Ciberseguridad

Tópicos avanzados en Redes de computadoras

Clase 2

Vlans, Ruteo y Túneles.

COMENZAMOS

Clase 2

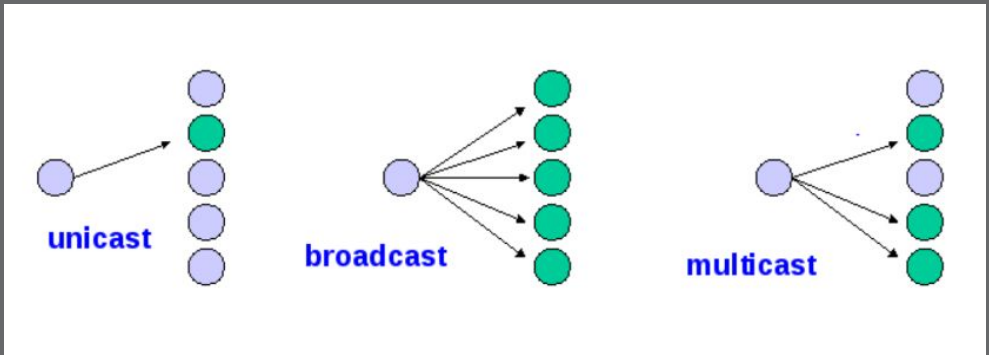
Contenidos de la clase

- Segmentación
- VLANs
- Ruteo
- NAT
- Túneles y VPNs

Segmentación

Tipos de tráfico

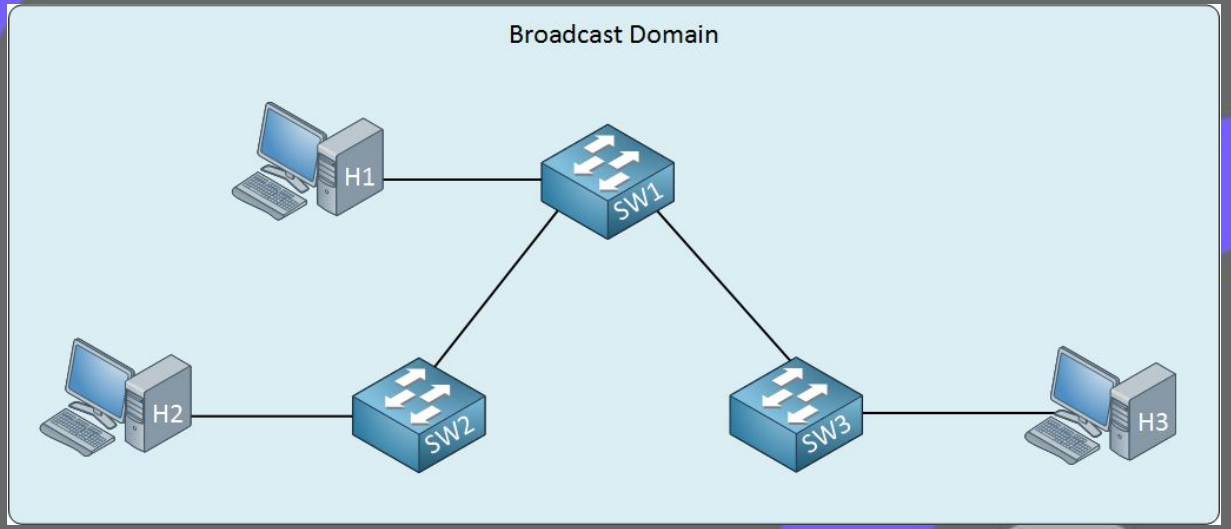
- Unicast: destino a un host/interfaz.
- Broadcast: destino a todos los hosts en una red.
- Multicast: destinada a un grupo de hosts en una red o varias redes.
- Anycast: destinada al primero que resuelva (al más “cercano”).



Redes Lan - Dominios de Broadcast

Un dominio de broadcast se refiere al grupo de dispositivos de red que son capaces de recibir información de broadcast entre sí.

Por ejemplo, esta red con 3 switches refiere al mismo dominio de broadcast, sumados los 3 hosts.



Funciones de un Switch

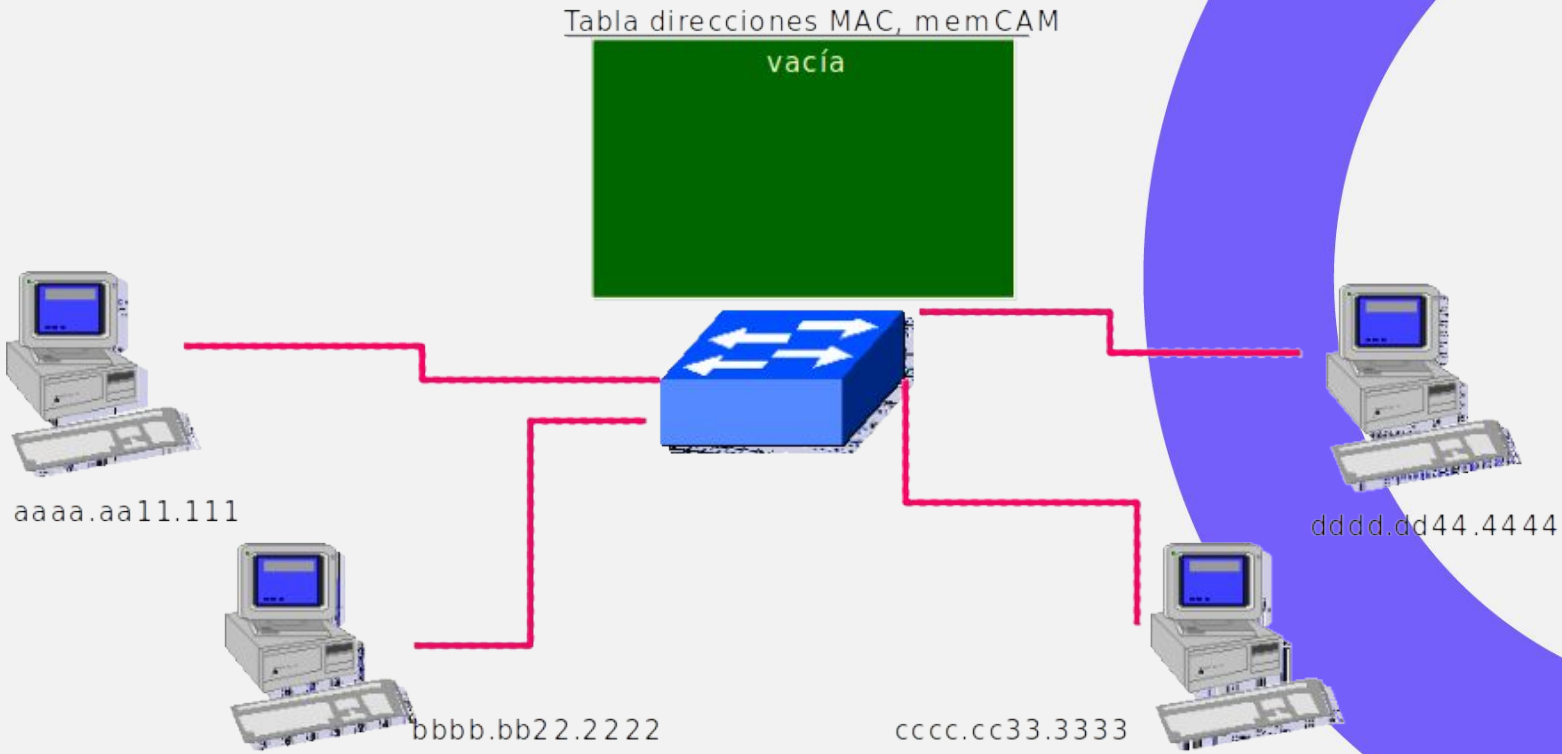
- Aprender direcciones MAC:
  El dispositivo guarda las direcciones MAC asociadas a cada puerto en una base de datos.

- Reenviar / filtrar paquetes:
  Al recibir una trama, el switch revisa su base de datos MAC para determinar a través de qué puerto puede alcanzar la dirección de destino.

- Evitar bucles de capa 2:
  Los switches administran los bucles de redundancia con STP. (Distintas versiones)

