

Introducción a Redes

Módulo 4

Enrutamiento

Enrutamiento

El **enrutamiento** es el proceso de **reenviar paquetes entre redes**, siempre **buscando la mejor ruta** (la más corta).

Para encontrar esa ruta más óptima, se debe tener en cuenta la tabla de enrutamiento y algunos otros parámetros como la métrica, la distancia administrativa, el ancho de banda, etc.


En una red los dispositivos que se encargan de llevar los paquetes entre redes son los '**router**'.
Un router es un dispositivo de capa 3 (Nivel de red) del modelo OSI.




Router

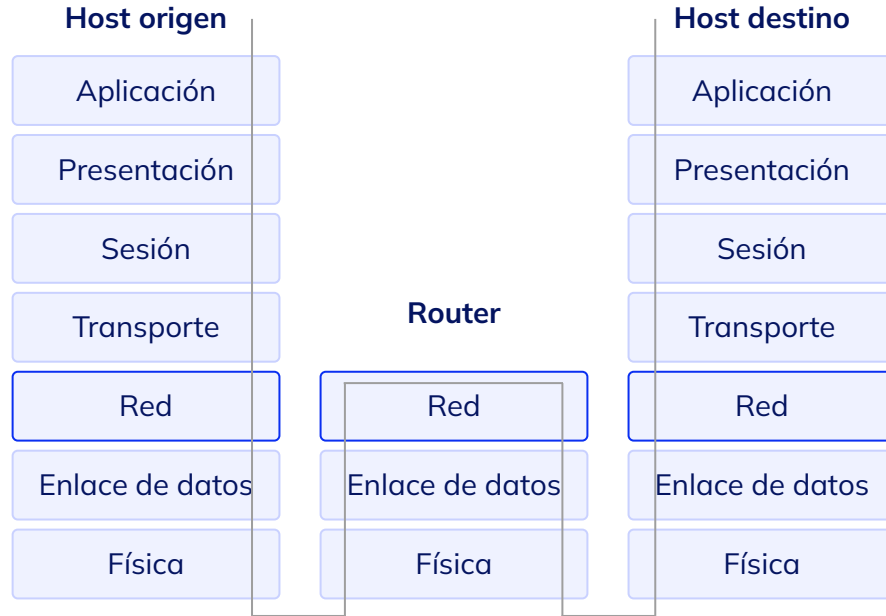
Su función es la de **establecer la mejor ruta** que destinará a cada paquete de datos para llegar a la red y al dispositivo de destino.

Es bastante utilizado para conectarse a Internet ya que conecta la red de nuestro hogar, oficina o cualquier red a la red de nuestro proveedor de este servicio.



La mayoría de los routers que se utilizan para el hogar y oficinas tienen incorporadas otras funciones adicionales al enrutador, como por ejemplo: punto de acceso inalámbrico, que permite crear y conectarse a una red Wifi; módem, que convierte las señales analógicas a digitales y viceversa; Conmutador, que conecta varios dispositivos a través de cable, creando una red local, a a este dispositivo se lo conoce como CPE.



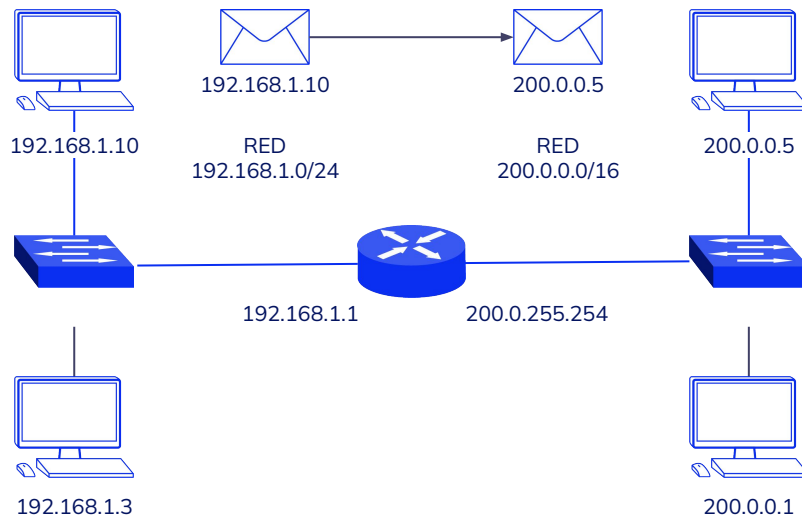


Funcionamiento

Consiste en enviar los paquetes de red por el camino o ruta más adecuada en cada momento. Para ello **almacena** los paquetes recibidos y **procesa la información de origen y destino** que poseen los datagramas (paquetes de datos con direcciones IP).

Un router recibe un paquete por una interfaz de red, y lo encamina hacia la red de destino por la interfaz vinculada a la red de destino. A diferencia de un *switch*, que tiene puertos de conexión, un router posee interfaces de red como cualquier host. Nótese que el router no está listado dentro de los dispositivos Ethernet, ya que este estándar abarca de capa 2 hacia abajo.

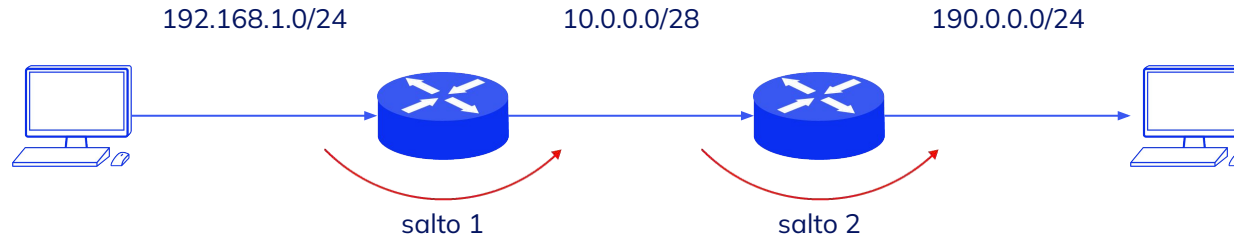
Ethernet determina las tecnologías físicas de transporte de tramas, no de paquetes.



Salto de red (*hop*)

En redes de computadoras, incluida la Internet, se produce un **salto** cuando se pasa un paquete de un segmento de red al siguiente. Los paquetes de datos pasan a través de los routers mientras viajan entre la fuente y el destino.

