



## Fundamentos de Redes

# Tema 2

## Capa de Red

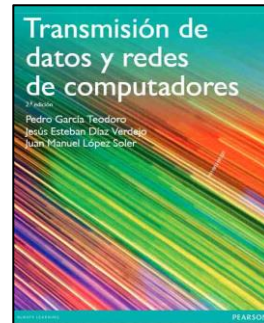
Antonio M. Mora García



# Bibliografía

## Básica

- © P. García-Teodoro, J.E. Díaz-Verdejo, J.M. López-Soler. Transmisión de datos y redes de computadores, 2ª Edición. Editorial Pearson, 2014. **CAPÍTULOS 6 y 9**



## Complementaria

- © James F. Kurose, Keith W. Ross. Redes de computadoras. Un enfoque descendente. 7º Edición. Editorial Pearson S.A., 2017. **CAPÍTULO 4**



# Índice

- © **2.1.** Funcionalidades
- © **2.2.** Conmutación
- © **2.3.** El protocolo IP
- © **2.4.** Asociación con la capa de enlace: El protocolo ARP
- © **2.5.** El protocolo ICMP
- © **2.6.** Cuestiones y ejercicios

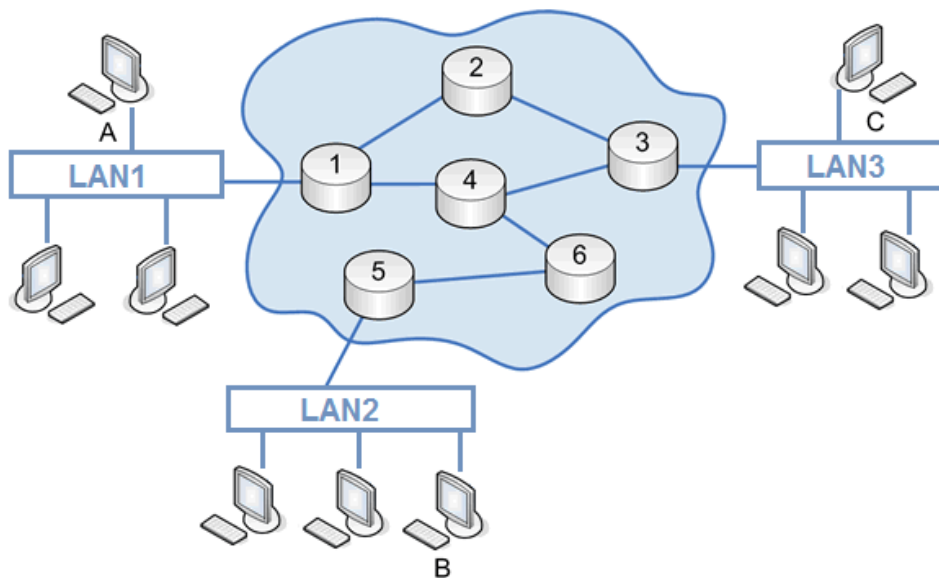
# TEMA 2. Capa de Red

- © **2.1. Funcionalidades**
- © 2.2. Conmutación
- © 2.3. El protocolo IP
- © 2.4. Asociación con la capa de enlace: El protocolo ARP
- © 2.5. El protocolo ICMP
- © 2.6. Cuestiones y ejercicios

# Funcionalidades

## FUNCIONES Y SERVICIOS EN TCP/IP

- El objetivo de la **capa de red** en Internet es la **interconexión de redes**, con independencia de la tecnología subyacente.
- En el modelo OSI el control de congestión se realiza en esta capa.



## EJEMPLOS DE PROTOCOLOS DE RED

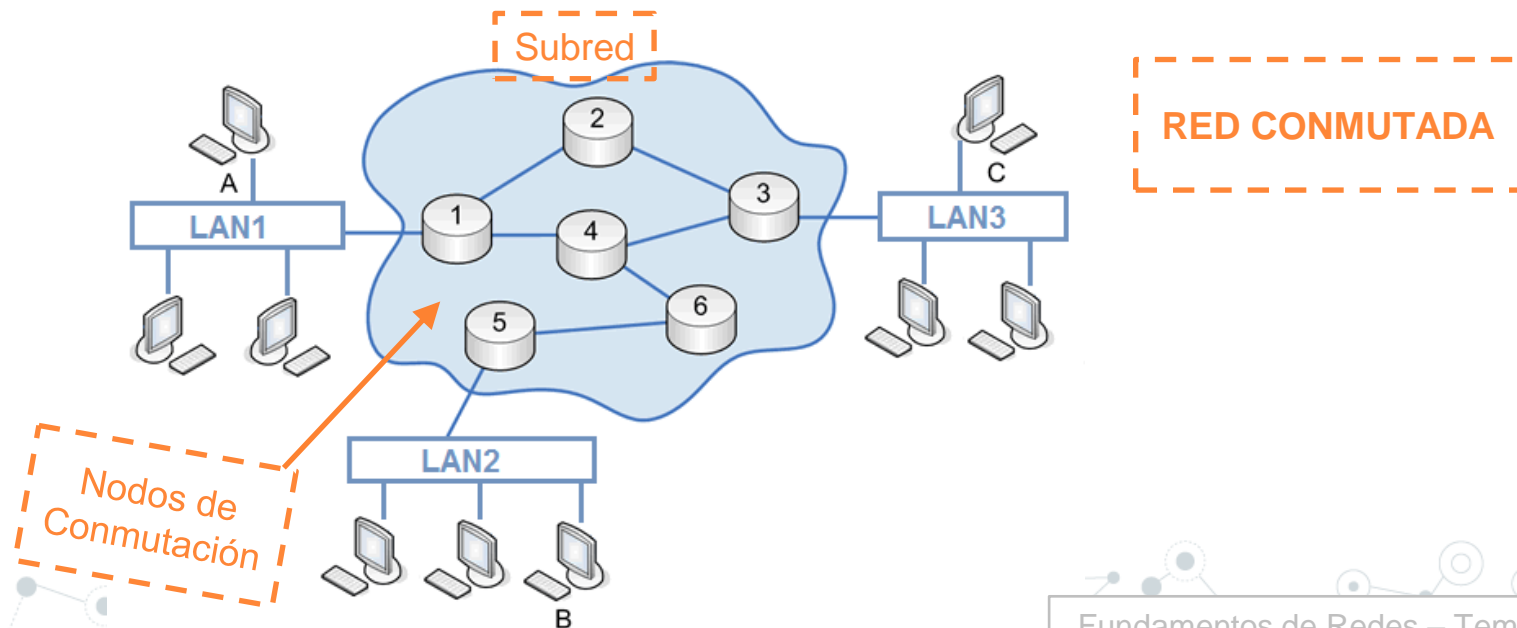
X.25 ([https://es.wikipedia.org/wiki/Norma\\_X.25](https://es.wikipedia.org/wiki/Norma_X.25))

IP

# Funcionalidades

## FUNCIONES Y SERVICIOS EN TCP/IP

- **Conmutación:** acción de cursar tráfico entre los nodos de la red.
- **Encaminamiento (*routing*):** encontrar la mejor ruta desde un origen a un destino.



# Funcionalidades

## FUNCIONES Y SERVICIOS EN TCP/IP

Funciones del protocolo TCP	
En el emisor	<ul style="list-style-type: none"><li>• Divide la información en paquetes</li><li>• Agrega un código detector de errores para comprobar si el paquete llega correctamente a su destino</li><li>• Pasa el paquete al protocolo IP para que gestione su envío</li></ul>
En el receptor	<ul style="list-style-type: none"><li>• Recibir los paquetes que pasa el protocolo IP</li><li>• Ordena los paquetes, y comprueba que están todos y que son correctos.</li><li>• Extrae la información útil de los paquetes</li><li>• Si detecta un paquete que no ha llegado o que es incorrecto, genera un paquete para ser enviado al emisor, indicándole que lo ha de enviar de nuevo.</li></ul>

# TEMA 2. Capa de Red

- © 2.1. Funcionalidades
- © **2.2. Conmutación**
- © 2.3. El protocolo IP
- © 2.4. Asociación con la capa de enlace: El protocolo ARP
- © 2.5. El protocolo ICMP
- © 2.6. Cuestiones y ejercicios



# ¿Qué es la conmutación?

- Proceso donde se pone en **comunicación un host con otro**, a través de una **infraestructura de comunicaciones común**, para la transferencia de información.
- Se necesita establecer un **sistema de comunicación** entre dos puntos, un **emisor (Tx) y un receptor (Rx)** a través de **equipos/nodos de transmisión**.
- Se determinará y **establecerá un camino** que permita **transmitir** información **extremo a extremo**.

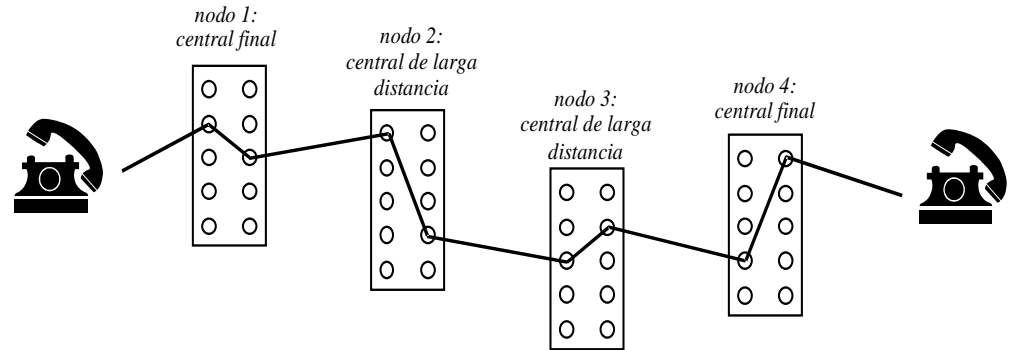
CONMUTACIÓN ⇔ REDIRECCIÓN

# ¿Qué es la conmutación?

- La conmutación para conectar redes entre sí funciona en la Capa 3 del modelo OSI (Capa de Red).
- Los **servicios** fundamentales que **emplean** técnicas de **conmutación** son:
  - Servicio telefónico
  - Servicio telegráfico
  - Servicio de datos
- **Tecnologías de conmutación:**
  - de circuitos
  - de paquetes (datagramas o circuitos virtuales)

# Conmutación de circuitos

- Consiste en el **establecimiento de un circuito físico previo al envío** de información, que **se mantiene abierto** durante **todo** el tiempo que dura la **transmisión**.
- El **camino físico se elige** entre los disponibles, **empleando** diversas **técnicas de señalización**: "por canal asociado" (si viaja en el mismo canal) o "por canal común" (si lo hace por otro distinto), encargadas de establecer, mantener y liberar dicho circuito.
- Ejemplo: *Red telefónica conmutada*



# Conmutación de circuitos

*Servicio  
orientado  
a conexión*

- **Pasos: (1) Conexión, (2) Transmisión, (3) Desconexión.**
- **Establecimiento del circuito:** el host emisor solicita a un cierto nodo de conmutación el establecimiento de conexión hacia un host receptor. Este nodo es el encargado de dedicar uno de sus canales lógicos al emisor. También será el encargado de encontrar los nodos intermedios para llegar al receptor, teniendo en cuenta ciertos criterios de encaminamiento, coste, etc...
- **Transferencia de datos:** una vez establecido el circuito exclusivo para esta transmisión, se transmite desde el emisor hasta el receptor conmutando sin demoras de nodo en nodo ( los nodos tienen reservado un canal lógico para ello) .
- **Desconexión del circuito:** Terminada la transferencia, el emisor o el receptor indican a su nodo de conmutación más inmediato que ha finalizado la conexión. Este nodo informa al siguiente de este hecho y luego libera el canal dedicado, así hasta liberar el canal dedicado completo en el otro extremo.

# Conmutación de circuitos

- **Ventajas:**

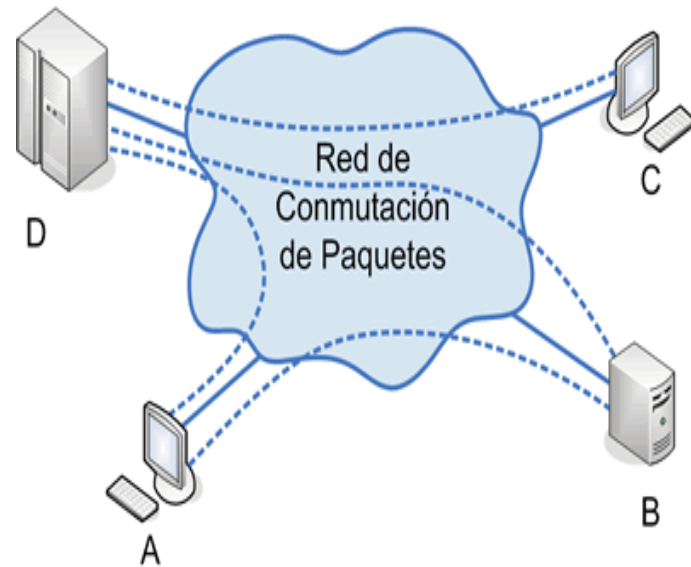
- Recursos dedicados (circuito en exclusiva).
- Facilita comunicaciones tiempo-real (voz y vídeo).
- No hay colisiones (no hay contienda por acceder al medio).
- No hay contención (el medio está disponible completamente → se transmite a la máxima velocidad posible).
- No hay encaminamiento (una vez establecido el circuito) ⇔ transmisión más rápida.
- Simplicidad de gestión en nodos (se recibe siempre por la misma entrada y se transmite siempre por la misma salida).

# Conmutación de circuitos

- **Desventajas:**
  - Retraso para establecimiento de la conexión (hay que resolver toda la ruta).
  - Bloqueo y posible infrautilización de recursos (la línea está reservada aunque no se aproveche).
  - Poca flexibilidad para adaptarse a cambios (no se reajusta la ruta si surgen posibles rutas alternativas mejores).
  - Poco tolerante a fallos (si falla un nodo del camino, se cae todo el circuito).

# Conmutación de paquetes

- **No** es **necesario** establecer una **conexión previa**.
- Un **paquete** consta de dos partes:
  - Datos útiles.
  - Información de control (para determinar la ruta a seguir a lo largo de la red hasta el destino).
- Los paquetes permanecen muy poco tiempo en memoria, por lo que resulta muy rápida.
- La conmutación de paquetes admite **dos variantes** distintas, según el modo de funcionamiento: **Datagrama** y **Circuitos Virtuales**



# Conmutación de paquetes

## PROCEDIMIENTO:

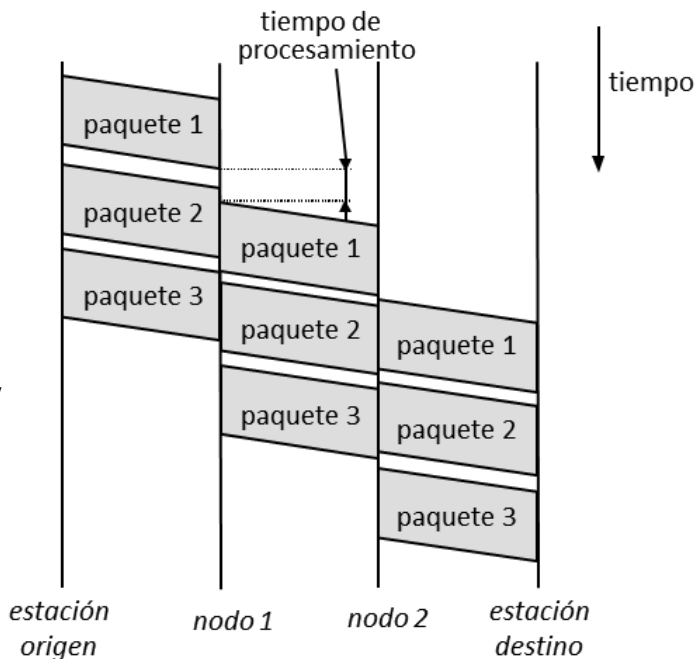
- Cuando un host quiere enviar **información** a otro lo **divide en paquetes**.
- Se lo **pasará a un nodo intermedio** que será el encargado de transmitirlo al siguiente hacia el destino.
- Cada **nodo intermedio** realiza las siguientes **funciones**:
  - **Almacenamiento y retransmisión** (*store and forward*): el paquete se detiene (se almacena) el tiempo necesario para procesarlo.
  - **Control de ruta** (*routing*): Selección de un nodo del camino por el que deben retransmitirse los paquetes para hacerlos llegar a su destino.
- Los **paquetes** toman **diversos caminos** pero nadie puede garantizar que todos los paquetes vayan a llegar en un momento determinado ni en un orden.



