

REDES DE ORDENADORES  
3º Grado Ing. Informática  
Universidad de Vigo



BLOQUE 2  
TEMA 5  
Tecnologías NAT y Proxy

## Problemática conexiones privadas-públicas-privadas



- Existen rangos de direcciones no gestionadas por el IANA. Estos rangos de direcciones están referidos al direccionamiento privado.
- Los routers de internet no definen rutas para esas direcciones:
  - ¿qué ruta seguir desde internet hasta la 192.168.1.0?.
- Además, el RFC1918 recomienda que cualquier router de Internet rechace cualquier información de enrutamiento referida a direccionamiento privado, lo que se traduce en que cualquier paquete cuya dirección IP origen o destino pertenezca a esos rangos, será eliminado:
  - “.... routing information about private networks shall not be propagated on inter-enterprise links, and packets with private source or destination addresses should not be forwarded across such links. Routers in networks not using private address space, especially those of Internet service providers, are expected to be configured to **reject** (filter out) routing information about private networks. If such a router receives such information the rejection shall not be treated as a routing protocol error.”
- El grado de aplicación de este RFC es muy alto.

## Problemática conexiones privadas-públicas-privadas



- Por tanto la problemática está en como acceder a un host que tenga dicha dirección, y por tanto en como un host con esa dirección accede a Internet.
- La solución se implementa en la capa de aplicación de los routers que hacen posible la comunicación pública-privada.
- Este router:
  - debe disponer de al menos una dirección pública que conecte a Internet, y una dirección privada que conecte a la organización.
  - no puede tener activada la RFC 1918.

# NAT



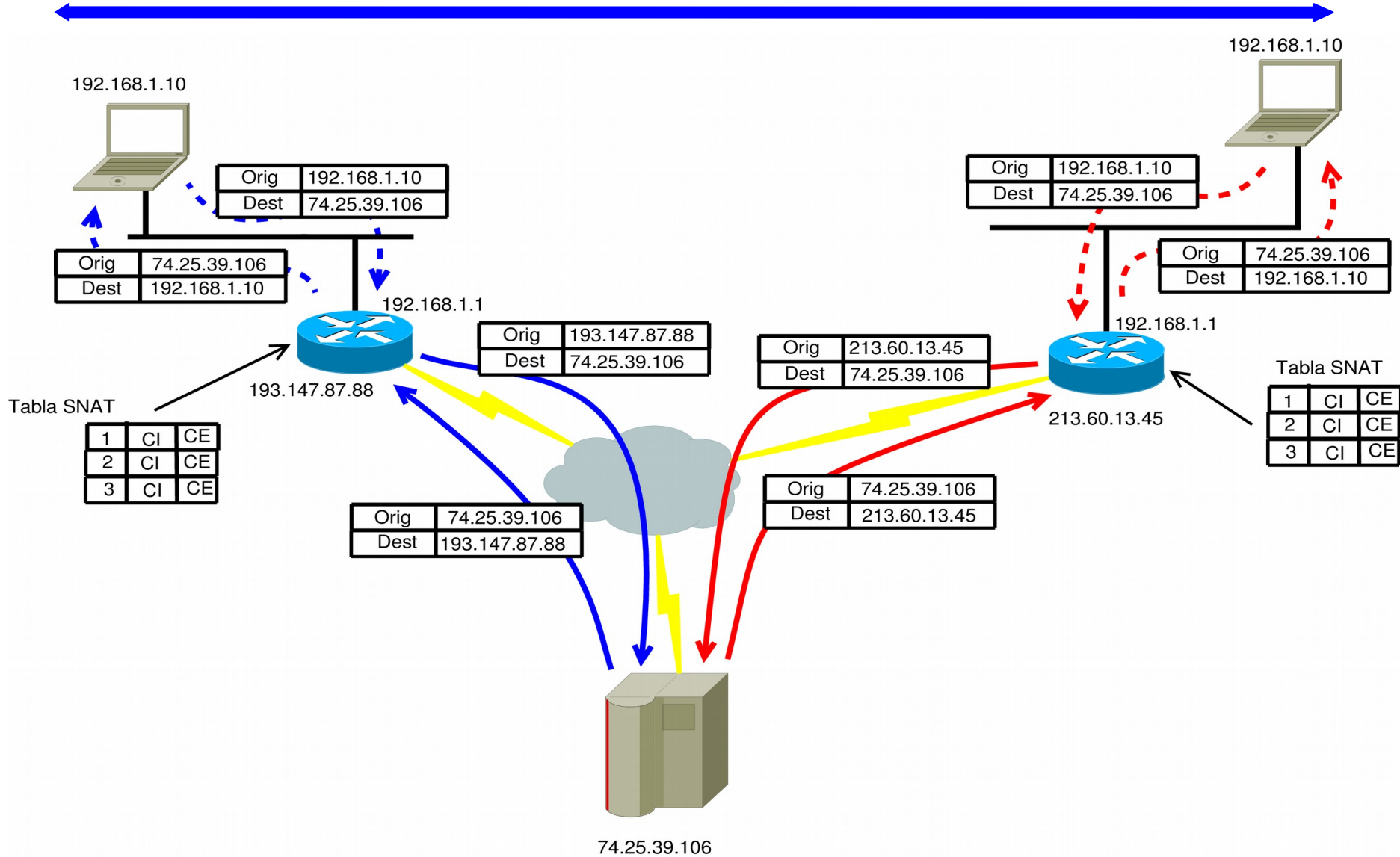
- El concepto de NAT se aplica a varias soluciones de interconexión de IP's privadas con IP's públicas. RFC 2663.
- Existen dos variantes básicas de los sistemas NAT:
  - NAT de origen (o Source NAT). Se representa normalmente como **SNAT**.
  - NAT de destino (o Destination NAT). Se representa normalmente como **DNAT**.

