

5.2. Verificación y monitorización de la red



Índice

Objetivos.....	4
Verificación del funcionamiento de una red mediante el uso de comandos	5
Verificación de la red	5
Protocolos y comandos de gestión.....	5
¿Y qué debemos verificar en la red?	6
Comandos para verificar la red.....	7
Sistemas operativos en red	7
Problemas de conectividad en la red.....	8
Comandos de consola.....	10
Comandos utilizados en Windows	10
Comandos utilizados en Linux	16
Ejercicio: practicar con ping	20
¿Por qué debemos tener las máquinas virtuales en modo " <i>bridged</i> "?.....	21
Monitorización de redes.....	23
Monitorización pasiva/reactiva y monitorización activa.....	23
Monitorización pasiva/reactiva.....	23
Monitorización activa	23
Pasos a seguir en la monitorización de la red.....	24
Información de monitorización.....	24
Mecanismos de monitorización	25
Protocolos y configuración de la red.....	26
Protocolos UDP y TCP.....	26
El protocolo UDP.....	27
El protocolo TCP.....	28
Ejemplo de conexiones TCP	28
¿Qué protocolo usa cada servicio?.....	29
El protocolo SCTP.....	29
Otros protocolos	30
El protocolo SNMP.....	30
Configurar los adaptadores de red.....	31

Configuración del adaptador de red en Windows	31
Configuración del adaptador de red en Linux.....	32
Adaptadores de red	33
Los adaptadores de red.....	34
Dispositivos de interconexión.....	36
Tipos de dispositivos de interconexión	37
Despedida	42
Resumen.....	42

Una vez configurada nuestra red, la función del administrador pasa por la observación y medida de su funcionamiento con el objetivo de obtener el mayor rendimiento posible de todos los elementos y del conjunto.

En este tema nos centraremos en conocer algunos de los comandos más comunes empleados para la monitorización y verificación del funcionamiento de una red sin pretender ser exhaustivos, pues siempre ha revisarse y ampliar el estudio sobre el sistema y la red particular sobre la que vayamos a trabajar.

Objetivos

Con esta lección perseguimos los siguientes objetivos:

1. Conocer los principales conceptos para poder verificar los elementos y el conjunto de nuestra red, pudiendo elaborar estadísticas con datos sobre el funcionamiento y el tráfico.
2. Conocer algunos de los paquetes software que existen en el mercado para la configuración o monitorización de la red.
3. Conocer aquellos comandos más comunes empleados para la monitorización y verificación del funcionamiento de una red.

Verificación del funcionamiento de una red mediante el uso de comandos

Verificación de la red

Al hablar de verificación de la red debemos tener en cuenta que siempre tendremos que adaptar esta tarea a la configuración concreta de la red que estemos verificando

Sin embargo, intentaremos señalar primero algunos aspectos generales a verificar. Luego nos centraremos en los comandos disponibles para ello y en las acciones concretas.

El administrador de la red podrá verificar, por ejemplo:

1. Verificación del **rendimiento de los equipos servidores**, en concreto:
 - El **rendimiento de las CPU**.
 - Sus **recursos de memoria** y el intercambio entre la memoria virtual del disco (paginación).
 - **Niveles de transferencia** de entrada/salida (por ejemplo demasiados accesos a disco, lentitud, etc.).
2. El **tráfico de la red**, tanto el de la propia red en un uso "normal" como generando pruebas de tráfico para medir su respuesta.
3. **Monitorización de los protocolos de red**, viendo el volumen de datos que hay en la red de cada protocolo.
4. **Verificación de los enrutamientos** (tablas y rutas).

Protocolos y comandos de gestión

El enorme crecimiento de Internet, las redes de área local y los sistemas distribuidos, entre otros factores, han propiciado la aparición de técnicas y protocolos especializados para gestionar la red.

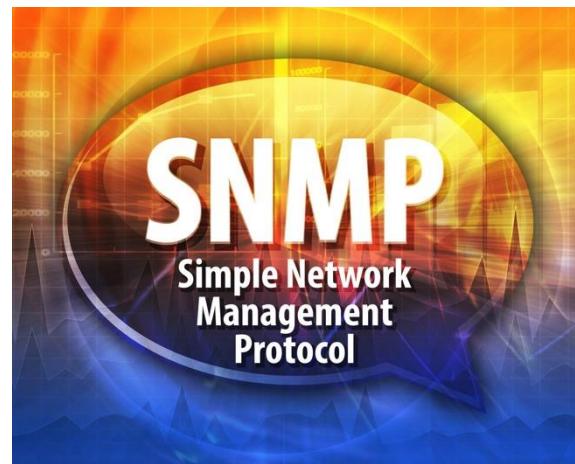
Estas técnicas y protocolos están destinados a facilitar la labor del administrador para, idealmente, permitirle gestionar a través de una consola toda la red.

Estos protocolos se encargan de **recoger información de los nodos de la red**, observar el tráfico y alertar de situaciones que requieran la actuación del administrador. En algunos casos también existen mecanismos de reconfiguración automáticos que permiten, por ejemplo, corregir problemas temporales en los enrutamientos o en la gestión del tráfico.

Para recoger la información sobre el estado de la red se instalan unos programas (conocidos como **entidades de gestión**) que se encargan de interrogar periódicamente a los elementos de la red (mecanismo de "**polling**") y recoger sus respuestas.

Para ello utilizan protocolos de gestión (como por ejemplo **SNMP** - "Simple Network Management Protocol", **RMON** –"Remote Monitoring", **SMON** – "Switching

Monitoring", etc.) almacenando la información en una base de datos ("Management Information Base" – **MIB**) a partir de la que se elaboran estadísticas e informes de rendimiento.



¿Y qué debemos verificar en la red?

Aunque cada administrador puede seguir sus propias directrices para verificar el funcionamiento de la red, en general se han definido una serie de áreas comunes, como son:

- La **configuración** y el **estado de los dispositivos** de la red.
- El **estado de las conexiones** y **rutas de encaminamiento**.
- El **rendimiento de la red** (estadísticas de tráfico, errores, etc.).
- La **seguridad de la red**.
- El **control de los fallos**.
- **Costes** de las comunicaciones (en recursos y/o monetarios).

En el mercado existen potentes herramientas software que facilitan la realización de esta gestión, proporcionando interfaces gráficas de mucha ayuda para el administrador de la red, y con la complejidad suficiente como para un estudio dedicado específico.

En nuestro caso, vamos a enfocarnos en aquellas labores que podemos realizar mediante comandos proporcionados por los S.O. más comunes para realizar una supervisión del estado de las comunicaciones de red en los propios equipos en los que se encuentran instalados.

Comandos para verificar la red

Aunque existen muchas herramientas de verificación y control de las conexiones de red, tanto propietarias como libres y gratuitas, en muchas ocasiones lo más rápido y asequible es recurrir a los propios comandos que nos ofrece el sistema operativo para estos fines.

Algunos de estos comandos son comunes y están disponibles en diferentes sistemas operativos (como el “ping” por ejemplo), y otros dependen del entorno (S.O.) que estemos usando. Al ser comandos a introducir a través de una consola/terminal, no tendrán entorno gráfico, y su respuesta será normalmente en modo texto en nuestra pantalla, pero su potencia y la valiosa información que aportan hacen muy recomendable ser capaz de trabajar con ellos.

En general, a través de comandos de red podremos obtener:

- **Información sobre la propia configuración** de red del equipo (dirección IP, máscara de subred, DNS, etc.).
- **Realizar pruebas de conectividad** y la capacidad para alcanzar destinos.
- **Verificar rutas** seguidas para llegar a un destino.
- Ver qué **unidades están siendo compartidas** en nuestro equipo.
- Hacer **pruebas de diagnóstico** sobre el funcionamiento de la red.

The screenshot shows a Windows Command Prompt window titled "Símbolo del sistema". The command entered is "C:\>ping www.telefonica.es". The output displays four successful ping responses from the IP 212.170.36.79 with TTL=241, followed by statistics: 4 packets sent, 4 received, 0 lost (0% loss), and a round-trip time of 11ms to 16ms with a mean of 13ms.

```
C:\>ping www.telefonica.es

Haciendo ping a www.telefonica.es [212.170.36.79] con 32 bytes de datos:
Respuesta desde 212.170.36.79: bytes=32 tiempo=12ms TTL=241
Respuesta desde 212.170.36.79: bytes=32 tiempo=11ms TTL=241
Respuesta desde 212.170.36.79: bytes=32 tiempo=16ms TTL=241
Respuesta desde 212.170.36.79: bytes=32 tiempo=13ms TTL=241

Estadísticas de ping para 212.170.36.79:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 11ms, Máximo = 16ms, Media = 13ms

C:\>
```

Más adelante veremos algunos ejemplos de estos comandos sobre sistemas operativos diferentes.

Sistemas operativos en red

En un sistema operativo en red los diferentes equipos (servidores y clientes) se mantienen “unidos” (conectados) a través de conexiones de red y comparten recursos hardware y software, por lo que las comunicaciones son un aspecto crucial para su funcionamiento.

Existen muchos sistemas operativos con esta funcionalidad; Windows Server, LAN Manager, Novell Netware, Unix, Linux, etc.

En muchos casos, cuando no hay un sistema en red (S.O. de servidor) y los equipos son autónomos, funcionando con su propio S.O. de forma independiente, entonces hablamos de “**grupo de trabajo**”.

Si existe un S.O. en red normalmente la gestión de los recursos se realiza de forma centralizada por el administrador, que además de encargarse de optimizar el aprovechamiento de los recursos, asignar permisos, etc., se encarga de la gestión de la propia red, esto es, monitorizar su funcionamiento, tomar medidas de seguridad y reaccionar frente a fallos o situaciones de congestión.

Un ejemplo de configuración de una red con dispositivos variados podría ser el siguiente:

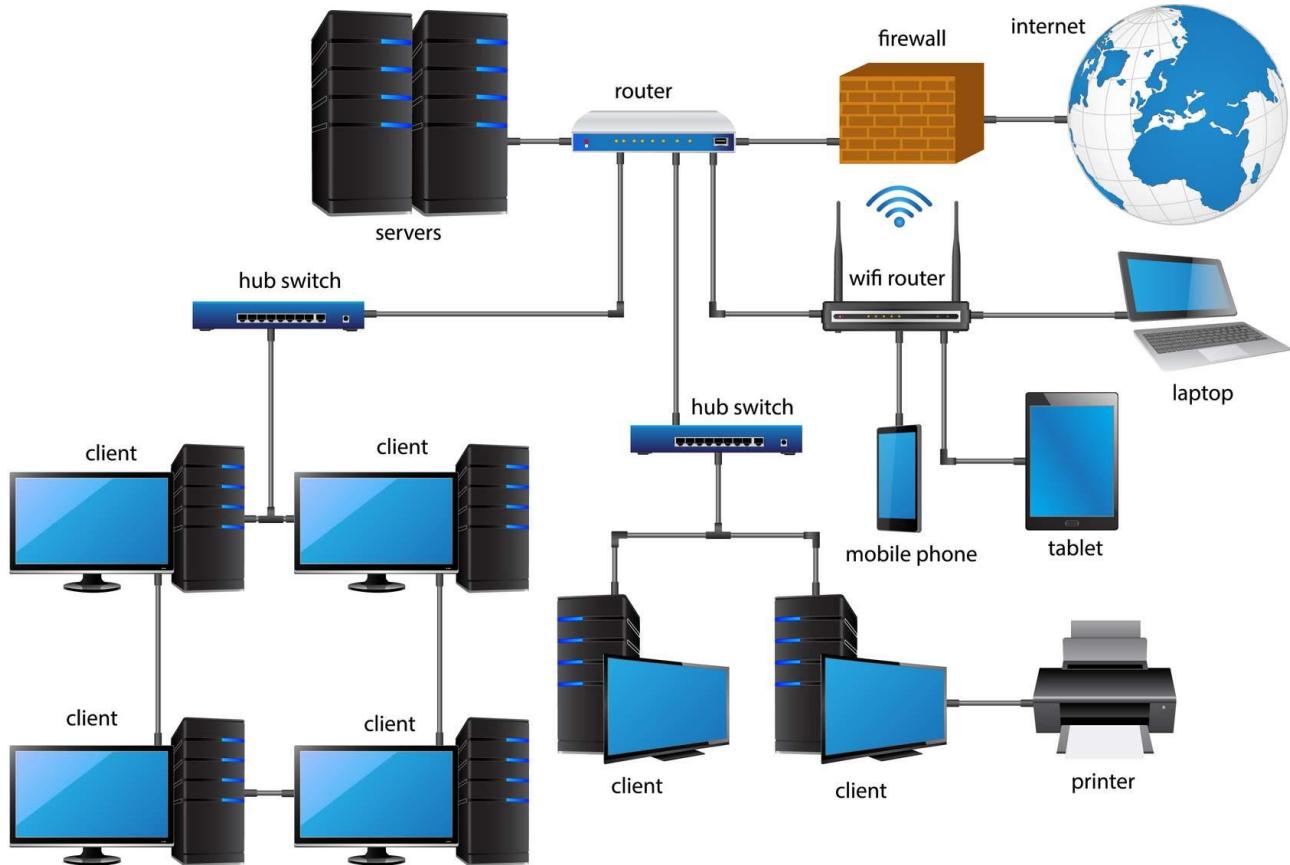


Diagrama de una red LAN.

Problemas de conectividad en la red

En general, e independientemente de las herramientas que utilicemos (programas o comandos de consola), al detectar problemas de conectividad en nuestra red debemos actuar diligentemente.