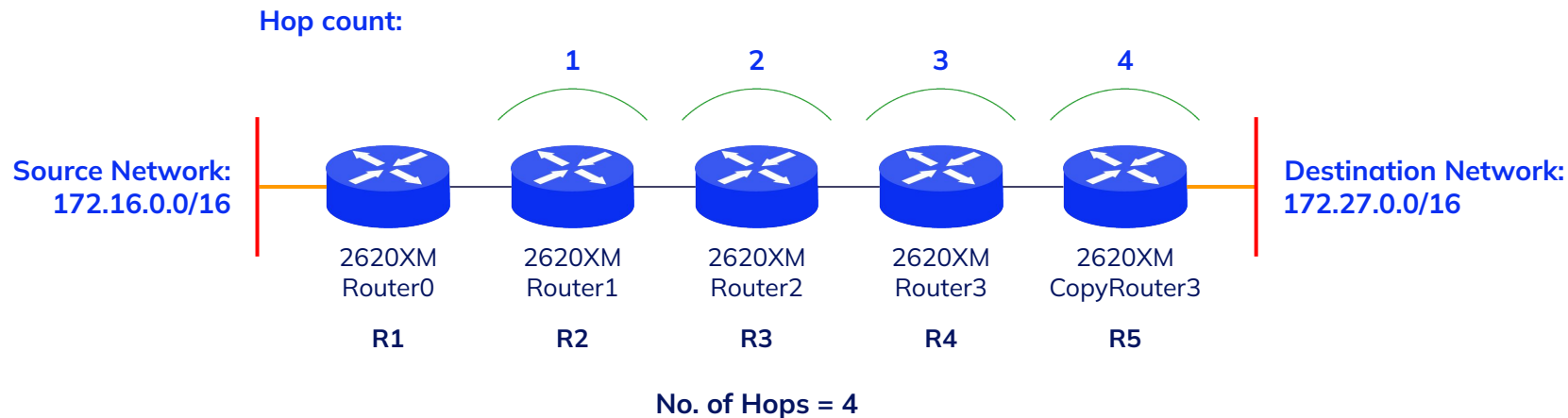
Dado que en cada salto se producen latencias de almacenamiento y reenvío y otras latencias, un gran número de saltos entre el origen y el destino implica un menor rendimiento en tiempo real.



Recuento de saltos (*hop count*)

El **recuento de saltos** se refiere al **número de dispositivos de red intermedios** por que pasa un paquete de datos.

Es una medida aproximada de la distancia entre dos hosts. Un recuento de 'n' saltos significa que 'n' dispositivos separan el host de origen del de destino.



Métrica

Los protocolos de enrutamiento se encargarán de buscar la mejor ruta hacia la red destino, pero tener en cuenta **el conteo de saltos no es del todo útil para determinar la ruta óptima de la red**, ya que no tiene en cuenta la velocidad, la carga, la fiabilidad o la latencia de ningún salto en particular, sino simplemente el recuento total.

No obstante, hay protocolos de enrutamiento, como el *Protocolo de Información de enrutamiento (RIP)*, que utilizan el recuento de saltos como única métrica.

Hop limit

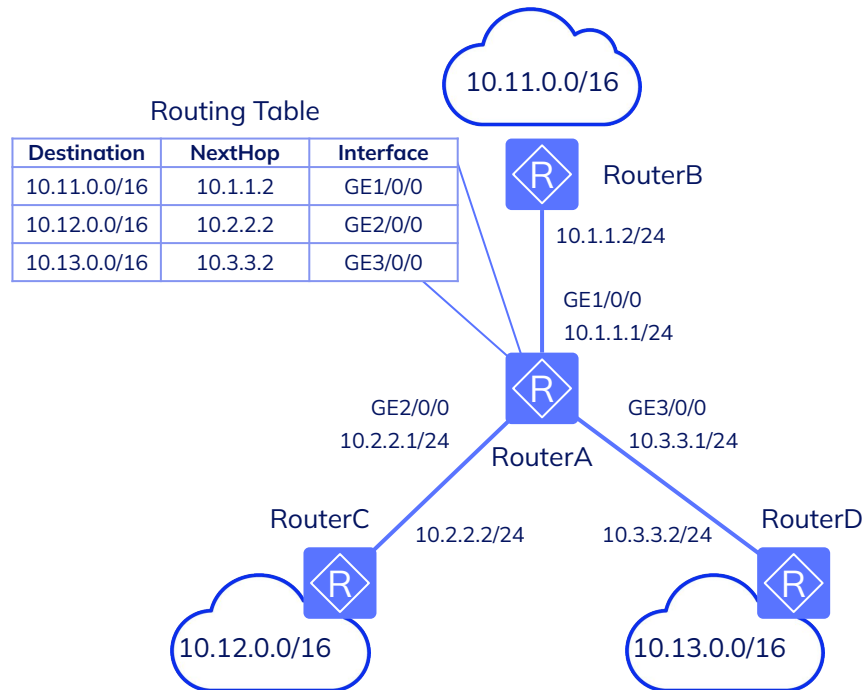
Conocido como el **tiempo de vida** (TTL) en IPv4, y **límite de saltos** en IPv6, este campo en el datagrama IP especifica un **límite** en el número de saltos que se permite a un paquete antes de ser desechado.

Los routers modifican paquetes IP a medida que se reenvían, disminuyendo el valor de los respectivos campos TTL o límite de saltos. Los routers no reenvían paquetes con un campo resultante de 0 o menos. Esto evita que los paquetes sigan un bucle infinito.

Next hop

Cuando un paquete de datos tiene una red de destino distinta a la de origen debe ser enviado a un dispositivo que se supone tiene acceso a otras redes, a este dispositivo se lo conoce como '**gateway**' y suele ser un router.

Cuando el paquete llega al *gateway* se determina si se conoce la red o si debe ser enviado a otro router que pueda conocer el camino de destino: el **siguiente salto** es la siguiente puerta a la que los paquetes deben ser reenviados a lo largo del camino a su destino final.



Trazado de ruta

El comando '**tracert**' puede ser utilizado para el número de hops del router de un host a otro. Los recuentos de hops son a menudo útiles para encontrar fallos en una red, o para descubrir si el enrutamiento es realmente correcto.

—



Tabla de enrutamiento

La **tabla de enrutamiento** es un sistema de registro del router que guarda la relación entre la red de destino y la interfaz de salida.

Es mantenida de forma **estática** (por el administrador de la red) o **dinámica** (con protocolos de enrutamiento). Se interpreta de la misma forma tanto si se trata de un host como de un router.



```
educacionit@athena:~$ route
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref      Uso Interfaz
default      gateway        0.0.0.0      UG     100    0        0 eno1
link-local   0.0.0.0        255.255.0.0  U      1000   0        0 eno1
172.17.0.0   0.0.0.0        255.255.0.0  U      0      0        0 docker0
192.168.1.0   0.0.0.0        255.255.255.0 U      100    0        0 eno1
```

Tabla de enrutamiento de un host con GNU/Linux, se indica cual es la interfaz por donde los paquetes pueden alcanzar el gateway.

Una tabla de enrutamiento suele contener la dirección de IP de una red de **destino** y la dirección de IP de la **siguiente puerta de enlace** en el camino hacia el destino final de la red. Al almacenar sólo la información del salto siguiente, el encaminamiento o el reenvío del salto siguiente reduce el tamaño de las tablas de enrutamiento.

Una determinada puerta de enlace sólo conoce un paso en el camino, no el camino completo a un destino.

También es clave saber que los siguientes saltos listados en una tabla de enrutamiento están en redes a las que la pasarela está directamente conectada, por lo tanto un paquete puede llegar a un destino siempre que exista una ruta, en una LAN un paquete jamás llegará a un destino de internet si no existe un router conectado directamente a la red pública.