# Conmutación de paquetes

## CONMUTACIÓN DE DATAGRAMAS:

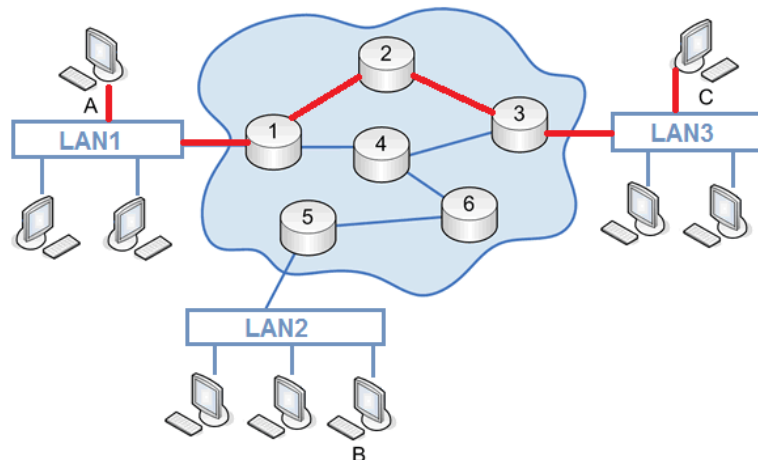
- **No hay conexión**
- Envío en unidades de datos (**paquetes**) **independientes**
- En cada salto: **almacenamiento y re-envío**
- Cada **paquete** debe contener las **direcciones origen y destino**
- Los **paquetes**, pueden seguir **rutas diferentes** y pueden llegar desordenados
- Ejemplo: **IP**



# Conmutación de paquetes

## CONMUTACIÓN CON CIRCUITOS VIRTUALES:

- **Orientado a conexión.**
- **Antes** de la transmisión se establece una **ruta entre el origen y el destino** (puede ser diferente en cada sentido).
- Se envían unidades de datos (**paquetes**) independientes.
- No se acaparan los **recursos** (se **comparten**).
- En cada salto: **almacenamiento y re-envío** (se debe comprobar si los recursos están libres).
- Los **paquetes llegarán ordenados**.
- Ejemplo: *ATM (Asynchronous Transfer Mode)*



# Conmutación de paquetes

## VENTAJAS DE CIRCUITOS VIRTUALES FRENTE A DATAGRAMAS:

- El **encaminamiento** en cada nodo **sólo se hace una vez** para todo el grupo de paquetes. Por lo que los paquetes llegan antes a su destino.
- Todos los **paquetes llegan en el mismo orden** del de partida ya que siguen el mismo camino.
- En cada **nodo** se realiza **detección de errores**, por lo que si un paquete llega erróneo a un nodo, éste lo solicita otra vez al nodo anterior antes de seguir transmitiendo los siguientes.

# Conmutación de paquetes

## DESVENTAJAS DE CIRCUITOS VIRTUALES FRENTE A DATAGRAMAS:

- En datagramas no hay que establecer la conexión → para **pocos paquetes**, es **más rápida** la **conmutación de datagramas**.
- Los **datagramas son más flexibles** → si hay congestión en la red, una vez que ya ha partido algún paquete, los siguientes pueden tomar caminos diferentes. En circuitos virtuales, esto no se hace.
- El envío mediante **datagramas es más fiable** → **si un nodo falla**, se perderá sólo un paquete. En circuitos virtuales se perderán todos (si no hay un mecanismo de recálculo de la ruta).

# TEMA 2. Capa de Red

- © 2.1. Funcionalidades
- © 2.2. Conmutación
- © **2.3. El protocolo IP**
- © 2.4. Asociación con la capa de enlace: El protocolo ARP
- © 2.5. El protocolo ICMP
- © 2.6. Cuestiones y ejercicios

# Introducción a IP

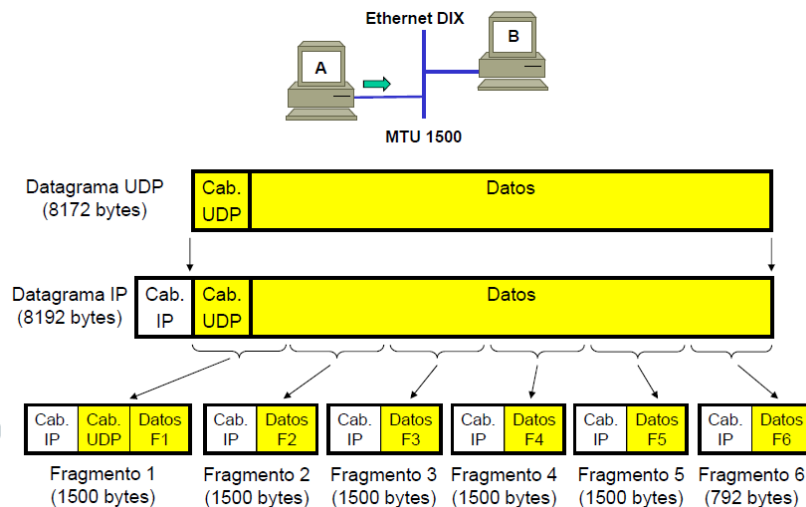
## IPv4

- Especificado en el RFC 791 (1349, 2474, 6864).
- Es un **protocolo** para la **interconexión de redes** (también llamadas subredes).
- Resuelve el **encaminamiento en Internet**: encontrar la ruta para llegar al destino.
- Es un protocolo **salto a salto**. Involucra a **hosts** y **routers**.
- Ofrece un servicio **no orientado a conexión** y **no fiable**:
  - No hay negociación o “handshake” → no hay una conexión lógica entre las entidades.
  - No existe control de errores, ni control de flujo, ni control de congestión.

# Introducción a IP

## IPv4

- La **unidad de datos** (paquete) de IP se denomina **datagrama**.
- IP es un protocolo de **máximo esfuerzo** (“*best-effort*”) o buena voluntad: los datagramas se pueden perder, duplicar, retrasar o llegar desordenados.
- IP **gestiona la fragmentación**: adaptar el tamaño del datagrama a las diferentes Maximum Transfer Units (MTUs) de las subredes hasta llegar al destino.



# Introducción a IP

- Cada entidad en Internet se **identifica por su dirección IP**.



Servidor  
Webmail  
130.206.192.39



Servidor  
Spotify  
78.31.8.101



[www.youtube.com](http://www.youtube.com)  
172.194.34.206



Buscar con Google

Voy a tener suerte

Google.es también en: [català](#) [galego](#) [euskara](#)

[www.google.com](http://www.google.com)  
172.194.34.209



ugr  
Universidad  
de Granada

[www.ugr.es](http://www.ugr.es)  
150.214.204.25  
dns3.ugr.es  
150.214.191.10  
pop.ugr.es  
150.214.20.3

Cada  
dirección IP  
es única en  
Internet



# Direcciones IP

## IPv4

- Una **dirección IP**  $\Leftrightarrow$  **etiqueta numérica** que **identifica**, de manera lógica a **una interfaz** de un sistema **dentro de una red** que utilice el protocolo IP.
- Internet adopta un **direccionamiento jerárquico** que simplifica las tablas de *routing*.
- Las direcciones IPv4 tienen **32 bits, agrupados en 4 bloques de 8 bits** cada uno.
- Se representan mediante **notación decimal** (entre 0 y 255) **separada por puntos**.

Ej: 200.110.23.77

# Direcciones IP

## IPv4

- Cada dirección IP tiene **dos partes** bien diferenciadas:
  - Un **identificador de la subred o prefijo** (parte izquierda de la IP)
  - Un **identificador del dispositivo** dentro de esa subred (parte derecha de la IP).
- Cada **subred** tiene un **identificador (o prefijo) único** en la intranet (red privada).
- Cada **dispositivo (interfaz)** tiene un **identificador único** en la subred.

# Direcciones IP

## IPv4

- La **máscara de red** es un patrón de '1s' que **determina qué bits** de la IP completa corresponden al **identificador de subred**.

Ejemplo:

Dirección IP: 200.27.4.112 → 11001000.00011011.00000100.01110000

Máscara: 255.255.255.0 → 11111111.11111111.11111111.00000000

- La máscara se puede representar de **forma compacta**, indicando el número de '1s' que tiene.

Ejemplo:

255.255.255.0 → 11111111.11111111.11111111.00000000 ⇔ /24

La dirección anterior con la máscara sería: 200.27.4.112/24

# Direcciones IP

## IPv4

- Dada una IP, para obtener la **dirección o identificador de la subred**, se realiza una **operación lógica “&” (AND)** con la **máscara de red**:

Ejemplo:

200.27.4.112 → 11001000.00011011.00000100.01110000

&

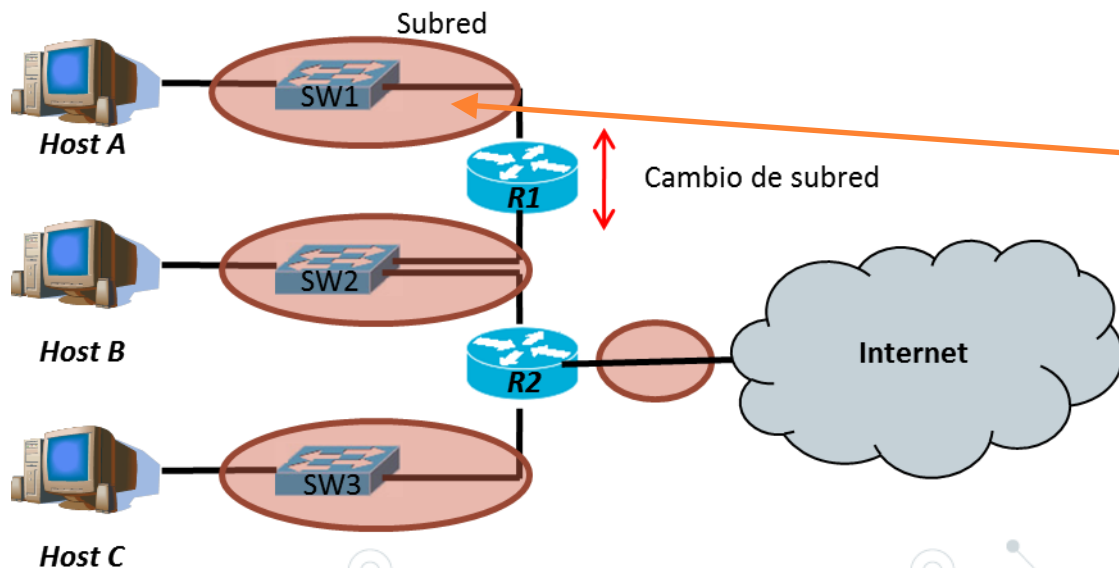
&

255.255.255.0 → 11111111.11111111.11111111.00000000

Subred → 200.27.4.0 ⇔ 11001000.00011011.00000100.00000000

# Subredes

- Podemos considerar Internet como un conjunto de subredes interconectadas
- ¿Qué es una subred? ¿Qué es un switch? ¿Qué es un router?

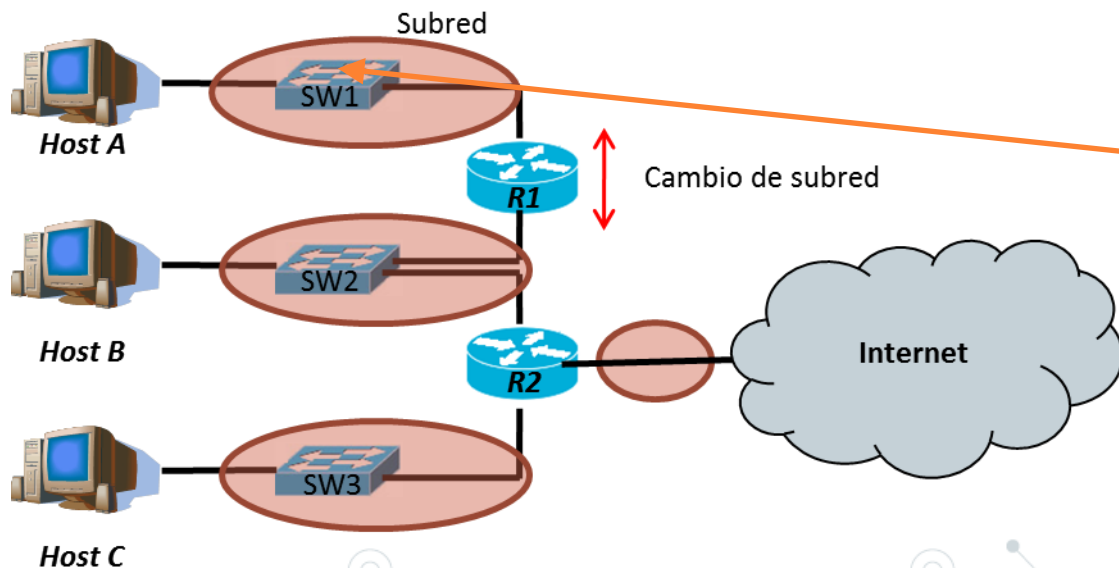


## SUBRED

Líneas de transmisión e infraestructura de red que permite la **conexión directa** de dispositivos IP sin intermediarios (un switch se considera transparente)

# Subredes

- Podemos considerar Internet como un conjunto de subredes interconectadas
- ¿Qué es una subred? ¿Qué es un switch? ¿Qué es un router?



## SWITCH

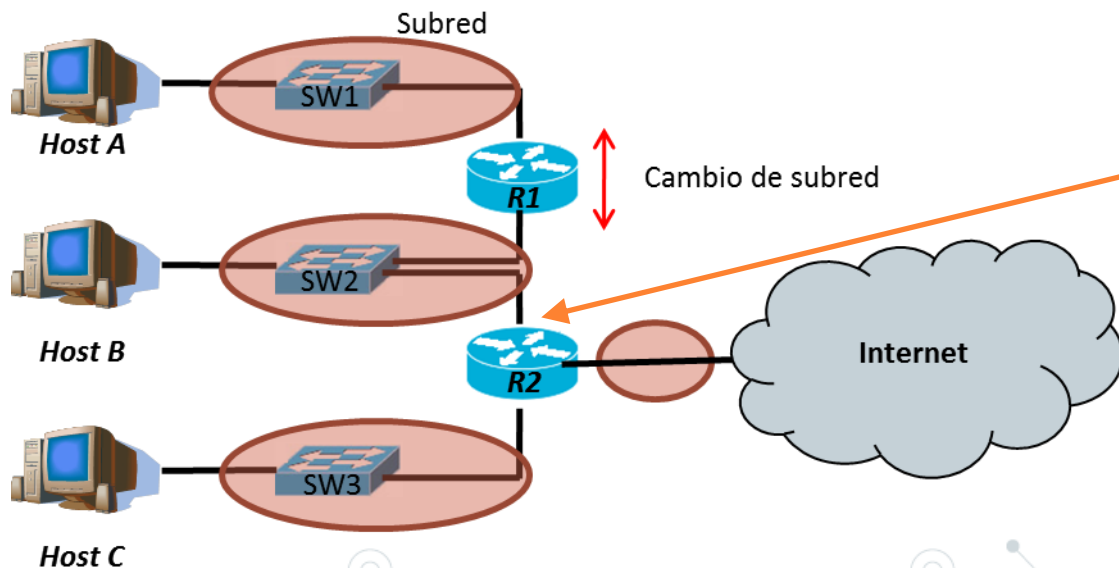
O conmutador.

Se usa para crear redes de computadoras. Son “transparentes”.

Trabaja a nivel de enlace (Capa 2 de OSI).

# Subredes

- Podemos considerar Internet como un conjunto de subredes interconectadas
- ¿Qué es una subred? ¿Qué es un switch? ¿Qué es un router?



## ROUTER

O encaminador.

