiente salto	
168.168.170.128	/25	*	Conexión directa red E
168.168.170.0	/25	*	Conexión directa red F
168.168.171.208	/30	*	Conexión directa subred con R_2
default	/0	R_2 (168.168.171.209)	Hacia Internet y otras subredes

R_6

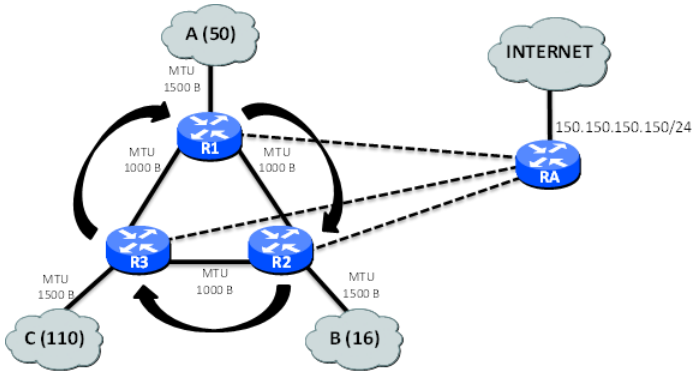
Destino	Máscara	Siguiente salto	
168.168.169.0	/24	*	Conexión directa red G
168.168.168.0	/24	*	Conexión directa red H
168.168.171.212	/30	*	Conexión directa subred con R_2
default	/0	R_2 (168.168.171.213)	Hacia Internet y otras subredes

Nota: En las tablas falta la columna “Interfaz”. También habría que indicar las interfaces ethX en el diagrama anterior.

Ejercicio 2

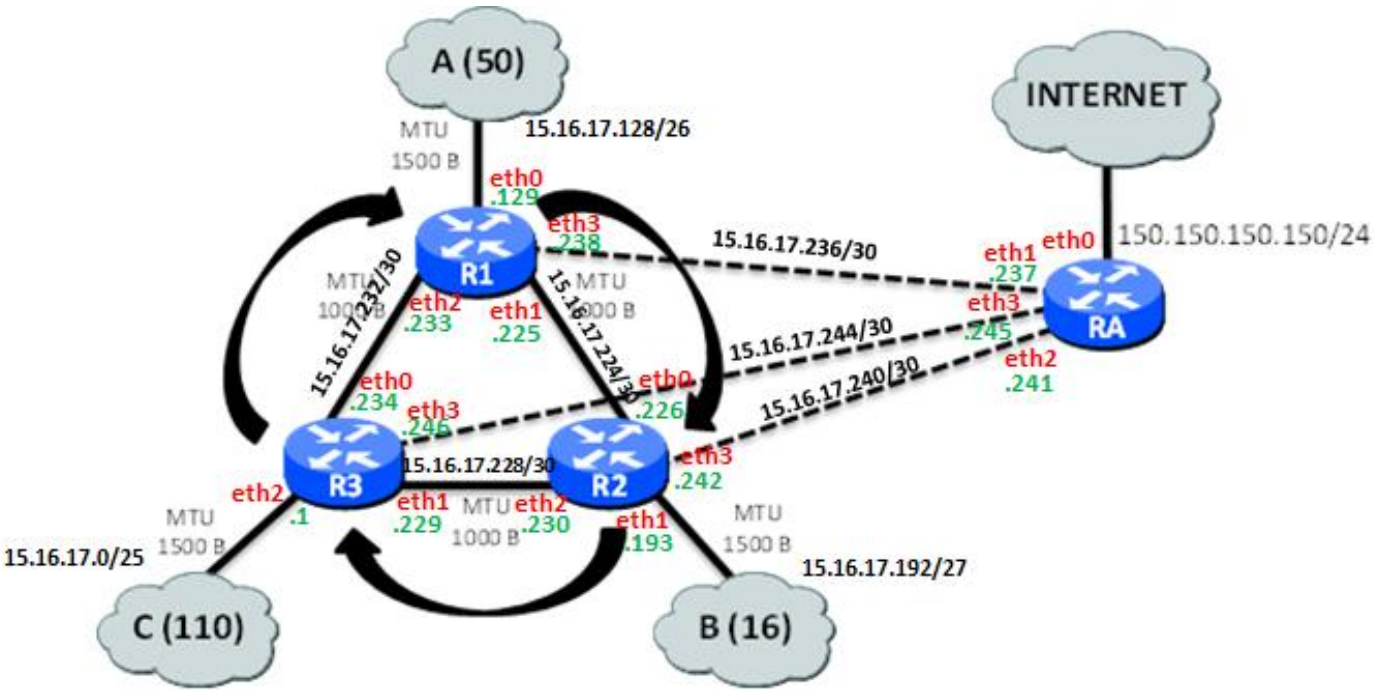
ASIGNACIÓN DE DIRECCIONAMIENTO Y ENCAMINAMIENTO IP

La siguiente figura muestra la topología de red de una empresa, que tiene contratado con su ISP el rango de direcciones 15.16.17.0/24. El número de ordenadores conectados a las redes A, B y C están indicados en la figura entre paréntesis.



a) Realice la asignación de direcciones IP tanto de equipos como de routers (incluyendo las redes entre los routers), utilizando direcciones públicas siempre que sea posible

b) Indique las tablas de encaminamiento de todos los routers de forma que, para el tráfico entre las redes A, B y C, se encamine de acuerdo a las flechas en la figura). Debe haber conectividad completa entre estas redes y hacia Internet.


A (50)
MTU
1500 B
15.16.17.128/26
eth0
.129
INTERNET
150.150.150.150/24
eth0
eth3
.238
15.16.17.236/30
eth1
.237
R1
RA
MTU
1000 B
15.16.17.232/30
eth2
.233
eth1
.225
15.16.17.224/30
eth3
.245
15.16.17.244/30
eth2
.241
eth0
.234
eth3
.246
eth0
.226
15.16.17.240/30
R3
R2
15.16.17.228/30
eth3
.242
eth2
.1
eth1
.229
MTU
1000 B
eth2
.230
eth1
.193
MTU
1500 B
MTU
1500 B
15.16.17.0/25
15.16.17.192/27
C (110)
B (16)

R_1

Destino	Máscara	Siguiente salto	Interfaz	
15.16.17.192	/27	R_2 (15.16.17.226)	eth1	Red B
15.16.17.0	/25	R_2 (15.16.17.226)	eth1	Red C
15.16.17.128	/26	*	eth0	Red A (directa)
15.16.17.224	/30	*	eth1	Directa
15.16.17.232	/30	*	eth2	Directa
15.16.17.236	/30	*	eth3	Directa
default	/0	R_A (15.16.17.237)	eth3	Internet y otras subredes

R_2

Destino	Máscara	Siguiente salto	Interfaz	
15.16.17.128	/26	R_3 (15.16.17.229)	eth2	Red A
15.16.17.0	/25	R_3 (15.16.17.229)	eth2	Red C
15.16.17.224	/30	*	eth0	Directa
15.16.17.192	/27	*	eth1	Red B (directa)
15.16.17.228	/30	*	eth2	Directa
15.16.17.240	/30	*	eth3	Directa
default	/0	R_A (15.16.17.241)	eth3	Internet y otras subredes

R_3

Destino	Máscara	Siguiente salto	Interfaz	
15.16.17.128	/26	R_1 (15.16.17.233)	eth0	Red A
15.16.17.192	/27	R_1 (15.16.17.233)	eth0	Red B
15.16.17.232	/30	*	eth0	Directa
15.16.17.228	/30	*	eth1	Directa
15.16.17.0	/25	*	eth2	Red C (directa)
15.16.17.244	/30	*	eth3	Directa
default	/0	R_A (15.16.17.245)	eth3	Internet y otras subredes