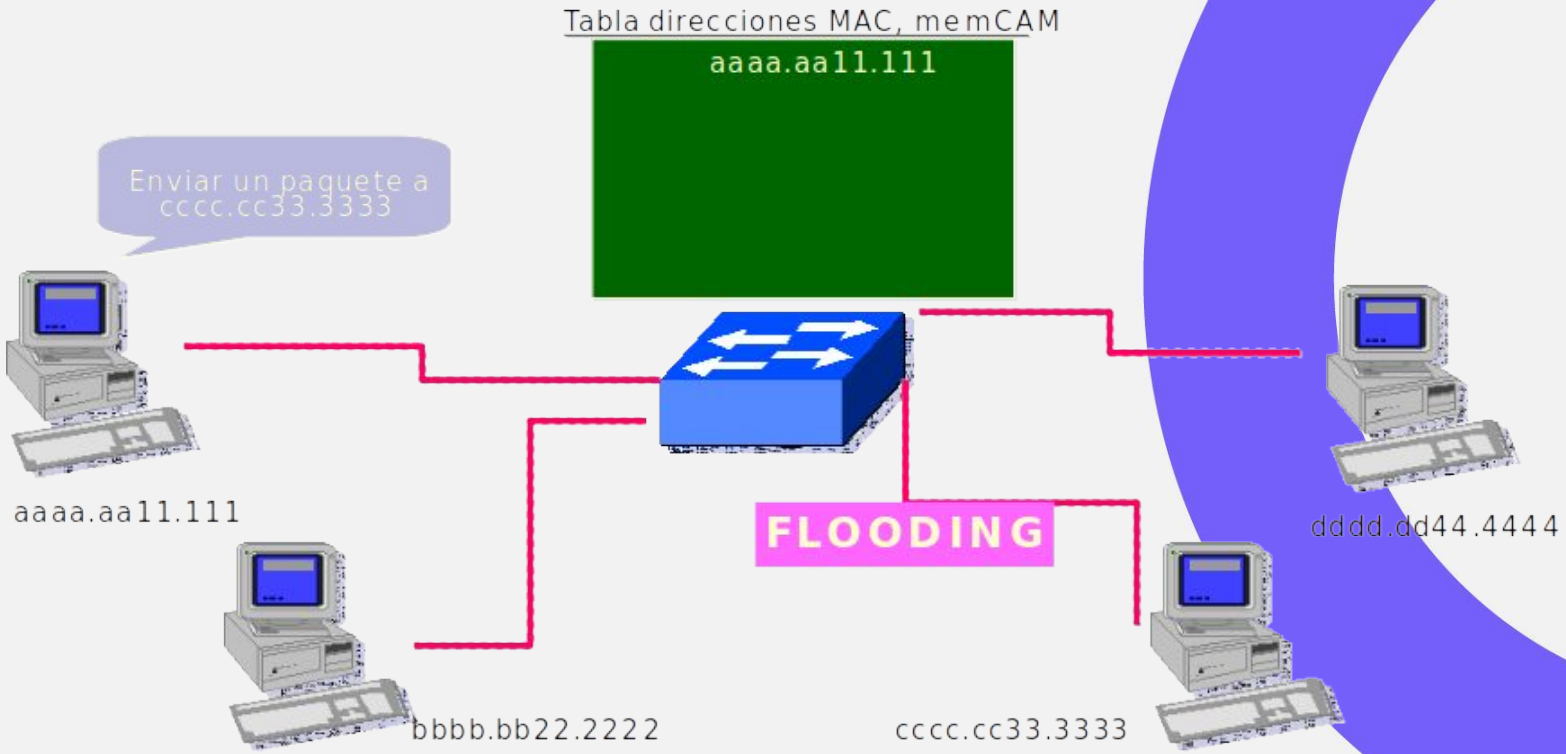
Switching

Aprendizaje de direcciones



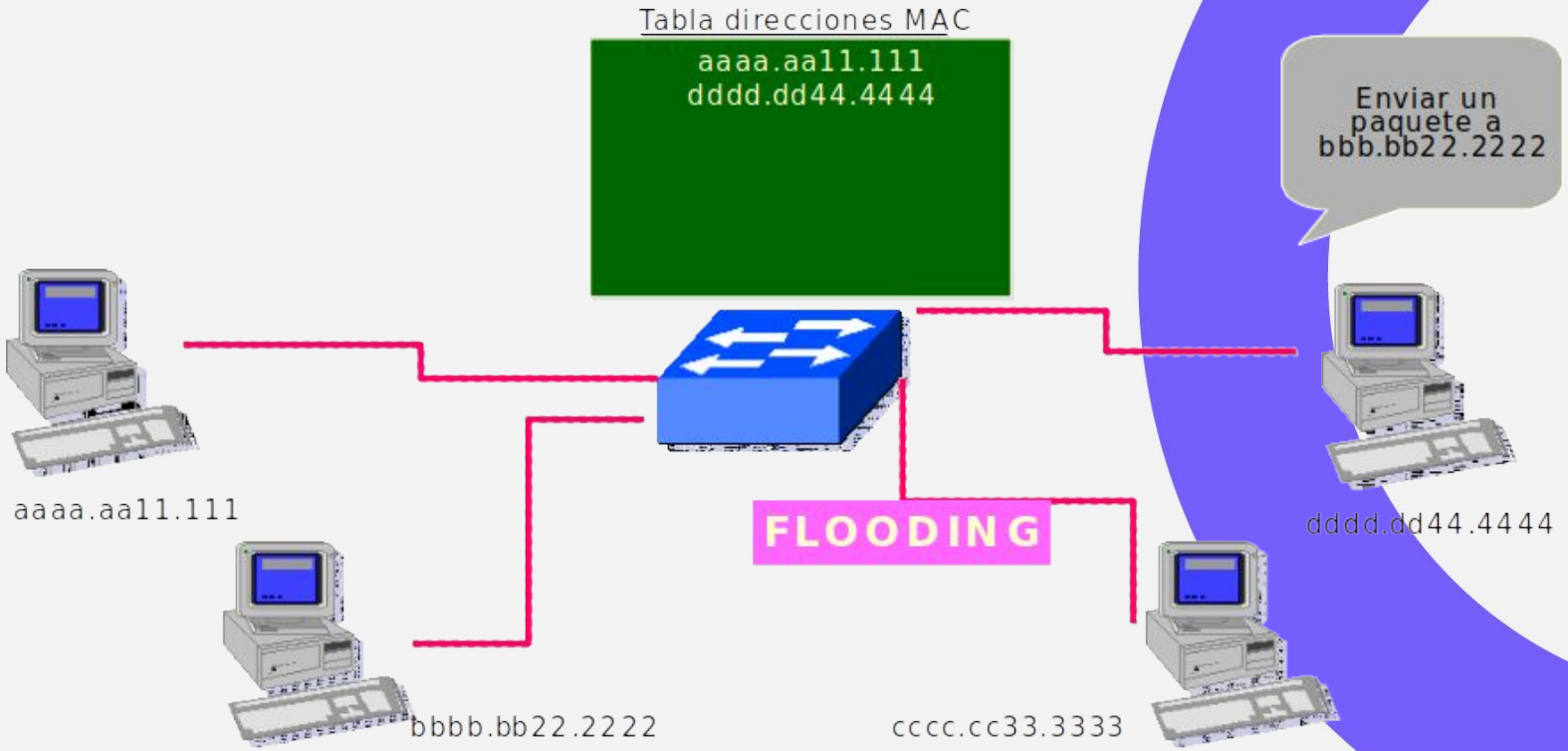
Switching

Aprendizaje de direcciones



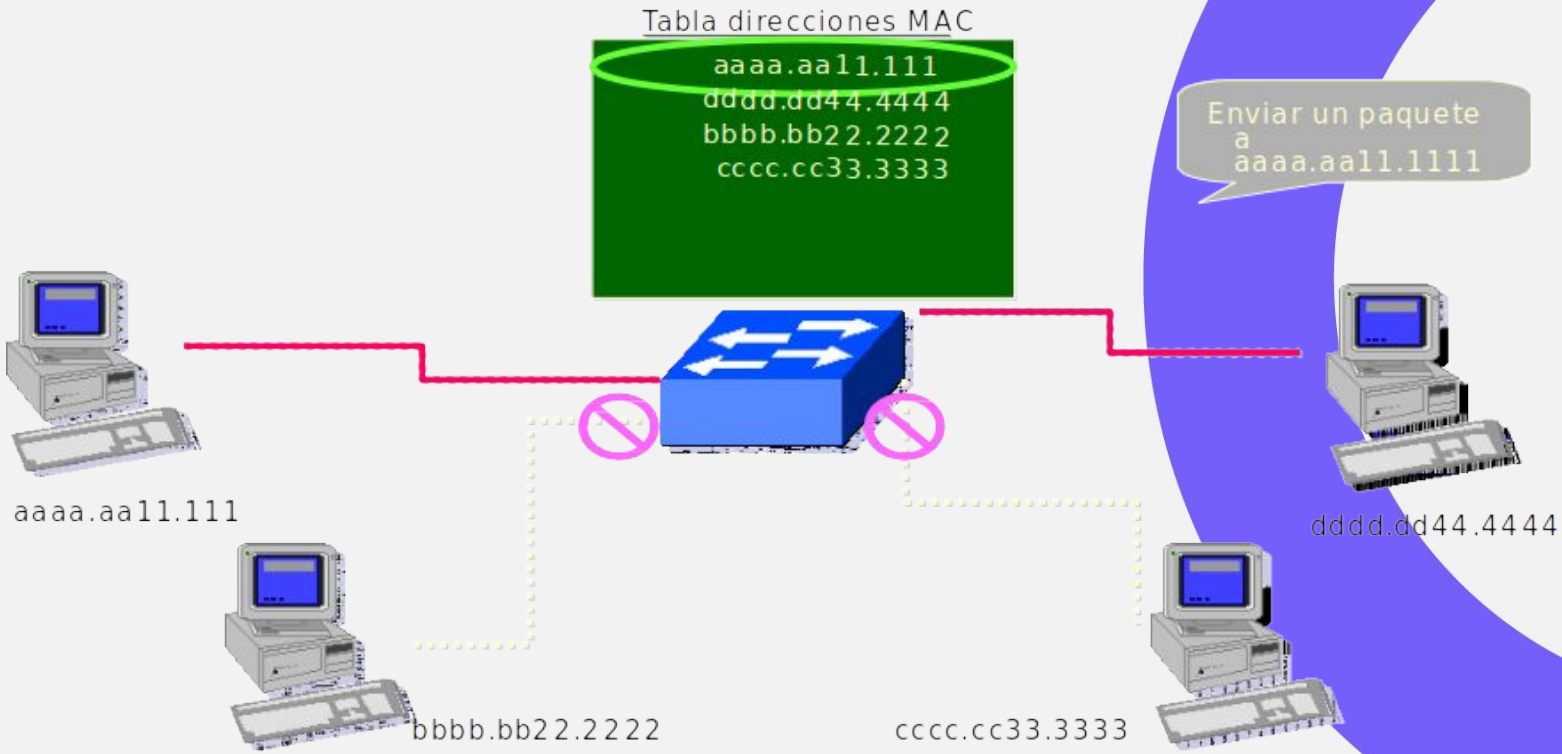
Switching

Aprendizaje de direcciones



Switching

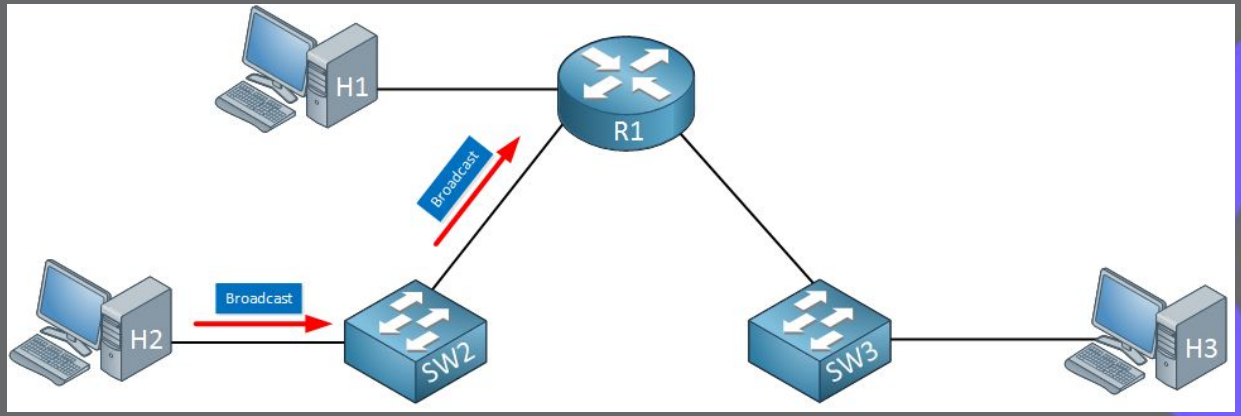
Reenvío / filtrado de paquetes



Redes Lan - Dominios de Broadcast

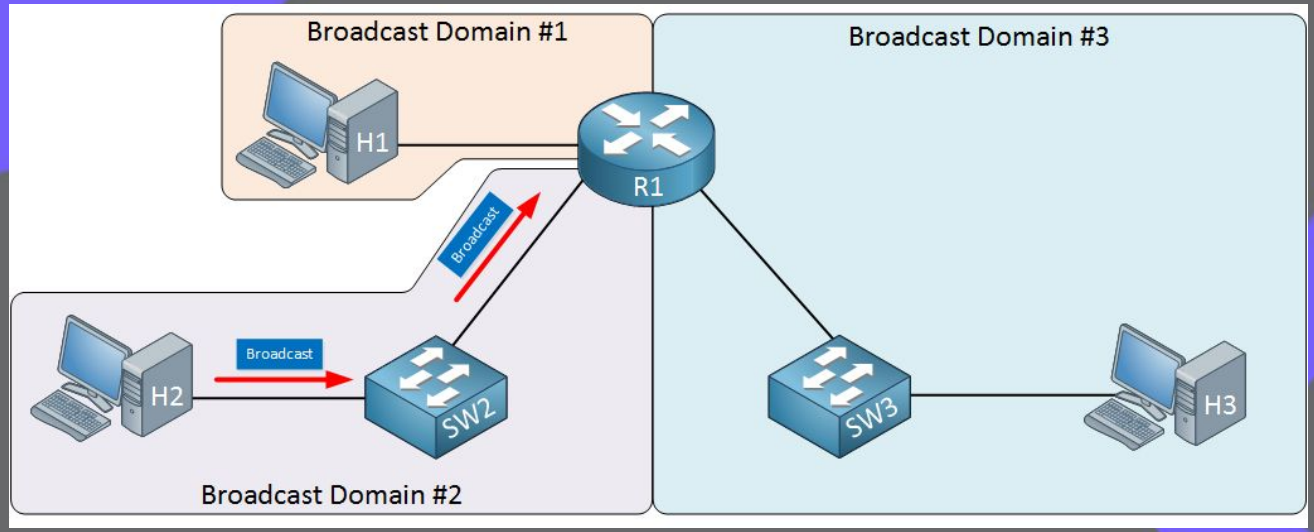
El tráfico de broadcast no es muy eficiente. Un solo dispositivo que envía mucho tráfico de broadcast afecta a todo el dominio y por consiguiente a todos los dispositivos en él, por lo que es una buena práctica limitar el tamaño de los dominios de broadcast.

¿Cómo se dividen los dominios de broadcast? Utilizando routers.



Redes Lan - Dominios de Broadcast

Los routers a diferencia de los switches, no reenvían el tráfico de broadcast: Dividen los dominios de broadcast.



Ahora hay 3 dominios de broadcast, uno en cada "lado" del router.

VLANs

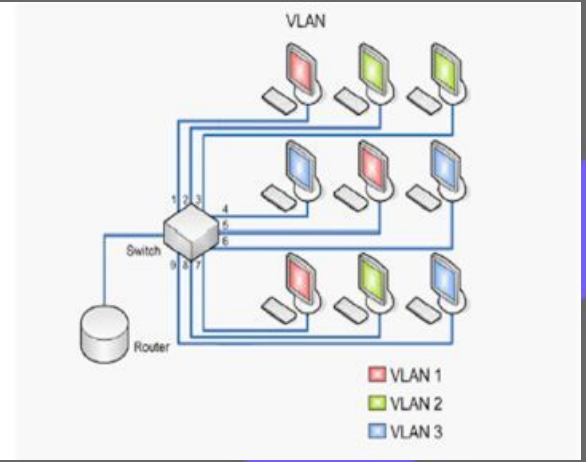
¿Se pueden crear múltiples dominios de broadcast en un switch?