Estructura

La tabla de enrutamiento presenta en su estructura los siguientes campos:

- **Destino:** direcciones IP de redes destinos donde sabe llegar el router.
- **Máscara:** máscara que determina el identificador de la red de la IP destino.
- **Interfaz:** identifica la interfaz de salida por donde debe de enviarse el datagrama.
- **Métrica:** parámetro en función del cual se escoge la mejor ruta (nº de saltos, ancho de banda, confiabilidad).
- **Gateway:** dirección del router utilizado para encaminar el datagrama hacia el destino. El encaminamiento puede ser de dos tipos, directo e indirecto, lo vamos a ver en las próximas slides.

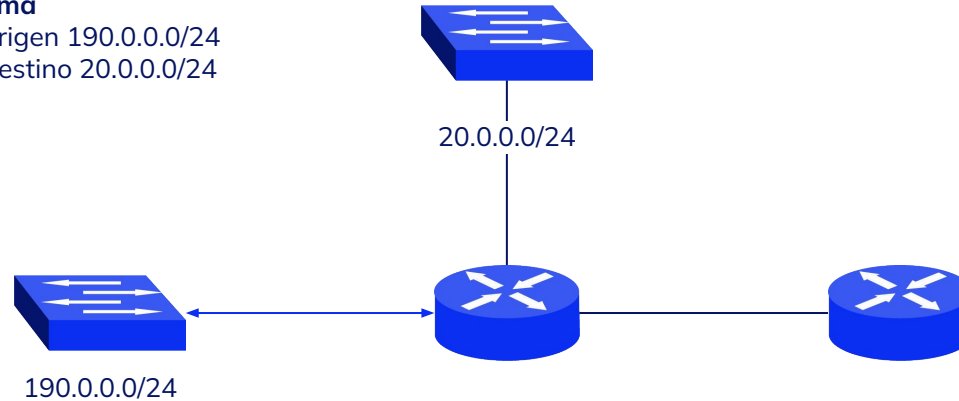


Encaminamiento directo: la interfaz de salida del router está en la misma red que la dirección destino del datagrama.

Datagrama

Red de origen 190.0.0.0/24

Red de destino 20.0.0.0/24



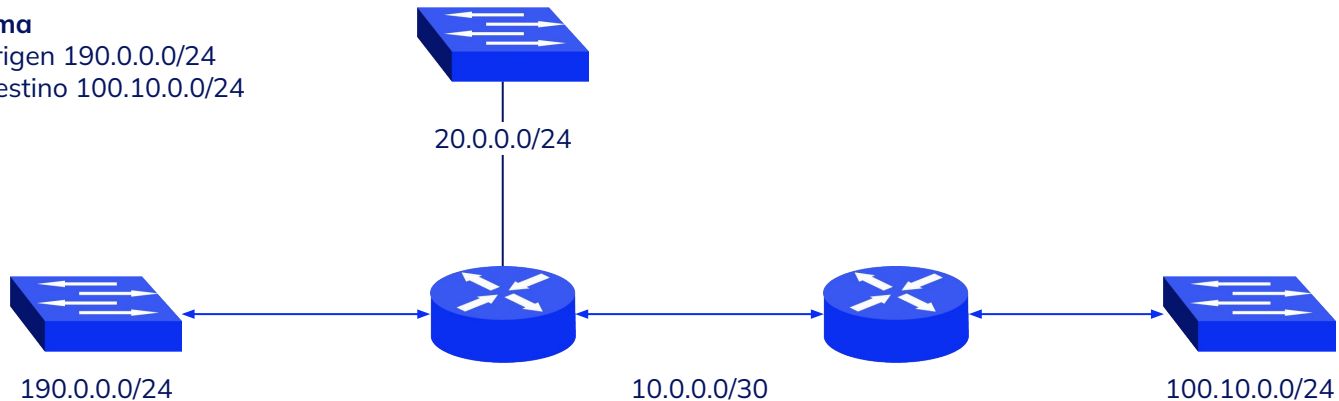
Los datagramas pasan de una red a otra, ambas redes están conectadas al mismo router.

Encaminamiento indirecto: la interfaz de salida no está en la misma red que la dirección destino del datagrama. El datagrama se envía a un router gateway que lo conducirá hacia su destino.

Datagrama

Red de origen 190.0.0.0/24

Red de destino 100.10.0.0/24

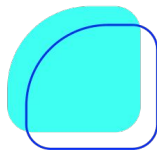


Los datagramas deben dar dos saltos para llegar a la red destino.

Funcionamiento

La tabla de enrutamiento o '*routing table*' contiene las **redes** que están **directamente conectadas** al router. Los paquetes que no tengan como destino de red una entrada en la tabla se envían a una ruta por defecto (*gateway*), esto evita que las tablas crezcan en tamaño ya que no podría albergar todas las rutas de todas las redes que conforman internet.

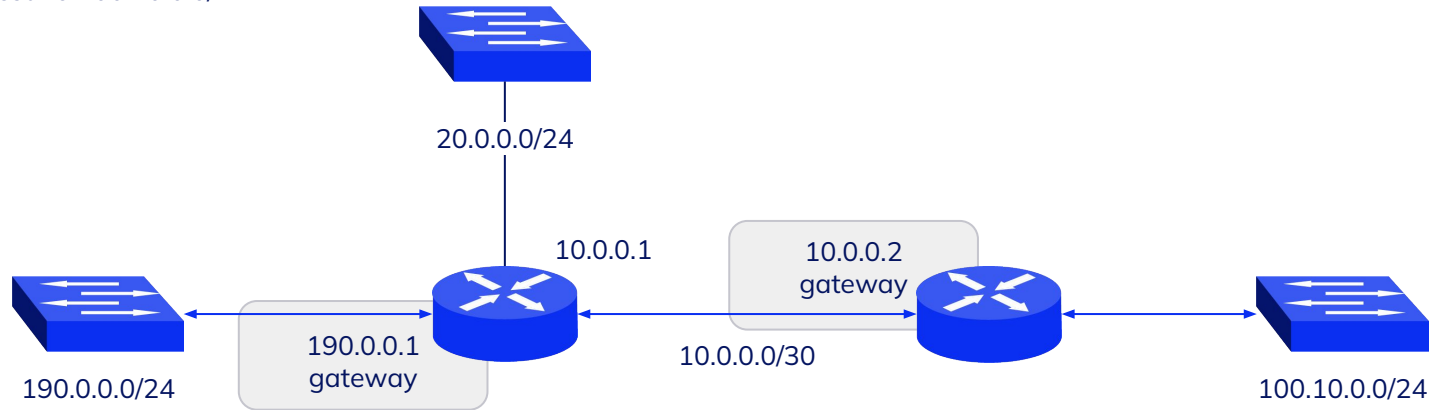
Un router normalmente especifica las rutas más cercanas, el resto de rutas se indican mediante una ruta o *gateway* por defecto. Al *gateway* por defecto se le envían aquellos datagramas que no se saben cómo encaminar.



Datagrama

Red de origen 190.0.0.0/24

Red de destino 100.10.0.0/24



Cada host, incluyendo los enrutadores, tiene definido un gateway por donde enviar paquetes cuya red destino no conozca.

