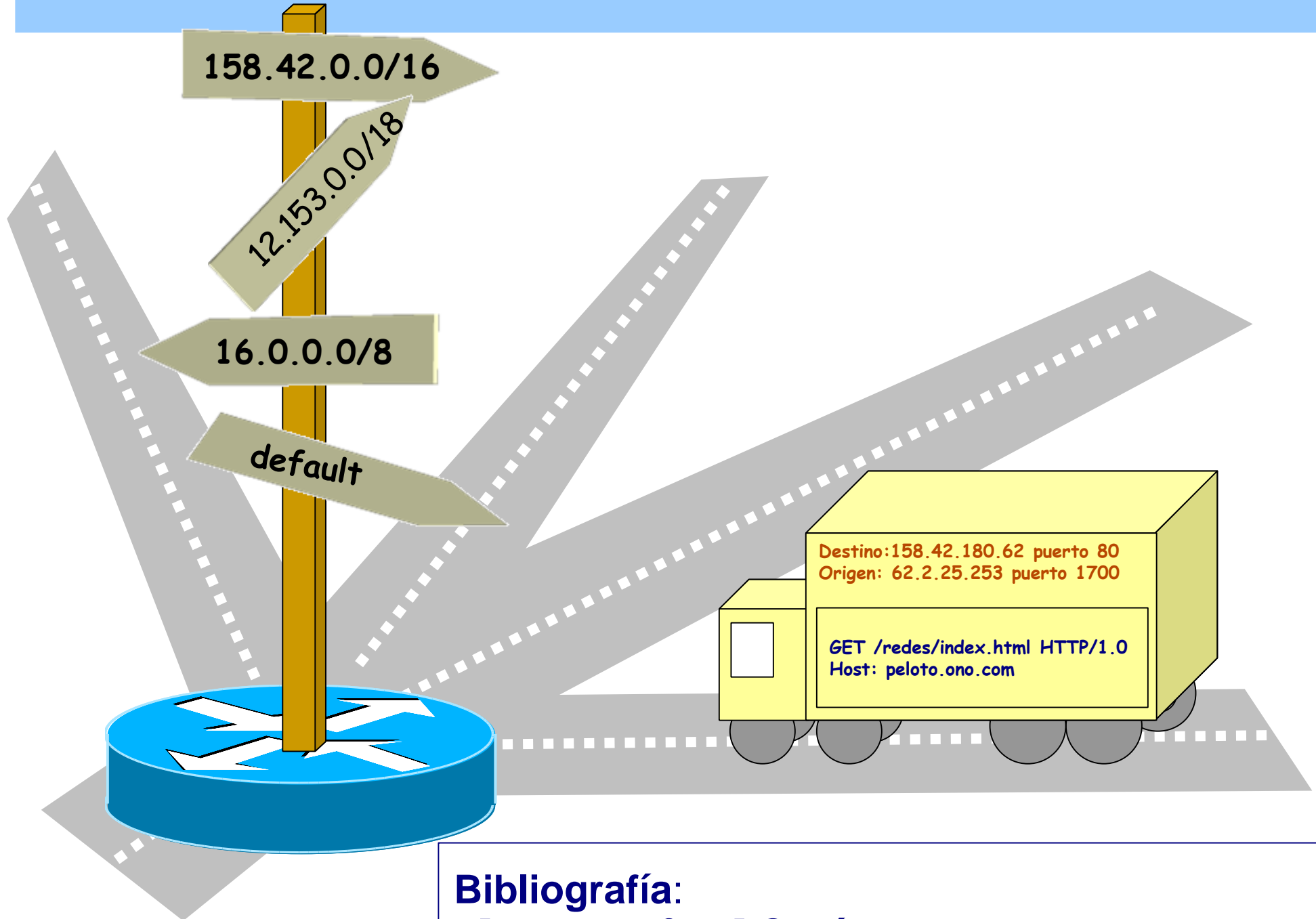


Tema 6: El nivel de red



Bibliografía:

- [Kurose 7ª ed] Capítulos 4 y 5

Objetivos

- Describir los modelos de servicio del nivel de red
- Explicar las características básicas del nivel de red y del protocolo IP:
 - El esquema de direccionamiento empleado actualmente en Internet, sus limitaciones y las soluciones aplicadas
 - Asignación de bloques de direcciones IP de forma eficiente (*subnetting*, *supernetting*)
- Describir cómo se realiza el encaminamiento en Internet
 - Tareas de reenvío/encaminamiento
 - Construcción de la tabla de encaminamiento mínima de dispositivo
 - Algoritmos de enrutamiento
- Explicar los protocolos DHCP, ICMP y el funcionamiento de NAT

Índice (I)

1. Introducción

- Modelos de servicio ofrecidos por el nivel de red
- Reenvío de datagramas

2. El protocolo IP v4

- Formato de un datagrama IP (fragmentación en prácticas)
- Direccionamiento IP v4
- (DHCP, NAT e ICMP en prácticas)

3. Protocolo IP v6

Índice (II)

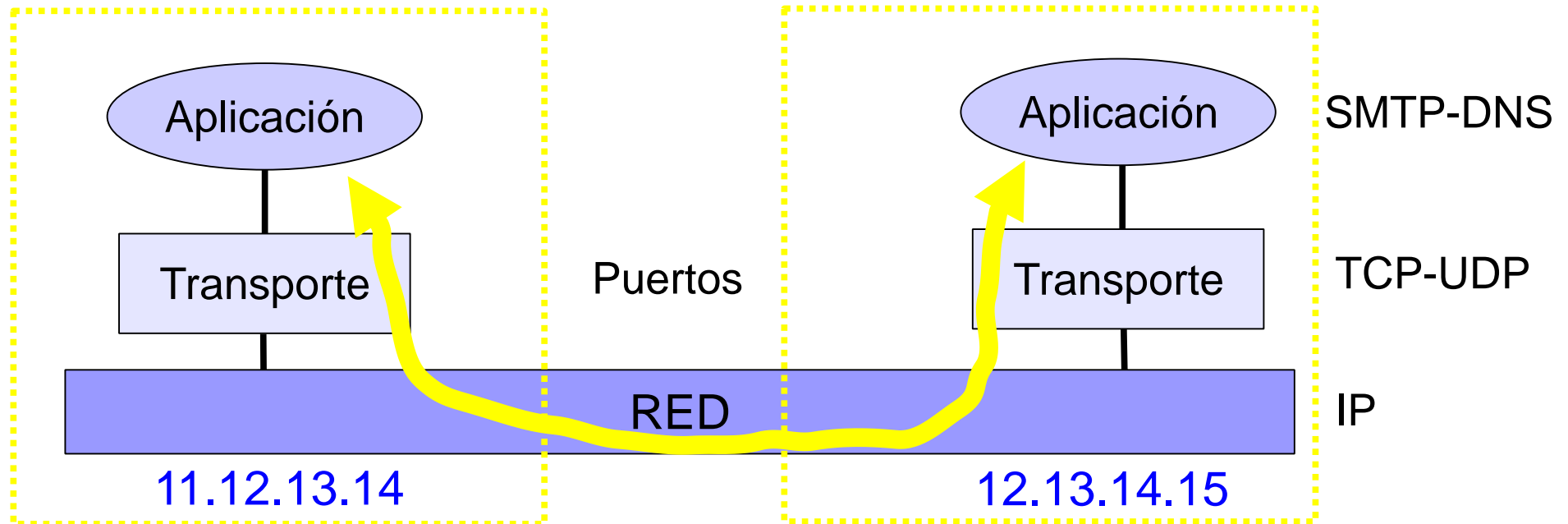
4. Algoritmos de enrutamiento

- Enrutamiento por estado del enlace
- Enrutamiento por vector de distancias

5. Enrutamiento en Internet

- OSPF
- BGP

1. Niveles de transporte y de red



- N. Transporte: se apoya en el **nivel de RED** para llevar los segmentos desde el origen al destino a través de Internet
- Protocolo para el diálogo en el nivel de red: **IP** (*Internet Protocol*)

Servicios del nivel de red

Cuando la capa de transporte le pasa a la de red un paquete, ¿qué puede esperar de ella?

- Servicios a nivel de paquete individual:
 - Entrega garantizada
 - Entrega garantizada con retardo limitado
- Servicios para un flujo de paquetes:
 - Entrega en orden
 - Ancho de banda mínimo garantizado
 - Fluctuación máxima garantizada
 - Servicios de seguridad
 - Confidencialidad, integridad, autenticación, etc.

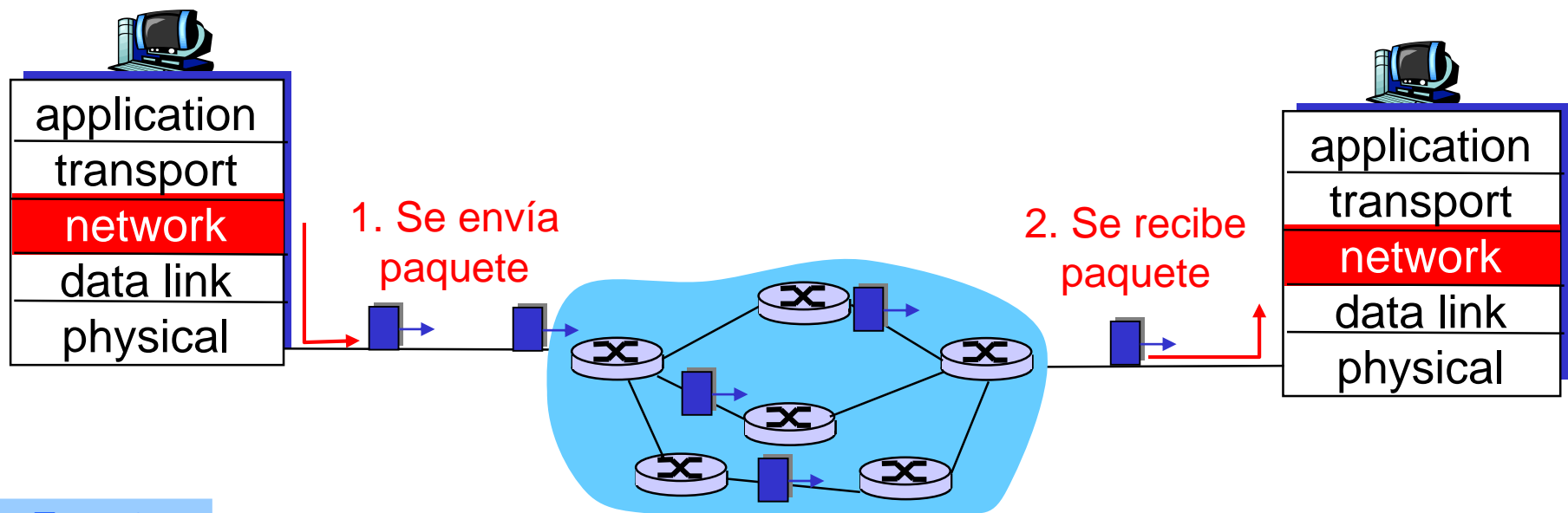
Tipos de servicios

Servicio orientado a la conexión o sin conexión:

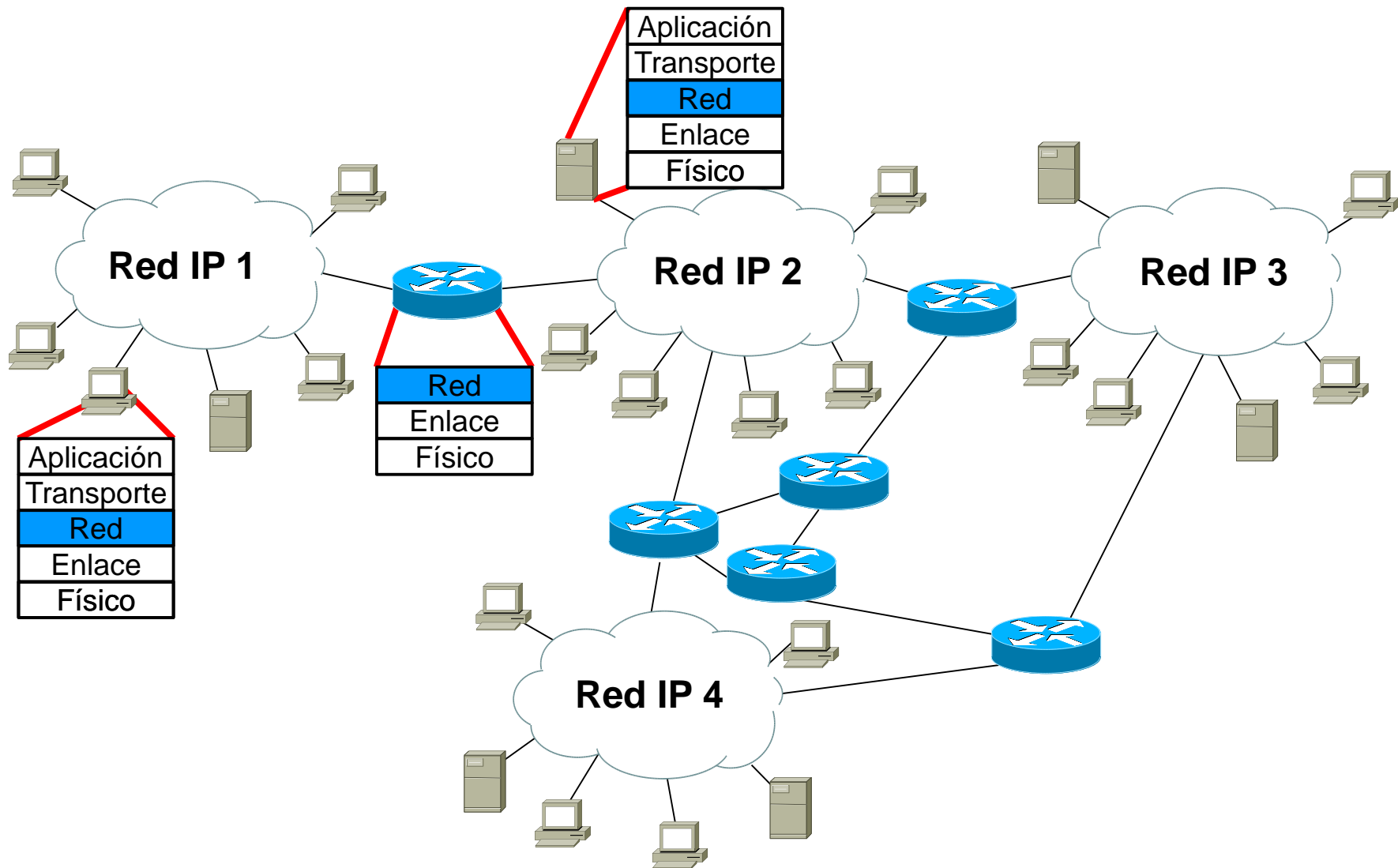
- Servicio con conexión:
 - Antes de enviar paquetes de un ordenador a otro, hay que establecer una conexión a nivel lógico
 - Los routers intermedios intervienen en el proceso
- Servicio sin conexión
 - Redes de datagramas
- Servicio con conexión en nivel de transporte vs. nivel de red
 - Transporte: entre dos procesos
 - Red: entre dos hosts (routers se involucran)

Redes de datagramas

- Cuando se quiere enviar un paquete, simplemente se envía, poniéndole la dirección del destino en la cabecera
- Los routers no guardan estado de la conexión del nivel de transporte
 - A nivel de red no existe el concepto de “conexión”
- Los paquetes se reenvían en los routers usando la dirección del host destino
 - Paquetes entre el mismo par origen-destino pueden seguir rutas diferentes
- Internet es una red de datagramas

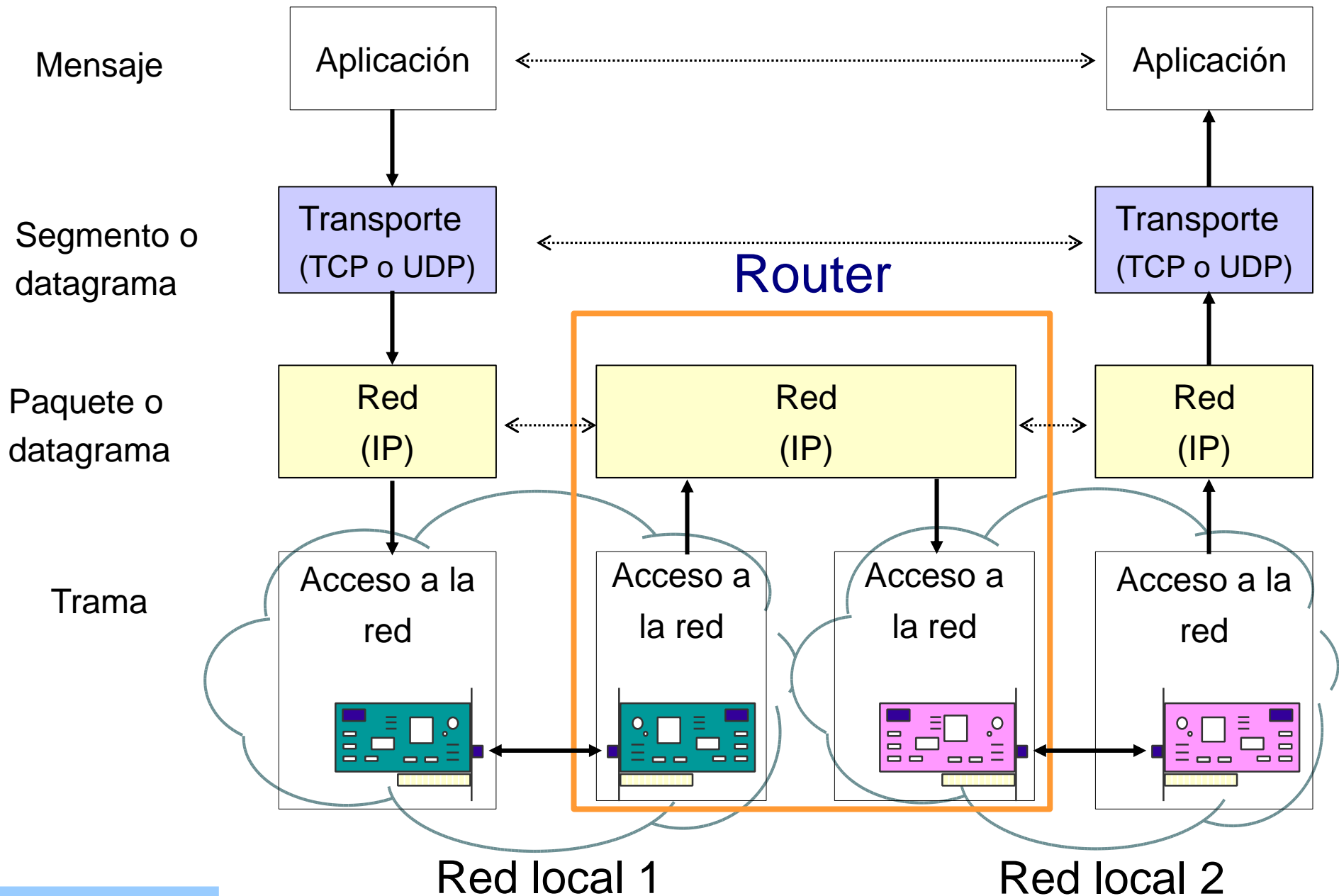


¿Cómo es Internet?



- Cada red IP puede tener distinta tecnología de red

Comunicación mediante la pila de protocolos



Encapsulado de datagramas IP

NIVEL DE APLICACIÓN

Datos aplicación

NIVEL DE TRANSPORTE

Cabecera
(UDP o TCP)

Datos aplicación

NIVEL DE RED

Cabecera
IP

Cabecera
(UDP o TCP)

Datos aplicación

NIVEL DE ENLACE

Cabecera
trama

Cabecera
IP

Cabecera
(UDP o TCP)

Datos aplicación

Cola
trama

Requisitos para la comunicación

- Unidad de intercambio común: **paquetes** (datagramas IP)
- Para poder llevar los paquetes a través de la red se necesita:
 - Identificar mediante direcciones a los sistemas que intervienen en la comunicación (**direcciones IP**)
 - Elegir una ruta en la red que permita alcanzar el destino (**encaminamiento**)
- El nivel de red se encarga de ambos problemas

Dos tareas a nivel de red

- *Forwarding* (reenvío): Un router debe mover los paquetes recibidos desde el enlace de entrada al enlace de salida adecuado
 - Suelen emplearse **tablas de reenvío**
- *Routing* (enrutamiento): Calcular la ruta a tomar por los paquetes de origen a destino.
 - Se calculan mediante **algoritmos de enrutamiento**
- En ocasiones se emplea el término **encaminamiento** indistintamente para ambas tareas