No suele ser muy conveniente comenzar a hacer cambios sin antes **detectar exactamente la causa del problema; la técnica de “ensayo-error” puede provocar más disgustos que soluciones.**

Por ello, de forma general, la secuencia recomendada sería:

1. **Localizar, aislar y comprobar el fallo/problema;** asegurarnos de que se produce y las condiciones en las que lo hace.
2. Realizar **pruebas de diagnóstico** para averiguar la causa del problema. Concretamente:
 - **Problemas hardware:** si falla alguna tarjeta de red, *router* o *switch*, etc.
 - **Problemas software:** alguna desconfiguración, conflicto de direcciones, protocolos incompatibles, etc.
 - **Problemas físicos:** problemas con el tendido de la red, cables, conectores, etc.
3. Definir la posible solución, y si hay varias, elegir una de ellas y reflejar el criterio empleado para la solución.
4. Hacer los cambios y modificaciones necesarias.
5. Comprobar la efectividad de los cambios o correcciones realizadas.
6. Realizar **pruebas de esfuerzo** sobre los equipos para asegurarnos de que funcionan en condiciones exigentes (evidentemente esto puede realizarse sin esperar a que ocurra un problema en la red).

Lo primero: localizar, aislar y comprobar el fallo o problema

Si estamos trabajando en un sistema operativo en red, con múltiples equipos clientes sobre uno o varios servidores, es posible (aunque no deseable) que para identificar la causa tengamos que reducir momentáneamente los servicios que están siendo utilizados.

Por ejemplo, podemos necesitar aislar un segmento de la red o desconectar algún equipo. En cualquier caso, **disponer de diagramas de red con la configuración física y lógica, es casi indispensable.**

Además, debemos intentar ubicar la causa del problema, al menos en el nivel al que afecta, y aunque no hay recetas "mágicas" debemos considerar, a modo de ejemplo:

- Si es un problema de **nivel físico** a menudo se pierde totalmente la conectividad o hay indisponibilidad total de alguna ruta.
- Si el problema es de **capa 2 (nivel de enlace)** podemos usar analizadores de red que detecten, por ejemplo, un elevado número de tramas con errores.
- Observar los mensajes en la consola durante los **procesos de arranque** puede darnos una idea de dónde radica el problema.
- En los problemas de **nivel de red** a menudo la causa está en una mala configuración o una degradación de las tablas de enrutamiento, problemas con el servidor DHCP (p. ej. direcciones insuficientes), o equipos conectados con una configuración inadecuada provocando conflictos de direcciones, etc.
- Si los problemas están a **nivel 4 o superiores** es común que exista una asignación incorrecta de puertos, o bien la comunicación funciona con unos servicios y no con otros.

Comandos de consola

A la hora de verificar el funcionamiento de las conexiones de red podemos utilizar, tanto herramientas gráficas del propio sistema operativo (o de terceros), como comandos de consola/terminal.

En el caso del uso de órdenes por consola/terminal, y sea cual sea el sistema en el que estemos trabajando, lo importante es **tener claro siempre qué es lo que buscamos** con el uso de un comando en particular, y no perder de vista el objetivo (problema) a resolver.

Ciertos comandos son comunes a distintos sistemas, algunos son bastante similares. Otros en cambio son totalmente diferentes entre los distintos entornos.

Vamos a centrarnos en los dos entornos más comunes y sobre los que existe más facilidad para realizar el entrenamiento, es decir, los sistemas **Windows** de Microsoft (aunque puede haber diferencias según la versión), y los sistemas **Linux** (sobre todo usando Ubuntu, que ya conoces), sabiendo que gran parte de lo visto para Linux será válido si nos movemos en entornos Unix.

El objetivo no es describir todos los comandos ni hacerlo en profundidad, sino aportar un conocimiento general que sirva al aprendizaje durante el estudio de esta unidad.



Comandos utilizados en Windows

Veamos algunos de los **comandos más comunes para Windows**, pulsa sobre cada una de las pestañas.

Ping

Quizás este sea el comando más sencillo para realizar pruebas de conectividad: **nos permite comprobar** que los protocolos de direccionamiento IP están funcionando y que somos capaces de “ver” (alcanzar) un determinado destino.

Ping (sin parámetros) envía cuatro pequeños paquetes de datos al destino y espera respuesta, registrando el tiempo que tarda en recibirse.

Podemos hacer “*ping*” sobre cualquier dirección IP o nombre de dominio, por ejemplo sobre la dirección “127.0.0.1” (*localhost* o dirección de *loopback*), y esto nos puede servir para verificar si la pila de protocolos TCP/IP está funcionando bien en nuestro equipo. También podemos hacer *ping*, por ejemplo, sobre la “puerta de enlace” (“*gateway*”) predeterminada, que en la instalación doméstica vía red local Ethernet suele ser el *router* de conexión a Internet (pero puede ser otra).

Veamos un ejemplo:

C:\ Símbolo del sistema

C:\Users\Alumno>ping 127.0.0.1

Haciendo ping a 127.0.0.1 con 32 bytes de datos:
Respuesta desde 127.0.0.1: bytes=32 tiempo<1ms TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1ms TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1ms TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1ms TTL=128

Estadísticas de ping para 127.0.0.1:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\Alumno>
C:\Users\Alumno>
C:\Users\Alumno>ping 192.168.43.1

Haciendo ping a 192.168.43.1 con 32 bytes de datos:
Respuesta desde 192.168.43.1: bytes=32 tiempo=3ms TTL=64
Respuesta desde 192.168.43.1: bytes=32 tiempo=6ms TTL=64
Respuesta desde 192.168.43.1: bytes=32 tiempo=3ms TTL=64
Respuesta desde 192.168.43.1: bytes=32 tiempo=5ms TTL=64

Estadísticas de ping para 192.168.43.1:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 3ms, Máximo = 6ms, Media = 4ms

C:\Users\Alumno>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=85ms TTL=54
Respuesta desde 8.8.8.8: bytes=32 tiempo=62ms TTL=54
Respuesta desde 8.8.8.8: bytes=32 tiempo=75ms TTL=54
Respuesta desde 8.8.8.8: bytes=32 tiempo=62ms TTL=54

Estadísticas de ping para 8.8.8.8:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos).

"ping" sobre el "localhost" (el propio equipo), con ello comprobamos que nuestra pila TCP/IP funciona.

Nos dice que se han enviado cuatro paquetes de 32 bits, que han sido devueltos con éxito. Para evitar que los paquetes viajen indefinidamente por la red se le asigna un "Time To Live" (TTL) que se decrementa en cada paso de un host a otro, en este caso al ser el mismo equipo local no se ha decrementado.

En este caso hacemos ping sobre el Gateway de nuestra conexión WIFI y nos da los tiempos min/max/medio que tardan los paquetes en regresar, y vemos también como se ha decrementado el TTL.

Con este ping podemos ver que somos capaces de alcanzar los servidores DNS de Google, y nos devuelve los paquetes de forma correcta.

Ipconfig

Comando muy útil que nos muestra **todos los datos de la configuración del protocolo TCP/IP** en el equipo (dirección IP asignada, máscara de subred, puerta de enlace, servidores DNS). Con la opción “/all” nos devuelve un informe detallado de la configuración de todas las interfaces y los puertos. Además nos permite renovar el servidor DHCP (en la configuración dinámica).

```
C:\Símbolo del sistema
C:\Users\Alumno>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet0:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::79a1:df5a:b344:9971%4
    Dirección IPv4. . . . . : 192.168.43.47
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.43.1

Adaptador de túnel Teredo Tunneling Pseudo-Interface:

    Sufijo DNS específico para la conexión. . . :
    Dirección IPv6 . . . . . : [2001:0:9d38:953c:2c7c:1b0e:3f57:d4d0]
    Vínculo: dirección IPv6 local. . . : fe80::2c7c:1b0e:3f57:d4d0%2
    Puerta de enlace predeterminada . . . . . : ::

C:\Users\Alumno>
```

Comando "ipconfig" que nos muestra la configuración IP de nuestro sistema Windows.

Direcciones IPV4 del equipo, máscara de subred y puerta de enlace predeterminada (Gateway).

Direcciones IPV6

Nslookup

Este comando se utiliza para **diagnosticar y solucionar problemas con los servidores DNS de una conexión del equipo**. Al insertar “nslookup” en la consola el sistema nos devuelve el nombre y la dirección IP del host que se encuentra configurado como servidor DNS local, y a continuación pasa a mostrar como “prompt” el signo “>” y si tecleamos “?” nos mostrará todas las posibles opciones del comando.

Una de las principales características de “nslookup” para resolver problemas con los DNS es su “modo depuración” (se invoca tecleando “set debug” o “set d2” con el prompt “>”). En dicho modo el programa nos devuelve informes con el detalle de los pasos que va siguiendo para ejecutar los comandos que le pedimos.

```
C:\Users\Alumno>nslookup
Servidor predeterminado: google-public-dns-a.google.com
Address: 8.8.8.8

> set debug
> telefonica.es
Servidor: google-public-dns-a.google.com
Address: 8.8.8.8

-----
Got answer:
HEADER:
    opcode = QUERY, id = 2, rcode = NOERROR
    header flags: response, want recursion, recursion avail.
    questions = 1, answers = 1, authority records = 0, additional = 0
    QUESTIONS:
        telefonica.es, type = A, class = IN
    ANSWERS:
    -> telefonica.es
        internet address = [212.170.36.79]
        ttl = 135 (2 mins 15 secs)

-----
Respuesta no autoritativa:
-----
Got answer:
HEADER:
    opcode = QUERY, id = 3, rcode = NOERROR
    header flags: response, want recursion, recursion avail.
    questions = 1, answers = 1, authority records = 0, additional = 0
    QUESTIONS:
        telefonica.es, type = AAAA, class = IN
    ANSWERS:
    -> telefonica.es
        AAAA IPv6 address = 2a02:9009:0:aa:aa01::
        ttl = 299 (4 mins 59 secs)

-----
Nombre: telefonica.es
Addresses: 2a02:9009:0:aa:aa01::
[212.170.36.79]
```

DNS configurado en las propiedades de la conexión IPV4

Le pedimos que nos muestre en detalle como se produce la traducción de este nombre y se accede a los DNS para obtener la dirección IP que nos devuelve luego (212,170,36,79)

Si la respuesta es “no autoritativa” lo que nos indica es que el DNS que ha respondido con la traducción no es el responsable de las direcciones de esa red sino que tiene una copia en caché de la traducción, fruto de otras consultas. Esto es lo que obtendremos en la mayoría de los casos porque los DNSs se guardan esta información y cuando no es necesario no “elevan” la petición al nivel superior.

Dirección IPV4 de “telefónica.es”, también nos da la IPV6

Netstat

El comando **netstat** es quizás uno de los más utilizados a la hora de verificar el funcionamiento de las conexiones de red. Netstat nos muestra todas las conexiones establecidas, los puertos y su estado, las aplicaciones que los utilizan, estadísticas, etc.

Entre sus opciones están, por ejemplo:

- **netstat -a**: muestra todas las conexiones y puertos abiertos.
- **netstat -n**: muestra puertos y direcciones en formato numérico.
- **netstat -b**: visualiza el programa que inició la conexión.
- **netstat -e**: visualiza estadísticas de datos enviados/recibidos.

```
C:\Users\Alumno>netstat
Símbolo del sistema
C:\Users\Alumno>netstat
Conexiones activas
Proto Dirección local     Dirección remota   Estado
TCP  192.168.43.47:50403  13.92.211.253:https ESTABLISHED
TCP  192.168.43.47:50436  152.199.20.1:https CLOSE_WAIT
TCP  192.168.43.47:50459  a104-126-85-80:https ESTABLISHED

C:\Users\Alumno>netstat -r
Símbolo del sistema
C:\Users\Alumno>netstat -r
ILista de interfaces
4...00 0c 29 59 fc 60 .....Intel(R) 82574L Gigabit Network Connection
1...00 00 00 00 00 00 e0 Teredo Tunnelling Pseudo-Interface
2...00 00 00 00 00 00 Software Loopback Interface 1

IPv4 Tabla de enrutamiento
Rutas activas:
Destino de red    Máscara de red    Puerta de enlace  Interfaz [Métrica]
0.0.0.0          0.0.0.0          192.168.43.1    192.168.43.47  25
127.0.0.0        255.0.0.0        En vínculo       127.0.0.1      331
127.0.0.1        255.255.255.255  En vínculo       127.0.0.1      331
127.255.255.255 255.255.255.255 En vínculo       127.0.0.1      331
192.168.43.0     255.255.255.0   En vínculo       192.168.43.47  281
192.168.43.47    255.255.255.255 En vínculo       192.168.43.47  281
192.168.43.255   255.255.255.255 En vínculo       192.168.43.47  281
224.0.0.0         240.0.0.0        En vínculo       127.0.0.1      331
224.0.0.0         240.0.0.0        En vínculo       192.168.43.47  281
255.255.255.255 255.255.255.255 En vínculo       127.0.0.1      331
255.255.255.255 255.255.255.255 En vínculo       192.168.43.47  281

Rutas persistentes:
Ninguno

IPv6 Tabla de enrutamiento
Rutas activas:
Cuando destino de red métrica    Puerta de enlace
2  331 ::/0                      En vínculo
1  331 ::1/128                   En vínculo
2  331 2001:::/32                 En vínculo
2  331 2001:0:9d38:953c:3c0e:1e67:3f57:d4d0/128 En vínculo
4  281 fe80::/64                 En vínculo
```

Arp

El comando **ARP** muestra y permite modificar las tablas de conversión de direcciones IP a direcciones que utiliza el protocolo de resolución de direcciones “ARP” (“Address Resolution Protocol”). El protocolo ARP se encarga de encontrar la dirección física que corresponde a una determinada dirección IP, y que se almacenan en una tabla en la caché del equipo para reducir el tiempo de consulta.