R_A

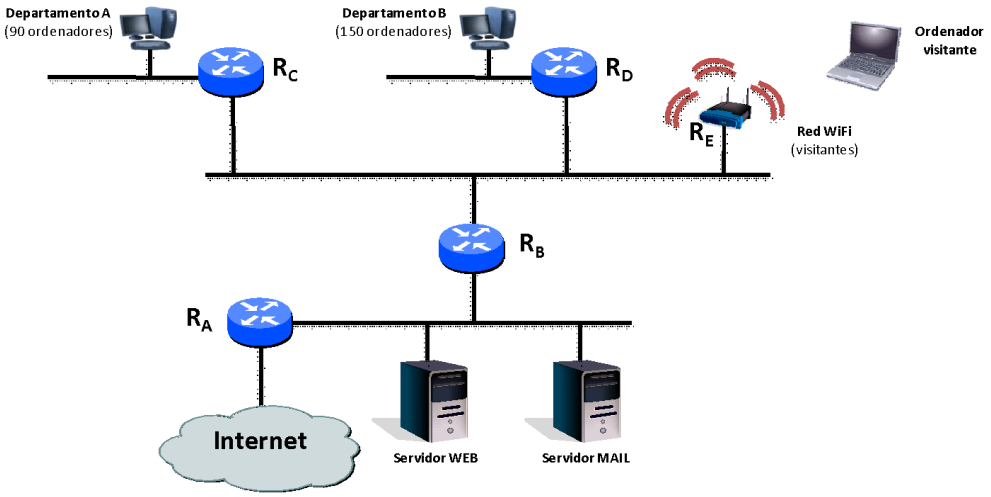
Destino	Máscara	Siguiente salto	Interfaz	
15.16.17.128	/26	R_1 (15.16.17.238)	eth1	Red A
15.16.17.192	/27	R_2 (15.16.17.242)	eth2	Red B
15.16.17.0	/25	R_3 (15.16.17.246)	eth3	Red C
15.16.17.224	/30	R_1 (15.16.17.238)	eth1	Subred R_1-R_2
15.16.17.228	/30	R_2 (15.16.17.242)	eth2	Subred R_2-R_3
15.16.17.232	/30	R_3 (15.16.17.246)	eth3	Subred R_1-R_3
150.150.150.0	/24	*	eth0	Red ISP (directa)
15.16.17.236	/30	*	eth1	Directa
15.16.17.240	/30	*	eth2	Directa
15.16.17.244	/30	*	eth3	Directa
default	/0	IP Gateway ISP	eth0	Internet

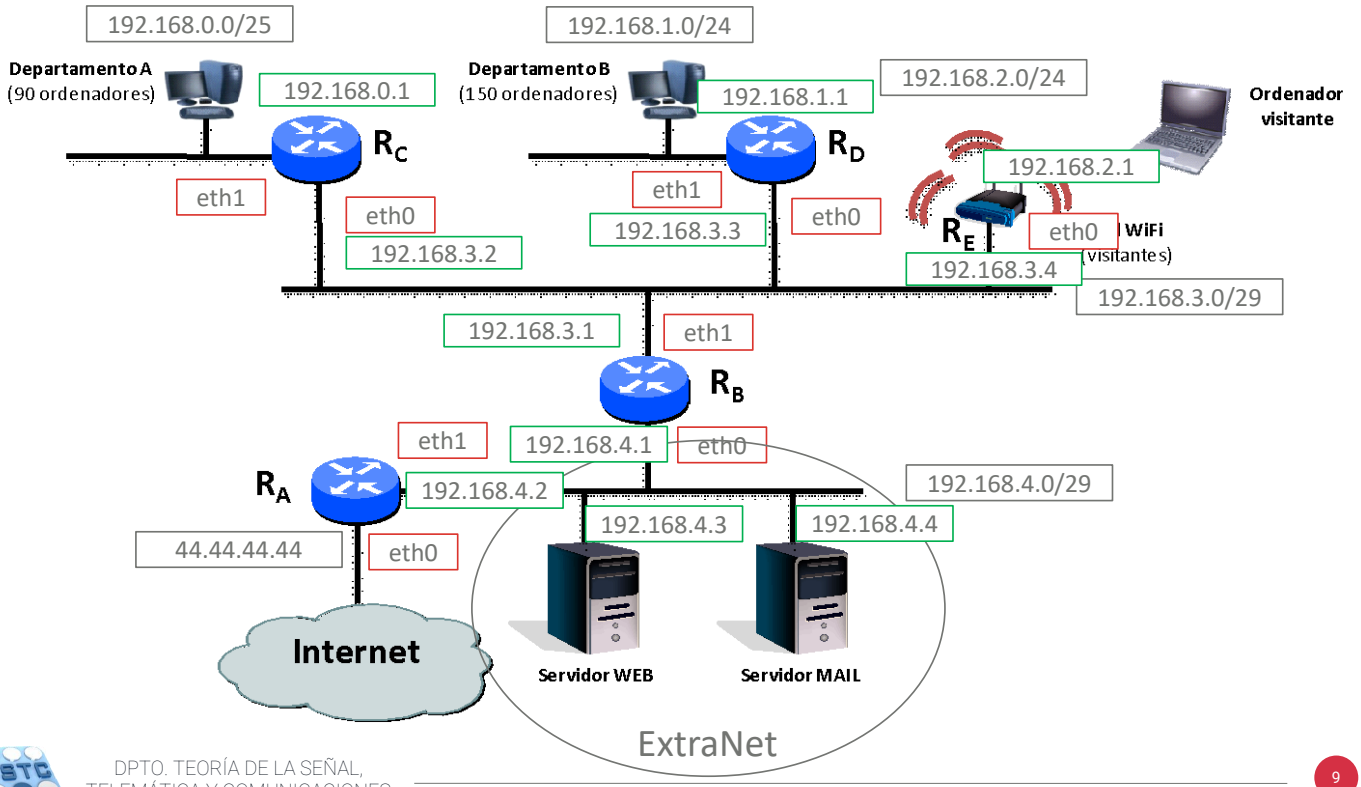
Nota: Los equipos de las redes A, B y C tienen como pasarela por defecto sus respectivos routers (R1, R2 y R3).

Ejercicio 3

ASIGNACIÓN DE DIRECCIONAMIENTO Y ENCAMINAMIENTO IP (Y NAT)

Dada la siguiente topología, que representa la red de una empresa, asigne direcciones IP a los diferentes equipos y redes, minimizando el número de entradas en las tablas de encaminamiento. El ISP sólo nos proporciona la dirección IP pública 44.44.44.44. Ajustar en lo posible las asignaciones al número de ordenadores.




192.168.0.0/25
Departamento A
(90 ordenadores)
192.168.0.1
RC
eth1
eth0
192.168.3.2
192.168.1.0/24
Departamento B
(150 ordenadores)
192.168.1.1
RD
eth1
192.168.3.3
eth0
192.168.2.0/24
Ordenador
visitante
192.168.2.1
RE
eth0
WiFi
(visitantes)
192.168.3.4
192.168.3.0/29
192.168.3.1
eth1
RB
eth1
192.168.4.1
eth0
RA
192.168.4.2
192.168.4.0/29
192.168.4.3
192.168.4.4
44.44.44.44
eth0
Internet
Servidor WEB
Servidor MAIL
ExtraNet

R_A

Destino	Máscara	Siguiente salto	Interfaz	
192.168.0.0	/22	R_B (192.168.4.1)	eth1	Agrupamiento .0.0 a .3.0
44.44.44.0	/24	*	eth0	Directa
192.168.4.0	/29	*	eth1	Directa
default	/0	IP Gateway ISP	eth0	Internet

R_B

Destino	Máscara	Siguiente salto	Interfaz	
192.168.0.0	/25	R_C (192.168.3.2)	eth1	Departamento A
192.168.1.0	/24	R_D (192.168.3.3)	eth1	Departamento B
192.168.2.0	/24	R_E (192.168.3.4)	eth1	Red de visitantes
192.168.4.0	/29	*	eth0	Directa
192.168.3.0	/29	*	eth1	Directa
default	/0	R_A (192.168.4.2)	eth0	Internet

R_C

Destino	Máscara	Siguiente salto	Interfaz	
192.168.3.0	/29	*	eth0	Directa
192.168.0.0	/25	*	eth1	Directa
default	/0	R_B (192.168.3.1)	eth0	Internet y otras subredes

R_D

Destino	Máscara	Siguiente salto	Interfaz	
192.168.3.0	/29	*	eth0	Directa
192.168.1.0	/24	*	eth1	Directa
Default	/0	R_B (192.168.3.1)	eth0	Internet y otras subredes

R_E

Destino	Máscara	Siguiente salto	Interfaz	
192.168.3.0	/29	*	eth0	Directa
192.168.2.0	/24	*	wlan0	Directa
default	/0	R_B (192.168.3.1)	eth0	Internet y otras subredes

Servidores WEB y MAIL

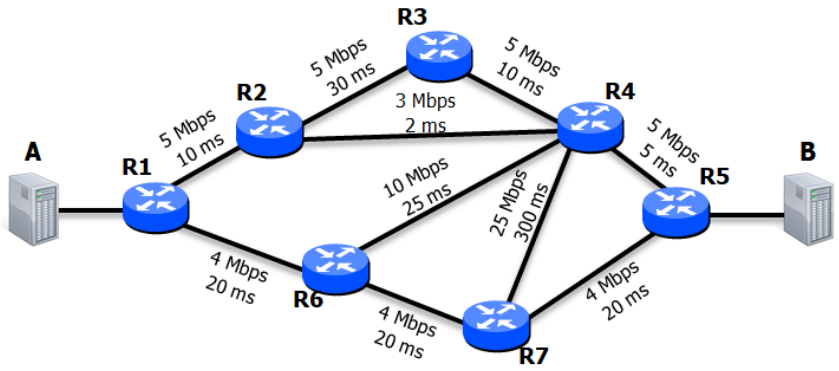
Destino	Máscara	Siguiente salto	Interfaz	
192.168.0.0	/22	R_B (192.168.4.1)	eth0	Agrupamiento .0.0 a .3.0
192.168.4.0	/29	*	eth0	Directa
default	/0	R_A (192.168.4.2)	eth0	Internet

Nota: Asumimos que para la red de visitantes es suficiente con /24 (192.168.2.0 /24). Asumimos que la IP de eth0 de RA es 44.44.44.44 /24, por lo que la subred correspondiente es 44.44.44.0 /24. Los ordenadores tienen su tabla de encaminamiento, aunque no se reporte.