## SNAT



- Consiste en enmascarar la dirección privada del origen con la dirección pública del router en el sentido salida hacia la red pública.
- En el router se mantiene una tabla que relaciona la conexión de transporte (IP destino, IP origen, protocolo, identificador de la conexión) privada con la conexión de transporte pública. Así se puede volver al host origen.
- Se activa normalmente en el Kernel del sistema operativo (Windows, Linux – UNIX) (ip\_masquerading), o viene por defecto (o como opción) en Sistemas operativos de red (IOS, LRP, OpenWRT).
- El cliente se configura para utilizar el router como dispositivo NAT de salida.

# SNAT




## SNAT



- Existen dos variantes básicas del SNAT:
  - Estático: el router dispone de mas IP's públicas de salida que IP's privadas a las que dar acceso a internet.
  - Dinámico: el router dispone de una sola IP pública y tiene que dar salida a mas de una IP privada. Este es el caso mas habitual.

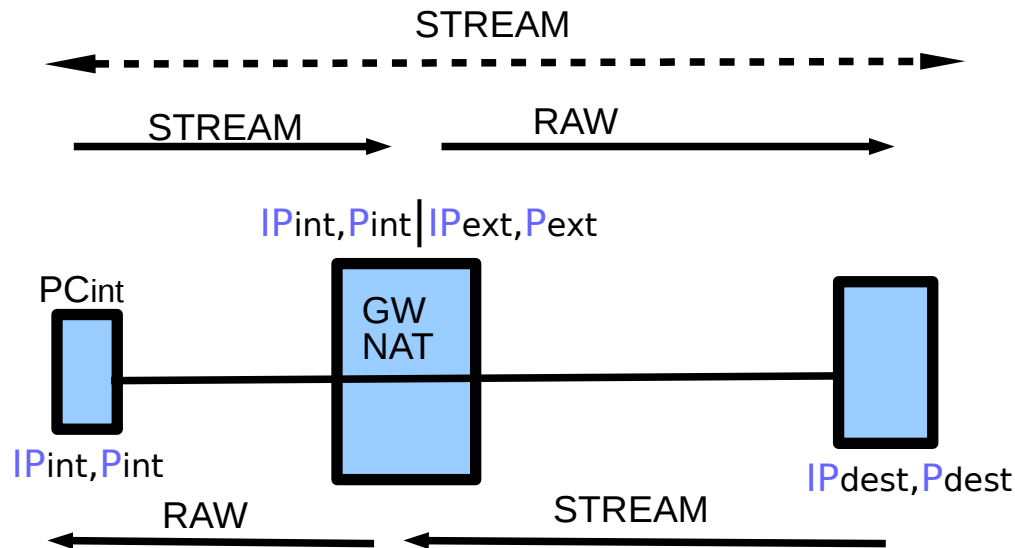
## SNAT

- 
- Los registros de las tablas SNAT internas tienen una vida limitada. Las tablas SNAT son dinámicas.
  - La identificación de conexión interna y externa no es única, pero en general consta de:
    - CI:  $IP_{\text{interna}}, P_{\text{interno}}$
    - CE:  $IP_{\text{externa}}, P_{\text{externo}}, IP_{\text{dest}}, P_{\text{dest}}$
  - La solución SNAT tiene algunos problemas:
    - Necesidad de procesamiento de forma muy rápida.
    - Necesidad de recalcular y modificar determinados campos de la cabecera IP en los routers.
    - Necesidad de soporte para determinadas aplicaciones con el destino en el campo de datos.
    - No funciona en sentido inverso, es decir, si el equipo con direccionamiento privado es un servidor público.