A menudo es útil para comprobar si podemos alcanzar un determinado equipo en nuestra red cuando existen direcciones duplicadas, etc.

```
C:\Users\Alumno>arp -a
Símbolo del sistema
C:\Users\Alumno>arp -a
Interfaz: 192.168.43.47 --- 0x4
Dirección de Internet      Dirección física  Tipo
192.168.43.1      18-f0-e4-3b-64-4c  dinámico
192.168.43.255    ff-ff-ff-ff-ff-ff  estático
224.0.0.22        01-00-5e-00-00-16  estático
224.0.0.251       01-00-5e-00-00-fb  estático
224.0.0.252       01-00-5e-00-00-fc  estático
239.255.255.250   01-00-5e-7f-ff-fa  estático
255.255.255.255   ff-ff-ff-ff-ff-ff  estático
```

Route

El comando “**route**” nos **muestra y manipula las tablas de enrutamiento de red** del equipo. Nos permite borrar entradas de las tablas de enrutamiento o, por ejemplo, forzar el uso de IPv4 o IPv6.

La forma sencilla de usarlo es escribir en la consola “**route print**” y pulsar “enter”, pero disponemos de muchas más opciones (para verlas, como de costumbre, podemos introducir “**route ?**”). En la figura podemos ver que el informe que nos da es prácticamente la misma información que con “**netstat**”.

```
C:\Users\Alumno>route print
=====
Lista de interfaces
 4...00 0c 29 59 fc 60 .....Intel(R) 82574L Gigabit Network Connection
 1........................Software Loopback Interface 1
 2...00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
=====

IPv4 Tabla de enrutamiento
=====
Rutas activas:
Destino de red      Máscara de red    Puerta de enlace  Interfaz   Métrica
  0.0.0.0            0.0.0.0          192.168.43.1    192.168.43.47  25
 127.0.0.0           255.0.0.0        En vínculo       127.0.0.1    331
 127.0.0.1           255.255.255.255  En vínculo       127.0.0.1    331
127.255.255.255  255.255.255.255  En vínculo       127.0.0.1    331
 192.168.43.0        255.255.255.0  En vínculo       192.168.43.47  281
 192.168.43.47      255.255.255.255  En vínculo       192.168.43.47  281
 192.168.43.255     255.255.255.255  En vínculo       192.168.43.47  281
  224.0.0.0           240.0.0.0        En vínculo       127.0.0.1    331
  224.0.0.0           240.0.0.0        En vínculo       192.168.43.47  281
 255.255.255.255  255.255.255.255  En vínculo       127.0.0.1    331
 255.255.255.255  255.255.255.255  En vínculo       192.168.43.47  281
=====
Rutas persistentes:
 Ninguno

IPv6 Tabla de enrutamiento
=====
Rutas activas:
Cuando destino de red métrica      Puerta de enlace
 2  331 ::/0                      En vínculo
 1  331 ::1/128                   En vínculo
 2  331 2001:::/32                En vínculo
 2  331 2001::0:9d38:953c:3c0e:1e67:3f57:d4d0/128
                                         En vínculo
 4  281 fe80::/64                 En vínculo
 2  331 fe80::/64                 En vínculo
 2  331 fe80::3c0e:1e67:3f57:d4d0/128
                                         En vínculo
 4  281 fe80::79a1:df5a:b344:9971/128
                                         En vínculo
 1  331 ff00::/8                  En vínculo
 4  281 ff00::/8                  En vínculo
 2  331 ff00::/8                  En vínculo
```

Tracert

Este comando nos **permite visualizar el camino que sigue un determinado paquete en la red** y si su avance se ha detenido en algún punto antes de llegar al destino.

Tracert determina la ruta seguida para alcanzar un determinado host mediante el envío de paquetes “eco” de protocolo **ICMP** (con un valor de tiempo de vida “TTL” – “Time To Live”) a la dirección de destino. En cada nodo de conmutación por el que pasan el *router* disminuirá (al menos en una unidad) el valor de TTL antes de reenviarlo, y si llega a cero antes de alcanzar el destino se devuelve al origen el mensaje “tiempo agotado”.

El comando **Tracert** nos muestra una lista de todos los *router* que han enviado el mensaje al origen, de forma que podemos ver “hasta dónde llega” y “por dónde pasan” nuestros paquetes IP dirigidos a una determinada dirección de destino.

```
C:\Users\Alumno>tracert www.microsoft.com
Traza a la dirección e13678.dsdp.akamaiedge.net [23.211.9.92] <-- Dashed line from destination
sobre un máximo de 30 saltos:
  1  *      *      *      Tiempo de espera agotado para esta solicitud.
  2  *      *      *      Tiempo de espera agotado para esta solicitud.
  3  86 ms   53 ms   57 ms  10.7.48.5
  4  58 ms   46 ms   57 ms  10.7.32.120
  5  109 ms   73 ms   57 ms  10.7.32.133
  6  100 ms   65 ms   49 ms  bcn-b2-link.telia.net [80.239.167.13]
  7  91 ms   60 ms   65 ms  mad-b1-link.telia.net [62.115.125.209]
  8  96 ms   72 ms   60 ms  vodafone-ic-309718-mad-b1.c.telia.net [80.239.128.182]
  9  97 ms   94 ms   90 ms  ae3-xcr1.mat.cw.net [195.2.30.90]
 10  116 ms   96 ms   88 ms  195.2.31.245
 11  105 ms   78 ms   86 ms  ae3-xcr2.fra.cw.net [195.2.10.209]
 12  95 ms   87 ms   94 ms  ae3-xcr1.dus.cw.net [195.2.25.9]
 13  272 ms   628 ms  216 ms  akamaaint-gw-xcr1.dus.cw.net [195.2.15.122]
 14  132 ms   98 ms   88 ms  a23-211-9-92.deploy.static.akamaitechnologies.com [23.211.9.92] <-- Dashed line from destination

Traza completa.

C:\Users\Alumno>
```

Con "tracert" podemos ver los puntos (enrutadores) por los que pasa la conexión hasta alcanzar el destino. El comando funciona enviando paquetes de protocolo ICMP hacia el destino, y pidiendo a los router intermedios que respondan cuando pasen por ellos (puede ser que alguno no responda y entonces nos aparece el mensaje de "tiempo agotado para esta solicitud" pero luego recibimos los siguientes). Al final, si todo va bien, alcanzaremos la IP del destino.

Netsh

Más que un comando es una verdadera utilidad por línea de comandos que nos permite, por ejemplo, **guardar la configuración de red actual** en un fichero (.cmp) y **cargarla** posteriormente o **exportarla** a otro equipo.

Netsh no es un comando fácil de entender. Para empezar, trabaja con "**contextos**", que son aspectos específicos de la configuración de red y que, al seleccionarlos, las acciones del comando devuelven diferentes resultados.

```
C:\Users\Alumno>netsh dump > c:\users\alumno\config_de_red.txt
C:\Users\Alumno>netsh
netsh>

Los siguientes comandos están disponibles:
Comandos en este contexto:
..           - Sube un nivel de contexto.
?            - Muestra una lista de comandos.
abort       - Descarta los cambios realizados estando en modo de configuración.
add         - Agrega una entrada de configuración a una interfaz.
advfirewall - Cambia al contexto 'netsh advfirewall'.
alias      - Agrega un alias.
branchcache - Cambia al contexto 'netsh branchcache'.
bridge     - Cambia al contexto 'netsh bridge'.
bye        - Sale del programa.
commit     - Confirma los cambios realizados en el modo de configuración.
delete     - Elimina una entrada de configuración de una interfaz.
dhcpclient - Cambia al contexto 'netsh dhcpclient'.
dnsclient  - Cambia al contexto 'netsh dnsclient'.
dump       - Muestra un script de configuración.
exec      - Ejecuta un archivo de script.
exit      - Sale del programa.
firewall   - Cambia al contexto 'netsh firewall'.
help       - Muestra una lista de comandos.
http       - Cambia al contexto 'netsh http'.
interface  - Cambia al contexto 'netsh interface'.
ipsec     - Cambia al contexto 'netsh ipsec'.
lan        - Cambia al contexto 'netsh lan'.
mbn        - Cambia al contexto 'netsh mbn'.
namespace  - Cambia al contexto 'netsh namespace'.
netio     - Cambia al contexto 'netsh netio'.
offline   - Establece el modo actual a Sin conexión.
online    - Establece el modo actual a En linea.
p2p       - Cambia al contexto 'netsh p2p'.
popd     - Extrae un contexto de la pila.
pushd    - Inserta el contexto actual en la pila.
quit      - Sale del programa.
ras        - Cambia al contexto 'netsh ras'.
rpc       - Cambia al contexto 'netsh rpc'.
set       - Actualiza las opciones de configuración.
show     - Muestra información.
trace    - Cambia al contexto 'netsh trace'.
```

config_de_red.txt: Bloc de notas

```
Archivo Edición Formato Ver Ayuda
=====
# Configuraci&on de interfaz
=====
pushd interface

popd
# Fin de la configuraci&on de interfaz

# -----
# Configuraci&on de IPHTTPS
# -----
pushd interface httpstunnel

reset

popd
# Fin de la configuraci&on de IPHTTPS

# -----
# Configuraci&on de IPv4
# -----
pushd interface ipv4

reset
set global icmpredirects=enabled
set interface interface="Ethernet (depurador de kernel)" forwarding=enabled advertise=enabled nud=enabled
set interface interface="Ethernet0" forwarding=enabled advertise=enabled nud=enabled ignoredefaultroute
```

Comandos "Net"

En realidad es un grupo de comandos, a través de los cuales podemos visualizar la red, los servidores disponibles y los grupos de trabajo a los que estamos conectados, los recursos que se encuentran compartidos, visualizar estadísticas, etc.

Para saber las opciones disponibles (varían de un sistema a otro) podemos teclear “net help” en la consola de comandos.

```
C:\Users\Alumno>net accounts
Tiempo antes del cierre forzado: Nunca
Duración mín. de contraseña (días): 0
Duración máx. de contraseña (días): 42
Longitud mínima de contraseña: 0
Duración del historial de contraseña: Ninguna
Umbral de bloqueo: Nunca
Duración de bloqueo (minutos): 30
Ventana de obs. de bloqueo (minutos): 30
Rol del servidor: ESTACION DE TRABAJO
Se ha completado el comando correctamente.

C:\Users\Alumno>net config workstation
Nombre del equipo \\DESKTOP-6PA7CHG
Nombre completo de equipo DESKTOP-6PA7CHG
Nombre de usuario Alumno

Estación de trabajo activa en NetBT_Tcpip_{5421EAAA-9F75-4C48-8FB2-C8A5F0093015} (000C2959FC60)

Versión del programa Windows 10 Enterprise Evaluation
Dominio de estación de trabajo WORKGROUP
Dominio de inicio de sesión DESKTOP-6PA7CHG

Tiempo de espera de COM (s) 0
Cuenta de envío de COM (bytes) 16
Tiempo de envío en COM (ms.) 250
Se ha completado el comando correctamente.

C:\Users\Alumno>
```

Con la opción “accounts” nos da información sobre la cuenta y los parámetros de duración de la contraseña, etc.

Nos muestra información sobre el propio equipo y el usuario que tiene activa la sesión, la versión del S.O., etc.

Comandos utilizados en Linux

Ahora veamos algunos de los **comandos más comunes para Linux**, pulsa sobre cada una de las pestañas.

Al igual que en entorno Windows, en Linux existen comandos muy útiles a la hora de verificar el funcionamiento de la red y sobre los que podemos combinar todas las opciones de gestión por terminal del sistema operativo. Pincha con el ratón sobre las pestañas para ver algunos ejemplos.

Hostname

Conocer el nombre de nuestro equipo (*host*) nunca fue más fácil, basta con teclear en el terminal el comando:

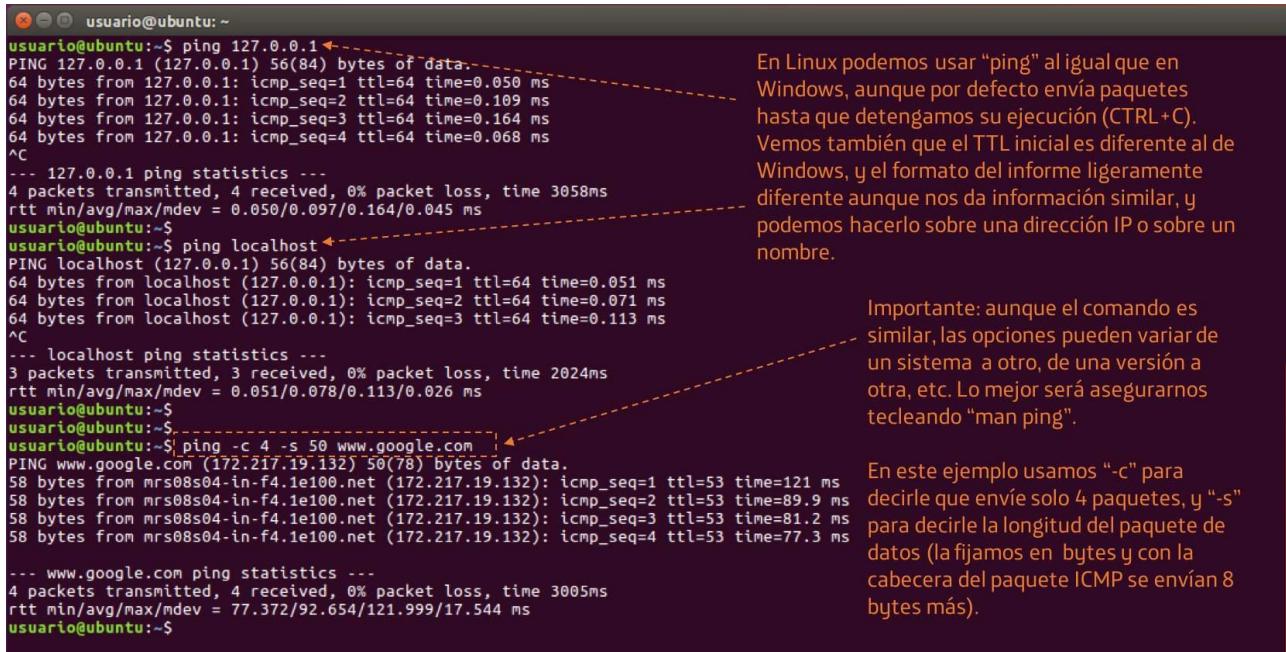
hostname

```
usuario@ubuntu: ~
usuario@ubuntu:~$ hostname
ubuntu
usuario@ubuntu:~$
```

Ping

Como habrás visto, este comando es común a ambos entornos (Windows y Linux/Unix), y la información que proporciona es prácticamente la misma. En entorno Linux el comando no envía solamente cuatro paquetes, sino que sigue enviándolos hasta que detenemos el proceso.

