

NATs: Recomendaciones

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Grado en Ingeniería en Tecnologías de
Telecomunicación, 3º

NATs y UDP: Mapeos

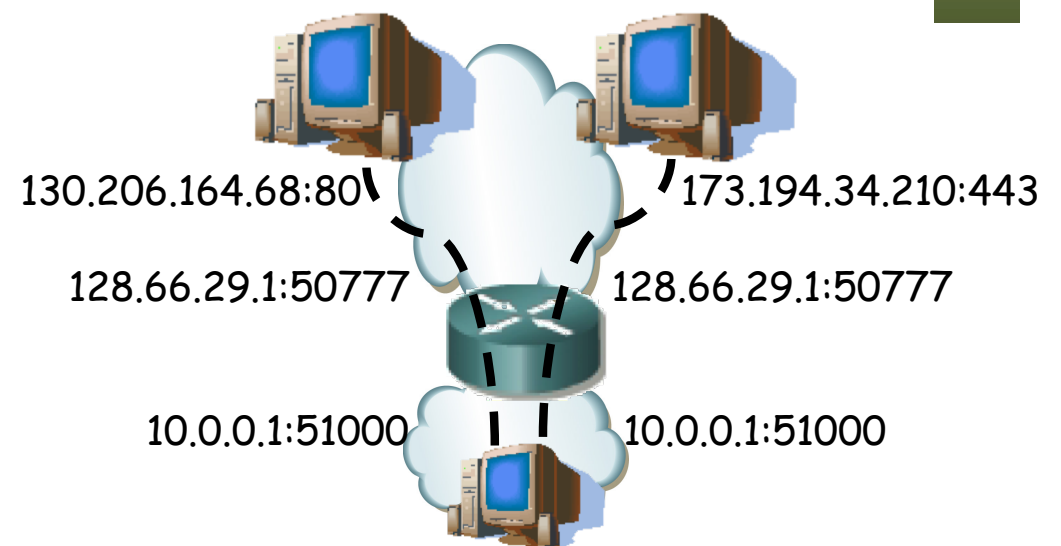
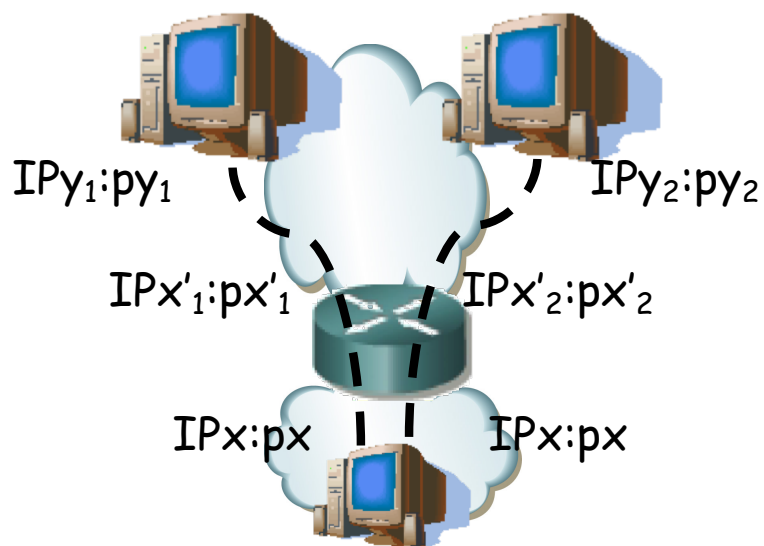
REQs UDP: RFC 4787

- A.k.a. BCP 127 (Enero 2007)
- “Network Address Translation (NAT) Behavioral Requirements for Unicast UDP”
- Contiene Requerimientos y Recomendaciones
- No quiere decir eso que todas las implementaciones las sigan (¡ni mucho menos!)
- Veamos algunas de ellas (...)



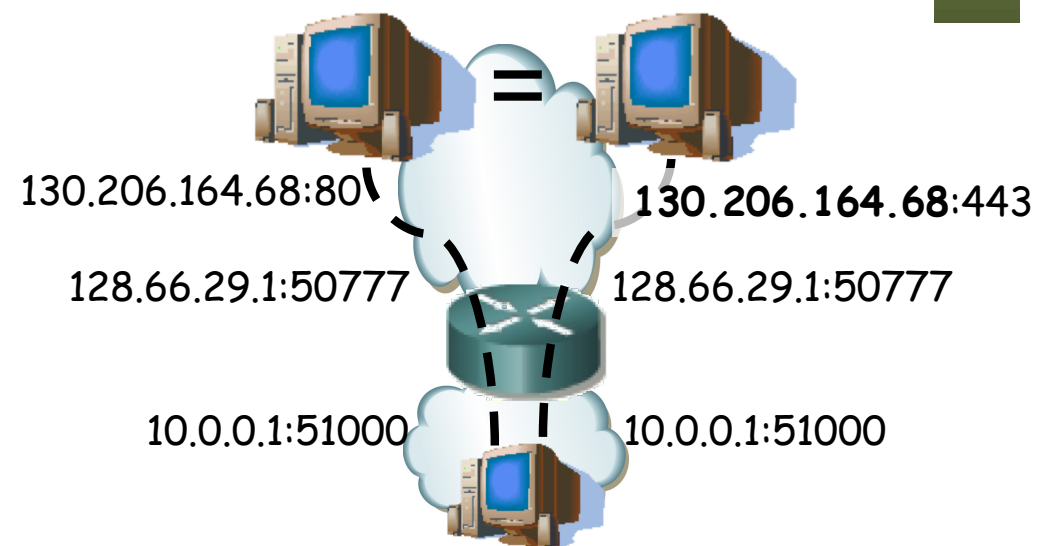
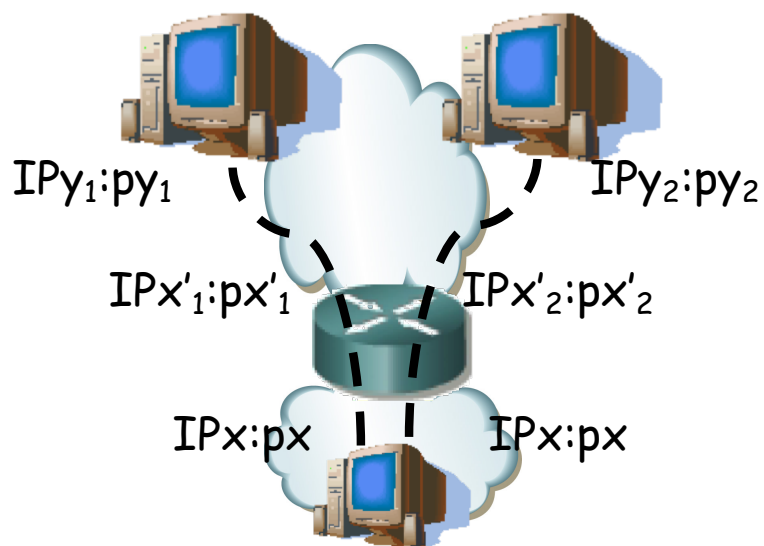
Mapeo de dirección y puerto