## SNAT

- Nivel de sockets (Caso conexiones TCP)



## SNAT

- 
- En función de cómo se resuelve la identificación de conexiones dentro de la pasarela NAT, .... existen varias implementaciones:
    - **CI**:  $IP_{\text{interna}}, P_{\text{interno}}$
    - **CE**:  $IP_{\text{externa}}, P_{\text{externo}}, IP_{\text{dest}}, P_{\text{dest}}$
  - existen varias implementaciones
    - **Full-cone NAT** (one-to-one NAT)
    - Address-restricted-cone NAT
    - Port-restricted-cone NAT
    - **Symetric NAT**

## SNAT

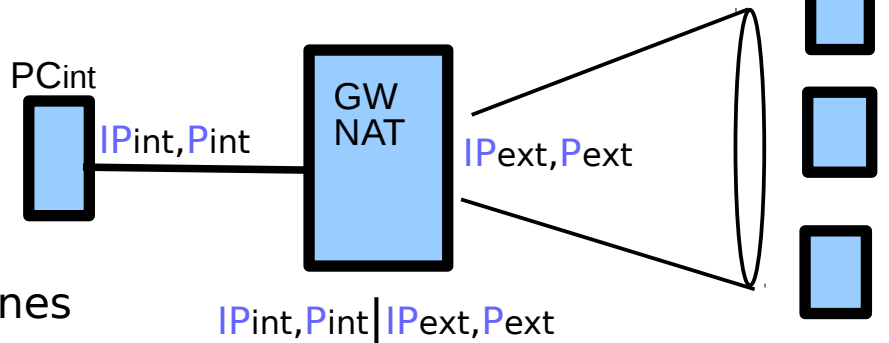
### • Full-cone NAT (one-to-one NAT)

- La  $IP_{dest}, P_{dest}$  no se tiene en cuenta. Una vez mapeadas las IPs y Puertos internos y externos, cualquier destino pasa a través de ese mapeo.

- Cualquier host que quiera enviar a la  $IP_{interna}, P_{interno}$  puede hacerlo a través de la  $IP_{externa}, P_{externo}$ .

- CI:  $IP_{interna}, P_{interno}$
- CE:  $IP_{externa}, P_{externo}$

- Pocos registros en la tabla NAT
- Sólo para conexiones UDP
- Adecuado para grandes corporaciones
  - Habilitan filtrados por Firewalling y Proxies.



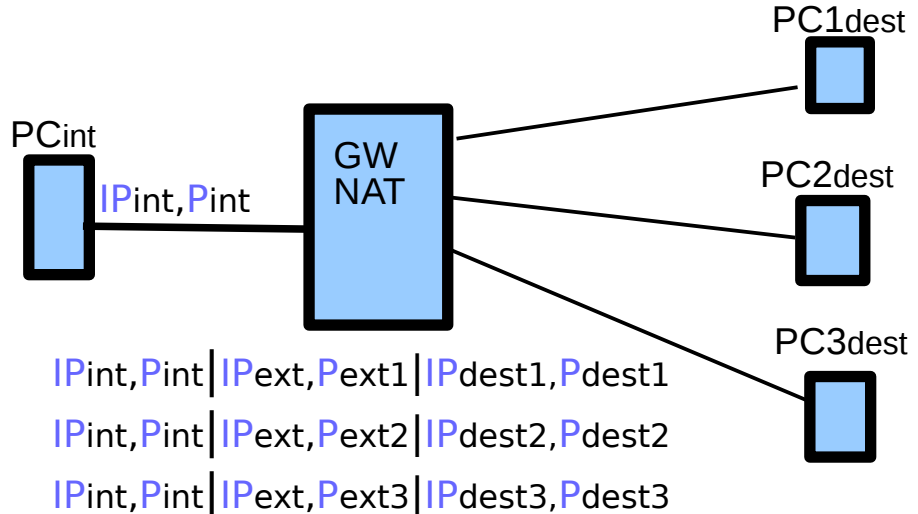
## SNAT

### • Symetric NAT

- La  $IP_{dest}$ ,  $P_{dest}$  si se tienen en cuenta. Cada conexión hacia un destino nuevo tiene un nuevo mapeo.
- Cualquier host externo que quiera enviar a la  $IP_{interna}$ ,  $P_{interno}$  tiene que haber recibido antes una conexión desde la  $IP_{externa}$ ,  $P_{externo}$ .

- $CI: IP_{int}, P_{int}$
- $CE: IP_{ext}, P_{ext}, IP_{dest}, P_{dest}$

- $P_{ext1}$ ,  $P_{ext2}$  ... pueden ser iguales
- **Muchos** registros en la tabla NAT
- Adecuado para SOHO
- Por defecto en TCP