El uso y la funcionalidad es idéntica: envía datagramas de protocolo ICMP y espera a que el otro host le conteste. Podemos utilizar un nombre de dominio o directamente una dirección IP.



```

usuario@ubuntu:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.109 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.164 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.068 ms
^C
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3058ms
rtt min/avg/max/mdev = 0.050/0.097/0.164/0.045 ms
usuario@ubuntu:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.051 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.071 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.113 ms
^C
--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2024ms
rtt min/avg/max/mdev = 0.051/0.078/0.113/0.026 ms
usuario@ubuntu:~$ 
usuario@ubuntu:~$ ping -c 4 -s 50 www.google.com
PING www.google.com (172.217.19.132) 50(78) bytes of data.
58 bytes from mrs08s04-in-f4.1e100.net (172.217.19.132): icmp_seq=1 ttl=53 time=121 ms
58 bytes from mrs08s04-in-f4.1e100.net (172.217.19.132): icmp_seq=2 ttl=53 time=89.9 ms
58 bytes from mrs08s04-in-f4.1e100.net (172.217.19.132): icmp_seq=3 ttl=53 time=81.2 ms
58 bytes from mrs08s04-in-f4.1e100.net (172.217.19.132): icmp_seq=4 ttl=53 time=77.3 ms
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 77.372/92.654/121.999/17.544 ms
usuario@ubuntu:~$ 
```

En Linux podemos usar "ping" al igual que en Windows, aunque por defecto envía paquetes hasta que detengamos su ejecución (CTRL+C). Vemos también que el TTL inicial es diferente al de Windows, y el formato del informe ligeramente diferente aunque nos da información similar, y podemos hacerlo sobre una dirección IP o sobre un nombre.

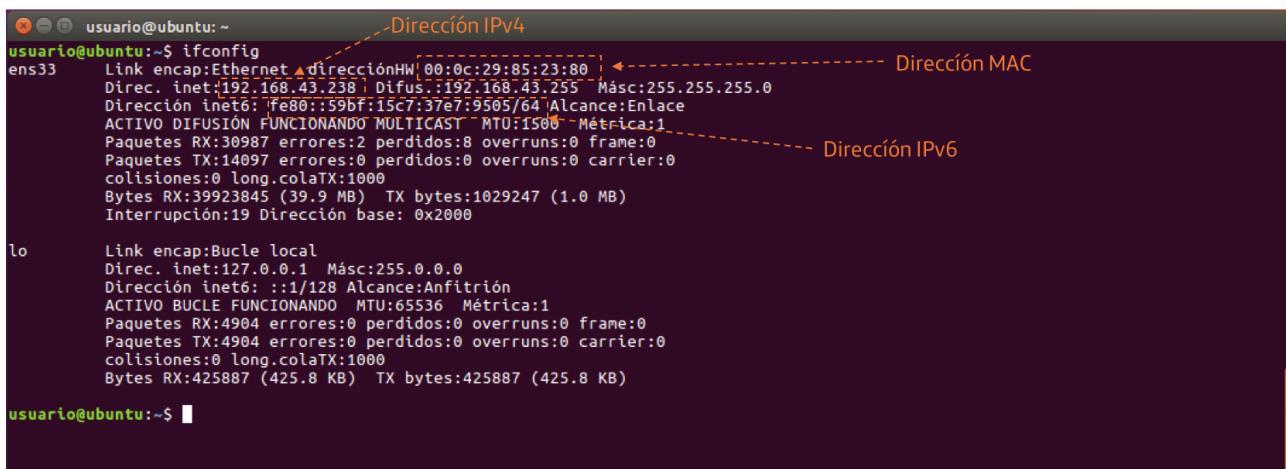
Importante: aunque el comando es similar, las opciones pueden variar de un sistema a otro, de una versión a otra, etc. Lo mejor será asegurarnos tecleando "man ping".

En este ejemplo usamos "-c" para decirle que envíe solo 4 paquetes, y "-s" para decirle la longitud del paquete de datos (la fijamos en bytes y con la cabecera del paquete ICMP se envían 8 bytes más).

Ifconfig

Ifconfig se usa para configurar las interfaces de red del *kernel*. Si lo introducimos sin parámetros nos visualiza el estado de las interfaces actuales, la dirección MAC, etc.

Como en todos los sistemas Unix/Linux, podemos obtener ayuda sobre el comando tecleando "*man ifconfig*". ¡Ojo!, aunque es muy parecido no debemos confundirlo con el "*ipconfig*" de Windows.



```

usuario@ubuntu:~$ ifconfig
ens3 Link encap:Ethernet HWaddr 00:0c:29:85:23:80
      Direc. inet:192.168.43.238 Brd:192.168.43.255 Másc:255.255.255.0
      Dirección inet6: fe80::59bf:15c7:37e7:9505/64 Alcance:Enlace
      ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
      Paquetes RX:30987 errores:2 perdidos:8 overruns:0 frame:0
      Paquetes TX:14097 errores:0 perdidos:0 overruns:0 carrier:0
      colisiones:0 long.colatTX:1000
      Bytes RX:39923845 (39.9 MB) TX bytes:1029247 (1.0 MB)
      Interrupción:19 Dirección base: 0x2000

lo Link encap:Bucle local
      Direc. inet:127.0.0.1 Másc:255.0.0.0
      Dirección inet6: ::1/128 Alcance:Anfitrión
      ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
      Paquetes RX:4904 errores:0 perdidos:0 overruns:0 frame:0
      Paquetes TX:4904 errores:0 perdidos:0 overruns:0 carrier:0
      colisiones:0 long.colatTX:1000
      Bytes RX:425887 (425.8 KB) TX bytes:425887 (425.8 KB)

usuario@ubuntu:~$ 
```

Ip route / ip route show

Con este comando conoceremos la configuración de las rutas y la puerta de enlace. Como sabemos, si tenemos destreza en el manejo de comandos podemos usar la potencia del sistema, por ejemplo, si queremos que nos muestre la línea en la que aparece la opción “*default*” debemos poner “*ip route show | grep default*”. En este ejemplo no es demasiado útil porque el resultado del comando es corto, pero sí será conveniente aprender estos “trucos” cuando los informes tengan un gran contenido.

```

usuario@ubuntu:~$ ip route
default via 192.168.43.1 dev ens33 proto static metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
192.168.43.0/24 dev ens33 proto kernel scope link src 192.168.43.238 metric 100
usuario@ubuntu:~$ ip route | grep default
default via 192.168.43.1 dev ens33 proto static
usuario@ubuntu:~$ 

```

Ruta por defecto
Origen por defecto para los envíos (IP del equipo)

Red

Toda la configuración Red

Cableada
Conectado - 10 Mb/s

Dirección física 00:0C:29:85:23:80
Dirección IPv4 192.168.43.238
Dirección IPv6 fe80::59bf:15c7::37e7:9505
Ruta predeterminada 192.168.43.1
DNS 192.168.43.1

Opciones...

Netstat

De nuevo nos encontramos con un comando común tanto en Windows como en Linux. “*Netstat*” nos permite visualizar todas las conexiones de red del equipo, incluyendo los *sockets* TCP, UDP y Unix (conectados o en espera).

```

usuario@ubuntu:~$ netstat -rn
Por defecto, los paquetes se enviarán al Gateway 192.168.43.1
Esta ruta es la que utiliza el Gateway (UG)
Tabla de rutas IP del Núcleo
Destino     Pasarela      Gmask   Indic   MSS Ventana irtt Interfaz
0.0.0.0     192.168.43.1  0.0.0.0   UG        0 0          0 ens33
169.254.0.0 0.0.0.0     255.255.0.0 U          0 0          0 ens33
192.168.43.0 0.0.0.0     255.255.255.0 U          0 0          0 ens33
usuario@ubuntu:~$ netstat -putn
(No todos los procesos pueden ser identificados, no hay información de propiedad del proceso
no se mostrarán, necesita ser superusuario para verlos todos.)
Conexiones activas de Internet (servidores w/o)
Proto Recib Enviad Dirección_local      Dirección remota      Estado      PID/Program name
tcp    0      0 192.168.43.238:49862  172.217.19.131:443  ESTABLECIDO 3651/firefox
tcp    0      0 192.168.43.238:40264  216.58.214.174:443  ESTABLECIDO 3651/firefox
tcp    0      0 192.168.43.238:60178  172.217.19.132:443  ESTABLECIDO 3651/firefox
tcp    0      0 192.168.43.238:47382  216.58.214.162:443  ESTABLECIDO 3651/firefox
tcp    0      0 192.168.43.238:55626  172.217.19.131:80  TIME_WAIT -
tcp    0      0 192.168.43.238:44092  216.58.214.163:443  ESTABLECIDO 3651/firefox
usuario@ubuntu:~$ netstat -putn
(No todos los procesos pueden ser identificados, no hay información de propiedad del proceso
no se mostrarán, necesita ser superusuario para verlos todos.)
Conexiones activas de Internet (servidores w/o)
Proto Recib Enviad Dirección local      Dirección remota      Estado      PID/Program name
tcp    0      0 :ubuntu:49862       par03s12-in-f131::https ESTABLECIDO 3651/firefox
tcp    0      0 :ubuntu:40264       nad01s26-in-f174::https ESTABLECIDO 3651/firefox
tcp    0      0 :ubuntu:60178       mrs08s04-in-f4.1e::https ESTABLECIDO 3651/firefox
tcp    0      0 :ubuntu:47382       nad01s26-in-f162::https ESTABLECIDO 3651/firefox
tcp    0      0 :ubuntu:55626       par03s12-in-f131.1:http TIME_WAIT -
tcp    0      0 :ubuntu:44092       nad01s26-in-f163::https ESTABLECIDO 3651/firefox

```

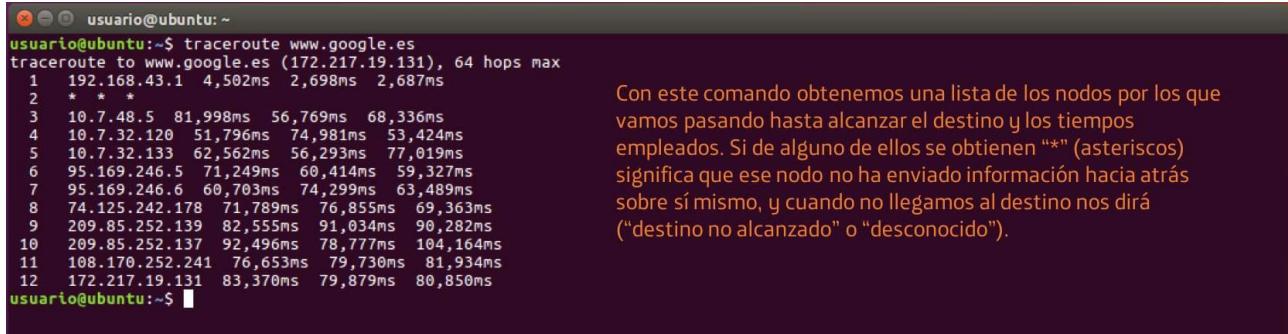
Para los destinos de estas redes no se usará el Gateway (son destinos locales), y tienen la interfaz “levantada” (U)

Desde nuestro equipo local (Ubuntu = 192.168.43.238) establecemos conexiones con un destino distante (172.217.19.131 = google.com) con nuestro navegador Firefox.

En función de los parámetros del mensaje podemos obtener diferente información.

Traceroute

Viene a ser equivalente al “*tracert*” de Windows. Permite mostrar todos los *hosts* por donde pasa un paquete en la red hasta llegar a su destino. Su sintaxis sería la que vemos en la imagen, aunque también podemos usar la IP, como ya sabemos.



