

Protocolo IP

Fundamentos de Redes de Ordenadores

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

Universidad Rey Juan Carlos

Octubre 2022



©2022 GSyC

Algunos derechos reservados.

Este trabajo se distribuye bajo la licencia

Creative Commons Attribution Share-Alike

disponible en <http://creativecommons.org/licenses/by-sa/4.0/deed.es>

Contenidos

- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

Contenidos

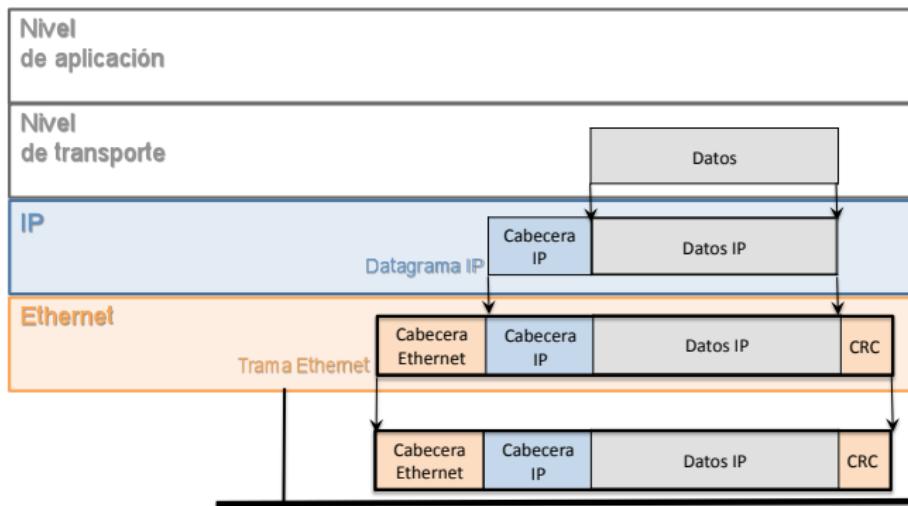
- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

Protocolo IP (Internet Protocol)

- IP es un protocolo de Nivel de Red que sigue un modelo:
 - **no orientado a conexión**: cada vez que hay que enviar datos se envían en un mensaje dirigido al destino, sin necesitar fases de establecimiento y liberación de conexión
 - **basado en datagramas**: cada mensaje dirigido a un destino se encamina por separado
 - **no fiable**: los mensajes pueden perderse, llegar duplicados, llegar desordenados.
- A la unidad de datos que envía IP se le llama **datagrama IP**.

Encapsulación

- Un datagrama IP se encapsula dentro de la parte de datos del paquete del nivel de enlace.
- Si el nivel de enlace es Ethernet, el datagrama IP viaja en la parte de datos de la trama Ethernet.



Contenidos

1 Protocolo IP

- Formato del datagrama IP
- Direcciones IP
- Tablas de encaminamiento IP

2 Protocolo ARP

3 Ejemplo de encaminamiento

- Comunicación entre máquinas vecinas
- Comunicación entre máquinas NO vecinas

4 El problema inverso al ARP

5 Protocolo ICMP

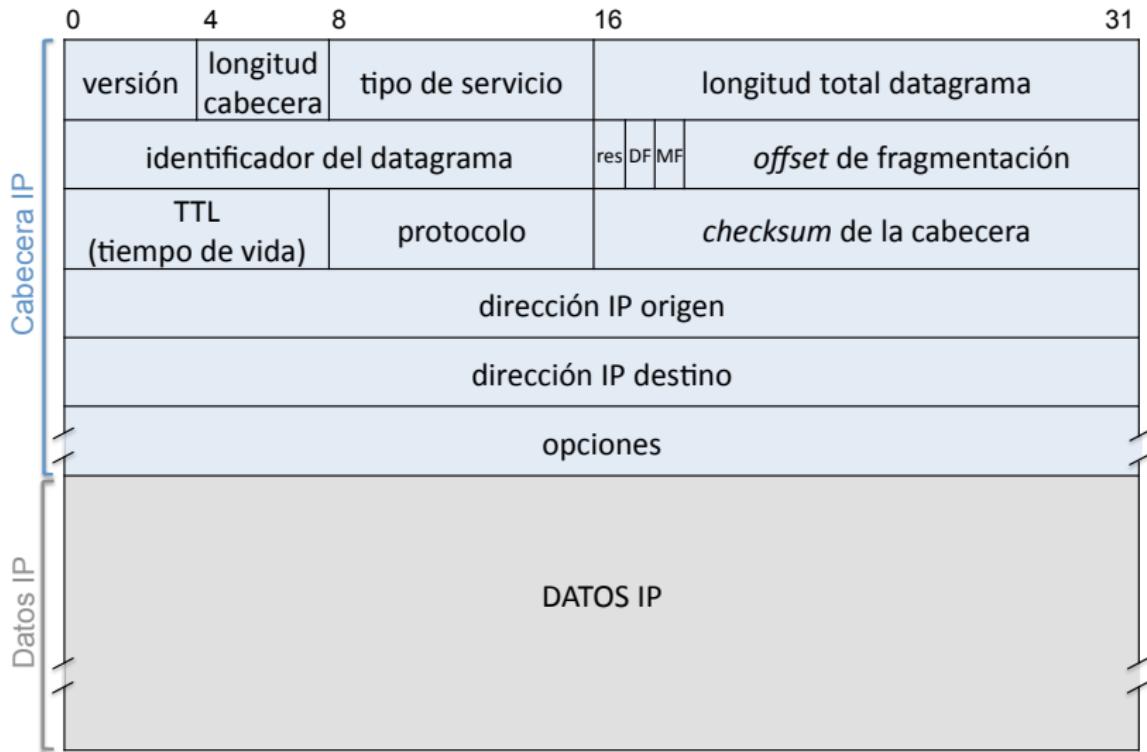
6 Asignación y planificación de subredes IP

7 Ejemplo: traceroute

8 Congestión

9 Referencias

Datagrama IP



Campos de la cabecera del datagrama IP (I)

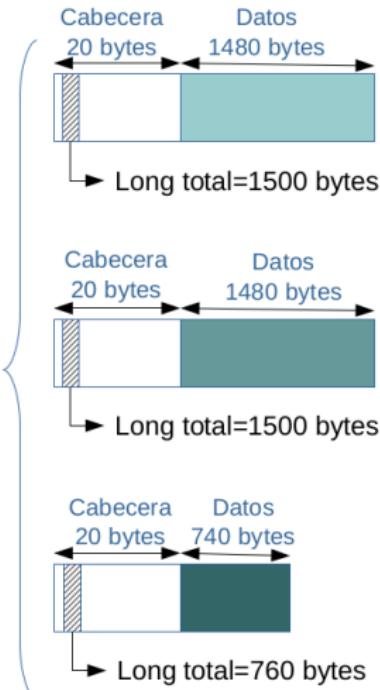
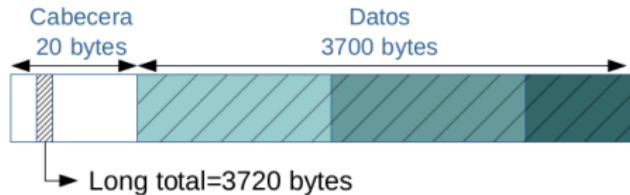
- **versión (4bits)**: Versión actual: 4. Próxima versión: 6 (al protocolo IP versión 6 se le conoce como “IPv6”).
- **longitud cabecera (4bits)**: Número de palabras de 32 bits que ocupa la cabecera. La cabecera puede tener una longitud variable debido al campo opciones. Normalmente la longitud de cabecera vale 5 (cabecera de 20 bytes, no hay opciones).
- **tipo de servicio (8bits)**: Pensado originalmente para poder “etiquetar” los datagramas y darles distinto tratamiento:
 - minimizar retardo
 - maximizar rendimiento
 - maximizar fiabilidad

Este campo ha sido redefinido de otras formas, pero normalmente es ignorado por los encaminadores.

- **longitud total datagrama (16bits)**: Tamaño del datagrama en bytes (cabecera + datos). Tamaño máximo: 65535 bytes. Dependiendo del tamaño máximo del campo de datos de la trama en el nivel de enlace, puede ser necesario fragmentar.

Campos de la cabecera del datagrama IP: datagramas de longitud > 1500 bytes

Si la longitud total del datagrama es
>1500 bytes no se puede enviar a través
de Ethernet. Es necesario fragmentar.

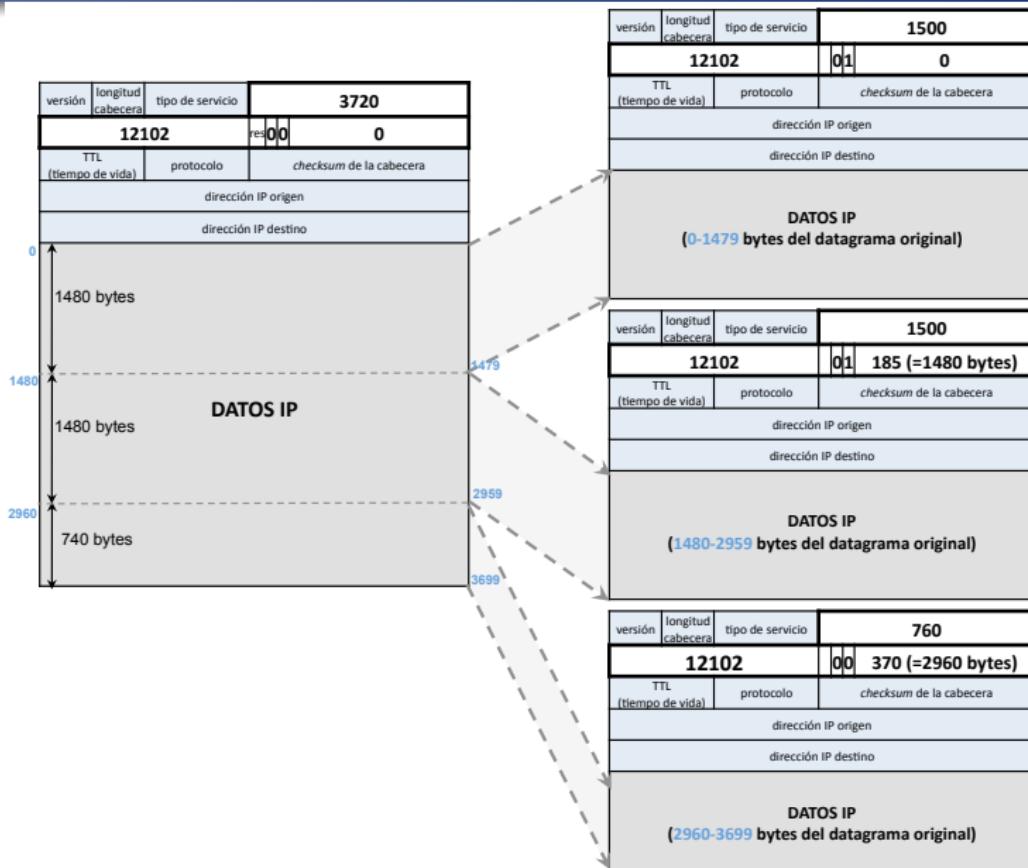


Campos de la cabecera del datagrama IP: fragmentación

Fragmentación: dividir en fragmentos un datagrama IP demasiado grande para encapsularlo en la unidad de datos del nivel de enlace. Los siguientes campos de la cabecera se usan para gestionar la fragmentación.

- **identificador del datagrama:** Número de 16 bits, diferente para cada datagrama que genera una máquina (se va incrementando de 1 en 1). Los distintos fragmentos del mismo datagrama tienen el mismo valor de identificador.
- **flag DF (*Don't Fragment*):** Cuando va a 1, indica que el datagrama no puede fragmentarse. Si no cabe en la trama, se descartará y se enviará un mensaje ICMP de error al origen del datagrama.
- **flag MF (*More Fragments*):** Cuando tiene valor 0 indica que el datagrama no está fragmentado o que se trata del último fragmento. Cuando va a 1, indica que el datagrama es un fragmento, y que quedan fragmentos posteriores.
- **offset de fragmentación:** Indica la posición relativa de los bytes de datos contenidos en este fragmento con respecto al datagrama original. Se mide en palabras de 8 bytes (todos los fragmentos excepto el último han de tener un tamaño del campo de datos múltiplo de 8 bytes).

Ejemplo de Fragmentación



Campos de la cabecera del datagrama IP: TTL, protocolo y checksum

- **TTL:** Indica el número máximo de encaminadores que el datagrama puede atravesar. Normalmente se inicializa a 64 en los datagramas recién creados. Cada encaminador resta 1 al TTL. Si al hacer la resta queda 0 se descarta el datagrama, enviándose a la dirección de origen un mensaje ICMP de error.
- **protocolo:** Indica el protocolo al que corresponden los datos del datagrama. Utilizado para demultiplexar en el destino:
 - 0x01: ICMP
 - 0x06: TCP
 - 0x11: UDP
 - ...
- **checksum:** Calculado sobre la cabecera del datagrama. Tiene el propósito de detectar:
 - Erratas en el SW de las máquinas por las que va pasando el datagrama, que podrían haberlo alterado (ej: una errata en la implementación de IP de ciertos modelos de *router*)
 - Errores en el HW de las máquinas por las que va pasando el datagrama, que podrían haberlo alterado (ej: un chip de memoria estropeado en un *router*).

Campos de la cabecera del datagrama IP: direcciones IP, opciones

- **direcciones origen y destino:** Direcciones de las interfaces de red de las máquinas origen y destino que se intercambian el datagrama.
- **opciones:** Cero o más opciones. Este campo se rellena con ceros por la derecha para asegurar un tamaño múltiplo de 32 bits. Los *routers* no están obligados a reconocer ni procesar las opciones.

Contenidos

1 Protocolo IP

- Formato del datagrama IP
- **Direcciones IP**
- Tablas de encaminamiento IP

2 Protocolo ARP

3 Ejemplo de encaminamiento

- Comunicación entre máquinas vecinas
- Comunicación entre máquinas NO vecinas

4 El problema inverso al ARP

5 Protocolo ICMP

6 Asignación y planificación de subredes IP

7 Ejemplo: traceroute

8 Congestión

9 Referencias

Direcciones IP

- Las direcciones IP tienen 32 bits.
- Se asigna una dirección IP a cada interfaz física conectada a Internet.
- Los 32 bits se expresan: cada byte en decimal (0–255), los bytes separados por puntos: 212.128.4.4.
- Una dirección IP está dividida en dos partes:
 - **Identificador de red:** conjunto de bits que identifica la red a la que está conectada una máquina.
 - **Identificador de máquina:** conjunto de bits único para cada máquina de una red.
- Ejemplo para la dirección 212.128.4.4:

Identificador de red	Id. de máquina
212.128.4	4

Máscara de red (*netmask*)

- Se utiliza para identificar hasta donde llega la parte de red de una dirección IP y, por lo tanto, donde comienza la parte de máquina.
- Se representa como un número de 32 bits. En binario:
 - hay unos en las posiciones del identificador de red
 - hay ceros las posiciones del identificador de máquina
- Para la siguiente dirección IP

Identificador de red	Id. de máquina
212.128.4	4

La máscara de red de esta dirección IP quedaría representada por:

Identificador de red	Id. de máquina
En binario: 1111 1111 . 1111 1111 . 1111 1111	0000 0000

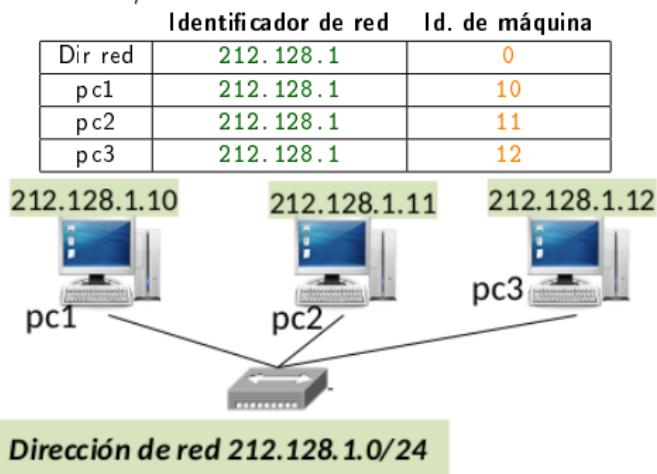
- La máscara se puede representar:
 - con la notación **decimal** separada por puntos (como si fuera una dirección IP)
 - con un **prefijo** que indica el número de unos de la máscara

Así, para el ejemplo anterior:

	Id. de red	Id. de máquina
Máscara en decimal:	255.255.255.	0
Máscara en modo prefijo:	/24	

Dirección de red

- Todas las máquinas de la misma comparten el mismo identificador de red y se diferencian por el identificador de máquina. Además, todas comparten la misma máscara.
- Se denomina **dirección de red** aquella que tiene un determinado id de red y el id de máquina está a cero.
 - Por ejemplo: 212.128.1.0/24



- En la red 212.128.1.0/24: pc1, pc2 y pc3 son máquinas vecinas y pertenecen a la misma red IP
- La dirección de red se usa en las tablas de *routing*, como veremos más adelante.

