

t_b es el tiempo de transmisión de 1 bit
 ↳ V_b es el bitrate (velocidad) $V_b = \frac{1}{t_b}$

OSI → Organización por capas

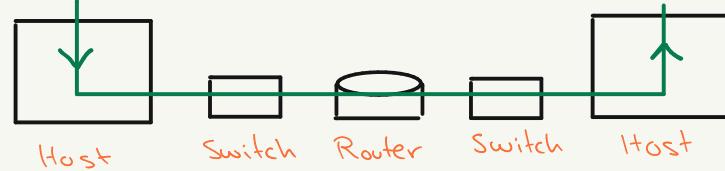
- proporciona servicios a la superior
- recibe servicios de la inferior
- se comunica con sus iguales

TCP → Capas 1 y 2 unidas (12) ← Físico o enlace
 Capas 5-7 unidas (57)

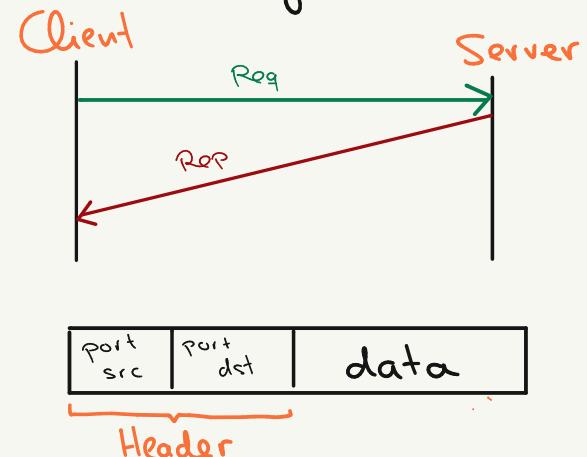
Host → 57 → 4 → 3 → 12

Switch → 12

Router → 3 → 12



Paradigma Cliente - Servidor



Puertos

0-1024 → Well known ports ← Necesitan root para acceder.

80 ← HTTP
 22 ← FTP

1024-2¹⁶ → Puertos efimeros ← No requieren permisos de root.

Latencia ← Tiempo que transcurre entre enviar y recibir

↳ En caso de mail nos da igual ±

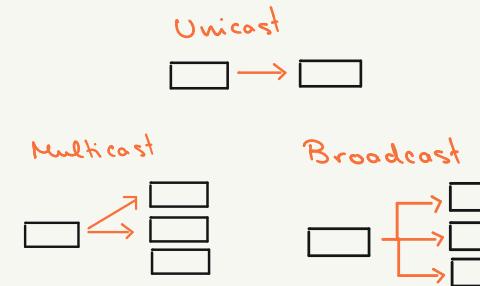
↳ En caso de VJ nos importa

↳ Drive necesita + ancho banda - latencia

Buscamos esto

Formas de comunicarse

- ↳ Unicast ← De un host a otro
- ↳ Multicast ← De un host a varios
- ↳ Broadcast ← De un host a todos



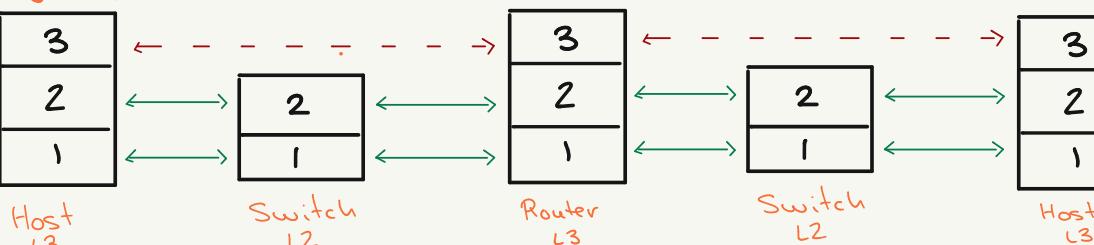
Redes IP

↳ Red de dispositivos de capa 3 (IP)