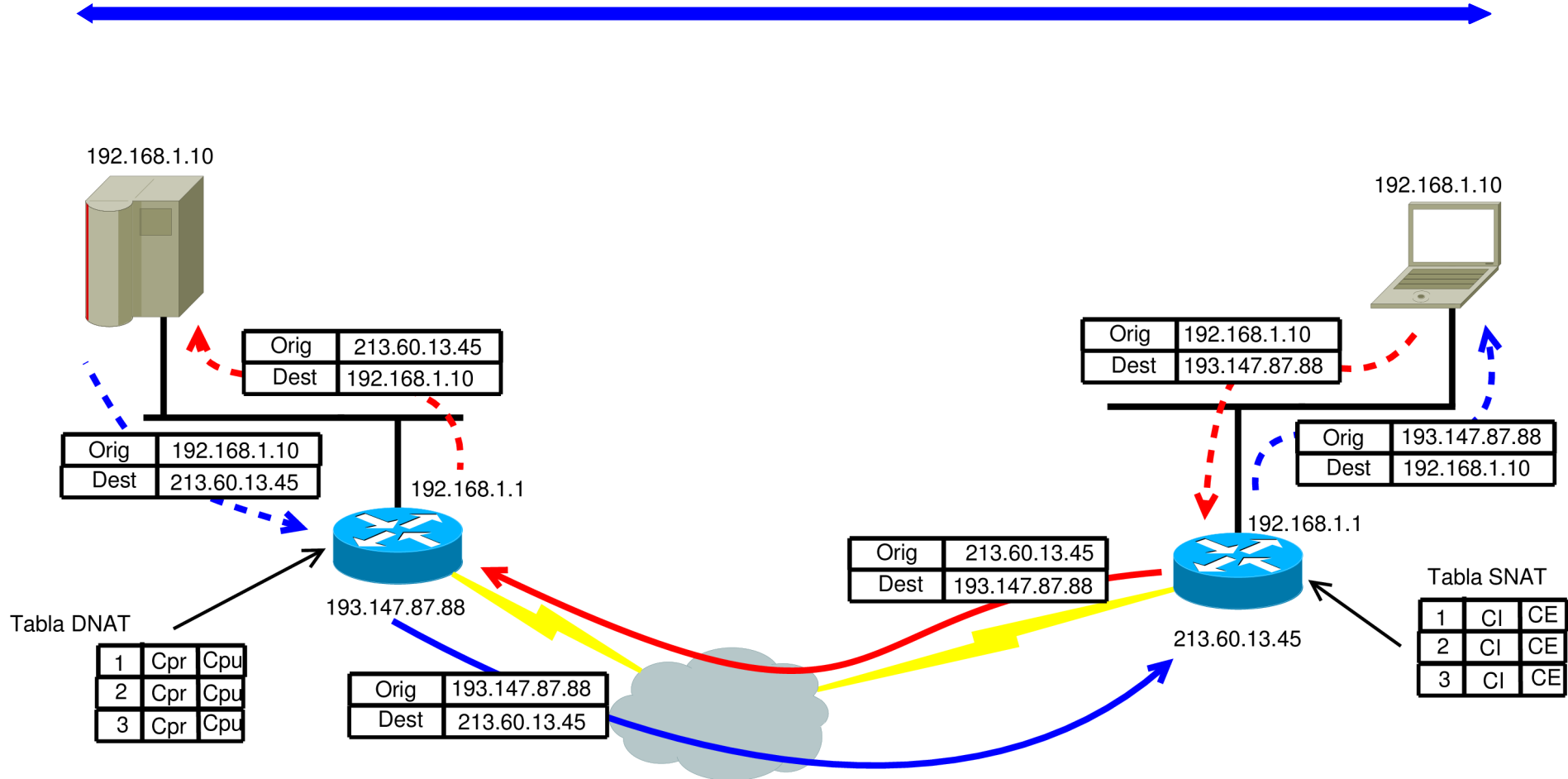


## DNAT



- Resuelve la conectividad pública->privada.
- Se le llama también NAPT o NAT reverso. No hay una solución uniforme, aunque la idea es similar en todas ellas.
- La solución DNAT trabaja sobre tablas estáticas:
  - Conexión privada: Puerto privado servidor, IP privada servidor.
  - Conexión pública: Puerto público, IP pública.
- Posibles problemas en servicios multipuerto.

# DNAT



## Conexiones Peer to Peer



- Problema de conexión a un servicio interno sin usar DNAT
- Servicios como Skype, Torrent, Gaming.
- Se involucra siempre un “well known server” S que ofrece información de direcciones y puertos entre los usuarios.
- No hay solución estándar.
- Se utiliza habitualmente el método “UDP Hole”
- Mucho mas complejo en TCP.


## Conexiones Peer to Peer (UDP Hole)



- Full cone NAT
  - Relativamente sencillo, al aceptar el GW NAT conexiones de cualquier origen una vez existe un UDP Hole de A hacia afuera.
- Simetric NAT
  - Si el  $P_{ext}$  se mantiene, el proceso sería:
    - A y B informan a S de sus conexiones.
    - A intenta ir a B, pero el GW NATB rechaza la conexión por no ser de un origen conocido (y no existir entrada en la tabla SNAT).
    - B intenta ir a A. En este caso el GW NATA acepta la conexión por existir una entrada en la tabla SNAT previa hacia B.

