



# REDES DE COMPUTADORES

Grado en Ingeniería en Informática

Curso académico 2016/2017

## Práctica 4. Encaminamiento IP avanzado

### CONTENIDOS

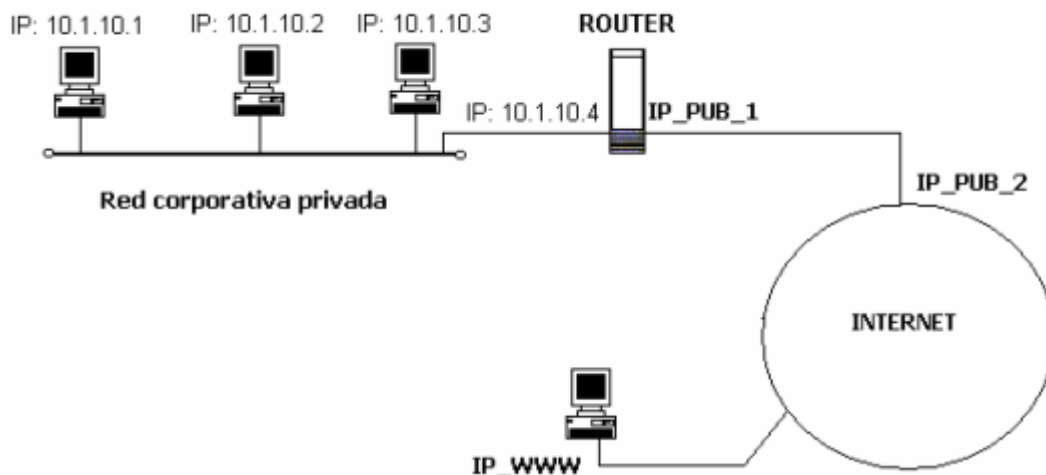
1. OBJETIVOS
2. DIRECCIONAMIENTO IP PRIVADO. MECANISMOS NAT/PAT
3. CONFIGURACIÓN DE LOS PARÁMETROS DE RED EN LINUX
4. HERRAMIENTAS DE RED EN LINUX
5. LISTAS DE CONTROL DE ACCESO (ACLs)
6. ACCESO REMOTO A UNA RED EMPLEANDO CONEXIONES VPN
7. CUESTIONES A REALIZAR
8. DOCUMENTACIÓN COMPLEMENTARIA

#### 1. Objetivos

- Analizar el mecanismo **NAT** (Traducción de Direcciones de Red) para proporcionar conectividad entre una red IP privada (LAN con direccionamiento IP privado) y una red IP pública (Internet).
- Conocer el funcionamiento de una red TCP/IP bajo el sistema operativo Linux.
- Conocer el manejo de herramientas de gestión de red en Linux.
- Conocer el funcionamiento de una lista de control de acceso (ACL – Access Control List).
- Conocer el funcionamiento de un acceso remoto mediante una VPN con PPTP.
- Configurar ACL's en un router Cisco y verificar su funcionamiento.

#### 2. Direccionamiento IP privado. Mecanismos NAT/PAT

El mecanismo NAT (Traducción de Direcciones de Red) tiene como objetivo conseguir que las estaciones de una red con direccionamiento IP privado puedan tener conectividad con Internet. Este mecanismo se describe en el documento RFC 1631 y se basa en un escenario como el que se presenta en la siguiente figura.



**Figura 1. Escenario de funcionamiento para NAT**

El **direccionamiento IP privado** son un conjunto de redes IP que no están asignadas a ningún equipo de Internet. En concreto, las redes IP privadas existentes son:

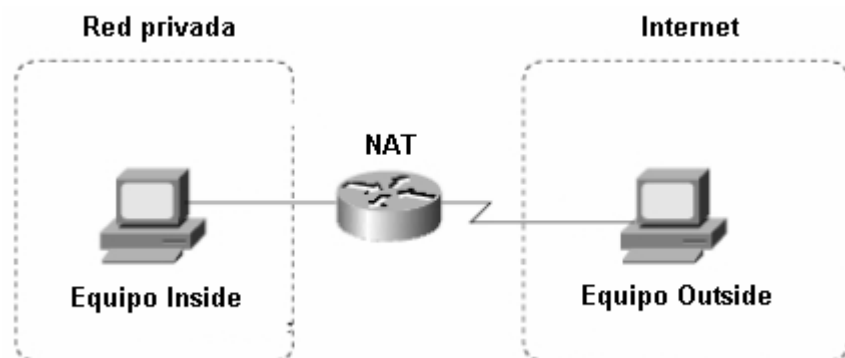
- a) 10.0.0.0/8
- b) 172.16.0.0/16 - 172.31.0.0/16
- c) 192.168.0.0/24 - 192.168.255.0/24

El resto de direcciones IP disponibles se denominan direcciones **IP públicas** y se caracterizan porque los routers de Internet son capaces de encaminar paquetes a esos destinos (además de que son direcciones únicas en Internet).

Un paquete IP con destino una dirección IP privada es rechazado por los routers del *backbone* (troncal) de Internet. El empleo del direccionamiento IP privado permite poder asignar direcciones IP dentro de una red privada sin restricciones ni costes, ya que disponer de una dirección IP pública (o legal) tiene un coste económico asociado.

El problema surge cuando se necesita conectar estas redes privadas con Internet. La conexión se realiza empleando un router que dispone de una o varias direcciones IP públicas asociadas. El router debe transformar los paquetes procedentes de la red privada modificando la dirección IP origen por las direcciones IP públicas que tiene asociadas.

Este mecanismo, Traducción de Direcciones de Red (NAT – *Network Address Translator*), precisa de una determinada nomenclatura.



**Figura 2. Nomenclatura en el mecanismo de NAT**

El router que implementa el mecanismo de NAT separa dos redes: **la red *inside*** y **la red *outside***. La red *inside* se corresponde con la red privada que tiene direccionamiento privado. La red *outside* es la red con direccionamiento público (Internet).

Además, se introducen los términos ***local*** y ***global***. El término *local* hace referencia a la red interna y el término *global* hace referencia a Internet. Así, pueden definirse los siguientes tipos de direcciones IP:


**INSIDE LOCAL:** Direcciones IP de las máquinas de la red privada en la red privada.

**INSIDE GLOBAL:** Direcciones IP de las máquinas de la red privada en Internet.

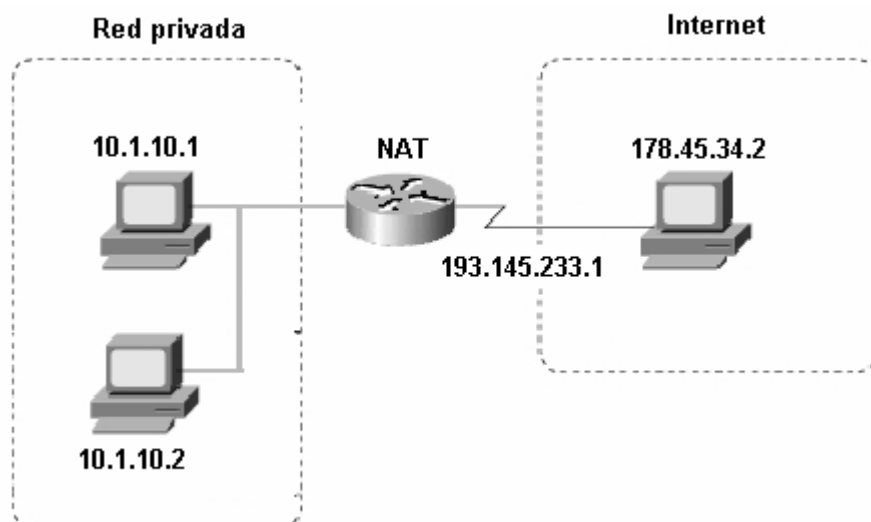
**OUTSIDE LOCAL:** Direcciones IP de las máquinas de Internet en la red privada.