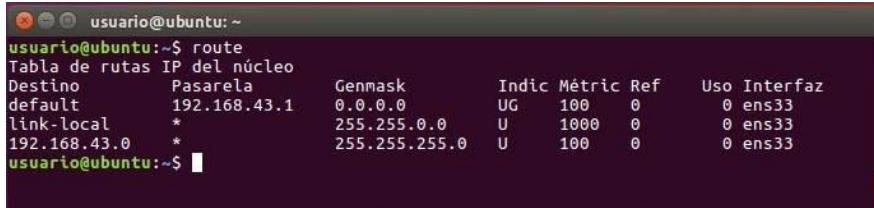
```
usuario@ubuntu:~$ traceroute www.google.es
traceroute to www.google.es (172.217.19.131), 64 hops max
 1  192.168.43.1  4,502ms  2,698ms  2,687ms
 2  *  *  *
 3  10.7.48.5  81,998ms  56,769ms  68,336ms
 4  10.7.32.120  51,796ms  74,981ms  53,424ms
 5  10.7.32.133  62,562ms  56,293ms  77,019ms
 6  95.169.246.5  71,249ms  60,414ms  59,327ms
 7  95.169.246.6  60,703ms  74,299ms  63,489ms
 8  74.125.242.178  71,789ms  76,855ms  69,363ms
 9  209.85.252.139  82,555ms  91,034ms  98,282ms
10  209.85.252.137  92,496ms  78,777ms  104,164ms
11  108.170.252.241  76,653ms  79,730ms  81,934ms
12  172.217.19.131  83,370ms  79,879ms  80,850ms
usuario@ubuntu:~$
```

Con este comando obtenemos una lista de los nodos por los que vamos pasando hasta alcanzar el destino y los tiempos empleados. Si de alguno de ellos se obtienen “*” (asteriscos) significa que ese nodo no ha enviado información hacia atrás sobre sí mismo, y cuando no llegamos al destino nos dirá (“destino no alcanzado” o “desconocido”).

Route

Comando que nos permite visualizar y manipular la información (añadir o eliminar registros) de las tablas de enrutamiento del equipo. Como habrás observado, es otro comando común con los sistemas de otros entornos como Windows, y su salida puede coincidir con la del comando *netstat*. Por ejemplo, si queremos añadir una determinada dirección IP como puerta de enlace a través de la conexión de red local, podríamos teclear:

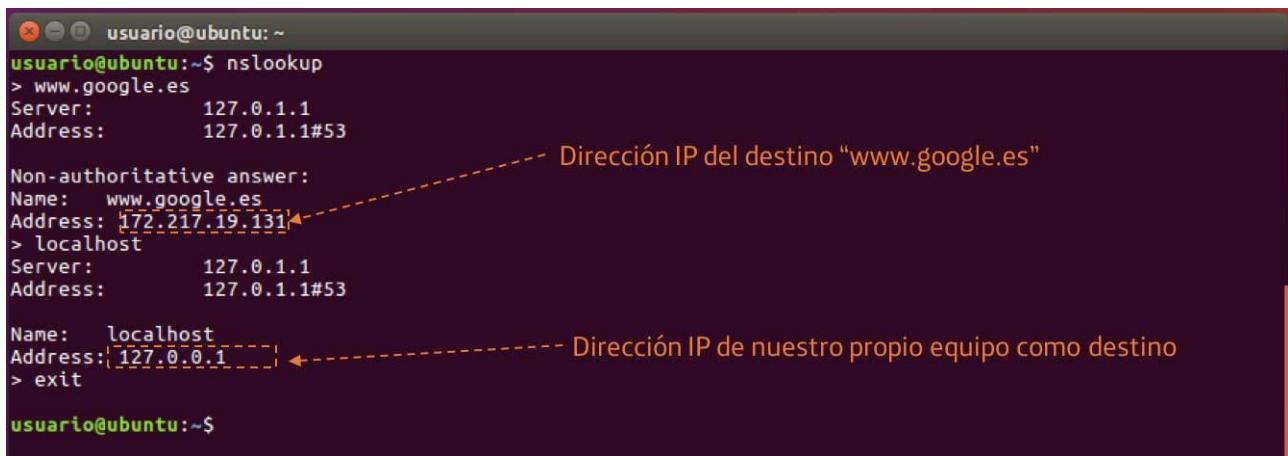
```
route add default gw 192.168.1.2 eth0
```



```
usuario@ubuntu:~$ route
Tabla de rutas IP del núcleo
Destino      Pasarela        Genmask     Indic Métric Ref    Uso Interfaz
default      192.168.43.1   0.0.0.0     UG      100    0        0 ens33
link-local   *               255.255.0.0  U       1000   0        0 ens33
192.168.43.0 *               255.255.255.0 U       100    0        0 ens33
usuario@ubuntu:~$
```

Nslookup

También existe en entorno Windows, y al igual que allí, el comando “*nslookup*” nos sirve para comprobar la coherencia de las tablas de traducción y ver si la traducción de la dirección IP al nombre es correcta. *Nslookup* nos permite interrogar a los DNS y nos devuelve la conversión de un nombre a IP o viceversa.



```
usuario@ubuntu:~$ nslookup
> www.google.es
Server:      127.0.1.1
Address:     127.0.1.1#53
Non-authoritative answer:
Name:   www.google.es
Address: 172.217.19.131
> localhost
Server:      127.0.1.1
Address:     127.0.1.1#53
Name:   localhost
Address: 127.0.0.1
> exit
usuario@ubuntu:~$
```

Dirección IP del destino “www.google.es”

Dirección IP de nuestro propio equipo como destino

Arp

ARP ("Address Resolution Protocol") es un protocolo de resolución de direcciones de la capa de enlace que se utiliza para conocer la dirección MAC (Ethernet) a la que corresponde una dirección IP. Funciona enviando un paquete de difusión ("ARP request") por la red (a la dirección de broadcast, MAC = FF FF FF FF FF FF) preguntando quién tiene la IP que buscamos, y la máquina que la contiene responderá desde su dirección MAC. Se establece entonces la correspondencia entre dirección "física" MAC y dirección IP, y con ello se mantiene una tabla en caché con las direcciones traducidas.

```

usuario@ubuntu: ~
usuario@ubuntu:~$ arp
Dirección          TipoHW  DirecciónHW      Indic Máscara     Interfaz
[192.168.43.1]    ether   [18:f0:e4:3b:64:4c] [C]  ens33
usuario@ubuntu:~$ arp -a
?:(192.168.43.1) en [18:f0:e4:3b:64:4c] [ether] en ens33
usuario@ubuntu:~$ usuario@ubuntu:~$
```

Dirección IP

Dirección MAC de la tarjeta Ethernet

C = entrada completa en la tabla de la caché ARP
M = entradas permanentes
P = entradas publicadas

Ejercicio: practicar con ping

Te proponemos un sencillo ejercicio con tus dos máquinas virtuales de Windows 10 y Ubuntu.

El ejercicio consiste en arrancar las dos máquinas virtuales al mismo tiempo y comprobar la conectividad entre ellas y con la máquina física, como si fueran tres equipos conectados en la misma red.

Requisitos:

- Las máquinas virtuales deben estar configuradas en modo "bridge" (más abajo te mostramos cómo hacerlo).
- El *firewall* de Windows debe estar desactivado (en la máquina física y en la máquina virtual) para que los paquetes ICMP que usa "ping" puedan ser recibidos y contestados.

Nota: si ves que tu equipo va muy lento arrancando las dos máquinas virtuales al mismo tiempo puedes hacer el ejercicio arrancando primero una, realizando la prueba y luego haciendo lo mismo con la otra, todo ello desde el equipo físico.

Secuencia a seguir

Te lo mostramos en el siguiente [vídeo](#). Lo que vamos a hacer es simplemente lo siguiente:

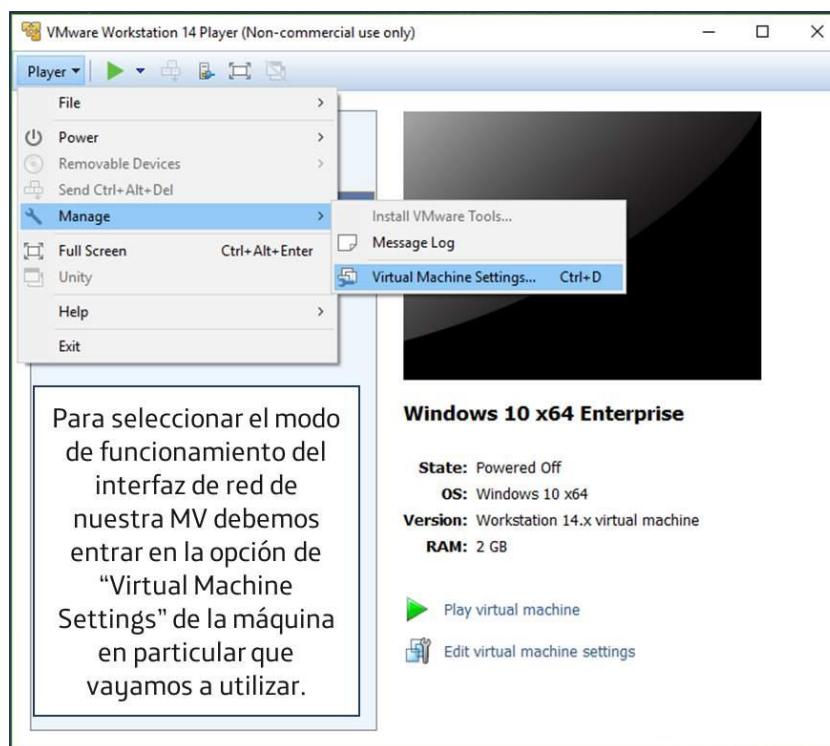
1. Visualizamos las direcciones IP de los tres sistemas, la máquina física y las dos máquinas virtuales. En los sistemas Windows usamos el comando "ipconfig" y en Linux el comando "ifconfig".
2. Usamos el comando "ping" para comprobar que desde cada sistema podemos ver/alcanzar los otros dos.

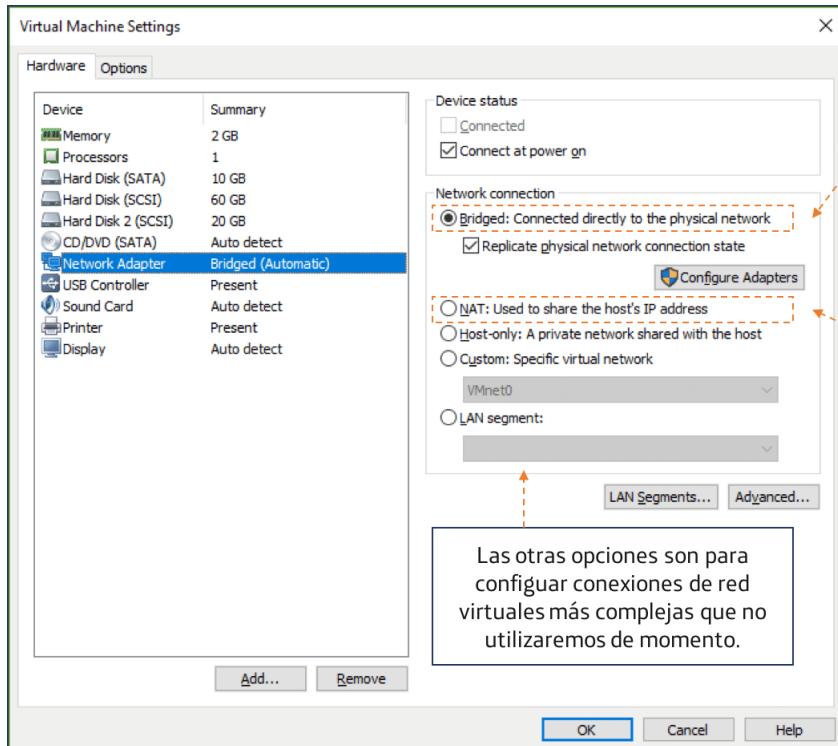
Nota: un detalle importante que debes tener en cuenta es que al mostrar la configuración IP de tu equipo físico usando el comando "ipconfig", en el resultado también aparecen los adaptadores de red virtuales de VMWare (VWnet1 y VWnet8 en el vídeo que acabas de ver).

Recuerda que es importante al acabar las prácticas volver a activar el *firewall* de Windows, sobre todo en tu máquina física.

¿Por qué debemos tener las máquinas virtuales en modo "*bridged*"?

El software de virtualización de VMWare nos ofrece diferentes modos de configuración de los adaptadores de red virtuales. En esta práctica necesitamos ponerlos en modo *bridged* para que funcionen como si fueran diferentes equipos conectados a la misma red. Te lo mostramos en un par de imágenes:





En modo "Bridged" nuestra máquina virtual funciona como si tuviera un interfaz directo (tarjeta de red) conectado a la red y obtiene una dirección IP independiente.

En modo "NAT" nuestra máquina virtual funciona utilizando para la conexión a internet el mismo interfaz y dirección IP de nuestro equipo físico, es como si se conectase al equipo físico y éste a la red.

Monitorización de redes

Monitorización pasiva/reactiva y monitorización activa

Monitorizar la red es realizar una verificación sistemática de su funcionamiento, analizando su rendimiento y la disponibilidad de los equipos y servicios que se prestan a través de ella, detectando y corrigiendo los problemas que puedan surgir.

En general podemos hablar de **dos tipos de monitorización**: monitorización pasiva/reactiva y monitorización activa.

Los parámetros a monitorizar pueden ser muy variados: desde la utilización del ancho de banda, el consumo de CPU y memoria en los *host*, el estado físico de las conexiones, los tipos de tráfico que están siendo gestionados en los nodos de conmutación (*routers*, *switches*, *hubs*, *firewalls*, etc.), y el estado de disponibilidad de los servicios en la red (correo, web, bases de datos, etc.).

Monitorización pasiva/reactiva

Se basa fundamentalmente en la **observación de la red sin modificar sus parámetros** de funcionamiento, a no ser como respuesta a un problema.

Hacemos este tipo de monitorización cuando usamos dispositivos para ver (capturar) el tráfico circulando por la red (por ejemplo “**sniffers**” o estadísticas de los **router o switches**). Con este tipo de verificación no estamos aumentando el tráfico que circula por la red, y simplemente observamos su funcionamiento cuando está en servicio.