Cuando el enrutador recibe un paquete se pasa a enrutar dicho paquete:

1. Extrae la IP destino.
2. Evalúa si la red de destino existe en la tabla.
3. Las entradas de la tabla se ordenan de mayor a menor bits en la máscara de red. Debido a este ordenamiento, la ruta por defecto será la última en mirarse si existe.
4. **Routing dinámico:** los enrutadores intercambian información con sus vecinos como la periodicidad con que se intercambian los paquetes de encaminamiento, el formato y contenido de estos paquetes, algoritmos asociados que permiten calcular el camino

óptimo para decidir la interfaz de salida (e.j., algoritmos de mínimo coste).

Routing estático: es el *routing* realizado por el administrador de la red, por lo tanto no es un sistema que responda automáticamente ante caídas de enlaces. Hay dos tipos de comandos que permiten introducir rutas en la tabla de *routing*: comandos que mapean IP sobre interfaces y comandos que añaden rutas hacia otras redes.

5. **La determinación de ruta:** en este punto se evalúa cuál es la mejor ruta a partir de las métricas recabadas por los protocolos y algoritmos de enrutamiento.

Topologías

Los enrutadores son la clave para que los paquetes IP lleguen de una red a otra, esto siempre que exista un camino, caso contrario se obtiene un mensaje de error “destino inalcanzable”. Esto es porque se ha superado la cantidad de saltos o dentro de la red ningún enrutador puede llegar a la red de destino.

Como los hosts y switches, los enrutadores se pueden organizar de manera que permitan redundancia, tolerancia a fallos o proveer caminos a segmentos de red según la cantidad de saltos o segmento de red con funciones específicas, por ejemplo la red de servidores.

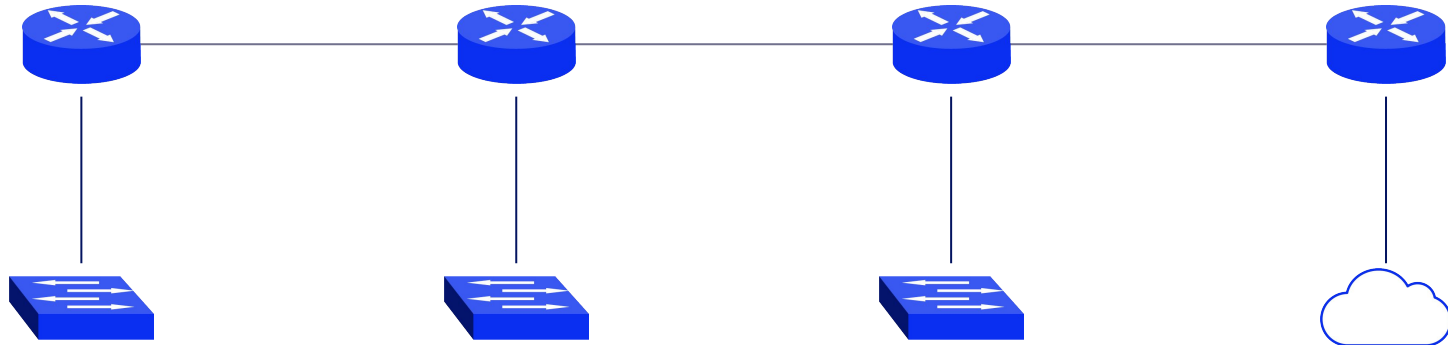
Por todo lo explicado en este capítulo podemos deducir lo siguiente:

- Se organizan bajo una **distribución física**, el interconectado, es decir la **topología física**.
- Se organizan bajo **redes lógicas** que comunican a los enrutadores entre sí, es decir la **topología lógica**.



Topología física

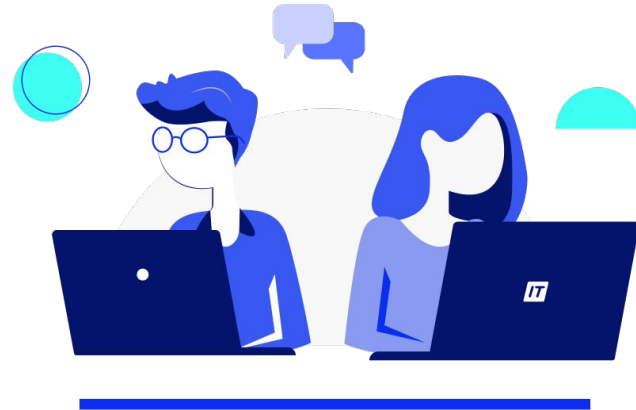
La siguiente red presenta una topología física con *backbone* en serie. La disposición física nos da la idea de cómo las tramas llegan de dispositivo a dispositivo.

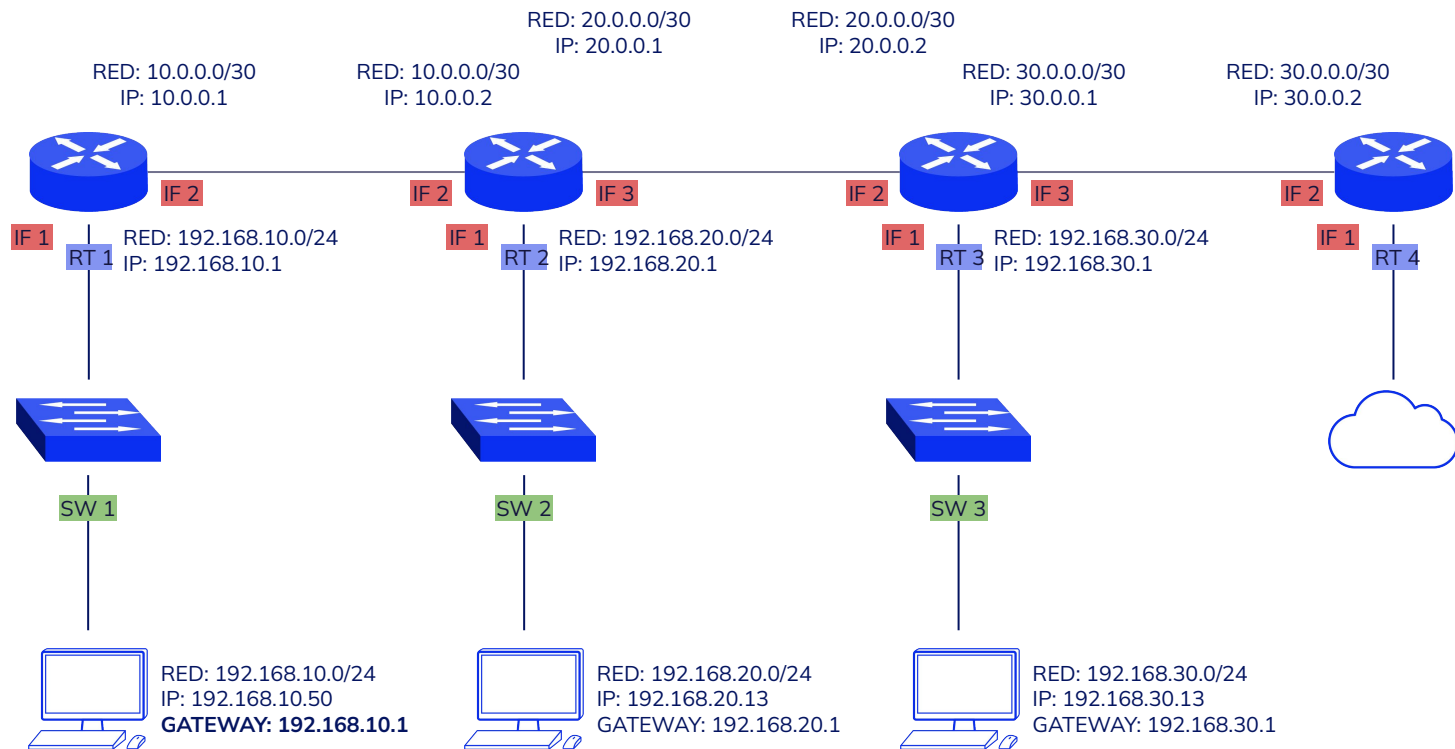


Topología lógica

Ya está definido cómo los dispositivos se interconectan entre sí a nivel físico, queda por determinar cómo los paquetes de datos llegan a nivel lógico.