Si, utilizando el concepto de VLAN - Virtual LAN

- **Definido en el protocolo IEEE 802.1Q.**
- **Cada VLAN es un dominio de broadcast independiente.**

VLANs

Equivale a “dividir “ un switch en varios más pequeños.



Objetivos:

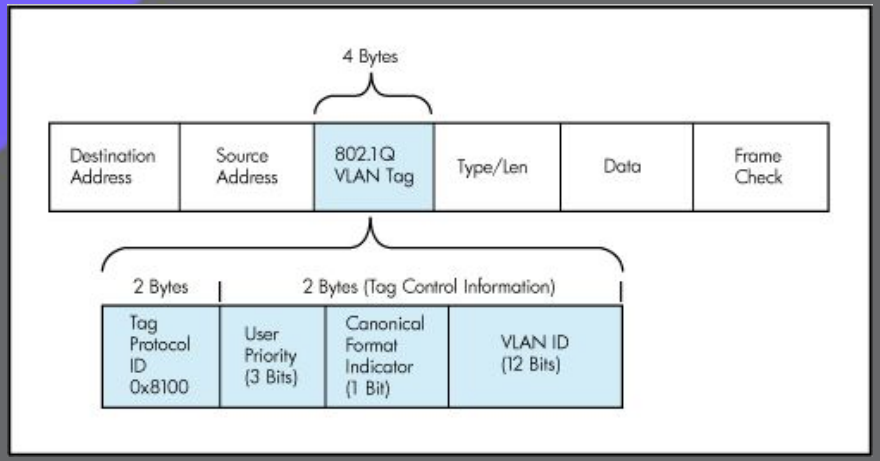
- Rendimiento (reducir tráfico broadcast)
- Gestión
- Seguridad

Características

- Múltiples dominios broadcast.
- Diferentes ports en diferentes VLANs.
- Para lograr conectividad se deben conectar mediante uplinks o routers.
  ¿Siempre necesito un router?

VLANs

Formato de una trama la trama 8021Q



Se utilizan 12 bits para definir el TAG de vlans, por lo que el número máximo de vlans posibles es 4096 (2^12).
Quedando el 0 y el 4095 como números reservados, se pueden usar los números de VLAN desde la VLAN 0001 hasta la VLAN 4094.

VLANs

Definimos 2 tipos de tráfico:

- Tráfico Untagged
  Todo el tráfico desde/hacia un dispositivo final que no participa activamente del esquema de VLANs, es tráfico “clásico” sin información de vlans.

- Tráfico Tagged
  El tráfico clasificado para pertenecer a alguna vlan determinada, viajará con el encabezado 8021q donde está definido el TAG de vlan al que pertenece.

Cuando el tráfico llega a un switch, este es el encargado de o bien colocarle un TAG y enviarlo con esta nueva información(si es necesario), o quitarle el TAG para realizar el envío hacia el dispositivo final. Esto depende del tipo de puerto en el switch.

VLANs

Tipos de puertos:

- Access
  Son puertos generalmente conectados a dispositivos finales que reciben y envían tráfico sin TAG. Se los configura como pertenecientes a una VLAN (Solo una vlan untagged por puerto).

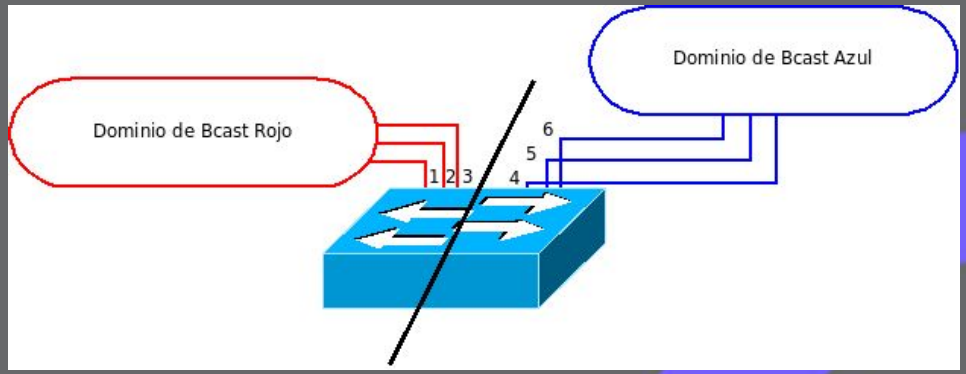
- Trunk
  Son puertos que generalmente se interconectan con otros Switches o Routers. Envían y reciben tráfico con TAGs, para informar al siguiente dispositivo a que VLAN corresponde el tráfico, usando un solo puerto para varias VLANs.

- Hybrid/General
  Algunos equipos tienen la capacidad de poder definir puertos que aceptan tráfico con y sin TAG. Lo que tiene TAG lo aceptan con ese TAG y lo que viene UNTAGGED lo colocan en una VLAN según su configuración (Sólo una vlan untagged por puerto).

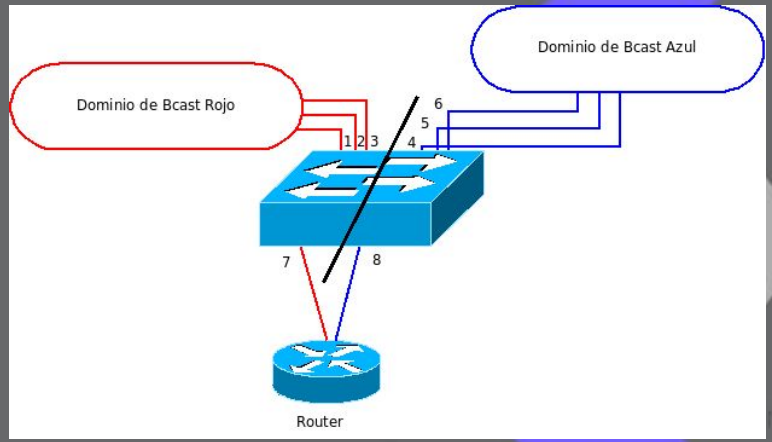
VLANs

VLANs Separadas

Cada puerto en una VLAN (dominio).
Puertos Access.

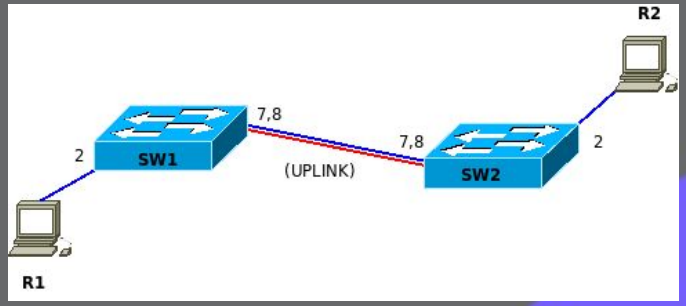


Ej de Interconectar VLANs mediante dispositivo L3 (router).

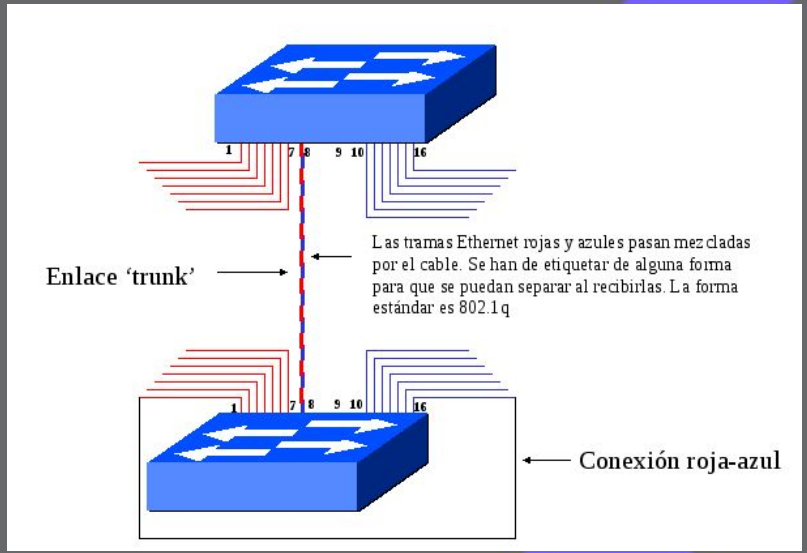


VLANs

VLANs (uplinks)
Hacer uplinks entre VLANs requeriría un enlace por VLAN si cada puerto solo está en una VLAN.
Puertos Access.

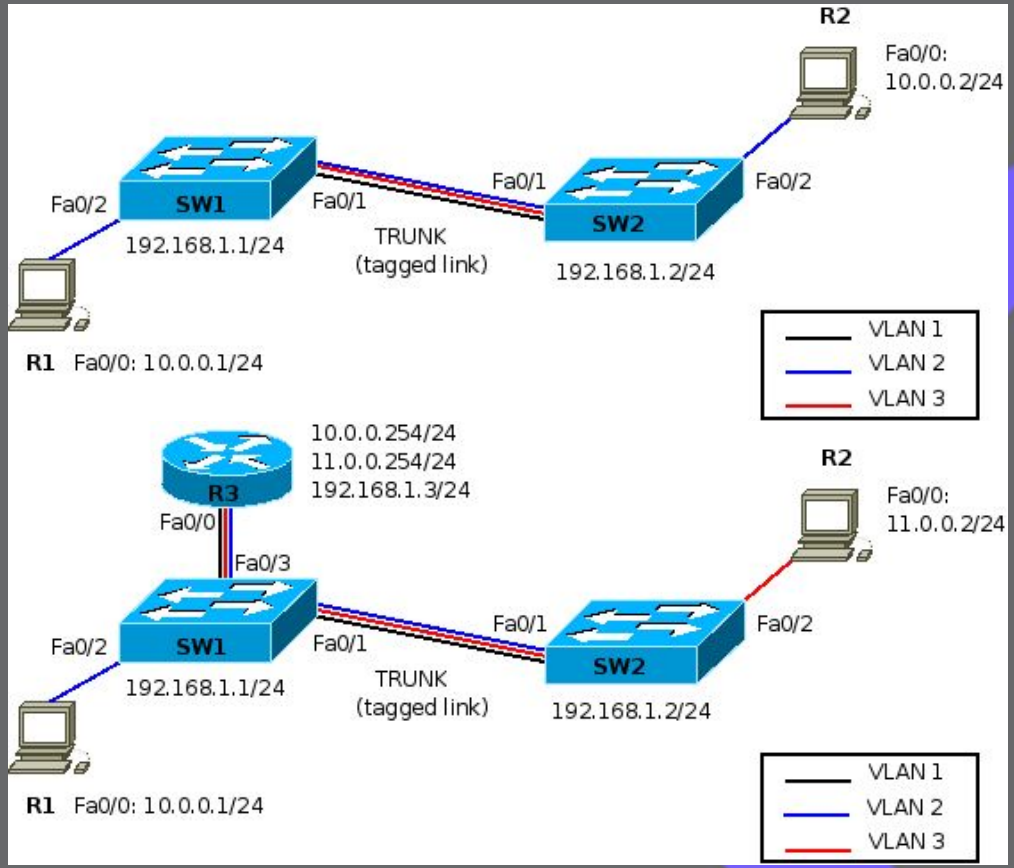


VLANs (tagging)
Puertos especiales de trunking, taggeados.
Puertos TRUNKs



VLANs

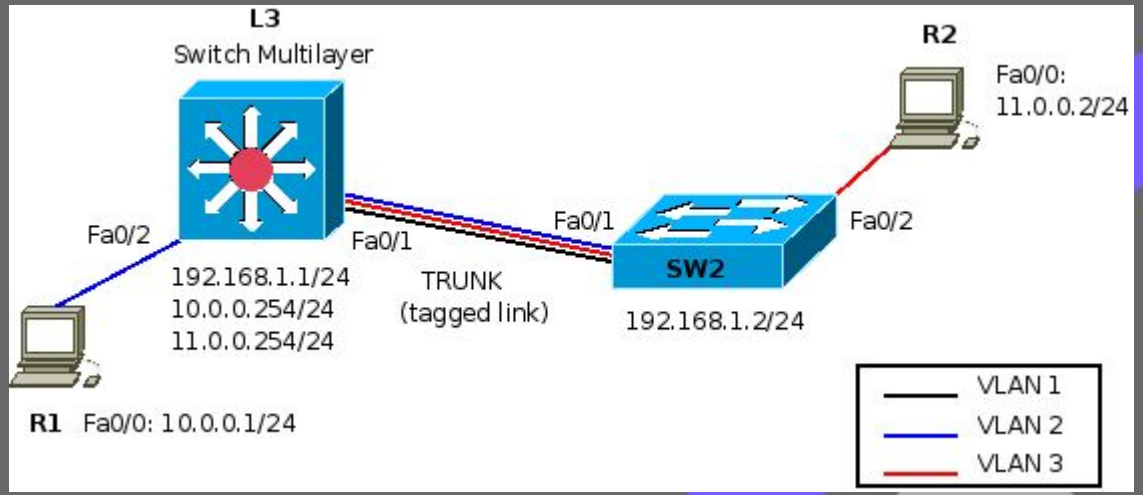
Modelo
“router on a stick”
para interconectar VLANs.