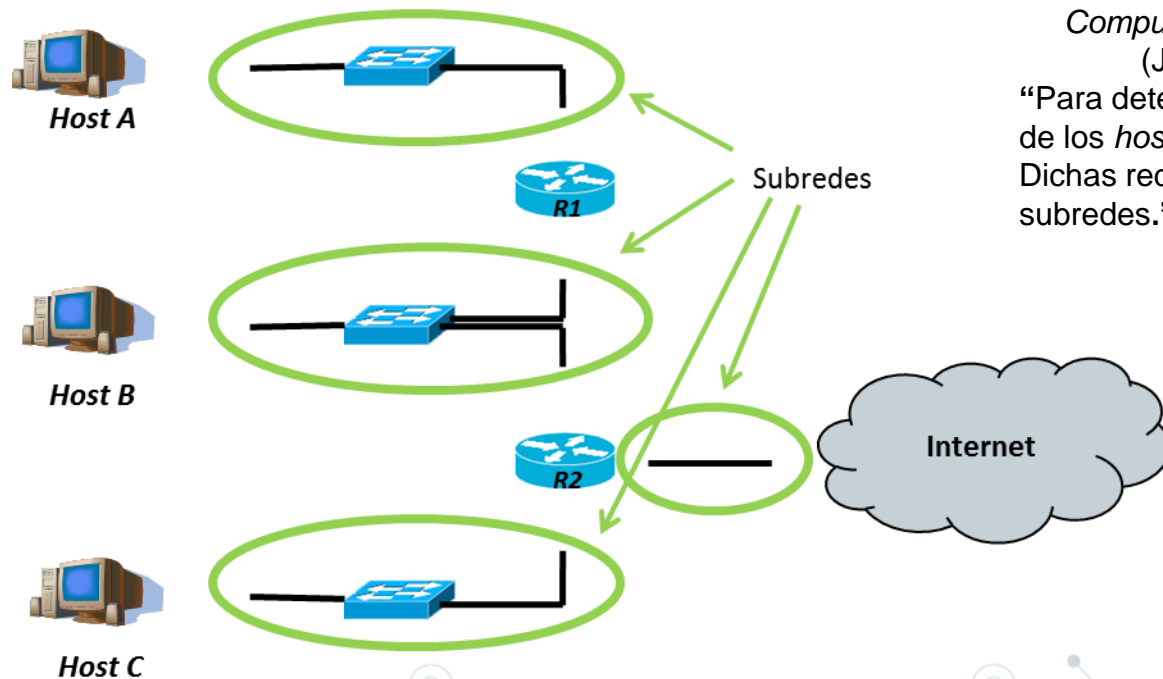
Se usa para conectar redes entre sí. Es un punto de separación, ya que limita el tráfico entre las redes.

Redirige los paquetes hacia el destino de una transmisión.

Trabaja a nivel de red (Capa 3 de OSI).

# Subnetting

- ¿Cómo determinar las subredes en un esquema de red?



*Computer Networking. A Top-down Approach.*

(James F. Kurose y Keith W. Ross)

“Para determinar las subredes, separe cada interfaz de los *hosts* y *routers*, creando redes aisladas. Dichas redes aisladas se corresponden con las subredes.”

Tendrán dirección IP  
cada una de las  
interfaces de los  
hosts y de los  
routers.  
Los switches no tienen  
dirección IP



# Subnetting

- ¿Cómo se elige la máscara? → Según el número de dispositivos que necesitemos direccionar en la subred, tal que se ajusta para no desaprovechar direcciones.

Dirección IP → 200.27.4.112 = 11001000.00011011.00000100.01110000

Máscara → 255.255.255.0 = 11111111.11111111.11111111.00000000

Número de dispositivos =  $2^{\text{número\_ceros}} - 2$

Ej: 8 ceros (/24) permite 254 dispositivos

El -2 viene de que la primera IP y última son reservadas

*Recuérdese:  
Cada subred tiene  
un identificador  
único en nuestra  
intranet*

# Subnetting

(Máscara /24)

*La dirección de Red/Subred tiene todo a 0s en la parte de host*

- 200.27.4.0 = 11001000.00011011.00000100.00000000 ➔ Reservada (subred)
- 200.27.4.1 = 11001000.00011011.00000100.00000001 ➔ Dispositivo #1
- ...
- 200.27.4.254 = 11001000.00011011.00000100.11111110 ➔ Dispositivo #254
- 200.27.4.255 = 11001000.00011011.00000100.11111111 ➔ Reservada (difusión)

*La dirección de Difusión/Broadcast tiene todo a 1s en la parte de host*

# Tipos de direcciones IP

## PÚBLICAS:

- Cada dirección se asigna a sólo 1 dispositivo (una interfaz) en toda la Internet global.
- Se asignan centralizadamente.

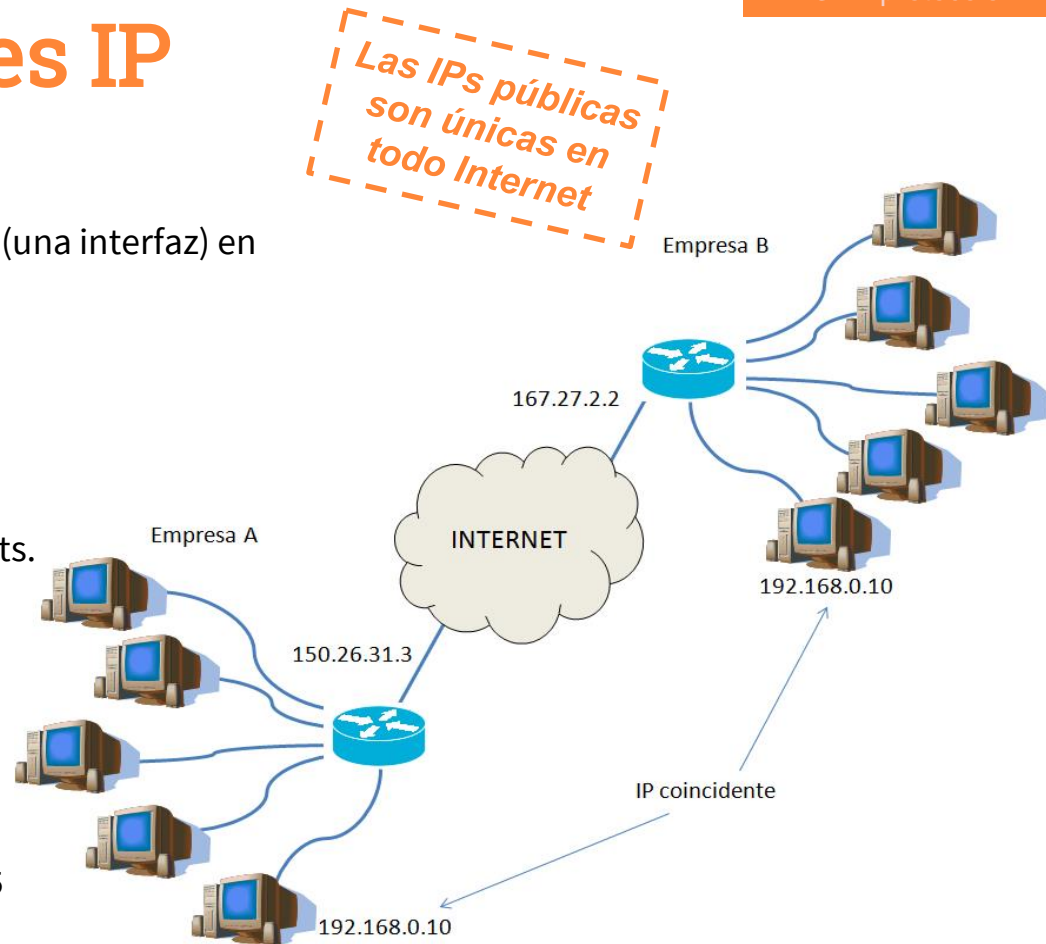
## PRIVADAS:

- Sólo sirven para tráfico dentro de las intranets.
- Se pueden repetir en distintas intranets.
- Las asigna el usuario según su criterio.
- Rangos de IPs privadas:

10.0.0.0/8 → de 10.0.0.0 a 10.255.255.255

172.16.0.0/16 → de 172.16.0.0 a 172.31.255.255

192.168.0.0/24 → de 192.168.0.0 a 192.168.255.255



# Clases de direcciones IP

- Especificadas en RFCs 1166 y 5737.
- Originariamente se definieron **5 clases de direcciones IP**.
- Clases **A, B, C** → **Jerárquicas a dos niveles**:

*identificador de red + identificador de dispositivo (host)*

Clase A	0	red (7 bits)	host (24 bits)
---------	---	--------------	----------------

Clase B	1 0	red (14 bits)	host (16 bits)
---------	-----	---------------	----------------

Clase C	1 1 0	red (21 bits)	host (8 bits)
---------	-------	---------------	---------------

Clase D	1 1 1 0	dirección grupo <i>multicast</i> (28 bits)	
---------	---------	--	--

Clase E	1 1 1 1 0	uso futuro	
---------	-----------	------------	--



Internet Assigned Numbers Authority



Internet Corporation for  
Assigned Names and Numbers

# Rangos de direcciones IP

- Según su clase:

A	→ 0.0.0.0 – 127.255.255.255	⇒ 128 redes x 16.777.216 <i>hosts</i>
B	→ 128.0.0.0 – 191.255.255.255	⇒ 16.384 redes x 65.536 <i>hosts</i>
C	→ 192.0.0.0 – 223.255.255.255	⇒ 2.097.152 redes x 256 <i>hosts</i>
D	→ 224.0.0.0 – 239.255.255.255	⇒ para <i>multicast</i>
E	→ 240.0.0.0 – 255.255.255.255	⇒ usos futuros

- Reglas especiales:

- host = 00...0 → identifica a una red, nunca es una dirección origen, no se usa para dispositivos
- host = 11...1 → difusión en la red especificada, es una dirección destino, no se usa para dispositivos
- 127.0.0.0 → autobucle (loopback)

- Reserva de direcciones privadas (RFC 1918):

A → 10.0.0.0 → 1 Red privada de Clase A

B → 172.16.0.0 – 172.31.0.0 → 16 redes privadas de Clase B

C → 192.168.0.0 – 192.168.255.0 → 256 redes privadas de Clase C

# Agotamiento de IPs

- Los bloques **de direcciones IPv4 se “agotaron”** ya (Nov. 2019)!!!
- Sólo quedan disponibles bloques /24 (256 direcciones) a /32 (1 dirección).
- Se van recopilando direcciones de sitios obsoletos, empresas que hayan desaparecido, proyectos terminados, hosting que ya no está en uso...

## IPv6

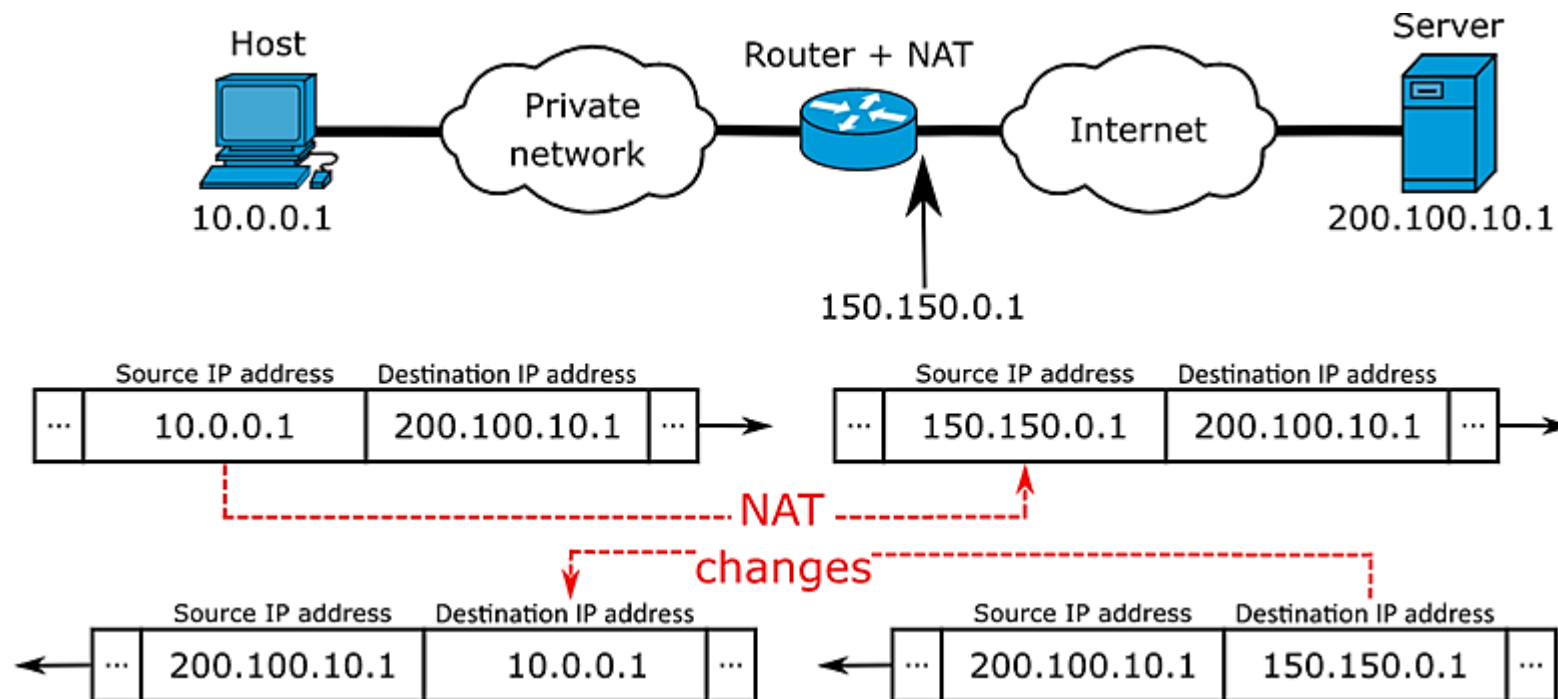
- IPv6 usa un esquema de **direccionamiento de 128 bits**.
- **Notación hexadecimal**. 8 grupos de 4 dígitos, separados por “:”.
- Cada dígito hexadecimal corresponde a 4 dígitos en binario (4 bits).
- **Rango:** 0000:0000:0000:0000:0000:0000:0000:0000    a
- FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
- 340.282.366.920.938.463.463.374.607.431.768.211.456 (340 sextillones) direcciones diferentes.
- **Compatible con IPv4.**

# NAT (Network Address Translation)

- RFC 1631, 2663, 3022.
- Consiste en **traducir un conjunto de direcciones IPv4 en otras**.
- Permite que una red con direccionamiento privado se pueda conectar a Internet (direccionamiento público).
  - Cambia la **dirección IP privada por una dirección pública** al reenviar un paquete hacia el exterior de la red (hacia Internet).
  - Cambia la **dirección IP pública por la correspondiente privada** al reenviar un paquete hacia el interior.
- Utiliza una **tabla de traducciones**, que contiene **direcciones IP y puertos**.

*Los puertos se asocian a los equipos de la red privada (para dirigir el tráfico entrante)*

# NAT (Network Address Translation)





# NAT (Network Address Translation)

## PROBLEMA DE LA ESCASEZ DE DIRECCIONES IP

- Se necesitan **m** direcciones pero se dispone de **n**, siendo  **$n < m$** .
- Si  **$n = 1$**  se denomina **enmascaramiento** (*masquerading*).
- Se usa en **ISPs**, para así poder **dar acceso a más usuarios que direcciones IP tenga el ISP**. Se supone que no todos los usuarios acceden simultáneamente. Las direcciones se asignan a los usuarios de forma dinámica.

## TIPOS DE NAT

- **SNAT (Source NAT)** → el origen de los datos está en la red privada; cambia la dirección IP de origen.
- **DNAT (Destination NAT)** → el origen de los datos está en la red pública; cambia la dirección IP de destino; requiere configurar en el router qué puerto irá dirigido a qué máquina.