Veamos la próxima slide.





Cada switch organiza un conjunto de hosts bajo una red propia y cada host puede enviar paquetes por el *gateway* por defecto de cada red.

DISPOSITIVO	RED	GATEWAY	ENRUTADOR	INTERFAZ DEL ROUTER
SW 1	192.168.10.0/24	192.168.10.1	RT 1	IF 1
SW 2	192.168.20.0/24	192.168.20.1	RT 2	IF 1
SW 3	192.168.30.0/24	192.168.30.1	RT 3	IF 1

Cada enrutador tiene una conexión física con el siguiente, al mismo tiempo cada enlace entre enrutadores forman una red lógica propia.

La siguiente tabla describe las interfaces de cada router junto con sus respectivas IP y la red de pertenencia:

INTERFAZ	RT 1	RT 2	RT 3	RT 4
IF 1	192.168.10.1/24	192.168.20.1/24	192.168.30.1/24	INTERNET
IF 2	10.0.0.1/30	10.0.0.2/30	20.0.0.2/30	30.0.0.2/30
IF 3	-	20.0.0.1/30	30.0.0.1/30	

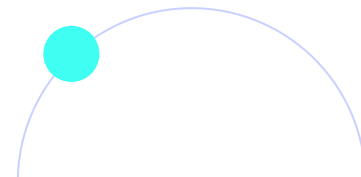
RED: 10.0.0.0/30**RED: 20.0.0.0/30****RED: 30.0.0.0/30**

Gateway

El *gateway*, también conocido como **puerta de enlace predeterminada**, es necesario para enviar un paquete fuera de la red local. Si la porción de red de la dirección de destino del paquete es diferente de la red del host de origen, el paquete tiene que hallar la salida fuera de la red original.

Para esto, el paquete es enviado al *gateway*. Este *gateway* es una interfaz del router conectada a la red local.

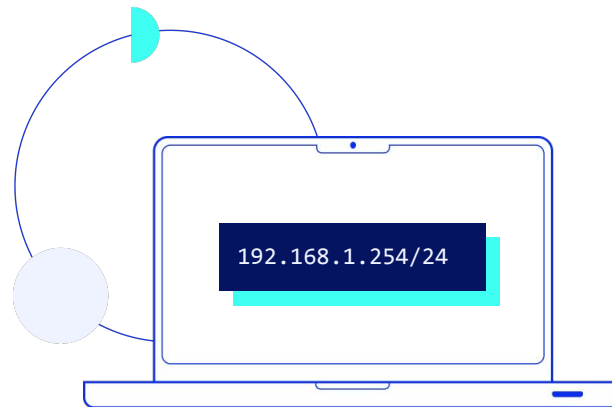
La interfaz del *gateway* tiene una dirección de capa de Red que concuerda con la dirección de red de los hosts. Los hosts están configurados para reconocer que la dirección es un *gateway*.