VLANs

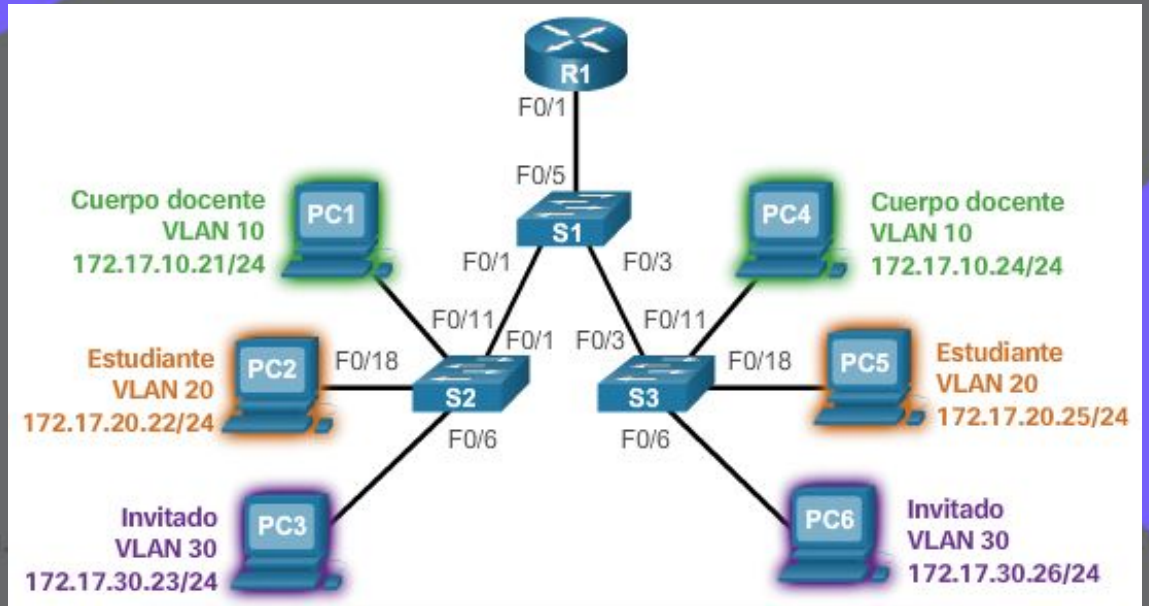
Modelo Switch Multicapa

Dispositivos L2/L3, switches-routers integrados. Mayor eficiencia.



VLANs

Cada VLAN corresponde a un dominio de broadcast, por lo tanto, cada VLAN se corresponderá con una SUBRED.
Las subredes de VLANs no son más que subredes normales, se les aplican todas las técnicas de subredes que vimos en la clase anterior: subnetting, vlsm, etc.



VLANs

Para interconectar VLANs, se necesita resolver la conectividad a nivel de Capa 3, por lo que necesito equipos que sepan “hablar” en capa 3. Es por eso que hablamos que para interconectar VLANs necesitamos un Router (o algún equipo que tenga la capacidad de realizar RUTEO).

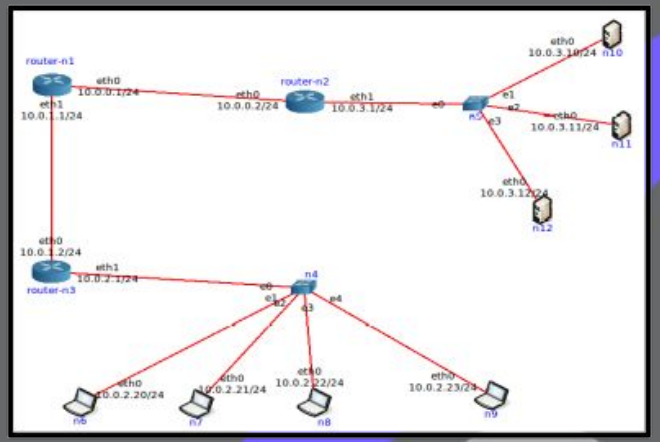
El equipo que interconecta las vlans hace uso de su tabla de rutas y deberá tener una interfaz, ya sea física o virtual (una interfaz de tipo VLAN), en cada vlan que quiera interconectar.

Ruteo

Redes, direcciones y ruteo

Todos los dispositivos en la red tienen tablas de rutas que les dicen qué camino deben tomar los paquetes que les llegan:

Los routers son los encargados de encaminar los paquetes entre distintas redes.

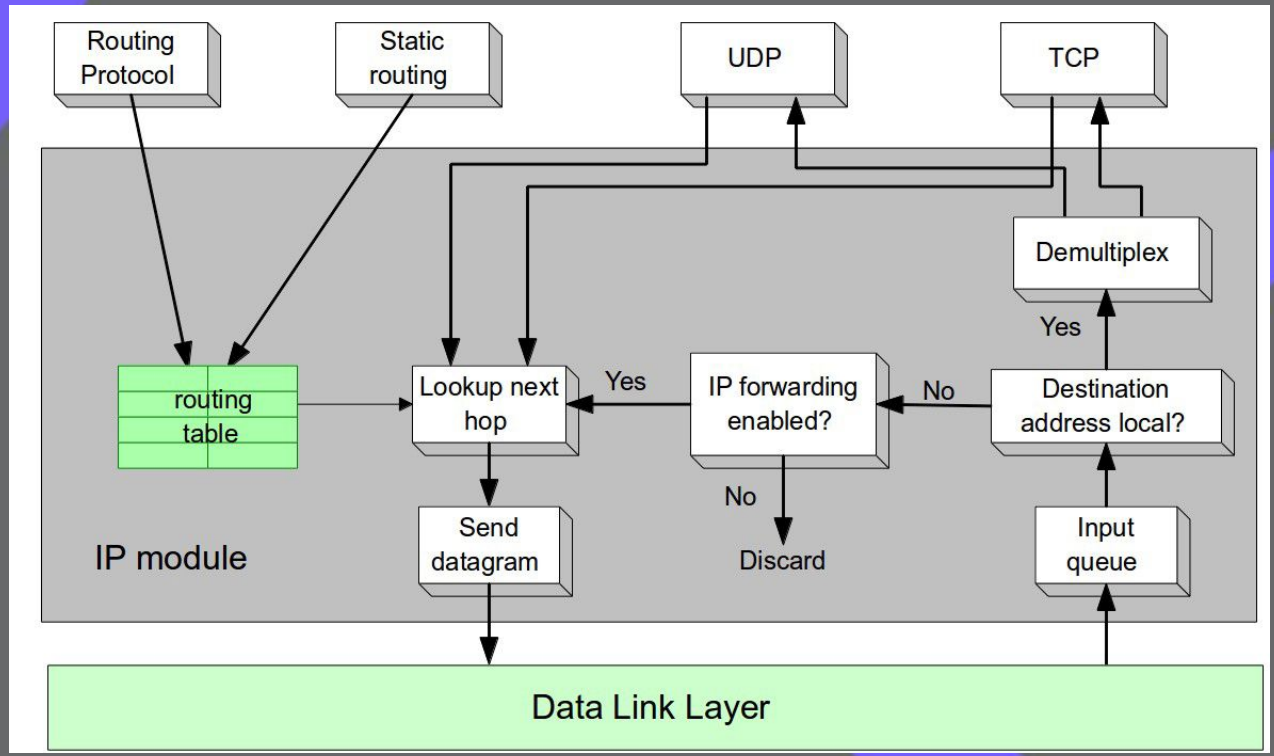



```
IPv4 Tabla de enrutamiento
===========================================================================
Rutas activas:
Destino de red        Máscara de red   Puerta de enlace   Interfaz  Métrica
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.81     25
        127.0.0.0        255.0.0.0      En vínculo       127.0.0.1    331
        127.0.0.1  255.255.255.255      En vínculo       127.0.0.1    331
  127.255.255.255  255.255.255.255      En vínculo       127.0.0.1    331
      192.168.0.0    255.255.255.0      En vínculo    192.168.0.81    281
     192.168.0.81  255.255.255.255      En vínculo    192.168.0.81    281
    192.168.0.255  255.255.255.255      En vínculo    192.168.0.81    281
        224.0.0.0        240.0.0.0      En vínculo       127.0.0.1    331
        224.0.0.0        240.0.0.0      En vínculo    192.168.0.81    281
  255.255.255.255  255.255.255.255      En vínculo       127.0.0.1    331
  255.255.255.255  255.255.255.255      En vínculo    192.168.0.81    281
===========================================================================
```

```
router n3$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        10.0.1.1        255.255.255.0   UG    20     0        0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
10.0.3.0        10.0.1.1        255.255.255.0   UG    30     0        0 eth0
router n3$
```

Ruteo

Los Routers tienen el forwarding habilitado, los hosts no.



Habilitar forward en Linux:

sysctl net.ipv4.ip_forward
sudo sysctl net.ipv4.ip_forward=1

Ruteo

Definiciones:

Tabla de ruteo: estructura en hosts y routers (gateways) que indica cómo despachar un mensaje. Perspectiva del vecino, siguiente salto.

Host: no despacha mensajes que recibe que no son para él. Despacha solo sus mensajes mirando su tabla de ruteo.

Router: Nodos intermedios, más de una interfaz, despacha mensajes mirando tabla de ruteo, desde cualquier interfaz.

Host multihome: tiene varias interfaces, no rutea.

Ruteo

```
router n3$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        10.0.1.1        255.255.255.0   UG    20     0        0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
10.0.3.0        10.0.1.1        255.255.255.0   UG    30     0        0 eth0
router n3$
```

Redes destino

Próximo salto

Máscara de red

Interfaz de salida del equipo

Ruteo

```
router n3$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        10.0.1.1        255.255.255.0   UG    20     0        0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
10.0.3.0        10.0.1.1        255.255.255.0   UG    30     0        0 eth0
router n3$
```

El Gateway representa el siguiente salto, en el camino, que el mensaje debe tomar para continuar hacia su destino final.

En el ejemplo vemos que la **tabla de ruteo** del **router n3**, para enviar un mensaje a **10.0.0.0**, tiene que encaminarlo hacia el equipo identificado como **10.0.1.1**

¿Cómo selecciona un router la ruta correcta?

Los routers usan la máscara de red para calcular a qué red pertenece una dirección IP

¿Por dónde envío un paquete a 10.0.3.56?

```
router n3$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        10.0.1.1        255.255.255.0   UG    20     0        0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
10.0.3.0        10.0.1.1        255.255.255.0   UG    30     0        0 eth0
router n3$
```

AND	10	0	3	56	← Dirección IP a consultar
	255	255	255	0	← Máscara de red
	10	**0**	**3**	**0**	← Dirección de red

Se usa la operación lógica AND a nivel binario

¿Cómo selecciona un router la ruta correcta?

AND	10	0	3	56	← Dirección IP a consultar
	255	255	255	0	← Máscara de red
	10	**0**	**3**	**0**	← Dirección de red

1 AND 1 = 1
1 AND 0 = 0
0 AND 0 = 0
0 AND 1 = 0

	00001010	00000000	00000011	00111000
AND	11111111	11111111	11111111	00000000
	00001010	00000000	00000011	00000000

AND lógico a nivel binario

¿Cómo selecciona un router la ruta correcta?

Se prueba **1 por 1 cada red y máscara** de la tabla de ruteo para saber con qué red coincide la dirección.
Esto se hace en orden: de máscara más específica a menos específica (es decir, de máscara con más cantidad de 1 a máscara con menos cantidad de 1).

163.10.10.0/24	255.255.255.0	11111111.11111111.11111111.00000000	**3ra**
163.10.48.0/26	255.255.255.192	11111111.11111111.11111111.11000000	**1ra MÁS ESPECÍFICA**
163.10.10.127/25	255.255.255.128	11111111.11111111.11111111.10000000	**2da**

Hagamos unos ejercicios...