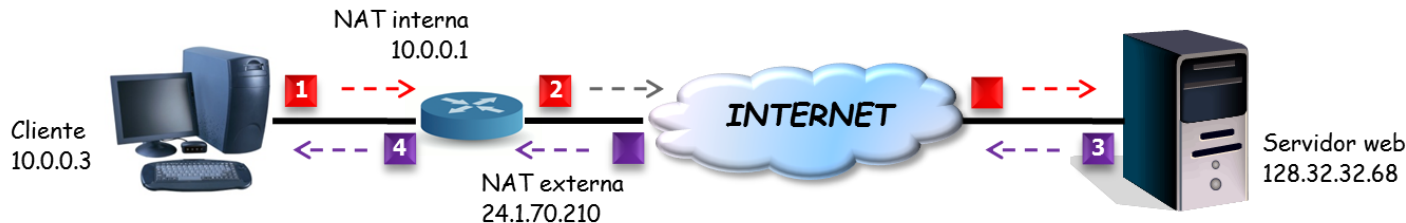
# NAT (Network Address Translation)

PROTO	TCP
SADDR	10.0.0.3
DADDR	128.32.32.68
SPORT	1049
DPORT	80
FLAGS	SYN
CKSUM	0x1636

1. El cliente intenta conectarse al servidor web 128.32.32.68 y envía un paquete SYN con su dirección IP interna 10.0.0.3 (privada).

PROTO	TCP
SADDR	24.1.70.210
DADDR	128.32.32.68
SPORT	40960
DPORT	80
FLAGS	SYN
CKSUM	0x2436

2. El dispositivo NAT ve la configuración del paquete, añade una nueva entrada a su tabla de traducción. Luego modifica el paquete usando su dirección IP externa (pública), cambia el puerto y el chequeo de integridad del paquete.



PROTO	TCP
SADDR	128.32.32.68
DADDR	10.0.0.3
SPORT	80
DPORT	1049
FLAGS	SYN, ACK
CKSUM	0x7841

4. El dispositivo NAT mira su tabla de traducción, y encuentra la que corresponde a direcciones y puertos origen/destino. Reescribe el paquete utilizando los puertos y direcciones internas.

Original	NAT
10.0.0.3:1049	24.1.70.210:40960
...	...

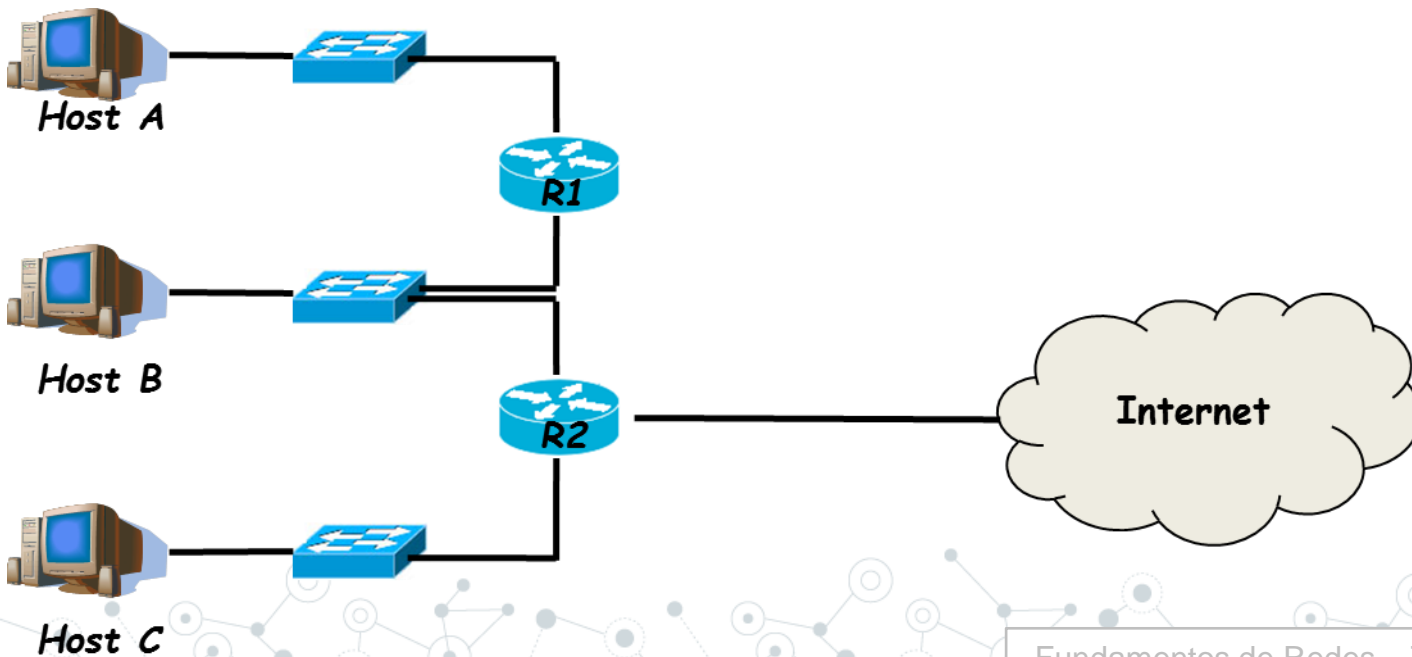
PROTO	TCP
SADDR	128.32.32.68
DADDR	24.1.70.210
SPORT	80
DPORT	40960
FLAGS	SYN, ACK
CKSUM	0x8041

3. El servidor responde con un paquete SYN, ACK. El paquete se envía a la dirección IP externa (pública) del dispositivo NAT.

# Ejercicio

## ASIGNACIÓN DE DIRECCIONES IP

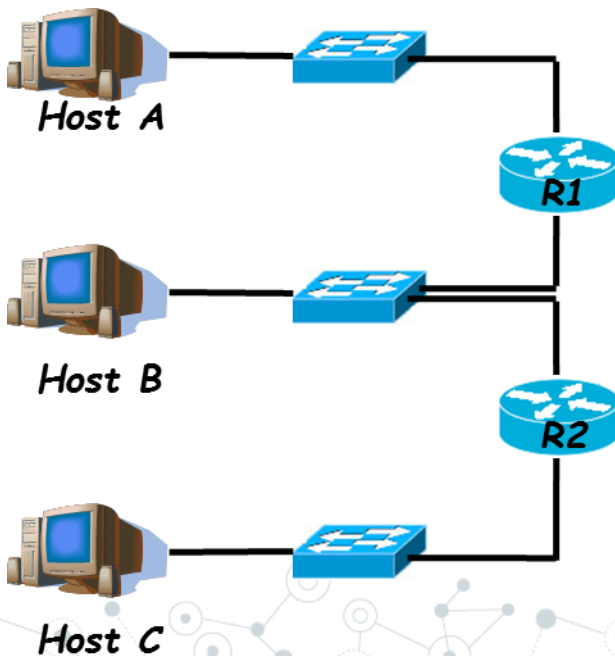
- Subredes corporativas: 30 dispositivos cada una, direcciones privadas en el rango 192.168.0.0
- Subred de acceso: dirección pública (ISP)



# Ejercicio

## ASIGNACIÓN DE DIRECCIONES IP

- Subredes corporativas: 30 dispositivos cada una, direcciones privadas en el rango 192.168.0.0
- Subred de acceso: dirección pública (ISP)

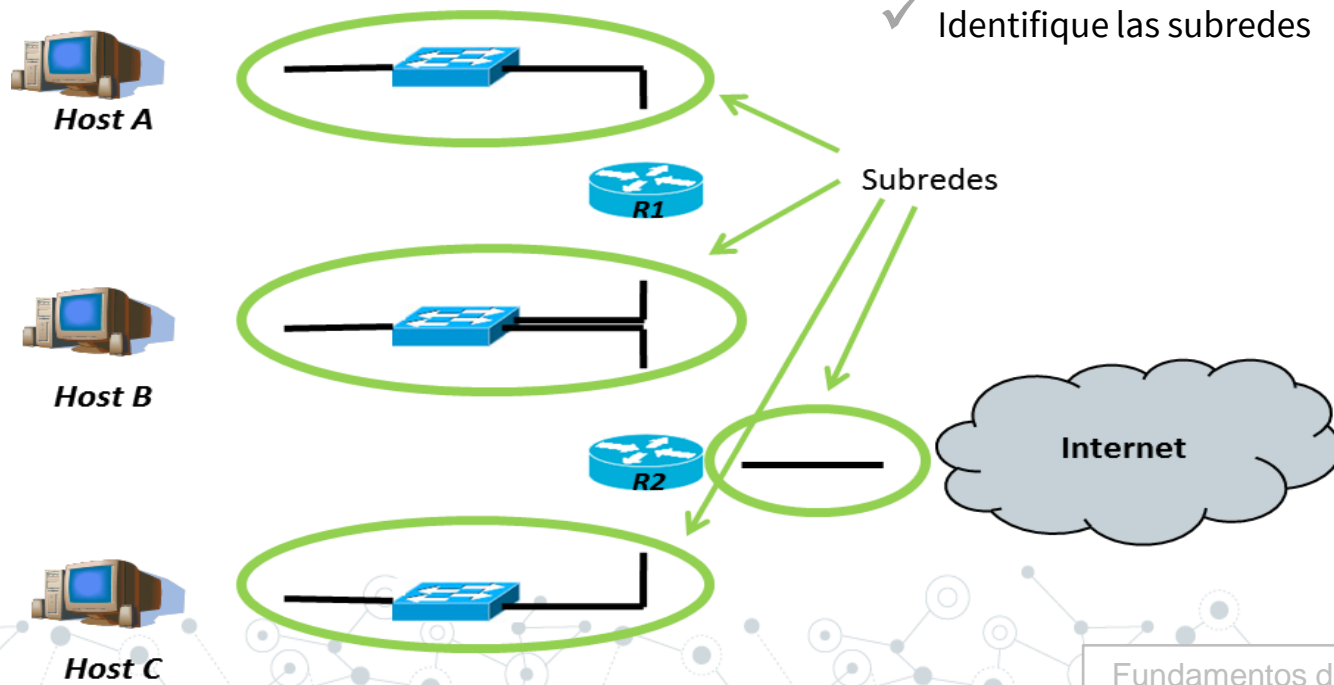


- ✓ Identifique las subredes
- ✓ Asocie una dirección de red y máscara a cada una
- ✓ Asigne una IP a cada interfaz

# Ejercicio

## ASIGNACIÓN DE DIRECCIONES IP

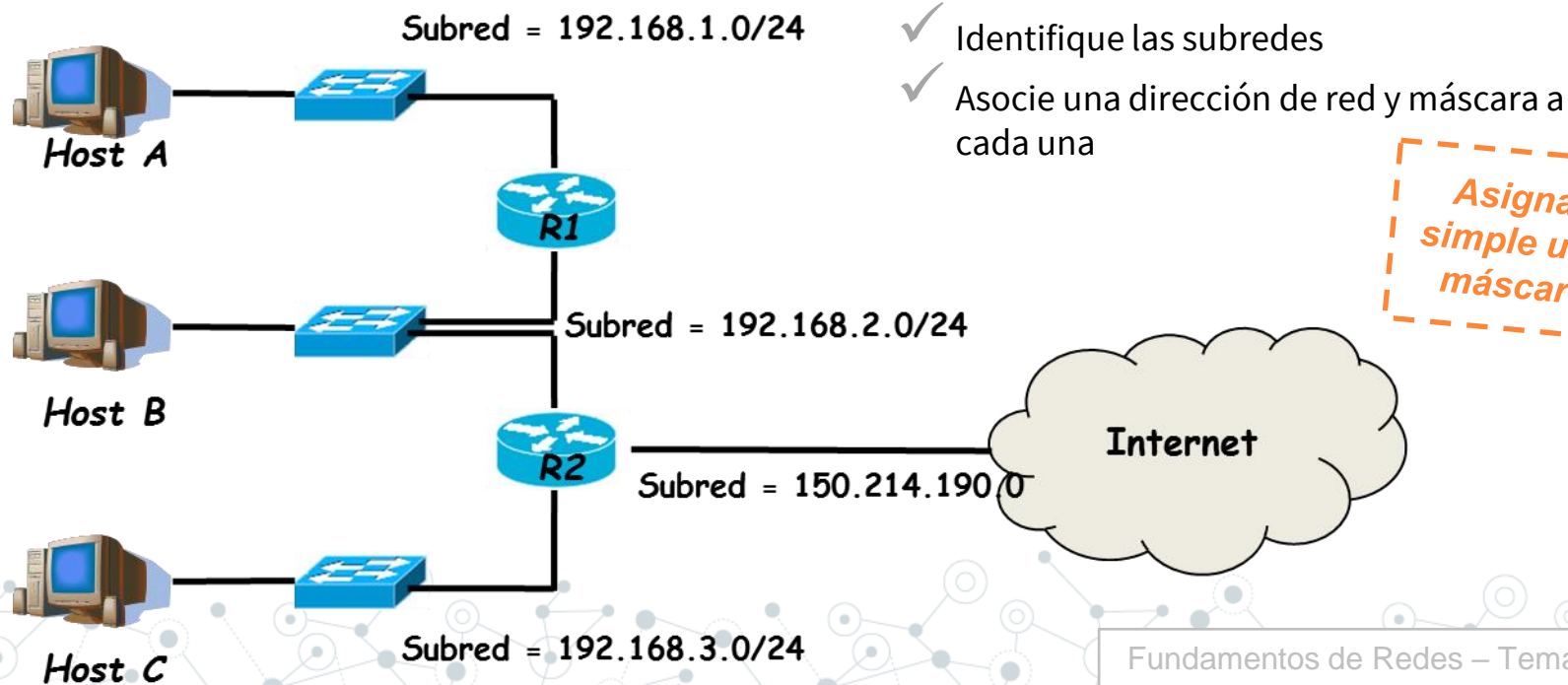
- Subredes corporativas: 30 dispositivos cada una, direcciones privadas en el rango 192.168.0.0
- Subred de acceso: dirección pública (ISP)



# Ejercicio

## ASIGNACIÓN DE DIRECCIONES IP

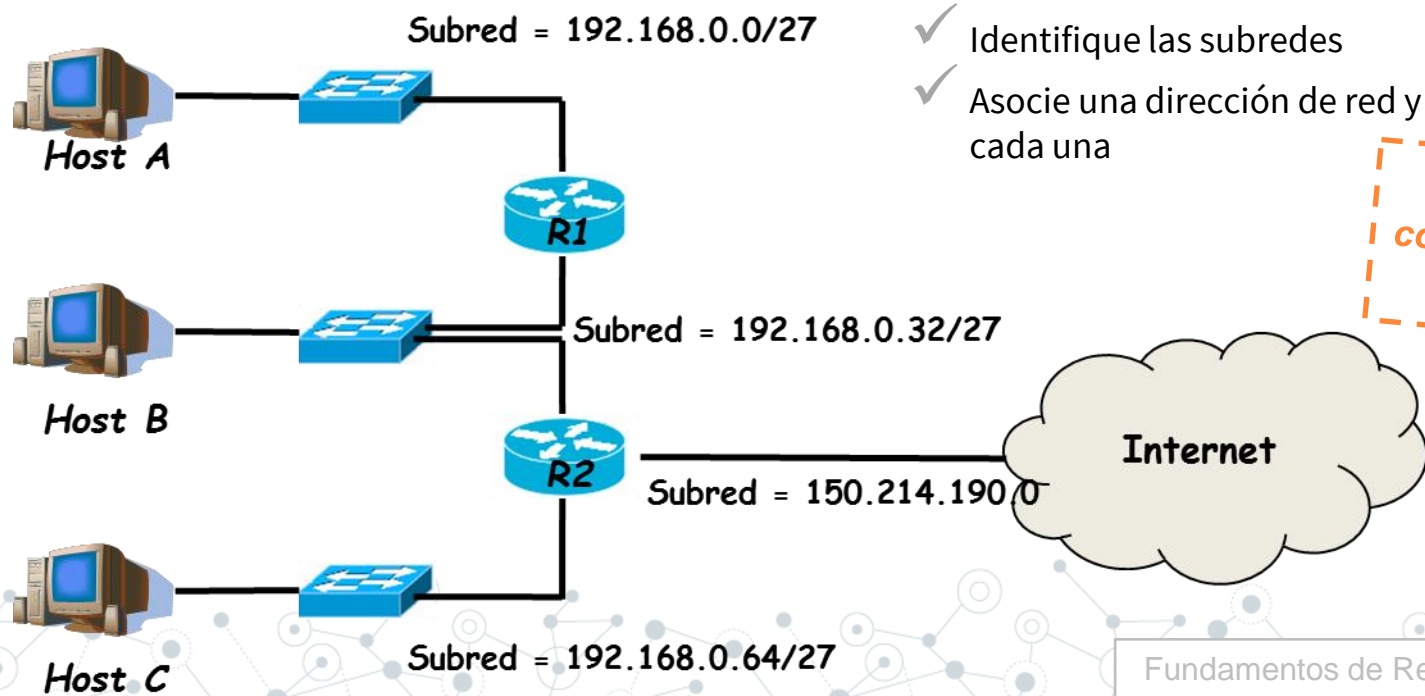
- Subredes corporativas: 30 dispositivos cada una, direcciones privadas en el rango 192.168.0.0
- Subred de acceso: dirección pública (ISP)