Podríamos incluir aquí la programación de alarmas y eventos de disparo/aviso (“**triggers**”), la fijación de umbrales de funcionamiento que al ser excedidos provoquen una comunicación al administrador, etc.

Monitorización activa

Hablamos de “monitorización activa” **cuando realizamos acciones sobre la red para observar su respuesta**. Por ejemplo, se puede insertar **tráfico con paquetes de prueba** dirigidos a determinados *hosts* o servidores y medir el tiempo de transmisión y/o respuesta de un determinado servicio. En estas mediciones es importante tener claro si un posible retardo se debe a un funcionamiento pobre de la red o de un equipo/servidor concreto, es decir, definir claramente los parámetros a medir y cómo medirlos para descubrir la posible causa raíz de cualquier fallo.

Este tipo de monitorización aumenta la carga de tráfico de la red, pero permite la realización de pruebas muy dirigidas a la verificación de los parámetros que se desean medir. Es importante tener en cuenta que la realización de las pruebas no sobrecargue la red por encima de su umbral de trabajo para no provocar situaciones de funcionamiento anómalo, salvo por ejemplo que queramos precisamente realizar “pruebas de esfuerzo” y llevar al límite a la propia red.

Pasos a seguir en la monitorización de la red

Los pasos a seguir son sencillos, aunque no siempre les prestamos la atención adecuada. Antes de ponernos a tomar “medidas” sobre la red, debemos tener claros los objetivos a conseguir y cómo hacerlo.

Los pasos propuestos serían:

1. Establecer **qué es lo que necesitamos monitorizar** y qué información obtener.
Monitorizamos básicamente:
 - **Sistemas y servicios:** si están disponibles o no.
 - **Recursos:** su disponibilidad y estado.
 - **Rendimiento:** por ejemplo un tiempo de realización, unidades procesadas por unidad de tiempo, etc.
 - **Cambios y configuraciones:** evolución sobre la configuración inicial.
2. Definir **cómo vamos a obtener esa información** y valorar el impacto que podemos ocasionar sobre la red y los servicios en funcionamiento, y también el esfuerzo necesario por nuestra parte.
3. Construir los **mecanismos de monitorización** (comandos, instalación de herramientas, etc.).
4. Realizar la monitorización.
5. **Procesar la información** obtenida, analizarla y tomar las decisiones de gestión pertinentes.

Información de monitorización

La información de monitorización puede ser muy variada y extensa, en función del método que empleemos, y podrá presentarse de forma gráfica o bien en forma de texto o datos "planos".

Distinguiremos entre la **información “estática”** (que no cambia en un plazo de tiempo corto), como puede ser la configuración de la red, las direcciones IP fijas de los servidores, etc., y aquella **información “dinámica”** que está continuamente cambiando o evolucionando, normalmente relacionada con el uso de los elementos de la red, como puede ser el volumen de tráfico transmitido, los destinos de los mensajes, etc.

A partir de la información dinámica obtenida de la red podemos hacer un **procesado estadístico** (más o menos complejo) **para interpretar los datos y obtener conclusiones**, por ejemplo obteniendo el valor medio de un parámetro, la cantidad total de paquetes transmitidos en 24 horas, etc.

Mecanismos de monitorización

Sondeo o “polling”

Se trata de hacer un acceso periódico a la información de gestión/monitorización. Si el acceso lo hacemos de forma remota a través de la propia red, estaremos introduciendo una cierta carga de tráfico de datos de gestión.

Notificaciones o “event reporting”

Los propios elementos de la red envían información al producirse ciertas circunstancias o eventos. Evidentemente esto supone una carga de trabajo para los procesadores encargados de realizar la monitorización y el reporte.

Sondas / proxys

Introducimos rutinas de exploración en ciertos elementos de la red, por los que pasa el tráfico o una parte de él, y que se encargan de recopilar la información además de realizar sus funciones dentro de la red.

Además, es importante tener en cuenta que:

- Si no disponemos de **datos “patrón”** sobre el funcionamiento de la red no podremos valorar la información obtenida; debemos conocer qué se considera “adecuado” o “bueno” en cuanto a un determinado valor, si existen acuerdos de nivel de servicio (SLA) que delimitan los parámetros a cumplir, etc.
- Si nunca hemos monitorizado la red anteriormente deberemos ir construyendo la base de **datos históricos**, como por ejemplo:
 - Carga típica en los enlaces.
 - Nivel de ocupación de los recursos.
 - Tiempo de indisponibilidad por período.
 - Etc.
- Es crucial **anotar todos los cambios** que se produzcan y mantener la información sobre la red actualizada.
- Establecer un **sistema de gestión de incidencias**.
- Mantener actualizada la información de consulta y soporte (documentación sobre los sistemas operativos, datos de los fabricantes de los equipos, etc.).

Protocolos y configuración de la red

Protocolos UDP y TCP

Ya conoces algunos conceptos sobre protocolos de la “familia TCP/IP”, como por ejemplo en qué consiste el encapsulamiento de los datos, lo que es un “puerto” y que son usados por las aplicaciones para intercambiar datos.

Para este intercambio de datos se usará el servicio a **nivel de aplicación**, que se encuentra soportado por todos los demás de la "pila de protocolos". En esa pila, el **protocolo IP** solamente se ocupa del enrutamiento y las direcciones IP identifican a los equipos (“hosts”) de origen y destino.

Ej. protocolos	MODELO TCP/IP
HTTP, HTTPS, FTPP, Telnet, SSH, POP3, IMAP, NTP, DHCP, PING, DNS, WINS	Nivel de Aplicación
TCP, UDP	Nivel de Transporte
IP, ARP, ICMP, IGMP	Nivel de Red
Ethernet, Token Ring, SDH, GSM, ATM	Nivel de acceso al Medio

A este nivel (capa 3) no existe distinción entre los usuarios, ni de los programas de aplicación que en un mismo ordenador pueden enviar/recibir los mensajes (paquete de datos).

Por encima del nivel de red (capa 3), los protocolos del nivel de transporte **TCP** (“*Transmission Control Protocol*”) y **UDP** (“*User Datagram Protocol*”) proporcionan un mecanismo que permite distinguir entre múltiples procesos en el mismo host.

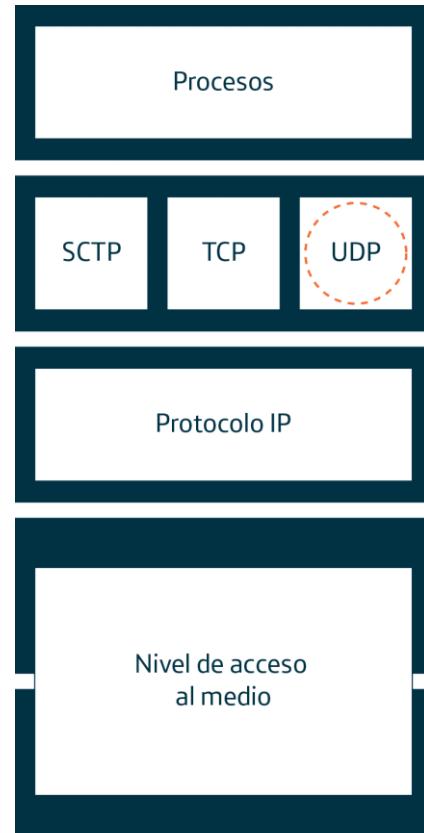
El protocolo UDP

Como hemos visto, el protocolo UDP proporciona un servicio “NO orientado a conexión” y sin garantías en la entrega.

En cada envío, además de los datos del proceso de usuario (aplicación), cada mensaje UDP contiene los números de **puerto origen y destino**, lo que permite que el software UDP de la capa de transporte reciba y entregue el mensaje al proceso de destino. Esta identificación de puerto se hace mediante un código de 16 bits llamado **número de puerto**, y ya hemos visto que los números de puerto se encuentran organizados en varios tipos/rangos.

UDP hace uso de la capa inferior (IP) para enviar el “**datagrama**” desde el un equipo (origen) a otro (destino), pero simplemente proporciona ese servicio de entrega, sin reconocimiento de los envíos y sin garantizar un orden en los mensajes entrantes, ni tampoco proporciona realimentación para controlar la velocidad del flujo de información entre las máquinas, por ejemplo.

Por tanto, los mensajes UDP pueden perderse, duplicarse o llegar desordenados, y no se establece ningún control sobre ellos, pudiendo darse el caso incluso de llegar a un ritmo más rápido de lo que el receptor pueda procesarlos.

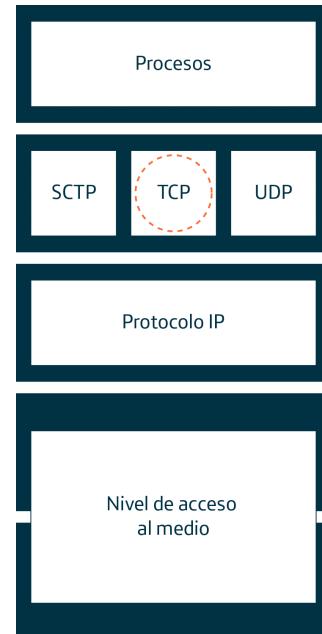


El protocolo TCP

A diferencia de UDP, el protocolo **TCP** (“Transmission Control Protocol”) proporciona un servicio de transporte **orientado a conexión y fiable extremo a extremo**, independiente de la red de transporte subyacente.

Al igual que en UDP, un mensaje completo TCP (incluyendo la cabecera y los datos) se encapsula en un datagrama IP y luego es enviado a través de la red. La unidad de transferencia entre entidades TCP (de los equipos de origen y destino) se denomina **segmento**.

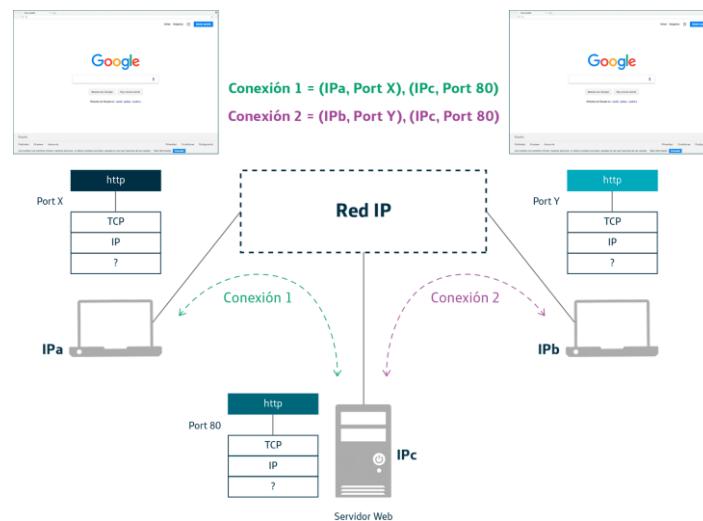
Los segmentos son intercambiados entre los hosts para establecer la conexión, transferir datos, enviar reconocimientos, notificar el tamaño de ventana y cerrar la conexión.



Ejemplo de conexiones TCP

En una comunicación el origen debe identificar el servicio que desea, especificando la dirección IP del host y su número de puerto TCP. De esta forma, cada "conexión" TCP estará identificada por dos parejas (dirección IP + puerto) para origen y destino.

Esta asociación es única para cada conexión, de forma que un determinado puerto TCP puede ser compartido por múltiples conexiones similares en la misma máquina, y así un servidor web, por ejemplo, podrá atender muchas peticiones simultáneas de orígenes diferentes.



¿Qué protocolo usa cada servicio?

Aunque para cada aplicación que queramos estudiar debemos asegurarnos de cuál es el protocolo que utiliza, e incluso puede haber aplicaciones y servicios que usen ambos protocolos en función de lo que vayan a hacer, podemos recordar que:

- TCP es usado por muchas de las aplicaciones más comunes de Internet, incluidas HTTP, SMTP, SSH y FTP.
- UDP se usa en el intercambio de información para protocolos como DHCP, BOOTP o DNS, en los que el establecimiento de conexión puede requerir el envío de muchos paquetes frente a la información útil, y también para transmitir audio y vídeo en tiempo real.

Servicio	TCP	UDP
Establecimiento y liberación de la conexión	✓	
Entrega en secuencia	✓	
Multiplexación de varias conexiones de transporte en un único servicio IP	✓	✓
Control de flujo	✓	
Reconocimientos extremo a extremo	✓	
Chequeo de errores	✓	✓
Retransmisión de segmentos		

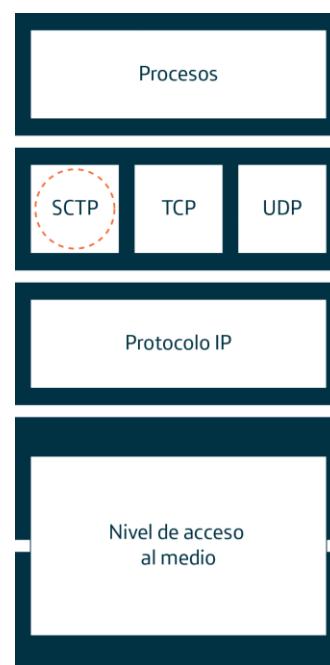
El protocolo SCTP

SCTP es un protocolo de capa de transporte que también está **orientado a conexión**; organiza los datos en bloques y a la conexión entre origen y destino se le llama **asociación**.

SCTP aporta fiabilidad en la entrega de los paquetes de datos, control de flujo y secuenciación.

Ofrece servicios similares a los de TCP, pero a diferencia de este, SCTP es un protocolo orientado a mensaje (como UDP), aunque opcionalmente permite en envío de mensajes fuera de orden, y además admite conexiones con sistemas de “host múltiple” o que tienen más de una dirección IP (lo que se conoce como “**multihoming**”).