¿Por dónde envío un paquete a 10.0.6.127?

¿Por dónde envío un paquete a 10.0.0.2?

```
A#show ip route
Codes: K - kernel route, C - connected, S - static, R -
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued rou

C>* 10.0.0.0/24 is directly connected, eth0, 00:00:22
C>* 10.0.1.0/24 is directly connected, eth1, 00:00:22
C>* 10.0.2.0/24 is directly connected, eth2, 00:00:22
R>* 10.0.3.0/24 [120/2] via 10.0.1.2, eth1, 00:00:21
R>* 10.0.4.0/24 [120/2] via 10.0.2.2, eth2, 00:00:21
C>* 10.0.5.0/24 is directly connected, eth3, 00:00:22
R>* 10.0.6.0/24 [120/2] via 10.0.5.2, eth3, 00:00:21
R>* 10.0.7.0/24 [120/3] via 10.0.2.2, eth2, 00:00:21
R>* 10.0.8.0/24 [120/3] via 10.0.5.2, eth3, 00:00:16
A#
```

Solución!

En la tabla anterior, solo hay redes /24 (255.255.255.0), por lo que ésta será la única máscara con la que se realizará el AND:

Expreso 10.0.6.127 en binario -->	00001010	00000000	00000110	01111111
Pruebo con la mascara 255.255.255.0 -->	11111111	11111111	11111111	00000000
Red Resultante --> **10.0.6.0**	00001010	00000000	00000110	00000000

Expreso 10.0.0.2 en binario -->	00001010	00000000	00000000	00000010
Pruebo con la mascara 255.255.255.0 -->	11111111	11111111	11111111	00000000
Red Resultante --> **10.0.0.0**	00001010	00000000	00000000	00000000

Hagamos unos ejercicios...

¿Por dónde envío un paquete a 10.0.6.126?

¿Por dónde envío un paquete a 10.0.6.130?

```
A# show ip route
Codes: K - kernel route, C - connected, S - static, R -
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHR
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued ro

C>* 10.0.0.0/24 is directly connected, eth0, 00:00:16
C>* 10.0.1.0/24 is directly connected, eth1, 00:00:16
C>* 10.0.2.0/24 is directly connected, eth2, 00:00:16
R>* 10.0.3.0/24 [120/2] via 10.0.1.2, eth1, 00:00:15
R>* 10.0.4.0/24 [120/2] via 10.0.2.2, eth2, 00:00:15
C>* 10.0.5.0/24 is directly connected, eth3, 00:00:16
R>* 10.0.6.0/25 [120/2] via 10.0.5.2, eth3, 00:00:15
R>* 10.0.6.128/25 [120/3] via 10.0.5.2, eth3, 00:00:12
R>* 10.0.7.0/24 [120/3] via 10.0.2.2, eth2, 00:00:15
```

Solución!

En la tabla, ahora hay redes con máscaras /25 (255.255.255.128) y /24 (255.255.255.0). Las máscaras /25 son más específicas, entonces empiezo probando con estas:

Expreso 10.0.6.126 en binario -->	00001010	00000000	00000110	01111110
Pruebo con la mascara 255.255.255.128 -->	11111111	11111111	11111111	00000000
Red Resultante --> **10.0.6.0**	00001010	00000000	00000110	00000000

Expreso 10.0.6.130 en binario -->	00001010	00000000	00000110	10000010
Pruebo con la mascara 255.255.255.128 -->	11111111	11111111	11111111	10000000
Red Resultante --> **10.0.6.128**	00001010	00000000	00000110	10000000

Ruteo

¿Cómo se llenan estas tablas de información?

Existen básicamente 2 opciones:

- **Manualmente**
  - Se usan las denominadas **rutas estáticas**, configuradas a mano en todos los equipos por parte de los administradores.

- **Protocolos de ruteo**
  - Un protocolo de ruteo es un programa distribuido que se encarga de que todos los participantes intercambien información de ruteo. Ejemplos: OSPF. Corren sobre IP.

Generalmente, se usa una combinación de ambos enfoques.

NAT

NAT (Network Address Translation)

Nuestro proveedor de Internet, por cuestión de costos (y de disponibilidad) brinda a cada uno de sus clientes 1 dirección IP pública para que éste pueda acceder a Internet.

Esta dirección IP es la que terminamos utilizando cuando navegamos en Internet.

Pero si yo en mi casa tengo varias PC, una smart TV, una tablet y 2 teléfonos, entonces, **¿cómo hacemos para poder tener todos Internet a la vez?**

Ésto se soluciona utilizando una técnica llamada NAT (Network Address Translation)

NAT (Network Address Translation)

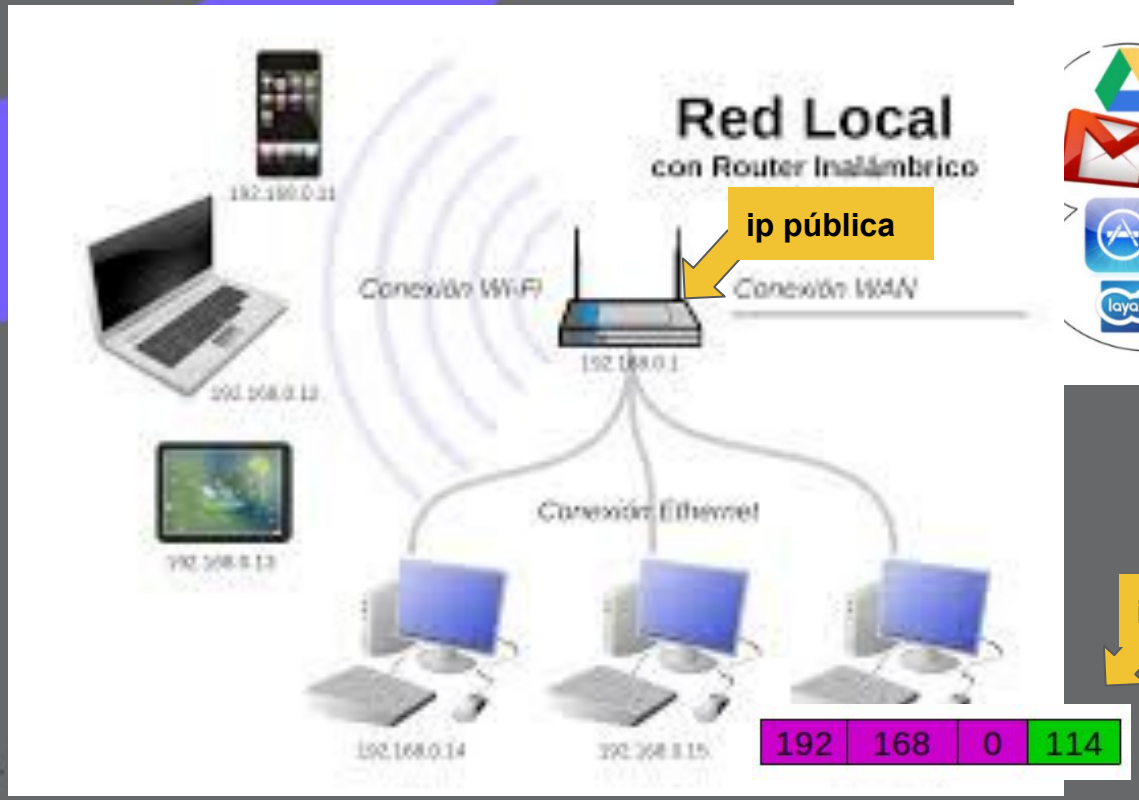
Nuestro Router-AP está conectado a 2 redes:

- Internet, usando la única dirección IP pública.
- Nuestra LAN, en la cual se utilizan direcciones privadas.

Nuestro **Router-AP**, es el que convierte las **direcciones privadas** de nuestra LAN hogareña a la **dirección pública** asignada.

NAT

- A cada paquete que sale, se le cambia la IP de origen por la ip_publica
- A cada respuesta recibida, se le cambia la IP destino por la de la PC que hizo el requerimiento



INTERNET

ip_privada_de_una_PC

NAT (Network Address Translation)

Problemas con IPv4:

- IPv4 tiene el espacio de direcciones “casi” agotado.

Soluciones temporales:

- CIDR: Tablas de ruteo.
- DHCP: direcciones escasas, facilidad de administración.
- NAT: direcciones escasas.

Solución definitiva:

- IPv6

NAT (Network Address Translation)

Traslación de direcciones de un espacio privado (no “enrutable” en Internet) a un espacio público.

Direcciones Privadas: RFC-1918:

- 1 Clase A: 10.0.0.0/8
- 16 Clases B: 172.16.0.0/12.
- 256 Clases C: 192.168.0.0/16.

Proceso definido en RFC-3022, hace obsoleta a RFC-1631.

NAT (Network Address Translation)

Una forma de realizarlo es: “one-to-one” (uno a uno), NAT básico:

- Se mapea una dirección IPv4 privada a una dirección IPv4 pública.
- Si se hace de forma estática requiere tantas direcciones públicas como privadas.
- Permite acceso en ambas direcciones.
- Si se hace de forma dinámica no es necesario, pero sí se requiere un timer por cada entrada. Limita el acceso simultáneo de acuerdo al pool de direcciones públicas.

NAPT (Network Address Port Translation)

NAT no es implementable cuando se tiene un pool chico de direcciones públicas o no se poseen direcciones públicas asignadas(caso hogareño).

En ese caso se debe trabajar con campos de la capa de transporte o del payload, es decir, los números de puerto.

NAPT es conocido como PAT (Port Address Translation): “one-to-many”.

Se utilizan los números de puertos de los protocolos para resolver el mapeo. Se pueden usar timers y sesión del protocolo.

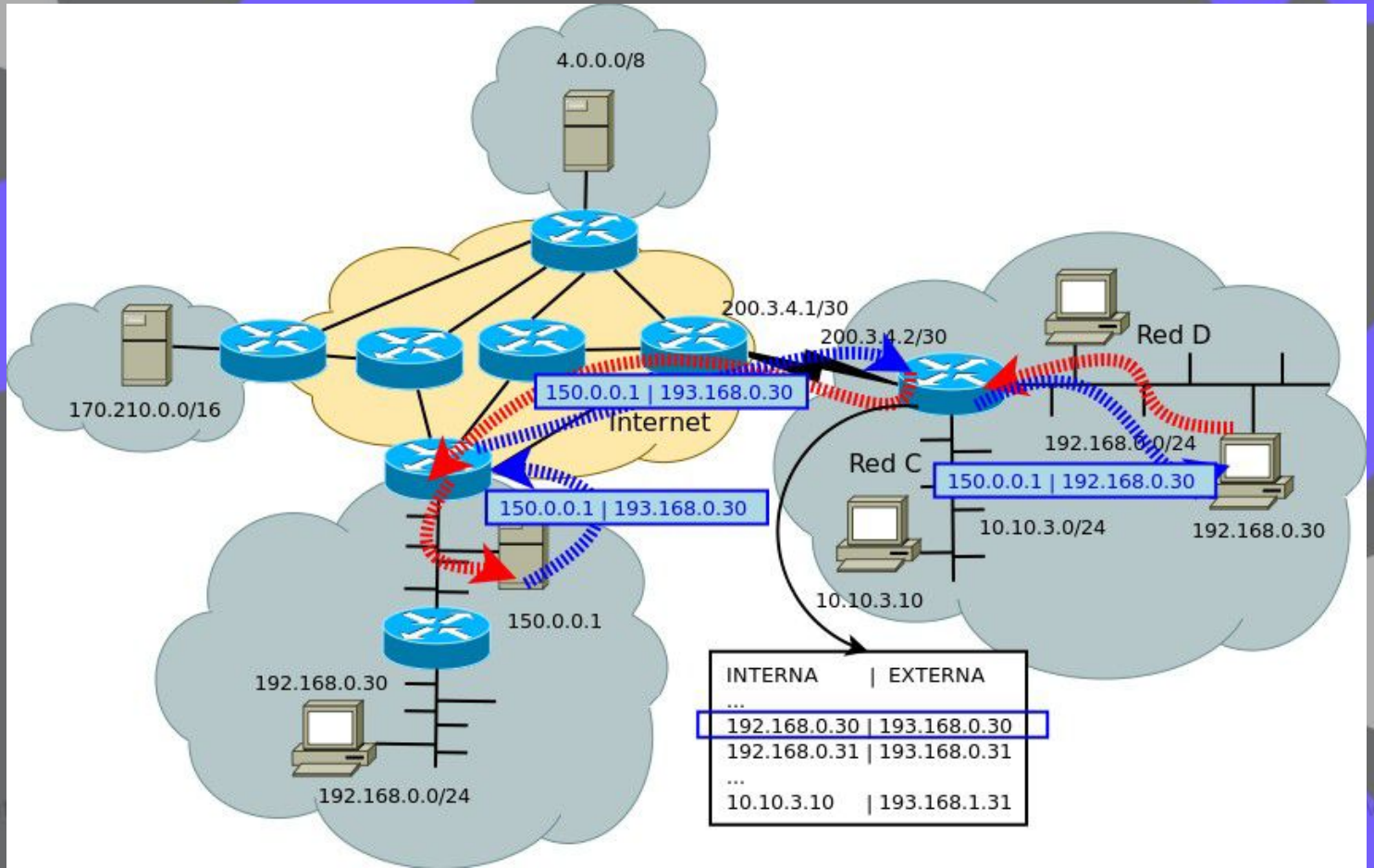
NAPT (Network Address Port Translation)

En la tabla de traslaciones se mantienen el protocolo y los puertos origen y destino.
Se intenta conservar el puerto origen, pero si está “ocupado” se debe reemplazar por otro.
El dispositivo debe “violar” los límites impuestos por la división en capas.