Tabla de encaminamiento

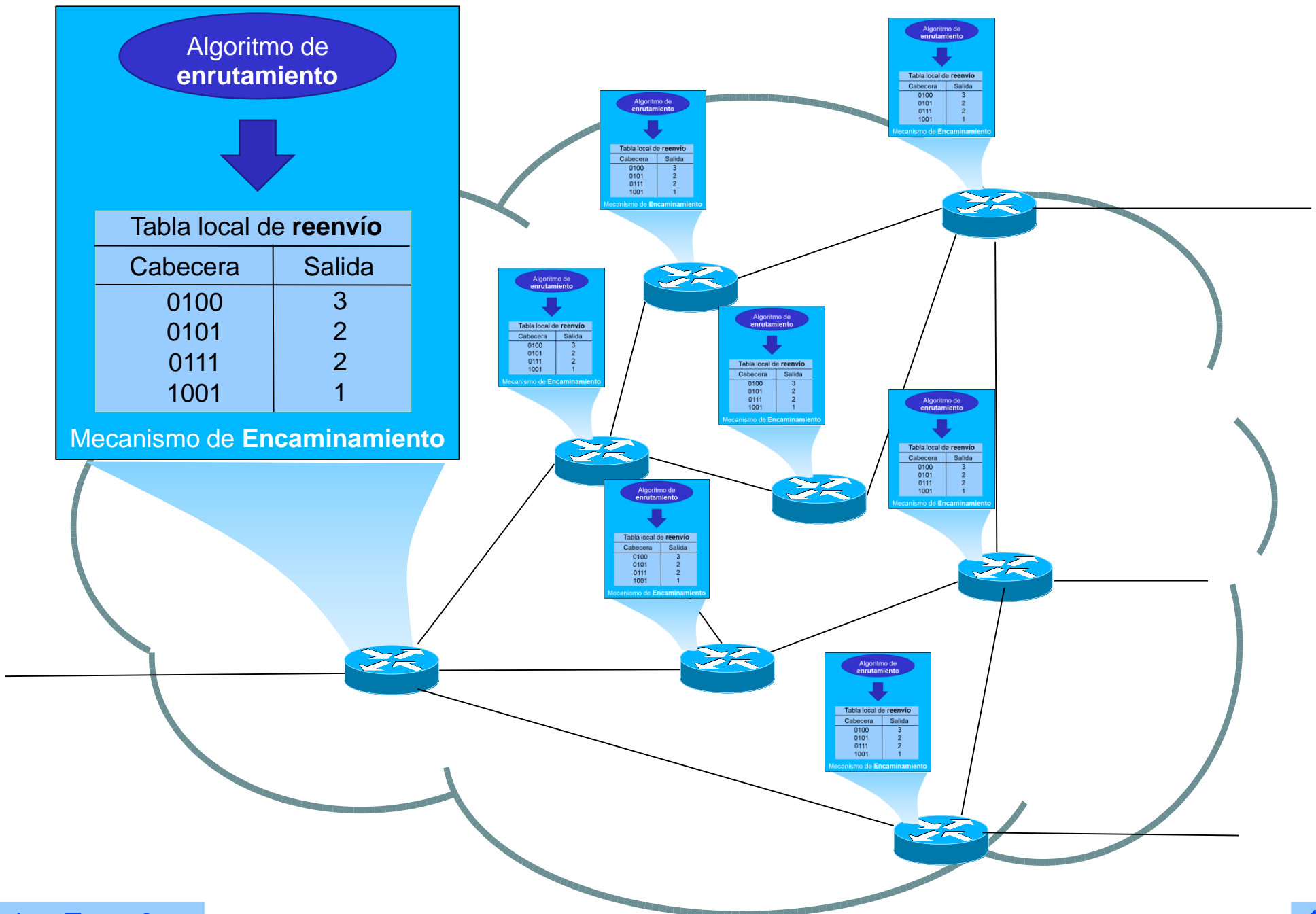


Tabla de encaminamiento

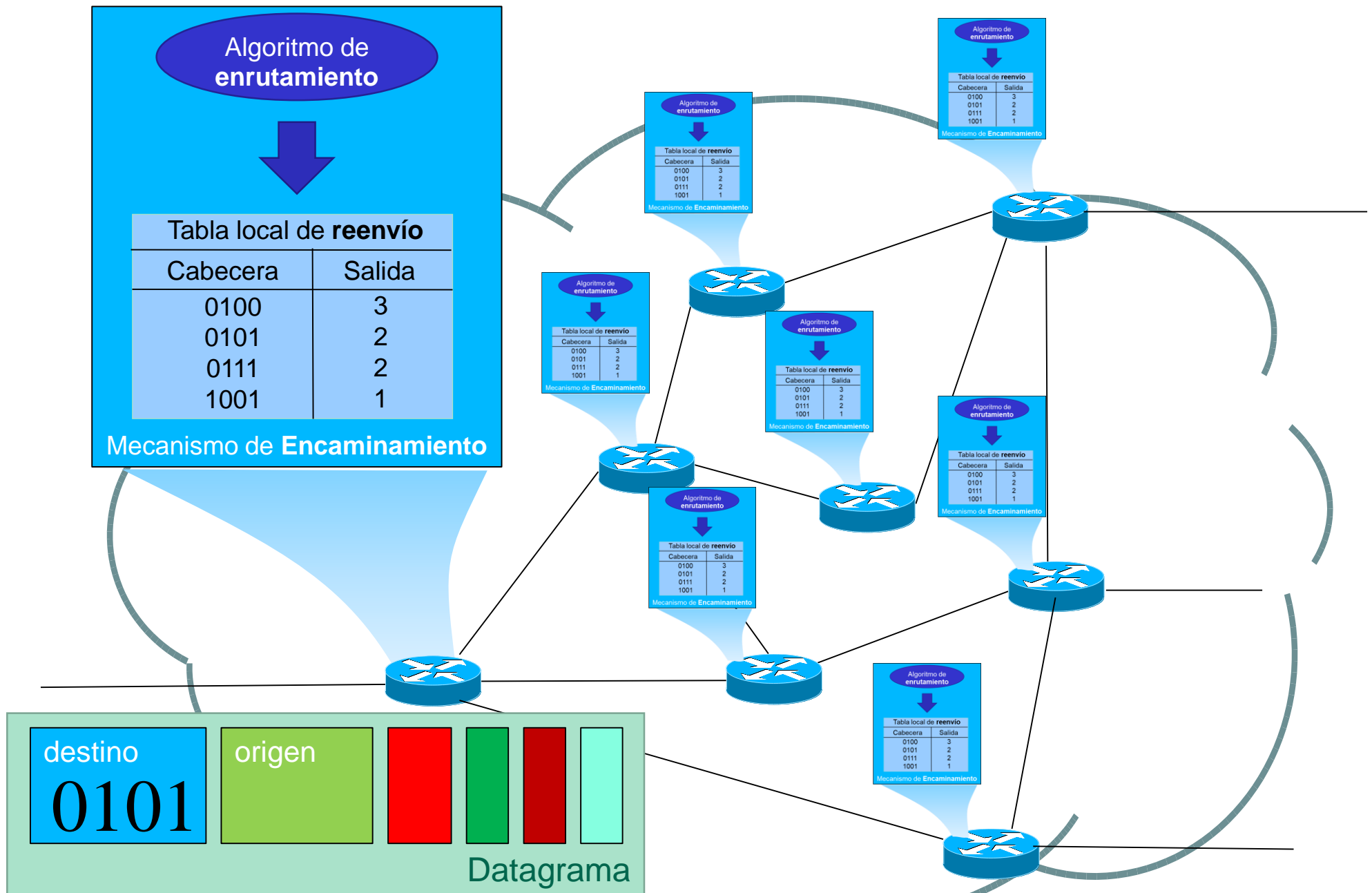


Tabla de encaminamiento

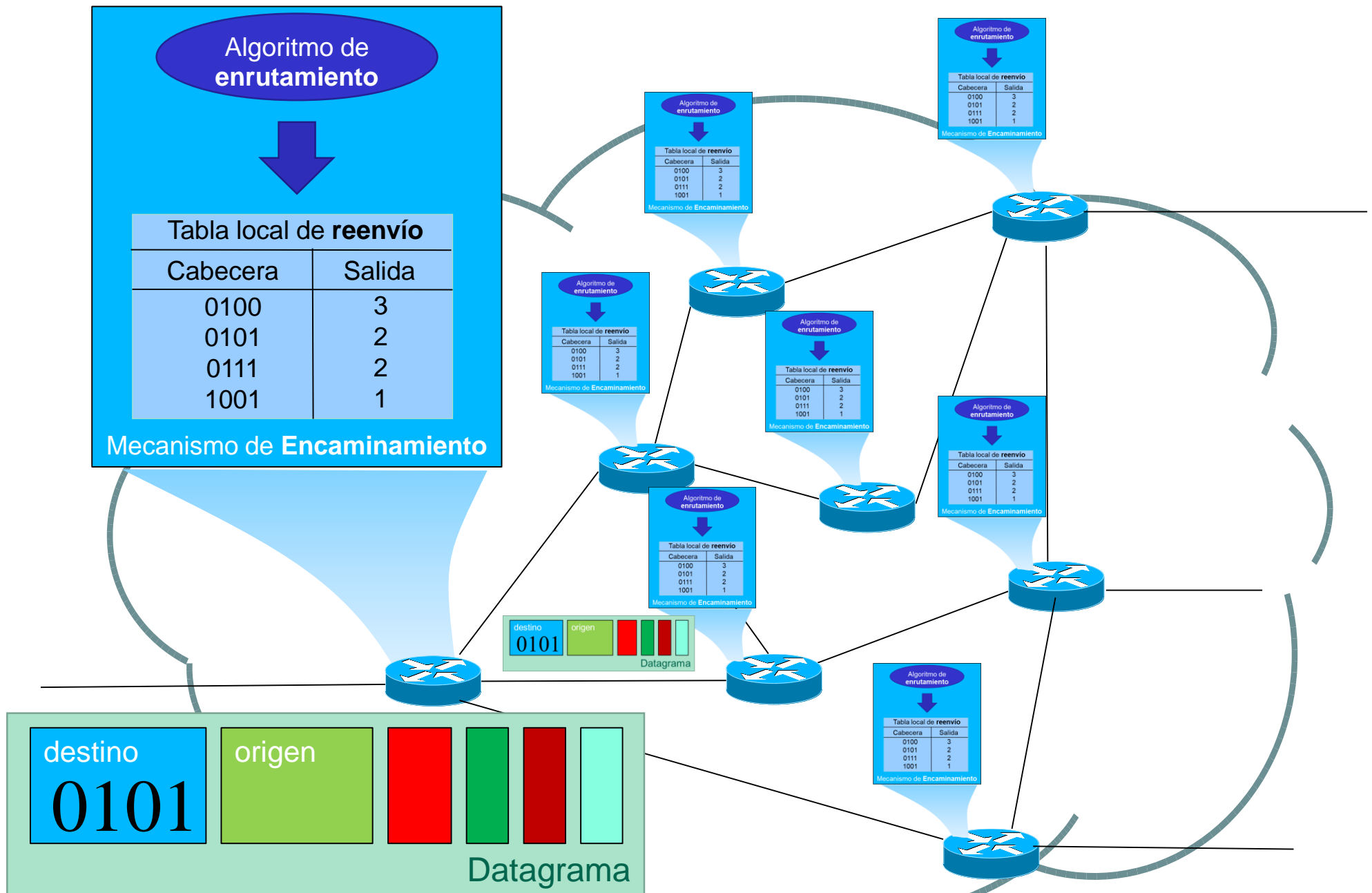


Tabla de encaminamiento

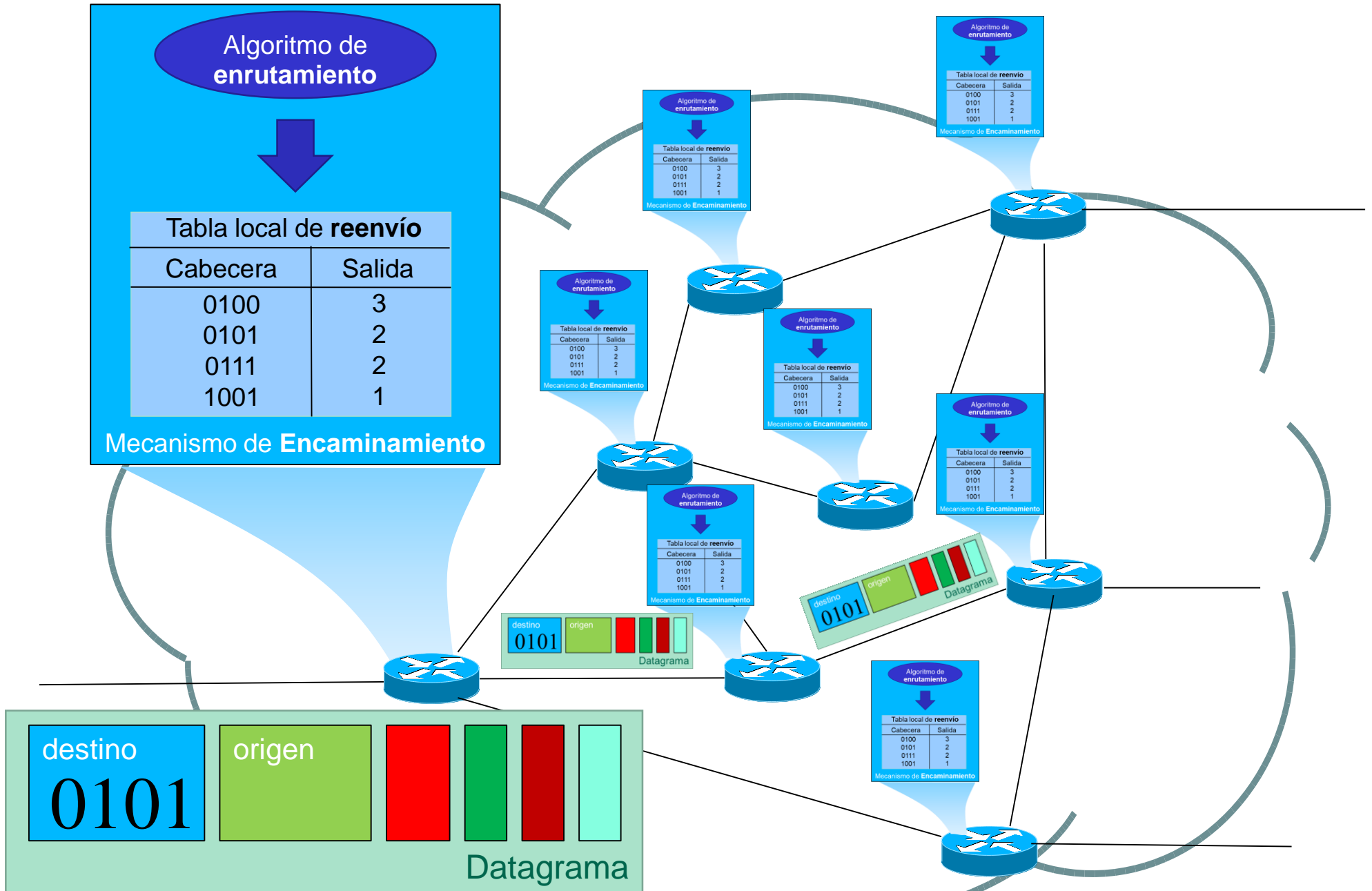
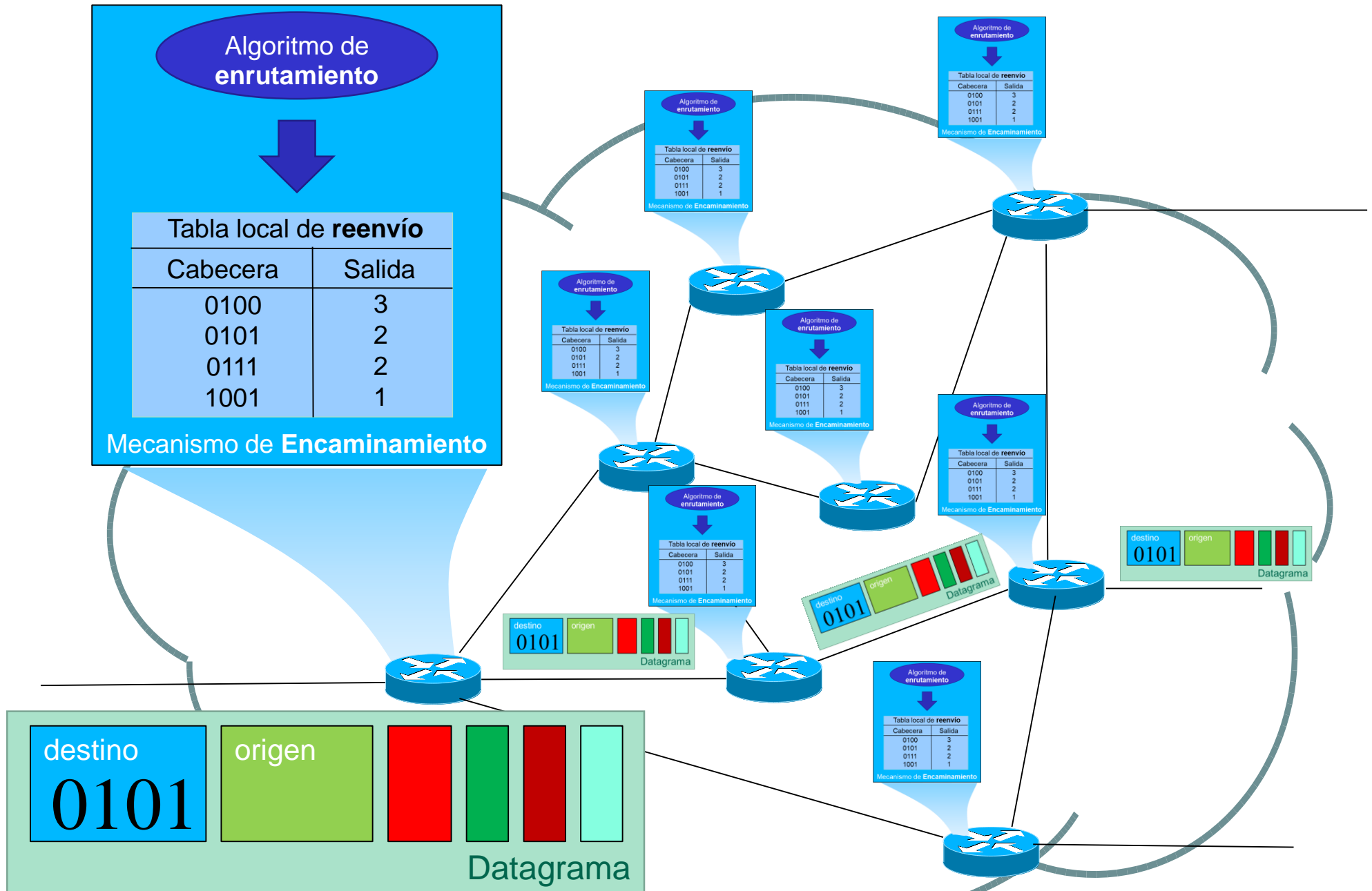
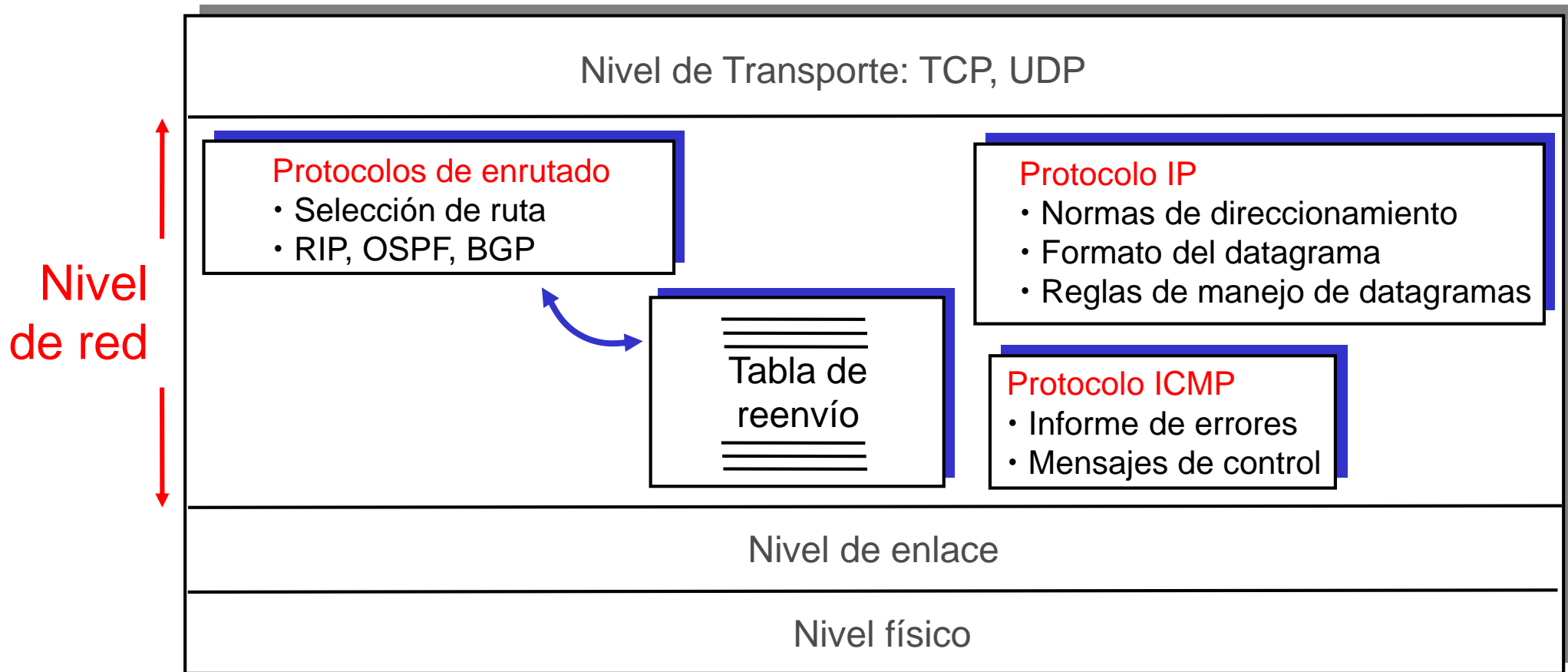


Tabla de encaminamiento



2. IP v4 (*Internet Protocol*)

Funciones del nivel de red:



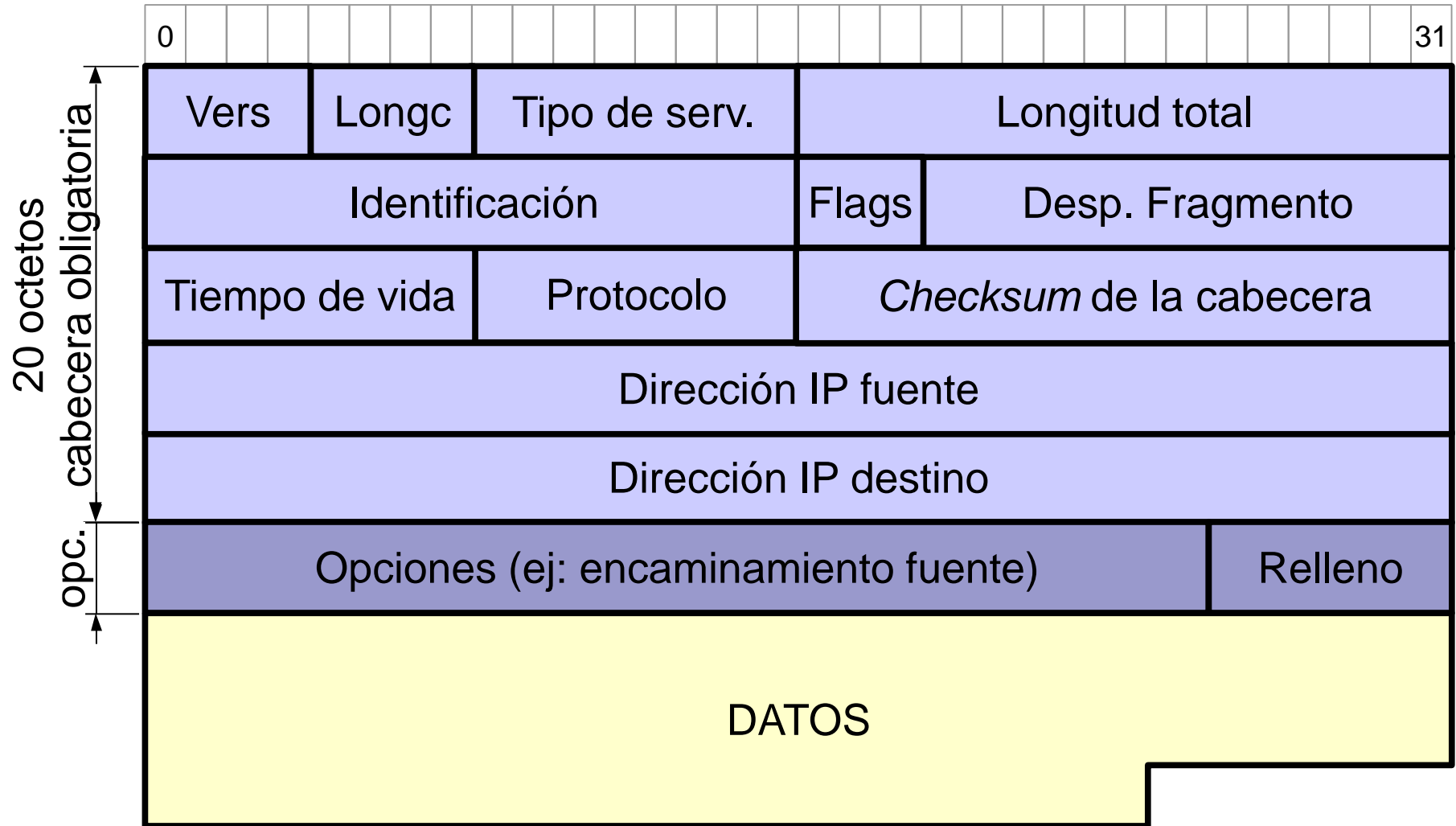
El protocolo IP v4

- IP es el protocolo central de la pila de protocolos TCP/IP
- Unidad básica de transferencia de datos: **datagrama**
- Servicio "best effort"
 - No fiable
 - Sin conexión:
 - No mantiene información de estado sobre los datagramas
 - Cada datagrama se maneja independientemente

Tareas de IP v4

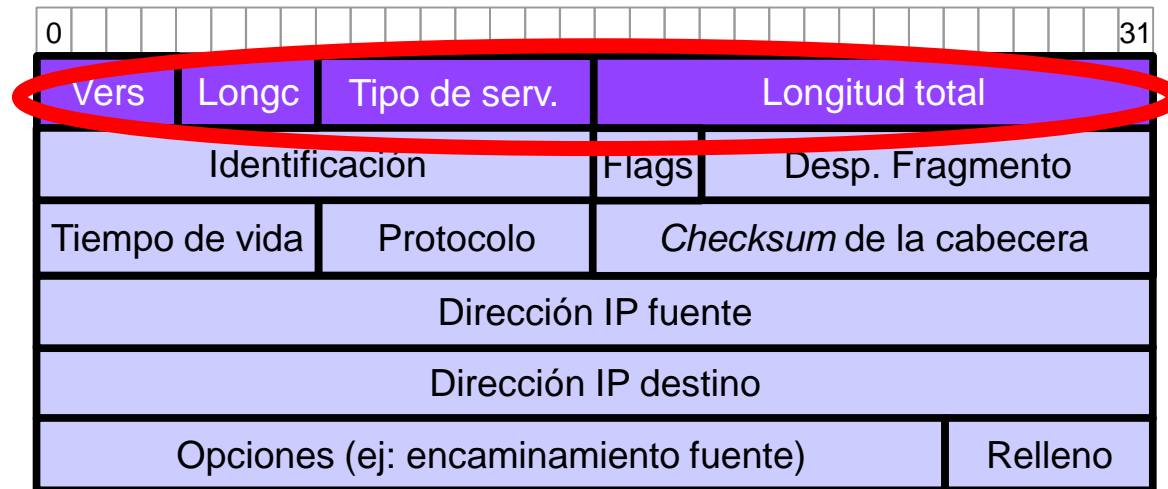
- IP realiza las funciones de encaminamiento
- IP determina las reglas de intercambio de datagramas entre computadores
 - Formato, reglas de procesamiento de datagramas, condiciones de error, etc.