Ejercicio 4

ENCAMINAMIENTO DINÁMICO

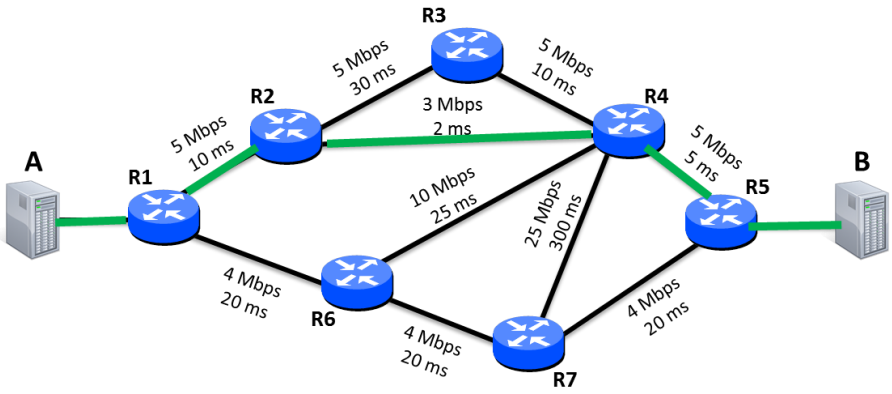
Dada la topología de la figura, explique qué ruta se utilizaría para mandar información entre el host A y el host B suponiendo:



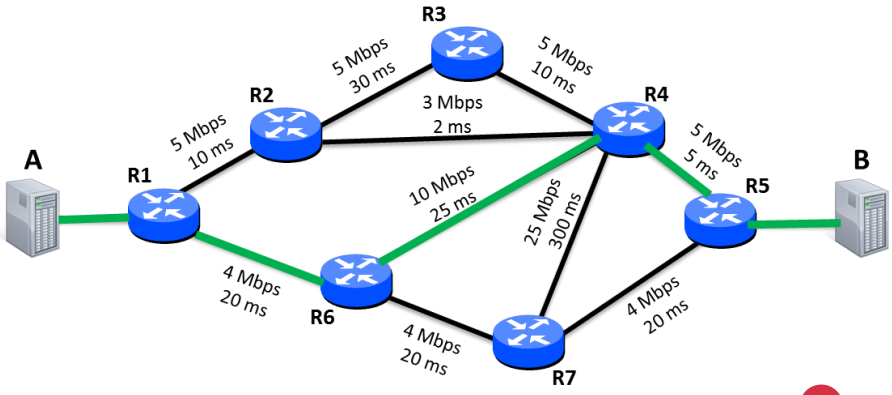
a) Que los routers implementan RIP. En el caso de que haya varias rutas posibles, explique cómo se elegiría la ruta a seguir en un caso real.

b) Que los routers implementan OSPF. En el caso de que haya varias rutas posibles, explique cómo se elegiría la ruta a seguir en un caso real.

RIP (menor número de saltos) → cualquier ruta con 3 saltos (por ejemplo, A-R1-R2-R4-R5-B)



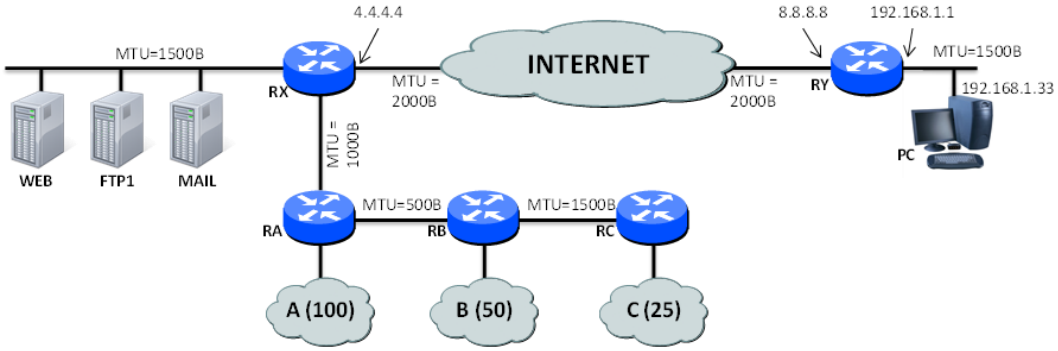
OSPF (menor coste en términos de 10^8/BW) → la ruta elegida sería A-R1-R6-R4-R5-B (coste = 55)



Ejercicio 5

FRAGMENTACIÓN (Y NAT)

La siguiente figura muestra la topología de red de una empresa conectada a Internet (parte izquierda), así como la red de un trabajador que se conecta desde casa (parte derecha). El ISP contratado por la empresa le asigna el rango 150.150.150.0/24.



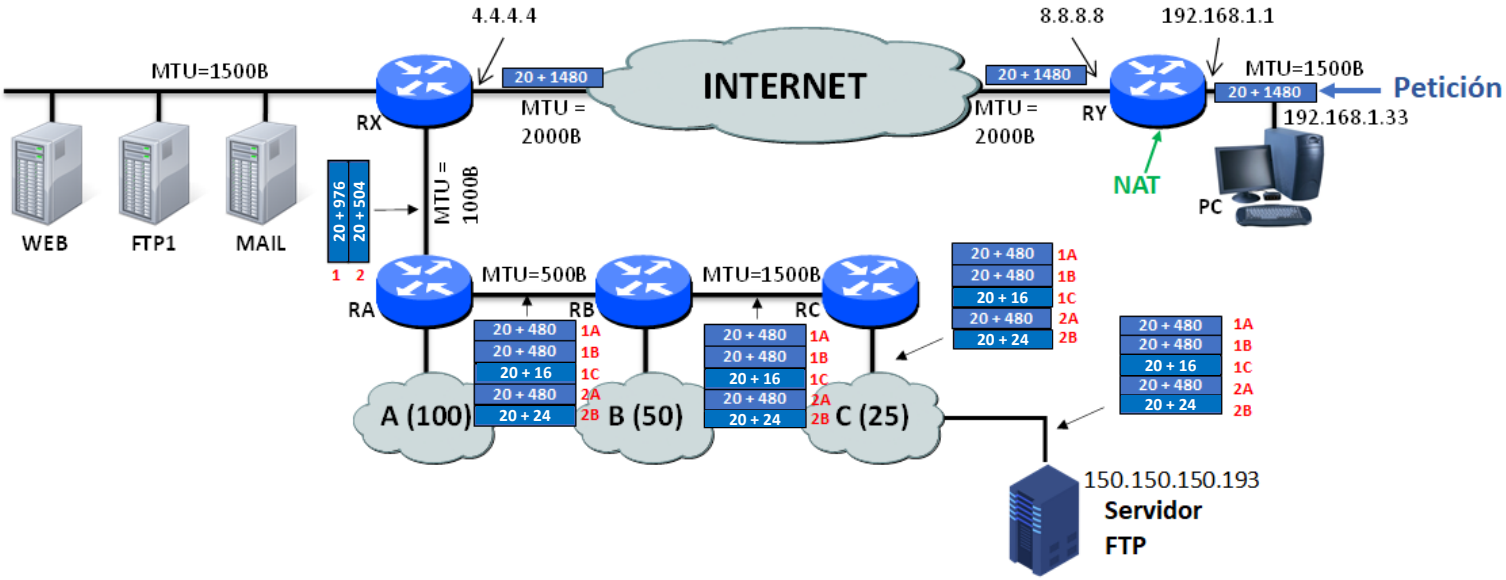
a) Asigne direcciones IP a todos los equipos de la empresa (incluyendo los routers) de forma que todas sean públicas.