Dirección de broadcast

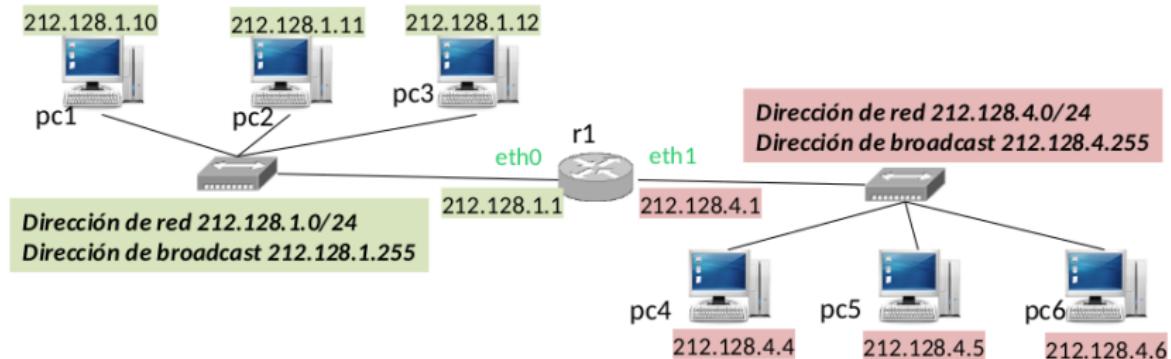
- La **dirección de broadcast** de una red se utiliza para comunicarse con todas las máquinas conectadas a dicha red.
- Se forma con el identificador de red y todos los bits del identificador de máquina **a 1**.
- Es el último número posible para dicha red.

	Identificador de red	Id. de máquina
Dir red/subred	212.128.1	0
pc1	212.128.1	10
pc2	212.128.1	11
pc3	212.128.1	12
...
Dir. broadcast de subred	212.128.1	255

- Dado un identificador de red, se pueden utilizar para máquinas cualquier identificador de máquina excepto el primero (reservado para la dirección de red) y el último (reservado para la dirección de broadcast).
- La dirección de broadcast puede utilizarse como dirección de destino de un datagrama IP para enviar dicho datagrama a todas las máquinas vecinas.

Interconexión de redes IP

- El dispositivo de interconexión de diferentes redes IP es el **router IP**.
- Las redes 212.128.1.0/24 y 212.128.4.0/24 están conectadas a través de r1.



- En la red 212.128.1.0/24:
 - pc1, pc2, pc3 y r1(eth0) son máquinas vecinas.
- En la red 212.128.4.0/24:
 - pc4, pc5, pc6 y r1(eth1) son máquinas vecinas.
- Las máquinas pc1, pc2 y pc3 no son vecinas de pc4, pc5 y pc6 ya que no están conectadas al mismo nivel de enlace.
- Las direcciones de red de 2 redes conectadas al mismo *router* NO tienen por qué ser parecidas, podrían ser completamente distintas.

Contenidos

1 Protocolo IP

- Formato del datagrama IP
- Direcciones IP
- Tablas de encaminamiento IP

2 Protocolo ARP

3 Ejemplo de encaminamiento

- Comunicación entre máquinas vecinas
- Comunicación entre máquinas NO vecinas

4 El problema inverso al ARP

5 Protocolo ICMP

6 Asignación y planificación de subredes IP

7 Ejemplo: traceroute

8 Congestión

9 Referencias

Máquinas que tienen activado el *routing*

Cualquier máquina IP puede estar o no configurada para hacer *routing*:

- Cuando **NO** lo está, los datagramas IP que recibe y no son para ella, **los descarta**.
- Cuando **SÍ** lo está, los datagramas IP que recibe y no son para ella **se tratan de encaminar** (es decir, se intenta reenviarlos para que progresen hacia su destino final).

Configuración normal:

- máquinas con **sólo una interfaz de red física**: NO tienen activado el *routing*
- máquinas con **dos o más interfaces de red físicas**: SÍ tienen activado el *routing*. A estas máquinas se les llama *routers*

Búsqueda en la Tabla de Encaminamiento

- Todas las máquinas tienen tabla de encaminamiento, aunque no tengan activado el *routing*.
- Cuando una máquina quiere enviar un datagrama IP a un destino (porque lo construye ella, o porque es un *router* y le ha llegado de otro sitio) consulta su **tabla de encaminamiento**.
- En la tabla se busca si encaja la IP destino en la primera columna de alguna entrada:
 - ① buscando si existe una **entrada con una dirección de máquina igual a la de destino del datagrama**,
 - ② si no, buscando si existe una **entrada con una dirección de red igual a la parte de red de la IP de destino del datagrama**,
 - ③ si no, buscando si existe una **entrada por defecto** (0.0.0.0 en la primera columna).
- Si no hubiera ninguna entrada adecuada en la tabla (ni siquiera una entrada por defecto), se descarta el datagrama IP

Ejemplo de Tabla de Encaminamiento

Las tablas de encaminamiento tienen el siguiente aspecto (ejemplo tomado de una máquina Linux):

<code>% route -n</code>			
Kernel IP routing table			
Destination	Gateway	Genmask	Iface
193.147.71.0	0.0.0.0	255.255.255.0	eth0
212.128.4.0	0.0.0.0	255.255.255.0	eth1
145.154.12.0	193.147.71.2	255.255.255.0	eth0
145.154.12.14	212.128.4.2	255.255.255.255	eth1
0.0.0.0	193.147.71.1	0.0.0.0	eth0

- Si el “Gateway” es 0.0.0.0 quiere decir que el Destino es alcanzable directamente sin pasar por ningún router intermedio.
 - Puede aparecer * en vez de 0.0.0.0 en la columna “Gateway”
- Si el “Destination” es 0.0.0.0 esa ruta es la **ruta por defecto**, aplicable cuando no hay una ruta más específica.
 - Puede aparecer default en vez de 0.0.0.0 en la columna “Destination”

Contenidos

- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

ARP (*Address Resolution Protocol*)

- El protocolo ARP permite averiguar la dirección Ethernet de una máquina sabiendo su dirección IP.
- Cuando el nivel IP va a enviar un datagrama con una cierta dirección IP de destino:
 - Si la dirección de destino es de la misma subred, esa máquina es directamente a quien hay que enviar la trama Ethernet que contenga el datagrama.
 - Si no, de la tabla de encaminamiento se obtiene la dirección IP del siguiente salto, que es el encaminador a quien hay que enviar la trama Ethernet que contenga el datagrama.
- En cualquiera de los dos casos, sólo se conoce la dirección IP de la máquina adyacente a la que pasar el datagrama, pero para formar la trama Ethernet se necesita conocer la dirección Ethernet de esa máquina.

Funcionamiento

Para conocer la dirección Ethernet de una máquina de la misma subred, dada su dirección IP, una máquina hace lo siguiente:

- ① Envía una trama Ethernet de *broadcast* que contiene una **solicitud ARP** en la que se incluye la dirección IP en cuestión.
- ② De entre todas las máquinas adyacentes, contestará aquella cuya IP va en la solicitud de ARP. Contestará con una trama Ethernet dirigida a quien hizo la pregunta, conteniendo una **respuesta ARP** indicando la dirección Ethernet pedida.

Cada máquina mantiene una **caché de ARP o tabla de vecinos** de correspondencias entre direcciones IP a direcciones Ethernet con los resultados de las solicitudes que va haciendo.

Formato del mensaje de ARP

- Mismos campos para solicitudes y respuestas:

Solicitud/Respuesta	Eth. Origen	IP Origen	Eth. Destino	IP Destino
---------------------	-------------	-----------	--------------	------------

- No hay que olvidar que el mensaje de ARP viaja dentro de una trama Ethernet (si es ése el nivel de enlace).
- En una **Solicitud**:
 - los campos **Origen** llevan los datos de la máquina que pregunta
 - de los campos **Destino** sólo va relleno **IP Destino**, ya que **Eth. Destino** es justo lo que se pregunta
- En una **Respuesta**,
 - los campos **Origen** llevan los datos de la máquina que responde, con lo que el campo **Eth. Origen** es la Ethernet por la que se preguntaba
 - los campos **Destino** llevan los datos de la máquina que hizo la solicitud

Contenidos

- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

Contenidos

1 Protocolo IP

- Formato del datagrama IP
- Direcciones IP
- Tablas de encaminamiento IP

2 Protocolo ARP

3 Ejemplo de encaminamiento

- Comunicación entre máquinas vecinas
- Comunicación entre máquinas NO vecinas

4 El problema inverso al ARP

5 Protocolo ICMP

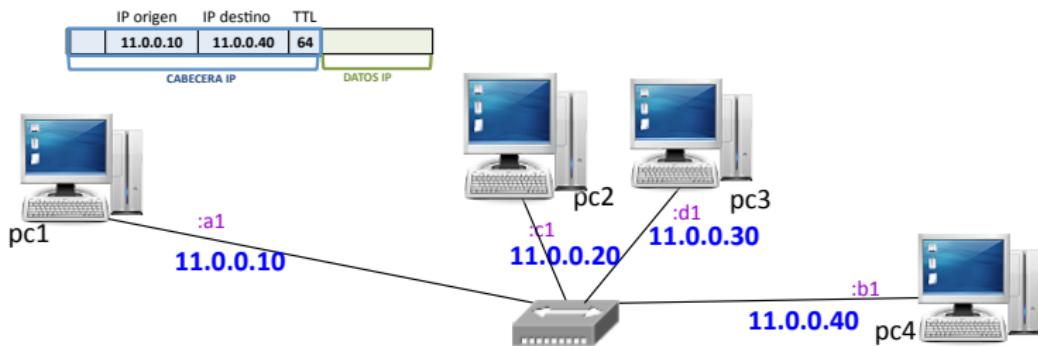
6 Asignación y planificación de subredes IP

7 Ejemplo: traceroute

8 Congestión

9 Referencias

Envío de pc1 a pc4



Envío pc1 -> pc4

- pc1 dispone de un datagrama IP para enviar a pc4

Envío de pc1 a pc4

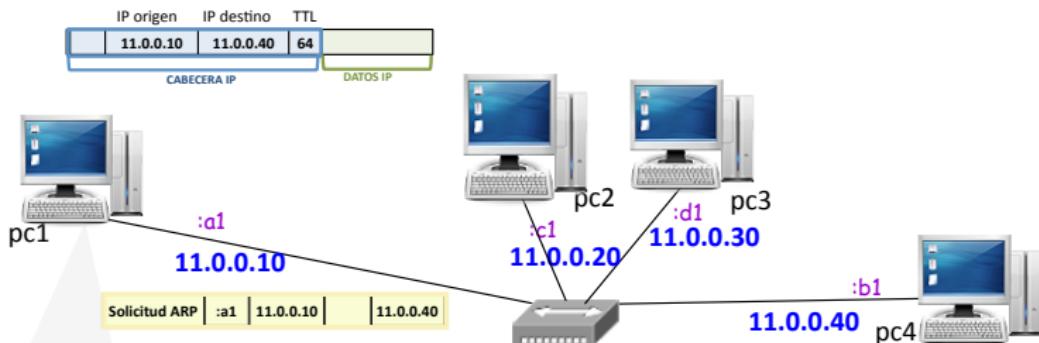


Tabla de encaminamiento en pc1		
Destino	Gateway	Máscara
11.0.0.0	0.0.0.0	255.255.255.0

Caché ARP en pc1		
IP	Ethernet	Interfaz

Envío pc1 -> pc4

- pc1 dispone de un datagrama IP para enviar a pc4
- pc1 consulta tabla de encaminamiento, necesita la dir Ethernet de pc4 para enviar la trama Ethernet. pc1 envía solicitud de ARP ¿Ethernet de pc4?

Envío de pc1 a pc4

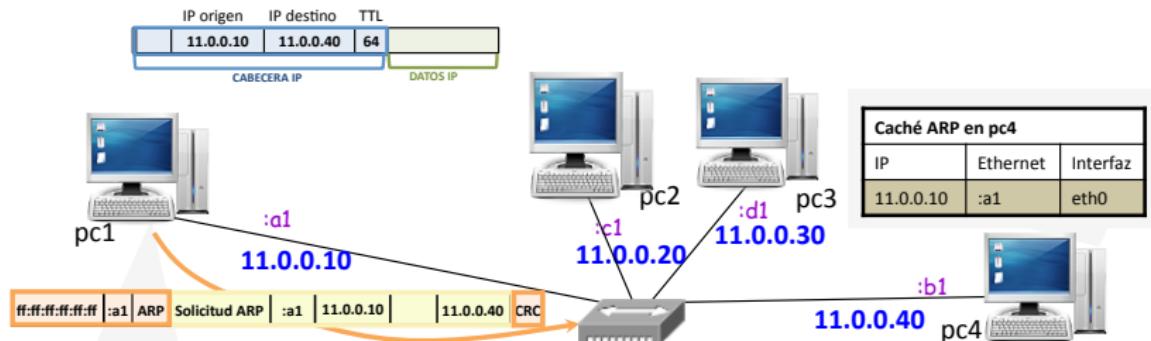


Tabla de encaminamiento en pc1		
Destino	Gateway	Máscara
11.0.0.0	0.0.0.0	255.255.255.0

Caché ARP en pc1		
IP	Ethernet	Interfaz

Envío pc1 -> pc4

- pc1 dispone de un datagrama IP para enviar a pc4
- pc1 consulta tabla de encaminamiento, necesita la dir Ethernet de pc4 para enviar la trama Ethernet. pc1 envía solicitud de ARP ¿Ethernet de pc4?
- La solicitud ARP se envía a la dir Broadcast Ethernet. La máquina pc4 aprende la asociación entre dir Ethernet y dir IP de pc1.