Formato de un datagrama IP



Descripción de los campos

- **Versión del protocolo IP:** (4 bits) actualmente IP v4
 - **Longitud de la cabecera:** (4 bits) tamaño de la cabecera del datagrama en palabras de 32 bits
 - **Longitud total del datagrama**
 - Incluye cabecera y datos
 - Tamaño máximo = 65.535 bytes
-
- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------------|--|--|--|-------|--|--|--|---------------|--|--|--|--|--|--|--|---------------------|--|--|--|--------------|--|--|--|---|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vers | | | | Longc | | | | Tipo de serv. | | | | | | | | Longitud total | | | | | | | | | | | | | | | |
| Identificación | | | | | | | | | | | | | | | | Flags | | | | Desp. Fragme | | | | | | | | | | | |
| Tiempo de vida | | | | | | | | Protocolo | | | | | | | | Checksum de la cabe | | | | | | | | | | | | | | | |
| Dirección IP fuente | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dirección IP destino | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opciones (ej: encaminamiento fuente) | | | | | | | | | | | | | | | | | | | | | | | | R | | | | | | | |



Tipo de servicio (I)

- **Tipo de servicio (TOS):**

- 3 bits para la prioridad, 4 bits para el tipo de servicio y un bit a cero.
- Los bits de tipo de servicio permiten al usuario solicitar las condiciones deseadas (Solo un bit a 1)
 - No se garantiza el tipo de servicio solicitado

Minimizar el retardo	1000
Maximizar la productividad	0100
Maximizar la fiabilidad	0010
Minimizar el coste	0001

Tipo de servicio (II)

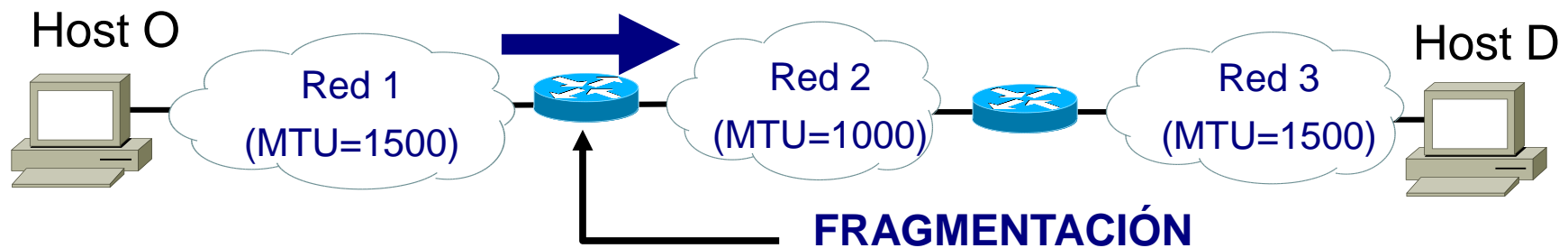
- Se recomienda el uso de los siguientes valores para el tipo de servicio, dependiendo de la aplicación:

Protocolo	TOS	Descripción
Telnet, SSH	1000	Minimizar retardo
Sesión control FTP	1000	Minimizar retardo
Sesión de datos FTP	0100	Maximizar productividad
NNTP	0001	Minimizar coste
IGP	0010	Maximizar fiabilidad

INFORMATIVO

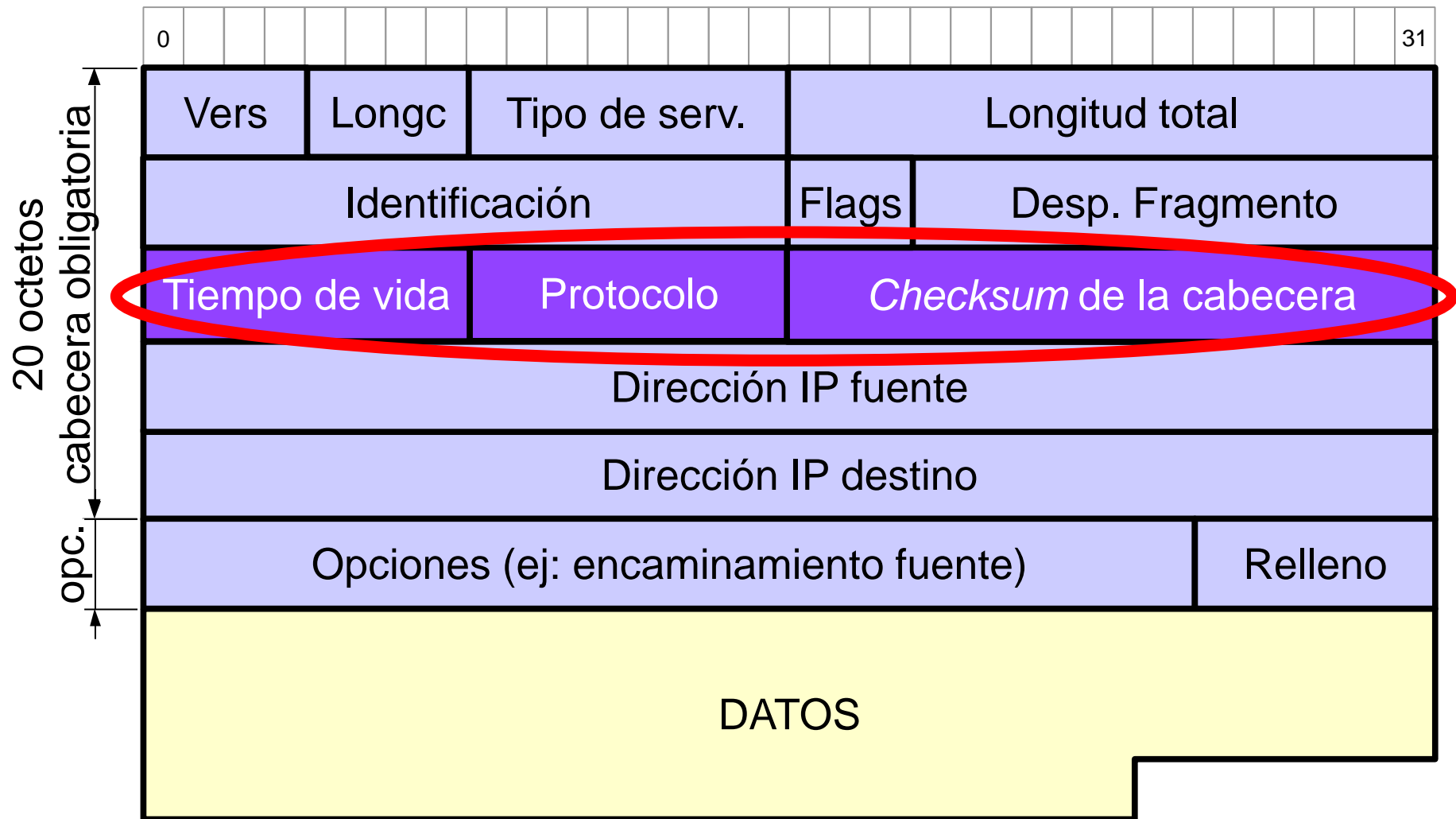
Fragmentación (práctica 2)

- En el nivel de enlace cada protocolo maneja un tamaño máximo de trama, limitando el tamaño de su campo de datos: **MTU** (*Maximum Transfer Unit*)
- Internet utiliza tecnologías heterogéneas
 - Utiliza múltiples MTUs



0																																31									
Vers				Longc				Tipo de serv.				Longitud total																													
Identificación												Flags				Desp. Fragmento																									
Tiempo de vida								Protocolo								Checksum de la cabecera																									
Dirección IP fuente																																									
Dirección IP destino																																									
Opciones (ej: encaminamiento fuente)																								Relleno																	

Otros campos de la cabecera IP



Tiempo de vida (TTL*)

- Los datagramas tienen un tiempo limitado de permanencia en la internet
- El TTL se inicializa en origen y se decrementa cada vez que el datagrama atraviesa un router
 - Valor inicial recomendado^{**}: 64
- Al llegar a **cero** el datagrama se **descarta**

(*) *Time to live*

(**) RFC1700: Assigned Numbers

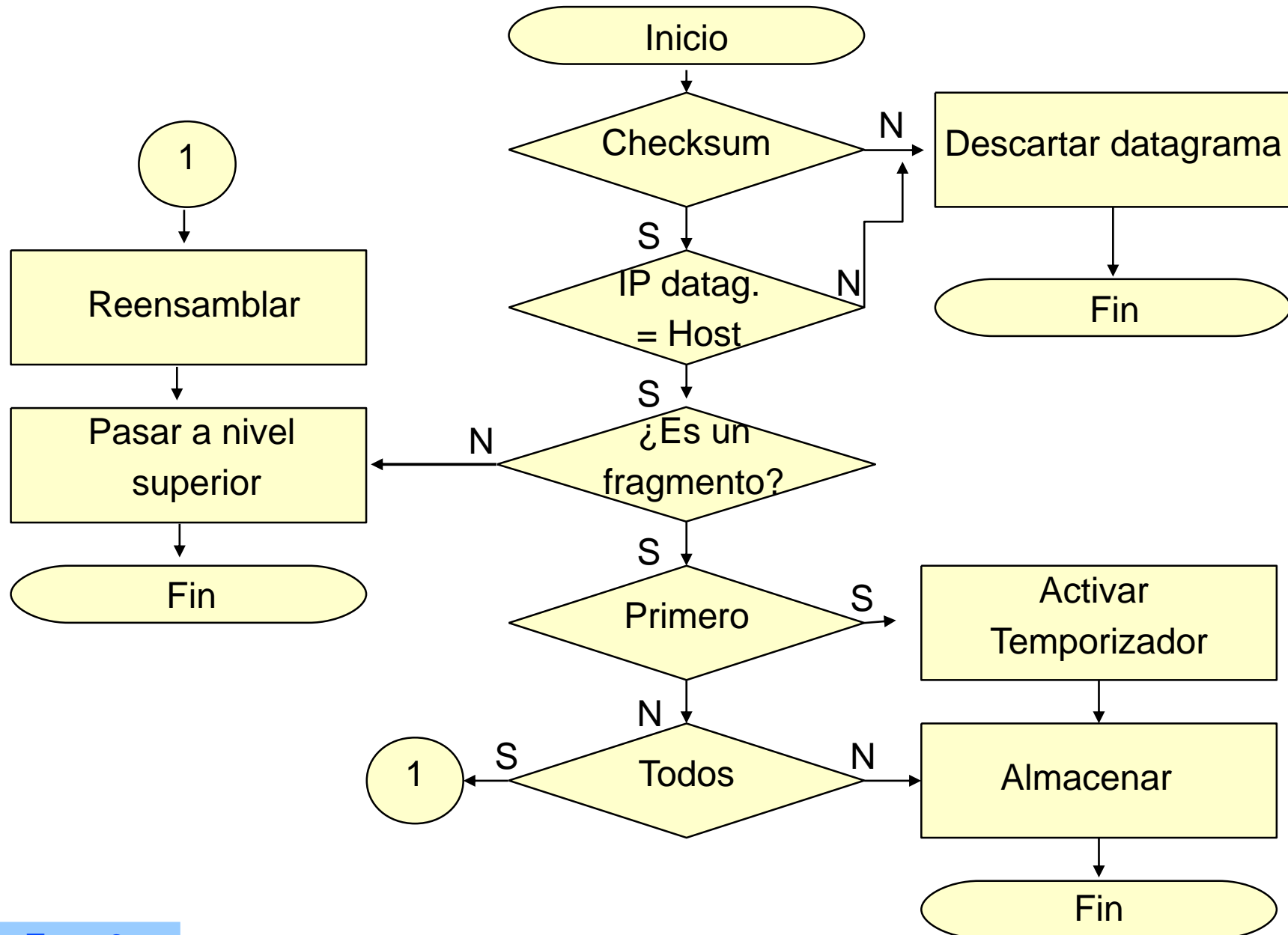
Otros campos de la cabecera IP (II)

- El campo **protocolo** indica el protocolo al que corresponden los datos:

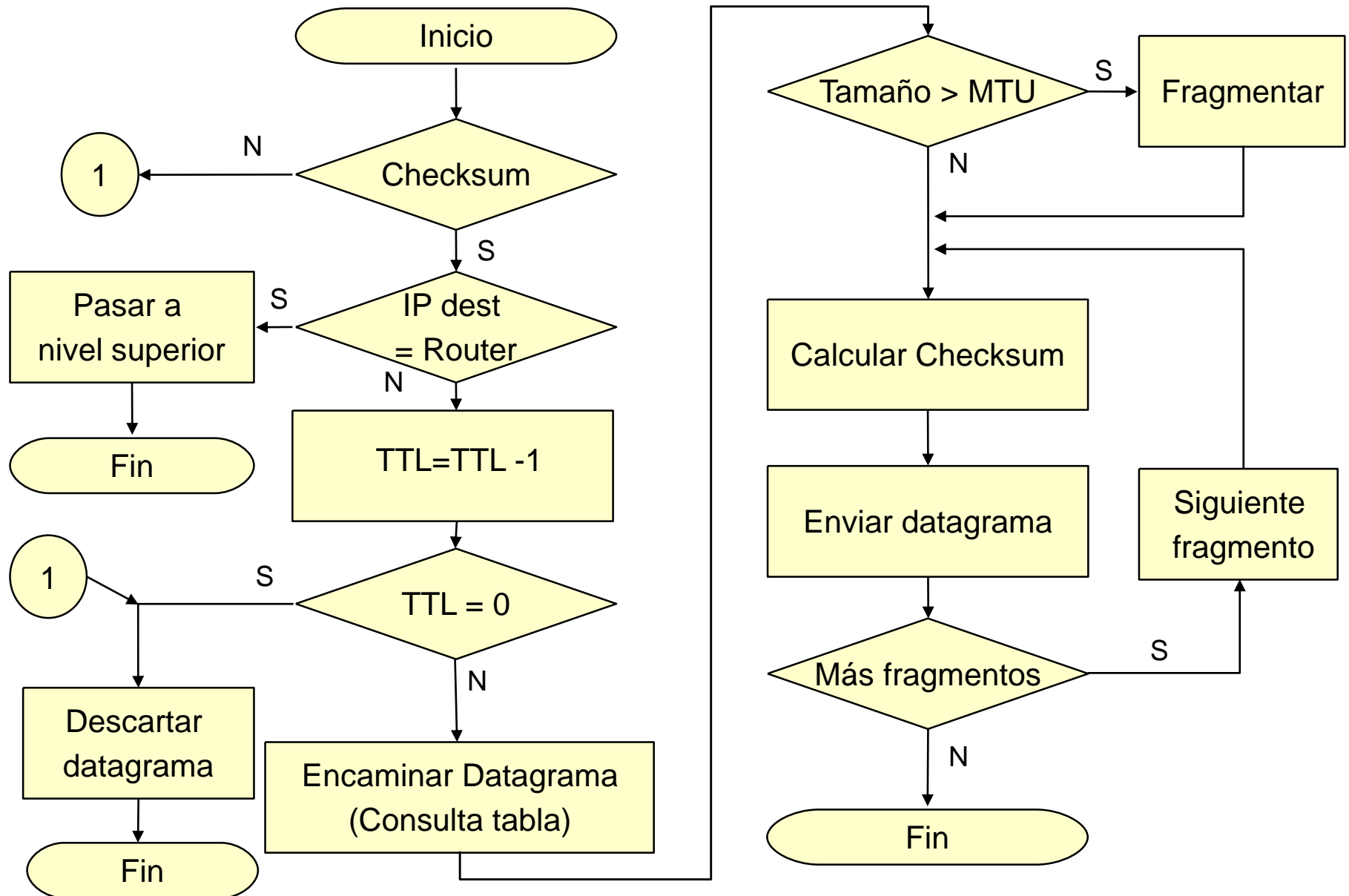
Protocolo	Valor del campo protocolo
TCP	6
ICMP	1
UDP	17

- El **checksum** incluye sólo la cabecera
 - ¿Qué pasa al atravesar un router?
- El campo de **opciones** se utiliza raramente
 - Permite especificar: encaminamiento fuente (uso de una ruta escogida), confidencialidad del datagrama, registro de la ruta (RR), etc.
 - Tiene longitud variable

Procesamiento de un datagrama IP en un computador



Procesamiento de un datagrama IP en un router



Direcciones IP v4: introducción (I) [K 4.3.3]

- Identifican un punto de conexión en una red, una máquina puede tener varias direcciones distintas
- Son direcciones virtuales (interpretadas sólo por el software)
- Tamaño fijo: 32 bits

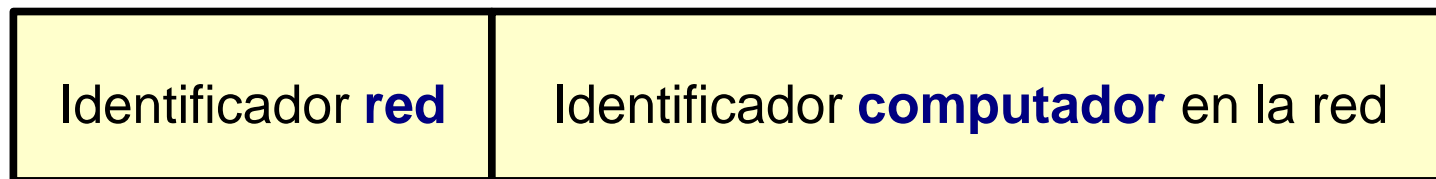
Direcciones IP v4: introducción (II)

- Se representan como cuatro números decimales obtenidos de los cuatro octetos que forman la dirección IP (**n1.n2.n3.n4**)

- *Ejemplo:*

- 10000000 00001010 00000010 00011110
- Se representa como **128.10.2.30**

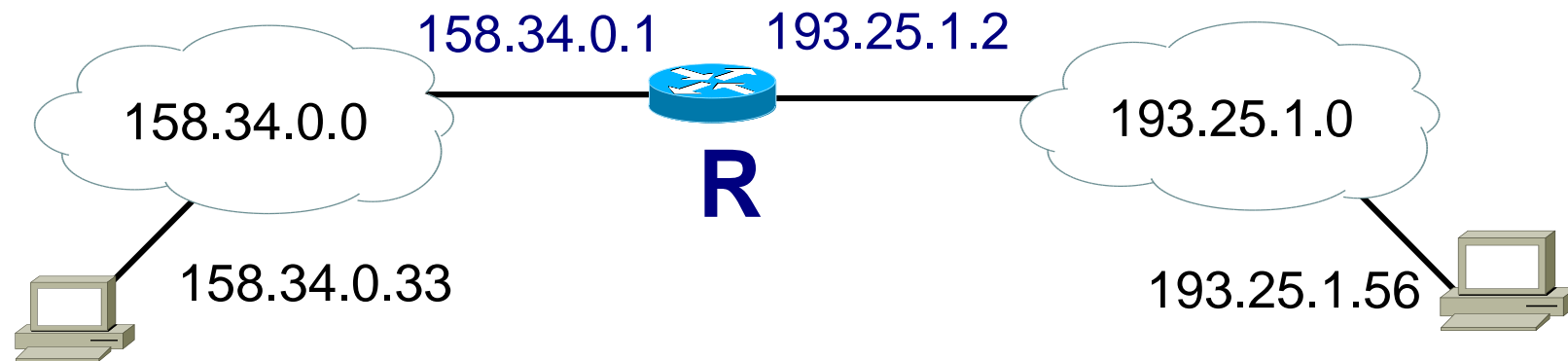
- Cada dirección IP tiene dos campos:



- Todos los sistemas (hosts y routers) conectados a la misma red IP comparten el mismo identificador de red
- **Cuestión:** ¿Este direccionamiento es plano o jerárquico?

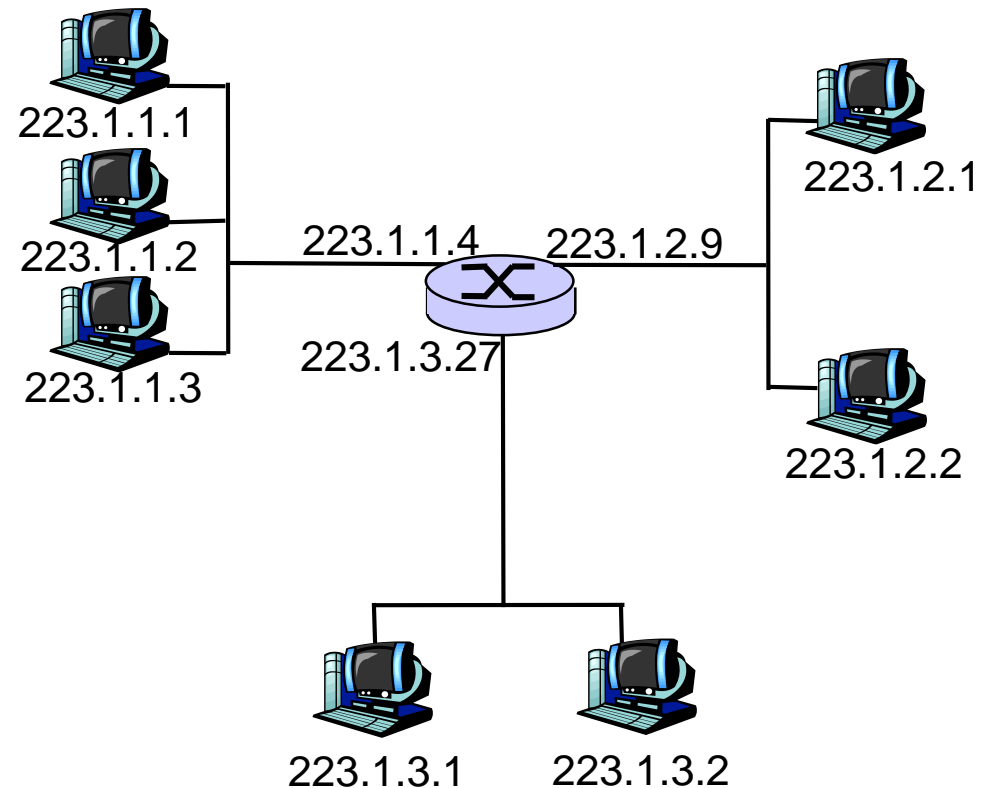
Routers y direccionamiento IP

- Cada router tiene al menos dos direcciones IP:
 - Tiene conexión a varias redes IP (mínimo dos)
 - Cada dirección IP tiene un identificador de red distinto



Direcciones IP y redes IP

- ¿Qué es una red IP?
 - Una red IP está formada por los interfaces de red que tienen el mismo identificador de red en su dirección IP
 - Los dispositivos de una misma red IP pueden intercambiar datagramas sin la intervención de un router