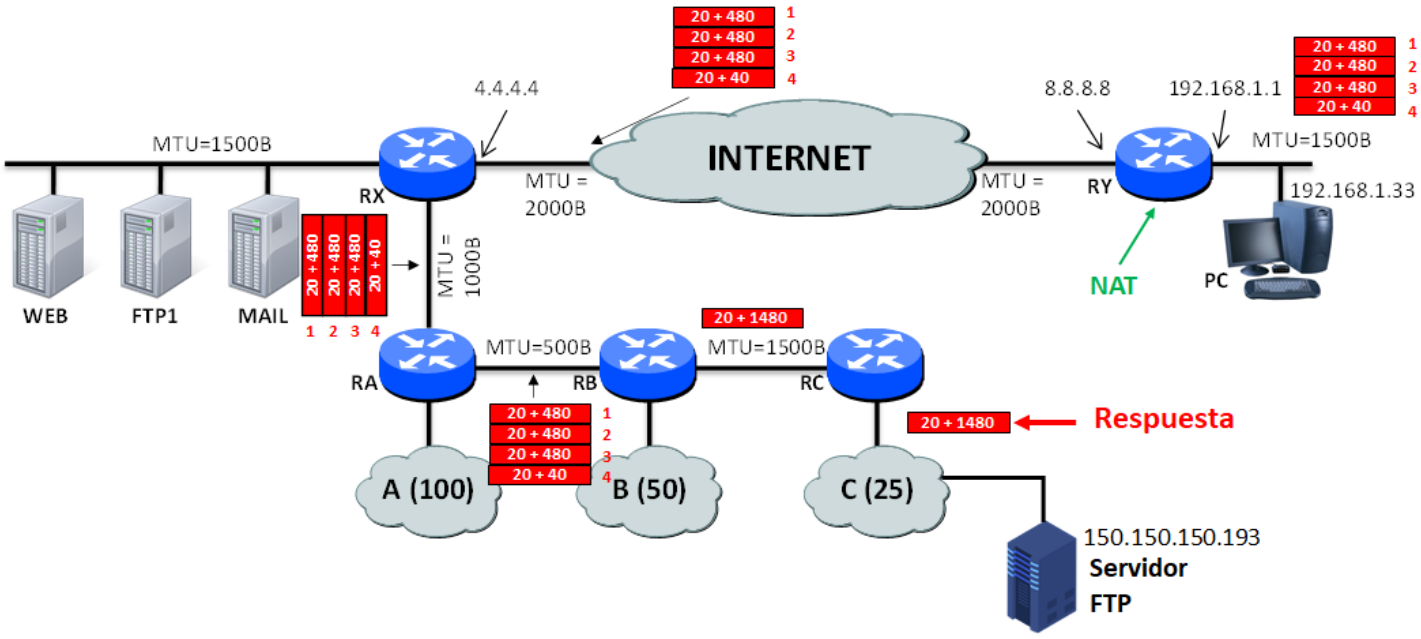
b) En la red C hay un servidor de FTP. El equipo PC (en casa del trabajador) quiere descargarse un fichero de este servidor. Suponga que se hace una petición con un datagrama IP y que se recibe una respuesta a dicha petición. Indique los valores de los diferentes campos (direcciones IP origen y destino; puerto origen y destino (21), identificador de paquete, offset, flag More Fragments). Suponga que tanto la petición como la respuesta tienen 1480 bytes de datos (incluyendo cabeceras de protocolos superiores, e.g. TCP). La cabecera IP tiene 20 bytes.

Red A: 150.150.150.0/25 → de .0 a .127
Red B: 150.150.150.128/26 → de .128 a .191
Red C: 150.150.150.192/27 → de .192 a .223

Red servidores (WEB, FTP1, MAIL): 150.150.150.224/29 → de .224 a .231

Red Rx-eth0 y Ra-eth2: 150.150.150.232/30
Red Ra-eth1 y Rb-eth2: 150.150.150.236/30
Red Rb-eth1 y Rc-eth1: 150.150.150.240/30


4.4.4.4
8.8.8.8
192.168.1.1
MTU=1500B
20 + 1480
INTERNET
20 + 1480
MTU=1500B
20 + 1480
Petición
MTU = 2000B
MTU = 2000B
RX
RY
192.168.1.33
NAT
PC
WEB
FTP1
MAIL
20 + 976
20 + 504
1 2
MTU = 1000B
RA
MTU=500B
RB
MTU=1500B
RC
20 + 480 1A
20 + 480 1B
20 + 16 1C
20 + 480 2A
20 + 24 2B
A (100)
B (50)
C (25)
150.150.150.193
Servidor
FTP


20 + 480
20 + 480
20 + 480
20 + 40
1
2
3
4
4.4.4.4
8.8.8.8
192.168.1.1
MTU=1500B
INTERNET
MTU = 2000B
MTU = 2000B
RX
RY
192.168.1.33
WEB
FTP1
MAIL
MTU = 1000B
NAT
PC
RA
RB
RC
MTU=500B
MTU=1500B
20 + 1480
Respuesta
A (100)
B (50)
C (25)
150.150.150.193
Servidor
FTP

Mensaje	Segmento	Tamaño (datos+ cabec.)	IP origen	IP destino	Puerto origen	Puerto destino	ID paquete	offset	Flag MF
Petición	PC → RY (MTU 1500B)	1480B + 20B	192.168.1.33	150.150.150.193	1037	21	1	0	0
Petición	RY → RX (MTU 2000B)	1480B + 20B	8.8.8.8	150.150.150.193	1037	21	1	0	0
Petición (fragm. 1)	RX → RA (MTU 1000B)	976B + 20B	8.8.8.8	150.150.150.193	1037	21	1	0	1
Petición (fragm. 2)	RX → RA (MTU 1000B)	504B + 20B	8.8.8.8	150.150.150.193	1037	21	1	122	0
Petición (fragm. 1A)	RA → RB (MTU 500B)	480B + 20B	8.8.8.8	150.150.150.193	1037	21	1	0	1
Petición (fragm. 1B)	RA → RB (MTU 500B)	480B + 20B	8.8.8.8	150.150.150.193	1037	21	1	60	1
Petición (fragm. 1C)	RA → RB (MTU 500B)	16B + 20B	8.8.8.8	150.150.150.193	1037	21	1	120	1
Petición (fragm. 2A)	RA → RB (MTU 500B)	480B + 20B	8.8.8.8	150.150.150.193	1037	21	1	122	1
Petición (fragm. 2B)	RA → RB (MTU 500B)	24B + 20B	8.8.8.8	150.150.150.193	1037	21	1	182	0
Se repiten los fragmentos 1A, 1B, 1C, 2A, 2B entre RB → RC y después entre RC → servidor FTP. Todos los campos serían iguales a las últimas 5 filas.									
Respuesta	Servidor → RC (MTU 1500B)	1480B + 20B	150.150.150.193	8.8.8.8	21	1037	1	0	0
Respuesta	RC → RB (MTU 1500B)	1480B + 20B	150.150.150.193	8.8.8.8	21	1037	1	0	0
Respuesta (fragm. 1)	RB → RA (MTU 1500B)	480B + 20B	150.150.150.193	8.8.8.8	21	1037	1	0	1
Respuesta (fragm. 2)	RB → RA (MTU 1500B)	480B + 20B	150.150.150.193	8.8.8.8	21	1037	1	60	1
Respuesta (fragm. 3)	RB → RA (MTU 1500B)	480B + 20B	150.150.150.193	8.8.8.8	21	1037	1	120	1
Respuesta (fragm. 4)	RB → RA (MTU 1500B)	40B + 20B	150.150.150.193	8.8.8.8	21	1037	1	180	0
Como el resto de los segmentos tienen una MTU mayor que 500B, ya no hay más fragmentaciones. Se repiten los fragmentos entre RA → RX y RX → RY, y ahí se desharía NAT (siguientes filas).									
Respuesta (fragm. 1)	RY → PC	480B + 20B	150.150.150.193	192.168.1.33	21	1037	1	0	1
Respuesta (fragm. 2)	RY → PC	480B + 20B	150.150.150.193	192.168.1.33	21	1037	1	60	1
Respuesta (fragm. 3)	RY → PC	480B + 20B	150.150.150.193	192.168.1.33	21	1037	1	120	1
Respuesta (fragm. 4)	RY → PC	40B + 20B	150.150.150.193	192.168.1.33	21	1037	1	180	0

2.4. Seminario 4 (Opcional)

Seminario 4

Creación de aplicaciones Cliente/Servidor

1. Objetivo

El objetivo de este seminario es introducir al alumno en los conceptos básicos para el desarrollo de aplicaciones cliente-servidor que utilizan sockets UDP y sockets TCP.