Envío de pc1 a pc4

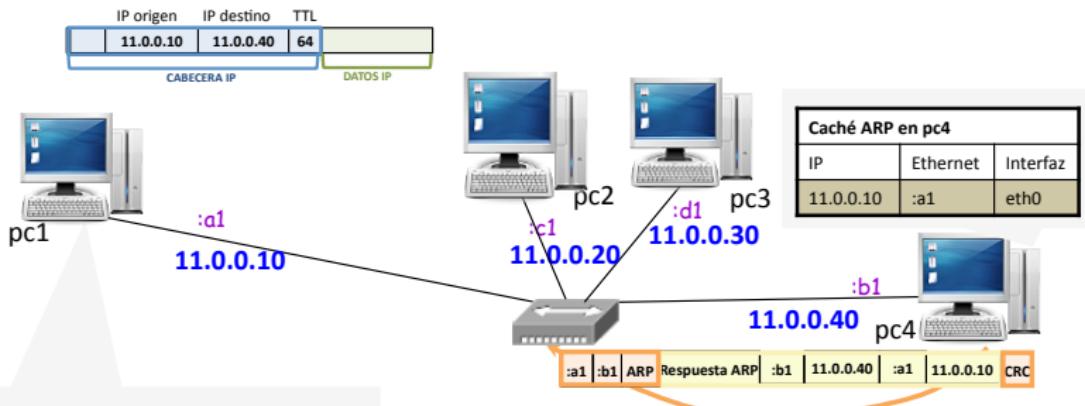


Tabla de encaminamiento en pc1		
Destino	Gateway	Máscara
11.0.0.0	0.0.0.0	255.255.255.0

Caché ARP en pc1		
IP	Ethernet	Interfaz
11.0.0.40	:b1	eth0

Envío pc1 -> pc4

- pc1 dispone de un datagrama IP para enviar a pc4
- pc1 consulta tabla de encaminamiento, necesita la dir Ethernet de pc4 para enviar la trama Ethernet. pc1 envía solicitud de ARP ¿Ethernet de pc4?
- La solicitud ARP se envía a la dir Broadcast Ethernet. La máquina pc4 aprende la asociación entre dir Ethernet y dir IP de pc1.
- pc4 responde a pc1 con mensaje ARP su dir Ethernet y pc1 la apunta en su caché ARP

Envío de pc1 a pc4

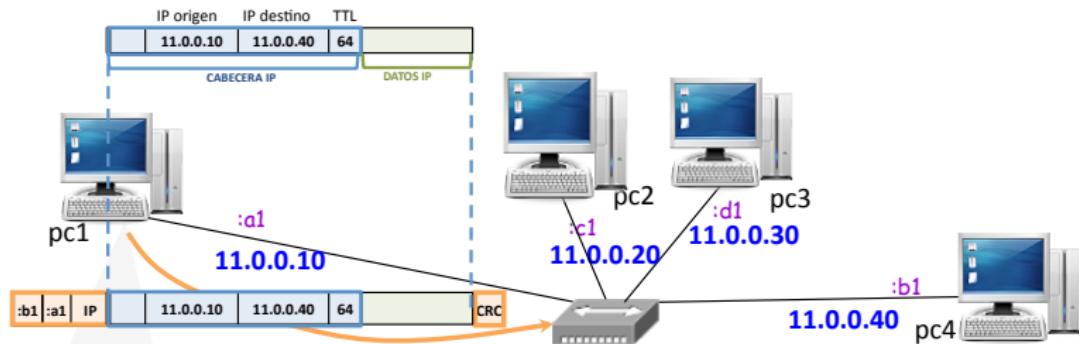


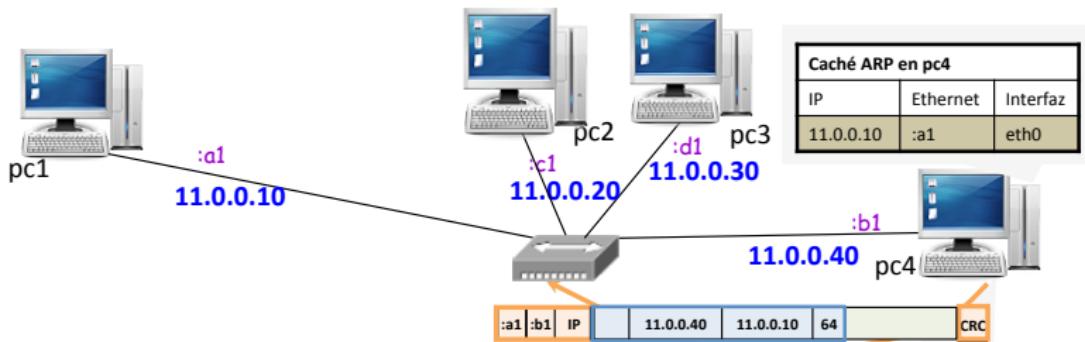
Tabla de encaminamiento en pc1		
Destino	Gateway	Máscara
11.0.0.0	0.0.0.0	255.255.255.0

Caché ARP en pc1		
IP	Ethernet	Interfaz
11.0.0.40	:b1	eth0

Envío pc1 -> pc4

- pc1 dispone de un datagrama IP para enviar a pc4
- pc1 consulta tabla de encaminamiento, necesita la dir Ethernet de pc4 para enviar la trama Ethernet. pc1 envía solicitud de ARP ¿Ethernet de pc4?
- La solicitud ARP se envía a la dir Broadcast Ethernet. La máquina pc4 aprende la asociación entre dir Ethernet y dir IP de pc1.
- pc4 responde a pc1 con mensaje ARP su dir Ethernet y pc1 la apunta en su caché ARP
- pc1 envía trama Ethernet a pc4

Envío de pc4 a pc1



Envío pc4 -> pc1

- Si ahora pc4 quiere "responder" a pc1 con un datagrama IP, lo prepara, mira su tabla de encaminamiento, y al tener ya en la caché de ARP la dirección Ethernet de pc1, construye la trama con el datagrama IP directamente: no se necesita ejecutar el protocolo ARP.

Contenidos

1 Protocolo IP

- Formato del datagrama IP
- Direcciones IP
- Tablas de encaminamiento IP

2 Protocolo ARP

3 Ejemplo de encaminamiento

- Comunicación entre máquinas vecinas
- Comunicación entre máquinas NO vecinas

4 El problema inverso al ARP

5 Protocolo ICMP

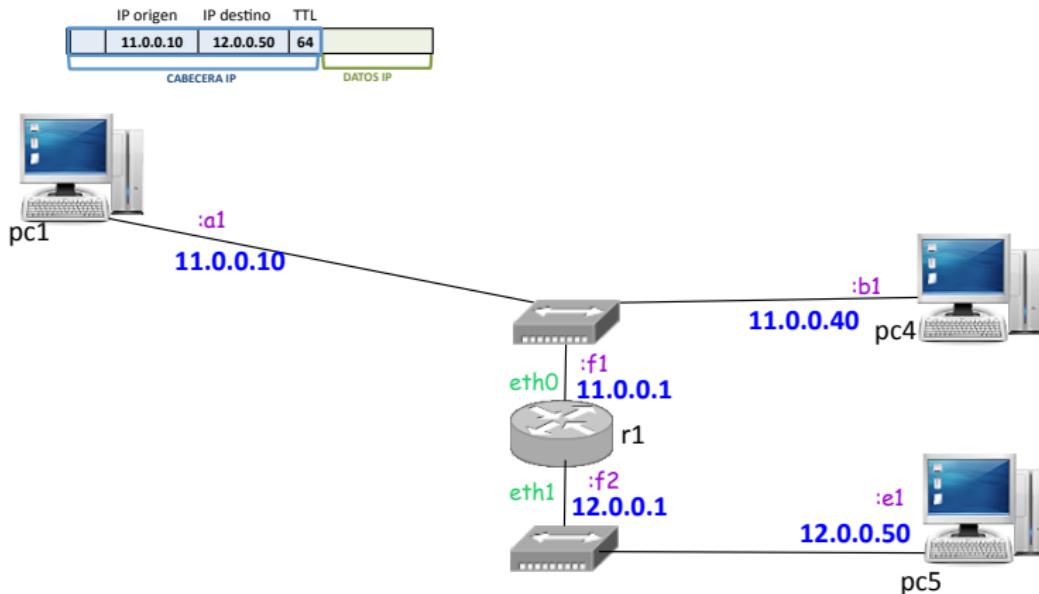
6 Asignación y planificación de subredes IP

7 Ejemplo: traceroute

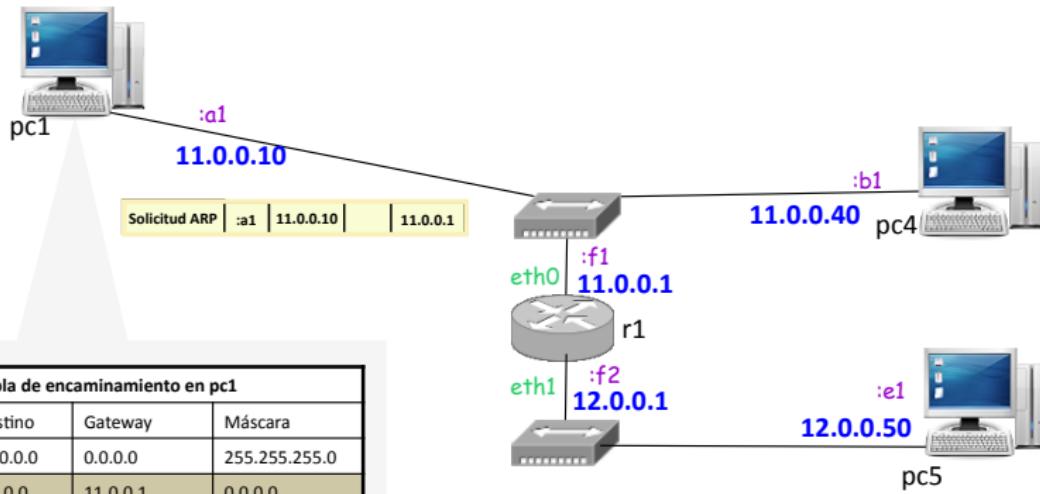
8 Congestión

9 Referencias

Envío de pc1 a pc5

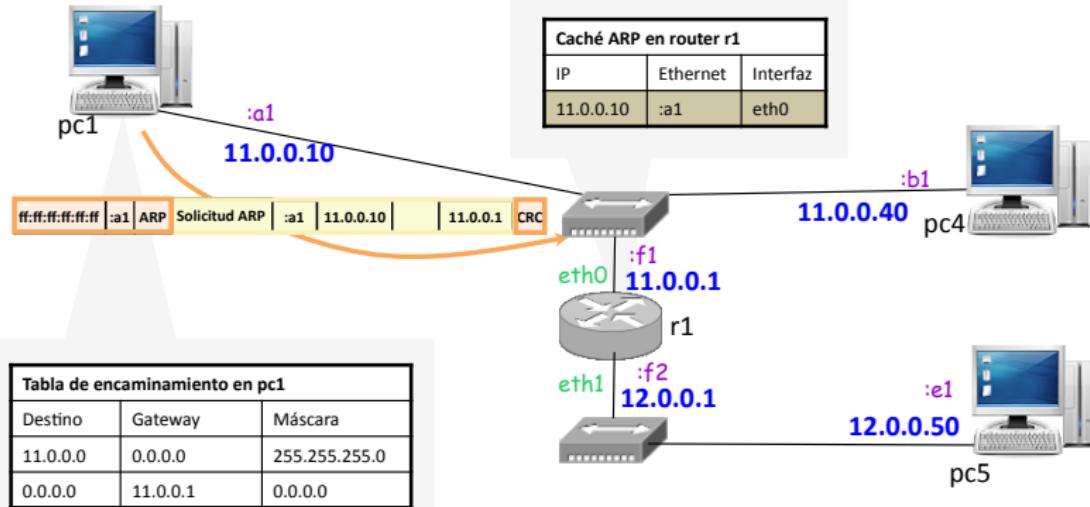


Envío de pc1 a pc5



Envío de pc1 a pc5

IP origen	IP destino	TTL	
CABECERA IP			DATOS IP
11.0.0.10			



Envío de pc1 a pc5

IP origen	IP destino	TTL	DATOS IP
11.0.0.10	12.0.0.50	64	

CABECERA IP DATOS IP

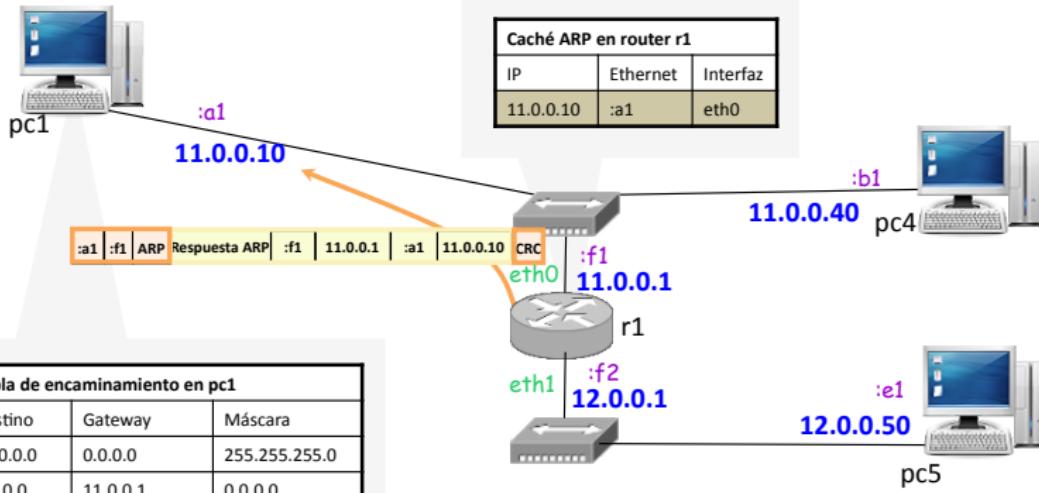


Tabla de encaminamiento en pc1		
Destino	Gateway	Máscara
11.0.0.0	0.0.0.0	255.255.255.0
0.0.0.0	11.0.0.1	0.0.0.0

Caché ARP en pc1		
IP	Ethernet	Interfaz
11.0.0.40	:b1	eth0
11.0.0.1	:f1	eth0

Envío de pc1 a pc5

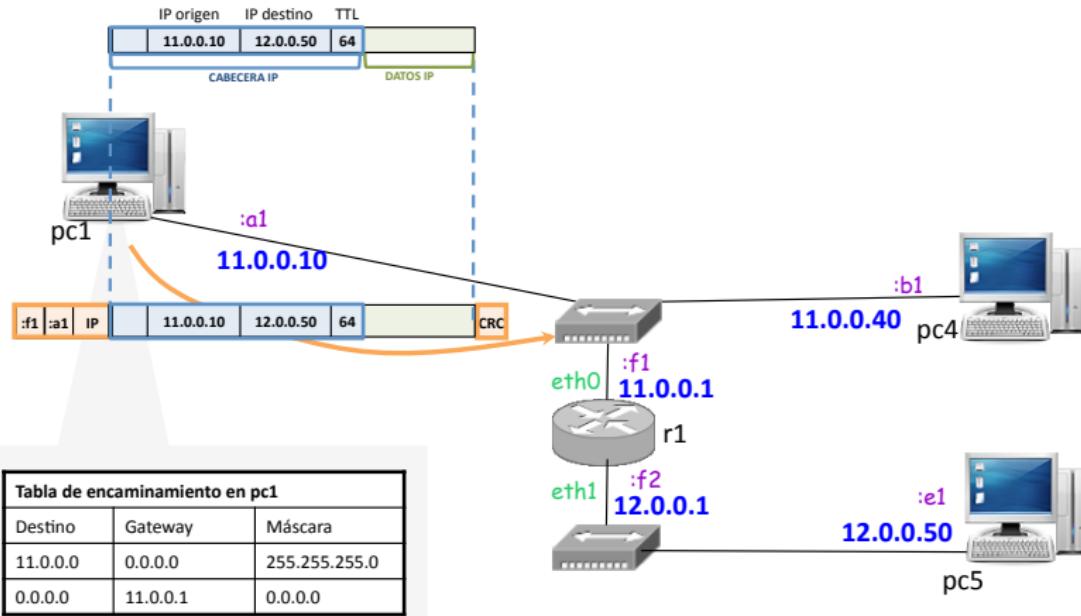
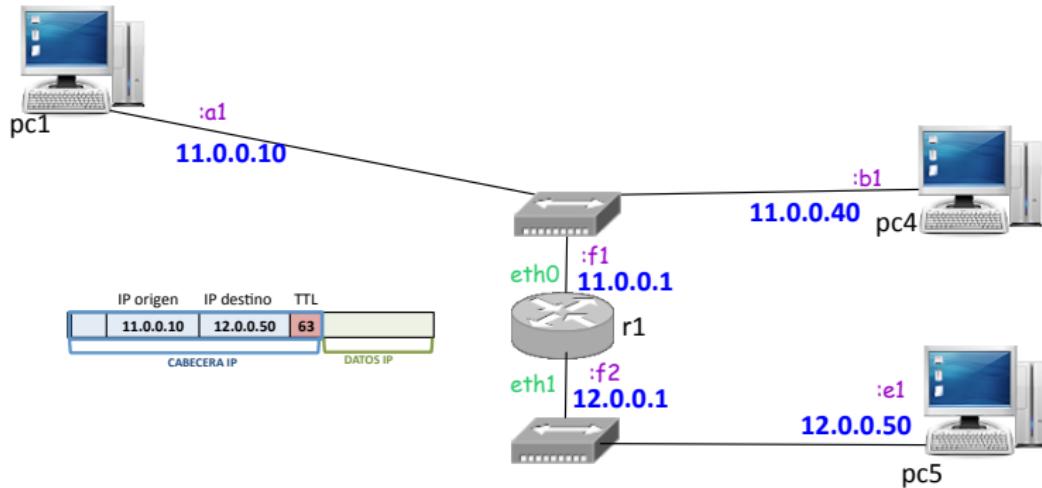