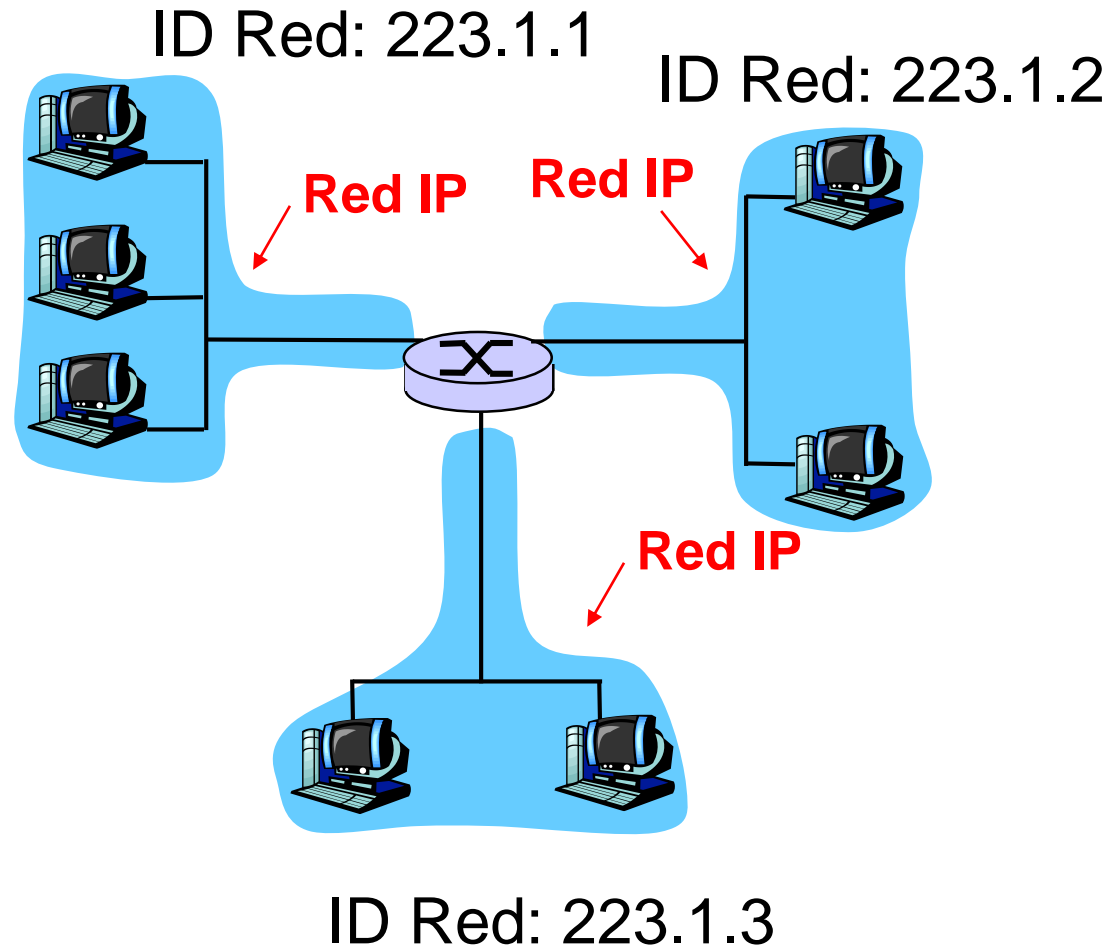


Red compuesta por tres redes IP

Direcciones IP y redes IP

Receta:

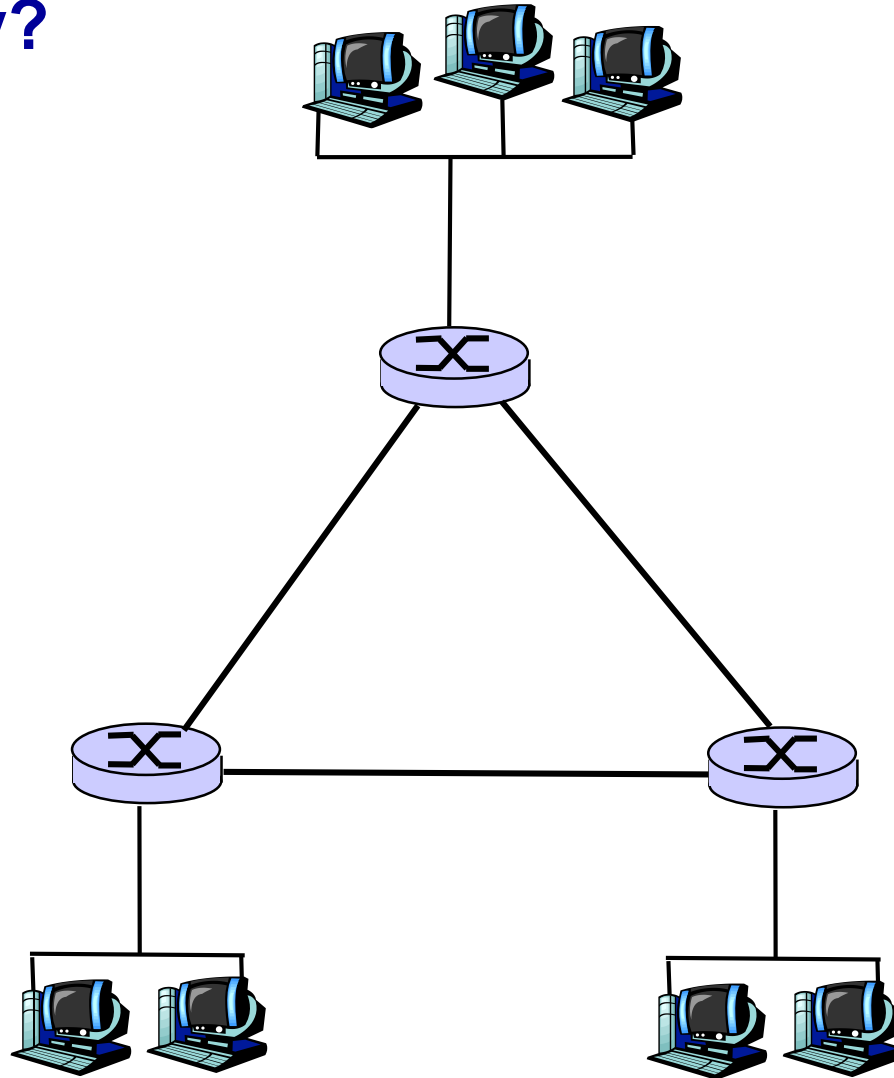
- Para determinar las diferentes redes IP, se separa cada interfaz de su correspondiente router, creando islas o redes aisladas
- Cada red aislada es una red IP diferente



Este identificador de red es de 24 bits

Direcciones IP y redes IP

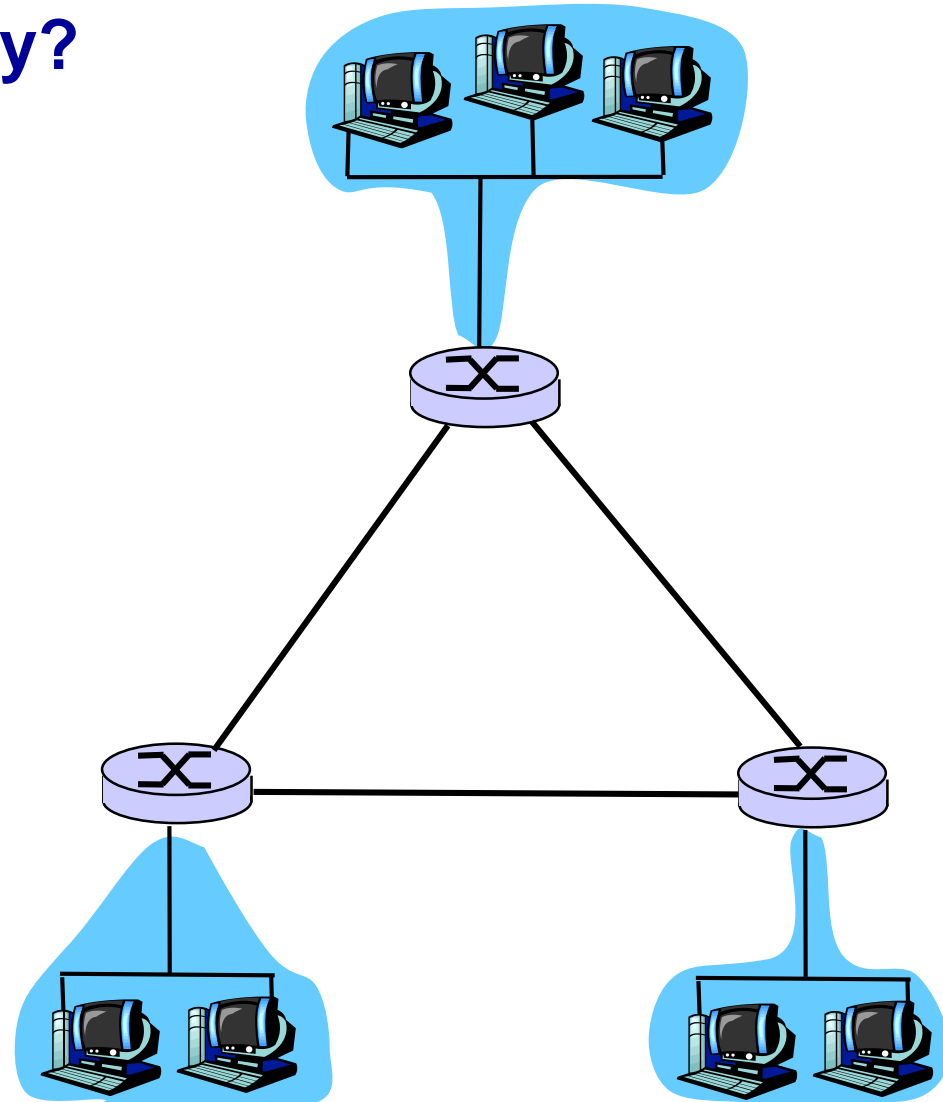
¿Cuántas redes IP hay?



Direcciones IP y redes IP

¿Cuántas redes IP hay?

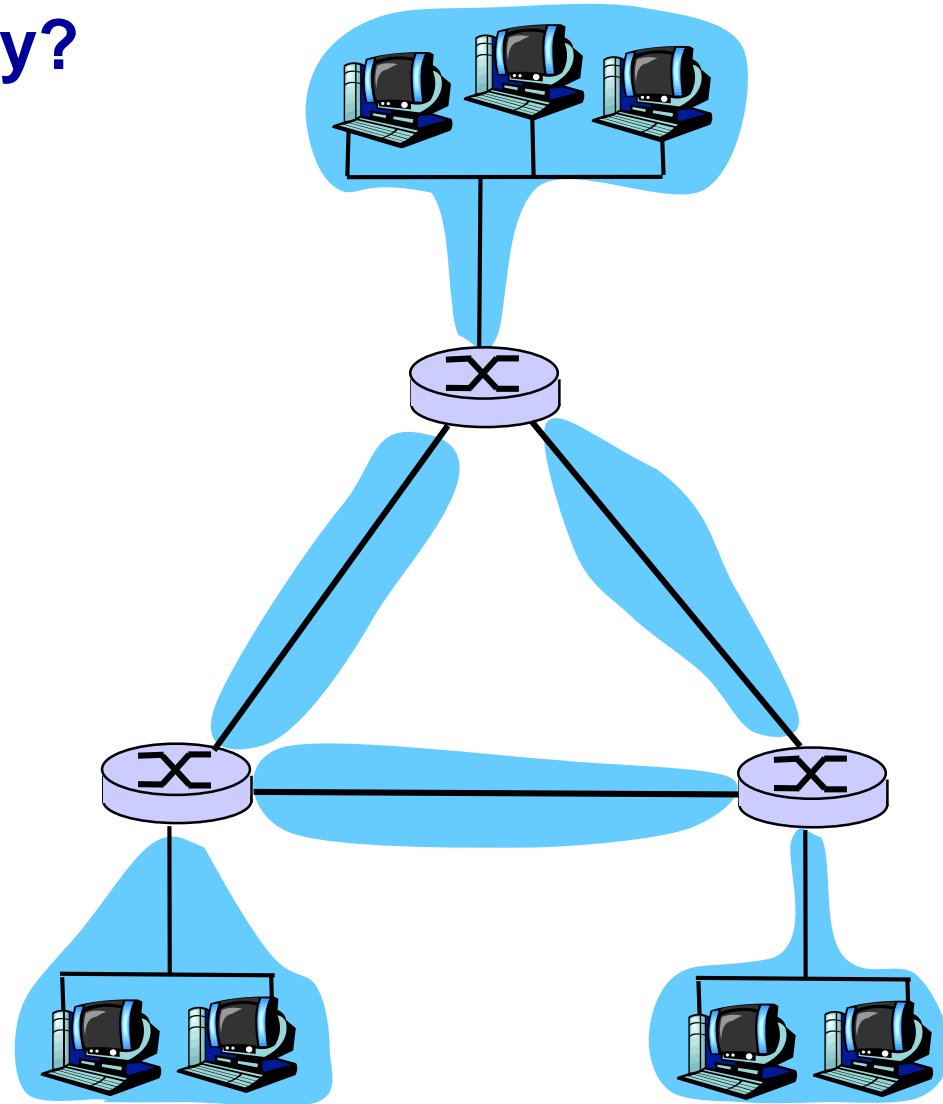
¿3?



Direcciones IP y redes IP

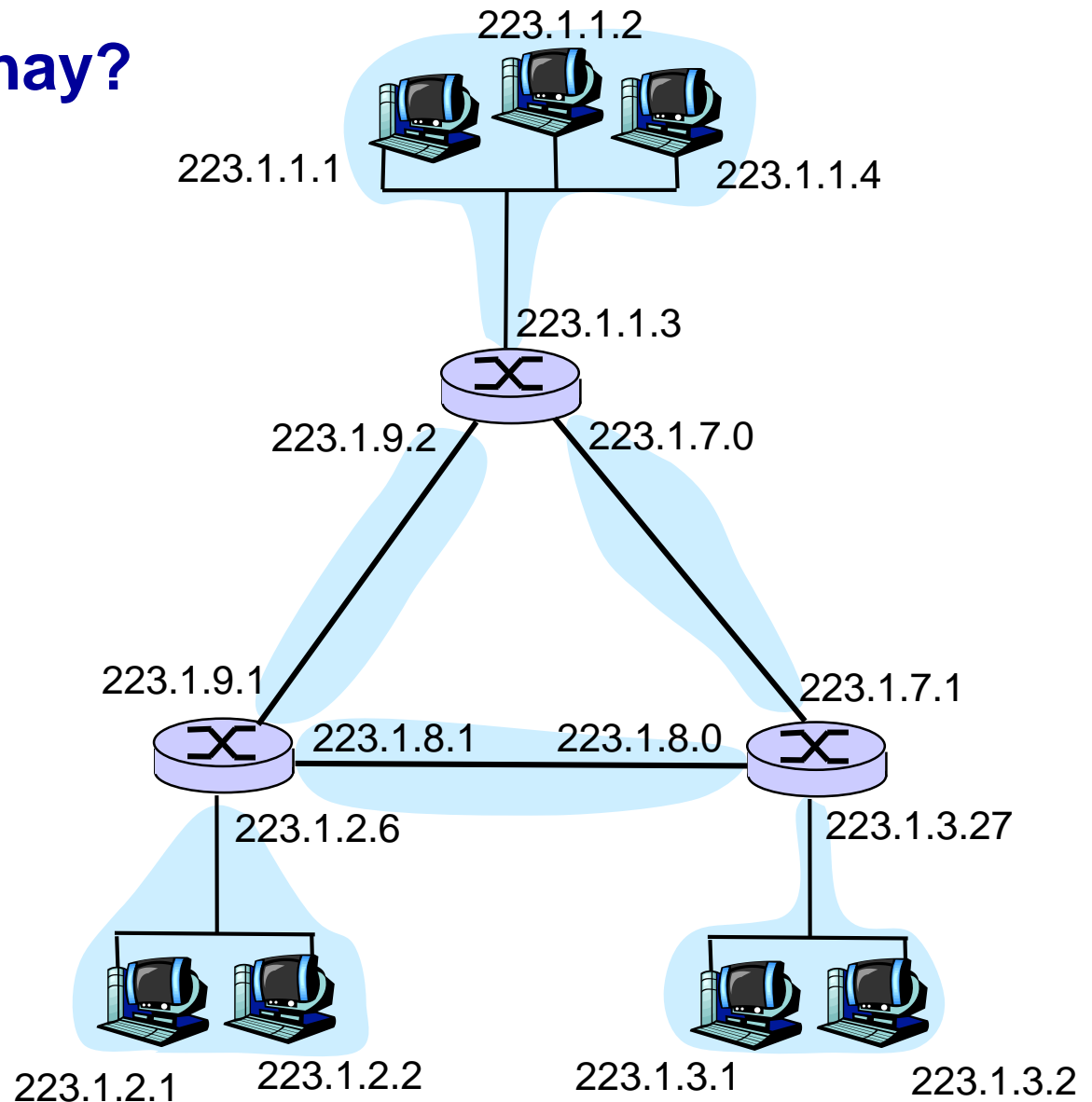
¿Cuántas redes IP hay?

!!!6!!!



Direcciones IP y redes IP

¿Cuántas redes IP hay?



Direcciones IP especiales

Id. red	Ceros
---------	-------

- **Dirección IP de una red**
 - **Campo de identificador de computador a 0's**
 - Se refiere a la red y no a sus computadores
 - No puede aparecer como dirección en un paquete

Direcciones IP especiales

127

Cualquier cosa

Todo ceros

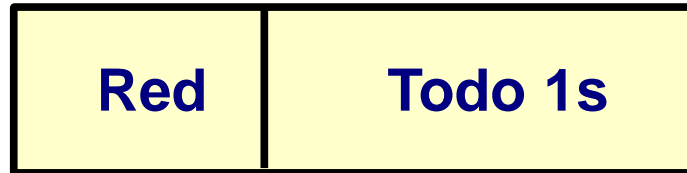
- **Dirección de loopback**

- Para depurar aplicaciones de red
 - No envía paquetes a través de la red
- Cuando en java empleamos la dirección IP “localhost” se traduce a la 127.0.0.1

- **La dirección de este host** (el que envía el paquete)

- Se utiliza como **dirección fuente** cuando el host obtiene su IP automáticamente a través de la red
- ¡Los protocolos utilizados se apoyan en IP!

Direcciones IP de difusión



- **Dirección de difusión dirigida**

- Para enviar una copia de un paquete a todos los hosts de una red IP



- **Dirección de difusión limitada**

- Difusión en la red IP donde está conectado el host
- Se utiliza como **dirección destino** durante el arranque del sistema si el host aún no conoce su dirección de red

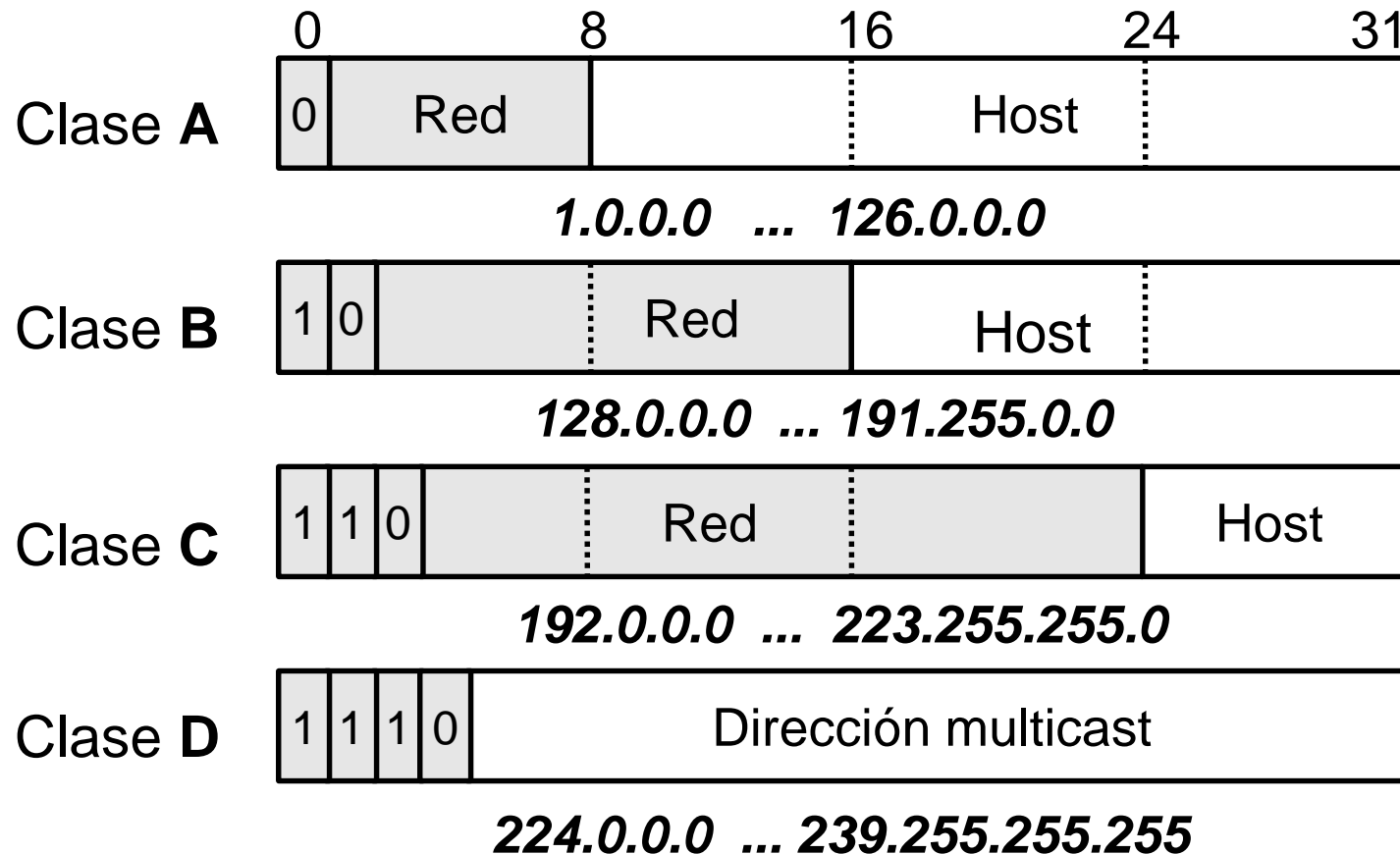
Direcciones privadas

- Direcciones privadas (RFC 1918)
- Estas direcciones no son encaminables por los routers de Internet
- 3 rangos de direcciones privadas
 - **ID Red de 8 bits: 10**
 - Rango: 10.0.0.0 – 10.255.255.255
 - **ID Red de 12 bits: 172.16**
 - Rango: 172.16.0.0 – 172.31.255.255
 - **ID Red de 16 bits: 192.168**
 - Rango: 192.168.0.0 – 192.168.255.255

Tipos de direccionamiento en Internet

- Dos tipos de direccionamiento en Internet, dependiendo de cómo se determine la longitud del identificador de red:
 - Direccionamiento con clases (obsoleto):
 - Los identificadores de red tienen longitudes fijas: 8, 16 y 24 bits
 - 3 clases de direcciones: A, B y C
 - Direccionamiento sin clases (***Classless InterDomain Routing***)
 - Se requiere una **máscara de red** para saber el número de bits que identifican a la red

Direccionamiento con clases (obsoleto)



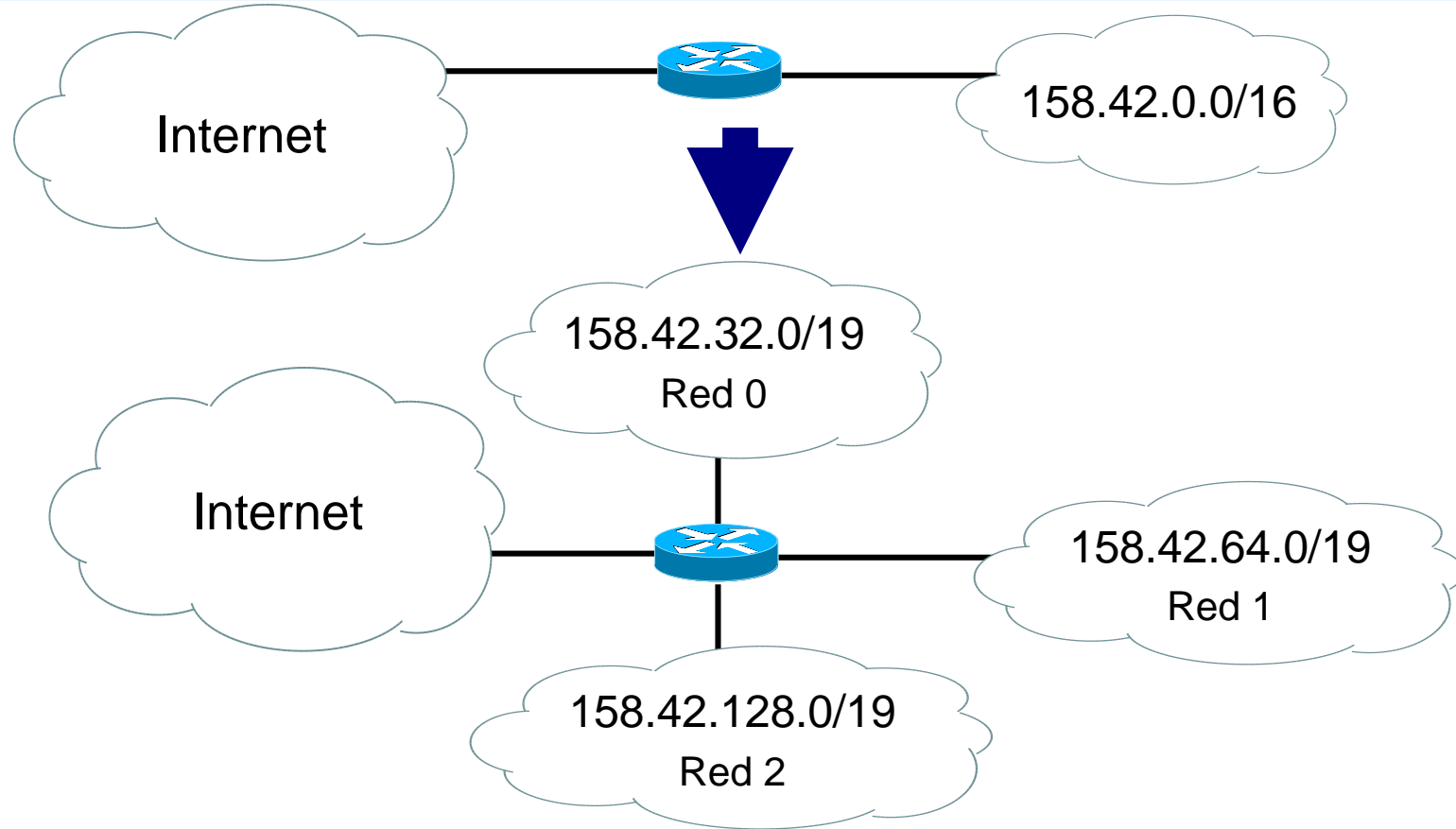
Direccionamiento sin clases (CIDR)

- Con la notación CIDR:
 - **Ya no hay clases**, las direcciones tienen:
 - **Algunos bits fijos (el prefijo de red)**
 - El tamaño de este campo no está limitado a unos pocos valores
 - **Algunos bits variables** (los bits de host)
 - Lo que era una dirección de red de clase A, B o C puede corresponder a varias **subredes**
 - Un bloque de direcciones de redes IP puede corresponder a una única red: **superred**
- Estas soluciones han ayudado a prolongar la vida de IP v4
- IP v6 utiliza esta visión sin clases

Direccionamiento sin clases (CIDR) (II)

- La dirección IP necesita delimitar cuántos bits pertenecen a la red
 - **Máscara de red**
 - **Prefijo de red**
- La máscara de red:
 - Permite saber qué bits corresponden al prefijo de red y cuáles al identificador del host
 - Tiene el mismo tamaño que la dirección IP (32 bits) y sus bits a 1 identifican el prefijo de red
 - Ejemplo para 20 bits a 1:
 - 255.255.240.0 ← Máscara
 - Dir IP/20 ← Prefijo
- **Cuestión:** ¿Ventajas de CIDR?