↳ Una red es la conexión de dispositivos de capa 3

Por lo tanto en modo solo habrá disp. capa ≤ 2

Ejemplo Red IP



Direcciones IP

- Version ← 4 (IPv4)
- IP Header Length ↓ IHL ← Longitud del header en palabras de 32 bits
- Type of Service ← TOS ← Distintos servicios
- Total Length ← Tamaño del datagrama en bytes
- Time to Live ← TTL ← Tiempo restante en circulación del datagrama
- Protocol ← Tipo de protocolo (ICMP, TCP, ...)
- Header Checksum ← Detección de errores
- SRC y DST ← Direcciones Origen y Destino

Time to Live ← Ayuda a conservar ancho de banda

Cuantos disp. L3 pueden enrutar un paquete

Si TTL=0, el paquete se descarta

Checksum (CKS)

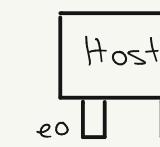
↳ Suma de todo el Header con el CKS=0

Esto lo hacen emisor y receptor para comprobar que no hay errores y el mensaje esté bien.

Dirección IP

net id	host id
--------	---------

El host id marca la interfaz, es decir, se puede tener mas de una por host.



Se expresan con 4 bytes entre puntos

10.10.27.1

Se puede saber de donde → País
 → Proveedor

Clases de Redes (como distribuir los 4 bytes)

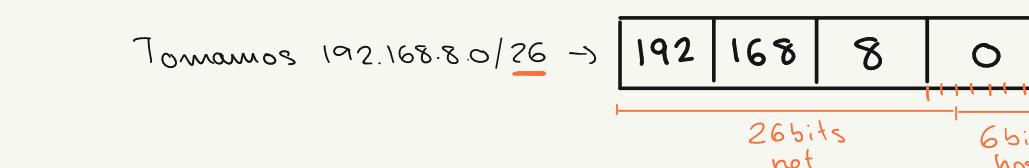
net id	host id	cuantos redos *	cuantos hosts **	range ip
Clase A	1	3	2^8	2^{24} 1-126
Clase B	2	2	2^{16}	2^{16} 128-191
Clase C	3	1	2^{24}	2^8 192-223

Ejercicio ejemplo

Private address block → 192.168.8.0/22

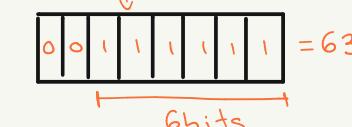
Sub-net X1 → 192.168.8.0/26

a) Cuantas interfaces se pueden tener? Cuál es el rango?



Tomando los 6 bits de host, podemos sacar el numero de direcciones y el rango:

192.168.8.0 ← red
 192.168.8.63 ← broadcast



b) Consigue todas las subredes y clasifícalas.

Subnet id	Prefix	mask	# IP addr	addr router
0000	192.168.8.0	/26	$2^6 - 2 = 62$	192.168.8.1
0001	192.168.8.64	/26	$2^6 - 2 = 62$	192.168.8.65
001X	192.168.8.128	/25	$2^7 - 2 = 126$	192.168.8.129
01XX	192.168.9.0	/24	$2^8 - 2 = 254$	192.168.9.1
1XXX	192.168.10.0	/23	$2^9 - 2 = 510$	192.168.10.1

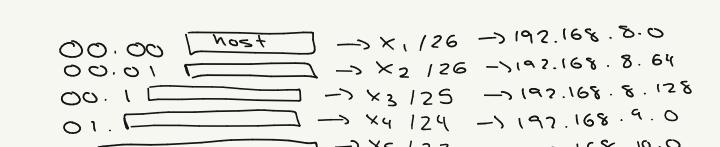
Para conseguir todas las redes de mayor tamaño, hay que partir de la base:

192.168.8.0/22

192.168.[00001000].0 ← Net id no se toca, así que hay que trabajar con los de host

↳ net id → host (10)

Se nos ha pedido empezar por X1(1/26), asigna:



Address Resolution Protocol (ARP)

Se basa en una tabla de IP-MAC para saber a qué dirección va cada uno.
↳ Si no se sabe la MAC, se guarda el dato en un buffer temporal hasta que lo tenemos.