Tabla de encaminamiento en pc1		
Destino	Gateway	Máscara
11.0.0.0	0.0.0.0	255.255.255.0
0.0.0.0	11.0.0.1	0.0.0.0

Caché ARP en pc1		
IP	Ethernet	Interfaz
11.0.0.40	:b1	eth0
11.0.0.1	:f1	eth0

Envío de pc1 a pc5



Envío de pc1 a pc5

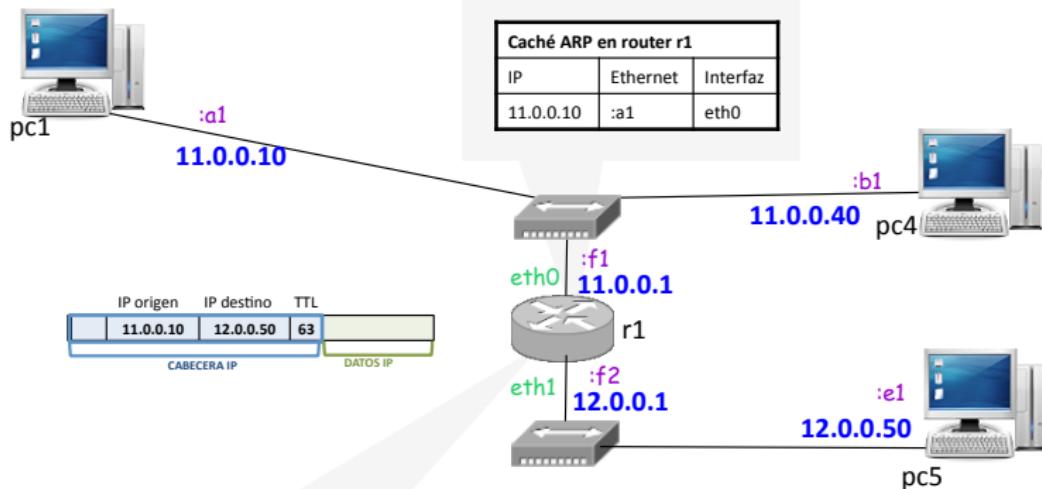
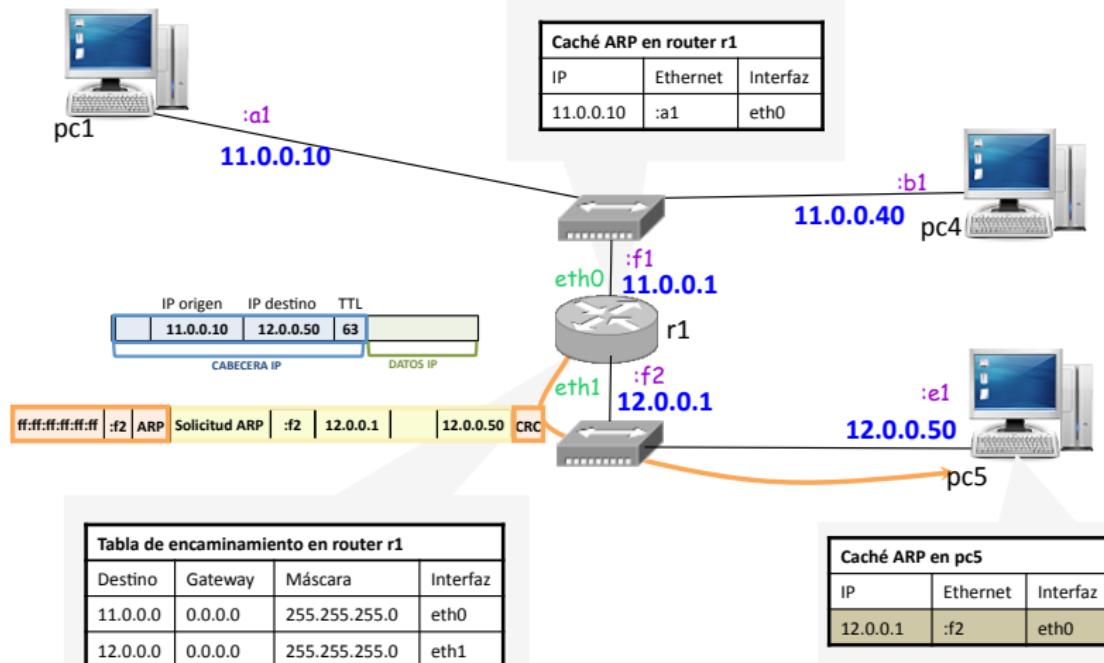
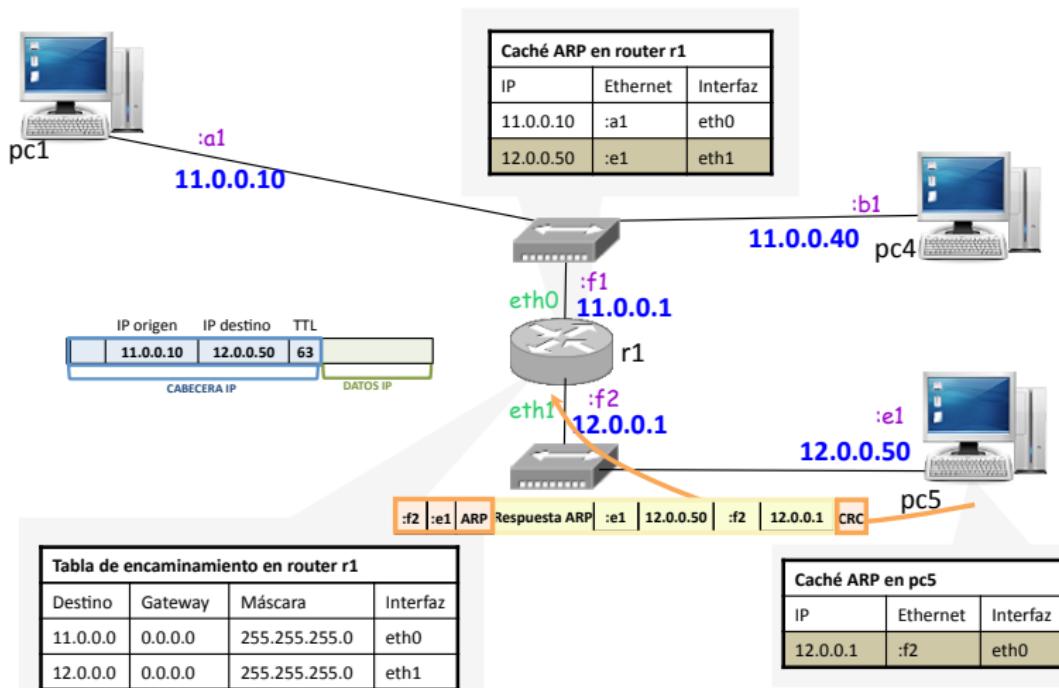


Tabla de encaminamiento en router r1			
Destino	Gateway	Máscara	Interfaz
11.0.0.0	0.0.0.0	255.255.255.0	eth0
12.0.0.0	0.0.0.0	255.255.255.0	eth1

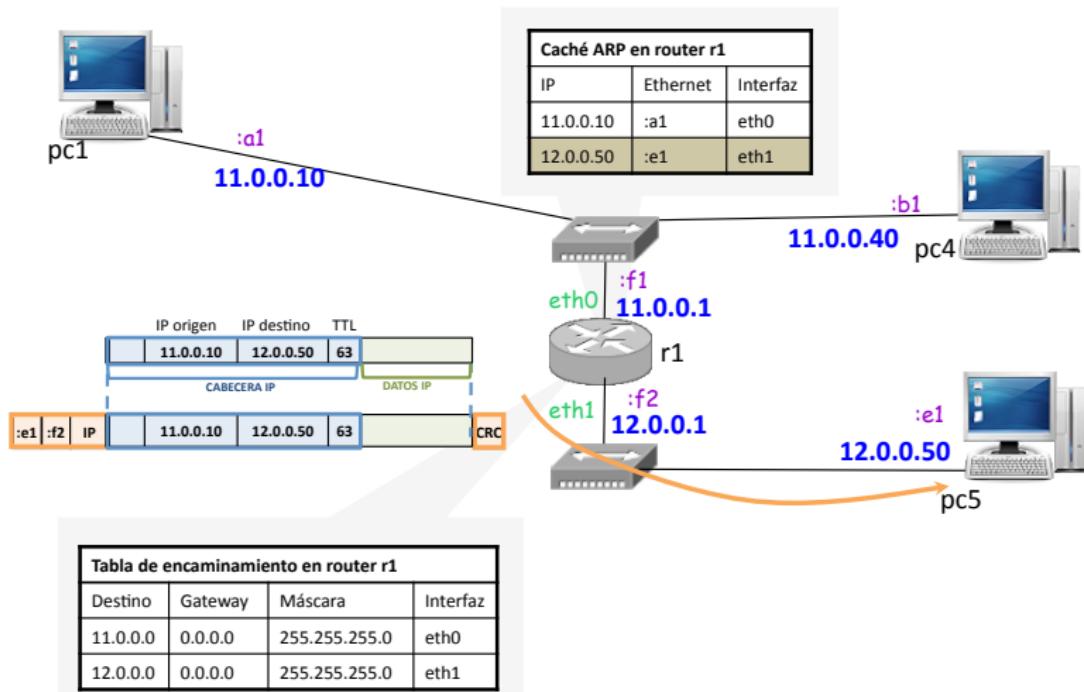
Envío de pc1 a pc5



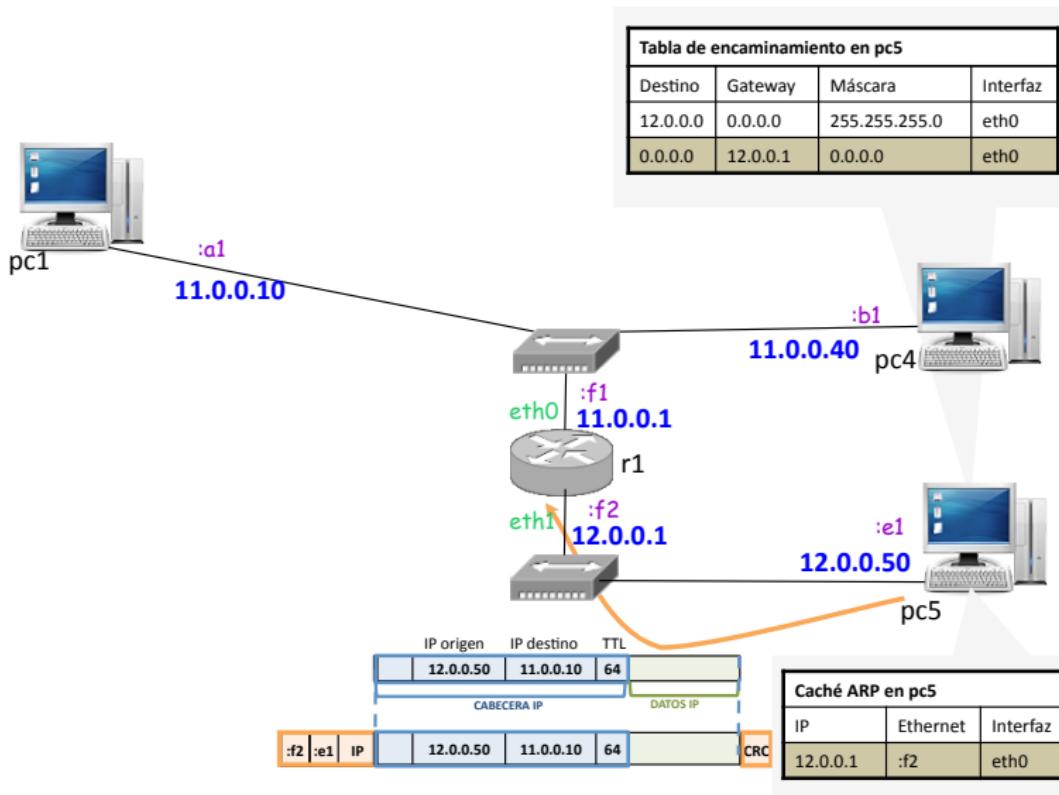
Envío de pc1 a pc5



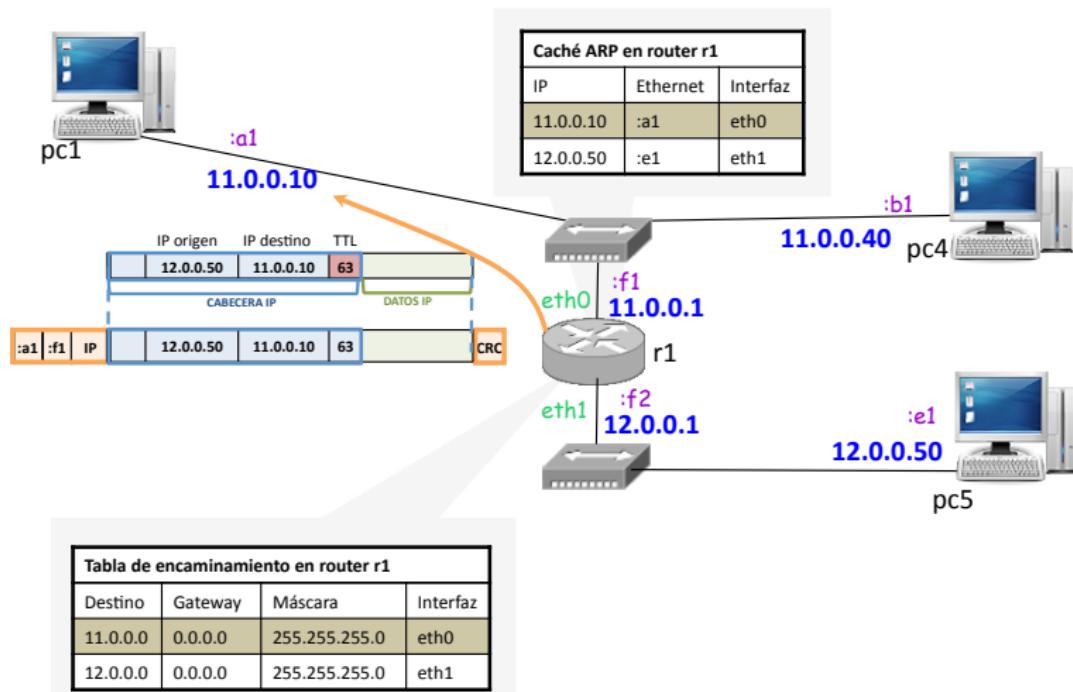
Envío de pc1 a pc5



Envío de pc5 a pc1



Envío de pc5 a pc1



Mecanismos adicionales

- Aprovechando que las solicitudes de ARP se envían con un *broadcast*, para cada solicitud que vea una máquina, si la IP Origen está en su caché de ARP, la máquina actualiza la entrada con la Eth. Origen que va en la solicitud.
- **ARP gratuito:** Una máquina puede enviar una solicitud ARP preguntando sobre su propia dirección IP. Propósitos:
 - detectar direcciones IP duplicadas
 - forzar a que los que tengan una entrada en su caché la actualicen
- **En Linux:** Solicitud de ARP para confirmar una dirección aprendida indirectamente:
 - Cuando A envía una solicitud de ARP preguntando por B, B (además de responder) aprende indirectamente la Ethernet de A y la guarda en su caché de ARP
 - Algunas implementaciones (como en Linux) a los 5 segundos intentan confirmar esta dirección: B hace una solicitud de ARP preguntando por la Ethernet de A
 - Estas solicitudes son especiales y no van dirigidas a la dirección de broadcast, sino a la máquina concreta cuya dirección se quiere confirmar (en el ejemplo, B envía la solicitud a la Ethernet de A).