Ejemplo de CIDR

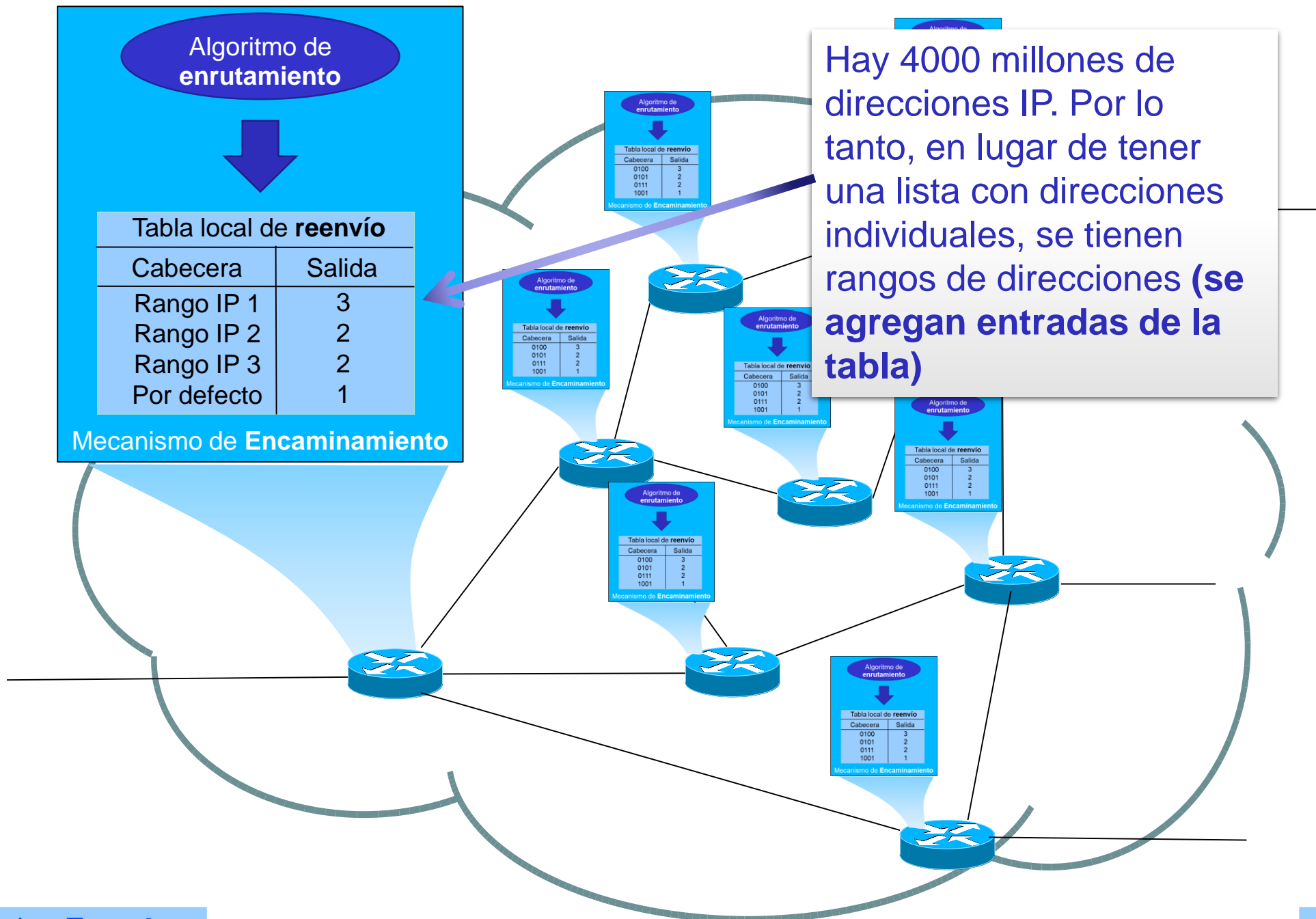


- Ejemplos para la red 2 (158.42.128.0):
 - Difusión:** 158.42.159.255
 - Algunos hosts :** 158.42.144.0, 158.42.128.255, 158.42.129.2
- Otra notación equivalente:
158.42.128.0/19

CIDR y reenvío

- Tablas de reenvío
 - Contienen información sobre los posibles destinos y cómo llegar a ellos
 - ¿Dónde están? En *routers* y *hosts*
 - ¿Cómo deben ser?
 - Compactas y pequeñas (en la medida de lo posible) para un funcionamiento más eficiente
 - Sólo con información sobre redes destino y los routers que nos conducen a ellas
 - Información en la tabla de reenvío:
 - Dirección IP red destino, máscara de red, dirección IP siguiente *router*, interfaz de salida

Reenvío de los datagramas



Reenvío de los datagramas

Rango de direcciones destino	Interfaz de salida
11001000 00010111 00010000 00000000 hasta 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 hasta 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 hasta 11001000 00010111 00011111 11111111	2
en otro caso	3

Reenvío de los datagramas

- Tener rangos en la tabla de encaminamiento no es la mejor opción
 - **Es más eficiente tener prefijos**

Prefijo de coincidencia	Interfaz de salida
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
en otro caso	3

- **Pero entonces ...**

- Dirección destino: 11001000 00010111 00010**110 10100001** ¿Qué interfaz?
- Dirección destino: 11001000 00010111 00011**000 10101010** ¿Qué interfaz?

Reenvío de los datagramas

Coincidencia con el prefijo más largo

Cuando se encamina un datagrama y existen varias coincidencias en la tabla de encaminamiento, se aplica la regla de coincidencia con el prefijo más largo

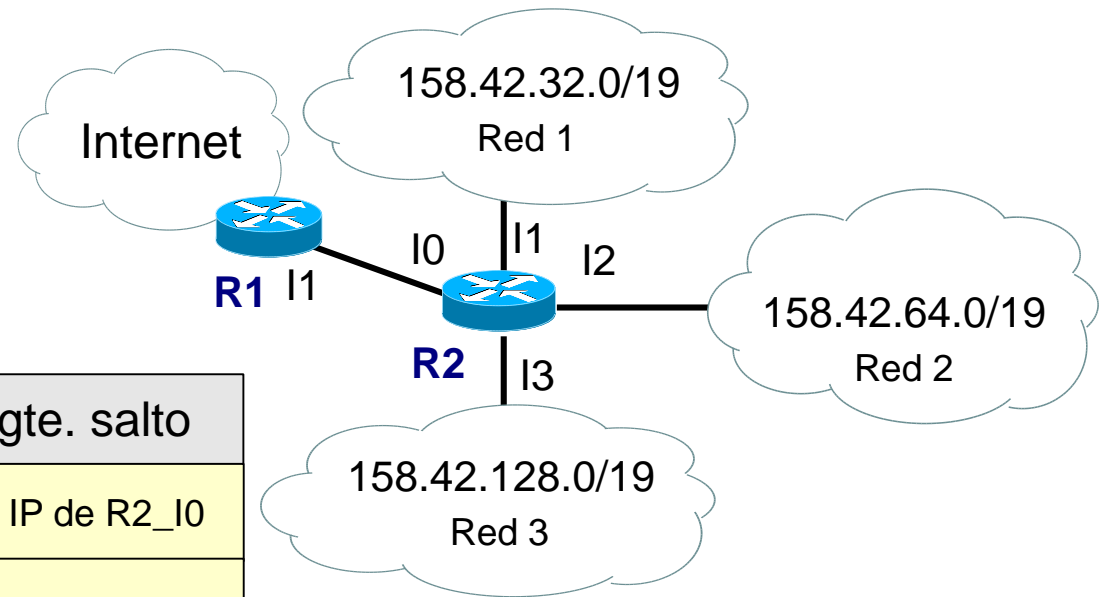
Prefijo de coincidencia	Interfaz de salida
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
en otro caso	3

- Dirección destino: 11001000 00010111 00010**110** 10100001 **Interfaz 0**
- Dirección destino: 11001000 00010111 00011000 **10101010** **Interfaz 1**

Ejemplo reenvío CIDR

- R1 verá una única red
 - Entrada en la tabla de encaminamiento:

Red destino	Máscara	Interfaz	Sgte. salto
158.42.0.0	/16	1	Dir IP de R2_I0
0.0.0.0	/0	0	Dir IP sgte router



- R2 verá las tres redes:

Red destino	Máscara	Interfaz	Sgte. salto
158.42.32.0	/19	1	0.0.0.0
158.42.64.0	/19	2	0.0.0.0
158.42.128.0	/19	3	0.0.0.0
0.0.0.0	/0	0	Dir. IP de R1_I1

1111 1111 1111 1111 1110 0000 0000 0000

Subredes: ejemplo (I)

- Tenemos una dirección de red 192.228.17.0/24
 - 192.228.17.0 = 11000000.11100100.00010001.00000000
 - Máscara inicial:
 - 255.255.255.0 = 11111111.11111111.11111111.00000000
 - Una red, 254 direcciones
- La dividimos en subredes
 - Pasamos bits del campo de host al campo de red
 - Nueva máscara de red: /27
 - 255.255.255.224 = 11111111.11111111.11111111.11100000
 - 3 bits de subred \rightarrow 8 redes (2^3)
 - 5 bits de host \rightarrow 30 hosts cada una (2^5-2)

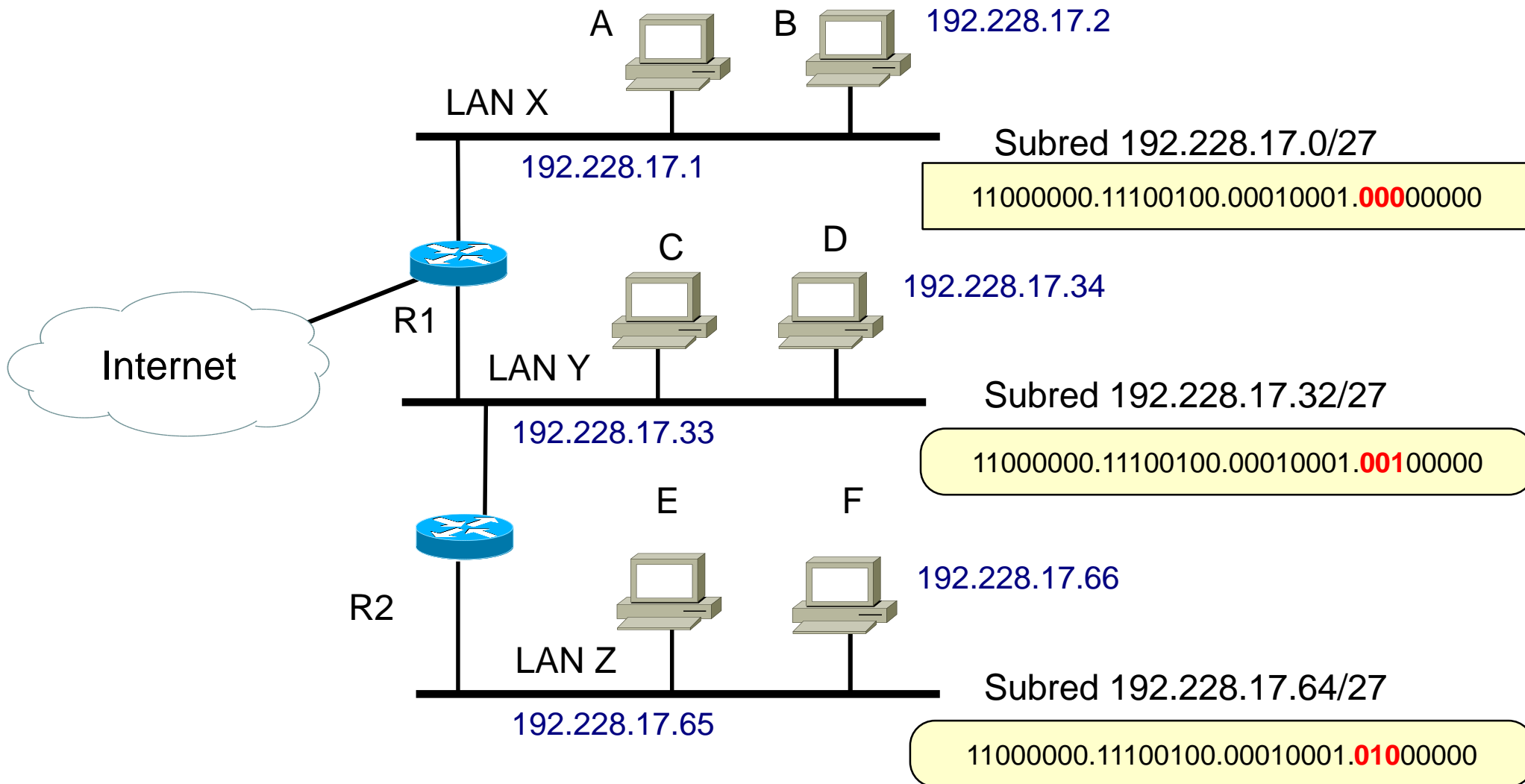
Subredes: ejemplo (II)

Red 192.228.17.0

11000000.11100100.00010001.00000000

Máscara de **subred**:
255.255.255.224

11111111.11111111.11111111.11100000



Superredes

- ¿Es posible juntar diferentes direcciones de red en una única dirección?
 - **Esto se conoce como agregación de redes**

- Ejemplo: Las 8 redes desde 194.32.136.0 a 194.32.143.0 tienen un prefijo común de 21 bits:

194.32.136.0/24 = **11000010 00100000 10001000** 00000000

194.32.137.0/24 = **11000010 00100000 10001001** 00000000

. . .

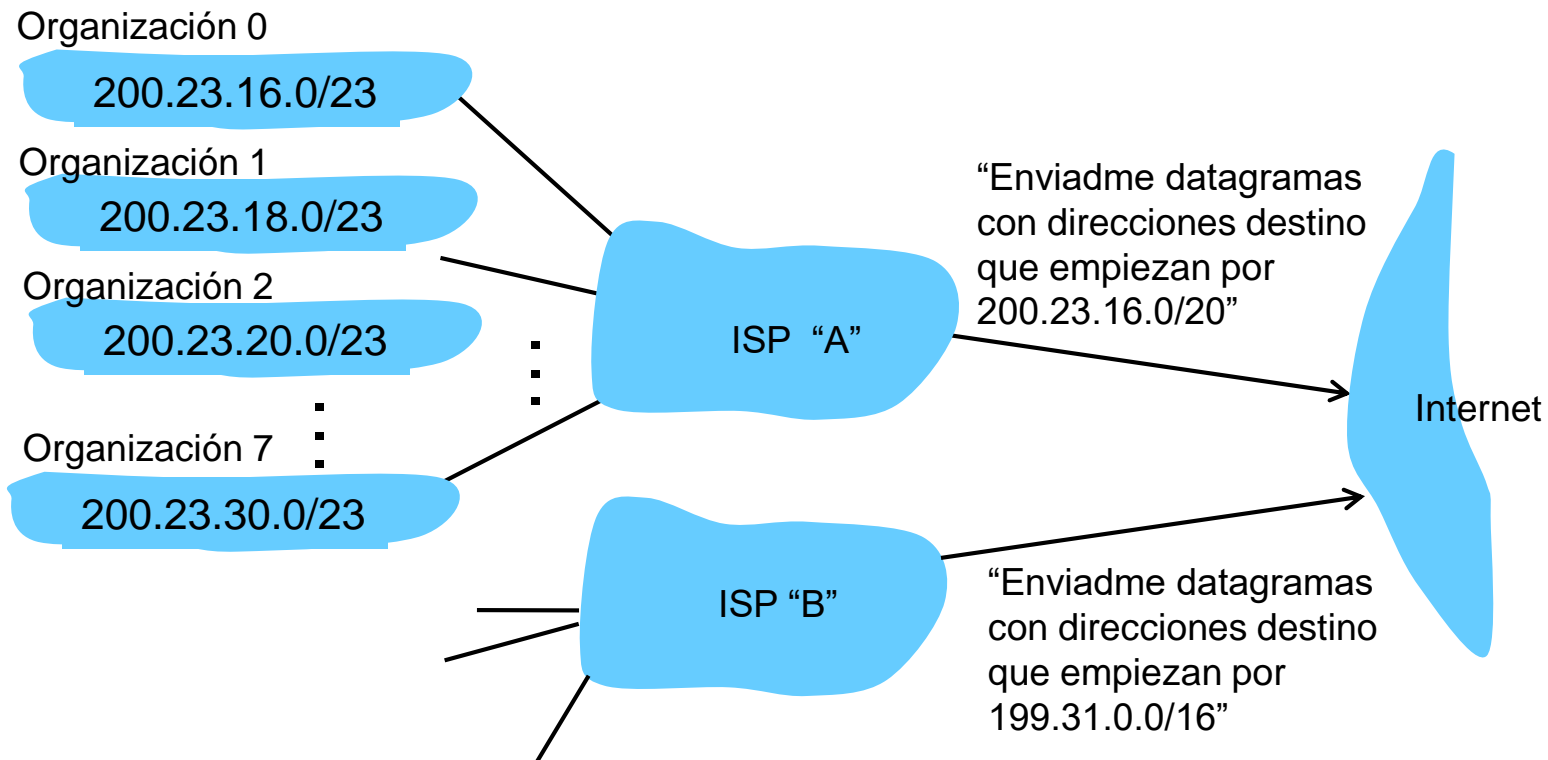
194.32.143.0/24 = **11000010 00100000 10001111** 00000000

- Este bloque de direcciones contiguas se puede expresar como una única (super)red:

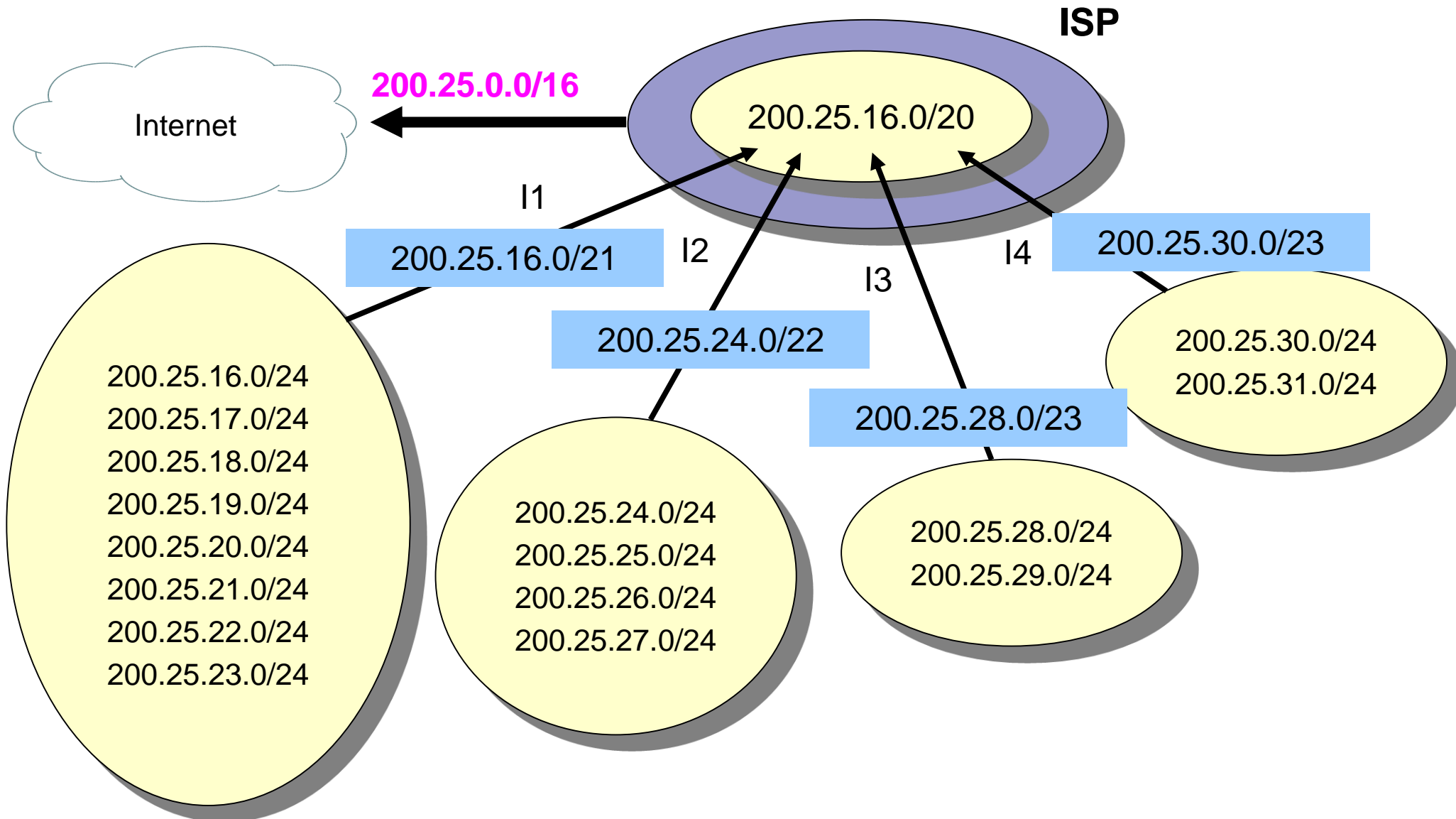
194.32.136.0/21

Superredes

- ¿Para qué sirve la agregación de redes?