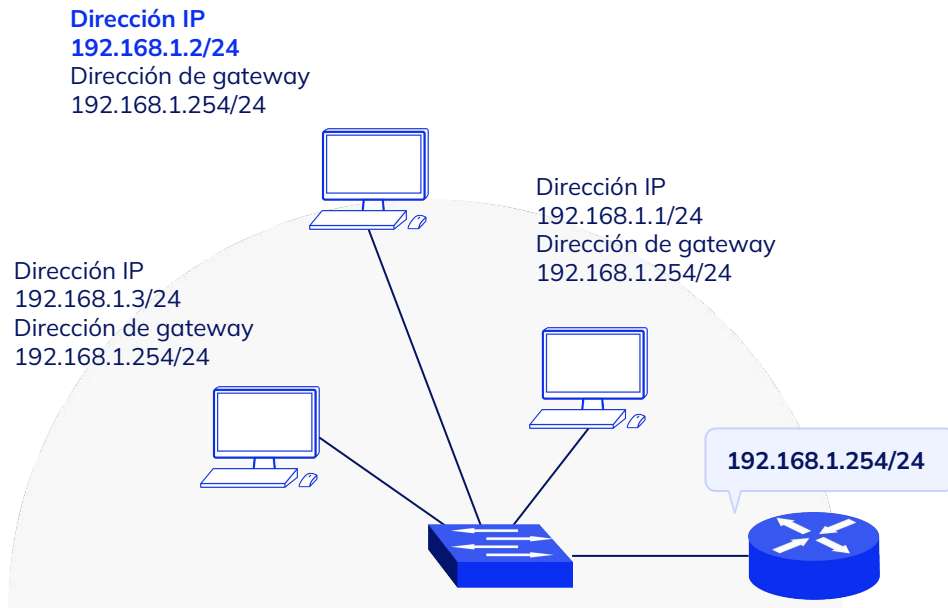


Gateway por defecto

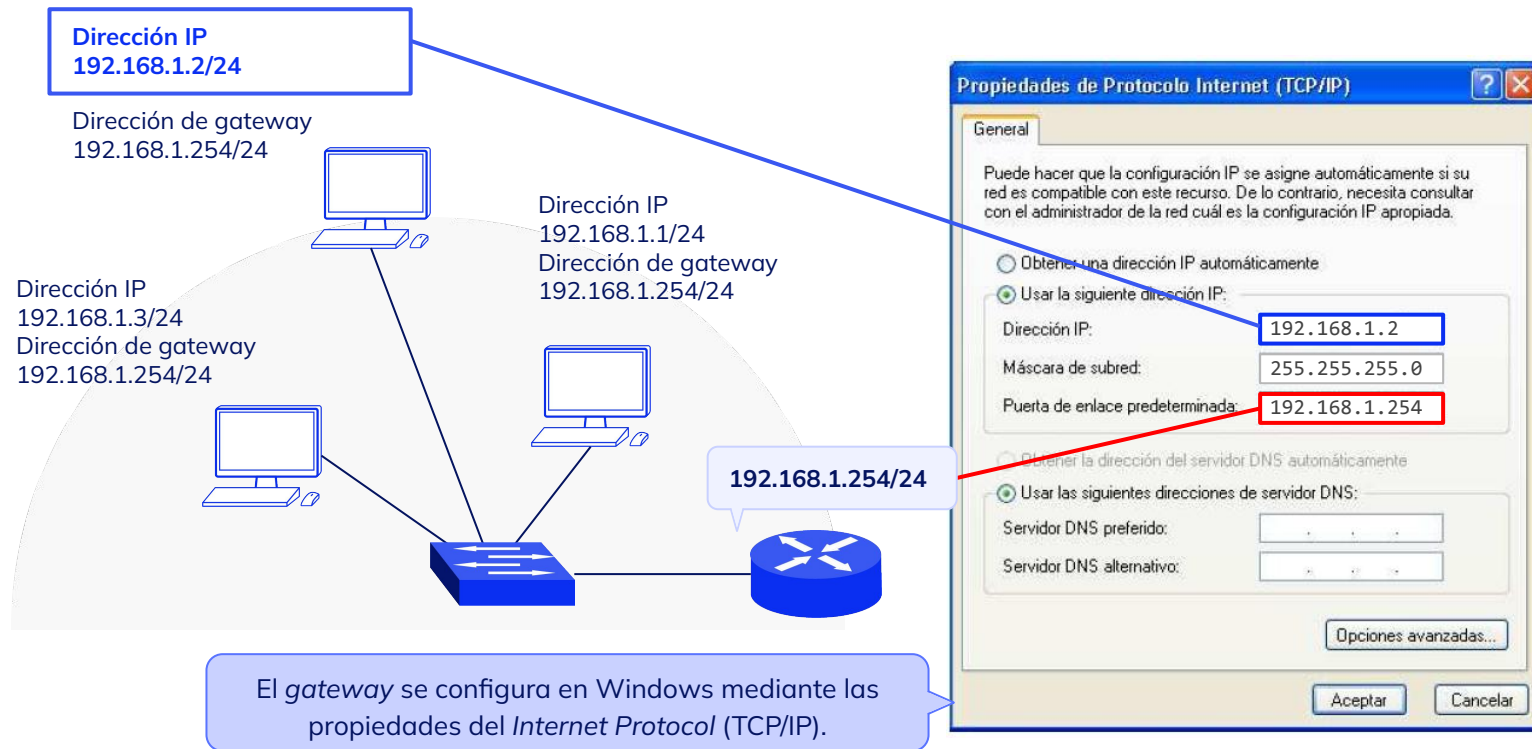
El *gateway* por defecto está configurado en el host. En una computadora con Windows, se usan las herramientas de las Propiedades del Protocolo de Internet (TCP/IP) para ingresar la dirección IPv4 de *gateway* por defecto.

Tanto la dirección IPv4 de host como la dirección de *gateway* deben tener la misma porción de red (y subred si se utiliza) de sus respectivas direcciones.





Todos los hosts de esta red poseen la misma dirección de gateway por defecto la dirección de la interfaz de gateway conectada a la red.



Ningún paquete puede ser **enviado sin una ruta**. Si el paquete se origina en un host o se reenvía por un dispositivo intermediario, el dispositivo debe tener una ruta para identificar dónde enviar el paquete.

Un host debe **reenviar el paquete** ya sea al host en la red local o al *gateway*, según sea lo adecuado. Para reenviar los paquetes, el host debe tener rutas que representan estos destinos. Un router toma una decisión de reenvío para cada paquete que llega a la interfaz del *gateway*. Este proceso es denominado **enrutamiento**. Para reenviar un paquete a una red de destino, el router requiere una ruta hacia esa red.

Si una ruta a una red de destino no existe, el paquete no puede reenviarse. La red de destino puede ser un número de routers o saltos fuera del *gateway*. La ruta hacia esa red sólo indicaría el router del siguiente salto al cual el paquete debe reenviarse, no el router final.

El proceso de enrutamiento usa una ruta para asignar una dirección de red de destino hacia el próximo salto y luego envía el paquete hacia esta dirección del próximo salto.



Network Address Translation (NAT)

La **traducción de direcciones de red**, también llamado **enmascaramiento de IP** o **NAT** (del inglés *Network Address Translation*), es un mecanismo utilizado por routers IP para cambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles. Consiste en **convertir**, en tiempo real, las direcciones utilizadas en los paquetes transportados.

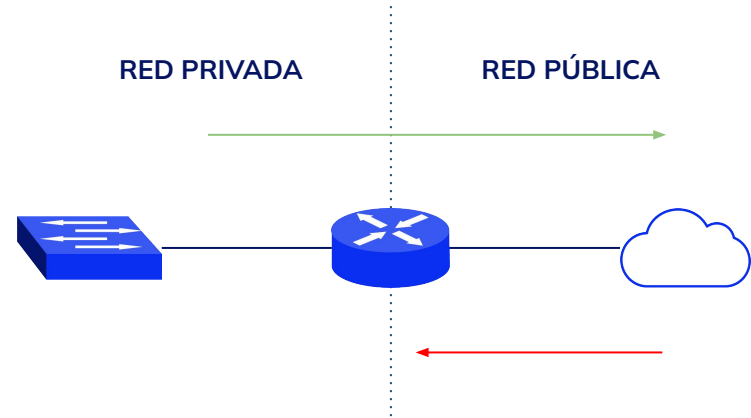
En un escenario normal al interconectar dos redes mediante un router estas se hacen visibles mutuamente, esto quiere decir que los paquetes son enrutados en todas direcciones gracias a la

tabla de enrutamiento. Pero por el contrario cuando un segmento de red se conecta a otra red mediante NAT, se denomina “red privada”, en principio, los hosts de la red privada pueden llegar a cualquier destino, pero los hosts de las redes del otro lado no pueden llegar a los hosts de la red privada de forma directa.



Este tipo de configuración de red es el que se usa en las configuraciones de acceso a internet:

- Desde la LAN se puede acceder a cualquier host (servidores y otros routers) dentro de la red pública.
- Desde la red pública no se puede acceder a los hosts de una red privada.