SCTP puede seleccionar y monitorizar rutas de envío de los mensajes, seleccionar una ruta “primaria” y verificar continuamente el estado de las alternativas.



Otros protocolos

Además de IP a nivel de red, y TCP y UDP en la capa de transporte, existen otros protocolos que proporcionan la funcionalidad completa de Internet y las redes IP.

Estos son algunos ejemplos

- **DCCP.** "Control de congestión de datagramas", se encarga del control dinámico de congestión (distinto del de TCP) y la entrega en secuencia de un flujo múltiple de datagramas. Está pensado para aplicaciones que requieren la semántica basada en flujo de TCP, pero no necesitan la entrega en orden ni la confiabilidad que ofrece TCP.
- **ICMP.** "Protocolo de mensajes de control de Internet", no permite corregir los errores, pero sí los notifica a los protocolos de capas cercanas. Lo usan los *router* para indicar que ha habido un problema en alguna entrega, y permite a los administradores conocer el estado de los nodos. Es el utilizado cuando usamos el comando "**ping**".
- **ARP.** "Protocolo de Resolución de Dirección", permite averiguar la dirección física de una tarjeta de interfaz de red correspondiente a una dirección IP.

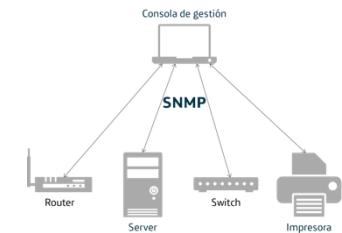


El protocolo SNMP

Es el protocolo utilizado para la **gestión de los equipos (hosts)** que forman las redes IP. Trabaja en el nivel/capa de aplicación y emplea para sus comunicaciones los mismos protocolos TCP/IP que las aplicaciones de usuario.

Sus funciones básicas son:

- **Gestión de fallos:** detección y corrección (si es posible) de problemas en la red, recepción de alarmas y realización de pruebas para diagnóstico.
- **Contabilidad:** análisis del uso de los recursos disponibles, evitando situaciones de bloqueo por monopolio de esos recursos.
- **Configuración:** gestión de los recursos y sus características de operación, administración de nuevos recursos, programación de alertas en tiempo real, generación de informes y estadísticas.
- **Rendimiento:** evaluación y seguimiento del comportamiento de los nodos y equipos que forman la red.
- **Seguridad:** soporte para la aplicación de políticas de seguridad.



Configurar los adaptadores de red

Como ya sabrás, para conectarnos a la red necesitamos disponer del hardware adecuado, bien sea una conexión por cable o inalámbrica.

Para ello necesitaremos tener la “**tarjeta de red**” o el “**adaptador de red**” adecuado. Puede incluso ser externo al equipo y conectarse por USB, por ejemplo, en aquellos equipos antiguos que no dispongan de adaptador incorporado.

Una vez hecha la instalación física (supongamos que ya la tenemos) es necesario disponer del **driver adecuado al hardware empleado** y para el sistema operativo que estemos utilizando. Es decir, necesitamos el **software controlador** que gestione ese hardware de conexión a la red.

Una vez detectado e instalado el *driver* adecuado al adaptador de red que vayamos a emplear, necesitaremos configurar los parámetros de conexión de los protocolos de red (principalmente IPv4 e IPv6). Esto podremos hacerlo a través del entorno gráfico del sistema operativo (GUI) o bien usando los comandos de consola/terminal que hemos visto anteriormente.

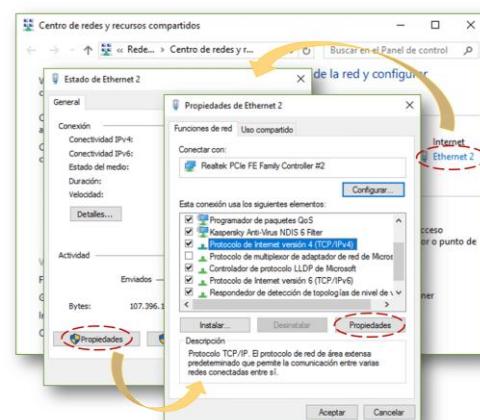
Evidentemente, el procedimiento concreto dependerá del S.O. en particular que usemos, por lo que veremos dos ejemplos, con un sistema propietario y uno libre.



Configuración del adaptador de red en Windows

En el caso de que lo hagamos a través del entorno gráfico deberemos acceder al "**Centro de redes y recursos compartidos**" y seleccionar la conexión de red a configurar, lo que implica elegir un adaptador para conectarnos a través de él (en la figura se elige la conexión “Ethernet 2”).

Para esa conexión elegiremos la opción de ver/modificar “Propiedades” y dentro de ella el/los protocolo/s que vayamos a usar, y para cada uno modificaremos a su vez los parámetros de conexión pulsando sobre el botón “propiedades” de la ventana.

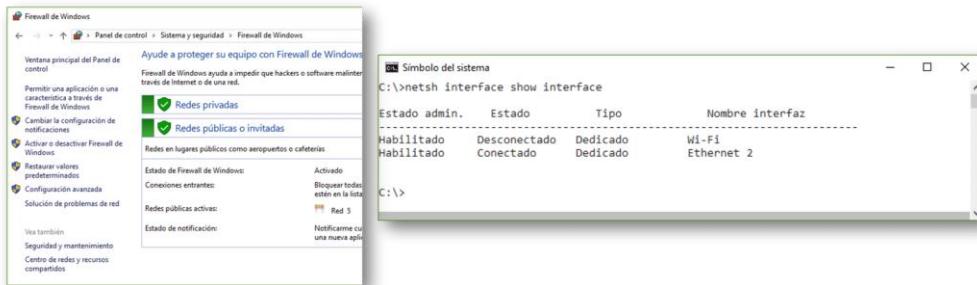


Los parámetros a configurar son los que hemos ido viendo a lo largo de la unidad, es decir:

- **Direcciones IPv4 e IPv6** del equipo (si lo usamos), y eligiendo si queremos que el equipo tenga una IP fija o se asigne dinámicamente por DHCP.
- **Máscara de subred**: correspondiente al tipo de red a la que nos conectamos.
- Dirección IP de la **puerta de enlace** predeterminada (**gateway**).
- **Direcciones de los servidores DNS** a utilizar (pueden ser DNS públicos o los que nos indique nuestro proveedor de acceso), o también la opción de configuración automática.
- En las **opciones avanzadas** elegir el orden para usar los DNS, las opciones de anexar sufijos, y las direcciones de servidores WINS si los usamos.
- La configuración deberemos hacerla para ambos protocolos IPv4 e IPv6 si alguna aplicación puede usarlo.

Una vez configuradas las conexiones a través de los adaptadores de red, debemos verificar el resto de la configuración de la red, por ejemplo **activar o desactivar el “firewall”**, o comprobar si necesitamos configurar los **“grupos de trabajo”** (podemos formar grupos de ordenadores que comparten recursos en red), aunque estos aspectos ya no tienen que ver con la configuración de los adaptadores, sino con el funcionamiento del sistema en red.

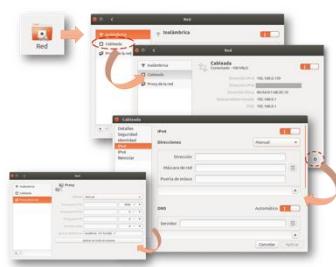
Para comprobar el estado de nuestras interfaces de red y conexiones podemos emplear el comando **“netsh”**.



Configuración del adaptador de red en Linux

Al configurar el adaptador para una conexión de red en Linux (Ubuntu en este caso) deberemos introducir básicamente los mismos datos que en cualquier otro sistema, ya que en definitiva se trata de parámetros de funcionamiento de la red. Ya sabemos que podemos hacerlo a través de la interfaz gráfica o mediante comandos de terminal.

Para proceder a la configuración abrimos la ventana de **"Configuración"** y dentro de ella la opción de **"Red"**. En los sucesivos menús elegiremos las conexiones que queramos configurar e iremos introduciendo los parámetros y direcciones IP, DNS, gateway, etc.



En el caso de que deseásemos realizar la configuración utilizando un **terminal de comandos**, podríamos editar los ficheros de configuración (los vimos anteriormente).

Por ejemplo, podríamos editar el archivo “**/etc/network/interfaces/**”(recuerda que es otro diferente en otras “distros” de Linux) y en él especificar si queremos una IP estática (mientras que en el ejemplo de la imagen está configurado para una conexión totalmente automática). Para hacerlo podemos teclear:

sudo nano /etc/network/interfaces

Y dentro del fichero poner la configuración deseada, por ejemplo:

```
auto eth0

iface eth0 inet static
    address 192.168.1.24
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
```

Además, deberemos configurar también los DNS en el fichero “**/etc/resolv.conf**” y para que la configuración se aplique reiniciar el adaptador de red con el comando:

/etc/init.d/networking restart

```
usuario@ubuntu:~$ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto eth0
iface eth0 inet static
    address 192.168.1.24
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1

usuario@ubuntu:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
usuario@ubuntu:~$ /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
usuario@ubuntu:~$
```

Adaptadores de red

Al hablar de las topologías y protocolos de red ya comentamos los distintos componentes de una red informática (servidores, switches, routers, etc.).

Vamos a verlos de nuevo ahora que conocemos su función y las necesidades de interconexión y protocolos que soportan las redes TCP/IP.

El primero de los elementos a tener en cuenta será el **adaptador de red**, también llamado **tarjeta de red o NIC (Network Interface Card)**, y es el elemento que nos permite conectar físicamente nuestro equipo a una red en particular.

El adaptador de red se conecta al medio físico de la red (cable, fibra, radio) y por lo tanto **puede haber en un mismo equipo varios adaptadores** diferentes si necesitamos conectarnos a varias redes.

Por ejemplo, en los equipos de usuario domésticos lo más normal es disponer de un **adaptador de red Ethernet y un adaptador de red inalámbrico** para la conexión a una red wifi.

Aún así, no siempre viene incorporado e integrado el adaptador de red. En ese caso hay que añadir una tarjeta independiente al equipo y, como hemos visto en el tema anterior, **configurarla adecuadamente** con los parámetros de conexión a la red.

Los adaptadores de red

El adaptador de red puede ser **interno o externo** y conectarse directamente al bus de comunicaciones de la CPU o bien a través de alguna interfaz de entrada / salida, como por ejemplo un puerto USB.

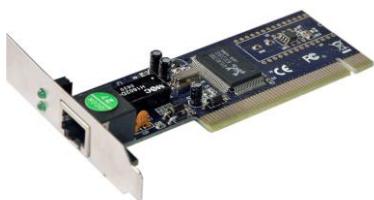
Como cualquier otro elemento de E/S, el adaptador de red **necesita del software controlador que lo gobierne y sirva de interfaz con el sistema operativo**, pues es al sistema operativo al que se dirigirán las aplicaciones para pedir el servicio de comunicación a través de la red a la que nos conectamos.



Adaptador de red wifi a través de USB.



Adaptador/tarjeta de red wifi interna.



Adaptador/tarjeta de red Ethernet con conexión vía RJ45.



Adaptador/tarjeta de red wifi interna con antena exterior.

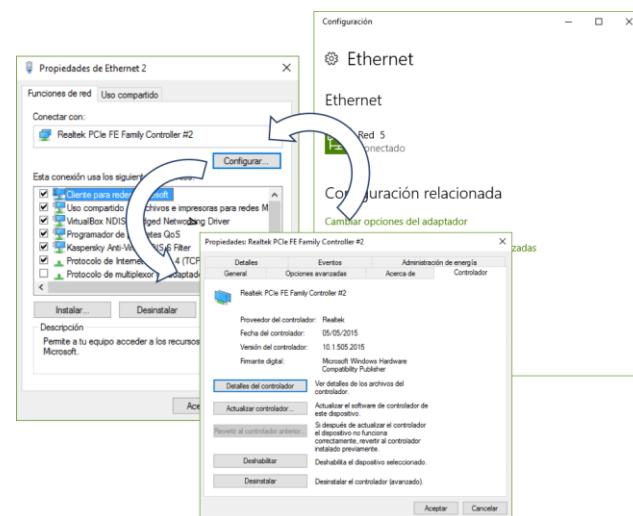
Es sobre el **software controlador** del adaptador sobre el que se montan otros programas (llamados **“packet drivers”**) para implementar todo el servicio de la pila de protocolos de comunicaciones. Si el adaptador que instalamos es del tipo **“plug & play”** (conectar y listo) el sistema operativo lo detectará automáticamente y normalmente nos pedirá autorización para instalar el controlador adecuado, que si no es encontrado deberemos suministrarlo de forma independiente.

Como podemos imaginar, no todos los adaptadores de red sirven para cualquier tipo de redes, e incluso en los que soportan varias clases del mismo tipo de red puede ser necesario configurarlos para un funcionamiento determinado.

Por ejemplo, una tarjeta Ethernet puede estar configurada para transmitir a 10 Mbps, a 100 Mbps, o bien detectar automáticamente la velocidad del medio.

Los **adaptadores de red** se pueden configurar a través de las opciones del entorno gráfico, bien en Windows a través del “Panel de Control” y las opciones de “Configuración de red”, o en el caso de Linux a través de las opciones de “Administración de red”.

Por supuesto, podemos utilizar los comandos de consola/terminal vistos en el tema anterior y la configuración a través de los ficheros de parametrización de las conexiones.



Dispositivos de interconexión

Para interconectar redes o segmentos de red entre sí se emplean una serie de dispositivos de interconexión que pueden trabajar a diferentes niveles en la escala de protocolos.