# Ejercicio

## ASIGNACIÓN DE DIRECCIONES IP

- Para direccionar 30 dispositivos → 5 bits en la parte de hosts.  $32-5=27$  bits para red → máscara /27
- Dirección pública ISP: 2 bits, /30, consideramos por ejemplo 150.214.190.0 (UGR)



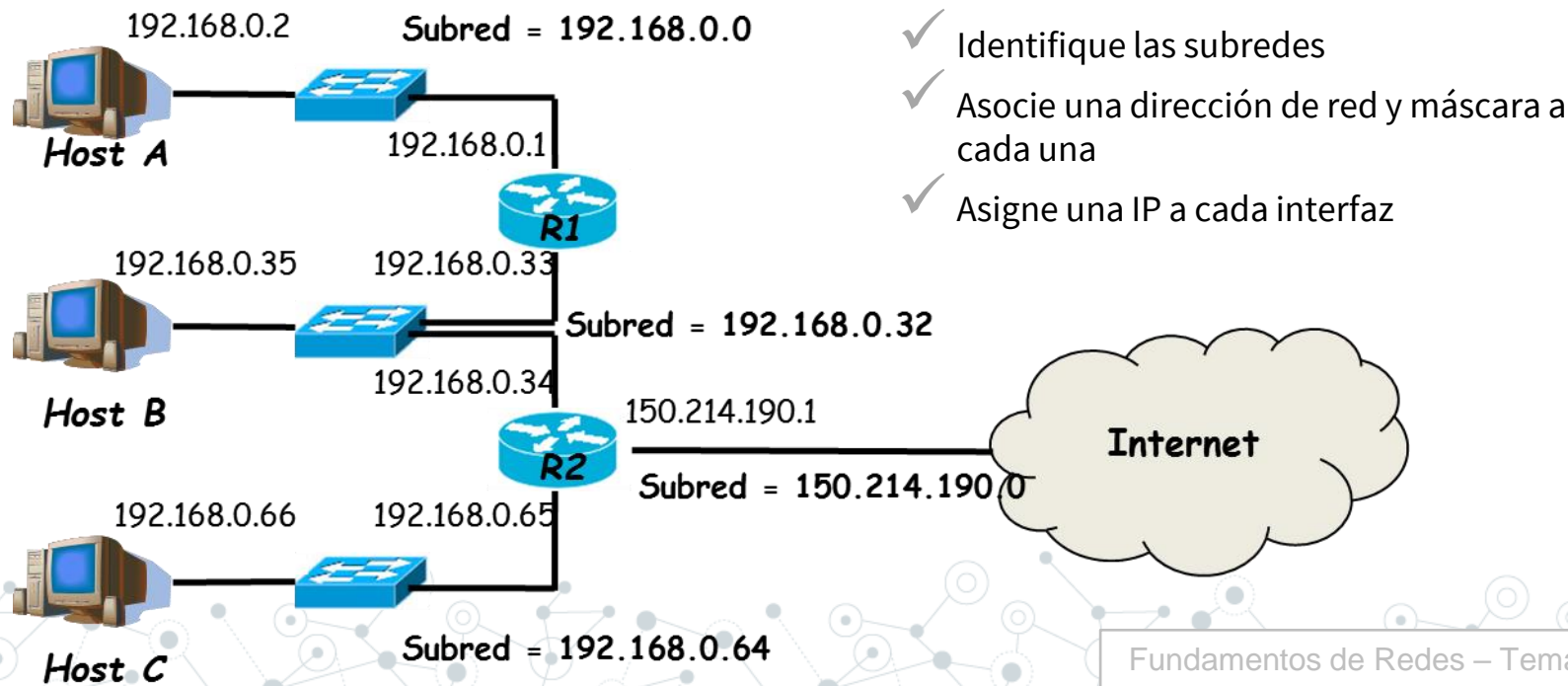
- ✓ Identifique las subredes
- ✓ Asocie una dirección de red y máscara a cada una

*Asignación compacta usando máscara /27*

# Ejercicio

## ASIGNACIÓN DE DIRECCIONES IP

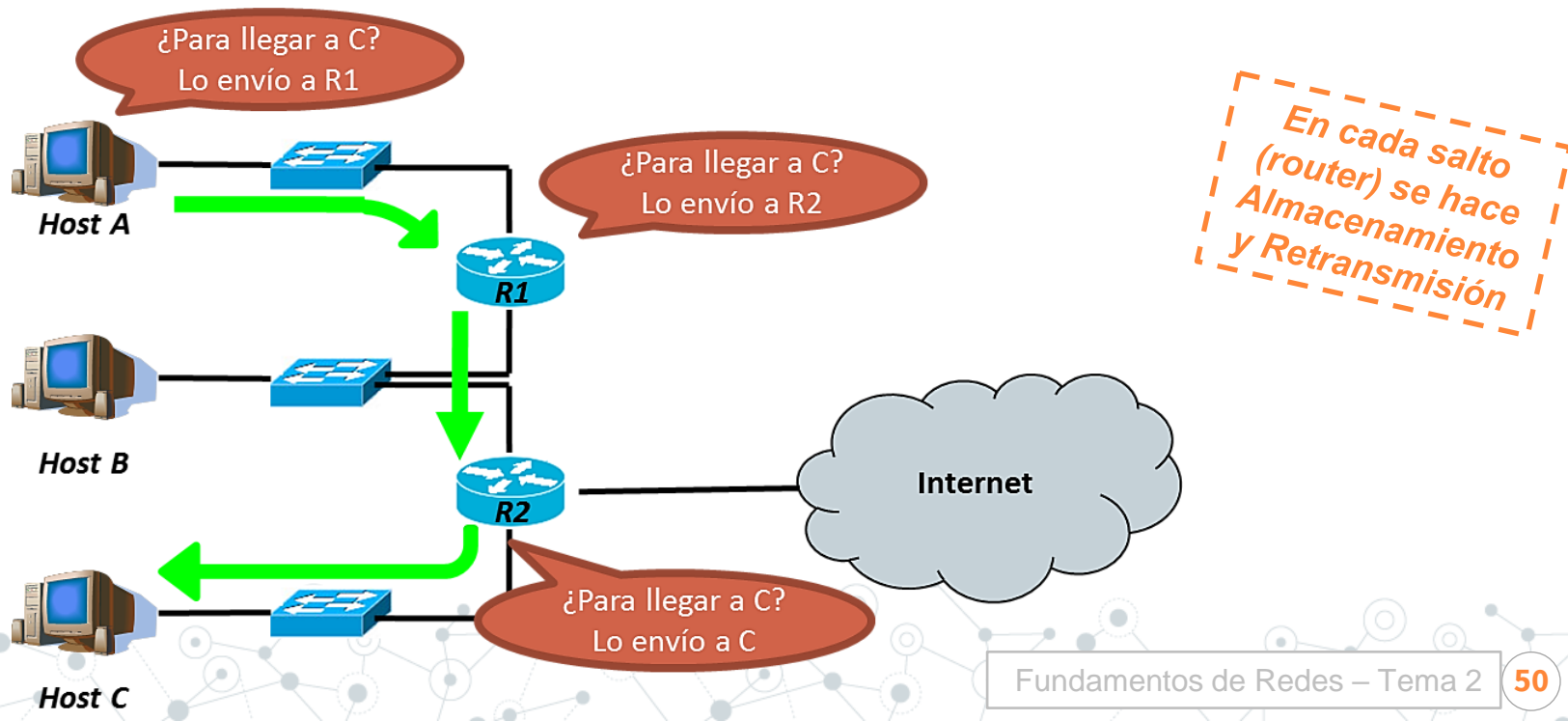
- Para direccionar 30 dispositivos → 5 bits en la parte de hosts.  $32-5=27$  bits para red → máscara /27
- Dirección pública ISP: 2 bits, /30, consideramos por ejemplo 150.214.190.0 (UGR)





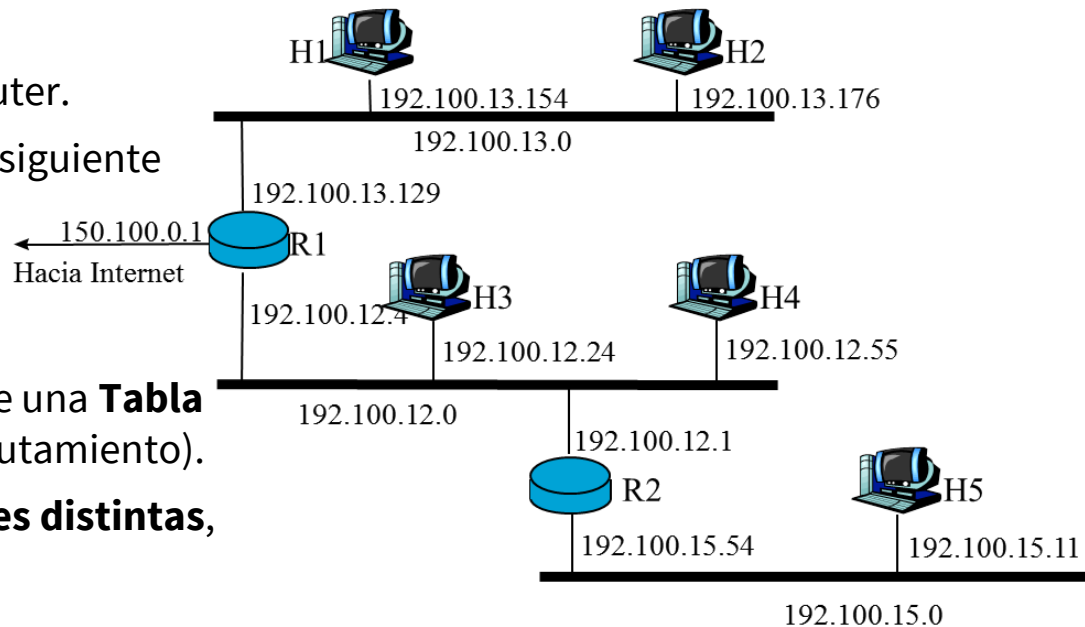
# Encaminamiento (Enrutamiento)

- **Encontrar el mejor camino** para llevar la información (paquete) de **un origen a un destino dado**.
- Se realiza **paquete a paquete** y **salto a salto**, en función de la IP destino del paquete y de las **Tablas de Encaminamiento** residentes en cada una de las entidades IP (host origen y routers).



# Encaminamiento (Enrutamiento)

- El encaminamiento se realiza **salto a salto** y **datagrama a datagrama** (IP es no orientado a conexión).
- Modos de encaminamiento:
  - **directo** → lo resuelve el propio router.
  - **no directo** → lo resuelve el router siguiente en la ruta.
- Cada dispositivo (host o router) tiene una **Tabla de encaminamiento** (o Tabla de enrutamiento).
- Un **router suele estar en varias redes distintas**, un host suele estar en solo una.



# Encaminamiento (Enrutamiento)

- Tabla de encaminamiento de R1**

Destino ( $D_i$ )	Salto siguiente ( $S_i$ )	Máscara ( $M_i$ )
127.0.0.1	* <b>Conexión directa</b>	255.255.255.255
192.100.12.0	*	255.255.255.0
192.100.13.0	*	255.255.255.0
192.100.15.0	192.100.12.1	255.255.255.0
Default	150.100.0.222	<b>0.0.0.0</b>

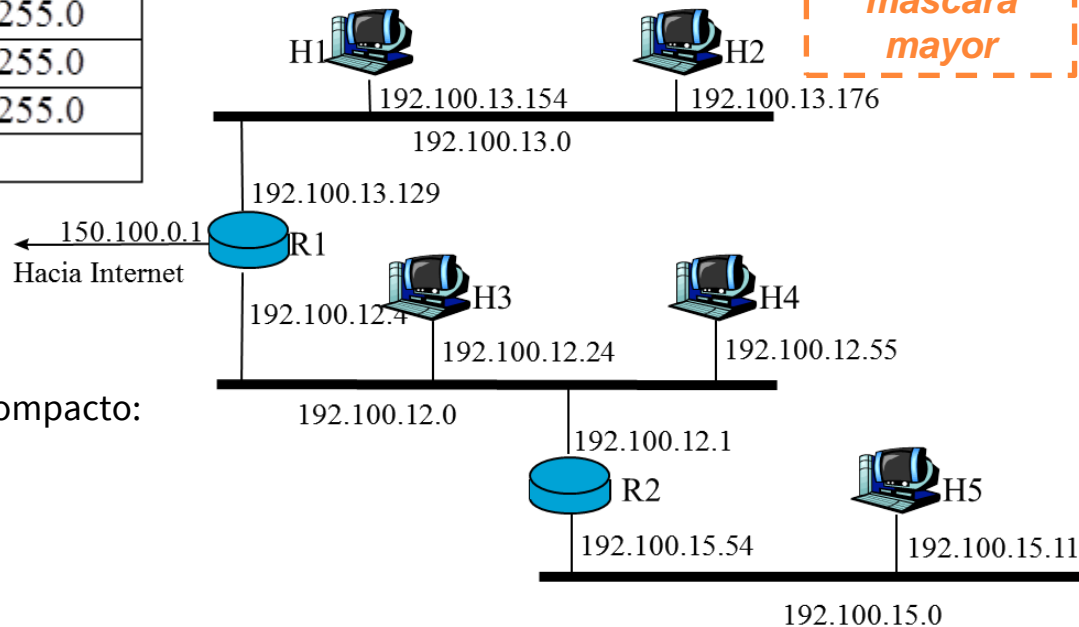
- ¿Faltaría alguna entrada?  
Una específica a la red 150.100.0.0/30

- La máscara se puede indicar en formato compacto:

255.255.255.0 ⇔ /24

255.255.255.192 ⇔ /26

255.255.255.252 ⇔ /30



Los destinos suelen ser subredes completas

Si hay dos entradas en conflicto se elige la más restrictiva ⇔ máscara mayor

# Encaminamiento (Enrutamiento)

i	Destino ( $D_i$ )	Salto siguiente ( $S_i$ )	Máscara ( $M_i$ )	Flags	Interfaz( $I_i$ )
1	127.0.0.1	*	255.255.255.255	H	lo
2	192.100.12.0	*	255.255.255.0	-	eth0
.	192.100.13.0	*	255.255.255.0	-	eth1
.	192.100.15.0	192.100.12.1	255.255.255.0	G	eth0
N	Default	150.100.0.222	0.0.0.0	G	eth2

## PROCESO DE ENCAMINAMIENTO (EN CADA NODO Y PARA CADA DATAGRAMA)

- Se extrae la dirección destino: IP\_DESTINO del datagrama
- Por cada entrada i con  $i=1, \dots, N$ , de la tabla de encaminamiento se calcula:  