- Mapeo establecido: $(IPx:px, IPx'_1:px'_1) \rightarrow IPy_1:py_1$
- A continuación se quiere enviar de $IPx:px$ a $IPy_2:py_2$
- *Endpoint-Independent Mapping*
 - Reutiliza el mapeo establecido con esa dirección y puerto internos a cualquier dirección IP y puerto externos
 - Es decir, $IPx'_1:px'_1 = IPx'_2:px'_2$ para cualquier valor de $IPy'_2:py'_2$
 - (...)



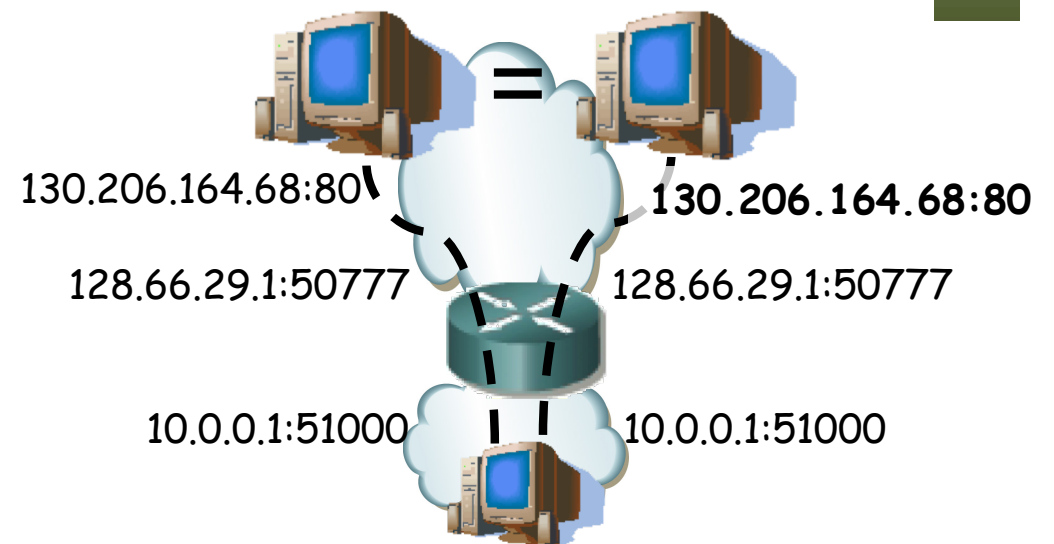
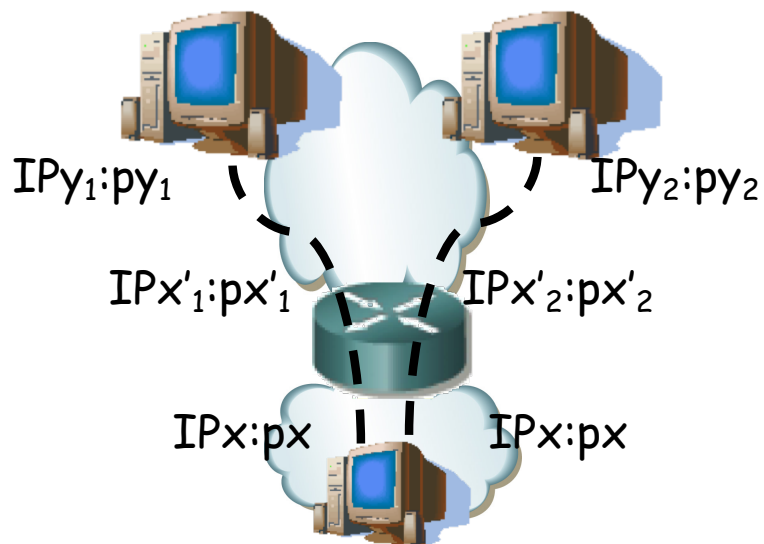
Mapeo de dirección y puerto

- Mapeo establecido: $(IPx:px, IPx'_1:px'_1) \rightarrow IPy_1:py_1$
- A continuación se quiere enviar de $IPx:px$ a $IPy_2:py_2$
- *Endpoint-Independent Mapping*
- *Address-Dependent Mapping*
 - Reutiliza el mapeo establecido con esa dirección y puerto internos siempre que vayan a la misma dirección externa
 - Es decir, $IPx'_1:px'_1 = IPx'_2:px'_2$ si y solo si $IPy'_2 = IPy'_1$
 - (...)