Routing con ARP

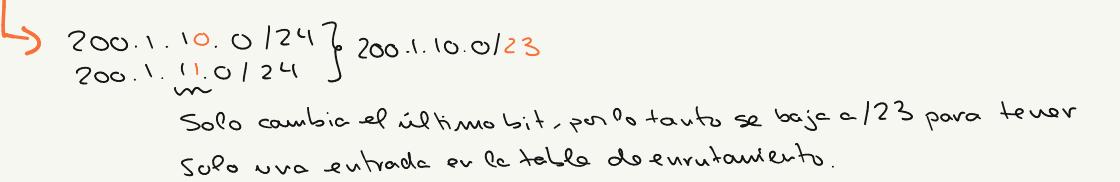
↳ Directo

En caso de enrutamiento directo (misma red), se manda el ARP, resuelve con la dirección de destino.

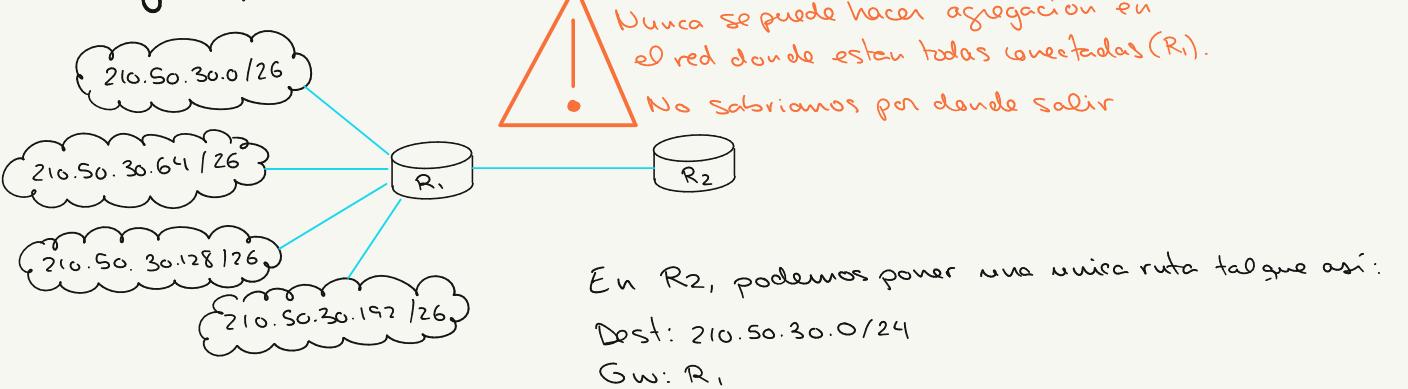
↳ Indirecto

Al no estar en la misma red, mandamos el ARP, resuelve con el gateway.