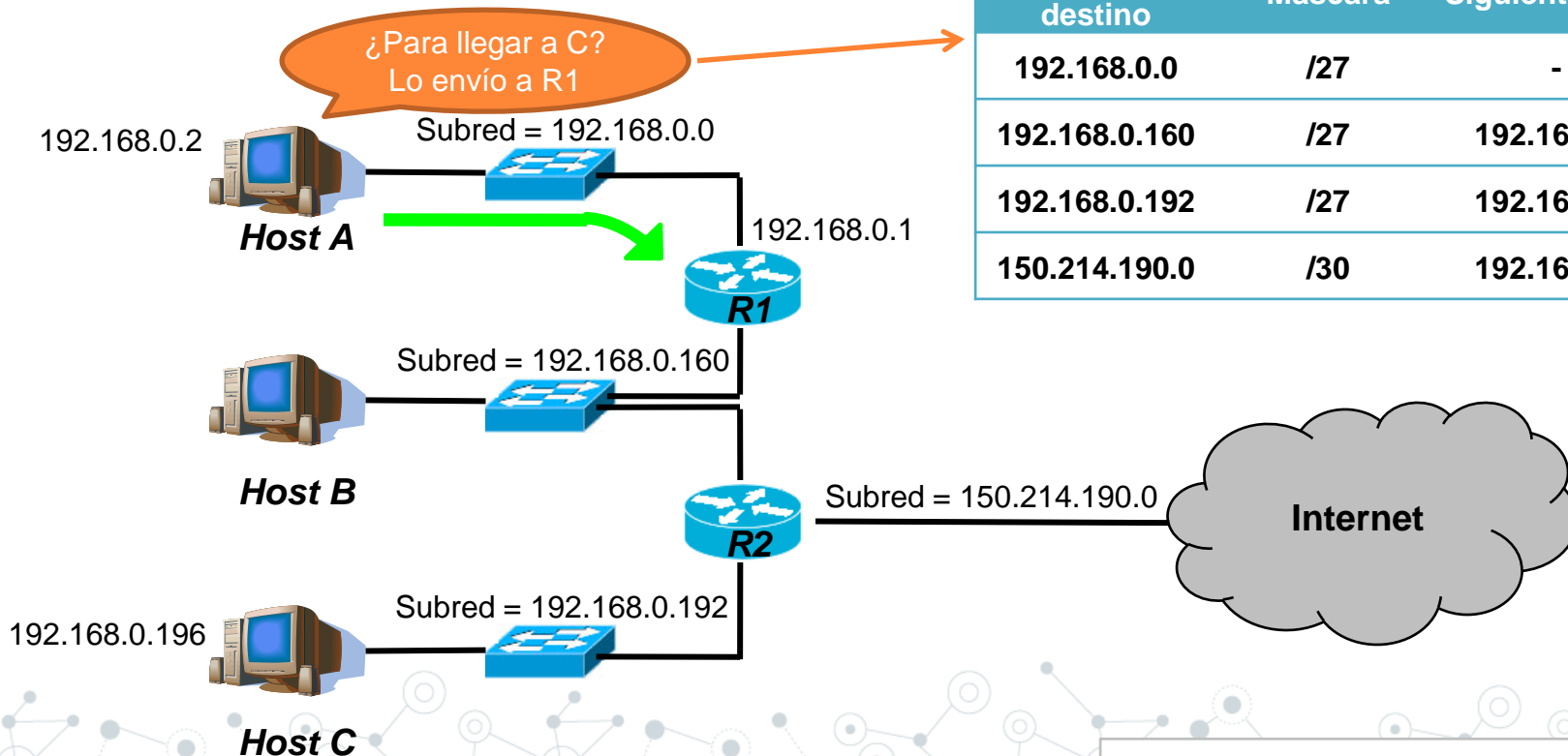
$$IPI = IP\_DESTINO \text{ AND } (\&) \text{ MASCARA}_i$$
- Si  $IPI = Di$  y  
 si es routing directo (\*) → reenviar el datagrama al destino final por la interfaz i  
 o si no es routing directo → reenviar el datagrama al salto siguiente por la interfaz i
- Si hay varias coincidencias se elige el destino con la máscara más larga (con más 1s)
- Si se ha barrido toda la tabla y no hay coincidencia con ninguna fila → error (posible mensaje ICMP)

# Ejemplo encaminamiento

## TABLA DE ENCAMINAMIENTO

Tabla de Host A

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1



# Ejemplo encaminamiento

- Se comprueba la **tabla de Host A**.
- Dirección de destino (DD): 192.168.0.196
- Para cada entrada (fila en la tabla)
- DD & Máscara = A
- ¿A = Dirección de destino en tabla?  
 SI → elegir el "Siguiete Nodo" → consultar TABLA ARP  
 NO → seguir buscando

*Esto se repite en cada router*

Tabla de Host A

Dirección IP destino	Máscara	Siguiete nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

➤ 192.168.0.196 & /27 = 11000000.10101000.00000000.110**00100** & /27 = 192.168.0.192  
 ¿192.168.0.192 = 192.168.0.0? NO

➤ 192.168.0.196 & /27 = 11000000.10101000.00000000.110**00100** & /27 = 192.168.0.192  
 ¿192.168.0.192 = 192.168.0.160? NO

➤ 192.168.0.196 & /27 = 11000000.10101000.00000000.110**00100** & /27 = 192.168.0.192  
 ¿192.168.0.192 = 192.168.0.192? SÍ ➔ **Siguiete Nodo = 192.168.0.1**

➤ 192.168.0.196 & /30 = 11000000.10101000.00000000.110001**00** & /30 = 192.168.0.196  
 ¿192.168.0.196 = 150.214.190.0? NO

# Ejemplo encaminamiento

## PROBLEMA

- La topología implica sólo un camino de salida desde Host A → ¿realmente necesitamos 4 entradas?

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

Podremos agrupar  
entradas de la tabla que  
tengan distinto destino,  
pero el mismo salto  
siguiente

Podemos agrupar entradas

# Ejemplo encaminamiento

## PROBLEMA

- La topología implica sólo un camino de salida desde Host A → ¿realmente necesitamos 4 entradas?

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

Podemos agrupar entradas

Buscamos los bits en común (iguales):

192.168.0.160 ⇔ 11000000.10101000.00000000.10100000

192.168.0.192 ⇔ 11000000.10101000.00000000.11000000

La máscara del agrupamiento indicará el número de bits iguales → /25

La dirección agrupada será la parte común y el resto de bits estarán a 0:

11000000.10101000.00000000.10000000

La entrada quedaría como:

192.168.0.128/25



# Ejemplo encaminamiento

## PROBLEMA

- La topología implica sólo un camino de salida desde Host A → ¿realmente necesitamos 4 entradas?

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1



Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.128	/25	192.168.0.1
150.214.190.0	/30	192.168.0.1

Podemos agrupar entradas

*No merece la pena agrupar direcciones muy diferentes, porque la entrada agrupada será muy genérica*

# Ejemplo encaminamiento

## PROBLEMA

- La tabla del ejemplo NO direcciona Internet (ej. [www.google.com](http://www.google.com) = 172.194.34.209)
- La topología implica sólo un camino de salida desde Host A → ¿realmente necesitamos 4 entradas?

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

¡¡Usar la entrada por defecto!! → /0

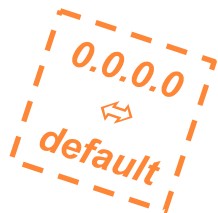
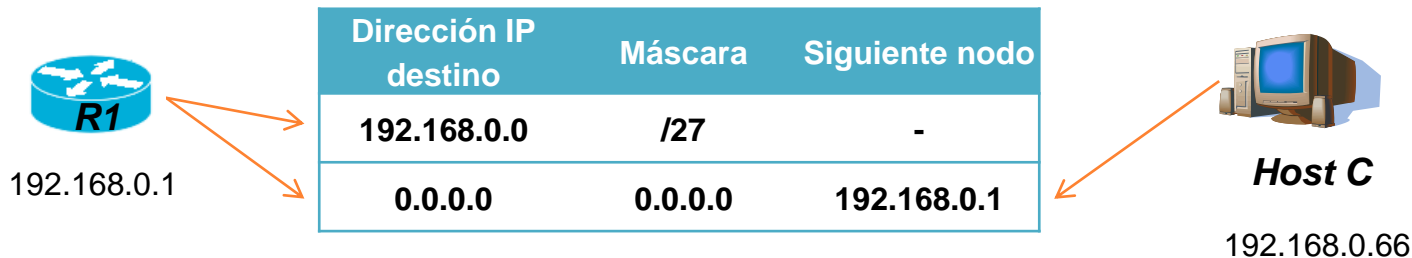
*La entrada por defecto se suele añadir para dirigir el tráfico hacia fuera de la red (hacia Internet)*

*Aunque en este ejemplo, se puede usar para dirigir el tráfico a las demás subredes también.*

# Ejemplo encaminamiento

## PROBLEMA

- La tabla del ejemplo NO direcciona Internet (ej. [www.google.com](http://www.google.com) = 172.194.34.209)
- La topología implica sólo un camino de salida desde A → ¿realmente necesitamos 4 entradas?



[www.youtube.com](http://www.youtube.com)  
172.194.34.206



Google es también en: [català](#) [galego](#) [euskara](#)

[www.google.com](http://www.google.com) =  
172.194.34.209



Servidor Spotify  
78.31.8.101

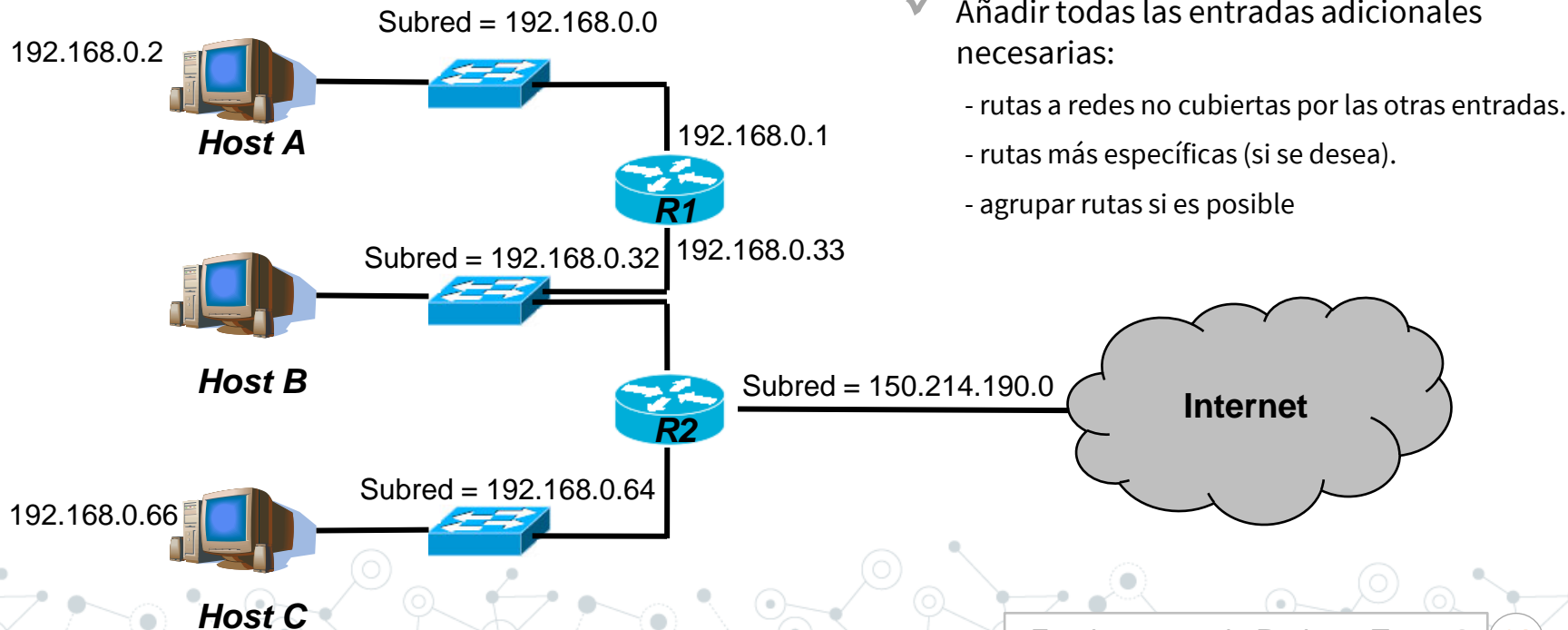


ugr  
Universidad  
de Granada

dns3.ugr.es = 150.214.191.10  
pop.ugr.es = 150.214.20.3

# Ejercicio

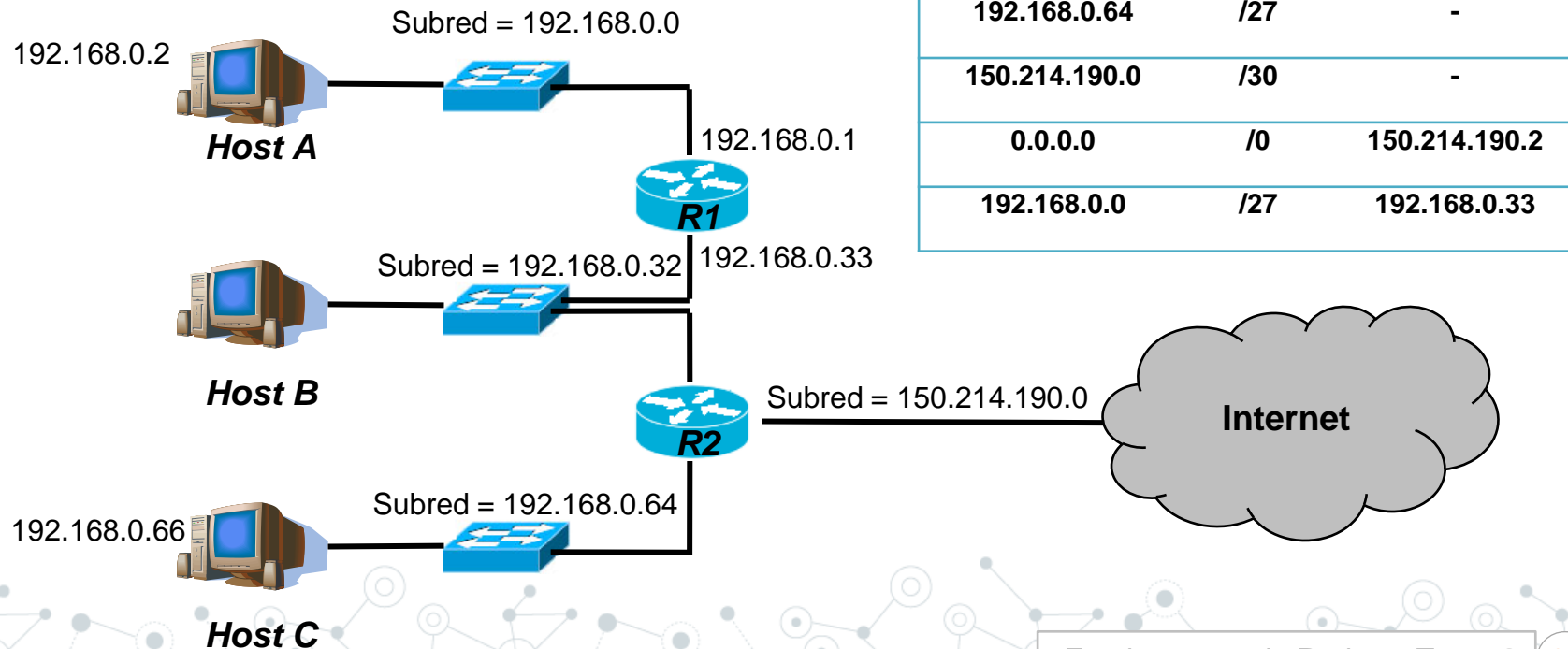
- Diseñar la Tabla de encaminamiento en R2



- ✓ Incorporar todas las redes directamente conectadas.
- ✓ Incorporar la entrada por defecto .
- ✓ Añadir todas las entradas adicionales necesarias:
  - rutas a redes no cubiertas por las otras entradas.
  - rutas más específicas (si se desea).
  - agrupar rutas si es posible

# Ejercicio

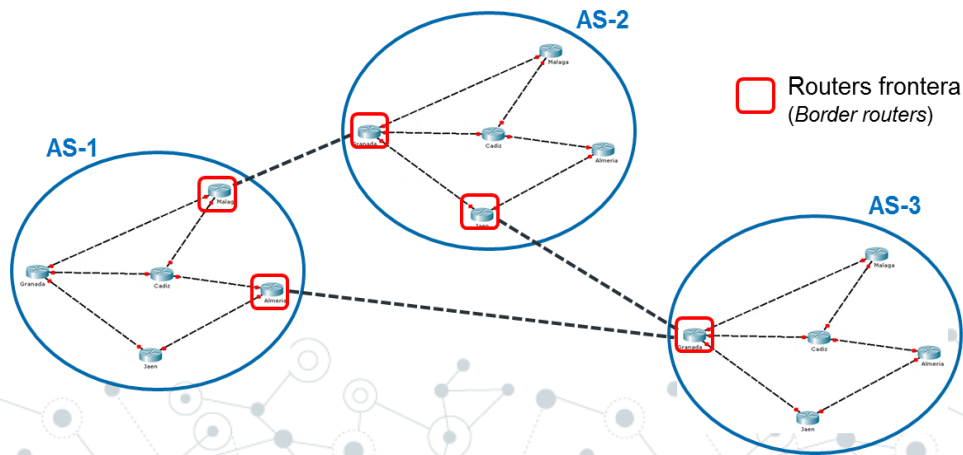
- Diseñar la Tabla de encaminamiento en R2



# Sistemas Autónomos

- Para **facilitar la administración** y **aumentar la escalabilidad** Internet se jerarquiza en **Sistemas Autónomos (SA)**.
- Un **SA es un conjunto de redes y routers** administrados por una autoridad.
- Cada **SA informa a los otros SA** de las **redes accesibles**.