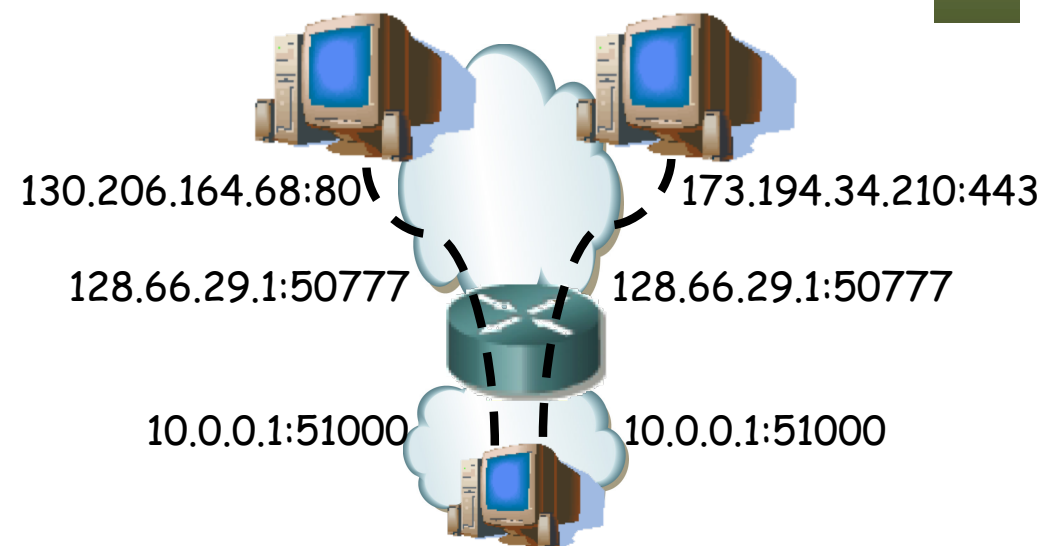
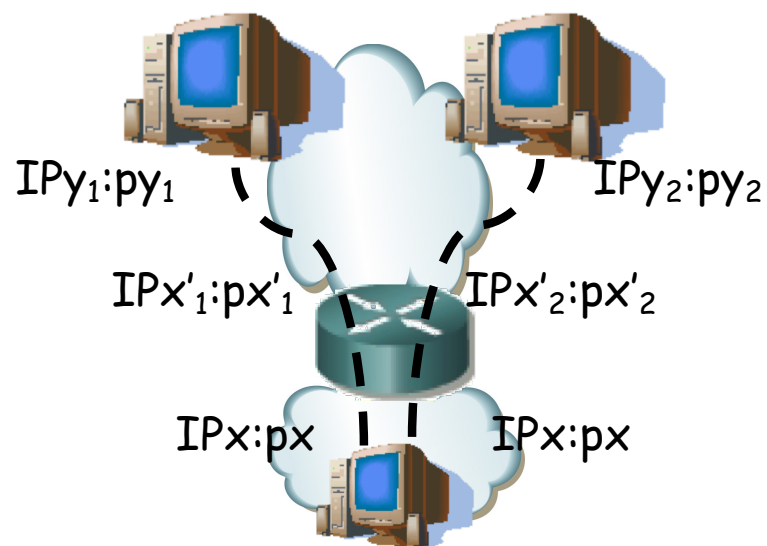
Mapeo de dirección y puerto

- Mapeo establecido: $(IPx:px, IPx'_1:px'_1) \rightarrow IPy_1:py_1$
- A continuación se quiere enviar de $IPx:px$ a $IPy_2:py_2$
- *Endpoint-Independent Mapping*
- *Address-Dependent Mapping*
- *Address and Port-Dependent Mapping*
 - Reutiliza el mapeo establecido con esa dirección y puerto internos siempre que vayan a la misma dirección y puerto externos
 - Es decir, $IPx'_1:px'_1 = IPx'_2:px'_2$ si y solo si $IPy'_2:py_2 = IPy'_1:py_1$



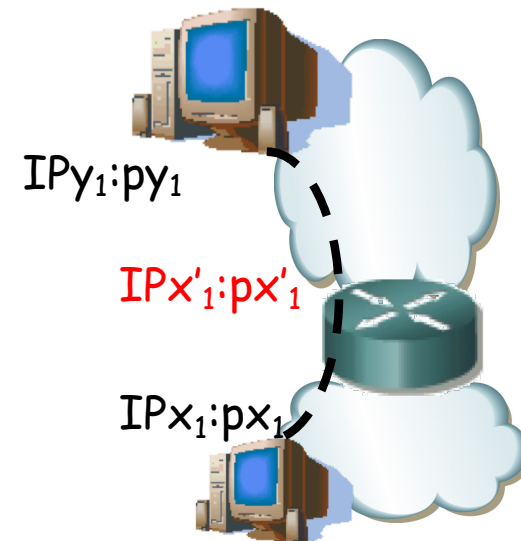
REQ: Endpoint-Independent

- BCP 127 **requiere** comportamiento *Endpoint-Independent*
- Necesario para métodos UNSAF (...)



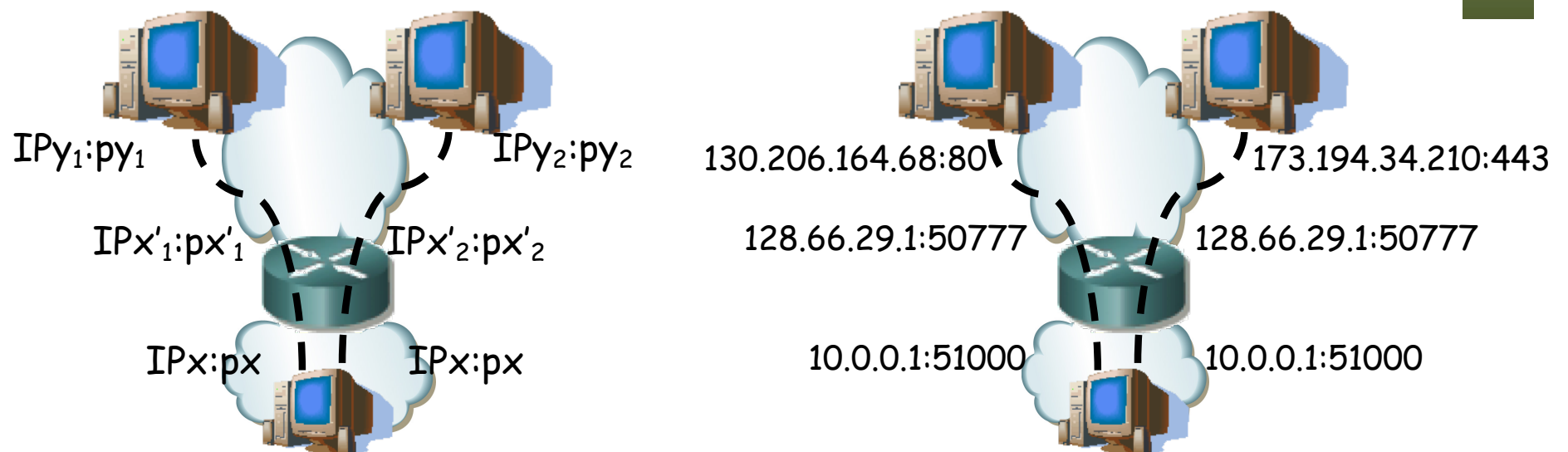
UNSAF

- UNilateral Self-Address Fixing (UNSAF)
- En ocasiones la aplicación necesita saber con qué dirección IP y puerto externo está comunicándose con el otro extremo
- Son métodos para que un extremo averigüe o fije la dirección y puerto con la que le puede acceder otro extremo
- Es decir, saber los valores de dirección y puerto públicos que va a emplear el NAT en la comunicación, tal vez para comunicarla en los datos del protocolo
- Se basan en heurísticos