Agregación y summarización



Ejemplo



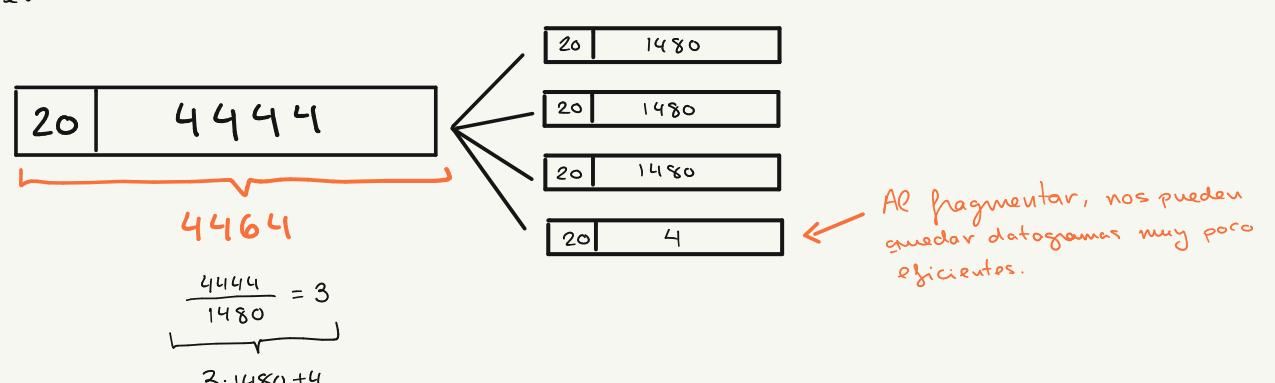
ARP req & rep

Cuando se necesita saber la MAC asociada a una IP, se envía un Broadcast "req" y se espera una respuesta "rep". ↳ Vuelve por Unicast y se añade a la tabla ARP.

Fragmentación IP

En ethernet solo se pueden transmitir 1500 bytes. Yo sabemos que el Header se lleva 20.
↳ Nos quedan 1480 bytes de payload

En TokenRing, el tamaño máximo es 4464. Así que si queremos mandar ese datagrama en Ethernet, habrá que hacer lo siguiente:



ICMP - Internet Control Message Protocol

- ↳ Ping usa ICMP echo request
- ↳ Traceroute usa ICMP

Se usan los errores de ICMP para resolver dudas. Por ejemplo:

- Si no sabemos el MTU necesario para fragmentar, mandamos el datagrama entero y con el bit de DF activado (don't fragment). ↳ ICMP devolverá "Error: fragmentation needed(1500)"

Alí tenemos el MTU que buscábamos

- Para usar tracercoute, se envían datagramas con TTL muy bajos. Por ejemplo, mandar un datagrama con TTL=1 nos muestra el error de "time exceeded" el primer router. El TTL=2, nos devolverá el segundo router, y así sucesivamente.

- Para medir el tiempo que tarda un message en llegar (delay), usamos echo request y contamos hasta recibir el reply (ping).

DNS - Domain Name System

Sirve para aprenderse URLs en lugar de IPs. Existe una base de datos que contiene los parámetros URL-IP.

Usa UDP en el puerto 53

DHCP - Dynamic Host Configuration Protocol

Las configuraciones de host contienen IP, Route default, DNS name, etc.

Sirve para asignar los IP de forma adecuada y dinámica (sin que tengas que hacerlo tú mismo).

Estos cambios pueden ser: Durante tiempo limitado o manualmente.

Manualmente: Se asocian IPs por MACs.

- Se basa en UDP pero se necesita que todos estén en la misma red.

Mensajes:

- Puedo tener + de un router, así →
- Server Discovery: Los clientes mandan broadcast @ IP 255.255.255.255 que + de una offer.
 - Offer: Router que recibe Discovery manda una oferta de IP al host.
 - Request: El host escoge la IP entre sus ofertas.
 - Acceptación: OKAY del router. (ACK)

Puertos

- Puedo tener + de un servidor DHCP.
- Servidores usan el puerto 67
 - Clientes usan el puerto 87

Algoritmos de Routing

El objetivo es añadir entradas a las tablas de enrutamiento.

- ↳ Estática: Manual, scripts, DHCP...
- ↳ Dinámica: Cambios automáticos por Algoritmos de Routing

Internet está organizado en Sistemas Autónomos (grupos de IPs conectados).

↳ Siempre tienen definida una política de enrutamiento

Hay +56000 AS identificados por su AS Number (ASN).

↳ Asignado por la IANA

Routing Algorithms

RA según ASes

- ↳ EGP: Exterior: Entre ASes distintos (BGPv4)
- ↳ Interior: Dentro de una AS (RIP, OSPF)
- ↳ IGP

RIP - Routing Information Protocol

- Matriz: Distancia (saltos) para llegar al destino.