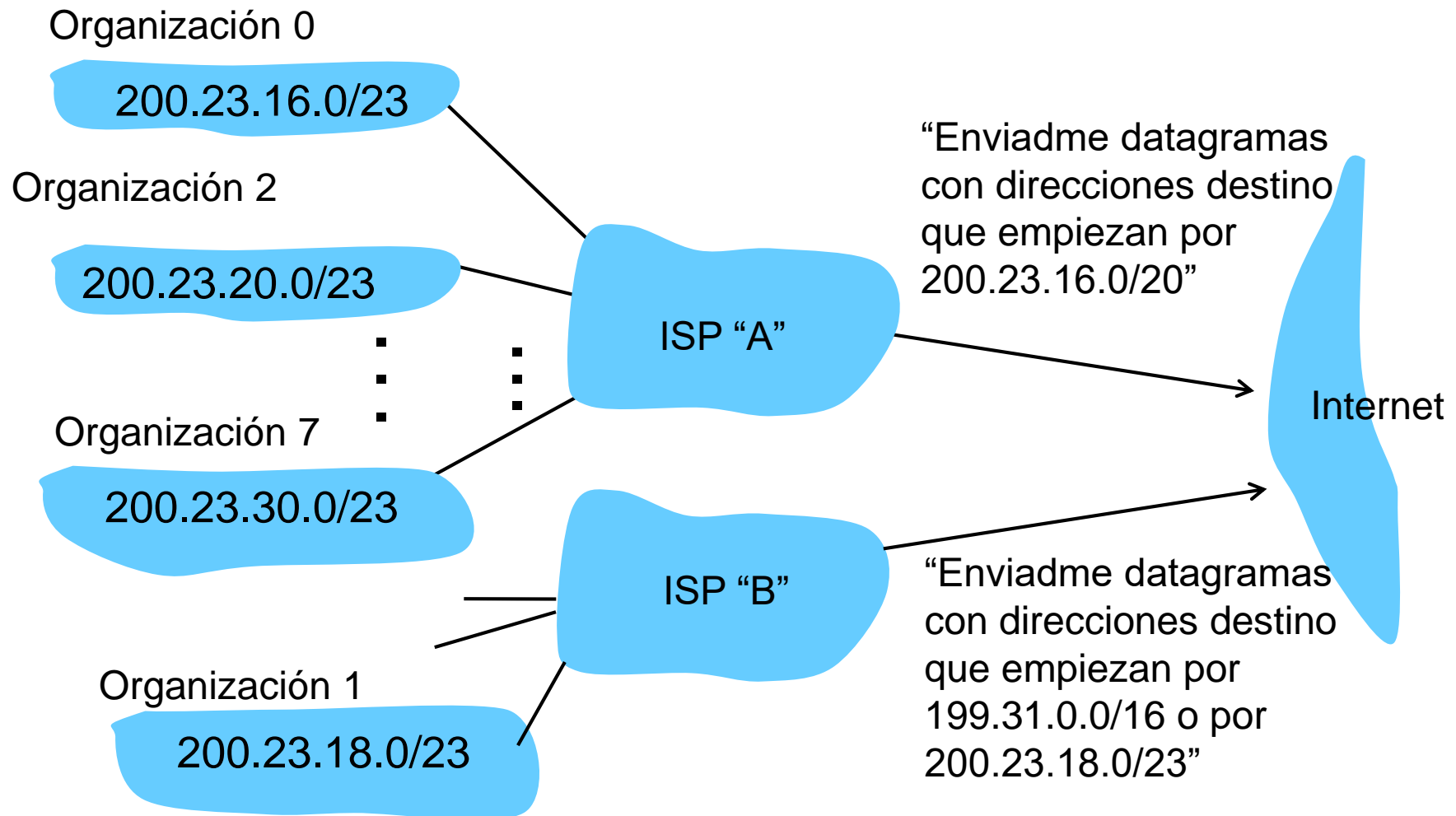


Superredes: varios niveles de jerarquía



Superredes: traslado de IPs

- El ISP "B" tiene una nueva ruta hacia la organización 1



Obtención de una dirección de red (I)

- ¿Cómo consigue un **host** una dirección IP?
 - Configurada **manualmente** por el administrador del sistema
 - Mediante el protocolo de configuración dinámica de host (**DHCP**, *Dynamic Host Configuration Protocol*)*

* RFC 2131

Obtención de una dirección de red (II)

- ¿Cómo se consigue un bloque de direcciones IP para la **red de una organización**?
 - A través de un ISP

Bloque del ISP	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/20
Organización 0	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/23
Organización 1	<u>11001000</u>	<u>00010111</u>	<u>00010010</u>	00000000	200.23.18.0/23
Organización 2	<u>11001000</u>	<u>00010111</u>	<u>00010100</u>	00000000	200.23.20.0/23
...	
Organización 7	<u>11001000</u>	<u>00010111</u>	<u>00011110</u>	00000000	200.23.30.0/23

Obtención de una dirección de red (III)

- ¿Cómo consigue un **ISP** un bloque de direcciones?
 - A través del ICANN (*Internet **C**orporation for **A**ssigned **N**ames and **N**umbers*)
 - En Europa delega en RIPE
 - Asigna direcciones IP
 - Gestiona los servidores raíz DNS
 - Asigna y resuelve disputas sobre nombres de dominio
- Pero ¡el ICANN ya no dispone de bloques libres para repartir!

3. IPv6: ¿Por qué una nueva versión?

- **Agotamiento del espacio de direcciones IP** en un futuro próximo
 - Internet está creciendo exponencialmente (su tamaño se dobla en menos de un año)
 - Se necesitan direcciones más largas
- **Nuevas aplicaciones** en Internet han provocado necesidades adicionales:
 - Audio y vídeo: retardos acotados, sincronización, calidad de servicio, etc.
- Tecnologías de colaboración: **comunicaciones dentro de un grupo**
- Soporte para **seguridad**

Características principales de IP v6

- Soporte de audio y vídeo: permite establecer rutas de alta calidad
 - Definición de flujos (un tanto ambigua)
- Formato de cabecera más flexible que en IP v4 para agilizar el encaminamiento:
 - Menos campos fijos que en IP v4
 - Cabeceras de extensión opcionales con un orden específico

Notación direcciones IPv6

- Direcciones de **128** bits
 - Teóricamente 6×10^{23} direcciones por m^2 de la Tierra
 - En la práctica unas 5000/ m^2
- **Tipos de direcciones**
 - **Unicast**: dirección de un computador
 - **Multicast**: dirección de un grupo de computadores (todos)
 - **Anycast**: dirección de un grupo de computadores (uno cualquiera del grupo, generalmente el mas cercano)
- Notación Hexadecimal
 - La dirección se divide en 8 bloques de 16 bits, y se codifican en hexadecimal (4 dígitos) y se separan mediante ":"
- Notación abreviada
 - Los ceros a la izquierda se pueden omitir
 - Las cadenas de ceros seguidas se pueden comprimir con los símbolos "::"
 - Solo una vez en una dirección.... ¿Por qué?

Notación direcciones IPv6 (II)

- Notación CIDR

- Prefijo y sufijo

- Ejemplos:

- ❖ FEDC:BA98:7654:3210:0123:4567:89AB:CDEF /64
- ❖ 1080:0000:0000:0000:0008:0800:200C:741A /64 =
1080::8:0800:200C:741A /64
- ❖ 21AB:0000:0000:000A:0000:0000:453E:BCDA /96 =
21AB::000A:0000:0000:453E:BCDA /96 ó bien
21AB:0000:0000:000A::453E:BCDA /96 pero nunca
21AB::000A::453E:BCDA /96

Direcciones IPv6: Ámbitos

- Zona de validez de la dirección (*scope*)
 - Link-local: dentro de una LAN
 - No atraviesan routers
 - Site-local: dentro de un sitio
 - Direcciones privadas → Intranets
 - Global: en todo Internet
 - Direcciones IP públicas

Tipo Dirección	Prefijo (bin)	Prefijo (Hex)
Reservada	0000 0000	0000:: /8
Unicast global	001	2000:: /3
Unicast enlace local	1111 1110 10	FE80:: /10
Unicast sitio local (en desuso)	1111 1110 11	FEC0:: /10
Unicast local (ULA)	1111 110	FC00:: /7
Multicast	1111 1111	FF00:: /10

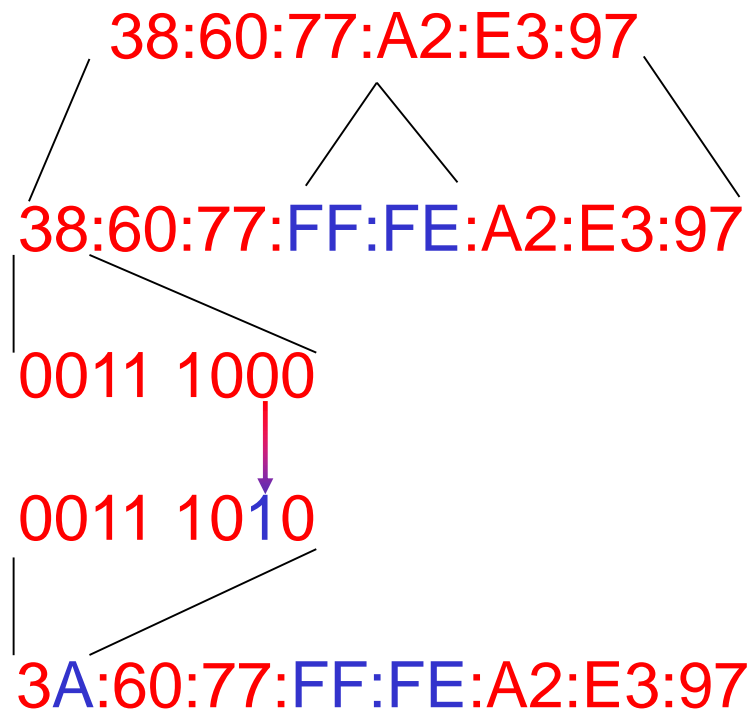
Direcciones *unicast* globales

- RFC 3587
- Equivalentes a direcciones públicas IPv4
- Jerárquica a tres niveles
 - Prefijo global → Sistema autónomo
 - Subred
 - Interfaz → Dirección MAC

001	Prefijo Global	Subred	Interfaz
3 bits	45 bits	16 bits	64 bits
2004:A128::		0032:	3A60:77FF:FEA2:E397

Direcciones *unicast* globales (II)

- Interfaz: La dirección MAC (48 bits) se convierte en un identificador EUI-64 ID (64 bits):
 - Insertar los bytes **FF FE** entre el tercero y cuarto de la dirección
 - Invertir el bit 7 del byte más significativo (Universal/local en IEEE)
- Ejemplo:



Direcciones *unicast* enlace local

- Definen direcciones privadas *unicast* que se usan en intranets, generalmente autoconfiguradas
 - Espacio plano de direcciones
 - Nunca se reenvían fuera del enlace
 - Permite la autoconfiguración y el descubrimiento de vecinos
- Formato:
 - El prefijo de formato ocupa 10 bits: 1111 1110 10 (FE80::/10).
 - El resto de los primeros 64 bits son 0
 - Identificador de host (64 bits)

1111 1110 1000 0000	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	Interfaz
16 bits	48 bits	64 bits
FE80	::	3A60:77FF:FEA2:E397

Direcciones *Multicast*

- Múltiples hosts en un ámbito determinado
- Solo tienen sentido como destino, nunca como origen
- Formato:
 - Prefijo: **FF::** /8
 - *Flags*: dirección permanente (0000) o temporal (0001)
 - Ámbito
 - Identificador del grupo *multicast*

1111 1111	Flags	Ámbito	Identificador de grupo
8 bits	4 bits	4 bits	112 bits

Permanente	0000
Temporal	0001

0000	Reservado
0001	Nodo local
0010	Enlace local
0101	Sitio local
1000	Organización local
1110	Global
1111	Reservado

Otras direcciones

- Dirección sin especificar: Indica que el host no tiene ninguna dirección asignada

$0:0:0:0:0:0:0:0 /128 = ::/128$

- Dirección de *loopback*: Similar a 127.0.0.1

$0:0:0:0:0:0:0:1 /128 = ::1/128$

- Direcciones IPv4 codificadas: En arquitecturas que emplean ambas versiones

- Formato: $::FFFF:<IPv4>$

- Ejemplo: $192.168.1.1 = ::FFFF:C0A8:0101$

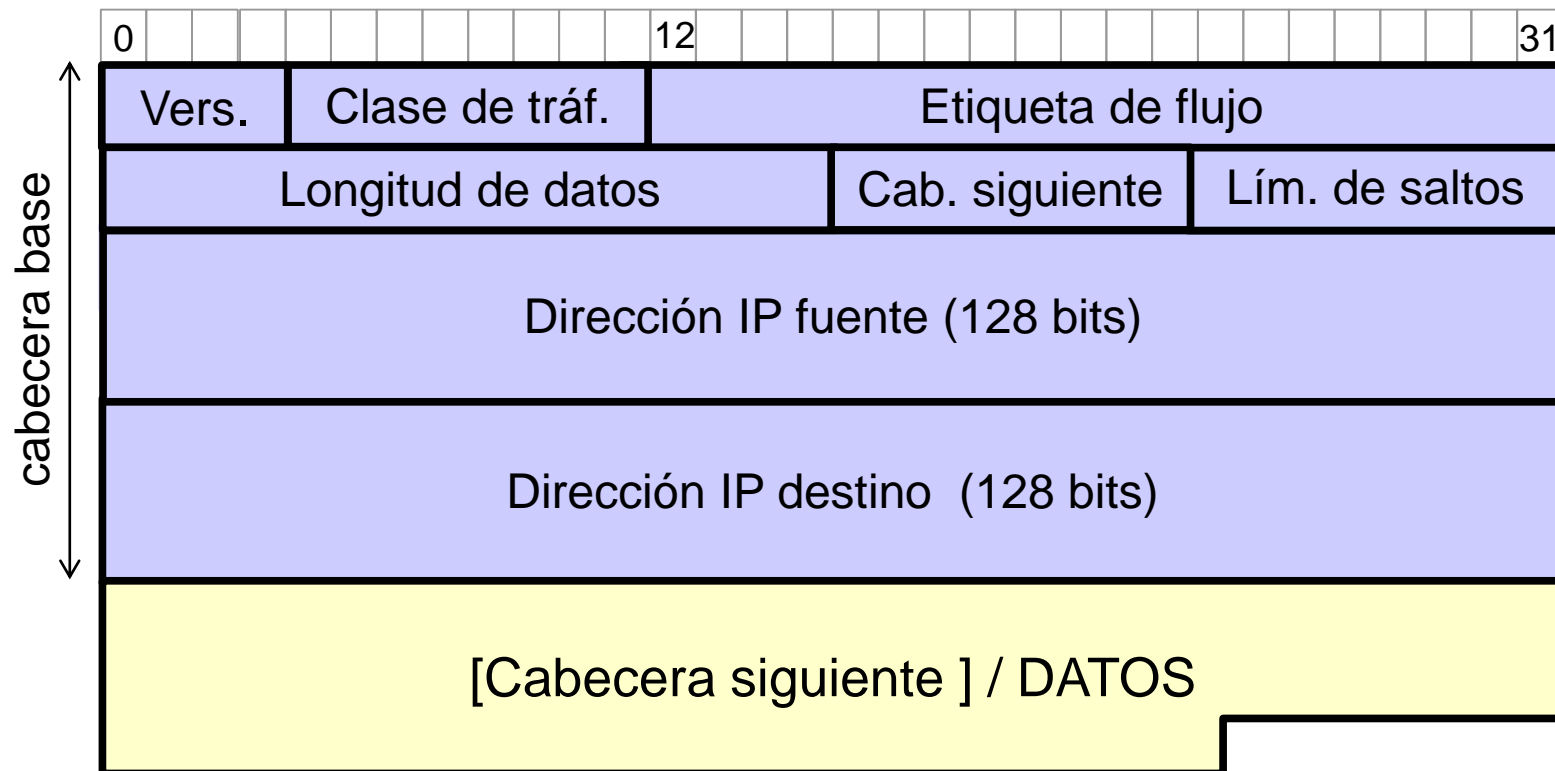
- Direcciones de documentación: Solo para ejemplos en docs

- Formato: $2001:DB8:: /32$

- Ejemplo: $2001:DB8:8:2A:: /32$

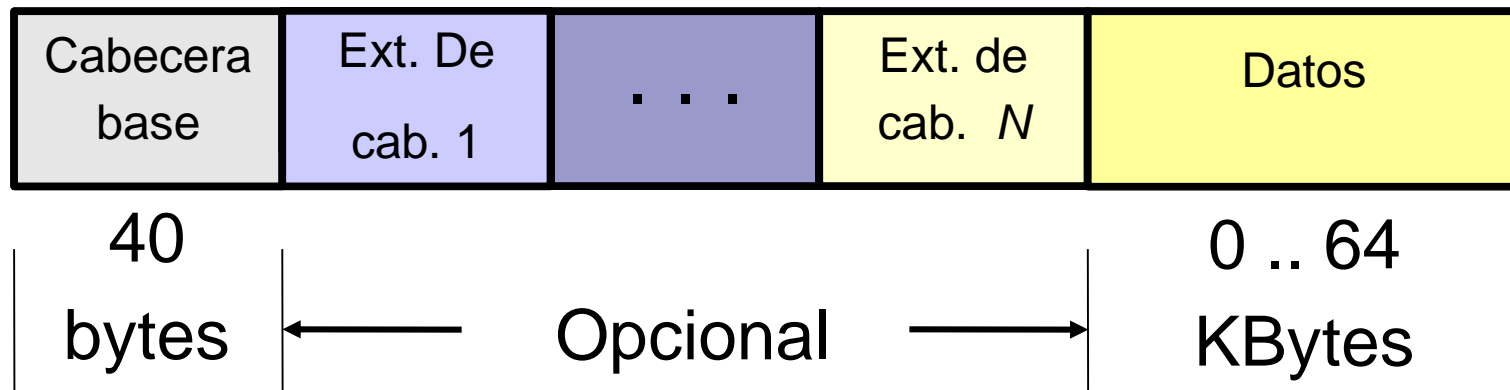
Formato de un paquete IP v6 (I)

- Etiqueta de flujo: Identifica paquetes del mismo flujo
- Clase de tráfico: para soportar calidad de servicio (QoS, *Quality of Service*)



Formato de un paquete IP v6 (II)

Sólo la cabecera base es imprescindible



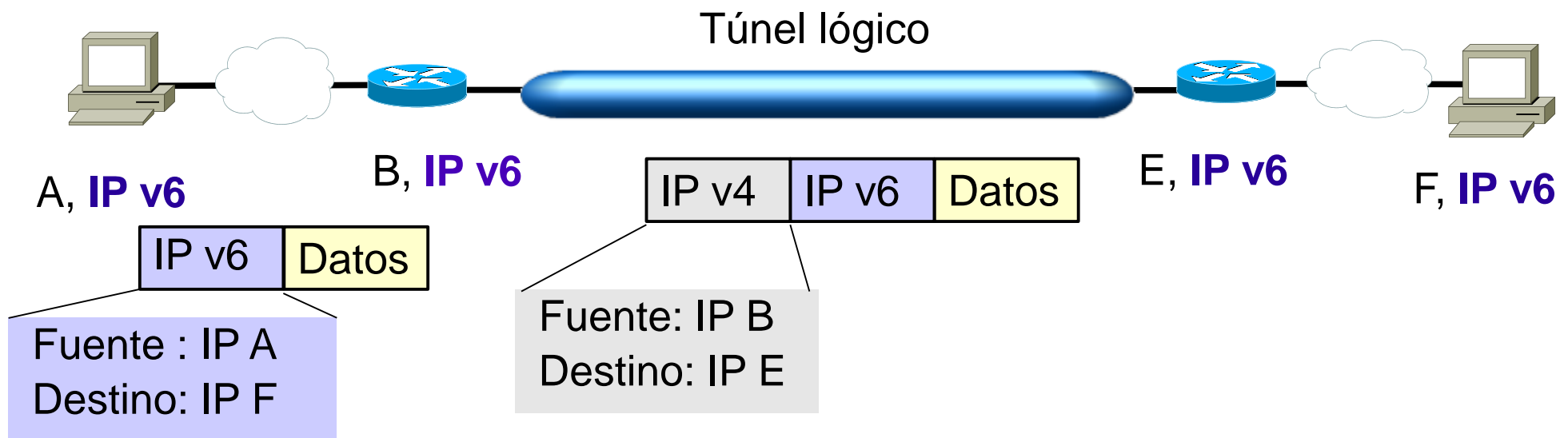
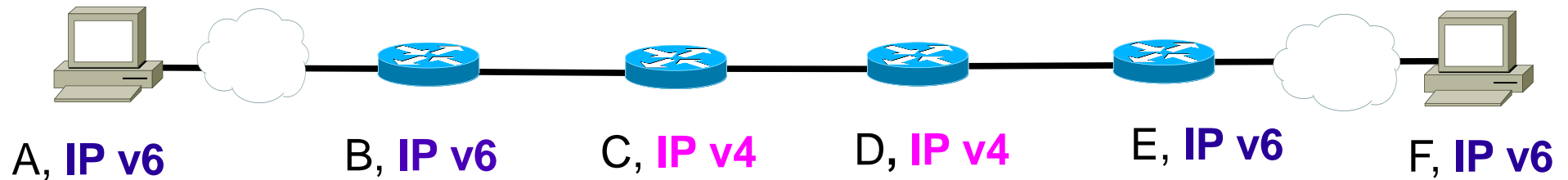
Otras diferencias con IP v4

- No se utiliza *checksum*
- Los routers no pueden fragmentar los paquetes
 - Si un paquete es demasiado grande el router envía un mensaje de error (ICMPv6)
- Nuevas características de seguridad
- ICMP v6: nueva versión de ICMP
 - Tipos adicionales de mensajes, como "Paquete demasiado grande"
 - Funciones de gestión de grupos (*multicast*)

Transición de la v4 a la v6

- Se ha previsto una fase de transición, durante la que puedan convivir ambos protocolos
- Durante el transitorio:
 - Las direcciones antiguas pueden codificarse fácilmente como direcciones nuevas
 - Se permitirán dispositivos capaces de encaminar paquetes IP v6 e IP v4
 - Las estaciones podrán utilizar los dos protocolos
 - El empleo de túneles permitirá encapsular tráfico IP v6 que tiene que atravesar zonas IP v4

Encapsulado de tráfico IP v6

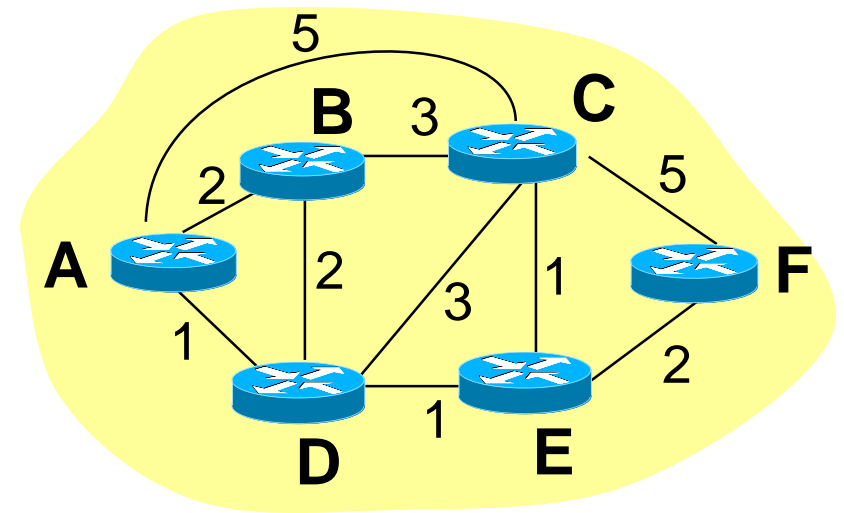


4.1 Algoritmos de enrutamiento: introducción

- Recordatorio:
 - Reenvío (*forwarding*): mover paquetes de la entrada del router a la salida apropiada
 - **Enrutamiento** (*routing*): determinar la ruta a tomar por los paquetes desde el origen al destino
 - Se calcula mediante **algoritmos de enrutamiento**
 - En Internet el intervalo de actualización típico está entre 1' y 5'
- **Objetivo del algoritmo de enrutamiento:**
 - **Determinar el 'mejor camino' (secuencia de routers) desde el origen al destino**

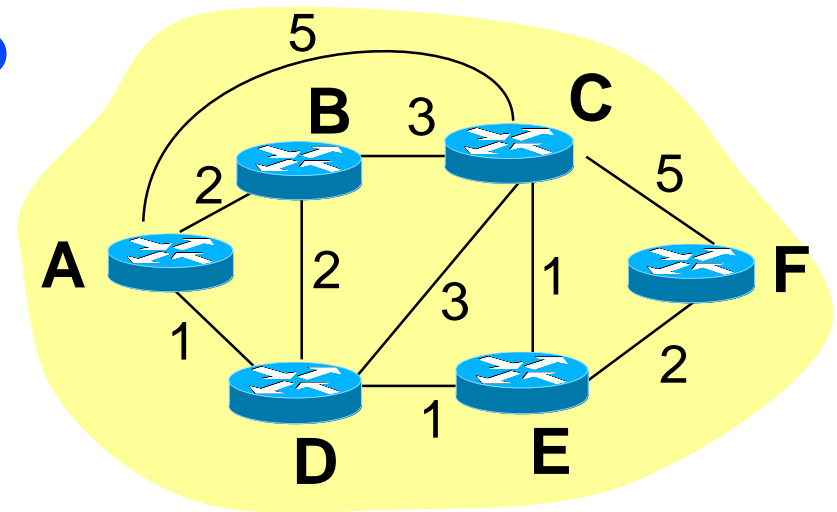
Representación de las rutas

- En los algoritmos de enrutamiento la red se representa mediante un grafo:
 - Vértices del grafo ➡ son los routers
 - Aristas del grafo ➡ son los enlaces entre routers



Coste de una ruta

- Simplificación:
 - Normalmente, consideraremos **grafos no dirigidos**:
 - $c(B,D)=c(D,B)=5$
- El coste puede calcularse atendiendo a diferentes criterios



Coste de una ruta $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

¿Cuál es la ruta de coste mínimo entre A y F?