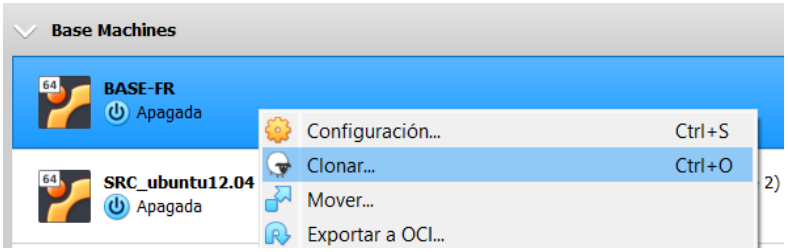
1.1. Información básica para la realización del seminario

En esta sección se ofrece la información básica y las referencias necesarias para llevar a cabo las tareas que se proponen en la práctica.

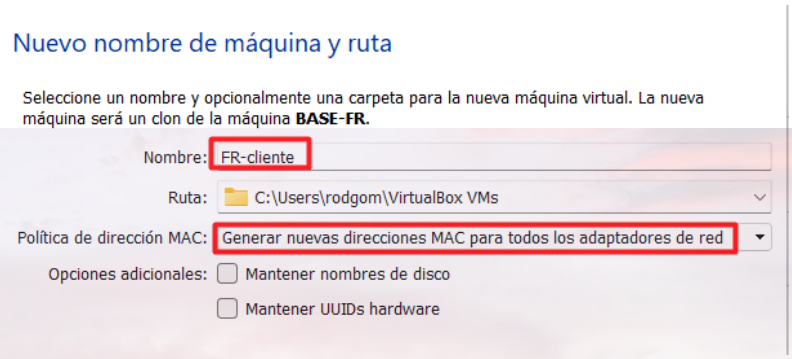
1.2. Acceso al sistema y elección del sistema operativo

Para la realización de este seminario no es necesario el uso de ningún sistema operativo en particular, ya que se realizará el lenguaje de programación *Python*, que es un lenguaje multiplataforma. No obstante, recomendamos la instalación del editor de código fuente *Visual Studio Code* [1] y la extensión de *Python*.

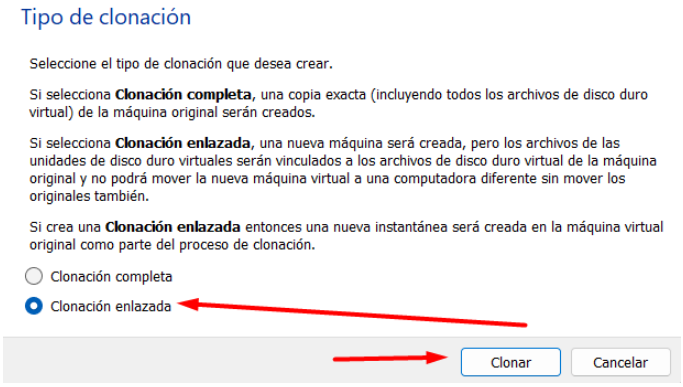
También es posible la utilización de la máquina virtual de seminarios de la asignatura de Fundamentos de Redes disponible en PRADO. Para ello, es necesario realizar una clonación de dicha máquina virtual. Para esto, se hace clic derecho sobre la máquina ya importada en Virtualbox y se escoge la opción Clonar.



Se puede cambiar el nuevo nombre de la máquina por FR-cliente/servidor, por ejemplo. Es muy importante que en este punto se seleccione la opción generar nuevas direcciones MAC para todos los adaptadores de red. Elegido esto se clica sobre Siguiente.



Por último, se escoge la opción clonación enlazada y se clica sobre Clonar.



Nota importante: Realizad la clonación clicando sobre el botón derecho desde la máquina original importada. Esa máquina será ahora la máquina base de esta clonación y no debe encenderse para minimizar los conflictos que esto pudiera generar. Sólo se utilizará la máquina clonada.

2. Códigos del cliente y servidor en UDP

Prepare los códigos tanto de este apartado como del siguiente para poder seguir la clase de seminarios con mayor fluidez. Los conceptos teóricos asociados serán repasados en la clase de seminarios.

Cread el archivo `clienteUPD.py` que contenga el siguiente código:

```
import socket
s_client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s_client.sendto(b'Hola clase', ('localhost',12345))
s_client.close()
```

Cread el archivo `servidorUPD.py` que contenga el siguiente código:

```
import socket
s_server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s_server.bind(('',12345))
data, clientaddr = s_server.recvfrom(4096)
s_server.close()
```

Estudie cada una de las líneas de código que componen ambos programas. Puede consultar para esto la siguiente documentación: https://docs.python.org/3/library/socket.html

Desde el servidor ejecute el programa con `py servidorUDP.py`
Desde el cliente ejecute el programa con `py clienteUDP.py`

Reto. ¿Qué habría que incluir en el código para que el servidor pudiera ver el contenido enviado por el cliente?

Reto. ¿Qué habría que incluir en el código para que el servidor pudiera responder al cliente "Bienvenido a clase"?

3. Códigos del cliente y servidor en TCP

Cread el archivo `servidorTCP.py` que contenga el siguiente código:

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("", 9999))
s.listen(1)
sc, addr = s.accept()
while True:
    recibido = sc.recv(1024)
    if recibido.decode() == "close":
        break
    print(str(addr[0]) + " dice: ", recibido.decode())
    sc.send(recibido)
print("Adios.")
sc.close()
s.close()
```

¿Qué indica el entero de la función *listen* del socket?

Cread el archivo `clienteTCP.py` que contenga el siguiente código:

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("localhost", 9999))
while True:
    mensaje = input("Mensaje a enviar >> ")
    s.send(mensaje.encode())
    if mensaje == "close":
        break
print("Adios.")
s.close()
```

Estudie cada una de las líneas de código que componen ambos programas. Puede consultar para esto la siguiente documentación: https://docs.python.org/3/library/socket.html

Desde el servidor ejecute el programa con `py servidorTCP.py`
Desde el cliente ejecute el programa con `py clienteTCP.py`
Debe escribir el mensaje a enviar al servidor.

¿Cuál es el mensaje que ha de escribir para que el programa del cliente finalice?

Reto. Rellene los huecos del fichero servidorWeb.py para que funcionen como un servidor web y se pueda acceder desde cualquier navegador con la url: *http://localhost:8080/* .

Reto. Montar un servidor que convierta en minúscula todos los mensajes que recibe del cliente y se reenvíe la conversión al cliente.

Reto. Montar un cliente/servidor resistente a fallos en la transmisión a nivel de aplicación. Con las siguientes características:

- El cliente envía un mensaje
- El servidor al recibirlo lo elimina con un 33% de probabilidad y si es así pide al cliente que lo reenvíe, si no, le dice que se ha recibido correctamente.
- El cliente reenvía el mensaje hasta que el servidor confirma que se ha recibido correctamente.

Para profundizar

Para estudiar en mayor profundidad los conceptos de cliente/servidor utilizando TCP o UDP se recomienda estudiar el capítulo 2 del libro *Computer Networking* [2].

Para ver cómo programar socket concurrentes con *Python* se recomienda el tutorial [3].

Bibliografía

[1] Visual Studio Code: https://code.visualstudio.com/
[2] James, Kurose, y Ross Keith. Computer Networking: A Top-Down Approach. Boston Munich, 2016. Capítulo 2.
[3] GeeksforGeeks. «Socket Programming with Multi-Threading in Python», 30 de septiembre de 2017. https://www.geeksforgeeks.org/socket-programming-multi-threading-python/.

2.5. Seminario 5

Fundamentos de Redes

Seminario 5: Resolución de problemas del Tema 3

Curso 2023/2024

Profesor: Jesús Minguillón, minguillon@ugr.es

UNIVERSIDAD DE GRANADA

DPTO. TEORÍA DE LA SEÑAL, TELEMÁTICA Y COMUNICACIONES

Ejercicio 1

Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Dadas dos entidades TCP (A y B) conectadas por una red cuya velocidad de transmisión es 100 Mbps, suponga segmentos de 1024 bytes y un RTT (Round Trip Time) constante de 2 mseg. Si A transmite masivamente datos a B ¿Cuánto tiempo tardará en transmitir 8 segmentos? Incluya el número de secuencia y de acuse en todos los segmentos TCP. Haga las suposiciones que estime necesarias.