**OUTSIDE GLOBAL:** Direcciones IP de las máquinas de Internet en Internet.

Con esta nomenclatura, el router que realiza NAT tiene que transformar las cabeceras IP's de los paquetes de la siguiente forma:

IP origen			IP destino			
INSIDE LOCAL	OUTSIDE LOCAL	DATOS		INSIDE GLOBAL	OUTSIDE GLOBAL	DATOS
OUTSIDE LOCAL	INSIDE LOCAL	DATOS		OUTSIDE GLOBAL	INSIDE GLOBAL	DATOS

Para entender mejor el funcionamiento de esta traducción, considérese el siguiente escenario de funcionamiento.



**Figura 3. Ejemplo de funcionamiento de NAT**

En el escenario de funcionamiento de la figura 3 las direcciones IP de la traducción de NAT serán:

**INSIDE LOCAL:** 10.1.10.1, 10.1.10.2

**INSIDE GLOBAL:** 193.145.233.1

**OUTSIDE LOCAL:** 178.45.34.2 (las direcciones IP de las máquinas de Internet son las mismas en la red privada que en Internet).

**OUTSIDE GLOBAL:** 178.45.34.2

Así, la traducción a realizar será:

IP origen	IP destino		IP origen	IP destino	
10.1.10.1	178.45.34.2	DATOS	193.145.233.1	178.45.34.2	DATOS
178.45.34.2	10.1.10.1	DATOS	178.45.34.2	193.145.233.1	DATOS

Hay que notar que los paquetes son traducidos en los dos sentidos de la comunicación.

Sin embargo el mecanismo de traducción indicando anteriormente tiene un grave problema, y es que será válido **sólo si hay UNA máquina de la red privada intercambiando datos con un determinado destino de Internet**. Si las estaciones 10.1.10.1 y 10.1.10.2 intercambian información con el mismo destino 178.45.34.2, las traducciones son incapaces de realizarse en el sentido Internet - red privada.

IP origen	IP destino		IP origen	IP destino	
10.1.10.1	178.45.34.2	DATOS	193.145.233.1	178.45.34.2	DATOS
10.1.10.2	178.45.34.2	DATOS	193.145.233.1	178.45.34.2	DATOS

El router es incapaz de conocer a qué máquina de la red interna tiene que entregar el paquete procedente del equipo de Internet.

Para solucionar este problema sería necesario tener una dirección IP pública para cada máquina de la red interna, por lo que no sería operativo. Sin embargo, ya que la mayor parte de las comunicaciones se fundamentan en la capa de transporte, pueden emplearse los números de puerto de las conexiones de transporte para identificar máquinas de la red interna empleando una única dirección IP pública en el router.

PAT (*Port Address Translation*) permite complementar a NAT para conseguir que una sola dirección **inside global** pueda ser compartida por varias estaciones de la red privada.

Al emplear PAT, se mantienen los mismos números de puerto de origen en la traducción de NAT, y será posible identificar máquinas distintas.

Supóngase que en el escenario de la figura 3, la estación 178.45.34.2 es un servidor web, y las dos estaciones de la red privada intentan acceder a él. La traducción quedaría en la forma:

IP origen	IP destino	P. origen	P. destino		IP origen	IP destino	P. origen	P. destino	
10.1.10.1	178.45.34.2	1075	80	DATOS	193.145.233.1	178.45.34.2	1075	80	DATOS
10.1.10.2	178.45.34.2	1090	80	DATOS	193.145.233.1	178.45.34.2	1090	80	DATOS

De esta forma puede emplearse el número de puerto de origen para identificar a qué máquina de la red interna está asociado el paquete. Existe un caso particular y es cuando el puerto origen empleado en los dos equipos de la red interna son los mismos (esto ocurre frecuentemente, ya que los números de puerto origen se generan secuencialmente por parte del sistema operativo de los equipos). En este caso, el router modifica el número de puerto origen en el paquete traducido, empleando uno que no esté siendo usado por otra traducción.

IP origen	IP destino	P. origen	P. destino		IP origen	IP destino	P. origen	P. destino	
10.1.10.1	178.45.34.2	1075	80	DATOS	193.145.233.1	178.45.34.2	1075	80	DATOS
10.1.10.2	178.45.34.2	1075	80	DATOS	193.145.233.1	178.45.34.2	1076	80	DATOS

La información acerca de las traducciones que realiza el router se almacena en su memoria, empleando una tabla denominada **tabla de traducciones**. Esta tabla tiene el siguiente formato:

<i>Inside global</i>	<i>Inside local</i>	<i>Outside local</i>	<i>Outside global</i>
193.145.233.1:1075	10.1.10.1:1075	178.45.34.2:80	178.45.34.2:80
193.145.233.1:1076	10.1.10.2:1075	178.45.34.2:80	178.45.34.2:80

Esta tabla puede consultarse en un router Cisco con el comando IOS **"sh ip nat translations"**.

El mecanismo NAT/PAT sólo es válido para el tráfico TCP y UDP. Sin embargo, el sistema operativo IOS de Cisco está desarrollado para permitir traducir tráfico que no emplee la capa de transporte (GRE, ICMP, etc.). El mecanismo de traducción en estos casos no está normalizado y es cada fabricante de hardware el que determina qué protocolos soporta en la traducción y cómo la realiza.

### 3. Configuración de los parámetros de red en Linux

Es conocido que en el sistema operativo Linux existen diferentes grupos de usuarios, y básicamente los podemos clasificar en dos grupos: aquellos que tienen privilegios de administración y los que no. De forma genérica, en Linux existe un único usuario con privilegios de administración que se denomina **root**. Accediendo a una estación Linux como usuario *root* tenemos acceso a todas las funcionalidades de configuración de red, herramientas de monitorización y gestión de red, etc.

El comando del sistema operativo que nos permite configurar un adaptador de red es **ifconfig**. El *kernel* de Linux (empleando los módulos o *drivers* de que disponga) reconoce los diferentes dispositivos de red disponibles en el equipo y los identifica con un nombre dependiendo del tipo de adaptador de red. Por ejemplo, los adaptadores de red Ethernet se denominan **eth**, los adaptadores de red Token Ring **tr**, los adaptadores de red de tipo punto a punto **ppp**, etc. Además, si el equipo dispone de más de un adaptador de red de un mismo tipo (por ejemplo en un router) se numeran sucesivamente: eth0, eth1, eth2,.....

Por ejemplo, para ver la configuración de los diferentes adaptadores de red de un equipo el alumno ha de ejecutar el siguiente comando:

```
usuario@host:~> /sbin/ifconfig
```

```
eth0    Link encap:Ethernet HWaddr 00:04:76:A4:35:D0
        inet addr:10.1.100.x Bcast:10.1.255.255 Mask:255.255.0.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:716484 errors:0 dropped:0 overruns:1 frame:0
        TX packets:256092 errors:0 dropped:0 overruns:0 carrier:0
        collisions:14480 txqueuelen:100
        RX bytes:159213386 (151.8 Mb) TX bytes:222387323 (212.0 Mb)
        Interrupt:10 Base address:0x7c80

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:181 errors:0 dropped:0 overruns:0 frame:0
        TX packets:181 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:10959 (10.7 Kb) TX bytes:10959 (10.7 Kb)
```

Para un equipo que disponga de un interfaz Ethernet se mostrará la dirección IP, máscara de red, dirección de broadcast, cantidad de paquetes transmitidos desde la activación del interfaz, paquetes recibidos, descartes de paquetes, errores, colisiones, etc. Además, el sistema operativo crea un dispositivo de red virtual denominado interfaz **loopback (lo)**. Este interfaz no está asociado a ningún interfaz de red físico pero se comporta como tal, permitiendo el funcionamiento de la arquitectura de red TCP/IP en la máquina.

Para configurar un interfaz ethernet con la dirección IP 172.20.43.221 en la subred 172.20.43.192/26 ha de emplearse el comando *ifconfig* con los parámetros:

```
usuario@host:~> /sbin/ifconfig eth0 172.20.43.221 netmask 255.255.255.192
```

Una vez configurado el interfaz, el sistema operativo añade una entrada en la tabla de rutas correspondiente a la red 172.20.43.192. Para visualizar la tabla de rutas del equipo se emplea el comando **netstat -rn**.