Criteria for the classification of algorithms

- Two fundamental questions about the routing algorithm:
 - ¿Es **escalable**?
 - Algorithms by *vector de distancias y estado del enlace* as maximum for 100 routers
 - ¿Es **centralizado** or **distribuido**?
 - Depends on where the tables are calculated
 - Centralized algorithms are difficult to scale, all our proposals will be **distributed algorithms**

Otros criterios para la clasificación

- Dependiendo de la frecuencia con que se actualizan las rutas:
 - **Estáticos**: suelen actualizarse manualmente
 - **Dinámicos**: son sensibles a cambios en el tráfico o en la topología de la red
- Los algoritmos de enrutamiento empleados en **Internet** suelen ser **dinámicos**

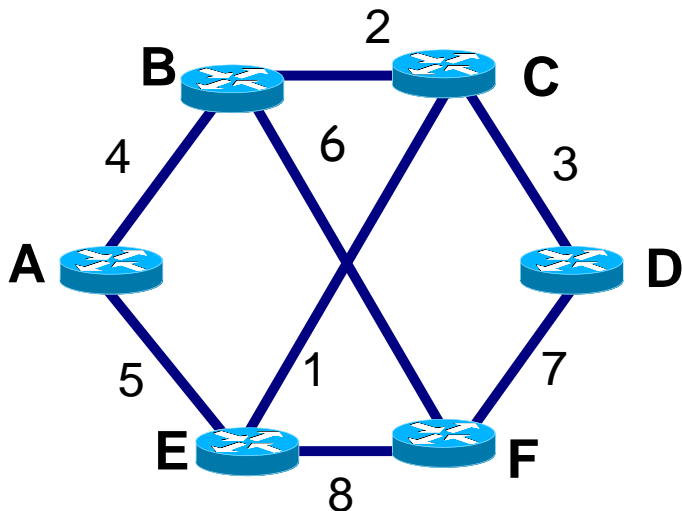
4.2 Enrutamiento por estado del enlace (I)

- Cada router debe:
 - Descubrir a sus vecinos y conocer sus direcciones de red
 - Medir el coste a cada vecino
 - Crear un paquete con la información
 - Enviar el paquete a **TODOS** los routers de la red
- Con los mensajes recibidos, **cada nodo**:
 - Construye el grafo de la red
 - Calcula el camino más corto (**Dijkstra**) a cada destino
 - Construye la nueva tabla de reenvío

Enrutamiento por estado del enlace (II)

El paquete de estado del enlace contiene:

- Identidad del emisor
- Número de secuencia
- Lista de vecinos y su distancia a ellos
- Tiempo de vida (TTL)



Paquetes de estado del enlace

A	
#Sec	
B	4
E	5

B	
#Sec	
A	4
C	2
F	6

C	
#Sec	
B	2
D	3
E	1

D	
#Sec	
C	3
F	7

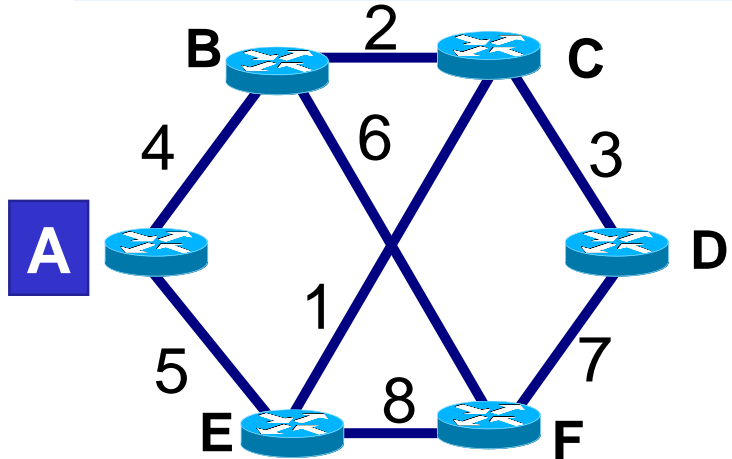
E	
#Sec	
A	5
C	1
F	8

F	
#Sec	
B	6
D	7
E	8

Notación para el alg. de Dijkstra

- $c(i,j)$: coste del enlace del nodo i al j
 - El coste es infinito si i y j no son vecinos
- $D(v)$: valor **actual** del coste mínimo del camino desde la fuente al destino v
- $p(v)$: nodo predecesor a lo largo del camino desde la fuente a v , o sea un nodo vecino de v
- N' : conjunto de nodos cuyo camino de coste mínimo se conoce definitivamente

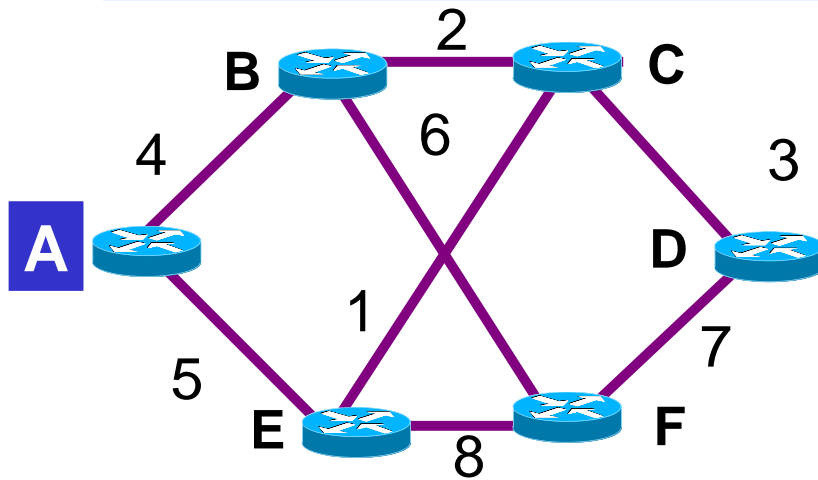
Ejecución del algoritmo de Dijkstra



- Para el nodo fuente A
- Al final, la lista de predecesores permite obtener los caminos más cortos a cada destino

Paso	N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	4, A	infinito	infinito	5, A	infinito
1	A, B		6, B	infinito	5, A	10, B
2	A, B, E		6, B	infinito		10, B
3	A, B, E, C			9, C		10, B
4	A, B, E, C, D					10, B
5	A, B, E, C, D, F					

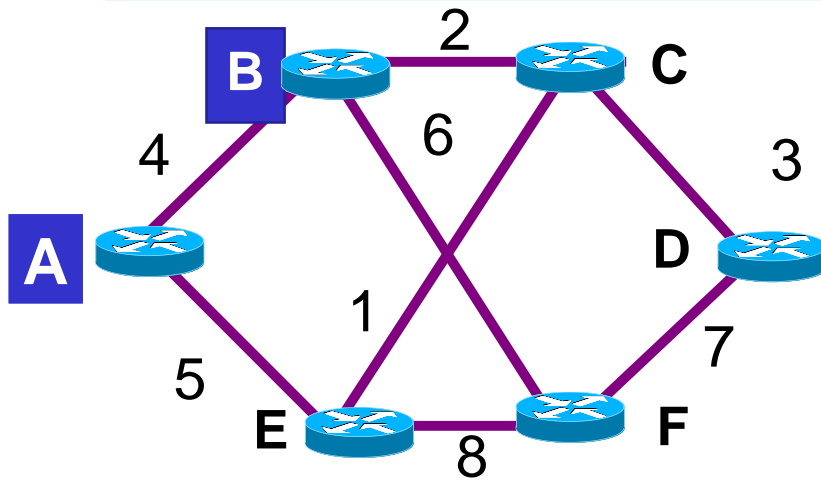
Ejecución del algoritmo de Dijkstra



• Para el nodo fuente A

Paso	N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	4, A	infinito	infinito	5, A	infinito

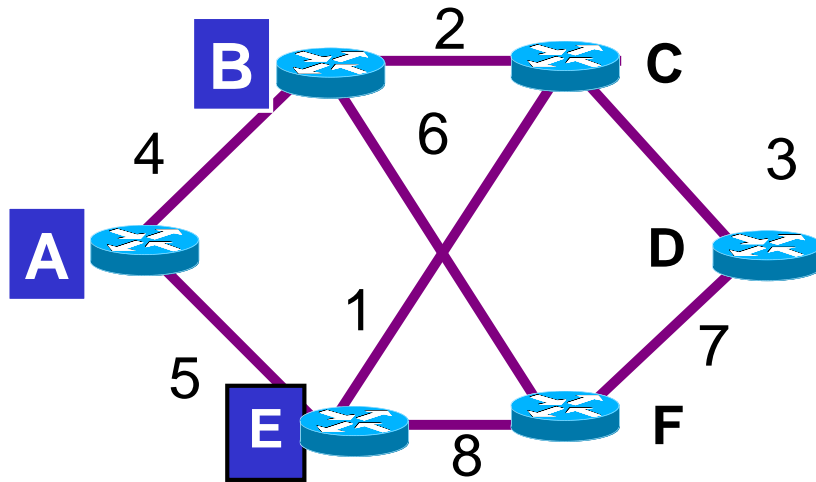
Ejecución del algoritmo de Dijkstra



• Para el nodo fuente A

Paso	N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	4, A	infinito	infinito	5, A	infinito
1	A, B		6, B	infinito	5, A	10, B

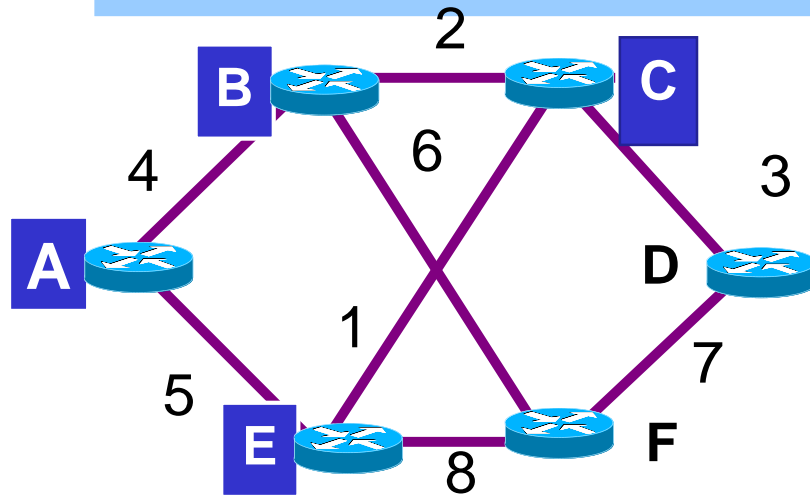
Ejecución del algoritmo de Dijkstra



• Para el nodo fuente A

Paso	N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	4, A	infinito	infinito	5, A	infinito
1	A, B		6, B	infinito	5, A	10, B
2	A, B, E		6, B	infinito		10, B

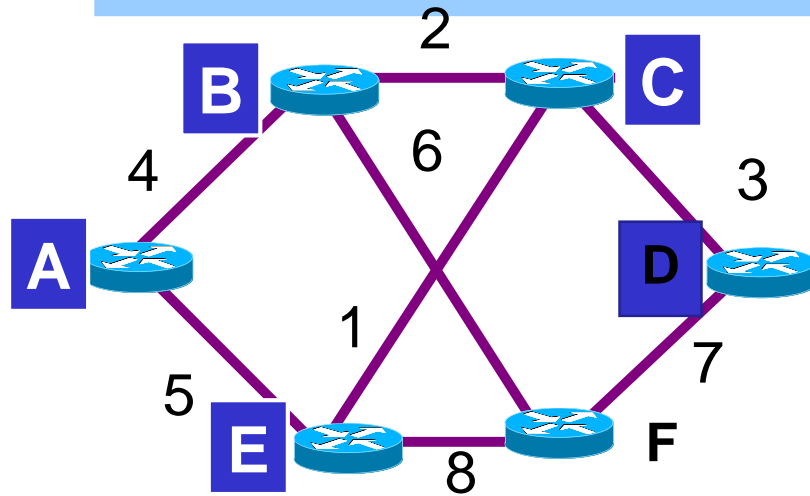
Ejecución del algoritmo de Dijkstra



• Para el nodo fuente A

Paso	N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	4, A	infinito	infinito	5, A	infinito
1	A, B		6, B	infinito	5, A	10, B
2	A, B, E		6, B	infinito		10, B
3	A, B, E, C			9, C		10, B

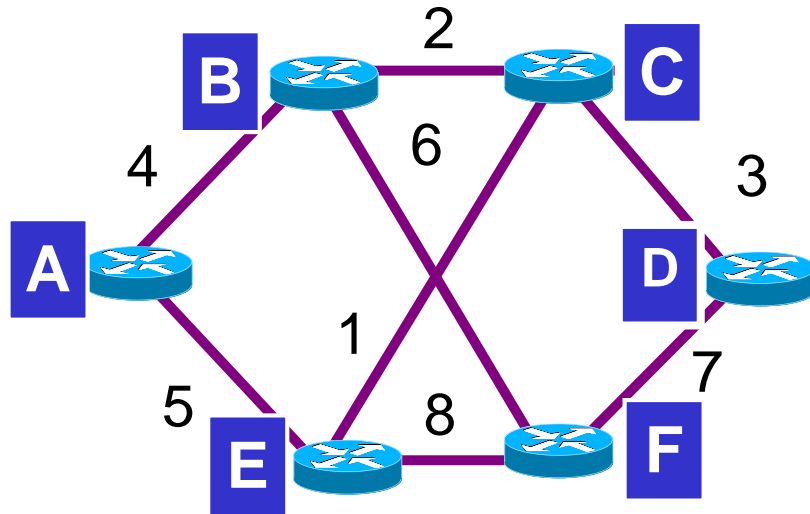
Ejecución del algoritmo de Dijkstra



- Para el nodo fuente A

Paso	N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	4, A	infinito	infinito	5, A	infinito
1	A, B		6, B	infinito	5, A	10, B
2	A, B, E		6, B	infinito		10, B
3	A, B, E, C			9, C		10, B
4	A, B, E, C, D					10, B

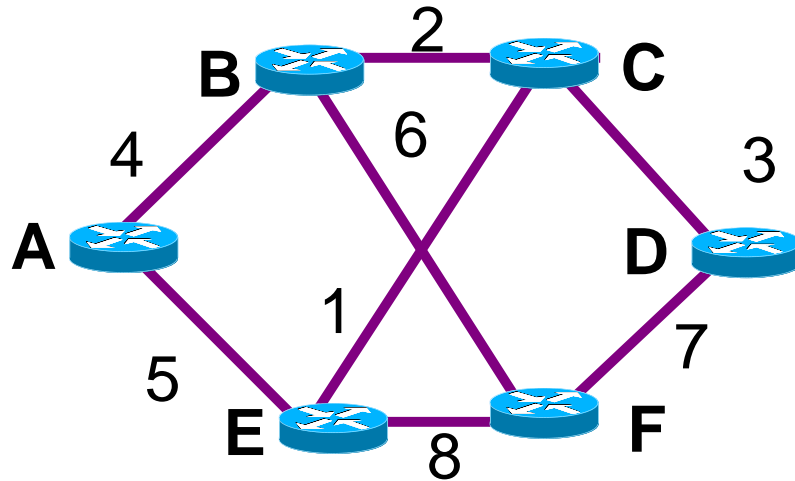
Ejecución del algoritmo de Dijkstra



- Para el nodo fuente A

Paso	N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	4, A	infinito	infinito	5, A	infinito
1	A, B		6, B	infinito	5, A	10, B
2	A, B, E		6, B	infinito		10, B
3	A, B, E, C			9, C		10, B
4	A, B, E, C, D					10, B
5	A, B, E, C, D, F					

Ejecución del algoritmo de Dijkstra



- Para el nodo fuente A
- Al final, la lista de predecesores permite obtener los caminos más cortos a cada destino

Paso	N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	4, A	infinito	infinito	5, A	infinito
1	A, B		6, B	infinito	5, A	10, B
2	A, B, E		6, B	infinito		10, B
3	A, B, E, C			9, C		10, B
4	A, B, E, C, D					10, B
5	A, B, E, C, D, F					

Dijkstra: ejemplo (II)

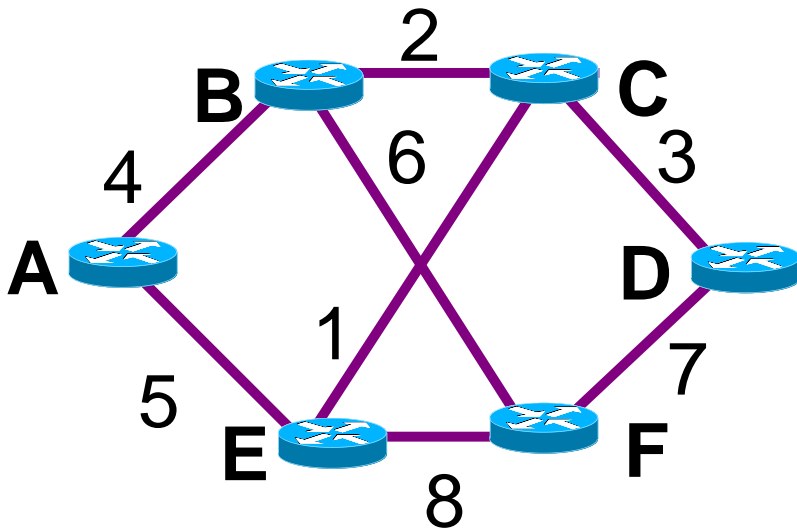


Tabla de reenvío de A

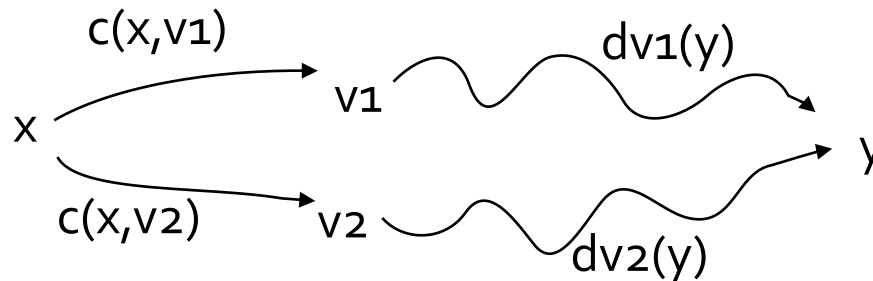
Destino	Enlace
B	(A,B)
C	(A,B) / (A,E)
D	(A,B)
E	(A,E)
F	(A,B)

4.3 Enrutamiento por vector de distancias

- Cálculo de la distancia a un destino
 - Coste de la ruta mínima de x a y (ecuación de Bellman-Ford):

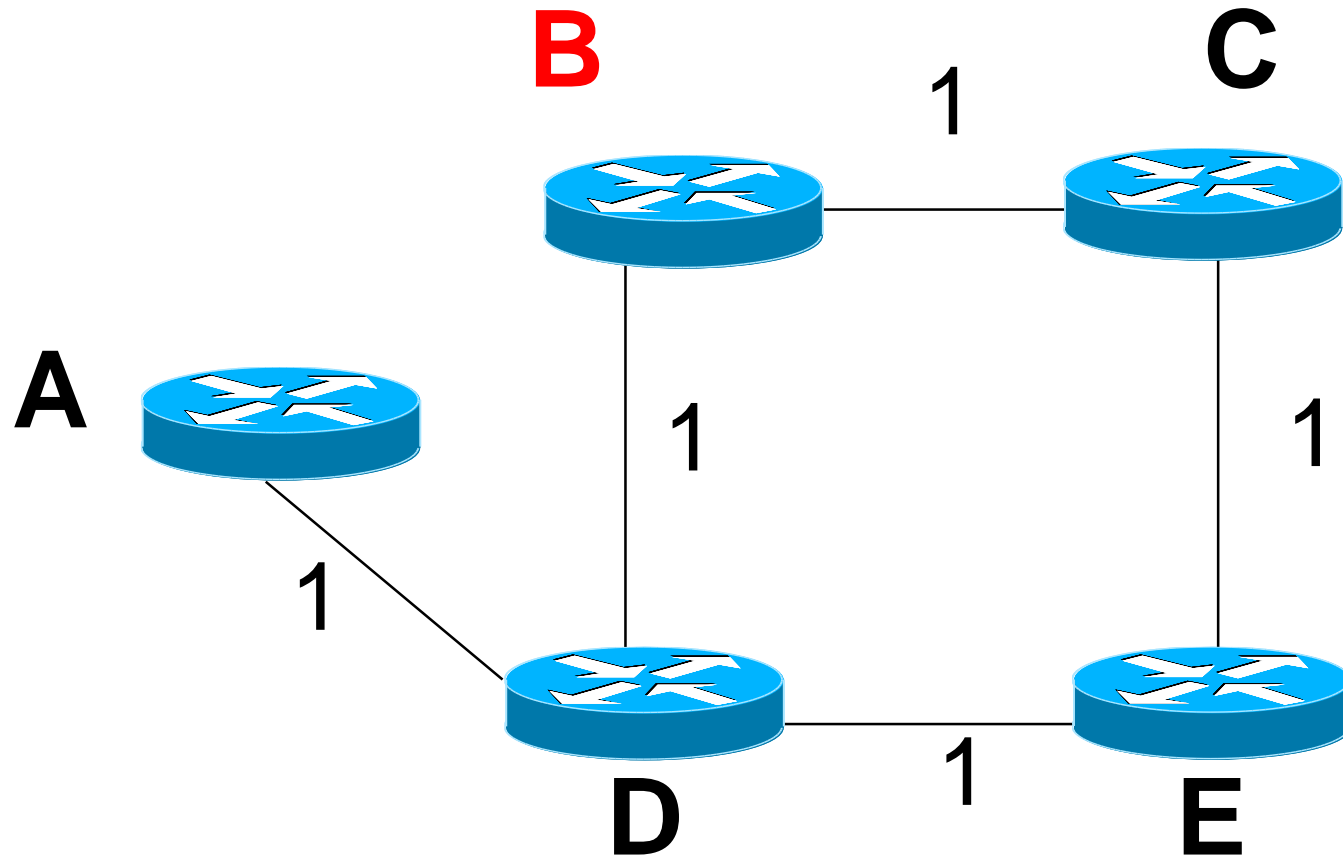
$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

min se calcula para todos los vecinos de x



- Con las distancias desde cada vecino se construye la tabla de distancias
- Y a partir de la tabla de distancias la de reenvío

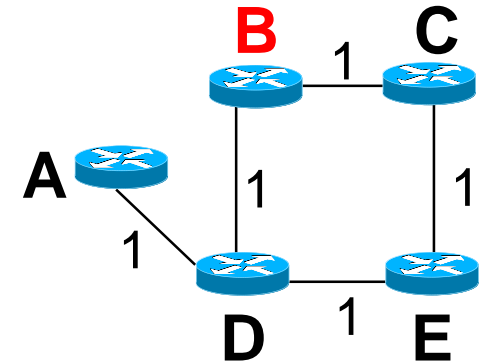
Ejemplo: tabla de distancias del nodo B



Ejemplo: tabla de distancias del nodo B

- **Información disponible en el nodo B:**

- **Vecinos:** C y D
- **Distancia a los vecinos:** $d_B(C) = 1$, $d_B(D) = 1$



- **Distancia del nodo B al nodo destino A**

$$d_B(A) = \min \{c(B,C) + d_C(A), c(B,D) + d_D(A)\} = \min \{1 + 3, \mathbf{1 + 1}\} = \mathbf{2}$$