Ejercicio 1

Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Dadas dos entidades TCP (A y B) conectadas por una red cuya velocidad de transmisión es 100 Mbps, suponga segmentos de 1024 bytes y un RTT (Round Trip Time) constante de 2 mseg. Si A transmite masivamente datos a B ¿Cuánto tiempo tardará en transmitir 8 segmentos? Incluya el número de secuencia y de acuse en todos los segmentos TCP. Haga las suposiciones que estime necesarias.

Evento	Acción del TCP receptor
Llegada ordenada de segmento, sin discontinuidad, todo lo anterior ya confirmado.	Retrasar ACK. Esperar recibir al siguiente segmento hasta 500 mseg. Si no llega, enviar ACK.
Llegada ordenada de segmento, sin discontinuidad, hay pendiente un ACK retrasado.	Inmediatamente enviar un único ACK acumulativo.
Llegada desordenada de segmento con núm. de secuen. mayor que el esperado, discontinuidad detectada.	Enviar un ACK duplicado, indicando el núm de secuen. del siguiente byte esperado.
Llegada de un segmento que completa una discontinuidad parcial o totalmente.	Confirmar ACK inmediatamente si el segmento comienza en el extremo inferior de la discontinuidad.

Ejercicio 1

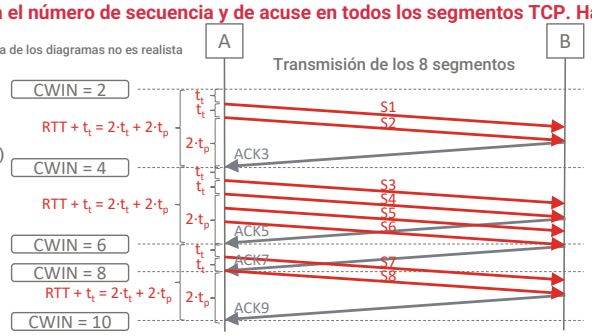
Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Dadas dos entidades TCP (A y B) conectadas por una red cuya velocidad de transmisión es 100 Mbps, suponga segmentos de 1024 bytes y un RTT (Round Trip Time) constante de 2 ms. Si A transmite masivamente datos a B ¿Cuánto tiempo tardará en transmitir 8 segmentos? Incluya el número de secuencia y de acuse en todos los segmentos TCP. Haga las suposiciones que estime necesarias.

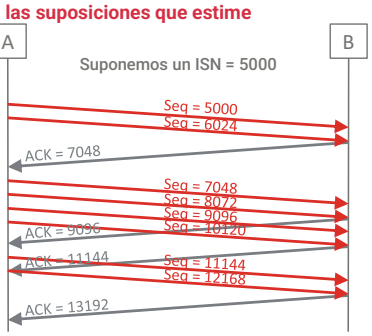
La escala de los diagramas no es realista

Datos:
V_t = 100 Mbps
Segmentos = 1024 B
RTT = 2 ms (constante)

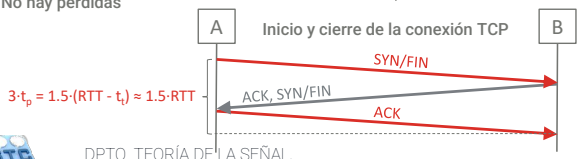
Suponemos:
t_t (ACK, SYN, FIN) ≈ 0
RTT simétrico
RTT = $t_t + 2 \cdot t_p$
RWIN → ∞
CWIN = 2
Inicio lento
No hay pérdidas



Transmisión de los 8 segmentos



Suponemos un ISN = 5000



t_t = 1024 B / 100 Mbps ≈ 0.082 ms

Si usamos la típica aproximación $t_{inicio} = t_{cierre} \approx 1.5 \cdot RTT$:
$t_{8seg} = t_{inicio} + t_{trans} + t_{cierre} \approx 1.5 \cdot RTT + 3 \cdot (RTT + t_t) + 1.5 \cdot RTT \approx$ **12.25 ms**

Si queremos afinar más, $t_{inicio} = t_{cierre} = 1.5 \cdot (RTT - t_t)$:
$t_{8seg} = t_{inicio} + t_{trans} + t_{cierre} = 1.5 \cdot (RTT - t_t) + 3 \cdot (RTT + t_t) + 1.5 \cdot (RTT - t_t) =$ **12 ms**

Ejercicio 2

Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Suponga el envío de un fichero grande sobre una conexión TCP y suponga que el RTT (tiempo de ida y vuelta) es constante.

a) Si CongWin es 1 MSS (tamaño del segmento) ¿cuánto tiempo como mínimo se necesitará para que CongWin sea 7 MSS? (suponga que no hay pérdidas y que no entra en la zona de prevención de congestión)

b) ¿Cuál será el tamaño de CogWin tras 6 RTTs?

c) Si CongWin es 101 MSS y está en la zona de prevención de la congestión ¿cuánto tiempo se necesitará para que CongWin sea 107 MSS?

d) ¿Cuál será el throughput medio tras 6 RTTs desde el inicio de la transmisión?

Ejercicio 2

Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Suponga el envío de un fichero grande sobre una conexión TCP y suponga que el RTT (tiempo de ida y vuelta) es constante.

a) Si CongWin es 1 MSS (tamaño del segmento) ¿cuánto tiempo como mínimo se necesitará para que CongWin sea 7 MSS? (suponga que no hay pérdidas y que no entra en la zona de prevención de congestión)

b) ¿Cuál será el tamaño de CogWin tras 6 RTTs?

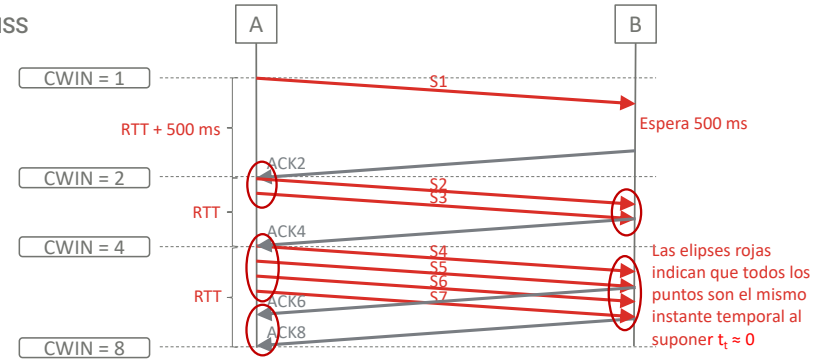
c) Si CongWin es 101 MSS y está en la zona de prevención de la congestión ¿cuánto tiempo se necesitará para que CongWin sea 107 MSS?

d) ¿Cuál será el throughput medio tras 6 RTTs desde el inicio de la transmisión?

a) Tiempo para pasar de CWIN = 1 MSS a CWIN = 7 MSS

Datos:
CWIN = 1 MSS
RTT constante
Inicio lento

Suponemos:
$t_t \approx 0$
RTT simétrico (mismo t_p de ida y vuelta)
$RTT = 2 \cdot t_p$
RWIN → ∞
No hay pérdidas



La escala del diagrama no es realista

$t_{CWIN=1\ MSS \rightarrow 7\ MSS} = 3 \cdot RTT + 500\ ms$

Ejercicio 2

Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Suponga el envío de un fichero grande sobre una conexión TCP y suponga que el RTT (tiempo de ida y vuelta) es constante.
a) Si CongWin es 1 MSS (tamaño del segmento) ¿cuánto tiempo como mínimo se necesitará para que CongWin sea 7 MSS? (suponga que no hay pérdidas y que no entra en la zona de prevención de congestión)
b) ¿Cuál será el tamaño de CogWin tras 6 RTTs?
c) Si CongWin es 101 MSS y está en la zona de prevención de la congestión ¿cuánto tiempo se necesitará para que CongWin sea 107 MSS?
d) ¿Cuál será el throughput medio tras 6 RTTs desde el inicio de la transmisión?

b) CWIN pasados 6 RTTs

Datos:
CWIN = 1 MSS
RTT constante
Inicio lento

Suponemos:
El primer RTT va con los 500 ms
Seguimos en inicio lento todo el rato
$t_t \approx 0$
RTT simétrico (mismo t_p de ida y vuelta)
$RTT = 2 \cdot t_p$
RWIN → ∞
No hay pérdidas

Cada RTT, se transmite (confirmaciones incluidas) una ventana entera, sea cual sea el tamaño de ventana (porque $t_t \approx 0$). En la zona de inicio lento, CWIN crece 1 MSS por segmento confirmado (i.e., se duplica el tamaño de la ventana cada vez que se confirma una ventana completa).