Destino	Máscara	Puerta de enlace	Interfaz
172.20.43.192	255.255.255.192	0.0.0.0	eth0

Para finalizar la configuración de la red es necesario especificar la puerta de enlace por defecto. Para ello se emplea el comando **route**.

```
usuario@host:~> /sbin/route add default gw 172.20.43.230
```

De esta forma, la tabla de rutas queda en la forma:

```
usuario@host:~> netstat -rn
```

Destino	Máscara	Puerta de enlace	Interfaz
172.20.43.192	255.255.255.192	0.0.0.0	eth0
0.0.0.0	0.0.0.0	172.20.43.230	eth0

Para obtener más información acerca de los comandos ifconfig, route y netstat se emplea el comando de ayuda **man** del sistema operativo (**man ifconfig**, **man route**, **man netstat**).

## 4. Herramientas de red en Linux

### 4.1 Ping

Como ya se mencionó en la práctica 2, la aplicación ping permite el envío de mensajes ICMP *Echo Request* y la recepción de mensajes ICMP *Echo Reply*. En Linux la aplicación ping tiene unas opciones diferentes de las estudiadas para Windows en la práctica 2. Las opciones de ping más relevantes son (dependiendo de la versión de Linux empleada):

**-c x** Se envían x mensajes *echo request* y se espera recibir x mensajes *echo reply*.

**-s l** Se envían paquetes *echo* con l bytes de datos.

**-t ttl** Se envían paquetes *echo request* con el campo tiempo de vida igual al valor ttl.

**-D** Se envían paquetes echo con el bit don't fragment de la cabecera IP activado.

### 4.2 Netstat

El comando *netstat*, además de visualizar la tabla de rutas del equipo, permite visualizar las conexiones del nivel de transporte existentes en la máquina y su estado: establecidas, en establecimiento, en liberación, en espera, en escucha.... Las opciones principales de este comando son:

**netstat**

**-rn** Visualiza la tabla de encaminamiento.

**-t** Visualiza la información relativa a conexiones TCP establecidas.

- u Visualiza la información relativa a conexiones UDP establecidas.
- l Visualiza los puertos TCP o UDP que están a la espera de recibir peticiones de conexión.
- a Visualiza toda la información referente a conexiones en la máquina, incluyendo los puertos en espera de conexiones.

### 4.3 Netcat

*Netcat* es una herramienta que permite establecer una conexión TCP o UDP con una máquina e intercambiar información. *Netcat* trabaja de forma interactiva, es decir, una vez establecida una conexión se envía la información procedente de la entrada estándar a través de la conexión (socket) y la información recibida a través de la conexión se visualiza en la salida estándar.

*Netcat* puede emplearse como sustituto a la aplicación **rexec** estudiada en la práctica 3. Para ello se establece con *netcat* una conexión al puerto 512 de la máquina con el servicio *rexec* activado y se le envía el comando a ejecutar en el formato adecuado, que es:

```
"\0usuario\0contraseña\0comando\0"
```

Por ejemplo, para ejecutar el comando **pwd** en el servidor 10.3.7.0 como usuario alumnos se ejecutará el comando:

```
usuario@host:~> echo -e "\0alumnos\0alumnos\0pwd\0" | /usr/bin/nc -w 1 10.3.7.0 512
```

El resultado del mismo será:

```
/home/alumnos
```

De igual forma, podemos emplear *netcat* como sustituto para la herramienta **udp**. Para utilizar el servicio UDP *echo* ejecutaremos:

```
usuario@host:~> echo -e "texto" | /usr/sbin/nc -u -w 1 10.3.7.0 7
```

Obteniendo como resultado:

```
texto
```

La forma de empleo de *netcat* es **nc [opciones] host puerto**, siendo las opciones más importantes:

- u Indica que el puerto **puerto** es de tipo UDP. Si no se indica esta opción se considera TCP.
- w **x** *Netcat* espera **x** segundos para cortar la conexión una vez que finaliza el intercambio de datos.

Con el comando **man** es posible obtener más información acerca de este comando.

### 4.4 Ngrep

*Ngrep* es una herramienta de red que trabaja de forma análoga al comando **grep** pero sobre el contenido de los paquetes que circulan en la red. *Grep* analiza las líneas de texto introducidas a través de la entrada estándar y visualiza aquellas en las que esté presente una expresión

regular dada. Por ejemplo, si queremos visualizar las líneas de un fichero donde aparece la cadena "documento" se ejecutará:

```
usuario@host:~> cat fichero | grep 'documento'
```

*Ngrep* realiza las mismas funciones, pero la cadena se busca dentro de los paquetes que circulan en la red. Si ejecutamos el comando:

```
usuario@host:~> sudo /usr/bin/ngrep -d eth0 -q "1234"
```

se visualizarán los paquetes capturados en el interfaz *eth0* y en cuyo contenido aparece la cadena "1234". Puede comprobarse que los paquetes ICMP *echo request* y *echo reply* serán visualizados por este filtro.

Algunas de las opciones para el comando *ngrep* son:

- q** Sólo visualiza los paquetes capturados que contengan la cadena especificada en *ngrep*.
- d** Selecciona el dispositivo donde se realiza la captura paquetes para realizar la búsqueda.

Esta herramienta es muy útil cuando se desea disponer de un sistema de alertas. Si conocemos una cadena en concreto que se intercambia en un paquete, es posible detectar cuando se produce el envío de ese paquete.

Es necesario disponer de permisos de administrador (root) para emplear esta herramienta.

## **5. Listas de control de acceso (ACLs)**

### **5.1 Introducción**

Las listas de control de acceso o ACLs, representan un mecanismo para clasificar tráfico dependiendo de las características del protocolo, tales como direcciones IP, puertos, bits de control de las cabeceras, etc.

Originalmente diseñadas con fines de control de acceso a recursos de la red, las ACLs son actualmente la piedra angular de muchos sistemas operativos de red, tales como el IOS de Cisco, actualmente el más extendido y convertido en un estándar de hecho.

Es sobre este sistema operativo de red (IOS), donde se realizarán todos los ensayos prácticos relativos al diseño y uso de listas de control de acceso.

### **5.2 Las listas de control de acceso**

#### **Finalidad**

Una lista de control de acceso permite clasificar un conjunto de paquetes que circulan a través de un dispositivo de nivel 2 (enlace) o nivel 3 (red), para luego poder actuar sobre ese patrón de flujo definido en dicha ACL.

Una vez clasificado el tráfico que nos interesa, podemos utilizarlo para algunas de las siguientes aplicaciones entre muchas otras:

- Control de acceso a recursos de la red: Seguridad.
- Control de acceso a los propios dispositivos de conmutación (Routers y Switches).
- Limitación de ancho de banda.
- Reserva de ancho de banda.
- Selección del grupo de direcciones IP privadas y públicas a utilizar por dispositivos con NAT o PAT.



- Encaminar el tráfico en función de una ACL, en lugar de la dirección IP destino de los paquetes.
- Definir qué tráfico debe ser autenticado (Solicitud de Usuario y Clave de acceso) y cual no.
- Definir qué tráfico se considera local y cual externo, para que los sistemas de seguridad (Cortafuegos e IDS (Intrusion Detection System)) actúen en consecuencia.
- Seleccionar qué tráfico debe ser encriptado por protocolos como IPSEC, permitiendo definir asociaciones de seguridad o SA.

### **Características de las ACL**

Una ACL es una colección secuencial de condiciones 'permite' y 'deniega' (permit and deny) que se aplican a los paquetes que la atraviesan. Si la ACL se aplica sobre una interfaz, entonces se permite o deniega el tráfico seleccionado.

Cuando un paquete llega a un interfaz que tiene aplicado una ACL, el procesador del router compara de forma secuencial los campos existentes en dicho paquete con los atributos definidos en cada línea de la ACL. Si coincide con alguna de ellas, entonces la comparación termina y la comprobación de la ACL devuelve un valor Verdadero o Falso (True o False), determinando si el paquete debe ser o no encaminado por el router. Si el paquete procesado no tiene coincidencia con ninguna línea de la ACL, el paquete es descartado (no encaminado).

Es por tanto crítico el orden en que se aplican las líneas de la ACL y, por tanto, es necesario tenerlo muy en cuenta en el diseño de la misma.