Contenidos

- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

El problema inverso al ARP

- A veces se da la situación inversa a la del ARP: se quiere conocer una IP dada la dirección Ethernet.
- Casi siempre se da este caso cuando una máquina, al arrancar, sabe su propia dirección Ethernet (está en la tarjeta Ethernet), pero no su dirección IP. Ejemplos:
 - Máquina sin disco duro en el que tener un fichero con su dirección IP
 - Máquina que acaba de conectarse a una red
 - No se desea configurar direcciones fijas en un fichero en cada máquina de una red
- Existen varios protocolos para resolver este problema. Todos funcionan de forma similar:
 - ① La máquina envía un *broadcast* con una **solicitud**, indicando su dirección Ethernet.
 - ② Alguna máquina de su subred le enviará una **respuesta**, indicándole cuál es su dirección IP.

Protocolos RARP, BOOTP, DHCP

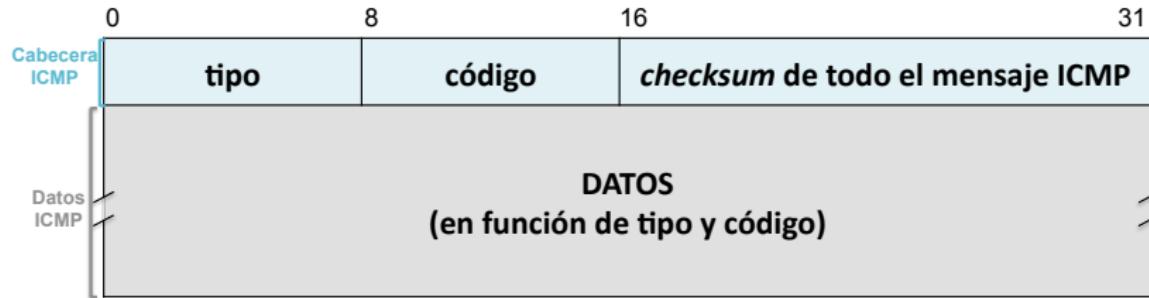
- Originalmente se usaba **RARP** (*Reverse Address Resolution Protocol*) protocolo muy simple con un formato de mensaje igual a ARP.
- Otro protocolo sencillo para el mismo propósito es **BOOTP** (*Bootstrap Protocol*), que permite, además, arrancar un sistema operativo a través de la red (en vez de leerlo del disco duro).
- El protocolo más usado para asignar una IP a una máquina al arrancar es **DHCP** (*Dynamic Host Configuration Protocol*), que permite, además de conocer la IP y arrancar un sistema operativo:
 - Configurar nombres de máquina, máscaras, puertas de enlace y otros servicios.
 - Realizar la asignación de direcciones de forma:
 - Manual: Cierta ethernet siempre tendrá la misma IP, indicada por el administrador.
 - Dinámica: El administrador reserva un conjunto de IPs, que se asignarán a las máquinas bajo demanda.

Contenidos

- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

ICMP (*Internet Control Message Protocol*)

- Este protocolo se utiliza para comunicar condiciones de error entre máquinas y para realizar algunas funciones de diagnóstico.
- Los mensajes ICMP se transmiten encapsulados dentro de datagramas IP.
- Formato de los mensajes ICMP:



Mensajes ICMP

Algunos mensajes ICMP:

tipo	código	descripción
0	0	respuesta de eco
3	0	destino inalcanzable: red inalcanzable
3	1	destino inalcanzable: máquina inalcanzable
3	3	destino inalcanzable: puerto inalcanzable
8	0	solicitud de eco
11	0	tiempo excedido: TTL = 0
12	1	cabecera IP incorrecta: falta una opción
13	0	solicitud de marca de tiempo
14	0	respuesta de marca de tiempo

Contenidos

- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

¿Quién asigna direcciones IP?

- Cinco organismos regionales: ARIN (Norteamérica), RIPE (Europa, Asia Central), AfriNIC (África), APIC (Asia y Zona del Pacífico) y LACNIC (Latinoamérica).



- RIPE delega en organismos regionales (normalmente por países).
- En España, el organismo es **Red.es**, Entidad Pública Empresarial adscrita al Ministerio de Ciencia y Tecnología.
- Una organización adquiere una (o más) direcciones de clase, y el administrador local de la organización reparte la dirección de clase entre todas sus máquinas.
- **ICANN** (Internet Corporation for Assigned Names and Numbers) es una organización internacional creada en 1998 para coordinar todas las tareas de asignación de nombres y direcciones.

Clases de direcciones IP

- Cuando se establece el direccionamiento IP, los identificadores de red son de una longitud fija.
- Las direcciones IP se dividen en 5 clases. Sólo las direcciones de las 3 primeras clases se utilizan para ser asignadas a máquinas.
- En cada clase es distinto el tamaño de los identificadores de red y de máquina.

				7 bits		24 bits	
Clase A	0		id red			id maquina	
Clase B	1	0		id red		id maquina	
Clase C	1	1	0		id red		id maquina
Clase D	1	1	1	0		id grupo multicast	
Clase E	1	1	1	1		reservado para usos futuros	

Rango de las direcciones de clase

- El rango de direcciones IP para las 5 **clases** de direcciones :

Clase	Rango	Máscara dec.	Prefijo
A	0.0.0.0 a 127.255.255.255	255.0.0.0	/8
B	128.0.0.0 a 191.255.255.255	255.255.0.0	/16
C	192.0.0.0 a 223.255.255.255	255.255.255.0	/24
D	224.0.0.0 a 239.255.255.255		
E	240.0.0.0 a 255.255.255.255		

Direcciones IP para redes privadas

- Existen unos rangos de direcciones IP reservadas para ámbito local, y que no son utilizables en máquinas conectadas directamente a Internet:

Desde	10.0.0.0	hasta	10.255.255.255
Desde	169.254.0.0	hasta	169.254.255.255
Desde	172.16.0.0	hasta	172.31.255.255
Desde	192.168.0.0	hasta	192.168.255.255

- Los encaminadores de Internet descartan los datagramas con destino a una de estas direcciones IP.
- Estas direcciones suelen denominarse **direcciones IP privadas**. Por oposición, el resto de direcciones reciben el nombre de **direcciones IP públicas**.

Classless Inter-Domain Routing (CIDR) (I)

- Inicialmente se asignaban las direcciones IP teniendo en cuenta el sistema de clases de direcciones:
 - Una dirección de clase A (/8): permite $\approx 2^{24}$ máquinas
 - Una dirección de clase B (/16): permite $\approx 2^{16}$ máquinas
 - Una dirección de clase C (/24): permite $\approx 2^8$ máquinas
- A cada organización se le asignaba una clase entera.
- Número de direcciones de clase diferentes (para asignar a organizaciones):
 - Hay 2^7 direcciones de clase A diferentes (y son la mitad de todas las direcciones IP)
 - Hay 2^{14} direcciones de clase B diferentes (y son la cuarta parte de todas las direcciones IP)
 - Hay 2^{21} direcciones de clase C diferentes (y son la octava parte de todas las direcciones IP)
- La diferencia tan grande en la cantidad de máquinas entre las diferentes clases hacía que una clase A fuera muy grande y una clase C muy pequeña.
- Este sistema de asignación provocó que las direcciones IP se agotaran muy rápidamente. (sólo quedaban libres clases C no contiguas).

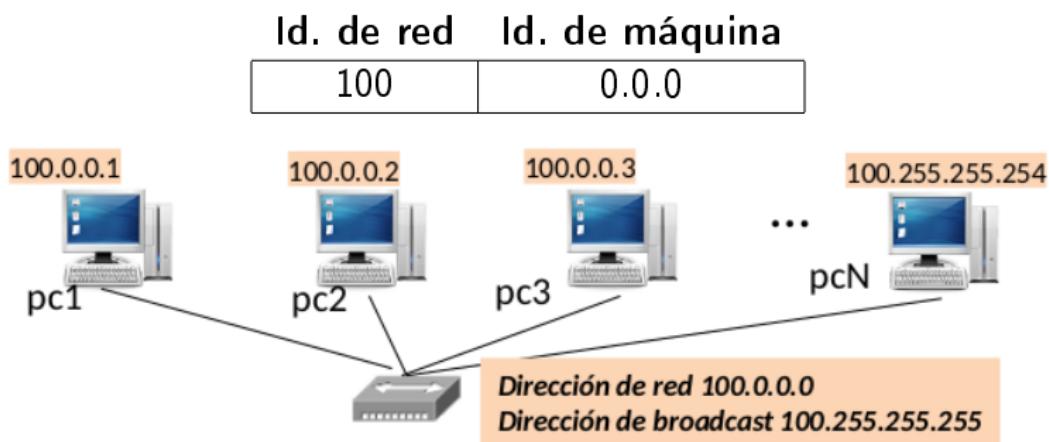
Classless Inter-Domain Routing (CIDR) (II)

- Otro problema del antiguo sistema de clases: Las tablas de encaminamiento de los *routers* del “centro de Internet” tenían demasiadas entradas: una entrada para cada dirección de clase diferente.
- El problema del agotamiento de direcciones y del tamaño de las tablas de encaminamiento hicieron cambiar el sistema de asignaciones a un sistema que no tuviera en cuenta las clases: **CIDR**.
- Ahora se asigna un prefijo dependiendo de la necesidad de máquinas de una determinada organización, sin necesidad de que el prefijo sea sólo uno de entre /8, /16 o /32.
- Ejemplo: a una organización se le puede asignar el prefijo: 130.0.0.0/22 (antes sería parte de una clase B). Este prefijo define 10 bits para la parte de máquina: $\simeq 2^{10}$ máquinas (menos 2).

Dir. IP de máquina	Id. de red (22 bits)	Id. de máquina (10 bits)
130.0.0.1	1000 0010 . 0000 0000 . 0000 00	00 . 0000 0001
130.0.0.2	1000 0010 . 0000 0000 . 0000 00	00 . 0000 0010
...
130.0.3.254	1000 0010 . 0000 0000 . 0000 00	11 . 1111 1110

Subdivisión del identificador de máquina: Subredes

- En ocasiones el identificador de máquina resulta un poco largo, lo que da lugar a demasiadas máquinas dentro de la misma red.
- Ejemplo: con un prefijo /8 son posibles $\simeq 2^{24}$ máquinas para esa red.
- Un mensaje de solicitud de ARP lo reciben todas esas máquinas.



Subredes

- En vez de tener todas las máquinas como vecinas, pueden repartirse en distintas **subredes**.
- El identificador de máquina se divide para separar las máquinas en diferentes subredes, obteniendo:
 - identificador de subred**
 - nuevo identificador de máquina**

Identificador de red	Id. de máquina	
Id. de red	Id. de subred	Nuevo Id. de máquina

- Cuando hay subredes la máscara cubre el nuevo id. de red. En estos casos a veces recibe el nombre de **máscara de subred**, pero en general se usan indistintamente los términos “máscara de red” y “máscara de subred”.
- Ejemplo: la dirección de red 100.0.0.0/8 quiere dividirse en 256 subredes, por tanto se necesitan 8 bits para identificar esas subredes, quedando 24 bits para identificar las máquinas de cada una de las subredes.

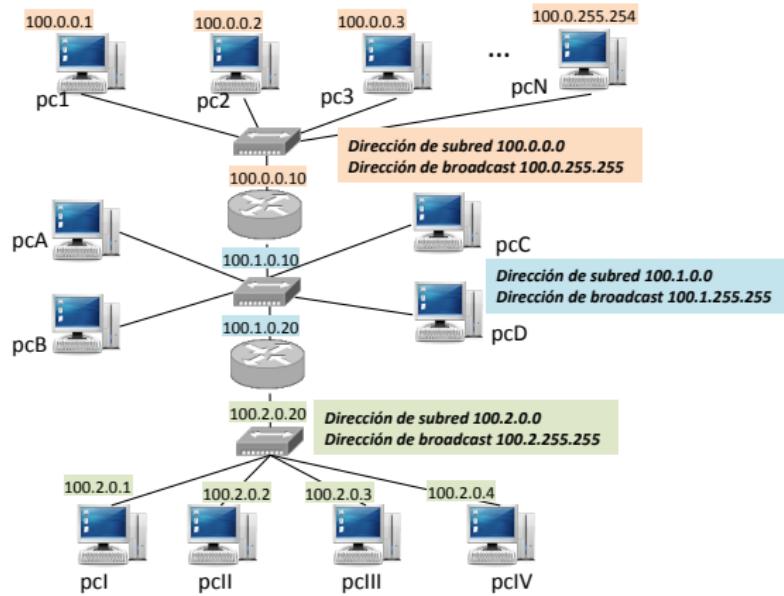
	Nuevo id. de red	Nuevo Id. de máquina	Máscara
	Id. de red	Id. de subred	
Subred 0	100	0	0.0
Subred 1	100	1	0.0
Subred 2	100	2	0.0
...
Subred 255	100	255	0.0

Subredes: ejemplo

Con $100.0.0.0/8$ se pueden planificar diferentes escenarios.