$CWIN_{t=6 \cdot RTT} = 2^6 =$ **64 MSS**

Ejercicio 2

Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Suponga el envío de un fichero grande sobre una conexión TCP y suponga que el RTT (tiempo de ida y vuelta) es constante.
a) Si CongWin es 1 MSS (tamaño del segmento) ¿cuánto tiempo como mínimo se necesitará para que CongWin sea 7 MSS? (suponga que no hay pérdidas y que no entra en la zona de prevención de congestión)
b) ¿Cuál será el tamaño de CogWin tras 6 RTTs?
c) Si CongWin es 101 MSS y está en la zona de prevención de la congestión ¿cuánto tiempo se necesitará para que CongWin sea 107 MSS?
d) ¿Cuál será el throughput medio tras 6 RTTs desde el inicio de la transmisión?

c) Tiempo para pasar de CWIN = 101 MSS a CWIN = 107 MSS

Datos:
CWIN = 101 MSS
RTT constante
Prevención de la congestión

Suponemos:
$t_t \approx 0$
RTT simétrico (mismo t_p de ida y vuelta)
$RTT = 2 \cdot t_p$
$RWIN \rightarrow \infty$
No hay pérdidas

Cada RTT, se transmite (confirmaciones incluidas) una ventana entera, sea cual sea el tamaño de ventana (porque $t_t \approx 0$). En la zona de prevención de la congestión, CWIN crece 1 MSS por ventana completa confirmada.

$t_{CWIN=101\ MSS \rightarrow 107\ MSS} = \mathbf{6 \cdot RTT}$

Ejercicio 2

Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Suponga el envío de un fichero grande sobre una conexión TCP y suponga que el RTT (tiempo de ida y vuelta) es constante.
a) Si CongWin es 1 MSS (tamaño del segmento) ¿cuánto tiempo como mínimo se necesitará para que CongWin sea 7 MSS? (suponga que no hay pérdidas y que no entra en la zona de prevención de congestión)
b) ¿Cuál será el tamaño de CogWin tras 6 RTTs?
c) Si CongWin es 101 MSS y está en la zona de prevención de la congestión ¿cuánto tiempo se necesitará para que CongWin sea 107 MSS?
d) ¿Cuál será el throughput medio tras 6 RTTs desde el inicio de la transmisión?

d) Throughput medio pasados 6 RTTs desde el inicio

Datos:
CWIN = 1 MSS
RTT constante
Inicio lento

Suponemos:
El primer RTT va con los 500 ms
Seguimos en inicio lento todo el rato
$t_t \approx 0$
RTT simétrico (mismo t_p de ida y vuelta)
$RTT = 2 \cdot t_p$
$RWIN \rightarrow \infty$
No hay pérdidas

Tras 1 RTT se ha transmitido 1 MSS (confirmación incluida), tras 2 RTTs se han transmitido 2 MSS más, tras 3 RTTs se han transmitido 4 MSS más, etc. Tras 6 RTTs, se han transmitido 1 + 2 + 4 + 8 + 16 + 32 = 63 MSS.

$Th_{t=6 \cdot RTT} = \mathbf{63\ MSS\ /\ (6 \cdot RTT + 500\ ms)}$

Ejercicio 3

Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Suponga dos entidades TCP A y B con la siguiente configuración: el valor inicial de la ventana de congestión es 3 MSS; el tamaño del buffer en recepción es 4 MSS; la aplicación receptora consume 1 MSS cada 30 milisegundos. Suponiendo que el round trip time (RTT) es de 10 milisegundos y que no hay pérdidas ni descartes de paquetes, ¿cuánto tiempo tarda la entidad A en enviar 8 segmentos TCP con datos a la entidad B? Considere despreciable el tiempo que se tarda en emitir los segmentos por los interfaces de red.

Ejercicio 3

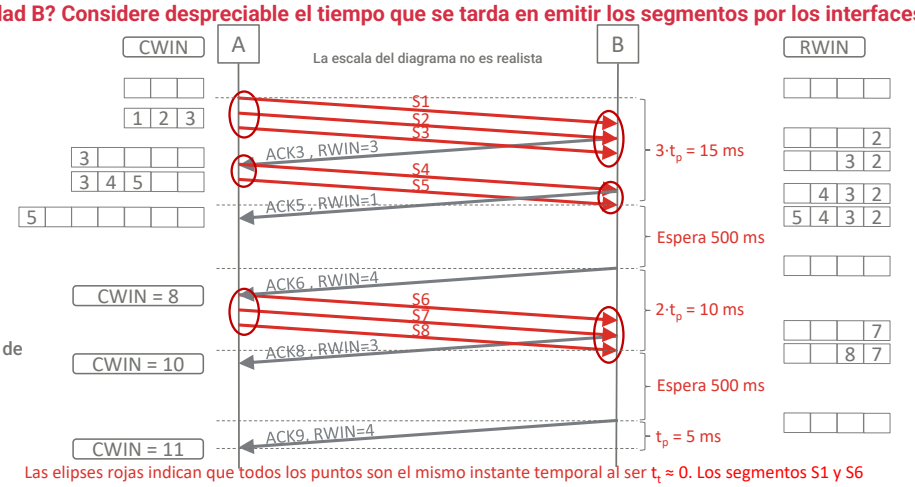
Ejercicio TCP: RTT, número de secuencia y acuse de recibo

Suponga dos entidades TCP A y B con la siguiente configuración: el valor inicial de la ventana de congestión es 3 MSS; el tamaño del buffer en recepción es 4 MSS; la aplicación receptora consume 1 MSS cada 30 milisegundos. Suponiendo que el round trip time (RTT) es de 10 milisegundos y que no hay pérdidas ni descartes de paquetes, ¿cuánto tiempo tarda la entidad A en enviar 8 segmentos TCP con datos a la entidad B? Considere despreciable el tiempo que se tarda en emitir los segmentos por los interfaces de red.

Datos:
CWIN = 3 MSS
RWIN = 4 MSS
RTT = 10 ms (constante)
Se consume 1 MSS en B cada 30 ms
$t_t \approx 0$

Suponemos:
RTT simétrico (mismo t_p de ida y vuelta)
RTT = $2 \cdot t_p$
Inicio lento
No hay pérdidas

Aquí no vamos a tener en cuenta tiempos de inicio y cierre de la conexión. Además, se considera que el proceso que toma los segmentos del buffer del receptor está disponible al inicio, por lo que el primer segmento que llega directamente pasa al proceso (no se almacena en memoria).



Las elipses rojas indican que todos los puntos son el mismo instante temporal al ser $t_t \approx 0$. Los segmentos S1 y S6 entran directamente al proceso que va consumiendo segmentos. La ventana de envío será siempre el mínimo entre CWIN y RWIN (restando al tamaño de ambas los segmentos enviados y no confirmados)

t_{8seg} = **1030 ms**

Ejercicio 4

Ejercicio Teórico

Explique las diferencias en objetivos y funcionamiento entre el control de flujo y el control de congestión en TCP. ¿Cómo ayudan los routers en el control de congestión de TCP? ¿Y en el control de flujo?

Ejercicio 4

Ejercicio Teórico

Explique las diferencias en objetivos y funcionamiento entre el control de flujo y el control de congestión en TCP. ¿Cómo ayudan los routers en el control de congestión de TCP? ¿Y en el control de flujo?

- **Control de flujo TCP.** Su objetivo es garantizar que el flujo que sale del emisor sea entregado, a la aplicación que hay por encima del TCP receptor, sin pérdidas, sin discontinuidades y en orden. El control de flujo permite al receptor de una transmisión controlar la tasa de envíos del emisor, para evitar que éste lo sature (llenando su buffer, por ejemplo) y pudieran llegar a perderse segmentos al tener que descartarse en la recepción. Este control lo realiza, por tanto, el receptor en una transmisión, por medio del campo WIN de la cabecera de TCP (es la ventana ofertada del receptor al emisor, la RWIN del ejercicio anterior), en el que se especifica el número de bytes que podría transmitir el emisor. Si este número menos el número de segmentos sin confirmar es cero, el emisor queda "bloqueado" (se queda sin ventana de envío para seguir transmitiendo segmentos) hasta recibir un nuevo valor de WIN que le permita continuar transmitiendo segmentos. El receptor se encarga de entregar todo ordenado y sin discontinuidades a la aplicación que tiene por encima. Incluso si hay alguna discontinuidad por pérdidas durante la transmisión, el receptor puede corregirla dejando el hueco correspondiente en su buffer (mientras sigue guardando segmentos posteriores), solicitando la retransmisión del segmento perdido y colocándolo en dicho hueco cuando llegue, entregando así todo completo y en orden a la aplicación que tiene por encima.