Además de las funciones básicas descritas se incorporan una serie de prestaciones adicionales entre las que destacan:

- Cada línea de una ACL dispone de contadores que nos indican el número de paquetes que han cumplido esa condición. Muy útil en la fase de depuración y optimización.
- Es posible asociar calendarios a cada línea, de modo que permanecerá activa durante el rango de tiempo definido (por ejemplo: Lunes-Viernes de 8 a 15, fines de semana, etc...).
- Se permite el uso de entradas dinámicas, que pueden ser introducidas desde dispositivos externos mediante comandos 'rexec', facilitando así la integración de los dispositivos de red con servidores de comunicaciones.

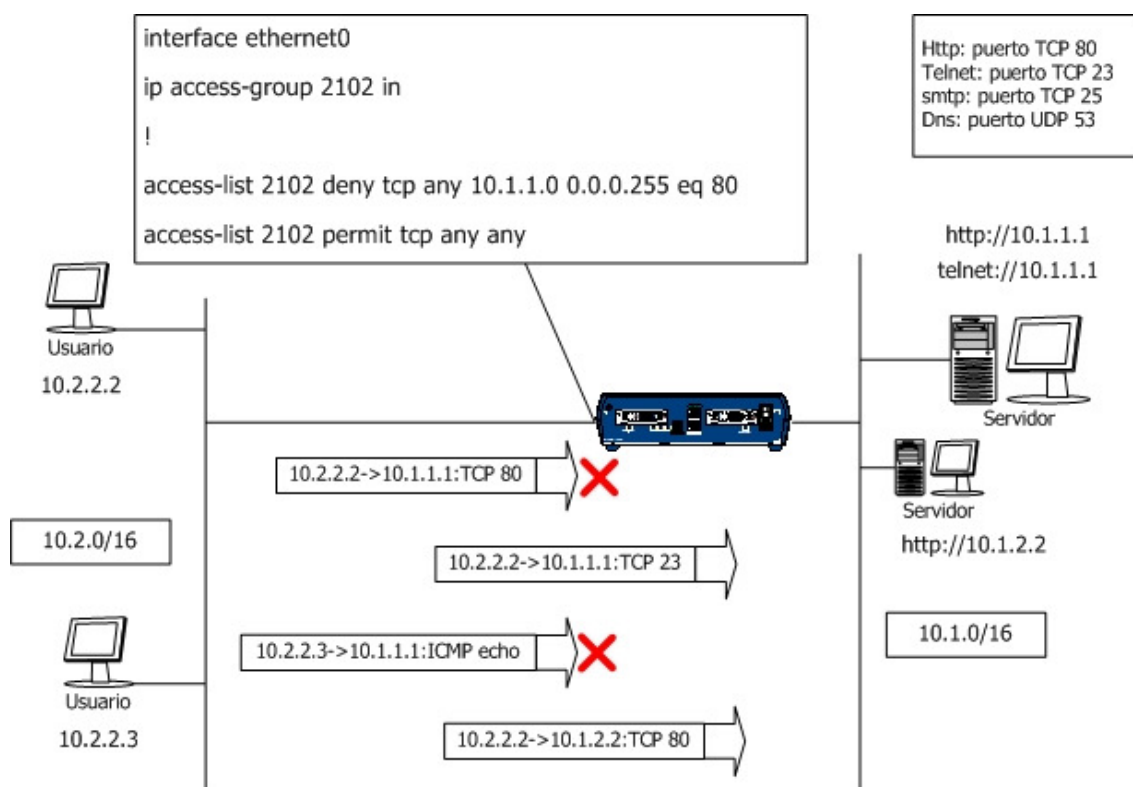
Dentro de la sintaxis utilizada para la construcción de ACLs, cabe considerar que los números utilizados definen su espectro de actuación. Es decir:

<1-99>	Lista IP standar
<100-199>	Lista IP extendida
<1100-1199>	Lista LAN Extendida con direcciones de 48-bit MAC
<1300-1999>	Lista IP standar (rango expandido)
<200-299>	Lista de acceso por campo 'type-code'
<2000-2699>	IP extended access list (rango expandido)
<700-799>	Lista LAN con direcciones de 48-bit MAC

En los ejercicios de esta práctica se empleará el rango 2000-2699 para la implementación de ACL.

### **Funcionamiento**

Veamos lo expuesto anteriormente mediante una ACL aplicada para filtrar el tráfico que entra en un interfaz de un router.



**Figura 4. Escenario 1 de filtrado con listas de control de acceso**

En el escenario de la figura 4 se representan dos redes separadas por un router.

En la interfaz de entrada del router, denominada Ethernet0, se encuentra aplicada la lista de acceso 2102, mediante la orden **'ip access-group 2102 in'**, donde la cláusula 'in' indica que se aplica solo al tráfico que entra por dicha interfaz. Pueden existir listas de entrada (in) y de salida (out) simultáneamente, aunque en la práctica solo se usan sobre un único sentido de comunicación.

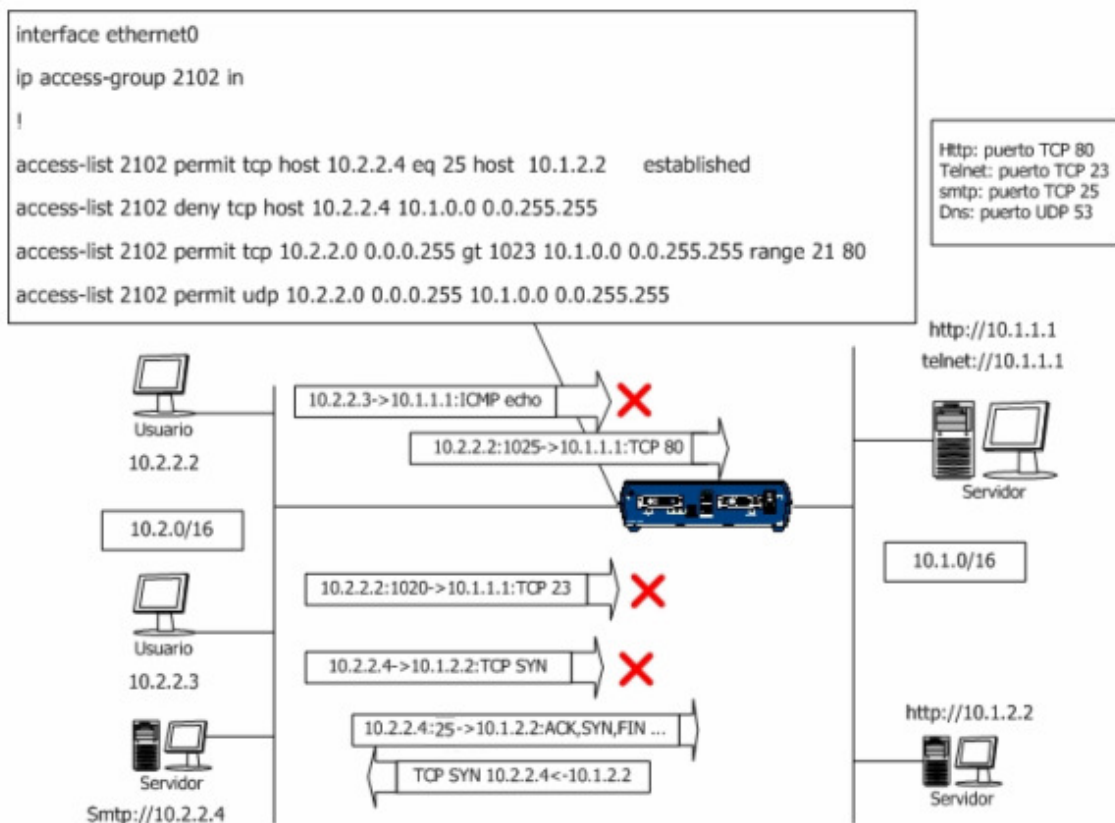
La ACL 2102 deniega en su primera línea todo el tráfico con cualquier dirección IP origen hacia el servicio http de cualquier máquina destino cuya dirección empiece por 10.1.1. , con lo que el servicio web del servidor con dirección 10.1.1.1 no es accesible desde cualquier red anterior al router.

Hay que hacer notar que el valor que acompaña a una dirección IP en la ACL se denomina 'wildcard' y se corresponde con el valor complementario a la máscara de red de la dirección IP. Es decir, cuando queremos indicar todas las direcciones IP 10.1.1.0 con máscara 255.255.255.0, el valor 'wildcard' que acompaña a la dirección IP en la ACL será 0.0.0.255.