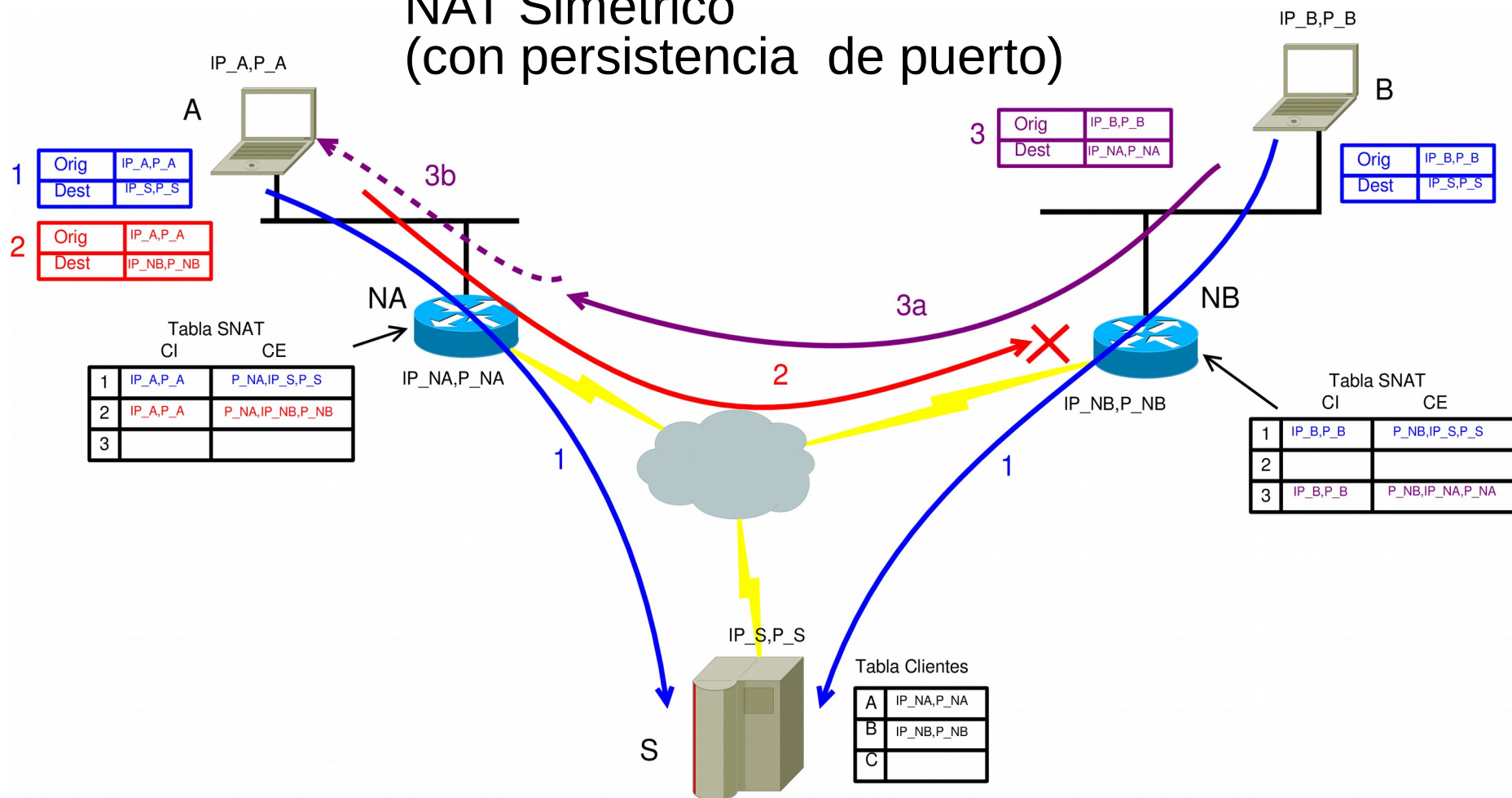


## Conexiones Peer to Peer (UDP Hole)

- 
- Simetric NAT
    - Si el  $P_{\text{ext}}$  no se mantiene:
      - Habitualmente se generan entradas en la tabla SNAT con puertos  $P_{\text{ext}}$  consecutivos.
      - A y B informan a S de sus conexiones.
      - A intenta ir a B, pero el GW NATB rechaza la conexión por no ser de un origen conocido (y no existir entrada en la tabla SNAT).
      - B intenta ir a A. En este caso el GW NATA también rechaza la conexión por no existir una entrada en la tabla SNAT previa hacia B.
      - A y B intentan conectarse a sus peers en puertos consecutivos. El primero que lo consiga establece la conexión e informa a su peer de que deje de intentarlo.
    - Si no se utilizan puertos consecutivos, no hay solución.