256 subredes con $\simeq 2^{16}$ máquinas cada una

Una red con todas las máquinas
 $(\simeq 2^{24}$ máquinas)



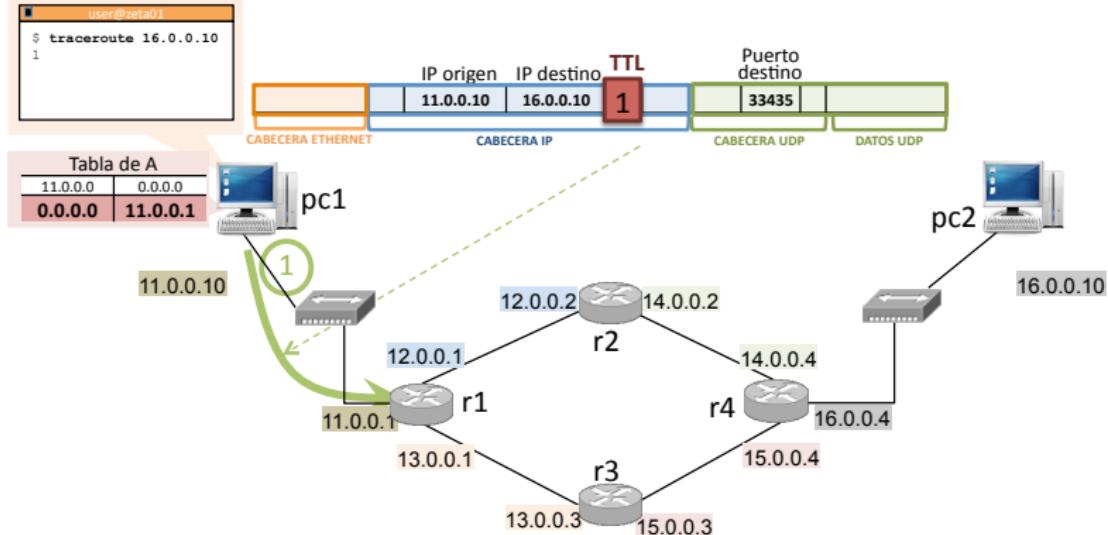
Contenidos

- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

Ejemplo: *traceroute*

- Muestra una aproximación de la ruta que siguen los datagramas IP hasta un destino.
- El nodo origen envía paquetes con TTL 1, 2, 3... (3 paquetes con cada TTL)
- El *router* que descarta el paquete envía al origen un ICMP tipo 11, código 0, lo que permite a la máquina que ejecuta el *traceroute* identificar la IP (y en su caso el nombre) de dicho *router*.

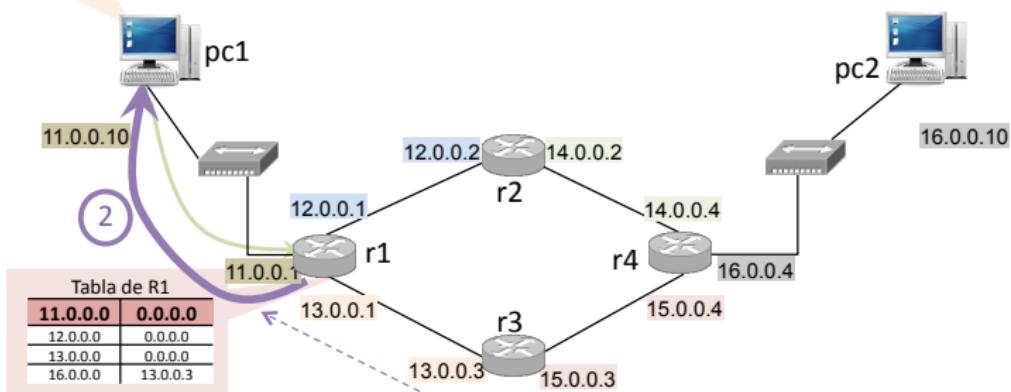
```
$ traceroute gsync.es
traceroute to gsync.es (193.147.71.64), 30 hops max, 40 byte packets
1 * * *
2 * * *
3 rediris-2.espanix.net (193.149.1.154)  10.971 ms  22.736 ms  7.346 ms
4 ESP.S01-0-0.EB-IRIS2.red.rediris.es (130.206.240.125)  8.363 ms  8.862 ms  24.718 ms
5 S00-0-0.EB-IRIS4.red.rediris.es (130.206.240.2)  13.994 ms  28.781 ms  8.976 ms
6 NAC4.S03-0-0.EB-Madrid0.red.rediris.es (130.206.240.130)  10.270 ms  9.866 ms  9.774 ms
7 cam-router.red.rediris.es (130.206.206.62)  17.183 ms  9.764 ms  16.176 ms
8 * * *
9 gsync.es (193.147.71.64)  22.846 ms  10.587 ms  14.046 ms
```



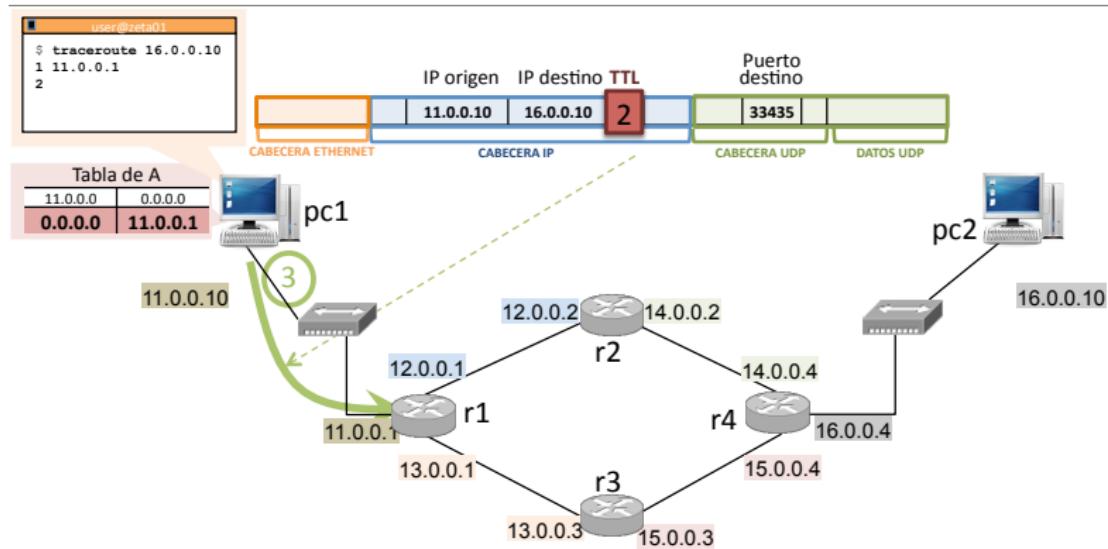
En **pc1** se ejecuta: `traceroute 16.0.0.10`:

- 1 **pc1** envía un datagrama a **pc2** con TTL 1 (*traceroute* envía por defecto 3 datagramas, pero lo veremos con 1 por simplicidad).

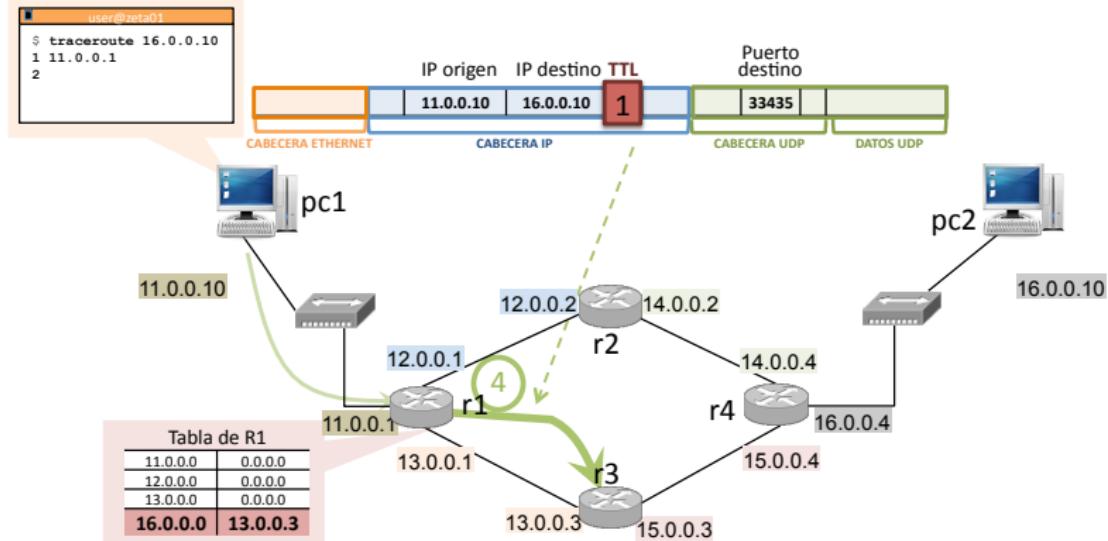
```
user@zeta01:
$ traceroute 16.0.0.10
1 11.0.0.1
```



- ② r1 envía un ICMP de TTL excedido al origen del datagrama (tipo 11, código 0), y pc1 muestra la dirección IP de origen de ese ICMP (r1) como primer salto en el camino pc1 → pc2.

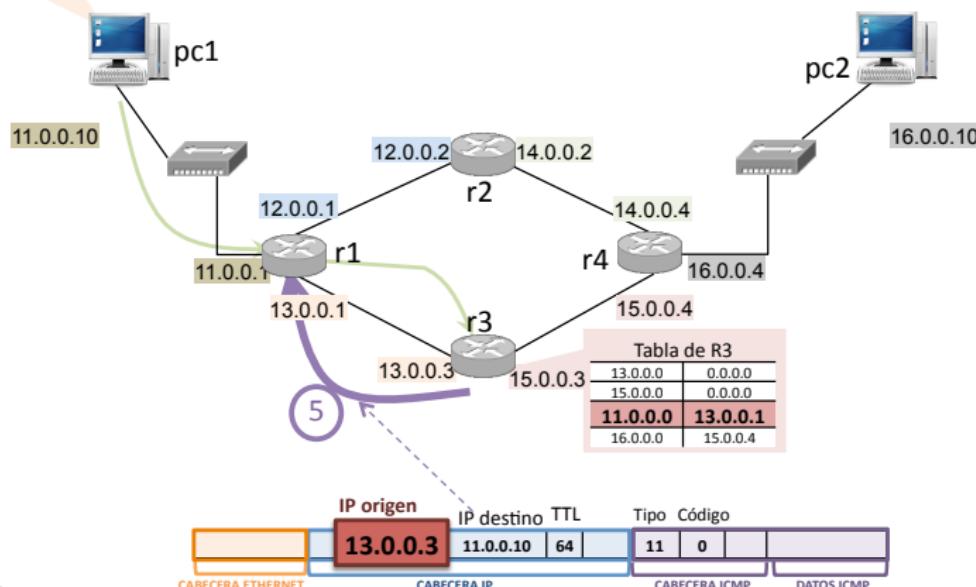


- ③ pc1 envía ahora un datagrama al destino (pc2) con TTL 2.

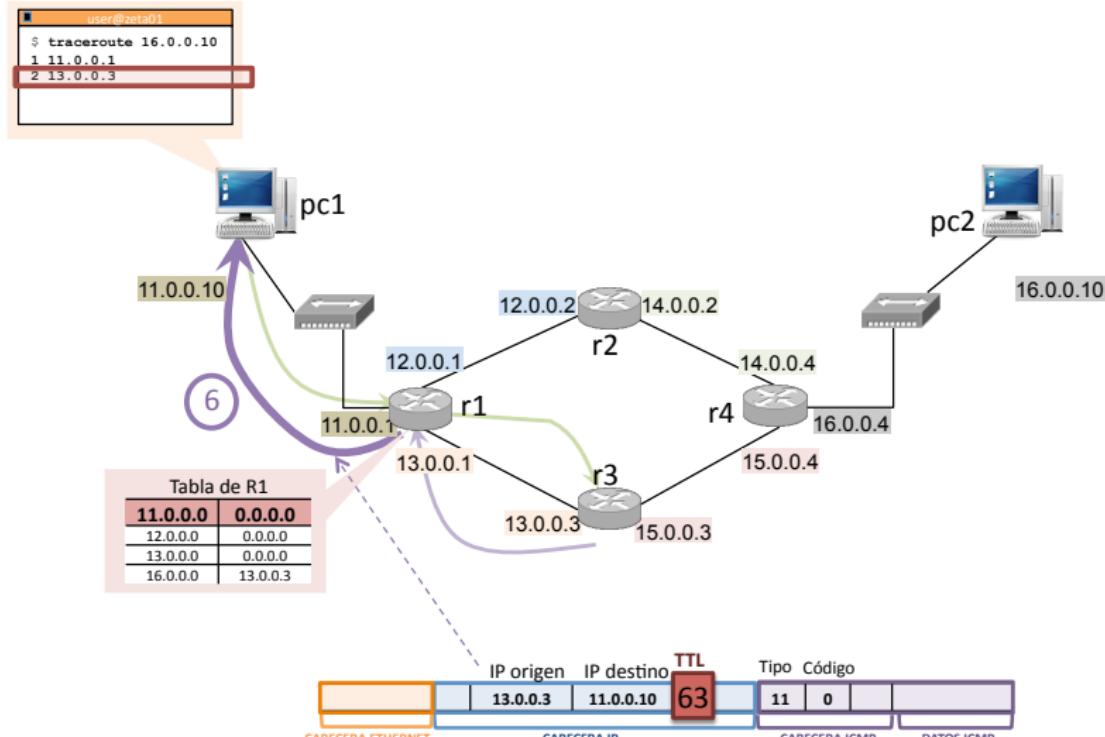


- ④ r1 disminuye en una unidad el TTL y lo reenvía hacia pc2 según su tabla de encaminamiento.

```
user@zeta01
$ traceroute 16.0.0.10
1 11.0.0.1
2
```



- 5) r3 envía un ICMP de TTL excedido al origen del datagrama, por la ruta que le indica su tabla de enrutamiento.