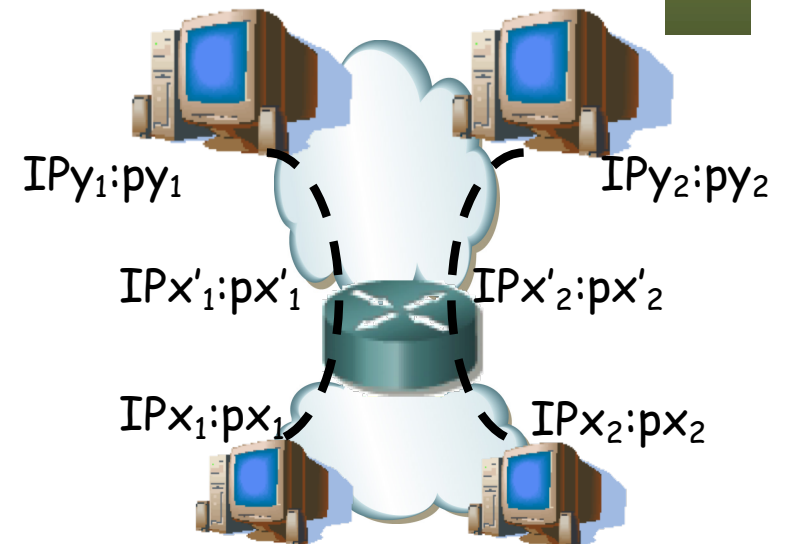
REQ: Emparejamiento de dir.

- Algunos NATs escogen la dirección pública de un *pool*
- Si lo hacen al azar para cada mapeo no mantienen la identidad (dirección IP) del host externamente
- Suele hacerse para ocultar a los usuarios
- Da problemas con algunas aplicaciones (ej: algunas sobre RTP)
- BCP 127: Requiere que la dirección externa en todos los mapeos de una dirección interna se mantenga



Mapeo de puerto

- *Port preservation*: NAT al mapear puerto: $px_1=px'_1$, $px_2=px'_2$
- Técnicas que usan ante colisión:
 - Sobrescribir el mapeo anterior
 - Asignar otra dirección pública si tienen un *pool*
 - Dejar de hacer *port preservation*
 - *Port overloading (PO)*: usar el mismo puerto aunque colisionen (!)
- Puertos: 0-1023 *well-known*, 1024-49151 *registered*, 49152-65535 *dynamic/private*
- Si requiere cambiar el puerto
 - Algunos emplean solo el rango dinámico
 - Otros emplean todos menos los *well-known*
 - Otros mantienen el que sea del mismo grupo si es *well-known*
- REQ BCP 127: No hacer *port overloading*
- Si no dos hosts internos no pueden usar mismo puerto con mismo servidor:puerto
- Recomendado: mantener el puerto en el mismo rango (0-1023 y 1024-65535)



NATs y UDP: Timers

Timers

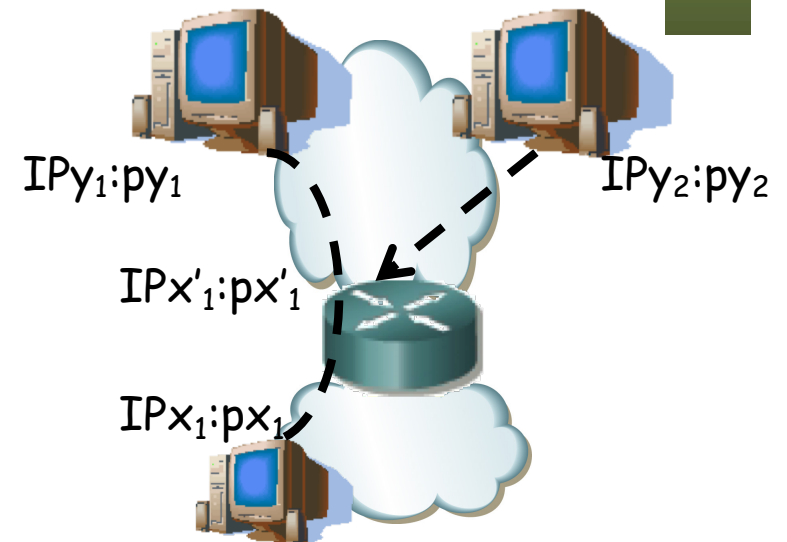
- Tiempo sin recibir paquetes de un mapeo tras el cual se elimina
 - En el rango de unos minutos (2-5min)
 - Si es de un servicio (puerto) conocido sobre UDP que se sabe emplea flujos cortos se puede reducir
-
- REQ: refrescar el timer ante paquetes salientes
 - Recomendado: refrescarlo ante paquetes entrantes (desventaja de que el externo puede mantenerlo indefinidamente)



NATs y UDP: Filtrado

Filtrado

- Mapeo establecido: $(IP_{x_1}:px_1, IP_{x'_1}:px'_1) \rightarrow IP_{y_1}:py_1$
- El filtrado determina qué hace el NAT con paquetes de $IP_{y_2}:py_2$ dirigidos a $IP_{x'_1}:px'_1$
- *Endpoint-Independent*
 - Reenvía cualquier paquete dirigido a $IP_{x_1}:px_1$ hacia $IP_{x_1}:px_1$
- *Address-Dependent*
 - Reenvía el paquete hacia $IP_{x_1}:px_1$ solo si $IP_{y_2}=IP_{y_1}$
- *Address and Port-Dependent*
 - Reenvía el paquete hacia $IP_{x_1}:px_1$ solo si $IP_{y_2}=IP_{y_1}$ y $py_2=py_1$
- Ejemplo: Una aplicación UDP (¿un juego?) podría cambiar al cliente a otro servidor
- Interesa que el nuevo servidor pueda contactar con el cliente si sabe $IP_{x'_1}:px'_1$
- Recomendado:
 - *Endpoint-Independent* para facilitar a las aplicaciones
 - *Address-Dependent* para mayor restricción