Cualquier paquete que no cumpla con la primera línea de la ACL, será comprobado con la segunda. Con más permisividad, la segunda línea permite todo el tráfico TCP entre cualquier dirección origen y destino.

Por último, es importante resaltar que siempre existe un 'deny' implícito al final de cualquier ACL, de modo que el tráfico que no cumpla con ninguna de las líneas de la ACL, se descarta automáticamente.

Veamos otro ejemplo, mediante el gráfico siguiente:



**Figura 5. Escenario 2 de filtrado con listas de control de acceso**

En la figura 5 se representa un escenario muy similar al de la figura 4, solo que en este caso, la ACL es diferente.

En la primera línea se permiten paquetes TCP procedentes de la dirección de máquina 10.2.2.4 (opción 'host') y puerto TCP origen igual a 25 (opción 'eq 25') hacia el servidor 10.1.2.2 siempre que el bit **ACK** esté activo (opción 'established').

Esto significa que en esta línea se permitirá solo aquel tráfico que se haya iniciado desde el servidor 10.1.2.2 hacia el puerto 25 de la máquina 10.2.2.4

Si recordamos la secuencia de paquetes para el establecimiento de una conexión TCP, el único segmento que no lleva activo el bit ACK es el primero, es decir el SYN; con lo que cualquier paquete de establecimiento de conexión desde 10.2.2.4 hacia 10.1.2.2 no cumplirá la primera línea y pasará a la segunda, donde será rechazado.

La segunda línea, efectivamente se encarga de rechazar cualquier paquete procedente de 10.2.2.4 y hacia el prefijo de red 10.1.0.0, que no cumpla con la primera condición, descrita anteriormente. Si el paquete cumple con esta línea, el proceso se acaba y se devuelve un valor False al proceso que utiliza la ACL 2102. Esto puede ocurrir en cualquier línea de la ACL, solo que si la línea contiene 'permit' se devolverá el valor True.

La tercera línea comprueba los paquetes con puertos TCP superiores al 1023, procedentes de direcciones origen que empiecen por el valor 10.2.2. y vayan dirigidos a destinos que empiecen por 10.1. con puertos destino comprendidos en el rango 21 a 80, ambos incluidos.

La cuarta línea y última, se encarga de permitir cualquier tipo de tráfico UDP entre las máquinas cuyas direcciones IP empiecen por los valores 10.2.2. y 10.1.

Nótese que el 'deny' implícito del final bloqueará cualquier otro tipo de tráfico IP como ICMP, GRE, IPSEC, etc.

Como resumen de lo expuesto, podemos esquematizar los siguientes pasos del modo siguiente:

1. Un paquete es procesado por una ACL
2. Se comprueba con la primera línea
3. Si se cumple la condición y la línea contiene 'permit' el proceso acaba y se devuelve el valor True al proceso que utiliza la lista
4. Si se cumple la condición y la línea contiene 'deny' el proceso acaba y se devuelve el valor False al proceso que utiliza la lista
5. Si no se cumple, entonces se ejecuta la línea siguiente
6. Si el proceso llega hasta el final de la lista sin haber acabado, entonces caerá en una línea implícita que siempre contiene un 'deny any any', y por tanto devolverá valor False.

En los dos ejemplos anteriores, hemos aplicado las ACL a los paquetes de entrada al interfaz Ethernet0 con la instrucción *'ip access-group 2102 in'*. Podríamos haberlas aplicado para el tráfico de salida con *'ip access-group 2102 out'*.

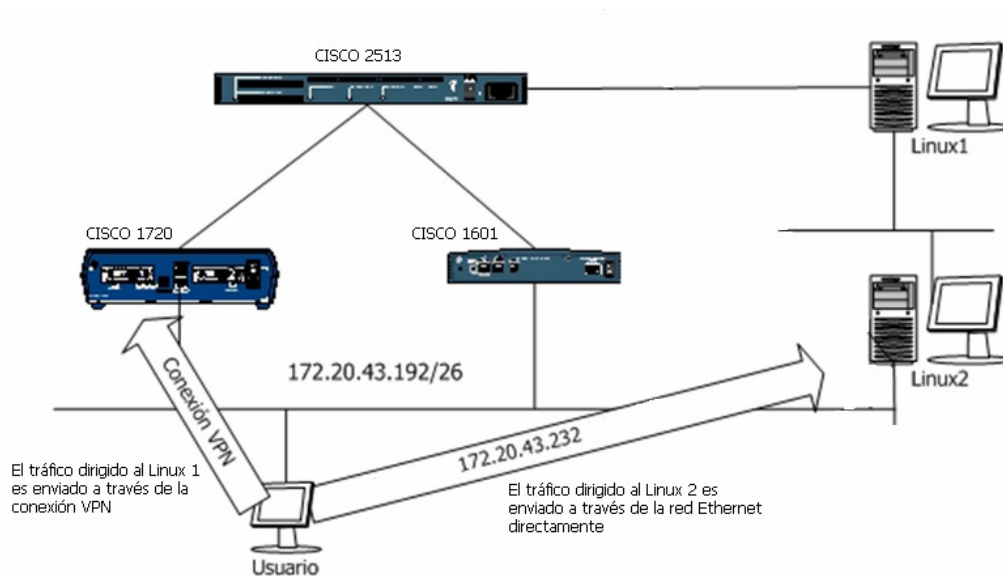
## **6. Acceso remoto a una red empleando conexiones VPN**

Para poder experimentar con el control del tráfico en un router CISCO hay que tener en cuenta que sobre un interfaz de un router (por ejemplo el interfaz ethernet del router CISCO 1720) sólo es posible aplicar una lista de acceso.

Esto supone una limitación a la hora de conseguir en el laboratorio que cada alumno pueda configurar una lista de acceso propia, pues no se podrá realizar simultáneamente. Para evitar este problema se van a emplear conexiones VPN (Virtual Private Network).

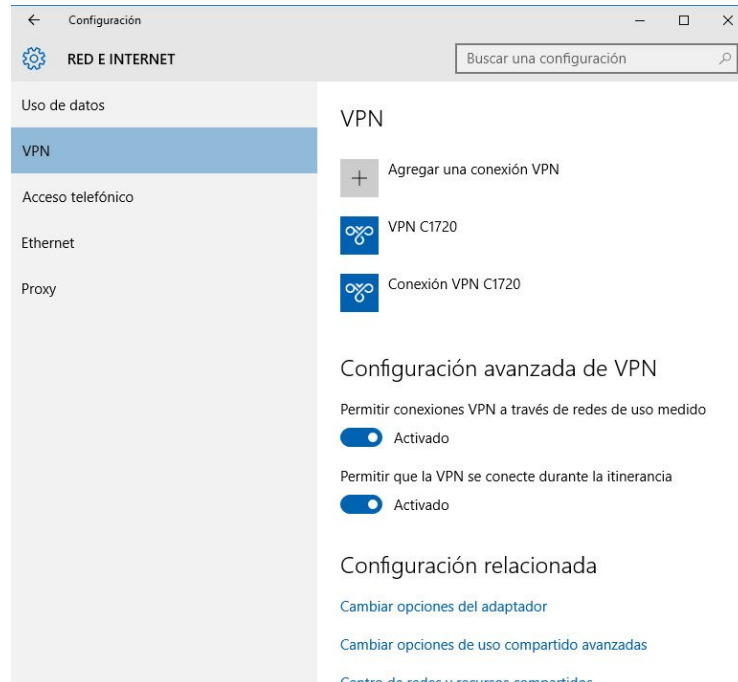
Una de las características que proporciona una VPN es establecer una conexión punto a punto desde un equipo a cualquier red. En cada conexión VPN el router asigna un interfaz virtual de red sobre el que se podrá aplicar una lista de acceso diferente para cada conexión.

Así, el alumno deberá realizar una conexión VPN al router CISCO 1720 estableciendo un enlace punto a punto. Esta conexión VPN está configurada para que todo el tráfico de paquetes intercambiado entre el PC del alumno y el router Linux 1 sea enviado a través de la conexión VPN al router CISCO 1720. Este modo de funcionamiento se describe en la siguiente figura.