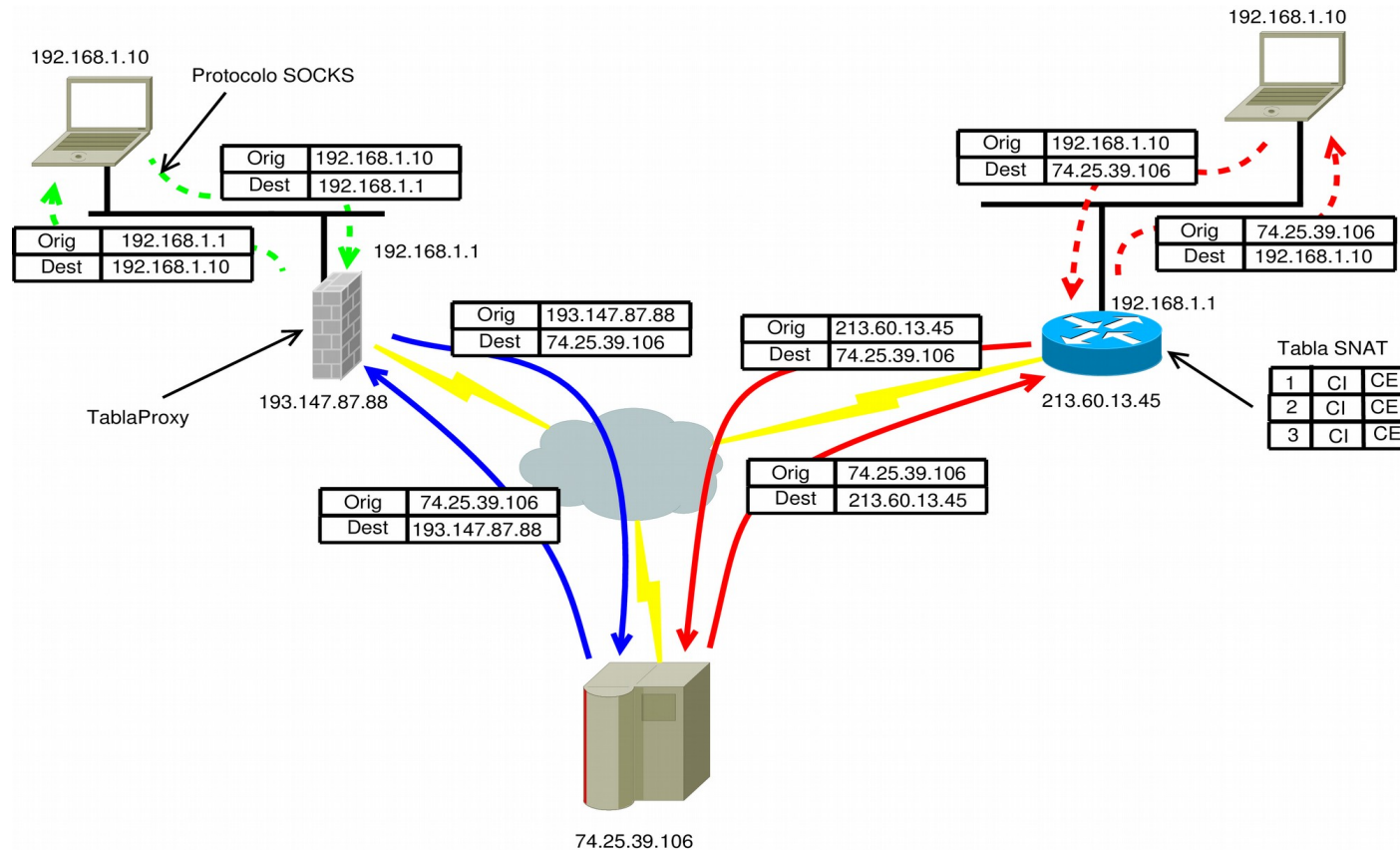
## Conexiones Peer to Peer (UDP Hole)

### NAT Simétrico (con persistencia de puerto)



## PROXY

- Resuelve algunos problemas de la conectividad privada->pública.
- Es una pasarela de aplicación. No encamina ni enmascara. Analiza la aplicación cliente y realiza el trabajo por ella.