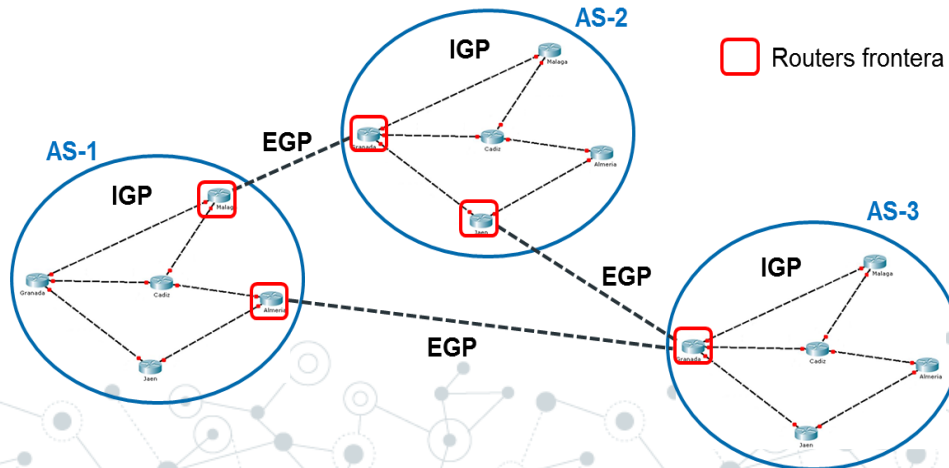
Existe un router responsable de esto, denominado **router exterior** (o *router frontera*).
- Cada **SA se identifica por un entero de 16 bits** (DESDE 2007 ES 32-BITS). Ej: Rediris → AS766



# Sistemas Autónomos

## INTERCAMBIO DE TABLAS

- Internet se **jerarquiza en Sistemas Autónomos**.
- Existe **encaminamiento dinámico** (mediante algoritmos automáticos).
- Se definen 2 niveles de encaminamiento (intercambio de tablas):
  - **Algoritmos IGP** → los que se usan dentro de un SA (el administrador tiene libertad de elección): **RIP, OSPF, HELLO, IGRP, EIGRP**
  - **Algoritmos EGP** → los que se usan entre SAs (norma única en Internet): **BGP**



# Algoritmos de Encaminamiento

## VECTOR DISTANCIA

- Los routers construyen su tabla de rutas con el único conocimiento de la distancia (métrica) y el siguiente salto (next hop) para llegar a la red de destino.
- Esta distancia puede ser un número que indica: longitud del enlace, número de saltos, latencia (tiempo medio) u otros valores.
- Requiere intercambiar información periódicamente con los routers vecinos para recalcular la distancia. Cada router envía su tabla de encaminamiento a los demás.
- Ejemplo: RIP

## ESTADO DEL ENLACE

- Los routers necesitan conocer previamente toda la topología de la red (conexiones existentes entre los nodos) para calcular el camino al destino y generar su tabla de enrutamiento.
- Ejemplo: OSPF

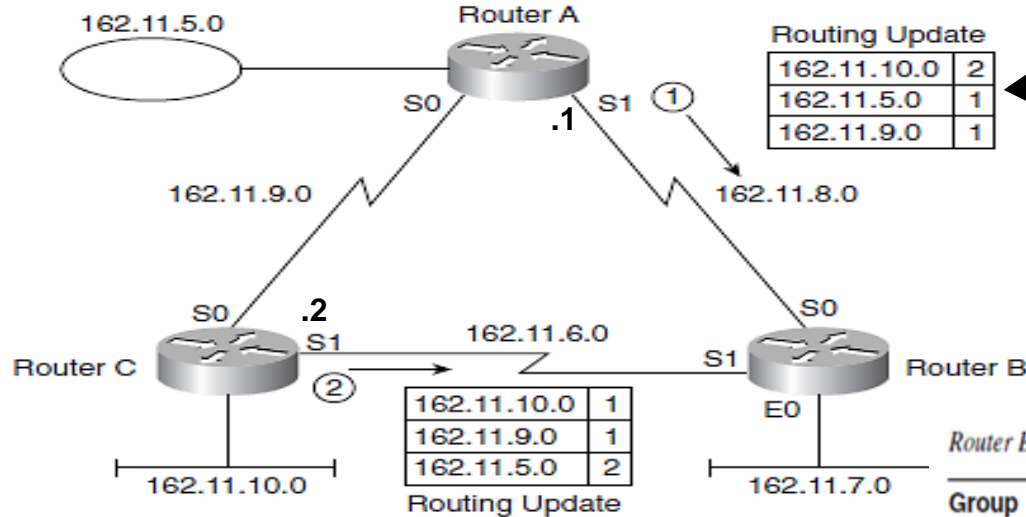


# RIP

- **Routing Information Protocol** (RFC 1058, 2453, 4822).
- Protocolo de la **capa de aplicación** (opera sobre **UDP** puerto 520).
- Adopta un **algoritmo vector-distancia** (métrica basada en **número de saltos**).
- No considera la congestión de la red ni la velocidad de los enlaces.
- Una red directamente conectada a un router tiene coste 1.
- **Máximo de 15 saltos** (16 sería considerada distancia infinita o no alcanzable).
- Periódicamente (por defecto **cada 30 segundos**) cada **router RIP recibe de todos sus vecinos y envía a todos sus vecinos** (dirección multicast 224.0.0.9) los **vectores-distancia para todos los posibles destinos**.
- De entre ellos, para un **destino dado**, se **selecciona como salto siguiente el vecino que anuncie el menor coste**, actualizando la métrica para ese destino sumando uno al coste anunciado (coste para alcanzar ese vecino desde el router actual).
- Problema convergencia lenta → las malas noticias tardan en propagarse.

# RIP (Ejemplo)

Routers A and C Advertising to Router B



1.- Router A envía actualización a Router B indicando en cuántos saltos Router B podría alcanzar estas redes

3.- Router B compila el mejor camino posible y actualiza el coste. Además, aparecen dos rutas de igual coste (métrica) a la misma red.

2.- RouterC envía actualización a RouterB indicando en cuántos saltos RouterC puede alcanzar estas redes

Router B Routing Table

Group	Outgoing Interface	Next Router	Metric
162.11.5.0	S0	162.11.8.1 (Rout A)	<del>1</del> 2
162.11.6.0	S1		<del>0</del> 1
162.11.7.0	E0		<del>0</del> 1
162.11.8.0	S0		<del>0</del> 1
162.11.9.0	S0	162.11.8.1, 162.11.6.2	<del>1</del> 2
162.11.10.0	S1	162.11.6.2 (Rout C)	<del>1</del> 2

# OSPF

- **Open Shortest Path First** (RFC 2328).
- Basado en **estado del enlace**.
- Se **publican los estados** por difusión/inundación.
- El **coste por defecto** que se considera en OSPF para cada enlace es: **coste =  $10^8/\text{BW}$** .

Ej: para un enlace con BW = 1 Mbps

$$\text{coste} = 10^8 / 10^6 = 100$$

- El **coste** de los enlaces **se podrá determinar en tiempo real** → un administrador o un algoritmo automático.
- Permite calcular **rutas alternativas** y hacer **balanceo de carga**. Se pueden considerar **distintas métricas**.
- Así se conseguirá **dar prioridad a unos enlaces** sobre otros ⇔ balanceo de carga

# OSPF

- Al conocer toda la red, las **rutas se calculan** usando un **algoritmo de Dijkstra**.
- **A partir de las rutas se construyen las tablas de encaminamiento** de cada router.
- Gestión en base a **áreas independientes de la red**.
- Se minimiza la difusión mediante **routers designados** (son los que envían y reciben el estado de la red).
- **Mejor convergencia**, ya que no hay que hacer cálculos sobre las rutas a difundir.
- Las **actualizaciones** se hacen **sólo cuando hay cambios en la red**.
- Maneja **distintas tablas (BD)**: vecinos, topología, rutas
- Mensajes: *hello, database description, link status request/update/ack*

# Formato Datagrama IP

0	4	8	16	19	31
V	LC	TS	longitud total		
identificación			I	desplazamiento	
TTL		protocolo	comprobación		
dirección IP origen					
dirección IP destino					
opciones				relleno	
datos					

cabecera

The screenshot shows the Wireshark interface with a packet capture filter set to 'ip'. The packet list shows a packet from 88.188.158.15 to 150.214.191.5. The packet details pane shows the following information:

- Frame 215: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
- Ethernet II, Src: Cisco\_b7:64:00 (00:07:0d:b7:64:00), Dst: Micro-St\_a8:f7:63 (00:24:21:a8:f7:63)
- Internet Protocol, Src: 88.188.158.15 (88.188.158.15), Dst: 150.214.191.5 (150.214.191.5)
  - Version: 4
  - Header length: 20 bytes
  - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  - Total Length: 75
  - Identification: 0xe87c (59516)
  - Flags: 0x02 (Don't Fragment)
  - Fragment offset: 0
  - Time to live: 106
  - Protocol: TCP (6)
  - Header checksum: 0xdb88 [correct]
  - Source: 88.188.158.15 (88.188.158.15)
  - Destination: 150.214.191.5 (150.214.191.5)
- Transmission Control Protocol, Src Port: wap-push-http (4035), Dst Port: 23691 (23691), Seq: 36, Ack: 36,
- Data (35 bytes)

# Formato Datagrama IP

0	16			31
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total	
Identificador			Flags	Posición de Fragmento
Tiempo de Vida	Protocolo		Suma de Control de Cabecera	
Dirección IP de Origen				
Dirección IP de Destino				
Opciones			Relleno	

## Versión:

0100 ⇔ 4

## Tamaño cabecera:

En palabras de 32 bits (entre 5 y 15) ⇔ entre 20 y 60 bytes.

## Tipo servicio:

Preferencia de envío (mínimo retardo, máximo rendimiento, mínimo coste).

## Longitud total:

Tamaño en bytes del datagrama completo (incluyendo datos).

# Formato Datagrama IP

0	16		31	
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total	
Identificador		Flags	Posición de Fragmento	
Tiempo de Vida		Protocolo	Suma de Control de Cabecera	
Dirección IP de Origen				
Dirección IP de Destino				
Opciones			Relleno	

**Identificador:**  
Número de orden del paquete en un mensaje.

**Flags:**  
Indican si hay fragmentación.

**Posición fragmento:**  
Desplazamiento del fragmento respecto del paquete original (para reconstruirlo).

# Formato Datagrama IP

0	16		31
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total
Identificador		Flags	Posición de Fragmento
Tiempo de Vida	Protocolo	Suma de Control de Cabecera	
Dirección IP de Origen			
Dirección IP de Destino			
Opciones		Relleno	

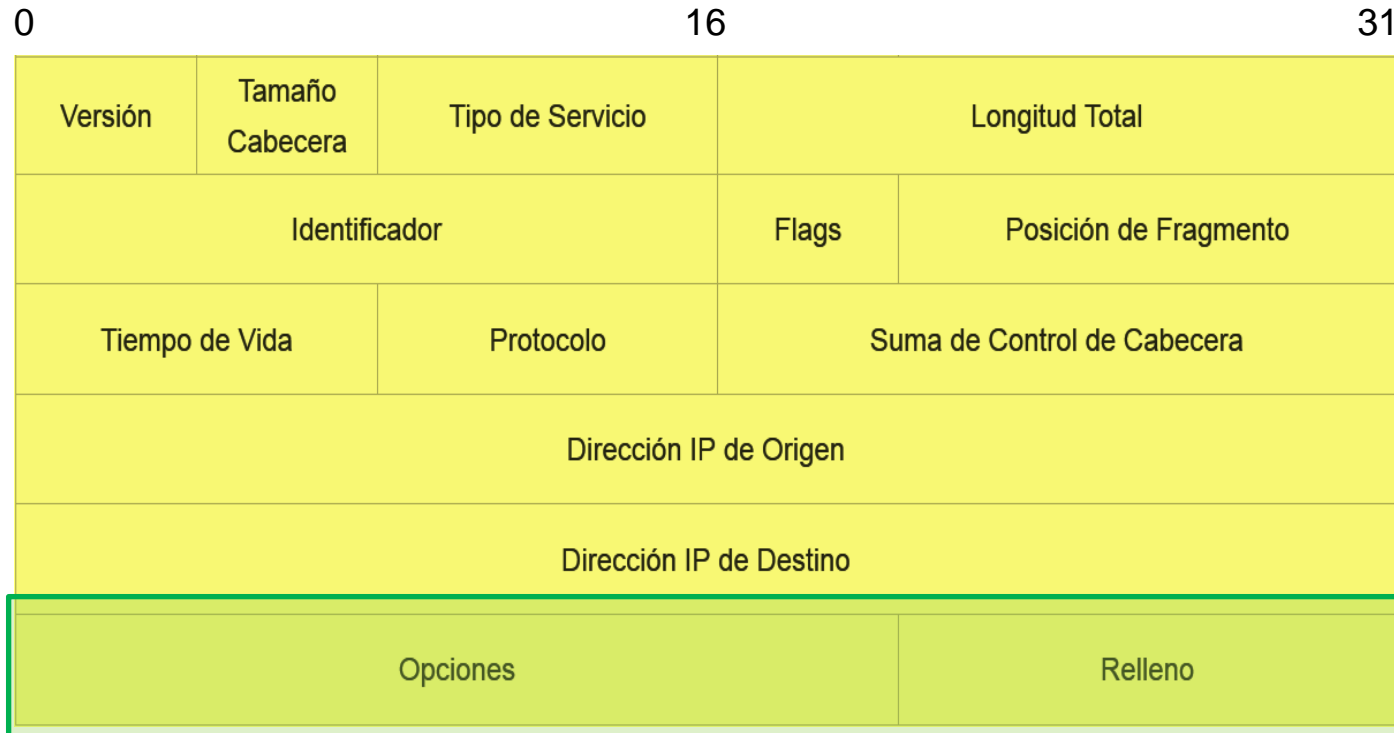
**Tiempo de vida (TTL):**  
Tiempo que puede estar el paquete en una red.

**Protocolo:** (RFC 3232)  
TCP, UDP, ICMP, etc

**Suma de control:**  
Número para comprobar la corrección de la cabecera.



# Formato Datagrama IP



## Opciones:

Hasta 40 bytes.  
Permite hacer funciones de test y depuración sobre la red (sello de tiempo, registro de ruta, etc).