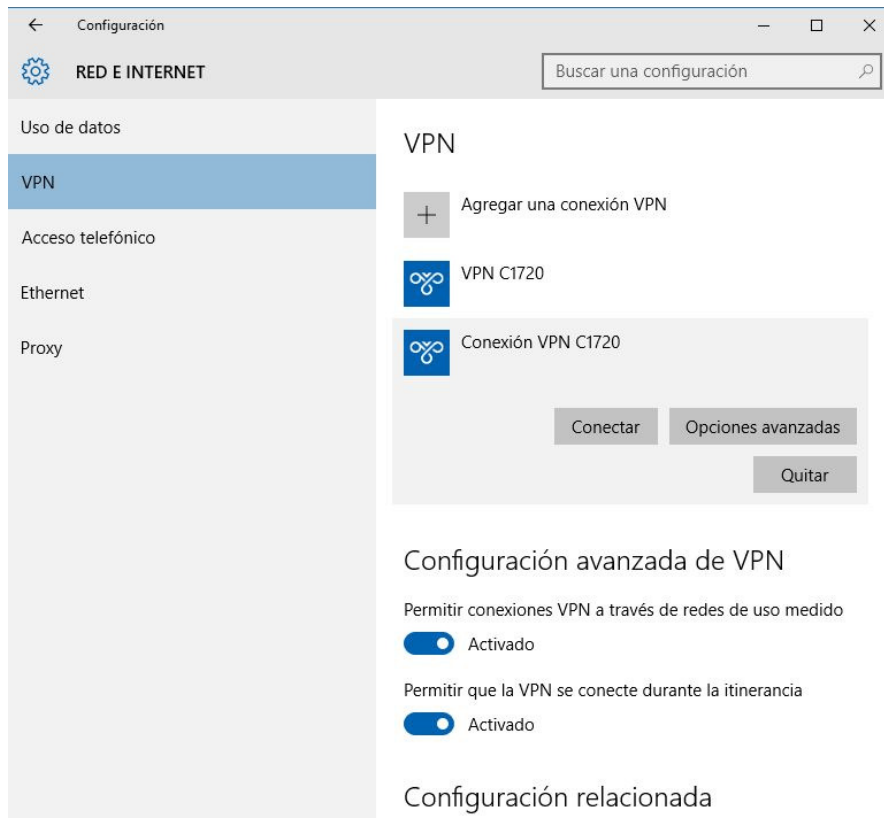


**Figura 6. Escenario de funcionamiento de accesos VPN y Listas de Control de Acceso**

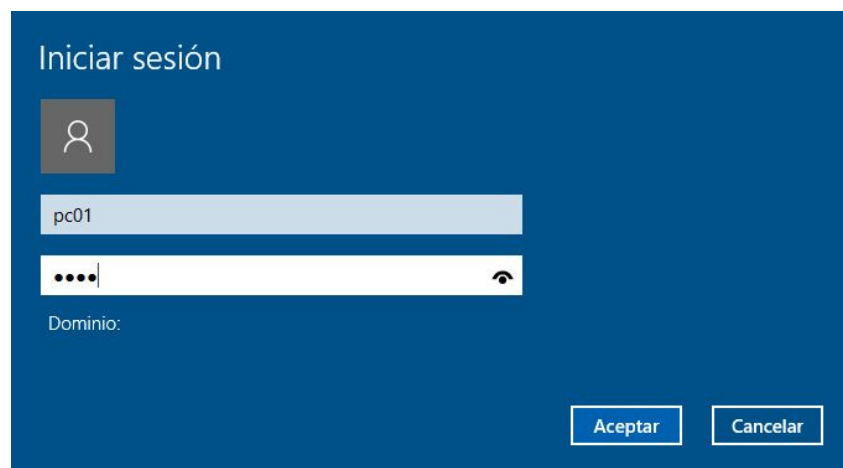
Para establecer una conexión VPN al router CISCO 1720 se empleará el cliente de Windows 10 de acceso VPN. Este cliente está configurado en el sistema y es accesible desde el escritorio, denominándose '**Conexión VPN C1720**'. Al ejecutarlo se despliega una ventana donde se selecciona el acceso citado.



Seleccionado la conexión '**Conexión VPN C1720**' se habilita el botón 'Conectar'.



Pulsando en el botón 'Conectar' se muestra una ventana de diálogo para indicar un nombre de usuario y contraseña.



Para permitir que todos los alumnos puedan acceder simultáneamente se ha creado una conexión VPN por equipo. Para cada PC, la conexión se establece empleando como nombre de usuario '**pcxx**', donde '**xx**' es el número de PC del 01 al 30, y contraseña la misma que el nombre de usuario.

Una vez realizada la conexión VPN, se nos asigna una nueva dirección IP que es la 10.1.0.x, donde x es el número del PC que empleamos. Esta dirección 10.1.0.x pertenece a la red 10.1.0.0/16, una red virtual conectada al router CISCO 1720. Hay que tener en cuenta que una vez establecida la conexión VPN, todo el tráfico que intercambiemos fuera de la red 172.20.43.192/26 (la red local Ethernet) empleará la dirección IP 10.1.0.x para nuestro PC.

Para que la conexión VPN deje de estar activa y liberarla, es necesario volver a la ventana de gestión de la conexión '**Conexión VPN C1720**' y pulsar el botón 'Desconectar'.

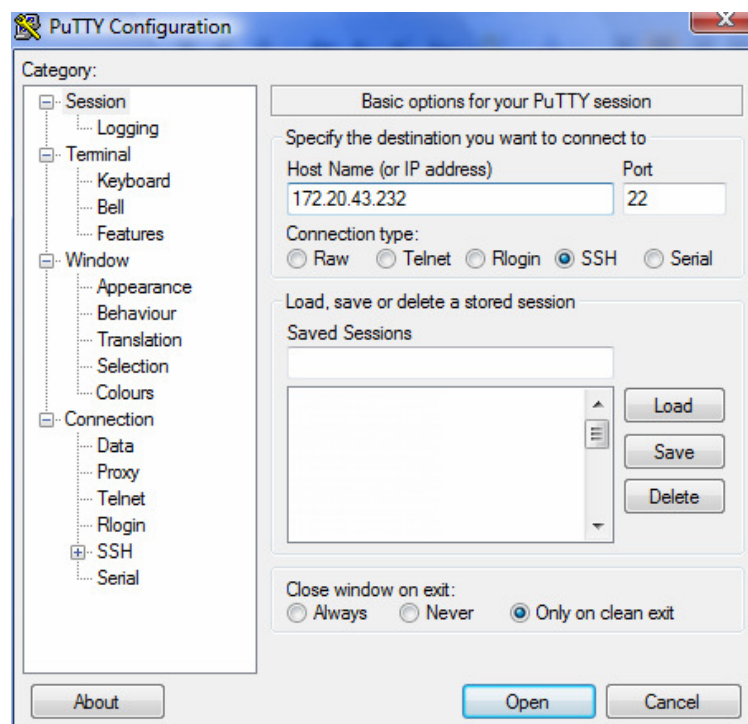


## 7. Cuestiones a realizar

### Acceso a un sistema linux

Para la realización de la práctica 4 se ha habilitado el acceso al router Linux2 del laboratorio. Desde cada PC se accederá al equipo Linux2 empleando una conexión **telnet segura**. 'Telnet' es un protocolo de acceso remoto a un equipo, accediendo a la consola del sistema en modo texto. El aspecto de seguridad se consigue cifrando los paquetes de información entre el cliente y el servidor con el estándar **TLS** (sustituto del no recomendado SSL). Para ello se emplea el servicio **SSH** (*Secure Shell*), que permite el cifrado de una conexión remota telnet.

Para el empleo del servicio SSH se empleará una aplicación ampliamente extendida que se denomina PUTTY.EXE. Esta aplicación está disponible en los PC's del laboratorio L24.