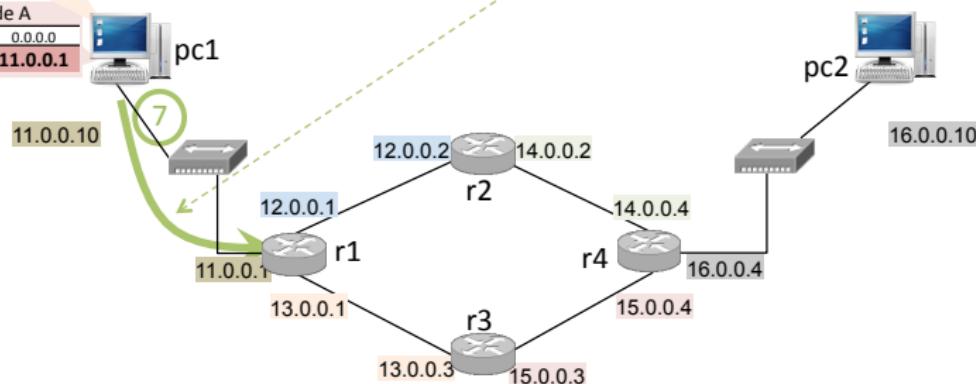
- 6 r1 reenvía el ICMP hacia pc1, tras disminuir en una unidad el TTL. pc1 muestra la dirección IP origen del ICMP recibido (r3) como segundo salto en el camino pc1 → pc2.

```
user@zeta01
$ traceroute 16.0.0.10
1 11.0.0.1
2 13.0.0.3
3
```

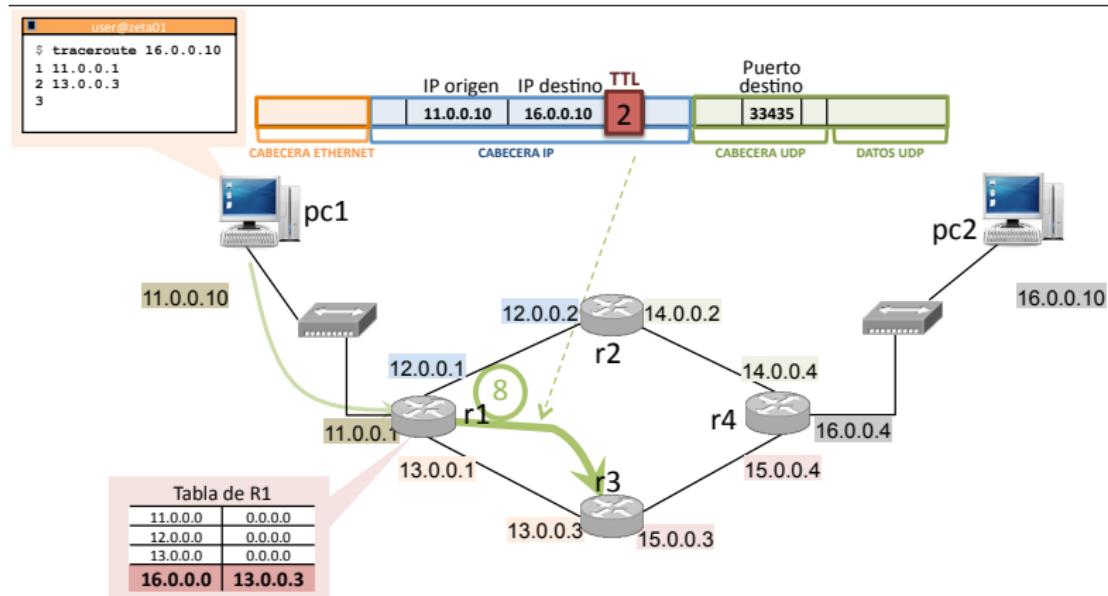


Tabla de A

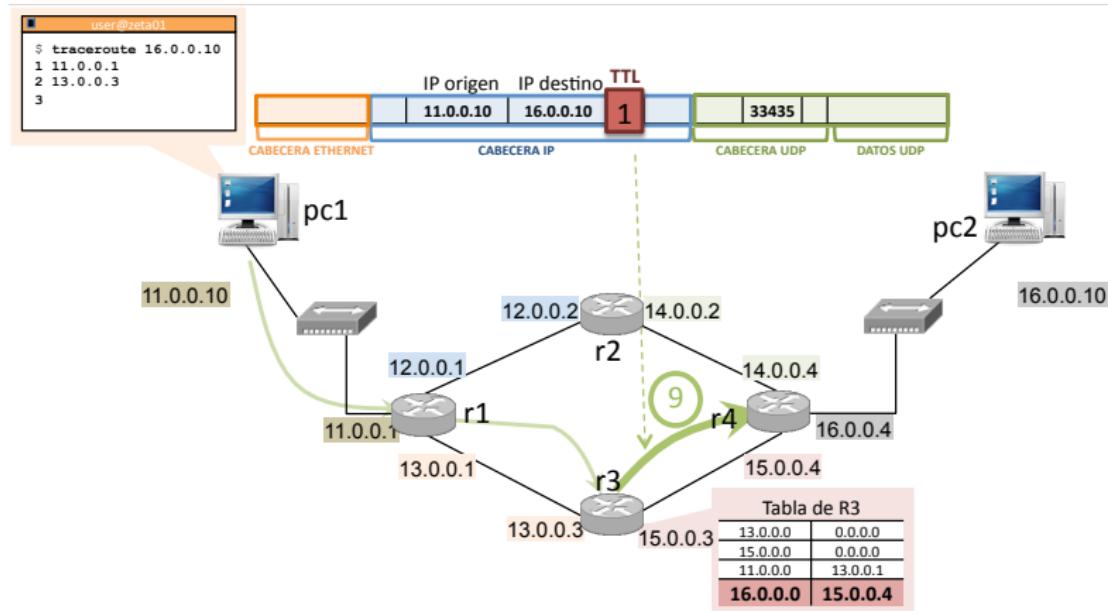
11.0.0.0	0.0.0.0
0.0.0.0	11.0.0.1



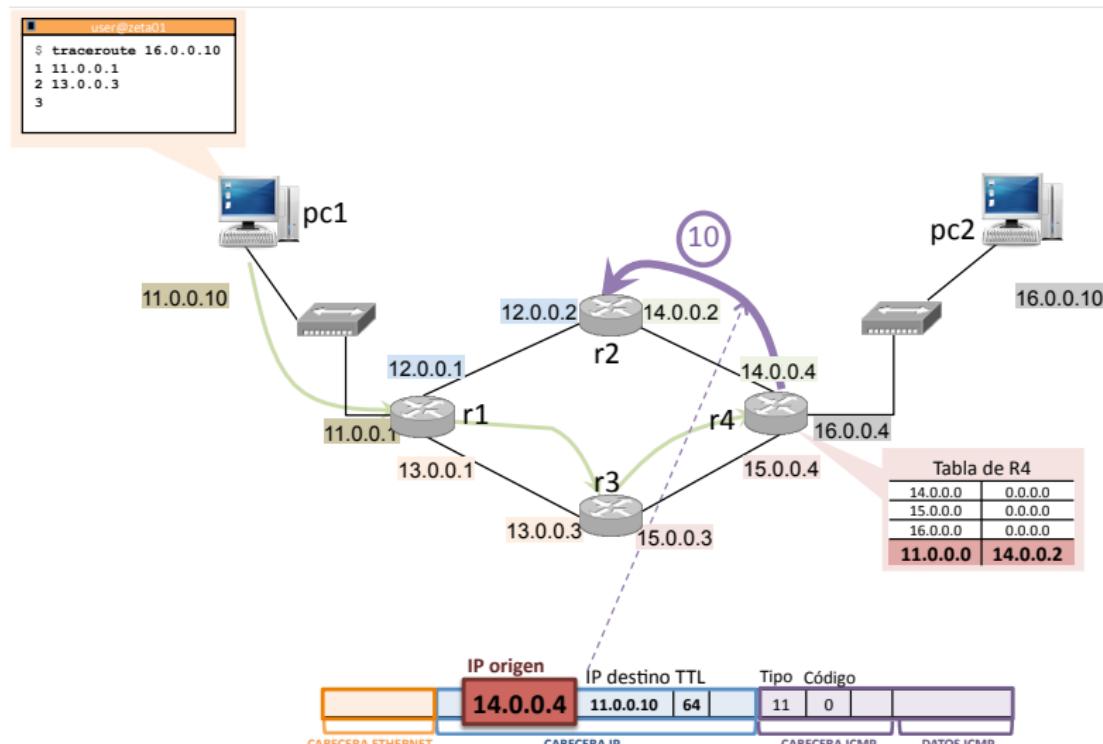
- 7 pc1 envía ahora un datagrama al destino (pc2) con TTL 3.



- 8 r1 disminuye en una unidad el TTL y lo reenvía hacia pc2 según su tabla de encaminamiento.



- 9 r3 disminuye en una unidad el TTL y lo reenvía hacia pc2 según su tabla de encaminamiento.



- 10 r4 envía un ICMP de TTL excedido al origen del datagrama, por la ruta que le indica su tabla de encaminamiento (por r2!).

```
user@zeta01
$ traceroute 16.0.0.10
1 11.0.0.1
2 13.0.0.3
3
```

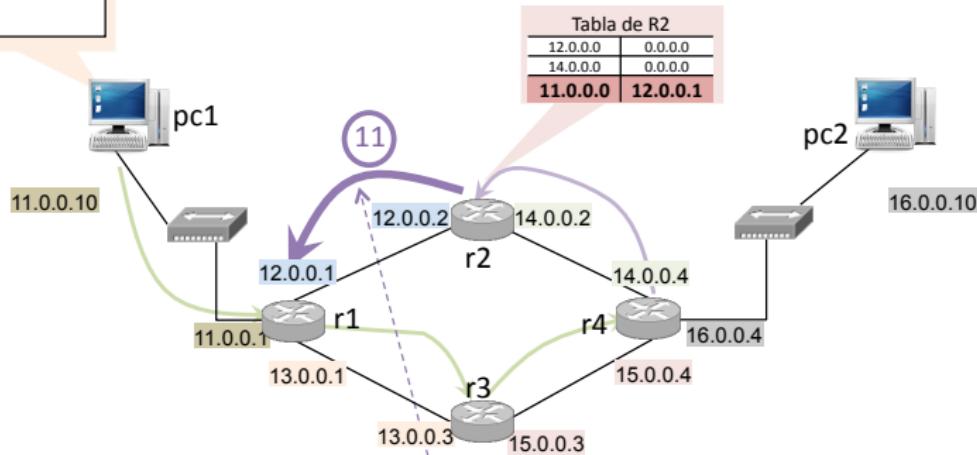
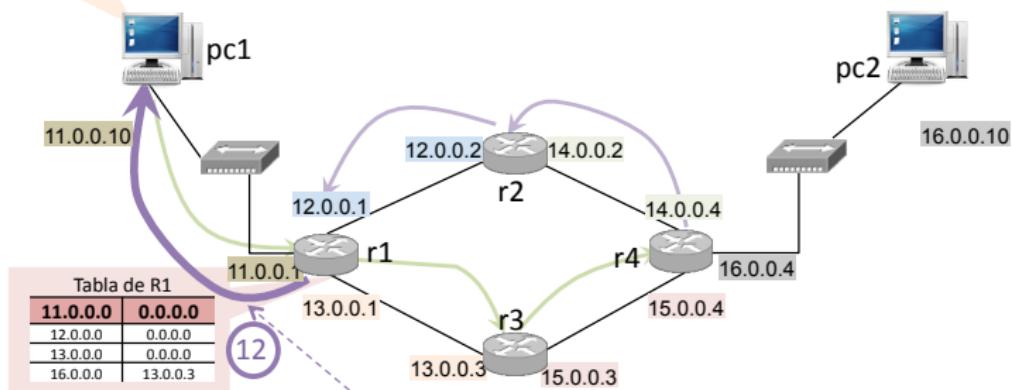


Tabla de R2	
12.0.0.0	0.0.0.0
14.0.0.0	0.0.0.0
11.0.0.0	12.0.0.1

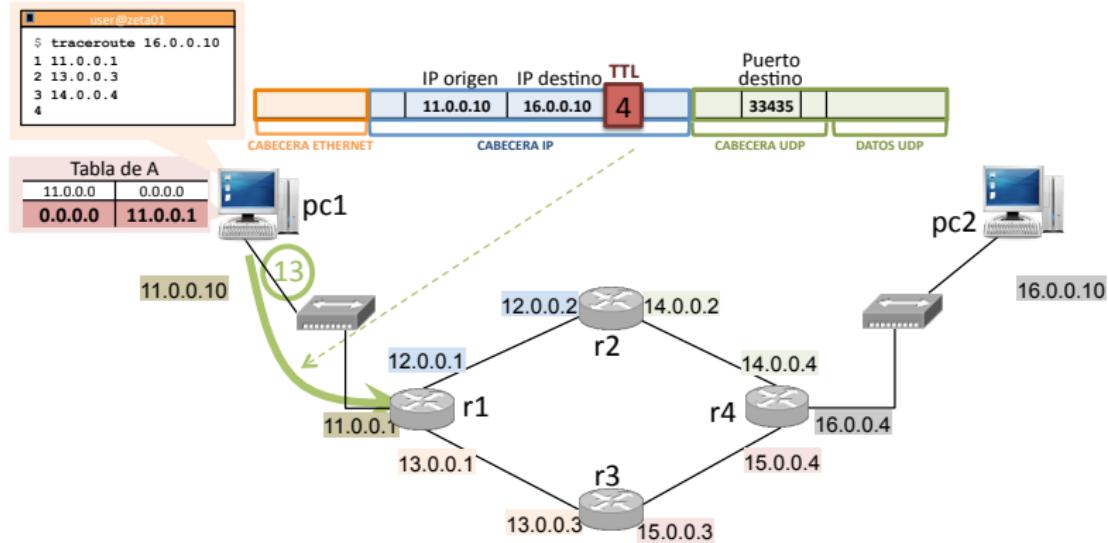


- 11 **r2** reenvía el ICMP hacia **pc1**, tras disminuir en una unidad el TTL, por la ruta que le indica su tabla de encaminamiento.

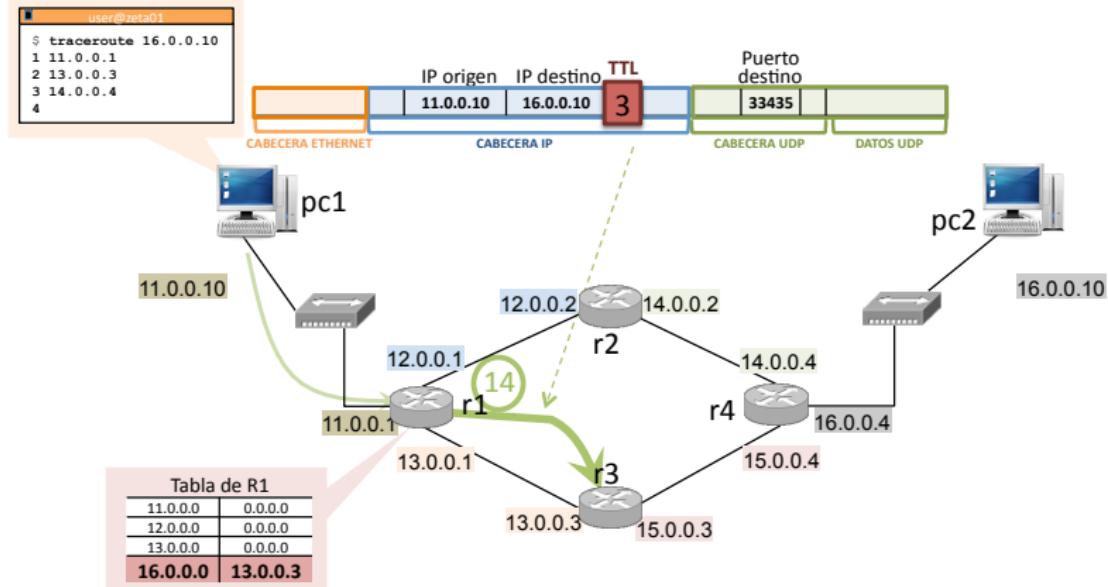
```
user@zeta01
$ traceroute 16.0.0.10
1 11.0.0.1
2 13.0.0.3
3 14.0.0.4
```



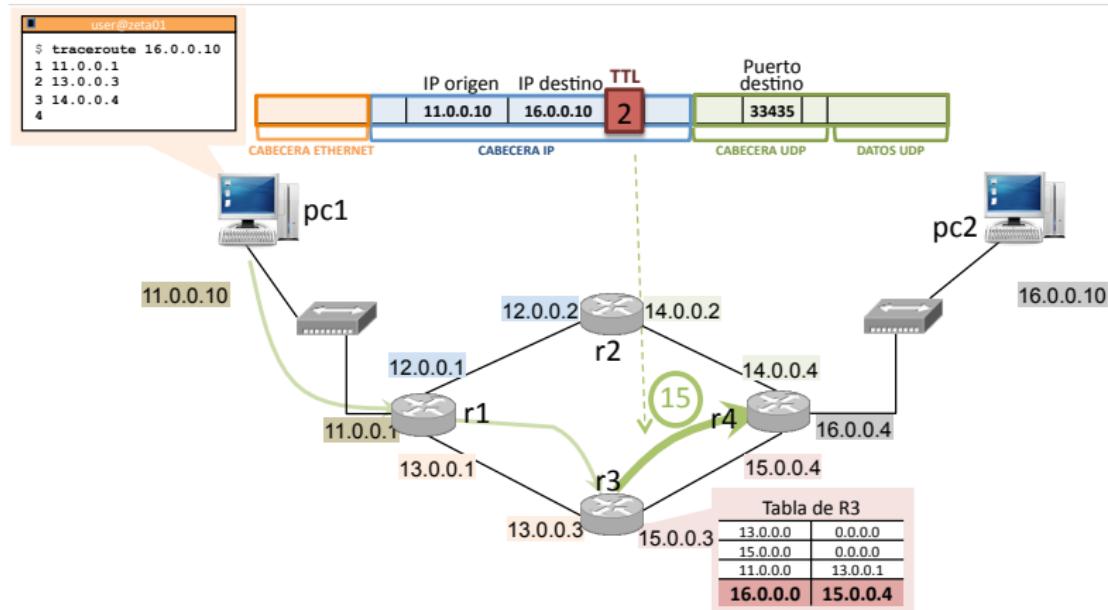
- 12 **r1** reenvía el ICMP, tras disminuir en una unidad el TTL. **pc1** muestra la dirección IP origen del ICMP recibido (**r4**) como tercer salto en el camino **pc1 → pc2**.