## PROXY

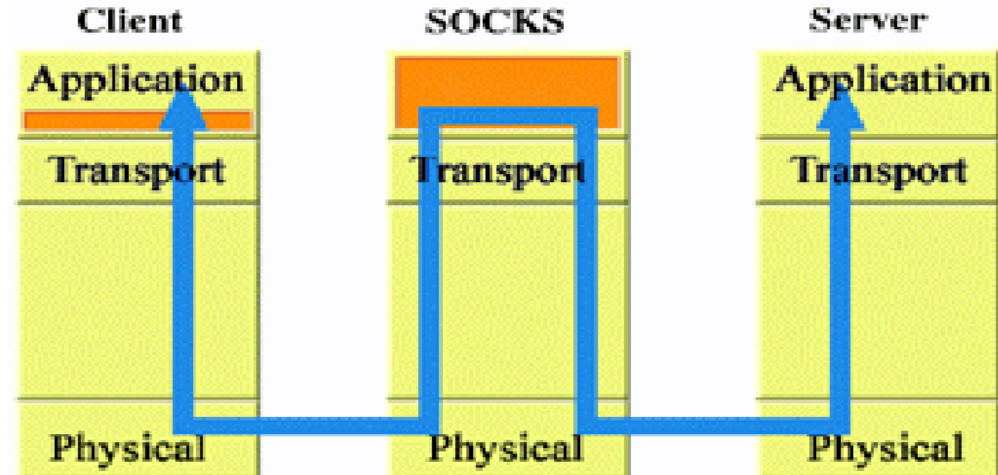


- Ventajas:
  - Aumenta la seguridad
  - Permite servicios avanzados:
    - control de usuarios
    - control de accesos por franjas horarias
    - control de accesos por contenidos
  - Usa una algoritmia de caché para el ahorro de consumo de caudal y mejorar el tiempo de respuesta.
- Desventajas:
  - Es específico de las aplicaciones de red mas comunes, pero no de todas.
  - La caché no tiene el destino completamente actualizado.
  - Impide algunos servicios avanzados al no utilizar todos los puertos de la aplicación.

## PROXY

### • Protocolo SOCKS

- Versión 4 y 5.
- La versión 5 (RFC 1928, 1996) implementa mayor seguridad que la 4 e incluye el transporte UDP, pero la funcionalidad es la misma.
- Permite resolver en parte el problema de los puertos no soportados por el proxy.
- La aplicación de usuario debe implementar ese protocolo: existen aplicaciones que permiten a cualquier aplicación utilizar el protocolo SOCKS.




## PROXY



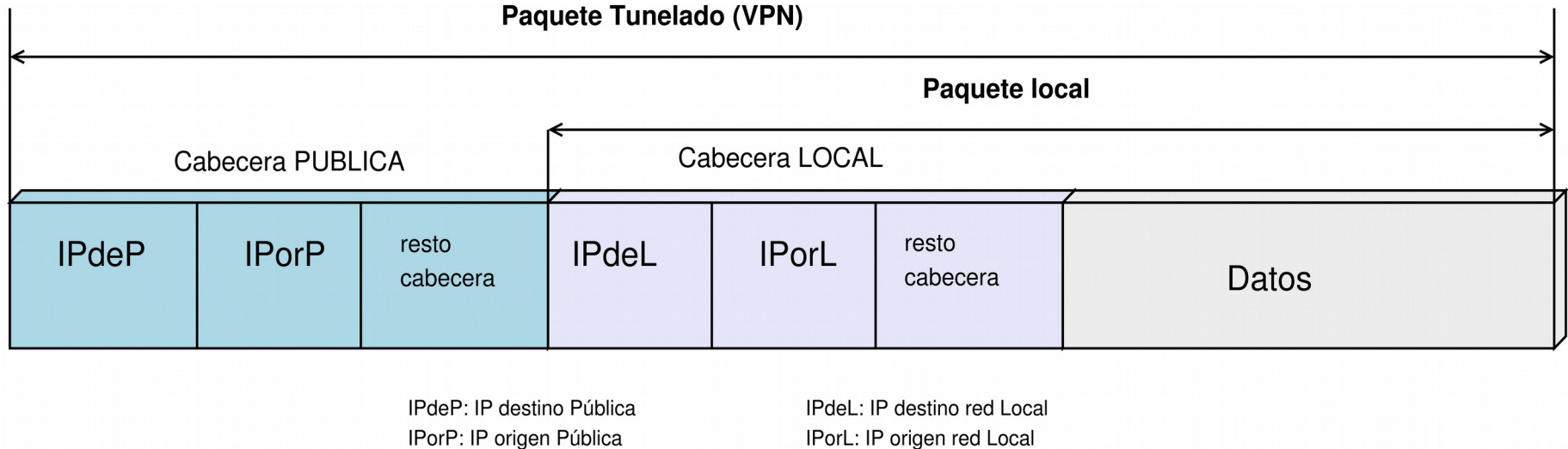
- Funcionamiento básico del protocolo SOCKS
  - Cliente SOCKS envía solicitud sobre TCP en el puerto especificado. La solicitud incluye:
    - Versión del protocolo SOCKS
    - Código de comando
    - Puerto destino
    - IP destino
    - Identificación de usuario
  - Servidor SOCKS envía respuesta a través de la conexión TCP establecida. La respuesta incluye
    - Campo nulo
    - Estado
    - Puerto destino
    - IP destino
  - Si la respuesta del servidor es positiva, los siguientes paquetes del cliente hacia el servidor SOCKS son redirigidos al puerto e IP destino.