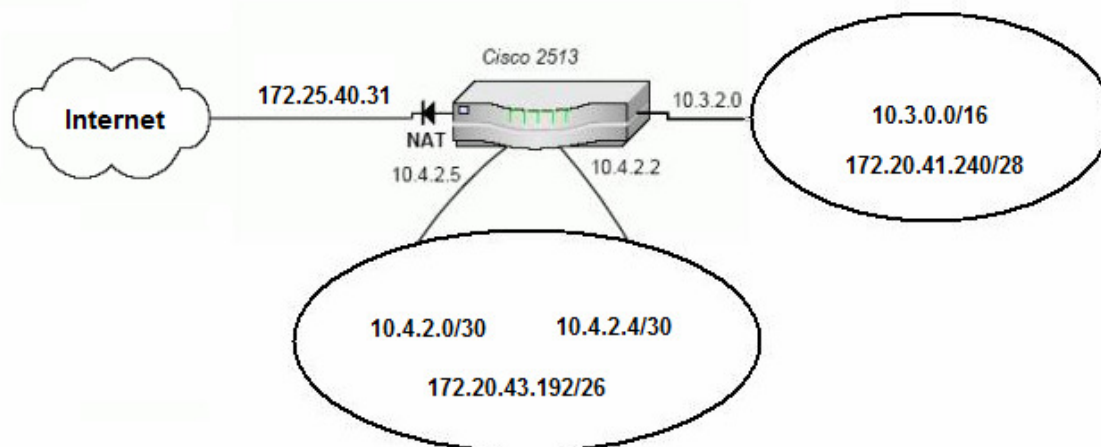


Para realizar la conexión SSH al equipo Linux2 hay que especificar en el campo 'Host Name' la dirección IP 172.20.43.232 y en el campo 'Port' el valor 22, el puerto asociado al servicio SSH. Para establecer la conexión sólo hay que pulsar en el botón **'Open'**.

El nombre de usuario y contraseña que se empleará para cada alumno será el mismo que se emplea en la conexión VPN. Es decir, el nombre de usuario será **'pcxx'**, donde **xx** es el número del PC, y la contraseña será la misma que el nombre de usuario.

## 7.1 NAT

En el esquema de red del laboratorio, el router Cisco 2513 emplea la funcionalidad de NAT para permitir la conectividad con Internet.



**Figura 7. Funcionamiento de NAT en el laboratorio de prácticas**

El router Cisco 2513 está configurado para que todo el tráfico de las redes IP del laboratorio que tenga como destino una dirección IP de Internet (IP pública) sea traducido empleando la dirección IP 172.25.40.31 (**Inside global**).

Para poder comprobar que se realizan las traducciones es necesario visualizar la tabla de traducciones del router Cisco 2513.

### 1. Visualización de traducción de paquetes

Desde el PC del alumno, con la aplicación Netcat (**nc.exe**) (**disponible en la carpeta 'Prácticas' de los materiales disponibles en UACloud. Debe ser descargada al directorio de trabajo donde se ejecutará el comando**) se enviarán paquetes TCP SYN a un servidor web de Internet y se visualizará la traducción de los paquetes SYN (el servidor web al que se accederá no existe y por tanto no se establecerá una conexión).

Cada alumno empleará una dirección IP de Internet distinta, en concreto **150.150.150.64+x**, donde **x** es el número de PC del alumno (es decir, 1, 2, 3, 4, etc.).

Para visualizar la tabla de traducciones NAT del router Cisco 2513 puede emplearse la aplicación **rexec** de la práctica 3 de la asignatura. Para ello, en el servidor Linux2 (172.20.43.232) se encuentra disponible el script **'NAT2513'** que visualiza el contenido de la tabla de traducciones NAT del router Cisco 2513. Para ejecutarlo puede emplearse el usuario "alumnos" y contraseña "alumnos".

Inicia el monitor de red en el PC del alumno para capturar el tráfico sin traducir generado por el alumno. Emplea un filtrado por la dirección IP 150.150.150.64+x.



Una vez iniciada la captura del tráfico se enviarán paquetes SYN al servicio web del destino 150.150.150.64+x con la aplicación **nc**.

**C:\nc -nv 150.150.150.64+x 80**

Espera la finalización de la aplicación nc que envía paquetes TCP SYN.

Empleando la aplicación **rexec**, ejecuta el comando '**NAT2513**' en el servidor Linux2 (la ejecución tiene un retardo de unos 20 segundos).

a) Determina cómo se realiza la traducción del paquete SYN en el router Cisco 2513.

Ten en cuenta que las traducciones están activas durante un cierto tiempo en el router. Si no se reciben paquetes la traducción se elimina de la tabla para dejarla libre a otras.

Empleando netcat, envía paquetes SYN con diferentes valores de puerto origen al mismo servidor. Para ello emplea la opción **-p puerto** de netcat. Nótese que el número de puerto origen será distinto para cada alumno.

**C:\nc -p 2064+x -nv 150.150.150.64+x 80**

**C:\nc -p 3064+x -nv 150.150.150.64+x 80**

**C:\nc -p 4064+x -nv 150.150.150.64+x 80**

Recuerda que **x** es el número de PC del aula en el que trabaja el alumno.

b) ¿ Cómo se emplea el puerto origen a la hora de realizar la traducción ?

## **2. Overload (sobrecarga) del puerto origen**

Cuando el router que emplea NAT recibe un paquete IP a traducir con un número de puerto origen que ya está empleando en otra traducción, se produce una sobrecarga del número de puerto. Es decir, el número de puerto origen se cambia por uno no empleado en la tabla de traducciones del router.

Trabaja en colaboración con un compañero para provocar una sobrecarga de puertos en el router.

Para ello debéis emplear el mismo número de puerto origen a un mismo destino. Un alumno ejecuta el comando:

**C:\nc -p 2064+x -nv 150.150.150.64+x 80**

Y el otro ejecuta:

**C:\nc -p 2064+x -nv 150.150.150.64+x 80**

Donde **x** es el mismo valor para ambos alumnos.

Por ejemplo, los alumnos que trabajan en los PC's **PC01** y **PC02** ejecutarán los comandos:

**C:\nc -p 2065 -nv 150.150.150.65 80**

**C:\nc -p 2065 -nv 150.150.150.65 80**

a) Determina qué puertos origen emplea el router Cisco 2513 en la traducción visualizando la información de la tabla de traducciones.

b) ¿ Qué ocurre si el acceso se produce a servidores web distintos ? ¿ Cómo se realiza la traducción ?

Continuando con el ejemplo de los alumnos en los PC's **PC01** y **PC02** se ejecutaría:

```
C:\nc -p 2065 -nv 150.150.150.65 80
```

```
C:\nc -p 2065 -nv 150.150.150.66 80
```

## 7.2. Gestión de red con el sistema operativo Linux

### 1. Configuración de las propiedades TCP/IP en Linux

Realizar una conexión con SSH al equipo Linux2 con el nombre de usuario asociado al PC del alumno.

Empleando los comandos **ifconfig** y **netstat** determina los interfaces de red de que dispone el equipo Linux 2 y cuál es su tabla de rutas.

- ¿Cuál es la puerta de enlace por defecto del router Linux 2 ?

### 2. Conexiones en la capa de transporte

Empleando el comando **netstat** determina la siguiente información acerca de conexiones en la capa de transporte.

- Determina las conexiones TCP establecidas en el Linux 2.
- Determina qué puertos TCP están en espera de recibir conexiones.
- Determina las conexiones UDP establecidas en el Linux 2.
- Determina qué puertos UDP están en espera de recibir conexiones.

### 3. Establecimiento de conexiones TCP y UDP

Empleando el comando **netcat** establece las conexiones TCP y UDP que se indican a continuación, comprobando que el resultado de las mismas es el esperado.

- Emplea el servicio UDP echo con **echo -e "Texto de prueba" | /usr/bin/nc -w 1 -nu 172.20.41.241 7**
- Emplea el servicio UDP daytime con **echo -e "Texto de prueba" | /usr/bin/nc -w 1 -nu 172.20.41.241 13**
- Emplea el servicio TCP rexec con **echo -e "\0alumnos\0alumnos\0ls -l\0" | /usr/bin/nc -w 1 -n 172.20.41.241 512**

## 7.3 Creación de listas de control de acceso (ACL) en el router CISCO 1720

En estas cuestiones se realizará una conexión SSH al equipo Linux2, donde se crearán las listas de acceso y se enviarán al router Cisco 1720.

Una vez creada en el router la lista de acceso, se realizará la conexión VPN al router Cisco 1720 empleando el icono del escritorio '**Conexión VPN C1720**' y el nombre de usuario y contraseña asociado a tu equipo. HASTA QUE NO SE REALICE LA CONEXIÓN VPN NO SE APLICARÁ EL FILTRADO ESPECIFICADO EN LA LISTA DE CONTROL DE ACCESO.