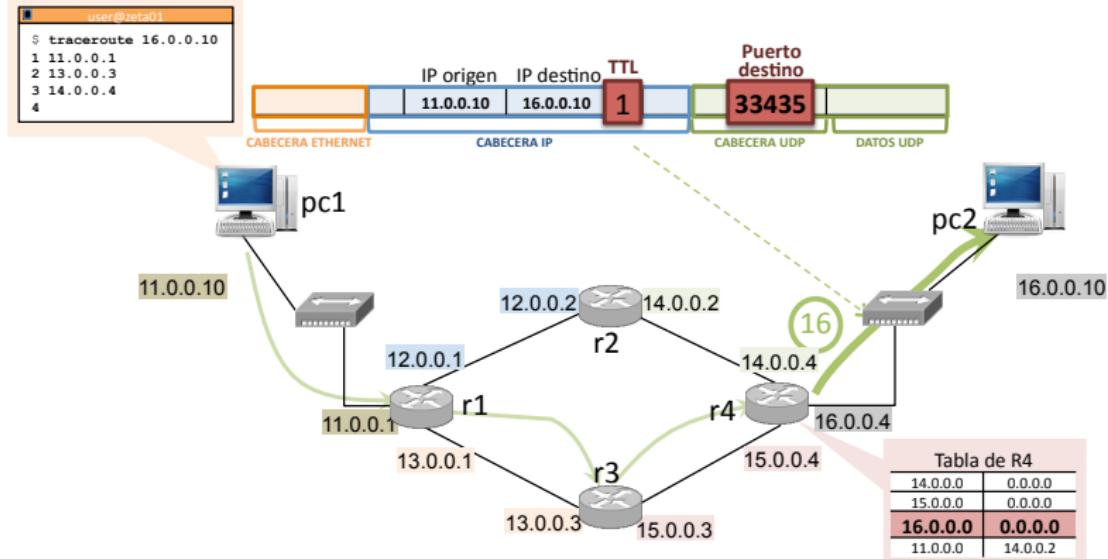
- 13 pc1 envía ahora un datagrama al destino (pc2) con TTL 4.



- 14 **r1** disminuye en una unidad el TTL y lo reenvía hacia **pc2** según su tabla de encaminamiento.

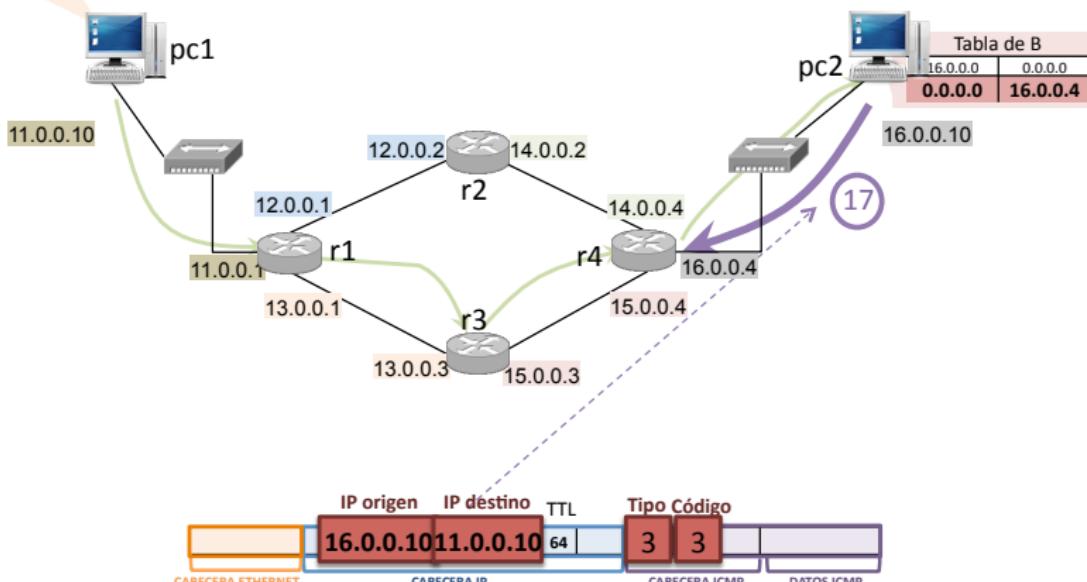


15 r3 disminuye en una unidad el TTL y lo reenvía hacia pc2 según su tabla de encaminamiento.

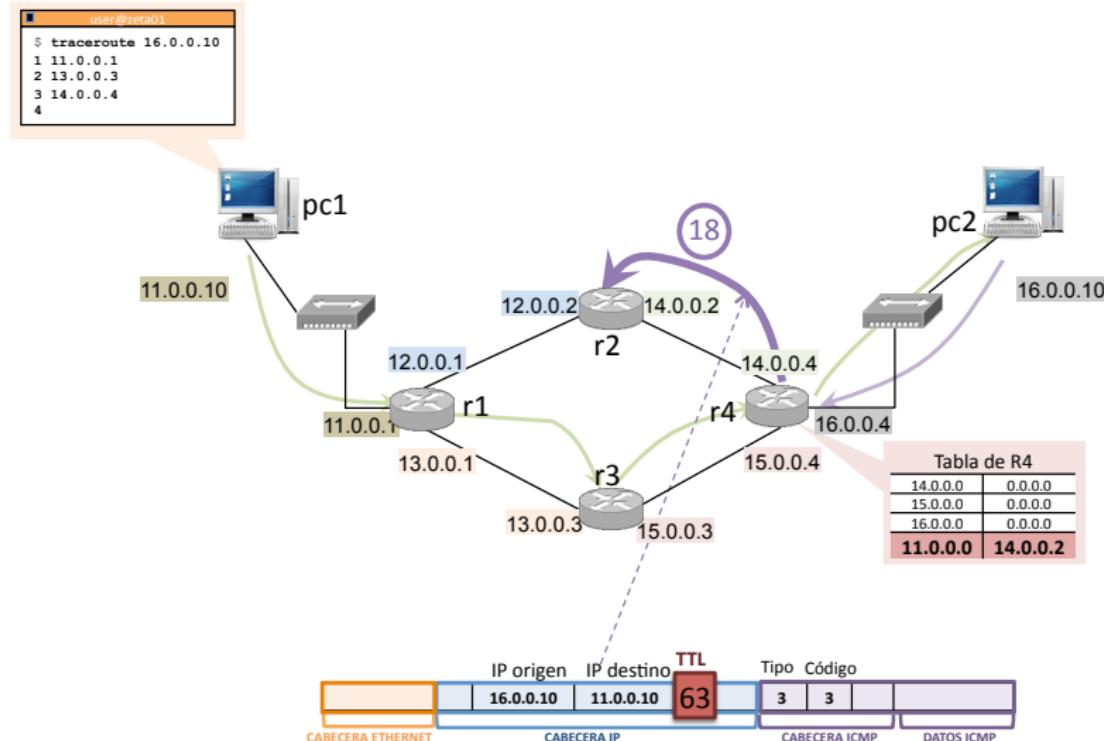


- 16 r4 disminuye en una unidad el TTL y lo reenvía hacia pc2 según su tabla de encaminamiento.

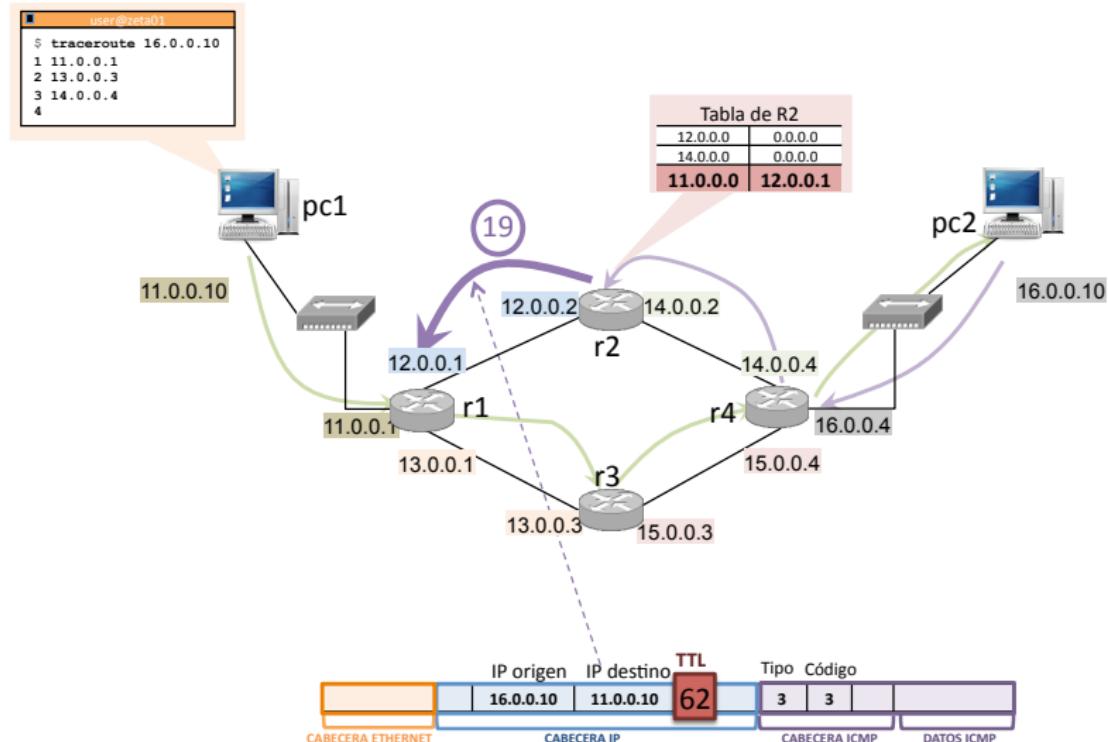
```
user@zeta01
$ traceroute 16.0.0.10
1 11.0.0.1
2 13.0.0.3
3 14.0.0.4
4
```



17 pc2 recibe el datagrama. Envía un ICMP de puerto inalcanzable al origen del datagrama.

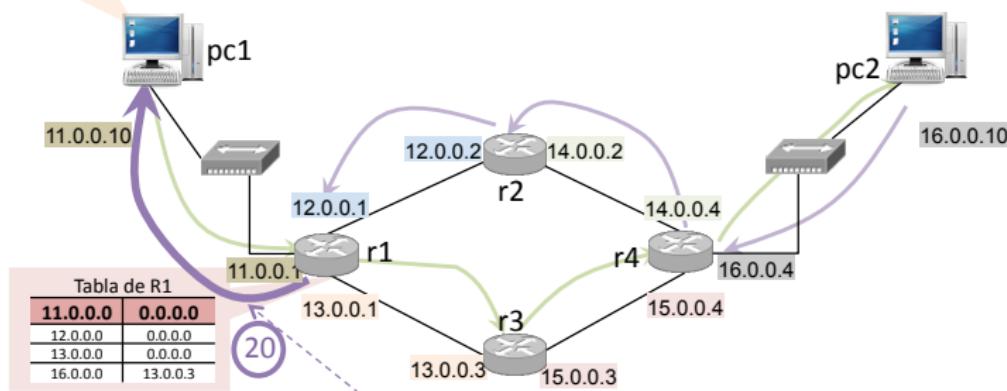


- 18 r4 reenvía el ICMP hacia pc1, tras disminuir en una unidad el TTL, por la ruta que le indica su tabla.



- 19 r2 reenvía el ICMP hacia pc1, tras disminuir en una unidad el TTL, por la ruta que le indica su tabla.

```
user@zeta01
$ traceroute 16.0.0.10
1 11.0.0.1
2 13.0.0.3
3 14.0.0.4
4 16.0.0.10
```



- 20 r1 reenvía el ICMP hacia pc1, tras disminuir en una unidad el TTL. pc1 muestra la dirección IP origen del ICMP recibido (pc2) y termina al ser el destino.

Resumen del funcionamiento de traceroute

- Por defecto traceroute envía 3 datagramas IP al destino con TTL=1, cuando reciba respuesta (o pasado un determinado tiempo, 5 segundos) enviará nuevamente 3 datagramas IP al destino con TTL=2, y así sucesivamente. En las figuras anteriores hemos mostrado el comportamiento enviando 1 solo datagrama con TTL=1, 1 con TTL=2... Este comportamiento se consigue ejecutando traceroute en la forma `traceroute -q1 <destino>`
- Cada *router* intermedio disminuye en una unidad el valor del campo TTL. Si TTL llega a cero, el *router* intermedio deberá enviar un mensaje ICMP encapsulado en un datagrama IP que indique que el TTL se ha excedido y que se ha descartado el datagrama inicial.
 - Si existe ruta para hacer llegar el mensaje ICMP a la máquina que inició el traceroute, la máquina origen podrá imprimir la dirección IP del nodo intermedio.
 - Si no existe ruta para hacer llegar el mensaje ICMP desde el nodo intermedio a la máquina que inició el traceroute, el mensaje ICMP se descartará y no llegará a la máquina origen. En este caso la máquina origen no podrá imprimir la dirección IP del nodo intermedio e imprimirá un *.
 - En cualquiera de los dos casos anteriores, se continuará el envío de datagramas IP incrementando en una unidad el valor del campo TTL.
- Cuando los datagramas IP lleguen al destino final, la máquina destino enviará mensajes ICMP indicando puerto inexistente y el origen al recibirlos terminará la ejecución de traceroute.

Contenidos

1 Protocolo IP

2 Protocolo ARP

3 Ejemplo de encaminamiento

4 El problema inverso al ARP

5 Protocolo ICMP

6 Asignación y planificación de subredes IP

7 Ejemplo: traceroute

8 Congestión

9 Referencias

Congestión en el protocolo IP

- El protocolo IP ofrece un modelo de nivel de red basado en datagramas.
- La principal fuente de pérdidas de paquetes en Internet se debe a la congestión de *routers*, que actúan descartando los paquetes que no les caben en sus *buffers*.
- IP ofrece un servicio no fiable: no se recupera de las pérdidas por congestión. Lo harán, en todo caso, protocolos de niveles superiores (en particular, TCP).
- IP no toma medidas especiales para prevenir o disminuir la congestión: será, en todo caso, labor de protocolos de niveles superiores (en particular, TCP).

Contenidos

- 1 Protocolo IP
- 2 Protocolo ARP
- 3 Ejemplo de encaminamiento
- 4 El problema inverso al ARP
- 5 Protocolo ICMP
- 6 Asignación y planificación de subredes IP
- 7 Ejemplo: traceroute
- 8 Congestión
- 9 Referencias

Referencias

- W. R. Stevens, **TCP/IP Illustrated, Vol. 1**: Cap. 3, Cap. 4, Cap. 6, Cap 9.
- A. Tanembaum, **Redes de Computadores (4^a ed.)**: Cap. 5 (5.6.1, 5.6.2, 5.6.3).
- J. F. Kurose, K. W. Ross, **Computer Networking: A Top-Down Approach (4th ed)**: Cap. 4 (4.4).