## Relleno:

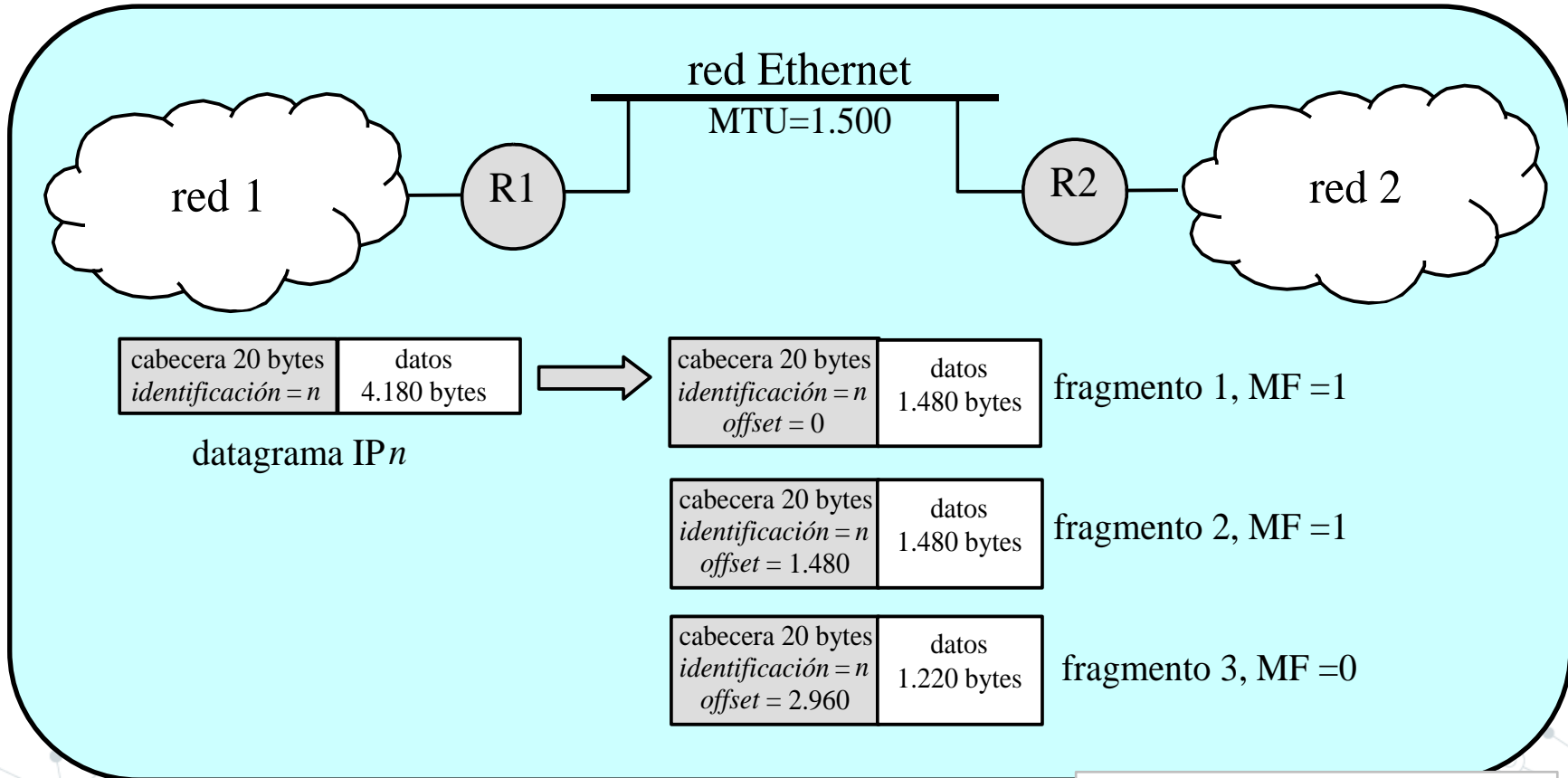
Bits a 0 para completar una palabra de 32 bits en la cabecera.

# Fragmentación IP

- Tamaño máximo:  $2^{16}-1 = 65.535$  bytes.
- Adaptarse a la MTU (Maximum Transfer Unit).
- Ensamblado en destino final:
  - desplazamiento:**  
offset respecto del comienzo del paquete.
  - indicadores (I):**  
“Don't Fragment”, “More Fragments”.

Nivel de enlace	MTU (bytes)
PPP normal	1500
PPP bajo retardo	296
X.25	1600 (RFC 1356)
Frame Relay	1600 (normalmente)
Ethernet DIX	1500
Ethernet LLC-SNAP	1492
Token Ring 4 Mb/s	4440 (THT 8ms)
Classical IP over ATM	9180

# Fragmentación IP



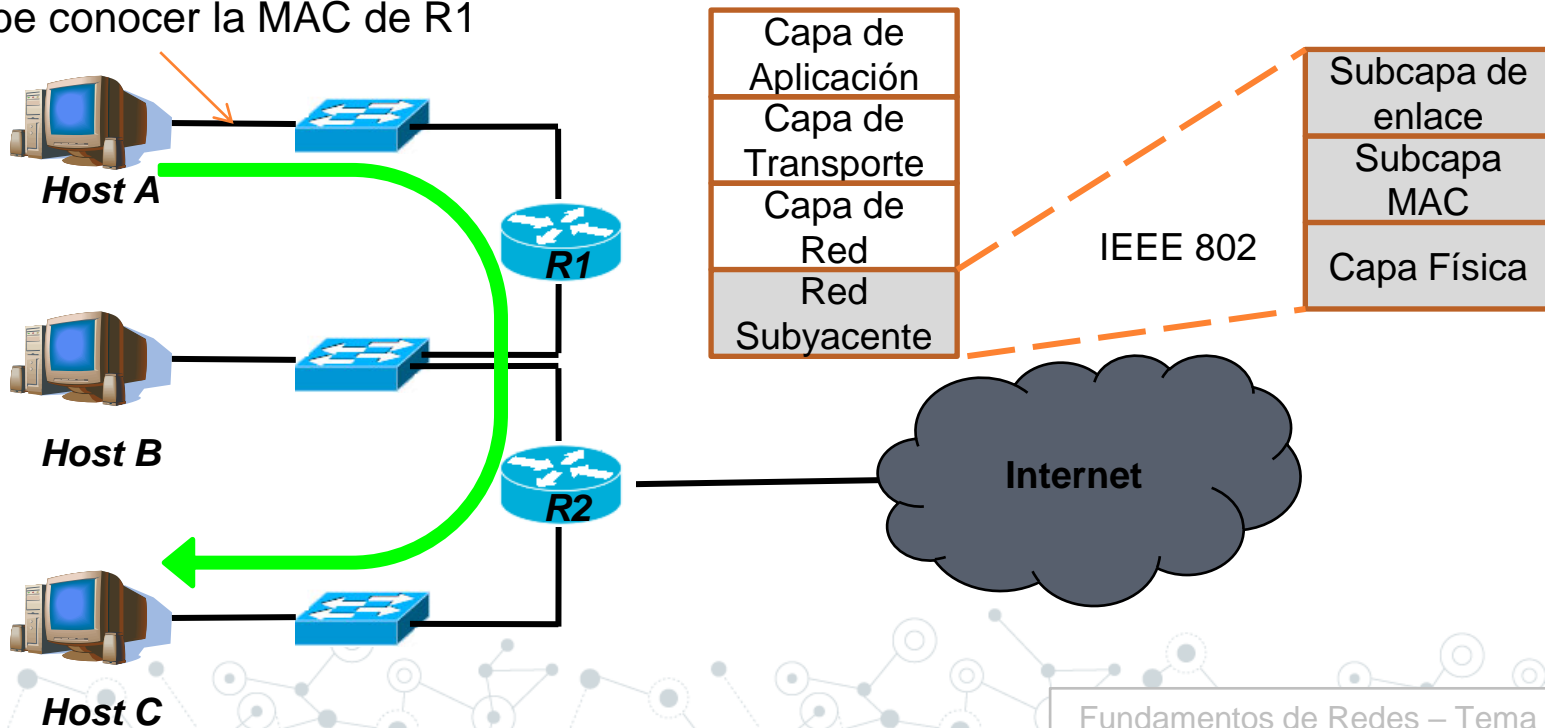
# TEMA 2. Capa de Red

- © 2.1. Funcionalidades
- © 2.2. Conmutación
- © 2.3. El protocolo IP
- © **2.4. Asociación con la capa de enlace: El protocolo ARP**
- © 2.5. El protocolo ICMP
- © 2.6. Cuestiones y ejercicios

# Direcciones MAC

- Para transmisiones a nivel de enlace (físicas).
- Tras la redirección IP → Enviar a la MAC del siguiente nodo

A debe conocer la MAC de R1



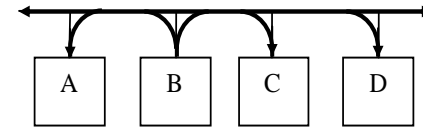
# ARP

- Tras la redirección IP → Enviar a la Medium Access Control (MAC) del siguiente nodo. Se usan en redes Ethernet (cableadas) y Wifi.
- Formato (6 bytes): HH-HH-HH-HH-HH-HH      Ej. 00-24-21-A8-F7-6A
- Son únicas, asignadas por IEEE en lotes de  $2^{24}$  para cada fabricante
- Dirección de difusión (broadcast) FF-FF-FF-FF-FF-FF

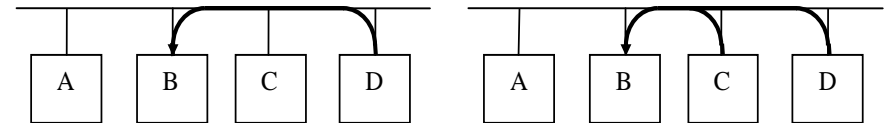
## ARP

- Address Resolution Protocol
- Obtener MAC a partir de IP:

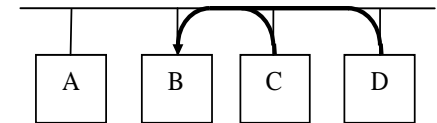
B pregunta MAC de D [(a) y (b)]



(a)



(b)



(c)

## RARP

- Reverse ARP (RARP)

Obtener IP a partir de MAC: (a) y (c)

# ARP

- Formato ARP

0	8	16	31
Htipo		Ptipo	
Hlen	Plen	Operación	
Hemisor (bytes 0-3)			
Hemisor (bytes 4-5)		Pemisor (bytes 0-1)	
Pemisor (bytes 2-3)		Hsol (bytes 0-1)	
Hsol (bytes 2-5)			
Psol (bytes 0-3)			

The image shows a Wireshark packet capture interface. The filter is set to 'arp'. The packet list shows a single packet (No. 6) at time 0.106885, source AsustekC\_a2:68:bd, destination Broadcast, protocol ARP, and info 'who has 150.214.191.10?'. The packet details pane shows the following structure:

- Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
- Ethernet II, Src: AsustekC\_a2:68:bd (90:e6:ba:a2:68:bd), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Address Resolution Protocol (request)
  - Hardware type: Ethernet (0x0001)
  - Protocol type: IP (0x0800)
  - Hardware size: 6
  - Protocol size: 4
  - Opcode: request (0x0001)
  - [Is gratuitous: False]
  - Sender MAC address: AsustekC\_a2:68:bd (90:e6:ba:a2:68:bd)
  - Sender IP address: 150.214.191.178 (150.214.191.178)
  - Target MAC address: 00:00:00\_00:00:00 (00:00:00:00:00:00)
  - Target IP address: 150.214.191.10 (150.214.191.10)

# TEMA 2. Capa de Red

- © 2.1. Funcionalidades
- © 2.2. Conmutación
- © 2.3. El protocolo IP
- © 2.4. Asociación con la capa de enlace: El protocolo ARP
- © **2.5. El protocolo ICMP**
- © 2.6. Cuestiones y ejercicios



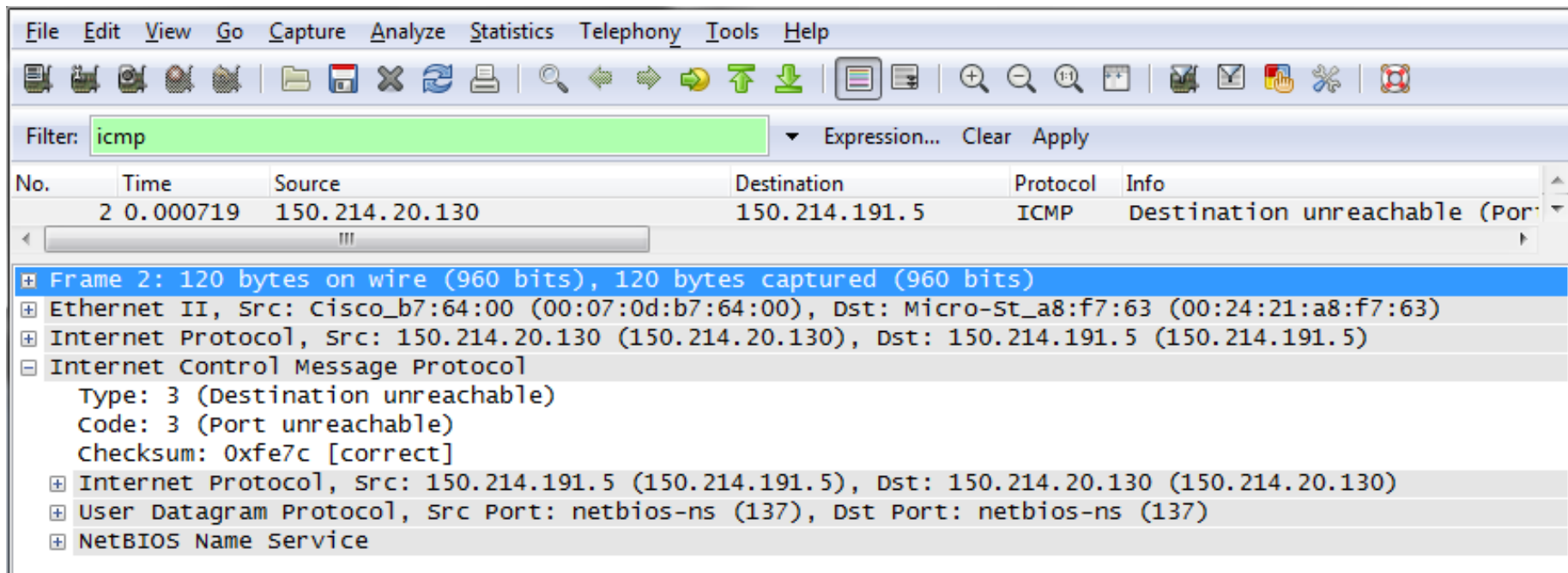
# ICMP

- Internet Control Message Protocol
- Informa sobre situaciones de error en IP → es un protocolo de señalización
- Suelen ir (excepto eco y solicitudes) hacia el origen del datagrama IP original
- ICMP se encapsula en IP
- Cabecera de 32 bits
  - Tipo (8 bits): tipo de mensaje
  - Código (8 bits): subtipo de mensaje
  - Comprobación (16 bits)

0	8	16
tipo	código	comprobación

## Mensaje ICMP

Campo tipo	Mensaje ICMP
8/0	Solicitud/respuesta de eco (ping)
3	Destino inalcanzable
4	Ralentización del origen
5	Redireccionamiento
11	Tiempo de vida excedido
12	Problema de parámetros
13/14	Solicitud/respuesta de sello de tiempo
17/18	Solicitud/respuesta de máscara de red



The image shows a Wireshark packet capture window. The filter bar at the top is set to 'icmp'. The packet list shows a single packet, Frame 2, at time 0.000719, from source 150.214.20.130 to destination 150.214.191.5, protocol ICMP, with info 'Destination unreachable (Port unreachable)'. The packet details pane shows the following structure:

- Frame 2: 120 bytes on wire (960 bits), 120 bytes captured (960 bits)
- Ethernet II, Src: Cisco\_b7:64:00 (00:07:0d:b7:64:00), Dst: Micro-St\_a8:f7:63 (00:24:21:a8:f7:63)
- Internet Protocol, Src: 150.214.20.130 (150.214.20.130), Dst: 150.214.191.5 (150.214.191.5)
- Internet Control Message Protocol
  - Type: 3 (Destination unreachable)
  - Code: 3 (Port unreachable)
  - Checksum: 0xfe7c [correct]
- Internet Protocol, Src: 150.214.191.5 (150.214.191.5), Dst: 150.214.20.130 (150.214.20.130)
- User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)
- NetBIOS Name Service

# ¿Preguntas?

0 comentarios, sugerencias, inquietudes