NATs y UDP: Otros aspectos

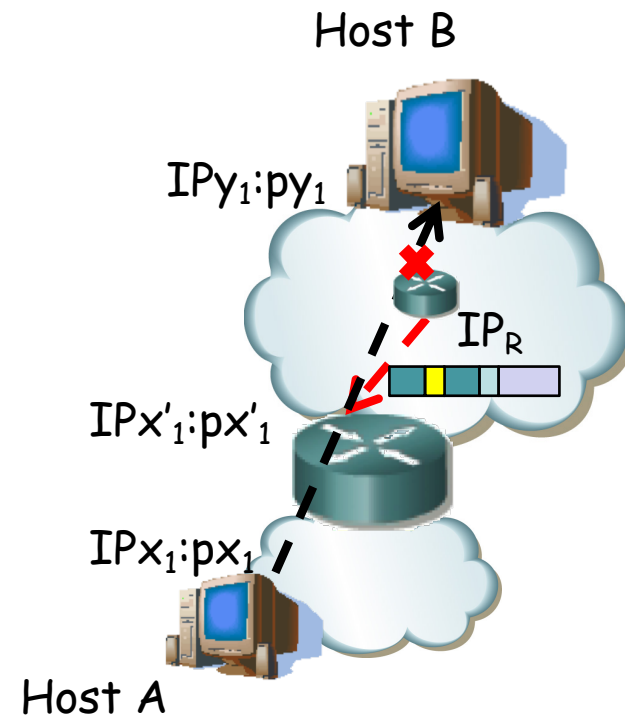
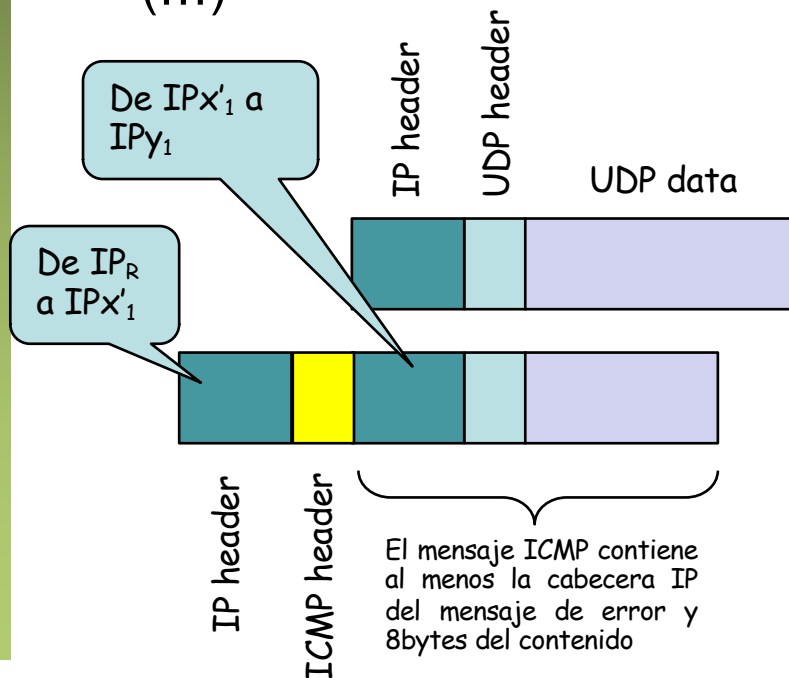
Determinismo

- Hay implementaciones de NAT que cambian de comportamiento según las circunstancias
- Por ejemplo:
 - Es *Endpoint-Independent* con *Port Overloading*
 - Pero si hay dos hosts internos en comunicación con el mismo externo lo cambia a *Address and Port-Dependent* sin *Port preservation*
- Un NAT que cambia su método de mapeo o de filtrado sin cambios de configuración se llama no determinista
- No deterministas suelen cambiar el comportamiento ante conflictos
- Se habla de su comportamiento primario, secundario, terciario, según el número de conflictos
- REQ: comportamiento determinista



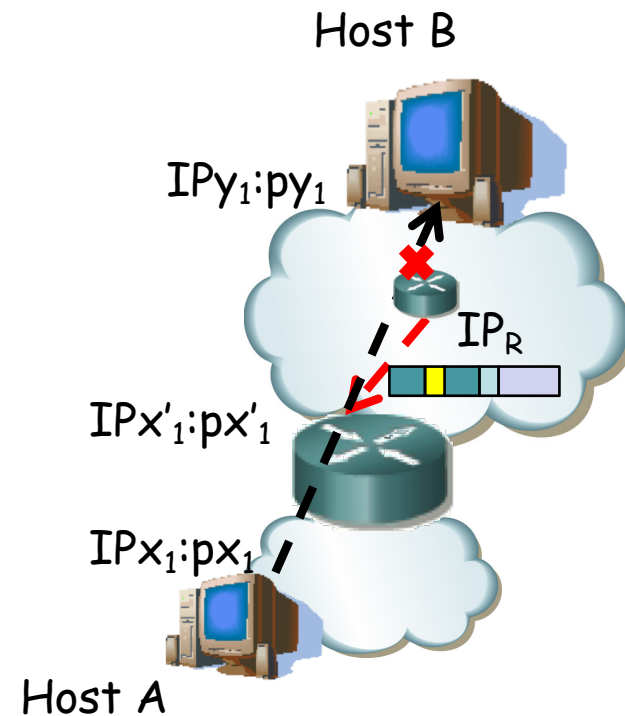
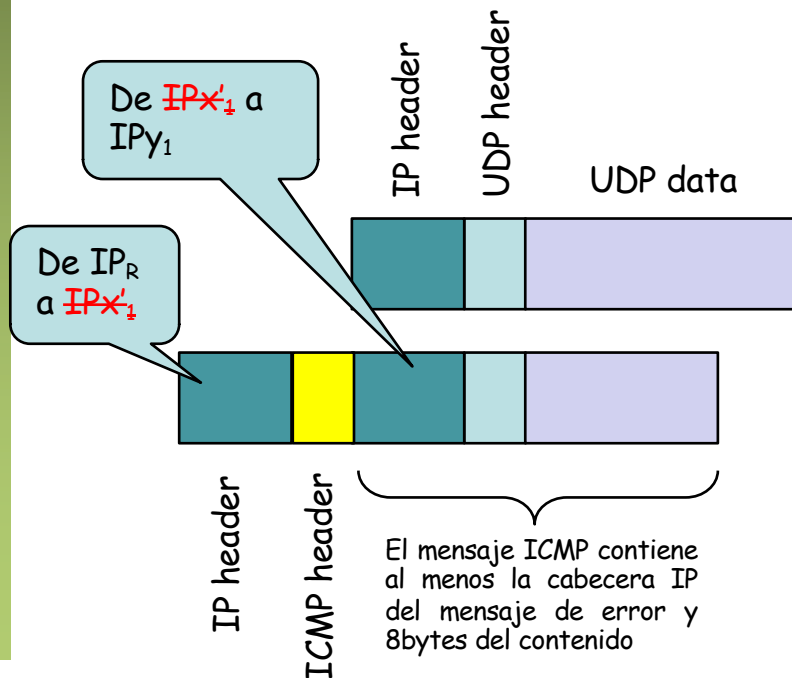
ICMP destino inalcanzable