Vectores de distancias de los nodos C y D

C	
A	3
B	1
D	2
E	1

$$d_B(C) = 1$$

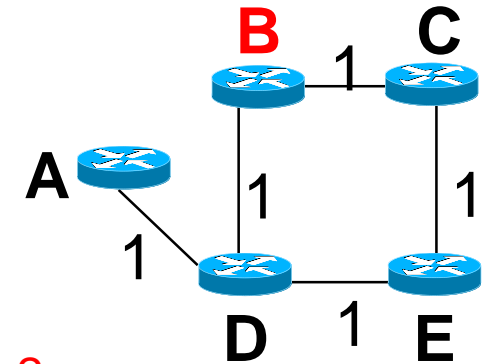
D	
A	1
B	1
C	2
E	1

$$d_B(D) = 1$$

Ejemplo: tabla de distancias del nodo B

- Información disponible en el nodo B:**

- Vecinos:** C y D
- Distancia a los vecinos:** $d_B(C) = 1$, $d_B(D) = 1$



- Distancia del nodo B al nodo destino A**

$$d_B(A) = \min \{c(B,C) + d_C(A), c(B,D) + d_D(A)\} = \min \{1 + 3, \mathbf{1 + 1}\} = \mathbf{2}$$

Vectores de distancias de los nodos C y D

C	
A	3
B	1
D	2
E	1

D	
A	1
B	1
C	2
E	1

$$d_B(C) = 1$$

$$d_B(D) = 1$$



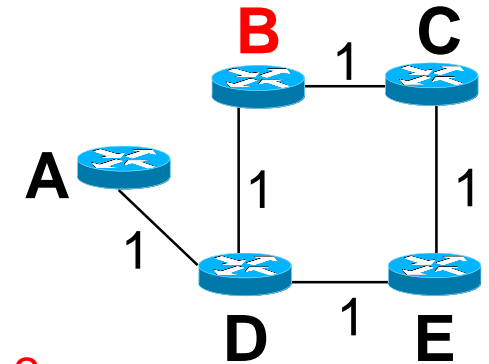
Tabla de distancias del nodo B

$D_B()$		Coste al destino vía	
		C	D
Destino	A	4	2
	C	1	3
	D	3	1
	E	2	2

Ejemplo: tabla de distancias del nodo B

- Información disponible en el nodo B:**

- Vecinos:** C y D
- Distancia a los vecinos:** $d_B(C) = 1$, $d_B(D) = 1$



- Distancia del nodo B al nodo destino A**

$$d_B(A) = \min \{c(B,C) + d_C(A), c(B,D) + d_D(A)\} = \min \{1 + 3, 1 + 1\} = 2$$

Vectores de distancias de los nodos C y D

C	
A	3
B	1
D	2
E	1

D	
A	1
B	1
C	2
E	1

$$d_B(C) = 1$$

$$d_B(D) = 1$$

Tabla de distancias del nodo B

		Coste al destino vía	
		C	D
Destino	$D_B()$		
	A	4	2
	C	1	3
	D	3	1
	E	2	2

Tabla de reenvío

Destino	Siguiente salto
A	D
C	C
D	D
E	C

Otro ejemplo de una tabla de reenvío

		Coste al destino vía		
Destino	$D_D()$	A	B	E
	A	1	3	3
	B	3	1	3
	C	4	2	2
	E	3	3	1

Destino	Siguiente
A	A
B	B
C	B
E	E

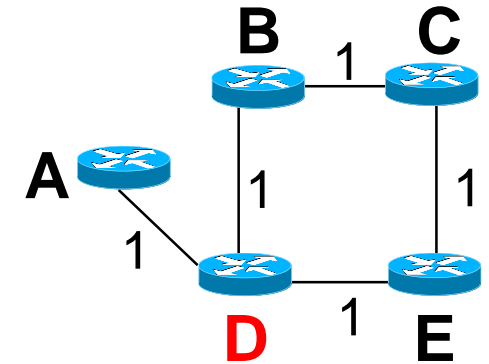


Tabla de distancias → **Tabla de reenvío**

El nodo vecino que proporciona el camino mínimo es el siguiente salto en la tabla de reenvío

Algoritmo de vector de distancias (I)

- $D_x(y)$ = coste mínimo **estimado** desde x hasta y
 - El nodo x mantiene el vector de distancias $D_x=[D_x(y); y \in N]$
- **Nodo x:**
 - Conoce el coste a cada vecino v: $c(x,v)$
 - Mantiene los vectores de distancia de sus vecinos. Para cada vecino v, x almacena $D_v=[D_v(y); y \in N]$
 - Si no conoce la distancia a un destino $D_x(y) = \infty$

Algoritmo de vector de distancias (II)

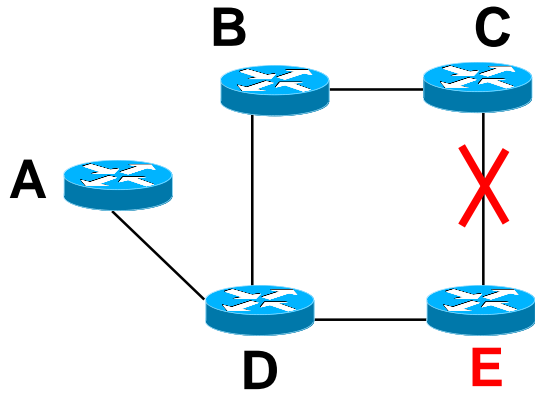
- Idea básica:

- Cada nodo envía periódicamente su vector de distancias (VD) a sus vecinos
- Cuando x recibe el VD de un vecino, actualiza, su propio VD (si procede) mediante la ecuación de B-F:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ para cada nodo } y \in N$$

- En condiciones normales, la estimación $D_x(y)$ converge al coste real mínimo $d_x(y)$
- Si el vector de distancias cambia x envía su nuevo vector a sus vecinos, y ellos a su vez pueden actualizar sus vectores de distancia

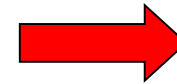
Vector de distancias: cambios en el coste de un enlace (I)



- Los cambios en los costes a veces se resuelven rápidamente

Coste al destino vía:

Destino	$D_E()$	
	C	D
A	4	2
B	2	2
C	1	3
D	3	1

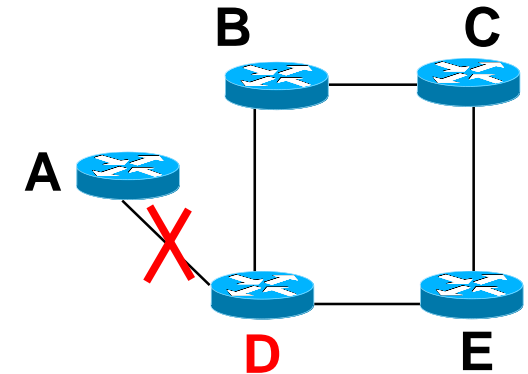


Coste al destino vía:

Destino	$D_E()$	
	C	D
A	∞	2
B	∞	2
C	∞	3
E	∞	1

Vector de distancias: cambios en el coste de un enlace (II)

En ocasiones la red puede tener problemas para estabilizarse



Destino	Coste al destino vía			
	$D_D()$	A	B	E
	A	∞	3	3
	B	∞	1	3
	C	∞	2	2
	E	∞	3	1

¡La ruta que tiene B utiliza el enlace DA!

- Aparecen bucles en el encaminamiento
- Problema de la **cuenta al infinito**
 - Soluciones:
 - Limitación del diámetro de la red
 - División horizontal con inversa envenenada

Vector de distancias: cambios en el coste de un enlace (III)

- Solución con inversa envenenada

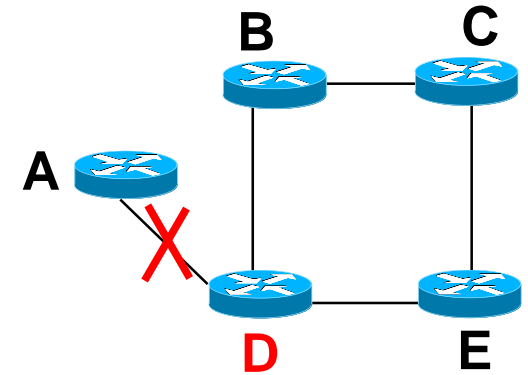
Coste al destino vía

$D_D()$	A	B	E
Destino			
A	∞	∞	∞
B	∞	1	3
C	∞	2	2
E	∞	3	1

Pero C ofrecerá a B una ruta a A a través de E



Más iteraciones hasta que C detecte que ni B ni E tienen ruta a A



5. Enrutamiento en Internet

- Nuestro estudio del enrutamiento ha sido idealizado. Hemos supuesto que:
 - Todos los routers son idénticos
 - La red es “plana”
- Escalabilidad:
 - Con millones de destinos posibles, los algoritmos vistos:
 - Tendrían problemas de convergencia
 - Generarían gran cantidad de tráfico, que inundaría los enlaces de Internet
 - Necesitarían tablas de encaminamiento inmensas
- Autonomía administrativa
 - Internet = red de redes
 - Cada administrador puede querer elegir el protocolo de encaminamiento
 - Tener un criterio distinto sobre la métrica más idónea

5.1 Solución: enrutamiento jerárquico

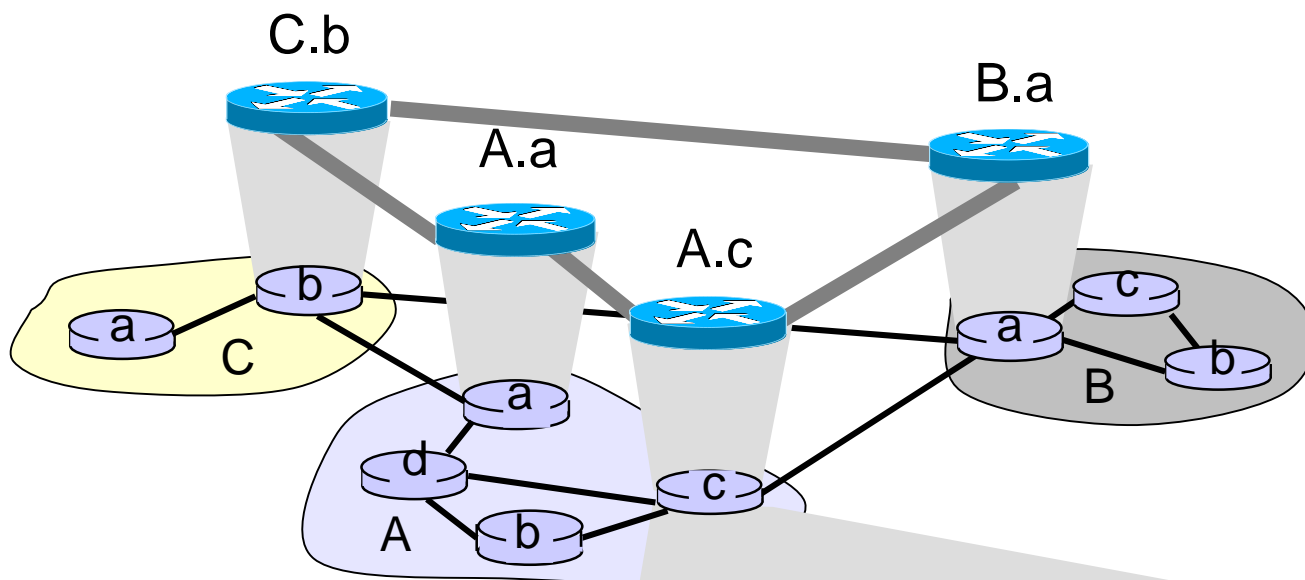
Dividir Internet en regiones, "sistemas autónomos"

- *Autonomous Systems (AS)*
- Enrutamiento intra-AS
 - Entre routers en el mismo AS
 - Todos los routers de un AS deben ejecutar el mismo protocolo de enrutamiento *intra-AS*
 - Los routers en diferentes AS pueden utilizar distintos protocolos intra-AS
 - Routers pasarela (gateway router): en la "frontera" del AS, se comunican con routers en otros AS
- Enrutamiento inter-AS
 - Entre distintos AS
 - Utiliza los routers pasarela (que además también emplean el protocolo intra-AS)

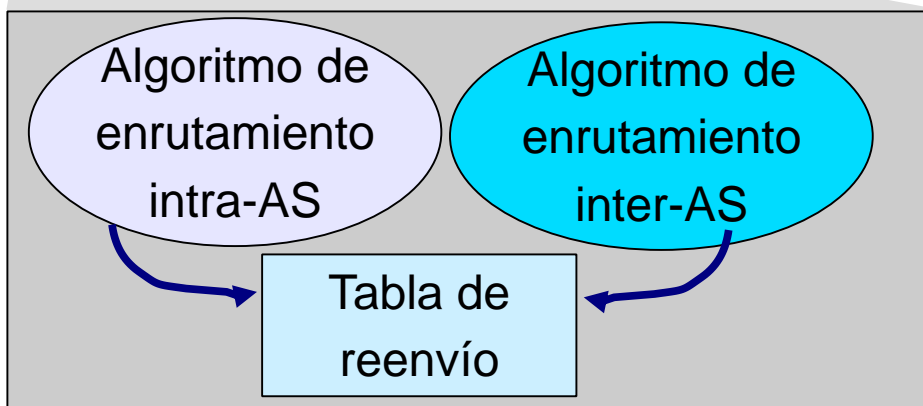
Enrutamiento intra-AS e inter-AS

Tablas de reenvío:
creadas con ambos
tipos de
enrutamiento

- Inter-AS para destinos externos
- Intra-AS para los destinos del AS



Enrutamiento
inter-AS, intra-AS

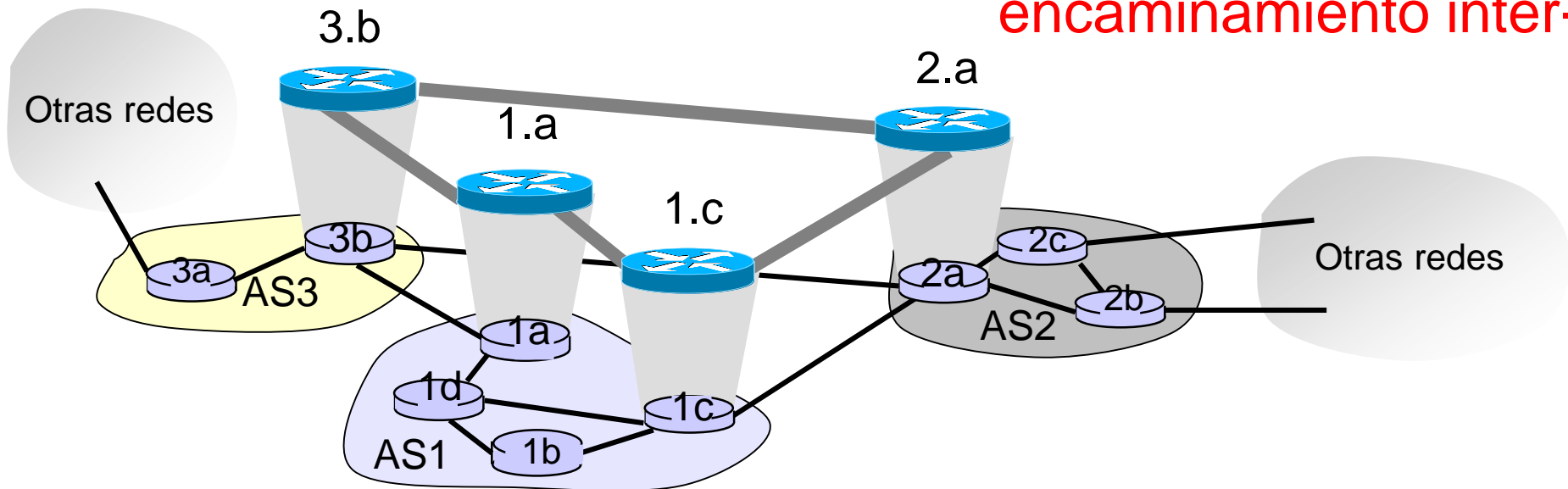


Tareas inter-AS

- Si un router en el AS1 recibe un datagrama destinado fuera de AS1:
 - Debería enviar el paquete a un router pasarela, pero ¿a cuál?

- **AS1 debe:**
 - Aprender qué destinos pueden alcanzarse a través de AS2 o de AS3
 - Propagar esta información a todos los routers de AS1

Esta **tarea** es **del encaminamiento inter-AS**



Protocolos para enrutamiento intra-AS

- También conocidos como *Interior Gateway Protocols (IGP)*
- Protocolos intra-AS más comunes:
 - Vector de distancias
 - RIP: *Routing Information Protocol* (RFC 2453, v2)
 - IGRP: *Interior Gateway Routing Protocol* (propiedad de Cisco hasta 2016)
 - Estado del enlace
 - **OSPF**: Open Shortest Path First (RFC 2328)

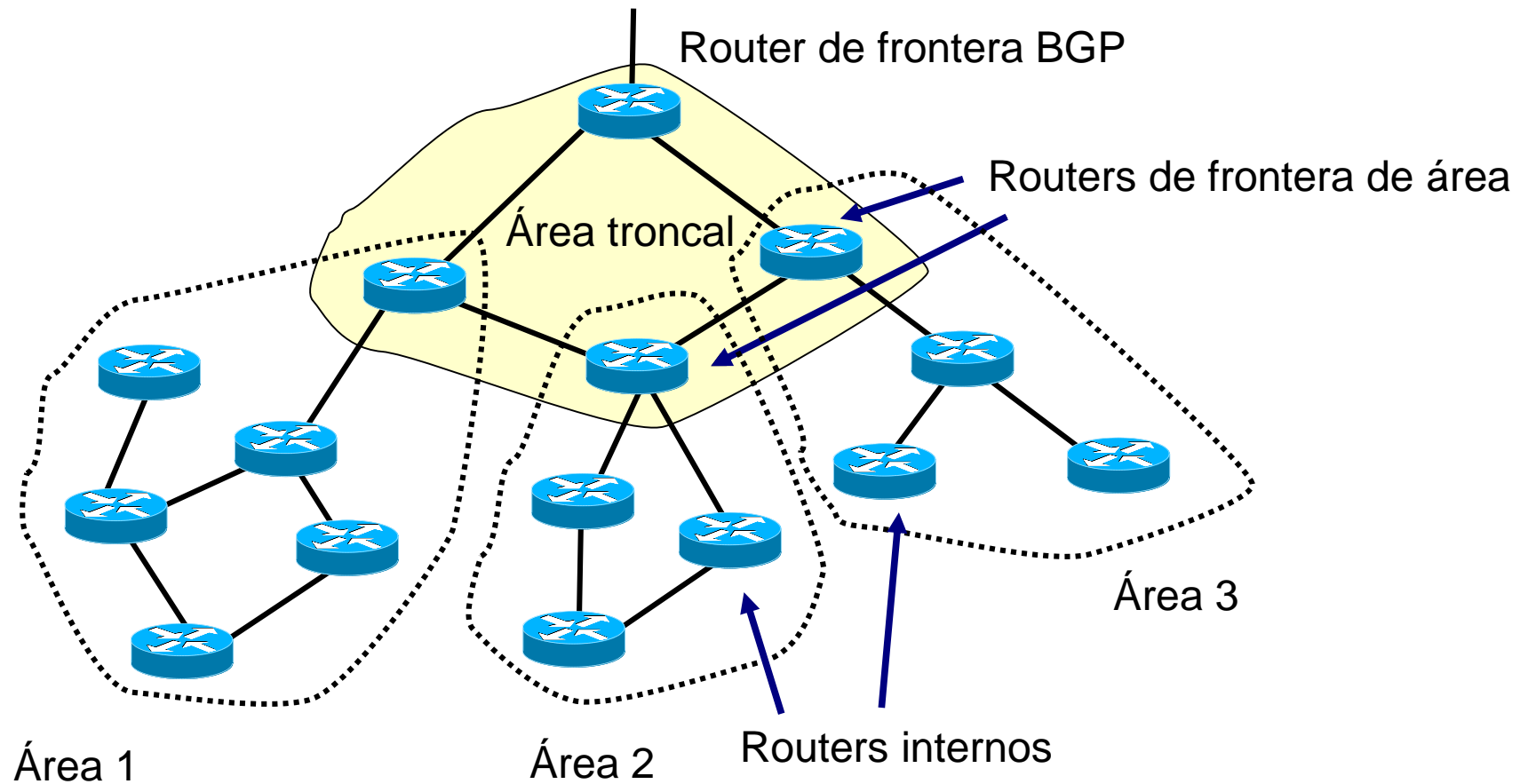
5.2 OSPF (*Open Shortest Path First*)

- “Open”: públicamente disponible (RFC 2328)
- Algoritmo de **estado del enlace**:
 - Difusión de paquetes de estado del enlace a todos los routers del dominio
 - Conocimiento de la topología del dominio
 - Una entrada en el paquete por router vecino
 - Cálculo de rutas mediante el algoritmo de Dijkstra
- La información se envía directamente sobre IP (en lugar de TCP o UDP)

Características “avanzadas” en OSPF

- **Seguridad**: autenticación de mensajes de encaminamiento
 - Para impedir suplantaciones
- Permite **múltiples caminos** con el mismo coste (sólo 1 camino en RIP)
- Permite utilizar **diferentes métricas** para el mismo enlace para diferentes tipos de de servicio (TOS)
- Soporte **multicast**
 - Anuncios específicos para destinos multicast
- En AS grandes se puede usar OSPF **jerárquico**

OSPF jerárquico

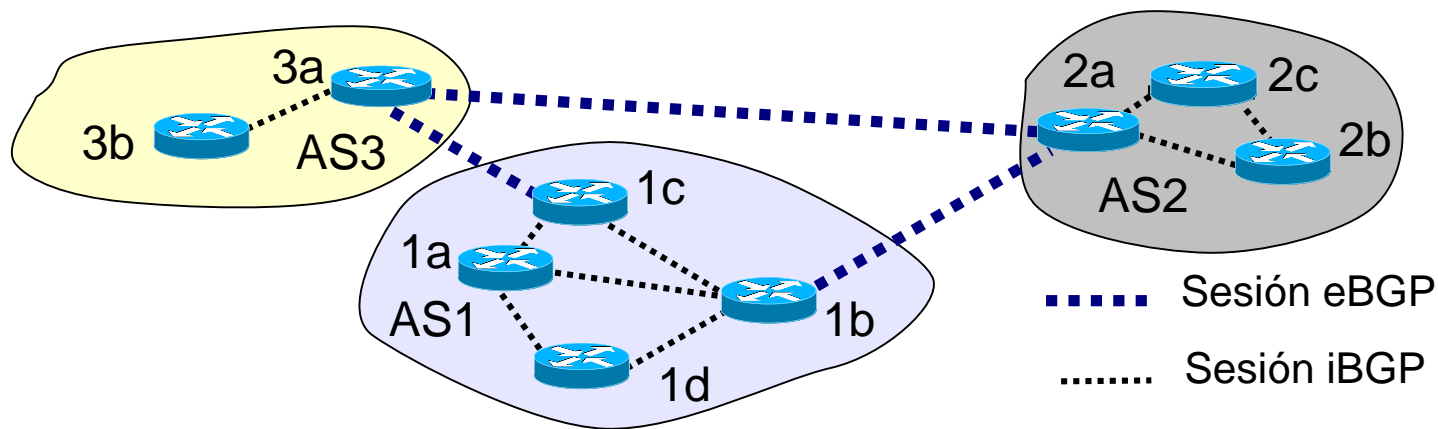


5.3 Enrutamiento entre AS en Internet: BGP

- **BGP (Border Gateway Protocol):** estándar “de facto”
 - Versión actual 4, RFC 1771
- BGP permite a cada AS:
 - Obtener información sobre redes destino que se pueden alcanzar a través de otros AS
 - Propagar esa información a todos los routers internos del AS
 - Obtener “buenas rutas” a las redes destino basadas en información de rutas y políticas de los AS
- Permite a una red informar de su existencia al resto de Internet

Distribución de información

- Con una sesión eBGP entre 3a y 1c, AS3 envía información de alcanzabilidad de prefijo (ej., 138.16.67/24) a AS1.
- 1c puede usar iBGP para distribuir este nuevo alcance de prefijo a todos los routers en AS1
- 1b puede entonces re-anunciar la información de alcance a AS2 a través de la sesión eBGP entre 1b y 2a
- Cuando un router aprende del nuevo prefijo, crea una entrada para ese prefijo en su tabla de re-envío.



Funcionamiento de BGP

- Es un algoritmo por **vector de caminos**:
 - Similar al protocolo por vector de distancias pero trabaja con rutas completas
 - Cada **AS** tiene un **número de identificación único**
 - Cada pasarela informa de la secuencia de AS que se atraviesan hasta alcanzar una red destino
 - Trabajar con rutas completas presenta ventajas:
 - Impide los **bucles** en el encaminamiento

BGP: ejemplo