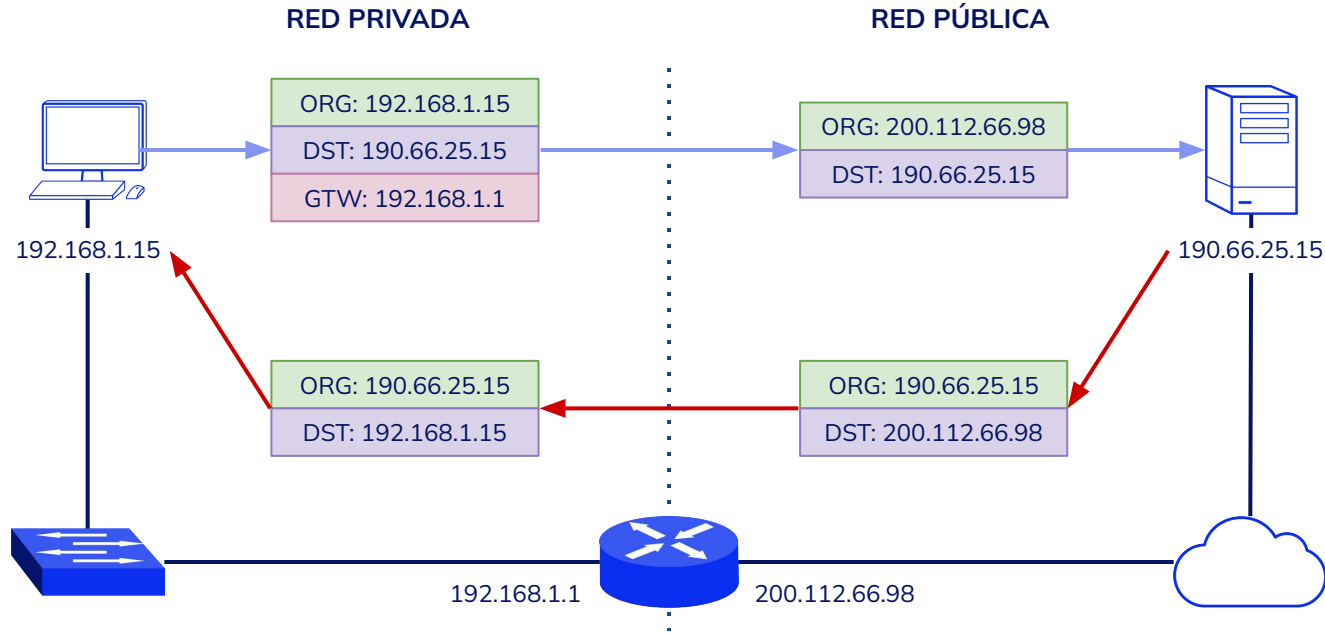
Funcionamiento

Un paquete cuyo destino sea una red distinta a la propia será **reenviado por el gateway** que es un **router NAT**. El router “enmascara” la dirección de origen por la dirección de la interfaz que está conectada a la siguiente red.

Bajo el modelo NAT los paquetes llegan al servidor como si hubiesen sido originados en el router. El servidor envía las respuestas a la IP pública del router, luego este traduce las direcciones para entregar el paquete al host correspondiente dentro de la red privada.

Podremos ver un ejemplo en la siguiente slide.





El paquete de datos sale del host de la red privada, para el host receptor quien envió el paquete fue un host con dirección 200.112.66.98 y es a quien dirigirá las respuestas.

Tipos de NAT

- **NAT de cono completo (*Full-Cone NAT*)**

En este caso de comunicación completa, NAT mapeará la dirección IP y puerto interno a una dirección y puerto público diferentes. Una vez establecido, cualquier host externo puede comunicarse con el host de la red privada enviando los paquetes a una dirección IP y puerto externo que haya sido mapeado. Esta implementación NAT es la menos segura, puesto que un atacante puede adivinar qué puerto está abierto.

- **NAT de cono restringido (*Restricted Cone NAT*)**

En este caso de la conexión restringida, la IP y puerto externos de NAT son abiertos cuando el host de la red privada quiere comunicarse con una dirección IP específica fuera de su red. La NAT bloqueará todo tráfico que no venga de esa dirección IP específica.

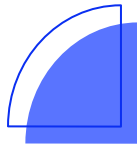
- **NAT de cono restringido de puertos**
(*Port-Restricted Cone NAT*)

En una conexión restringida por puerto NAT bloqueará todo el tráfico a menos que el host de la red privada haya enviado previamente tráfico a una IP y puerto específico, entonces solo en ese caso esa IP/puerto tendrán acceso a la red privada.



- **NAT Simétrica (*Symmetric NAT*)**

En este caso la traducción de dirección IP privada a dirección IP pública depende de la dirección IP de destino donde se quiere enviar el tráfico.



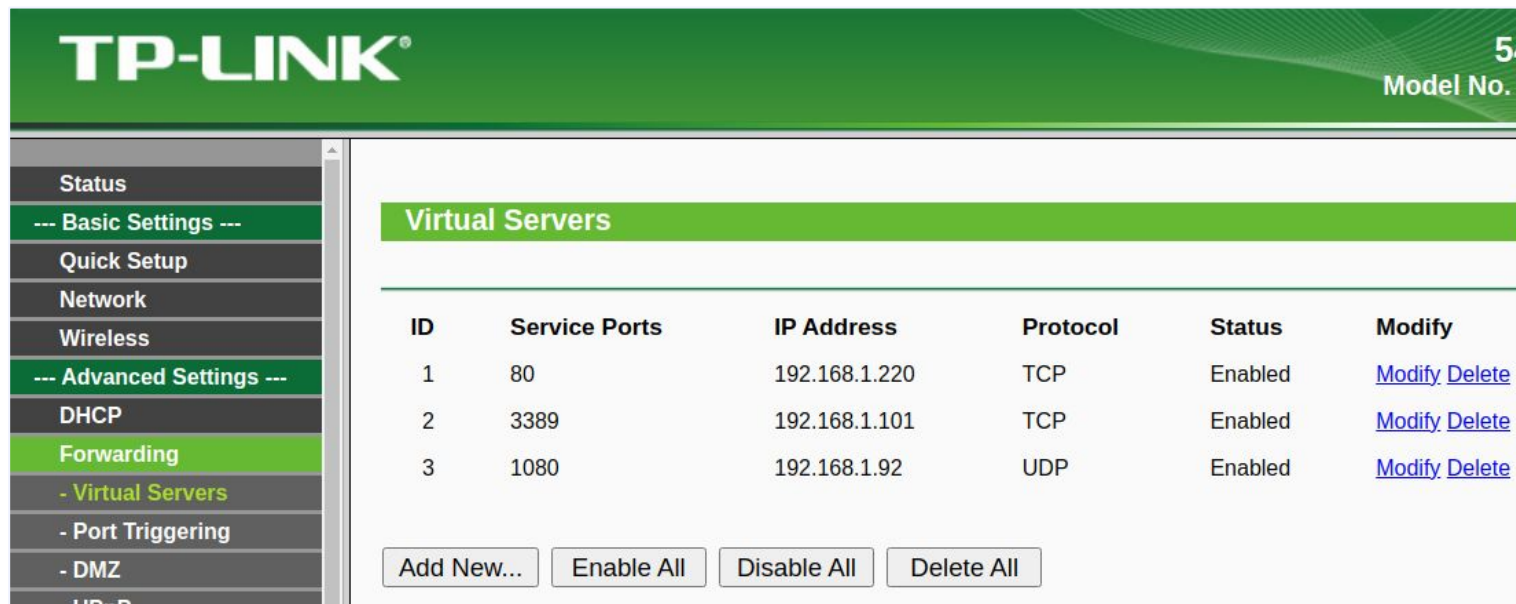
Reenvío de puertos (*Port forwarding*)

El **reenvío de puertos**, también conocido como “apertura de puertos” y “mapeo de puertos”, consiste en permitir que determinados servicios de una red privada sean **accesibles** desde redes externas.

La apertura de puertos se utiliza cuando el segmento de red detrás de un enrutador NAT debe recibir paquetes, por lo tanto peticiones de conexión desde hosts que están fuera de la red.

La mayoría de los routers hogareños (aquellos que integran wifi, switch, etc) trabajan en modo NAT siendo el puerto “WAN” el que separa la red privada de la red pública. En estos dispositivos podemos hacer la apertura desde el panel de configuración web.





The screenshot shows the TP-Link router's web management interface. The top header is green with the TP-LINK logo and the model number 54. The left sidebar contains a menu with options: Status, --- Basic Settings ---, Quick Setup, Network, Wireless, --- Advanced Settings ---, DHCP, Forwarding (highlighted), - Virtual Servers, - Port Triggering, - DMZ, and UPnP. The main content area is titled 'Virtual Servers' and displays a table with three entries. Each entry has columns for ID, Service Ports, IP Address, Protocol, Status, and Modify. Below the table are four buttons: Add New..., Enable All, Disable All, and Delete All.