- Puede ser devuelto por un router intermedio
- Dirección IP origen del ICMP no coincide con la del mapeo
- El mapeo era para un datagrama UDP desde host A a host B (y ciertos puertos), que se puede utilizar para datagramas en sentido contrario
- Pero no es UDP, es ICMP lo que se recibe
- No debería descartarlo
- Debería aplicarle el mapeo y reescribir los datos (el paquete contenido tiene la dirección pública, no la interna)
- (...)



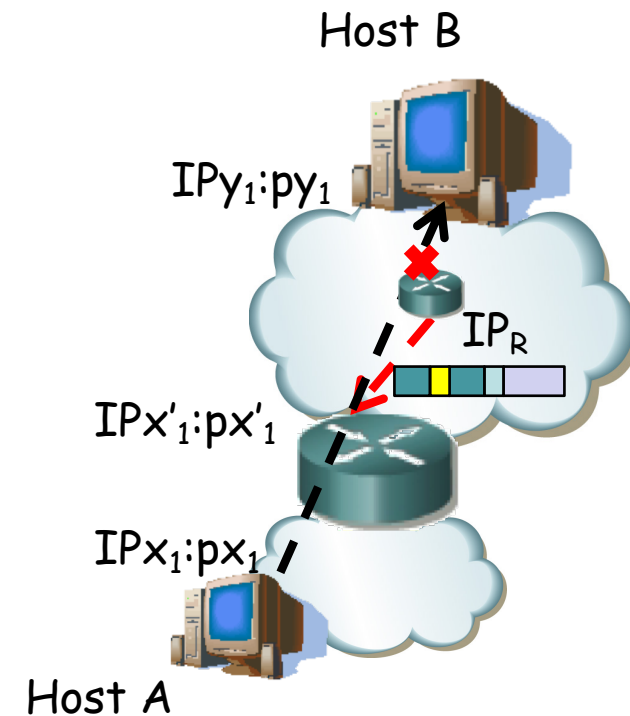
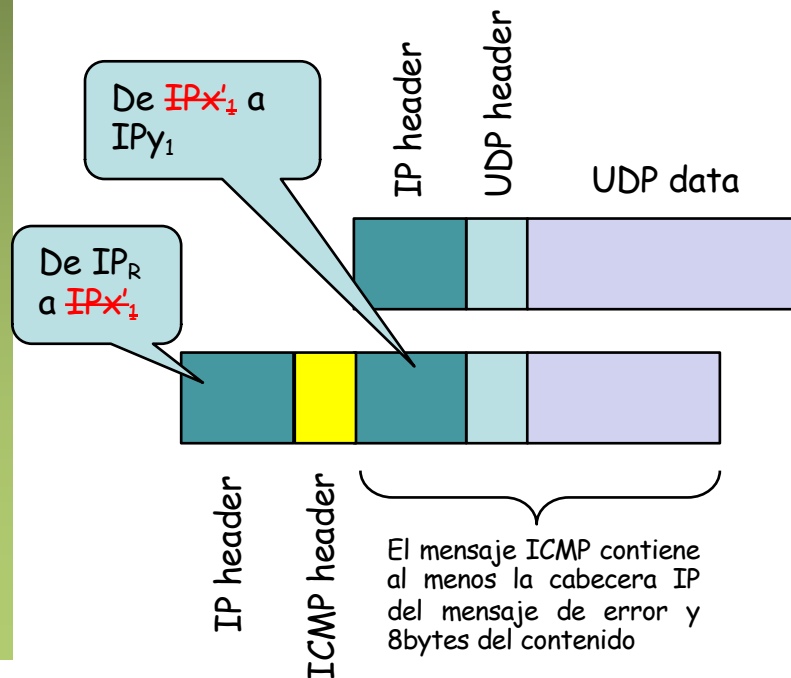
ICMP destino inalcanzable

- Es decir, en el contenido de los datos ICMP está la cabecera IP y los puertos UDP del mapeo
- Debe buscarlos y deshacer el cambio del NAT también en ellos
 - La cabecera IP externa, en lugar de ir a IPx'_1 debe ir a IPx_1
 - La cabecera IP interna, en lugar de ir **desde** IPx'_1 debe ir desde IPx_1
 - La cabecera UDP interna, debe deshacer el cambio de puerto que hubiera hecho