Por otra parte, para validar el funcionamiento de la ACL creada se emplearán las aplicaciones **ping**, **udp.exe** y **netcat (nc)** en Windows. La ayuda acerca del empleo de la herramienta netcat puede obtenerse ejecutando el comando **nc -h**. De las muchas opciones que permite netcat,

emplearemos la que nos permite enviar paquetes TCP SYN al número de puerto y máquina que escojamos. El formato de uso de netcat será:

**C:\nc -zvn DIRECCION\_IP PUERTO**

**Para crear y validar una ACL, será necesario seguir de forma estricta los siguientes pasos:**

1. Acceso SSH al equipo Linux2 con la cuenta asociada al PC del alumno.
2. Creación de un fichero de texto mediante un editor de Linux (vi, joe) en el cual aparezcan las líneas de la ACL a configurar, comenzando siempre por una línea que contenga la cláusula '**no access-list <número>**'.
3. El número utilizado en la ACL debe comenzar por 20 seguido del número de 2 dígitos del PC desde donde se está realizando la práctica. Por ejemplo 2003, para el pc03.
4. El nombre del archivo debe comenzar por el número de la acl seguido del prefijo .acl. Por ejemplo, 2003.acl para el pc03.
5. Para enviar la ACL al router CISCO 1720 se utilizará el siguiente comando: **acl1720 20xx.acl**, donde xx es el número de PC y **20xx.acl** debe ser el nombre del fichero creado que contiene la ACL.
6. **Realizar la conexión VPN.**
7. Desde el PC local, mediante 'ping' y 'netcat' se pueden realizar comprobaciones.
8. Desde la conexión con Linux2, se puede ejecutar el comando **cmd1720 20xx** para ver el estado de los contadores de la ACL.

En la figura siguiente se representa un ejemplo con el PC 02 del laboratorio. En este caso, su ACL sería la 2002, y el nombre del fichero 2002.acl:

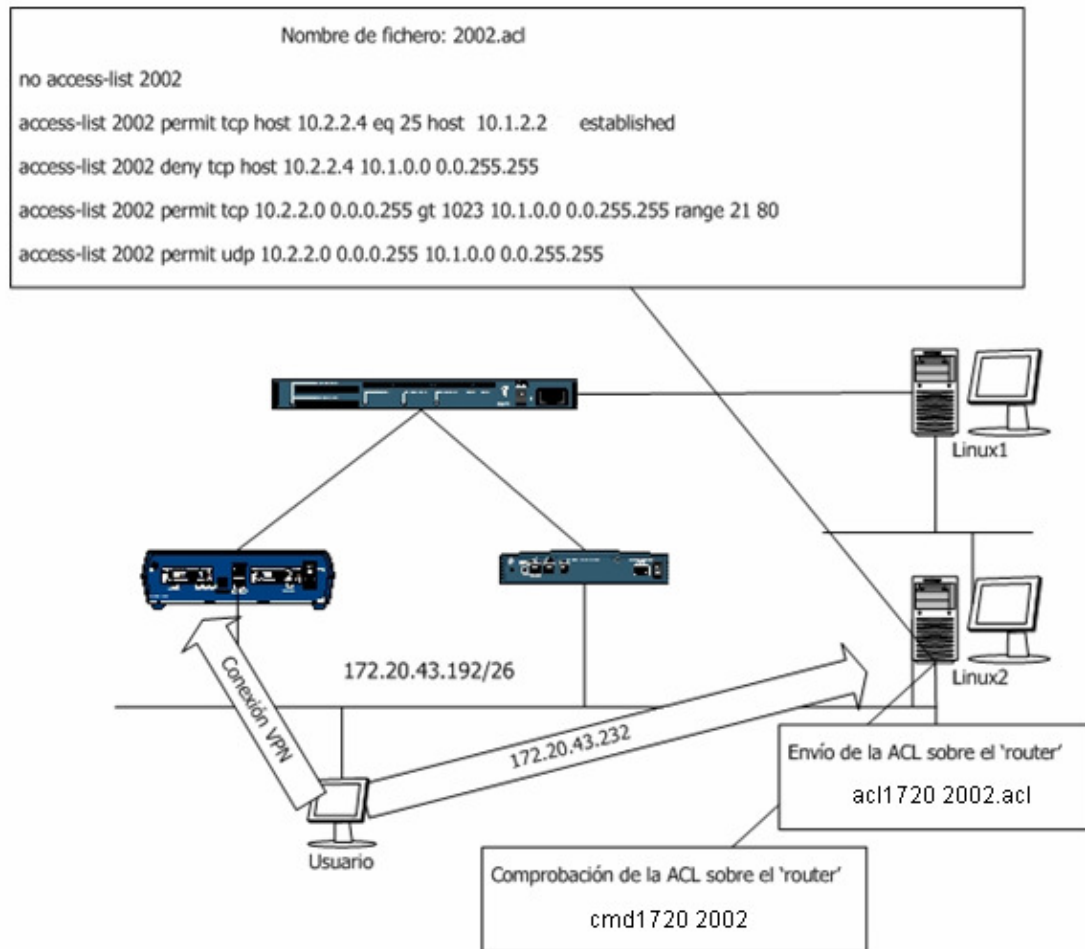


Figura 4

## 1. Creación de una ACL permisiva

Se realizará una ACL con las siguientes características:

- Se permite todo el tráfico TCP.
- Se permite todo el tráfico UDP.
- Se permite todo el tráfico ICMP.

Verificar la lista con las siguientes acciones:

- Comprobar si es posible realizar conexiones al puerto TCP 23 del equipo 10.3.7.0 (comando a emplear: '**C:\nc -zvn 10.3.7.0 23**').
- Comprobar si es posible realizar conexiones al puerto UDP 7 del equipo 10.3.7.0 empleando la aplicación **UDP.EXE**.
- Comprobar si **ping** tiene conectividad con el equipo 10.3.2.0.
- En el equipo Linux2 y mediante la utilidad '**cmd1720 20xx**' comprobar los contadores de paquete permitidos y no permitidos por la lista de acceso en el router CISCO 1720. Para cada línea de la lista de acceso creada se indica el número de paquetes que han llegado al router y han coincidido con la condición especificada en la línea.

## 2. Creación de una ACL básica

Se realizará una ACL con las siguientes características:

- No permitir conexiones TCP al puerto 80 de las direcciones IP 172.0.0.0/8.
- No permitir conexiones TCP ni UDP al equipo 10.3.7.0
- Permitir conexiones UDP a las direcciones 172.0.0.0/8.
- Permitir conexiones TCP a cualquier destino.
- Bloquear el resto del tráfico.

Verificar la lista con las siguientes acciones:

- Comprobar si es posible el acceso web a servidores de la Universidad de Alicante ([dfests.ua.es](http://dfests.ua.es), [www.dlsi.ua.es](http://www.dlsi.ua.es), [www.dccia.ua.es](http://www.dccia.ua.es), [www.dtic.ua.es](http://www.dtic.ua.es), etc.). ¿ Por qué ?
- Comprobar si es posible el acceso web a servidores externos a la Universidad de Alicante ([www.google.es](http://www.google.es), [www.upv.es](http://www.upv.es), etc.). ¿ Por qué ?
- Comprobar si es posible acceder al servicio rexec del equipo 10.3.7.0 (Linux 1) con la aplicación **REXEC.EXE** ¿ Existe alguna forma de acceder al servicio rexec del equipo Linux 1 ?
- Comprobar si es posible acceder al servicio UDP ECHO del equipo 10.3.7.0. ¿ Sería accesible de alguna forma ?
- ¿ Con qué máquinas del laboratorio hay conectividad con la aplicación **ping** ?
- ¿ Se recibe algún mensaje de error ICMP ? ¿ De qué tipo ? ¿ A qué crees que es debido ?

## 8. Documentación complementaria

- Documentación **man** de Linux Red Hat.
- **RFC 2663** IP Network Address Translator (NAT). IETF.
- **RFC 2637** PPTP Point-to-Point Tunneling Protocol. IETF.
- **RFC 2661** L2TP Layer 2 Tunneling Protocol. IETF