- ↳ 1 es directamente conectado
- ↳ 2 es un salto
- ↳ 16 es infinito (no llega)

- Se envían updates Router-Router cada 30 segundos. Si a los 180s no responde un Router, se considera DOWN.

- RIPv2 permite máscaras variables y usa multicast.

Problemas

↳ Se envía toda la tabla de enrutamiento por Router (too much)

↳ los cambios tardan en verse

Count To Infinity

En este caso, si perdemos R3, R2 dirá que N4 está a 16(inf), pero verá que por R1, puede llegar en 3 saltos (falso) y ahí tenemos bucle porque de R1 volverá a R2.

Split Horizon + Poisoned Reverse

↳ Avisa y elimina de otros Routers las entradas que pasen por él para llegar al inaccesible.

Triggered Updates

Se manda el aviso de Router Redido antes de los 30s de avisos de RIP.

NAT - Network Address Translation

NAT traduce una red privada (o cualquiera) a uno público

interno externo

Se cambia el src en caso de enviar o dst al recibir.

SNAT (Source NAT) se usa para acceder a Internet por parte de un usuario interno.

- ↳ Usa una tabla que tiene las traducciones
- ↳ Necesitamos una IP pública por cada privada.

DNAT (Destination NAT) es NAT pero la conexión la inicia una dirección externa. Por lo tanto, cada servidor tendrá una IP pública. (Esta dirección tiene que conocerse y no cambiar)

Ventajas:

- Reducir el nº de IP públicas a comprar (se pueden reutilizar)
- Aumenta seguridad (escoge quien tiene acceso a Internet)
- Facilidad de red

Entradas tabla NAT

Estáticas: Añadidas manualmente

Dinámicas: Para redes con muchos hosts → Tiene un timeout

Mediante el uso de TCP/UDP, NAT usa también puertos (PAT).

Con PAT, ahora puedo tener 60.000 comunicaciones por cada IP (una por puerto). P.ej.: @192.168.1.10:1022 IP Port

use puertos diferentes (> 1024)

Seguridad

Objetivo →
 Confidentialidad
 Integridad
 Disponibilidad

Al inicio, no se buscó la seguridad, y se han ido añadiendo parches.

Soluciones → Firewall
VPN

Firewall

- Sistema o grupo de sistemas para proteger una red

- App del Router que decide que paquete pasa y cual no. ← Firewall más básico

- Filtro de paquetes mediante TCP/UDP

Se consigue mediante una lista de acceso (ACL)

↳ El IP del datagrama recorre la ACL y ve si puede pasar.

Confidencialidad
Integridad
Disponibilidad

Al inicio, no se buscó la seguridad, y se han ido añadiendo parches.

VPN - Virtual Private Network

VPN se basa en → Autenticación
Cryptografía
Túneles → Node te garantiza bandwidth / delay

Tipos de túneles

- **IPinIP:** Básico
- **GRE:** Generic (header adicional)
- **PPTP:** Point-to-Point
- **IPsec:** Ofrece seguridad

} Nueva cabecera IP.

Al estar añadiendo una cabecera, la nueva se llama **Externa**, y se quita al entrar al túnel

Cabecera Ext → **Src: Inicio túnel**
→ **Dst: Fin túnel** } El túnel crea una nueva interfaz **Tun0**, pero virtual.
Es decir, todo aparte que ve por túnel, pero ivá por estos datos

Autenticación
Cryptografía
Bandwidth / delay

Problemas

- **Fragmentación:** Al ampliar el Header, reduce el tamaño de los datagramas y puede causar fragmentación.
- **ICMP:** Si salte algún error, se devuelve a la @src, que ahora es el túnel.

Autenticación
Cryptografía
Bandwidth / delay

Soluciones

- **Estado túnel:** Estado en que entre el Router que abre un túnel para direccional los errores de ICMP de vuelta al host.
- **Fragmentación:** El túnel fragmenta si es necesario.