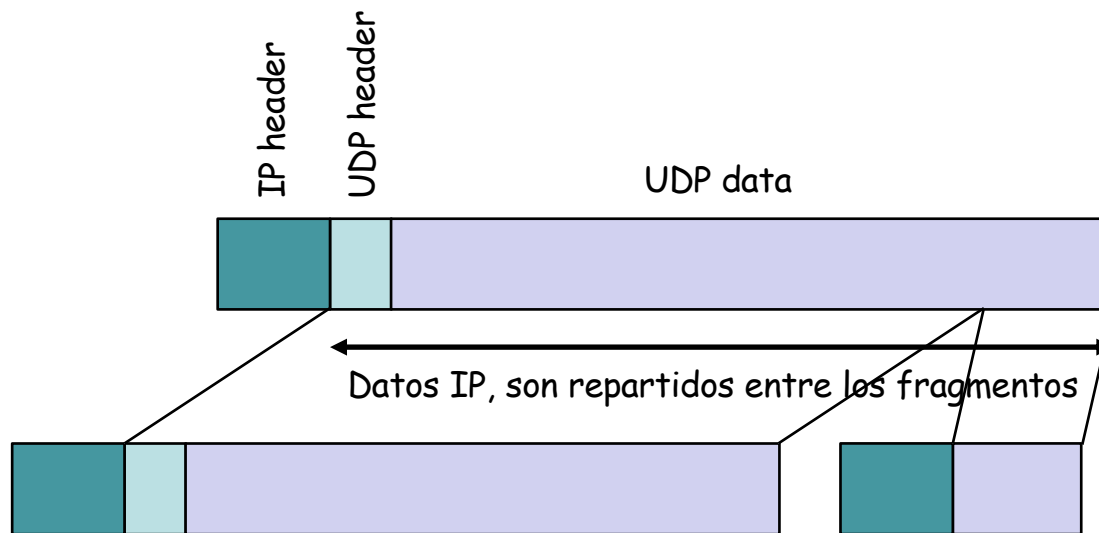
ICMP destino inalcanzable

- Debe hacerlo mientras exista el mapeo
- REQ: No debe eliminar el mapeo por recibir un ICMP
- Permite así UDP Path MTU Discovery y traceroute



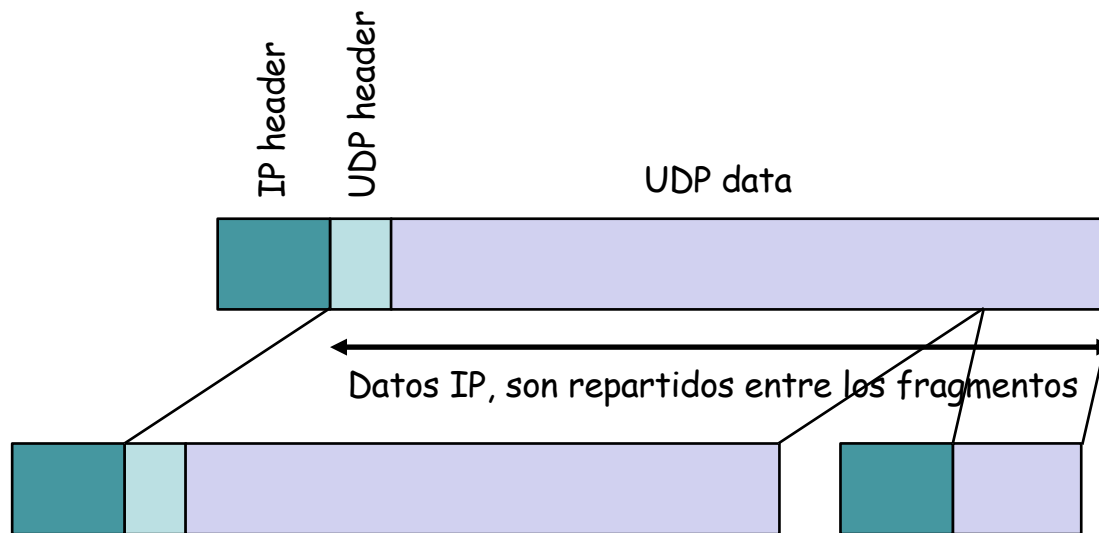
Recibiendo fragmentos

- Si recibe fragmentos del exterior la cabecera de transporte está solo en el primero (que puede no llegar el primero)
- Para ellos no se puede localizar el mapeo correspondiente al no tener la cabecera de transporte
- Al ver el primer fragmento (que contiene la cabecera de transporte) debe recordar el identificador de fragmentación
- Eso le permite deshacer el mapeo para el segundo fragmento
- ¿Y si el segundo fragmento llega primero? ¿Guardarlo hasta recibir el primero?



Recibiendo fragmentos

- NAT “*Received Fragments Ordered*”
 - Solo es capaz de reenviar correctamente los fragmentos si han llegado en orden
- NAT “*Received Fragments Out of Order*”
 - Es capaz de reenviar fragmentos correctamente aunque le lleguen desordenados
- REQ: ser *out of order*