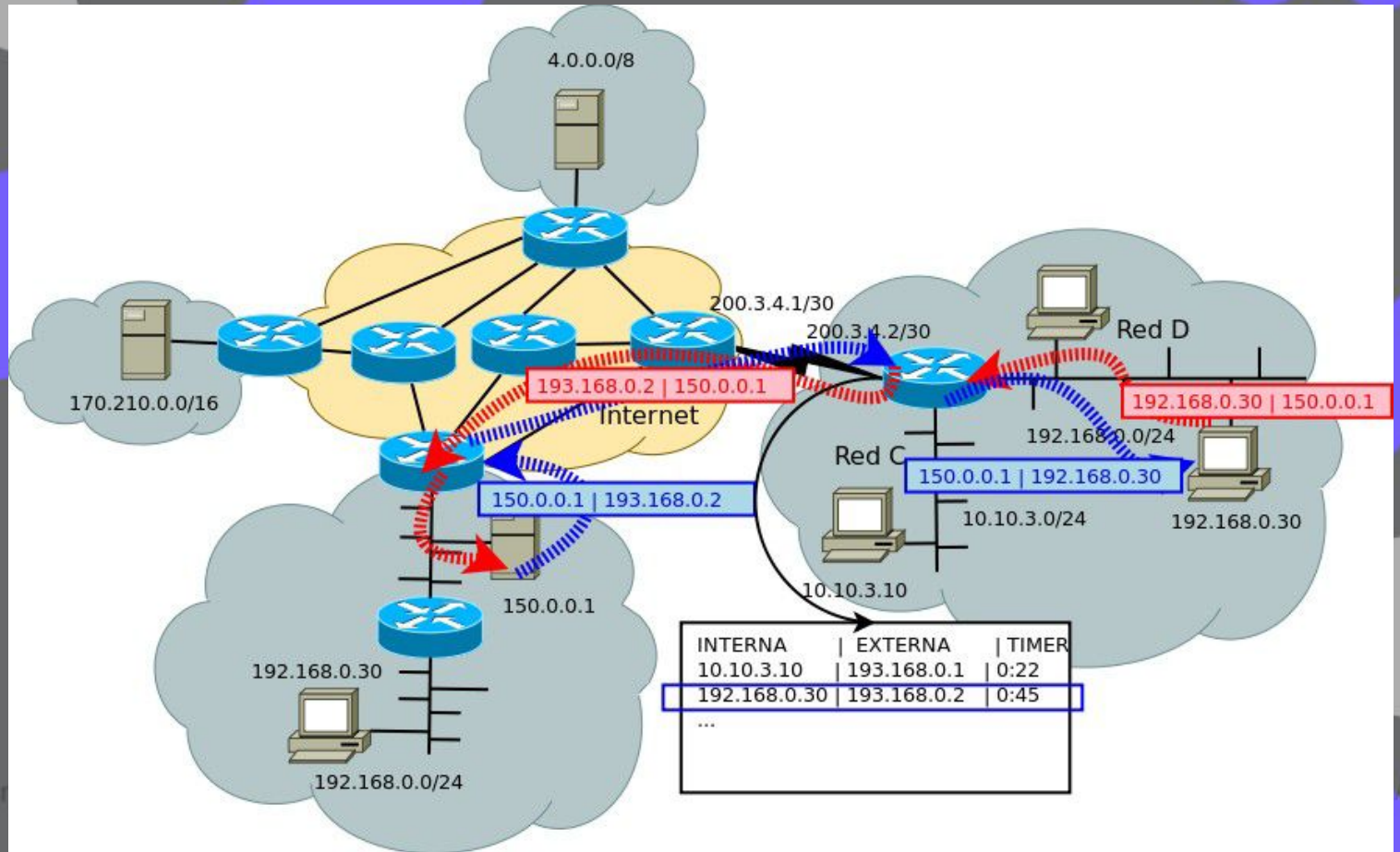
Dos alternativas:

- Utilizando un pool de direcciones y haciendo PAT sobre este.
- Utilizando la dir. IP externa y haciendo overloading/masquerading sobre esta.

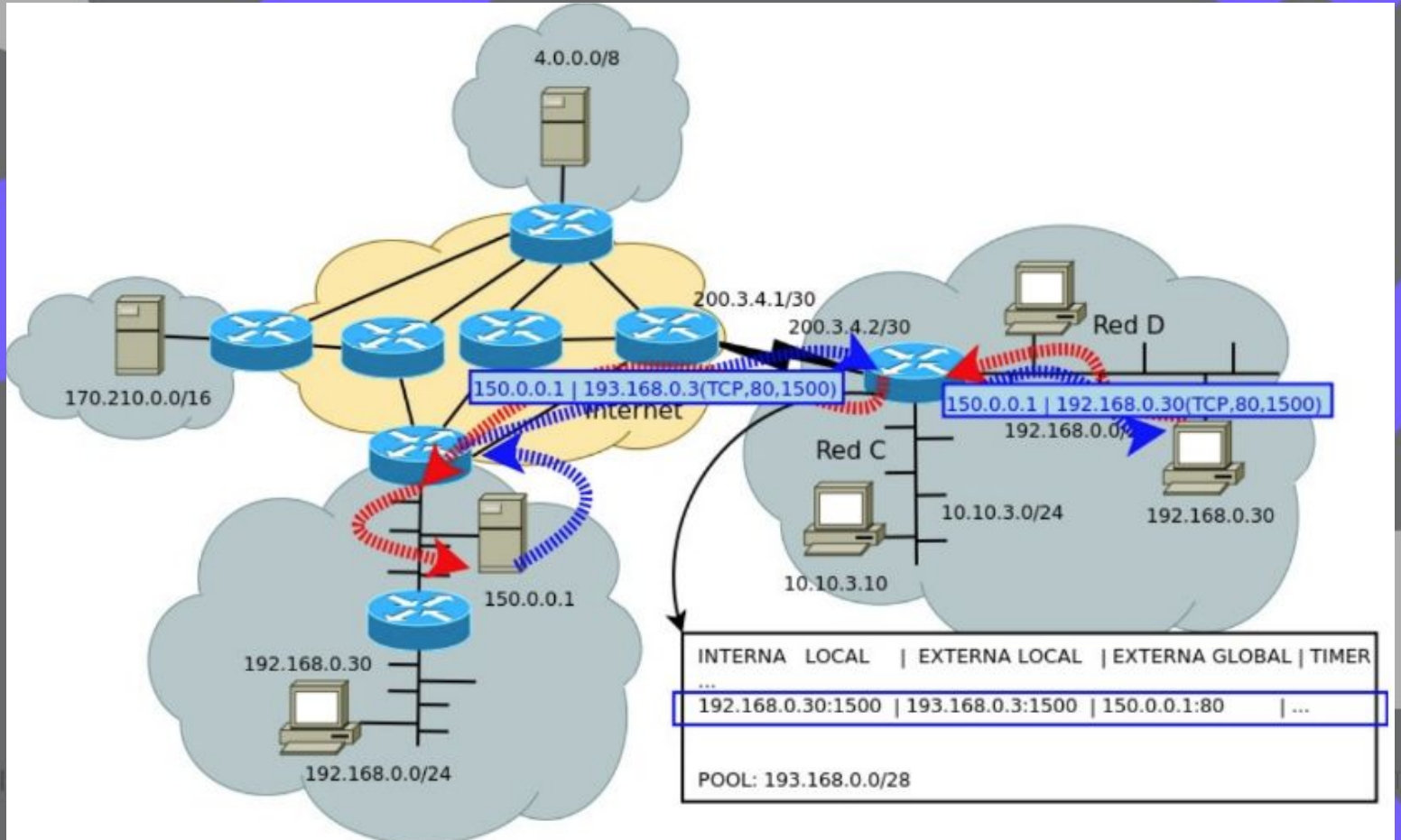
NAT Estatico - 1-to-1



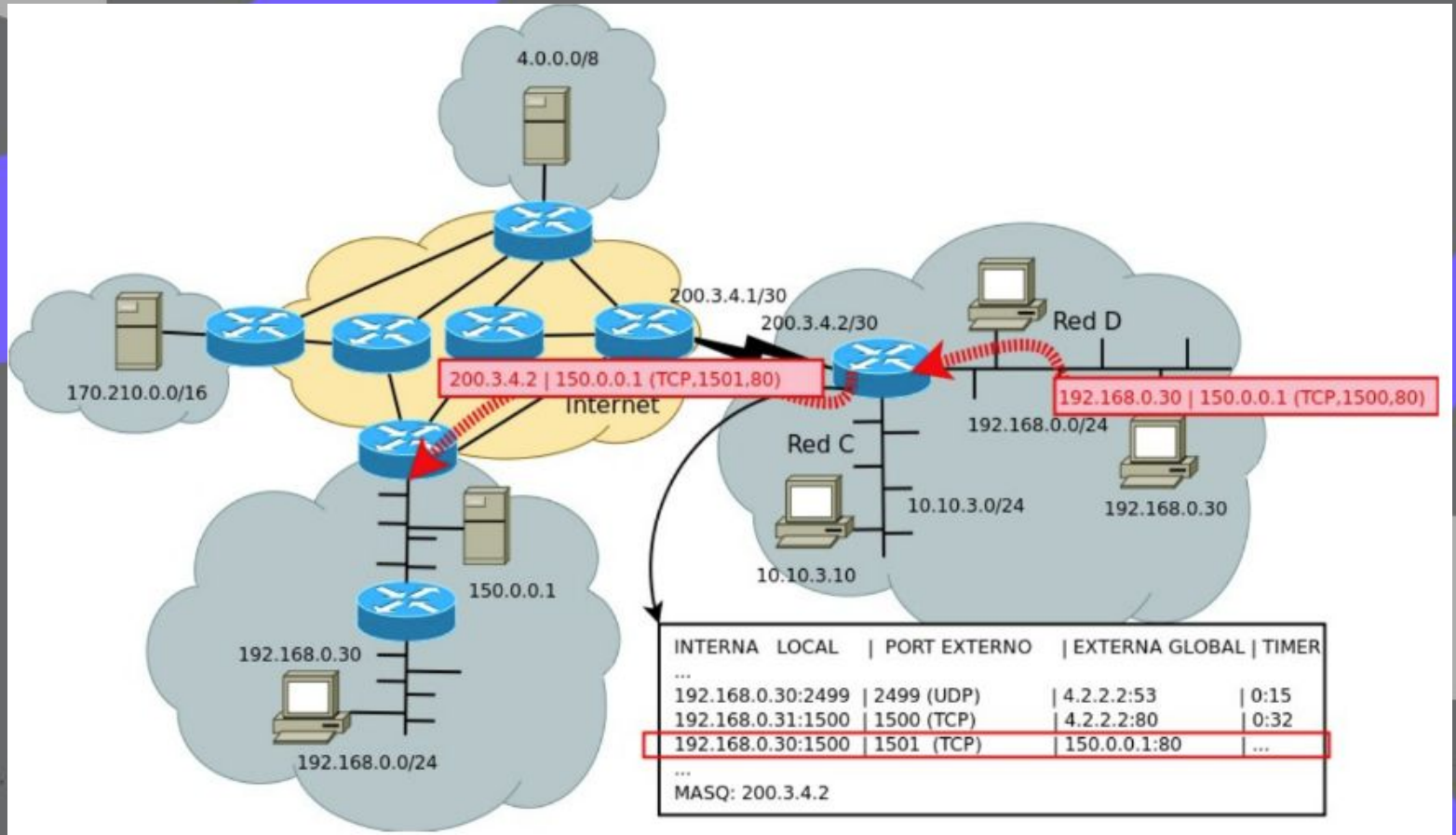
NAT Dinámico - n-to-n (Pool de direcciones)