## PROXY

- 
- Proxy – NAT (proxy transparente)
    - Es un servicio NAT dirigido a un proxy, en la misma máquina o en otra. El servicio NAT genera las peticiones al proxy y enruta los paquetes a ese servicio.
    - El usuario sólo ve un servicio NAT, y no suele conocer la existencia del proxy, pues no requiere configuración en el equipo de usuario.
    - Utilizado por proveedores ISP.
  - Proxy reverso:
    - Permite establecer conectividad pública-privada y redirigir a voluntad las solicitudes a diversos servidores internos.
    - Utilizado por ISP para sus servicios.

# Tunneling

- Estructura de información dentro de otra. Nivel 3 coloca la información local de la conexión dentro del paquete de datos de la estructura IP.

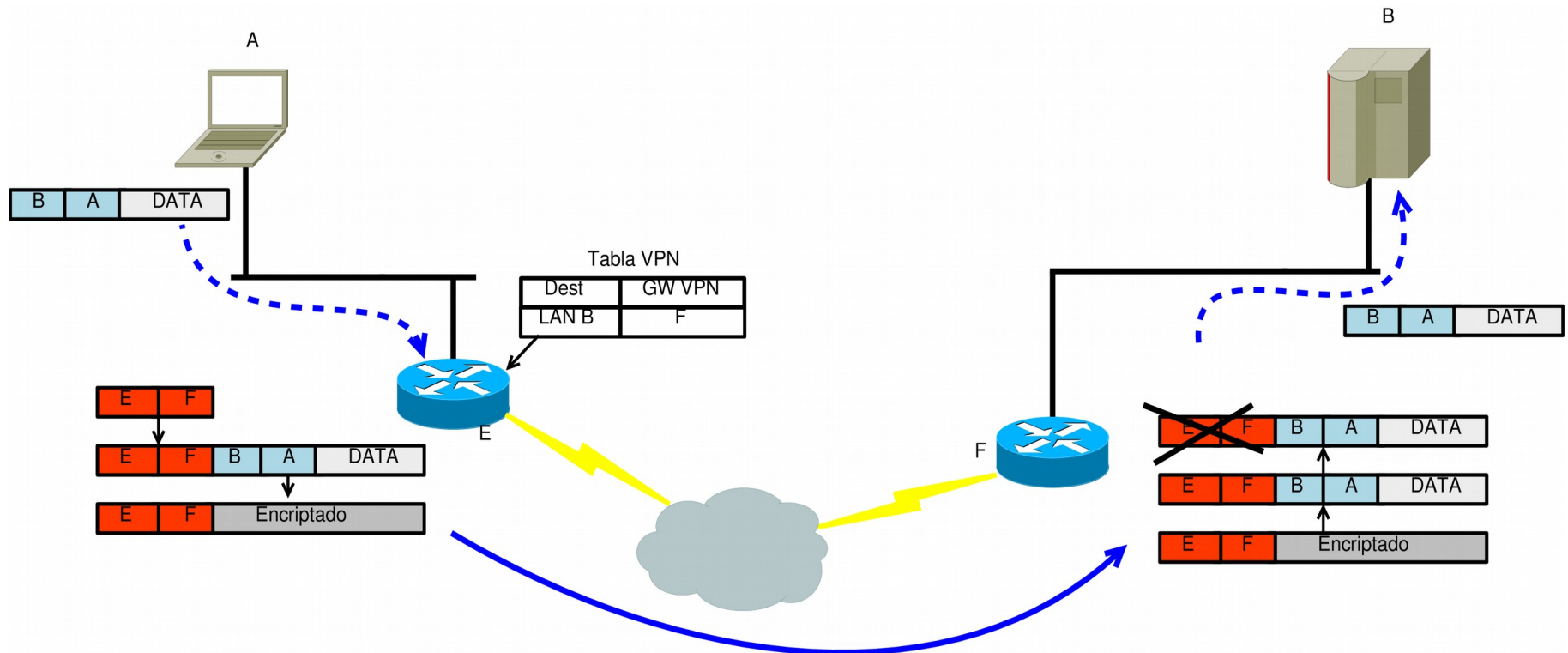


- Normalmente va encriptada.
- Es útil para conectar diversas oficinas remotas de una organización a través de redes públicas con sensación de estar en la misma red local: **VPN**



# Tunneling. VPN

- VPN: Virtual Private Network
  - Tunneling
  - Autenticación
  - Encriptación



## Tunneling/VPN



- Es una arquitectura Cliente/Servidor
- Los extremos cliente/servidor mantienen una tabla de rutas alternativa para redirigir los paquetes a las redes locales a través de redes públicas.
- Los extremos cliente/servidor encriptan/desencriptan el paquete local.
- Existen soluciones puenteadas y enrutadas

## Tunneling. VPN

### • Topologías

- Peer2Peer: todo el proceso se hace por software en los equipos terminales. Uno de los equipos tiene que estar visible en Internet, o se necesita una pasarela central.
- Acceso entre subredes
- Acceso remoto desde PC a una subred (Llanero solitario).