Según la funcionalidad de estos dispositivos reciben un nombre, como pueden ser: los repetidores (**repeaters**), puentes (**bridges**), commutadores (**switches**), enrutadores (**routers**), concentradores (**hubs**) o pasarelas (**gateways**).

Por supuesto, no debemos olvidarnos de los **medios físicos**, esto es, cables (del tipo que sean, conductores metálicos o de fibra) o enlaces vía radio. Si queremos conectar dos redes (o partes de una red) que emplean diferentes tipos de medios de transmisión, forzosamente deberemos tener algún dispositivo que sea capaz de pasar la información de un medio a otro.

A estos dispositivos que adaptan la señal y la convierten, por ejemplo de señal eléctrica a óptica, se le suele llamar "**transductores**", y solamente actúan a nivel físico, sin interpretar los protocolos de nivel superior.

Los dispositivos de interconexión se encargan de que la información que circula por una red pase a ser transmitida a través de otra red, que puede ser del mismo tipo que la otra o diferente en algún nivel.

Un dispositivo de interconexión tendrá entonces que **realizar diferentes funciones según el nivel al que trabaje**, es decir, a nivel físico (adaptando señales a medios de transmisión), a nivel de enlace o de acceso al medio (capa 2, convirtiendo tramas), a nivel de red (manejando los direccionamientos de ambas redes), etc.



MODELO TCP/IP	
Nivel de Aplicación	La interconexión puede ser más sencilla o complicada, según lo parecidas o diferentes que sean las redes a interconectar, llegando al punto que si ambas redes son totalmente diferentes hay que hacer una conversión completa de protocolos para pasar de una a otra. Estaríamos hablando de llegar hasta el nivel de aplicación, desempaquetando y volviendo a empaquetar la información del usuario.
Nivel de Transporte	
Nivel de Red	
Nivel de acceso al Medio	En el caso de tener que conectar dos redes de área local (LAN) a través de una red de área amplia (WAN), a menudo se suele hablar de la interconexión LAN-WAN como "acceso a la red" de área amplia (por ejemplo cuando conectamos nuestra red local a Internet), pero en el fondo lo que estamos haciendo es interconectando dos redes entre sí.

Tipos de dispositivos de interconexión

Repetidores (repeaters)

A menudo se incluyen al hablar de dispositivos de interconexión de redes, pero un repetidor realmente lo que hace es prolongar la extensión de una red o un segmento de una red.

Trabajan exclusivamente a nivel físico y se encargan de amplificar la señal cuando es débil, haciendo que pueda alcanzar una distancia mayor en la red.

Existen repetidores para todo tipo de señal y conexión a los diferentes medios de transmisión, ya sea para una señal de un cable de red local Ethernet, o un repetidor óptico para señales a través de fibra, etc. Los más conocidos actualmente quizás sean los repetidores de señal wifi, que se usan cuando la señal de radio de la red inalámbrica no alcanza toda la extensión en la que se encuentran nuestros equipos y necesitamos conectarlos vía wifi a la red.



Concentradores (hub)

Un concentrador (**hub**) se encarga de reproducir la señal que recibe por un puerto de entrada por todos sus otros puertos (hace “*broadcast*”). Actúa también a nivel físico, al igual que los repetidores.

El número de puertos que puede tener un *hub* puede ser pequeño (4, 8) o bastante elevado (p. ej. 24), y a veces también se les llama “repetidores multipuerto”. Es un **dispositivo “pasivo”** (no interpreta protocolos) que actúa como punto de conexión central entre las ramas de red conectadas a sus puertos, de forma que si un cable de una de las ramas tiene un problema, este no afecta a las demás.

También debemos recordar que un concentrador nos permite tener una red configurada físicamente “en estrella”, mientras que en realidad su funcionamiento es el de una red “en bus” (estructura lógica).

Al mismo tiempo que reproduce la señal sobre sus puertos, el concentrador puede servir de amplificador, al emitirla con un nivel adecuado en sus salidas, y llegado el caso podría evitar el uso de algún repetidor.



Comutadores (switch)

Puede parecer muy similar al concentrador, pero en realidad es bastante diferente. El **switch** también **reproduce la información que recibe por una de sus entradas**, pero no sobre todas las demás, sino solamente **por aquella en la cual se puede alcanzar el equipo de destino**.

Para hacer esto el comutador debe **leer la información de nivel 2** de las tramas de información, esto es, debe leer las **direcciones MAC** de origen y destino de la trama, y saber en cuál de sus ramas se encuentra ese destino, para lo cual tendrá que disponer de tablas almacenadas de las direcciones MAC de los equipos conectados a los segmentos de cada uno de sus puertos.

Es decir, un *switch* hace “**filtrado de tráfico**” (cosa que no hace un *hub*) y tiene capacidad de reconocer los equipos que se conectan al identificarse con sus MAC.

Al funcionar de esta manera se aprovecha mejor el ancho de banda disponible en la red, pues no se envía información innecesaria por cada segmento, sino solo al que debe llegar a los equipos que están en él.

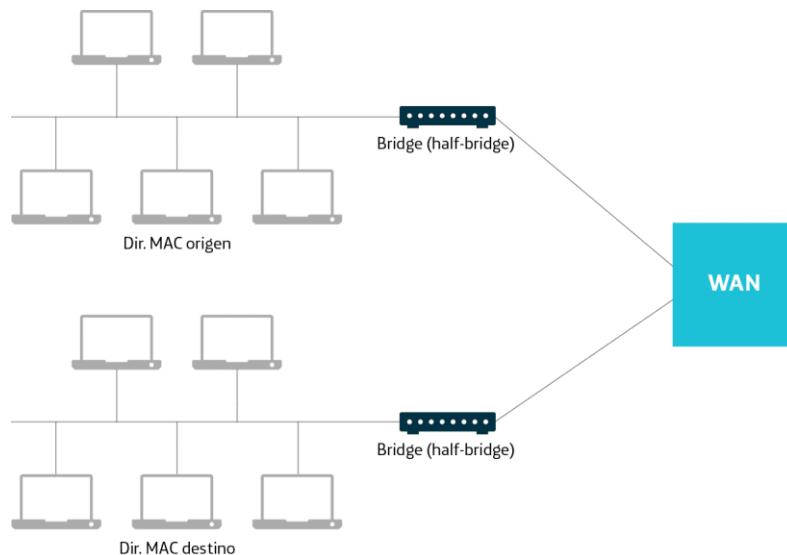


Puentes (bridge)

Los puentes, al igual que los *switches*, **actúan a nivel de capa 2 (nivel de enlace)**, y se utilizan para interconectar segmentos de una misma red o bien redes con protocolos de capa 2, iguales o diferentes (aunque ya no es lo más habitual pero podríamos conectar una red Ethernet a una red Token Ring a nivel 2) y con medios de transmisión similares o también diferentes (por ejemplo una red por cable y una inalámbrica).

A primera vista parece existir poca diferencia entre un *switch* y un *bridge*, pues ambos trabajan a nivel de capa 2, hacen filtrado de tráfico y en el mercado hay una gran variedad de productos con ambos nombres. Intentaremos señalar algunas diferencias:

- Los *bridge* suelen hacer la commutación por software, mientras que los *switch* la hacen por hardware (circuitos ASIC especializados en ello), ganando en velocidad.
- El número de puertos de un *bridge* suele ser más bajo (p. ej. 2 a 16) que el de un *switch* (decenas de puertos).
- Los *switches* de última generación aportan unas capacidades de commutación mucho más altas.
- Un *bridge* a menudo también sirve para interconectar dos redes a través de una red de área amplia, poniendo uno en cada lado de la conexión, y a los que se suele llamar “**half bridge**”.



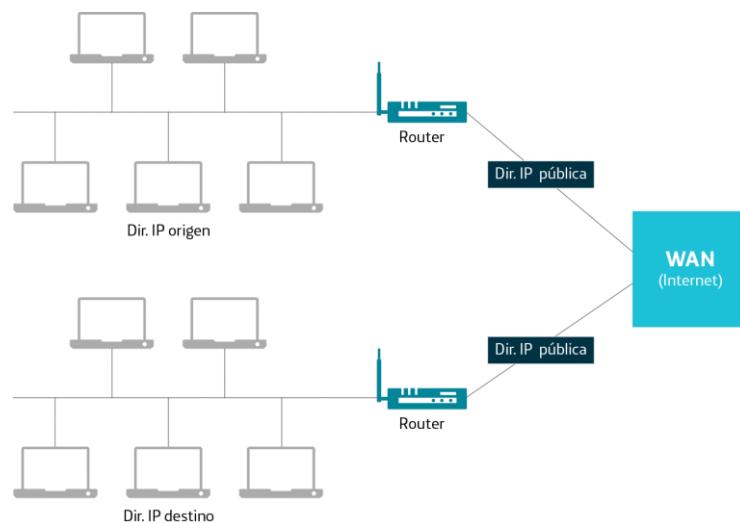
Enrutadores (router)

Los “**router**” trabajan a nivel de red, es decir, **manejan información de direccionamiento global a nivel de toda la red**. En el caso de las redes IP, los “**router IP**” se encargan de encaminar los paquetes de información en base a sus direcciones IP de origen y destino por alguna de sus “**rutas**” disponibles.

De lo anterior se deduce que un *router* es un equipo que debe tener almacenadas “**tablas de enrutamiento**” a nivel de red, con la posibilidad de tener diferentes caminos posibles (**rutas**) para alcanzar un determinado destino (es lo deseable, pues de esta forma si alguno de ellos falla, pueden seguir enviándose paquetes hacia el destino a través de otra ruta). Así funcionan los *router* del “interior” de Internet.

En las instalaciones domésticas a lo que estamos más acostumbrados es a disponer de un “**router de conexión a Internet**”, que parecería no tener “rutas” y solamente un camino de salida hacia el exterior (así suele ser), pero en realidad en la mayoría de los casos puede actuar también como *router* entre todas sus entradas, de forma que podemos formar una red interna doméstica y conectar diferentes equipos en nuestra “oficina”.

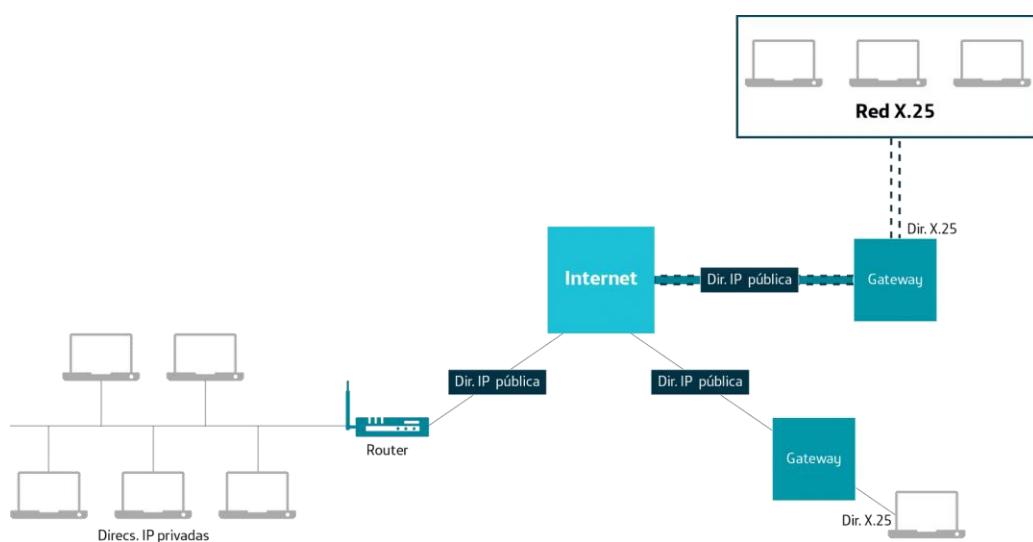
En este caso el *router* sabrá cuándo un paquete va dirigido hacia una IP “externa” y lo enviará a Internet y cuándo debe enviarlo por una “ruta interna” para alcanzar una IP a través de otro de sus puertos. ¡Ojo!, esta función siempre puede estar desactivada por configuración en el *router*.



Pasarelas (gateway)

Los “**gateway**” son equipos que se utilizan para **interconectar redes con protocolos totalmente distintos**. Es decir, trabajan al nivel más alto de la pila de protocolos; a **nivel de aplicación**. Para hacer la conversión de protocolos tienen que “desempaquetar” totalmente la información recibida hasta eliminar todas las cabeceras de los niveles de un protocolo, y volver a encapsular esa información de usuario con las cabeceras del protocolo de la otra red. Esto es, hacen una traducción completa de protocolos.

IMPORTANTE: el término “**gateway**” o “**puerta de enlace**” también se utiliza en las conexiones domésticas para identificar al equipo que nos da acceso a otra red, y no debemos confundirnos por el hecho de que un mismo término se use en entornos similares pero diferentes. En este caso se está refiriendo al *router* que nos da acceso a Internet, y que en realidad está trabajando a nivel de capa de red y es donde, por ejemplo, se realiza al traducción de direcciones IP privadas a dirección IP pública (servicio conocido como “**NAT – Network Address Translation**”).



Despedida

Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado:

- Recuerda que es tarea fundamental del administrador la comprobación de que las comunicaciones del sistema funcionan con normalidad y seguridad. Para ello dispone de comandos y herramientas que verifican las conexiones, el tráfico de la red, los enrutamientos, etc.
- Cada aplicación/servicio necesitará hacer uso de diversos protocolos de red. Conocer cuáles utiliza nos ayudará a detectar la causa de los posibles problemas de conectividad que puedan surgir.
- Monitorizar la red es realizar una verificación sistemática de su funcionamiento, analizando el rendimiento y la disponibilidad de los equipos y servicios que se prestan a través de ella.