NAPT - Pool de direcciones



NAPT - Overloading/Masquerade

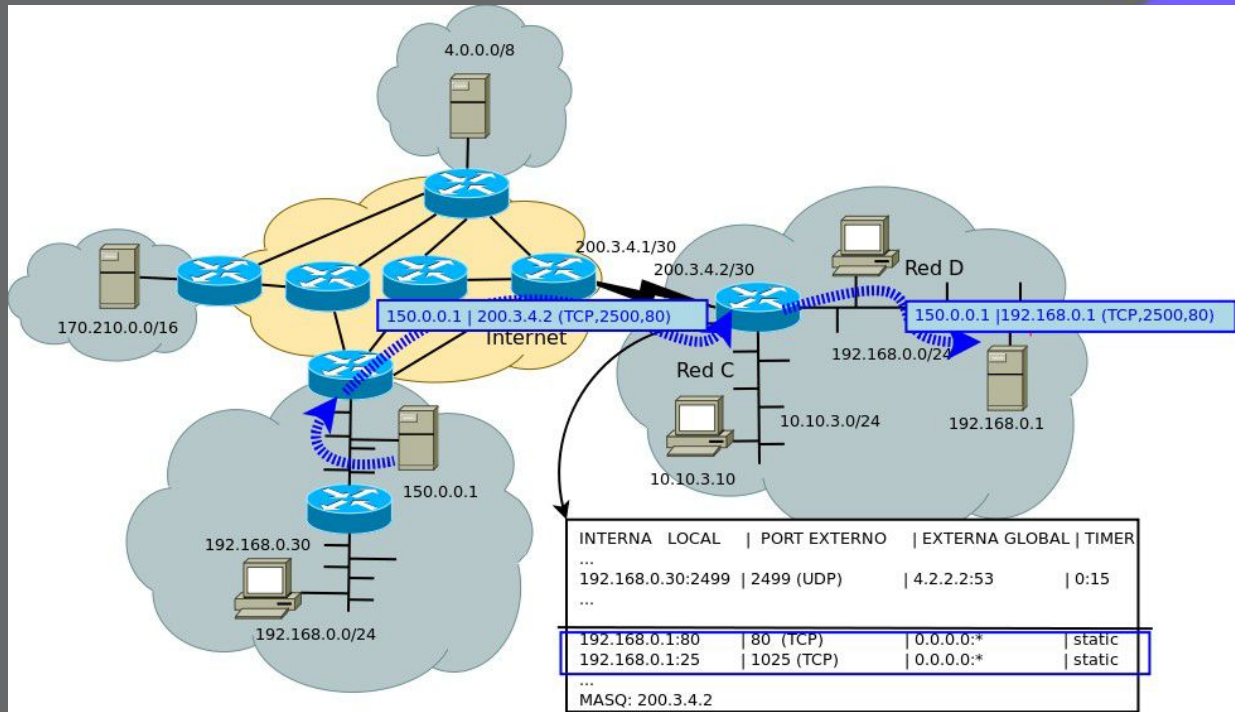


Port Forwarding

Overloading/Masquerade no permiten acceso desde “afuera” hacia “adentro”, solo se permiten entrar tráfico de conexiones generadas internamente.
Mediante Port Forwarding (Reenvío de puerto) se permite acceder a servicios en una red privada desde “afuera”.

No se requiere NAT estático, se implementa con NAPT y mapeo reverso estático de puertos.

Se define a qué dirección y puerto se enviará el tráfico que se reciba en una dirección y puerto del router.



NAT - Consideraciones

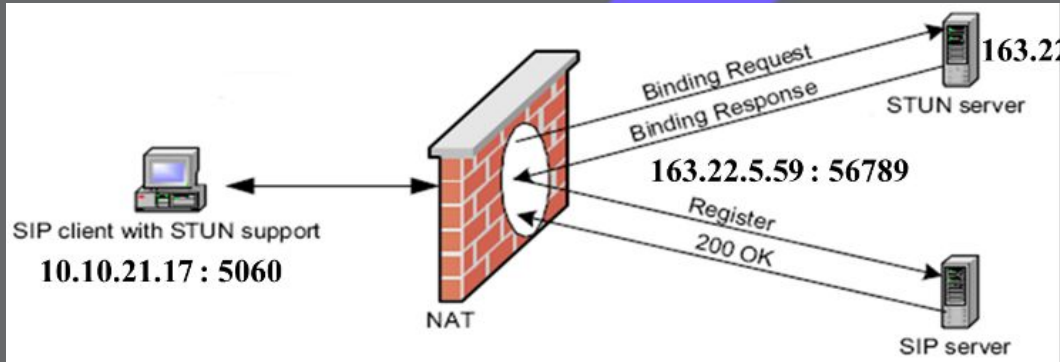
- Carga de procesamiento: Todas las conexiones deben ser traducidas ida y vuelta. Consumo de Recursos extra.
- Logs y Auditoría: Todas las conexiones desde la misma dirección IP.
- “Sensación de Seguridad”: Se suele pensar que al estar la dirección oculta, se está libre de riesgos, por lo que los controles se relajan.
- Funcionamiento de algunos servicios: Algunas aplicaciones no se llevan bien con la técnica de NAT. Un ejemplo clásico es VoIP.
  - Posibles soluciones: NAT Helpers, Protocolo STUN.

STUN - Session Traversal Utilities for NAT

Un servidor STUN permite a los clientes encontrar sus direcciones públicas, el tipo de NAT del cual ellos están atrás y el puerto global asociado por el NAT con el puerto local específico.
Usado principalmente para VoIP y WebRTC.

Un cliente STUN envía una petición a un servidor STUN y este informa al cliente la IP pública usada y qué puerto ha sido abierto por NAT para permitir el tráfico entrante a la red del cliente (realizar el matching de la tabla de nat).
Esta información es usada para configurar la comunicación SIP entre el cliente y el proveedor VOIP.

El protocolo STUN está definido en el RFC 3489. Por defecto utiliza el puerto UDP 3478.



Túneles y VPNs

Túneles

La técnica de tunneling consiste en encapsular un protocolo de red sobre otro (protocolo de red encapsulador) creando un túnel dentro de una red.

El establecimiento de dicho túnel se implementa incluyendo una PDU (unidades de datos de protocolo) determinada dentro de otra PDU con el objetivo de transmitirla desde un extremo al otro del túnel sin que sea necesaria una interpretación intermedia de la PDU encapsulada.
De esta manera se encaminan los paquetes de datos sobre nodos intermedios que son incapaces de ver en claro el contenido de dichos paquetes. El túnel queda definido por los puntos extremos y el protocolo de comunicación empleado

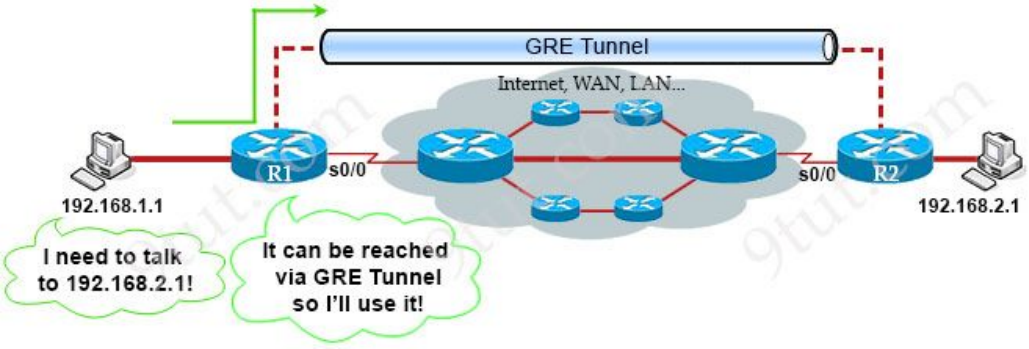
Mecanismos Usuales

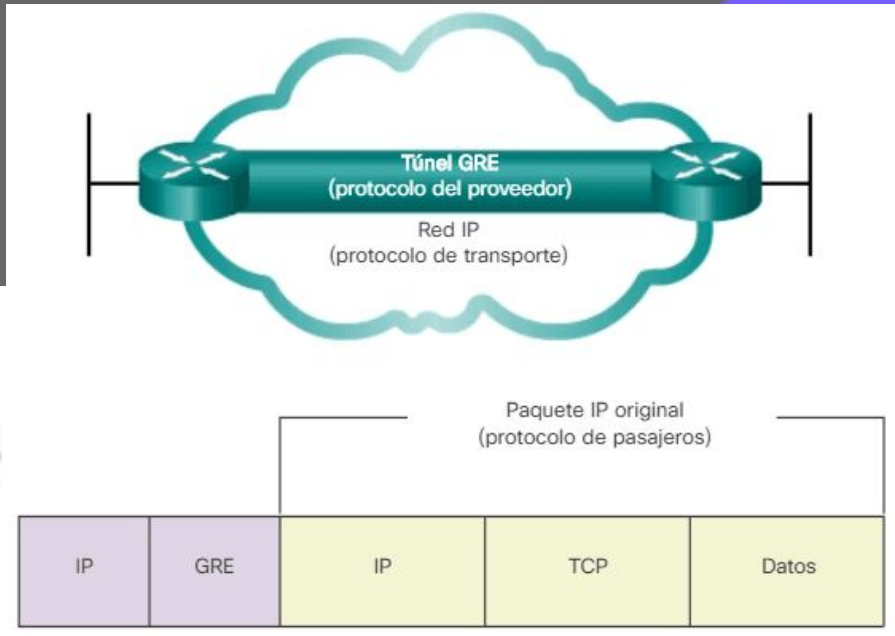
- IP/IP. RFC 3168
- Generic Routing Encapsulation, GRE. RFC 1701
- Layer 2 Tunneling Protocol, L2TP. RFC 2661
- IPsec. RFC 4301

Túneles - GRE

La encapsulación de routing genérico (GRE) es un ejemplo de un protocolo de tunneling de sitio a sitio básico y no seguro.
GRE es un protocolo de tunneling que fue desarrollado por Cisco para encapsular una amplia variedad de tipos de paquete de protocolo dentro de túneles IP, lo que crea un enlace punto a punto virtual entre los routers en puntos remotos sobre IP.

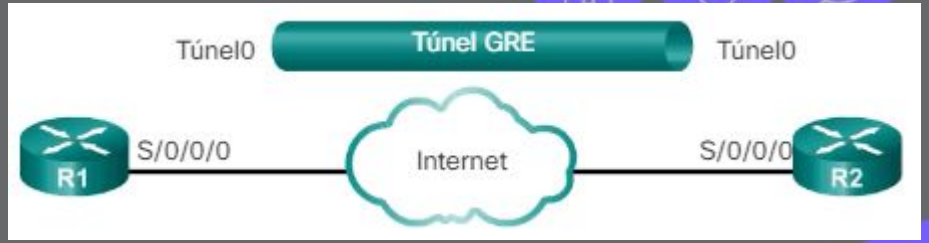
Puede encapsular varios
tipos de paquete de protocolo dentro de un túnel IP.