ID	Service Ports	IP Address	Protocol	Status	Modify
1	80	192.168.1.220	TCP	Enabled	Modify Delete
2	3389	192.168.1.101	TCP	Enabled	Modify Delete
3	1080	192.168.1.92	UDP	Enabled	Modify Delete

[Add New...](#) [Enable All](#) [Disable All](#) [Delete All](#)

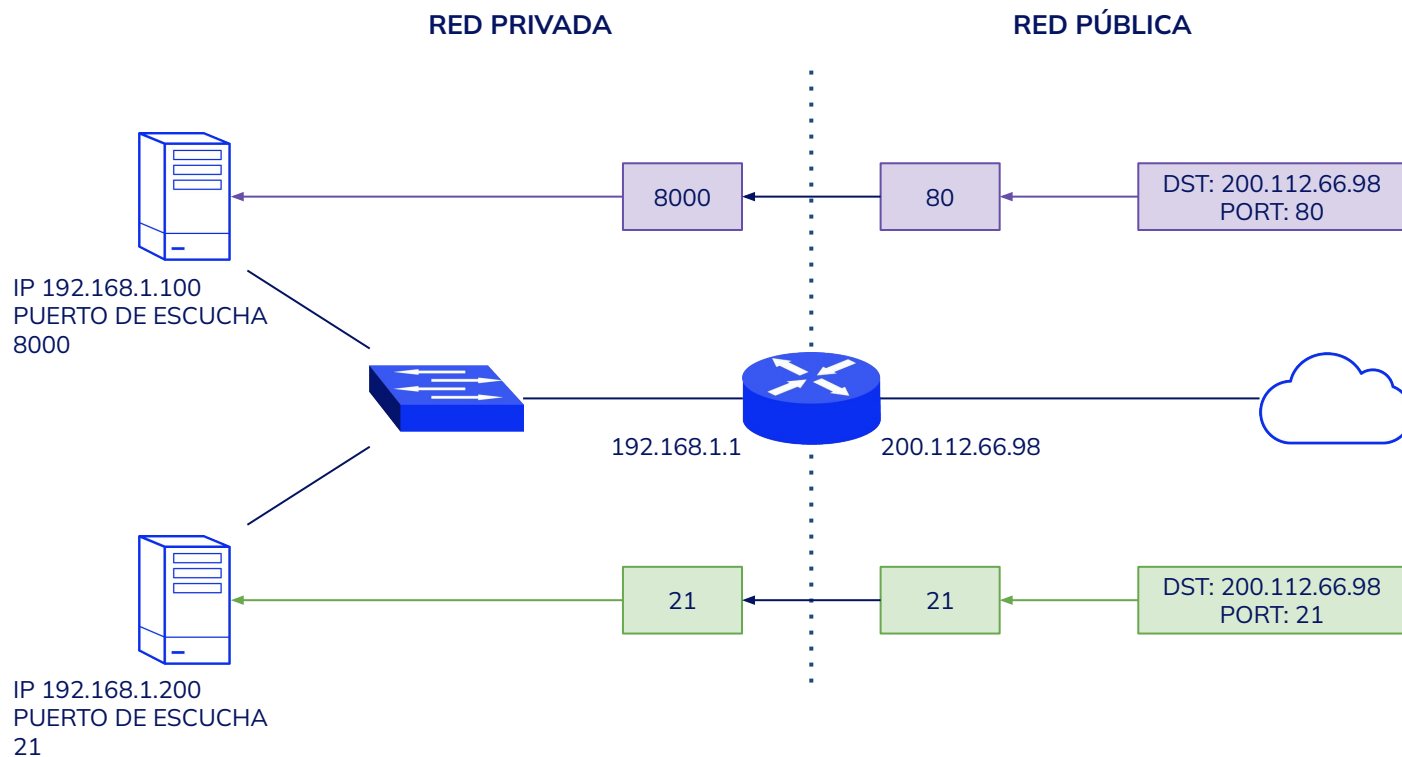
Mapeo de puertos en un router TP-Link.


Funcionamiento

Los mensajes además de tener una IP de destino tienen un **puerto de destino**, lo que permite establecer las conexiones en las que las aplicaciones intercambiarán información.


En una red NAT los hosts dentro de la red privada son inaccesibles, lo que es accesible es la dirección IP pública del router. La apertura consiste en indicar que los paquetes que tengan como destino un puerto y un protocolo determinado se reenvían a un host dentro de la red privada.







Puerto del router	Servicio	Host destino	Puerto de escucha del host	Protocolo
80	Web App	192.168.1.100	8000	TCP
21	FTP	192.168.1.200	21	TCP

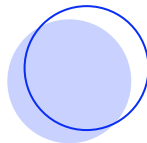


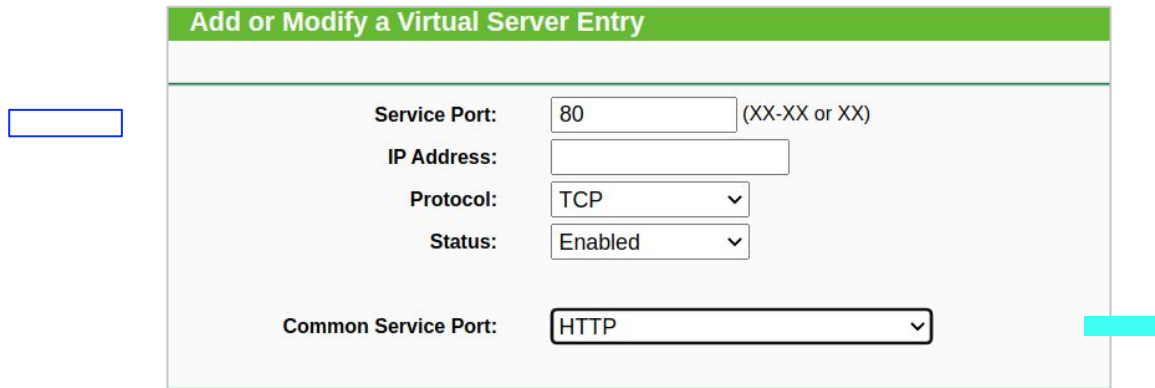
Puertos y protocolos

La **apertura de puertos** se puede realizar según los siguientes criterios:

- Puertos de escucha en la red pública.
- Puertos de escucha de los hosts dentro de la red.
- Protocolos de transporte TCP/UDP.
- Rango de puertos.

Normalmente los enrutadores tienen perfiles predefinidos según el tipo de servicio.





Add or Modify a Virtual Server Entry

Service Port: (XX-XX or XX)

IP Address:

Protocol: ▼

Status: ▼

Common Service Port: ▼

Cabe destacar que la **apertura de puertos** apunta a **redireccionar un puerto externo** al puerto de un host dentro de la red.

Van de uno a uno, no sería posible abrir un puerto y que apunte a dos direcciones IP diferentes, a nivel lógico es imposible.

**¡Sigamos
trabajando!**