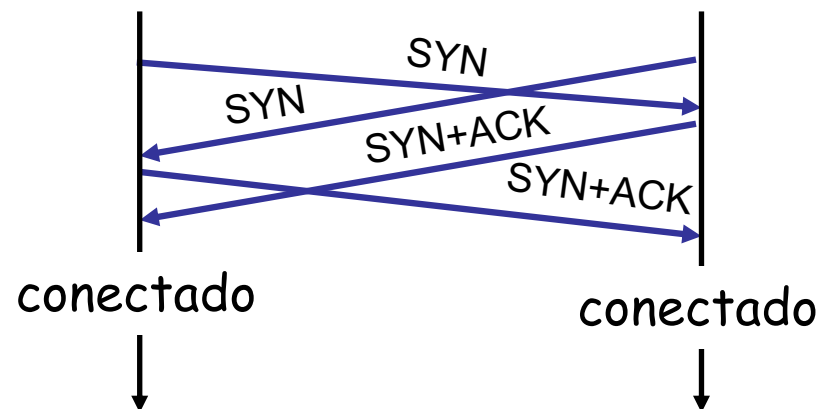
No hay cabecera de transporte



NATs y TCP

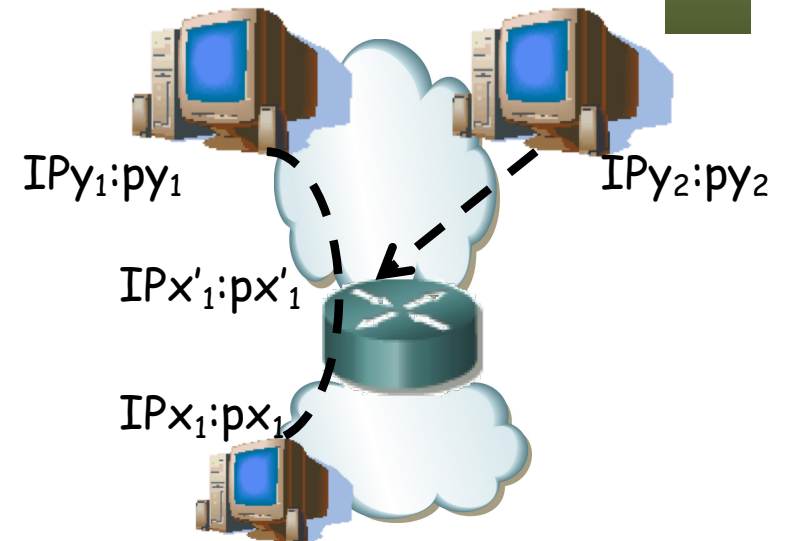
REQs TCP: RFC 5382

- BCP 143 “NAT Behavioral Requirements for TCP” (2008)
- REQ: Comportamiento “*Endpoint-Independent Mapping*”
 - Es decir, mantener mismos IP:puerto externos para mismos internos independiente del destino externo
 - Necesario para UNSAF
- REQ: Debe soportar todas las secuencias de paquetes TCP; en concreto debe manejar correctamente un *simultaneous open*
 - Que no impida un SYN entrante (sin ACK) si hay mapeo
 - Permite hacer *hole punching* para comunicar dos hosts ambos tras NATs (...)



BCP 143

- Filtrado recomendado (como para UDP):
 - *Endpoint-Independent* para facilitar a las aplicaciones (cualquier host externo puede enviar a IP:puerto externo)
 - *Address-Dependent* para mayor restricción (solo el mismo host)



BCP 143: *Session Refresh*

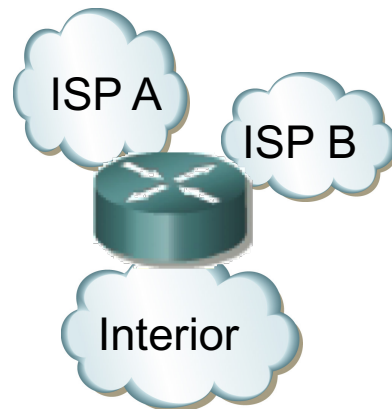
- El mantenimiento de estado es sensible a ataques y agotamiento de recursos
- Además una conexión TCP puede permanecer establecida indefinidamente sin enviar tráfico
- NAT podría eliminar mapeos de sesiones que estima han terminado en el orden:
 - Sesiones en que un extremo se ha reiniciado
 - Sesiones cerradas
 - Sesiones en proceso de establecerse
- El NAT podría comprobar si un extremo se ha reiniciado enviándole un *keep-alive* (recibiría un RST)
- REQ: si no puede determinar si los extremos siguen activos puede abandonar la sesión tras un tiempo
- Ese tiempo debe ser al menos de 2h04m para las establecidas



NATs: Otros aspectos

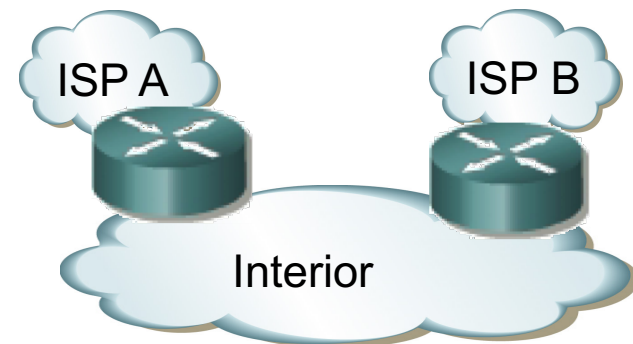
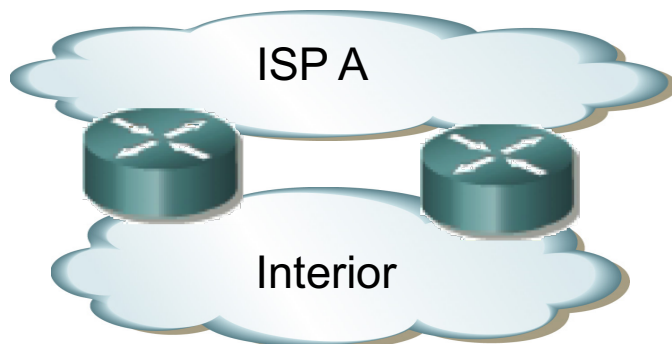
Protección y NATs

- Mantiene el mapeo para una sesión así que todo el tráfico debe pasar por él
- No debe haber asimetría, los dos sentidos de la comunicación deben pasar por el NAT
- Normalmente el NAT está en la frontera de un *stub*
- Es un punto único de fallo
- ¿ Multihomed ?
 - Protección con enlaces a varios operadores
 - La dirección externa será diferente así que los mapeos fallarán salvo que se enrute de vuelta al NAT por el segundo operador



Protección y NATs

- Mantiene el mapeo para una sesión así que todo el tráfico debe pasar por él
- No debe haber asimetría, los dos sentidos de la comunicación deben pasar por el NAT
- Normalmente el NAT está en la frontera de un *stub*
- Es un punto único de fallo
- ¿ Múltiples NATs ?
 - Protección ante fallo del equipo y de enlace si van a diferente ISP
 - Podría emplearse para cada sesión el más cercano al destino
 - Si uno falla debería encaminarse el tráfico por el otro
 - Las sesiones entonces deben estar sincronizadas
 - De nuevo problema con la dirección externa



Seguridad y NATs

- Hosts internos no son directamente accesibles si no inician ellos la comunicación
- Cambian direcciones y puertos así que no funcionan con mecanismos de seguridad que los firmen (IPSec)
- Sí funcionan con aplicaciones que no basen la seguridad en direcciones IP o puertos (SSH, TLS)



Protocolos sobre IP

- Campo protocolo 1 byte (256 valores)
- Hay ya al menos 140 reservados (2016)
- Pero un NAT soporta traslación solo para TCP/UDP/ICMP
- Con el despliegue que hay de NATs, el empleo de otros protocolos hoy en día no tendría alcance global
- Nuevos protocolos acaban implementándose sobre UDP que da un servicio de datagramas como IP