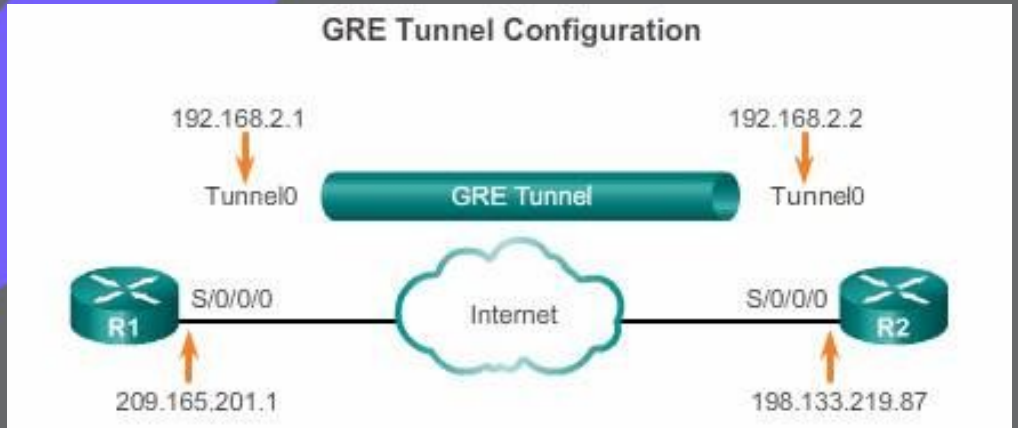
Túneles - GRE

Las características de GRE son las siguientes:



- GRE se define como un estándar IETF (RFC 2784).
- En el encabezado IP externo, se utiliza el número 47 en el campo de protocolo para indicar que lo que sigue es un encabezado GRE.
- La encapsulación de GRE utiliza un campo de tipo de protocolo en el encabezado GRE para admitir la encapsulación de cualquier protocolo de capa 3 del modelo OSI. Los tipos de protocolo se definen en RFC 1700 como «EtherTypes».
- GRE en sí misma no tiene estado; de manera predeterminada, no incluye ningún mecanismo de control de flujo.
- GRE no incluye ningún mecanismo de seguridad sólido para proteger su contenido.
- El encabezado GRE, junto con el encabezado de tunneling IP que se indica en la ilustración, crea por lo menos 24 bytes de sobrecarga adicional para los paquetes que se envían por túnel.

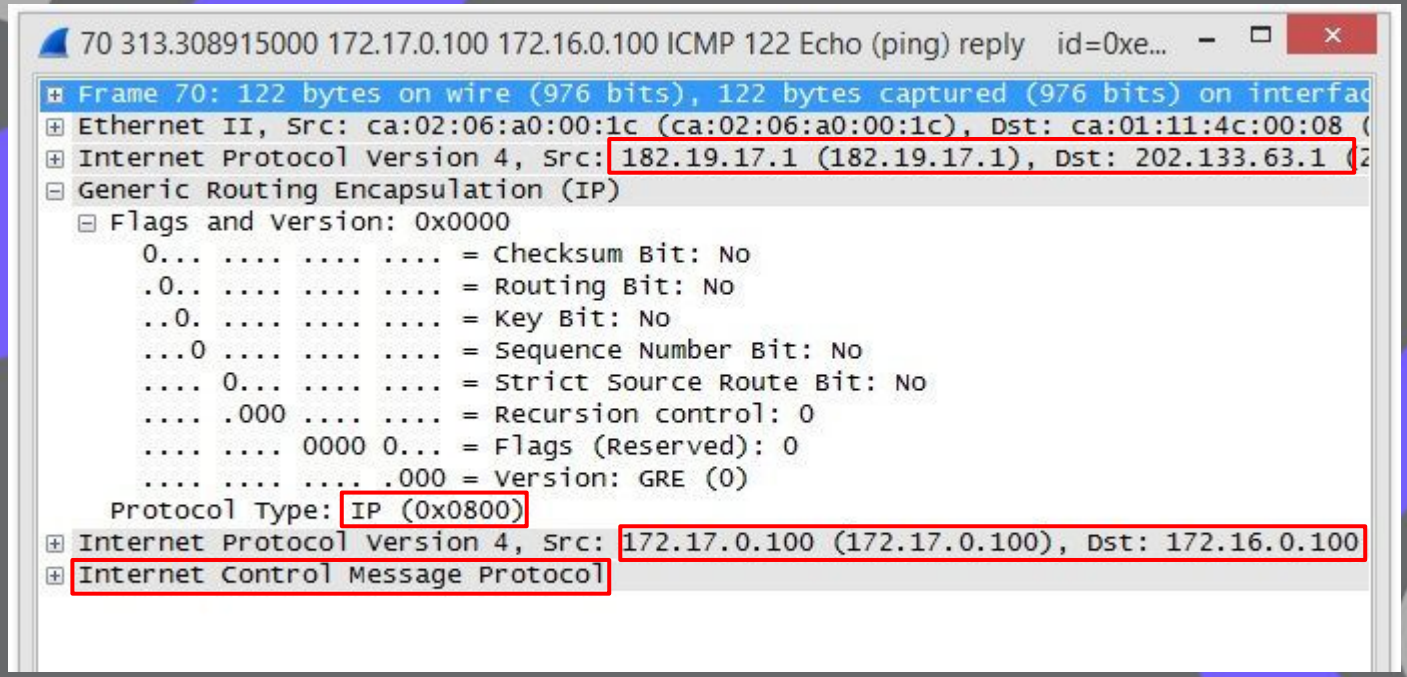
Túneles - GRE



R1 configuration:

```
R1(config)# interface Tunnel0
R1(config-if)# tunnel mode gre ip
R1(config-if)# ip address 192.168.2.1 255.255.255.0
R1(config-if)# tunnel source 209.165.201.1
R1(config-if)# tunnel destination 198.133.219.87
```

Túneles - GRE


70 313.308915000 172.17.0.100 172.16.0.100 ICMP 122 Echo (ping) reply id=0xe...
Frame 70: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interfac
Ethernet II, Src: ca:02:06:a0:00:1c (ca:02:06:a0:00:1c), Dst: ca:01:11:4c:00:08 (
Internet Protocol Version 4, Src: 182.19.17.1 (182.19.17.1), Dst: 202.133.63.1 (2
Generic Routing Encapsulation (IP)
Flags and Version: 0x0000
0... .... = Checksum Bit: No
.0.. .... = Routing Bit: No
..0. .... = Key Bit: No
...0 .... = Sequence Number Bit: No
.... 0... .... = Strict Source Route Bit: No
.... .000 .... = Recursion control: 0
.... 0000 0... = Flags (Reserved): 0
.... .000 = Version: GRE (0)
Protocol Type: IP (0x0800)
Internet Protocol Version 4, Src: 172.17.0.100 (172.17.0.100), Dst: 172.16.0.100
Internet Control Message Protocol

VPNs

Según la RFC 2764:
"... a VPN is simply defined as the emulation of a private Wide Area Network (WAN) facility using IP facilities (including the public Internet, or private IP backbones)."

Una VPN (Virtual Private Network) es una técnica para interconectar o extender una o más redes LAN a través de una red pública o no controlada como internet.

Completa analogía con redes físicas.
VPN no introduce nuevos paradigmas.

VPNs

Pero... ¿Eso no es un túnel?

Si, muchas tecnologías de VPN hacen uso de la estrategia de tunelizado y encapsulamiento, pero se diferencian de ellos en que la estrategia de VPN encripta el tráfico entre los extremos de la conexión, cosa que el tunelizado en sí mismo no realiza.

Ej: IPSEC vpn, SSL VPN, etc.

VPNs

Características

- Transporte transparente
  El tráfico de la VPN no debe relacionarse con el del backbone de IP
  El tráfico puede ser multiprotocolo. Opera por encima del backbone IP. Tunelizado.
  El direccionamiento del cliente es independiente del del backbone (ej IP privadas)

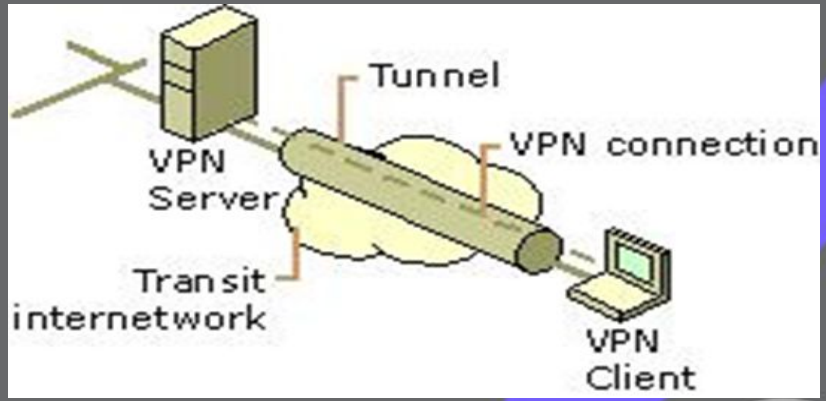
- Seguridad
  El usuario provee su seguridad. La red solo provee el enlace.
  El proveedor del servicio de VPN se encarga de la seguridad.
  Depende del tipo de VPN establecida

- Túneles
  Los dos primeros requerimientos generan su necesidad. Conforma el enlace punto a punto.

VPNs

Elementos:

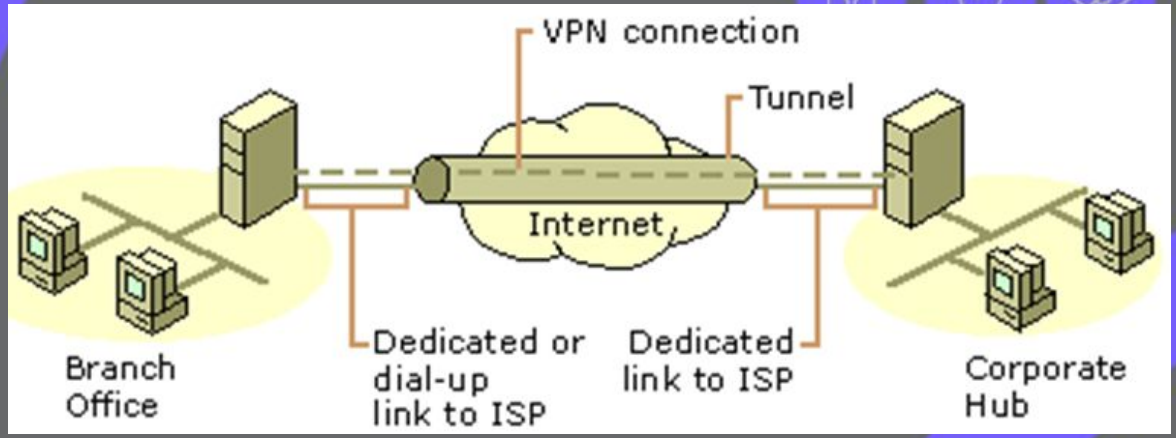
- VPN client: El dispositivo que inicia la conexión VPN a un VPN Server.
- VPN server: El dispositivo que acepta conexiones VPN de clientes.
- Tunel: La etapa de la conexión en la que se encapsulan los datos. Define el protocolo de tunneling (ej PPTP, L2TP)

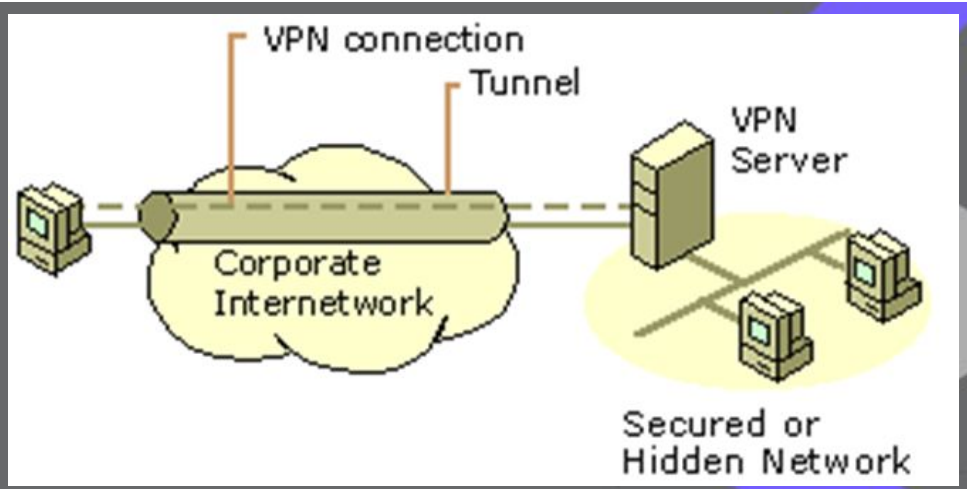


VPNs

- VPN connection. La etapa de la conexión en la que se encriptan los datos. Para conexiones seguras los datos se encriptan y encapsulan en la misma etapa de la conexión. Si bien los datos pueden encapsularse y no encriptarse, en ese caso no estamos en presencia de una VPN.

- Transit internetwork. La red pública o compartida por la que viajan los datos encapsulados





Esto es todo por hoy.

¿Preguntas?

www.linti.unlp.edu.ar

LINTI
Laboratorio de Investigación en
Nuevas Tecnologías Informáticas
Facultad de Informática
UNIVERSIDAD NACIONAL DE LA PLATA
Calle 50 esq. 120, 2do Piso. Tel: +54 221 4223528