- **Control de congestión TCP.** Se lleva a cabo por el emisor y su objetivo es intentar evitar la congestión de la red durante la transmisión que, de producirse, conllevaría la pérdida de segmentos. Dicho control se hace aplicando un mecanismo de gestión de los segmentos a enviar, en el que se considera una ventana de congestión (CWIN) que, junto con la ventana ofertada por el receptor (RWIN), define la ventana de envío disponible, es decir, cuántos segmentos pueden ser transmitidos consecutivamente sin recibir confirmaciones. En general, va aumentando mientras no haya muestras de congestión en la red. Si se produce un evento relacionado con la congestión, como la pérdida de algún segmento, la ventana de envíos se reducirá en una proporción. El control de congestión lo realiza el emisor en una transmisión.

- **Routers.** Los routers no incorporan funcionalidades para realizar un control de flujo ni de congestión a nivel de capa de transporte. Si bien se podría considerar que pueden ayudar al control de congestión, mediante los mecanismos de conformación de tráfico (se controla la tasa de generación de paquetes sobre la red) o con los métodos que implementan para definir perfiles de tráfico con determinada prioridad, asociados a Calidad de Servicio (Quality of Service, QoS). Por ejemplo, se puede dar prioridad a transmisiones asociadas a streaming de vídeo sobre la red que gestione el router, evitando saturar la misma con otro tipo de tráfico.

2.6. Seminario 6

Fundamentos de Redes (Grado en Ingeniería Informática y Doble Grado en Ingeniería Informática y Matemáticas)

Seminario 6: Resolución de problemas de los Temas 4 y 5

Curso 2023/2024

Profesora: Julia Caleya Sánchez, jcaleyas@ugr.es

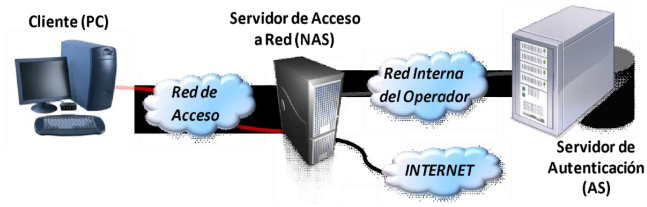
UNIVERSIDAD DE GRANADA

DPTO. TEORÍA DE LA SEÑAL, TELEMÁTICA Y COMUNICACIONES

Ejercicio 1

SEGURIDAD

La figura y mensajes siguientes describen un hipotético protocolo utilizado para permitir el acceso de un cliente a Internet a través de un Servidor de Acceso a Red (NAS). El Servidor de Autenticación (AS) guarda en una base de datos las claves secretas que se solicita a los usuarios para poder acceder a Internet.



Aceptadas la disponibilidad y validez de las claves públicas involucradas en base a la existencia de una entidad superior confiable, responda razonadamente:

a) **¿Qué servicios de seguridad se proporcionan en el protocolo descrito?**

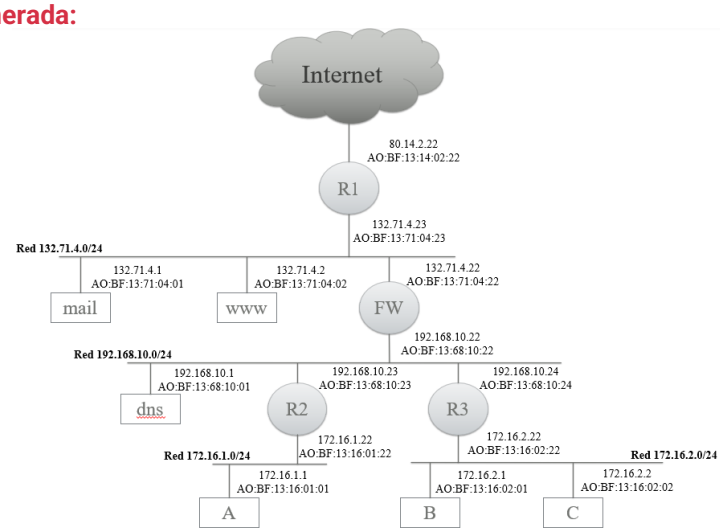
b) **¿Qué debilidades presenta el esquema propuesto? En su caso, ¿cómo podrían evitarse?**

PC -> NAS: $Kpub_{NAS}$(petición acceso + usuario)
NAS -> PC: desafío
PC -> NAS: $Kpub_{NAS}$(MD5(usuario + K_{PC-AS} + desafío))
NAS -> AS: petición autenticación + usuario + desafío + MD5(usuario + K_{AS-PC} + desafío)
AS -> NAS: petición aceptada + $Kses_{PC-NAS}$ + K_{PC-AS}($Kses_{PC-NAS}$) o petición rechazada
NAS -> PC: $Kpriv_{NAS}$(petición aceptada + K_{PC-AS}($Kses_{PC-NAS}$)) o $Kpriv_{NAS}$(petición rechazada)
PC -> NAS: $Kses_{PC-NAS}$(datos a enviar)
NAS -> Internet: datos a enviar
Internet -> NAS: datos de respuesta
NAS -> PC: $Kses_{PC-NAS}$ (datos de respuesta)
Siendo:
$Kpub_X$: cifrado con la clave pública de X
$Kpriv_X$: cifrado con la clave privada de X
K_{X-Y}: la clave secreta entre X e Y
$Kses_{X-Y}$: la clave secreta de sesión entre X e Y
MD5 una función hash

Ejercicio 2

DNS Y SMTP

Dada la topología adjunta correspondiente a una red corporativa, en la que se especifican tanto las direcciones IP como las MAC de cada uno de los dispositivos que la forman, analice el tráfico generado al hacer un acceso de correo electrónico desde el host "C" al servidor "mail", especificando en una tabla, y para cada trama Ethernet generada:



a) **Las direcciones hardware (físicas) origen y destino.**

b) **Las direcciones IP origen y destino contenidas en el paquete IP encapsulado.**

c) **En su caso, los puertos origen y destino de la unidad de datos del protocolo (PDU) de transporte, así como los flags activos y campos de secuencia y ACK.**

d) **El tipo de mensaje de que se trata.**

NOTA: suponga todas las tablas ARP son conocidas y, por simplicidad utilice sólo los dos últimos de los 6 octetos de las direcciones físicas de las NIC (interfaces o tarjetas de red).

Ejercicio 3

VELOCIDAD DE TRANSMISIÓN EN SERVICIOS TELEMÁTICOS

Una sucursal con 50 empleados en Granada tiene una red interna basada en *FastEthernet* (100 Mbps) que se conecta a Internet con una red de acceso ADSL de 0,5 Mbps de subida y 1,5 Mbps de bajada. Cada empleado, en el desempeño de su trabajo, realiza un promedio de 2000 solicitudes de información a la hora a un servidor de Base de Datos ubicado en la central del banco, en Madrid, donde cada solicitud supone el envío por parte del servidor de un promedio de 10 registros de 1 KB cada uno. Adicionalmente, la modificación de datos tras algunas de estas solicitudes supone el envío promedio de 100 actualizaciones, de 10 registros de media, a la hora desde la sucursal al servidor. El resto de los servicios telemáticos se restringe. Calcule el promedio de la velocidad de transmisión requerida. ¿Es la velocidad del enlace de acceso suficiente?

Ejercicio 4

HTTP

Compare el rendimiento en términos temporales de HTTP persistente y no persistente considerando los siguientes parámetros:

- **Descarga de una página web con 10 objetos incrustados**
- **Tiempo de Establecimiento de conexión TCP: 5 ms**
- **Tiempo de Cierre de conexión TCP: 5 ms**
- **Tiempo de solicitud HTTP: 2 ms**
- **Tiempo de respuesta HTTP (página web u objeto): 10 ms**

Ejercicio 5

ESPECIFICACIONES PARA APLICACIONES TELEMÁTICAS

Discuta las características de las siguientes aplicaciones en términos de su tolerancia a la pérdida de datos, los requisitos temporales, la necesidad de rendimiento mínimo y la seguridad:

- **La telefonía móvil.**
- **WhatsApp.**
- **YouTube.**
- **Spotify.**
- **Comercio electrónico.**

3 Referencias

- Diapositivas de clase.
- Wuolah.