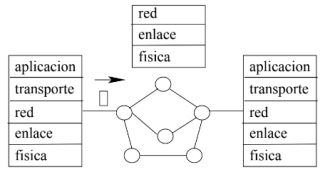
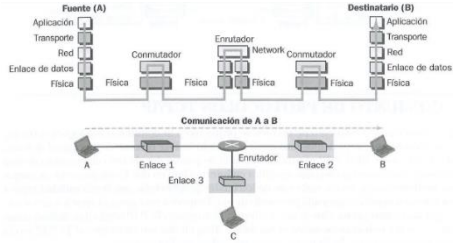
TEMA 4: CAPA DE RED

INTRODUCCIÓN





- La CAPA DE RED es la encargada de llevar los paquetes que le pasa la capa de transporte del host origen al host destino.
  - Está implementada tanto en los **sistemas finales** como en los **routers intermedios**.

COMUNICACIÓN EN LA CAPA DE RED

- El **host origen:**
  1. Acepta un paquete de la capa de transporte.
  2. Lo encapsula en un datagrama (es decir, le añade la cabecera de la capa de red).
  3. Entrega el segmento a la capa de enlace.
- El **host desino**:
  1. Acepta el datagrama de la capa de enlace.
  2. Desencapsula el datagrama (le retira la cabecera de la capa de red).
  3. Entrega el segmento a la capa de transporte.
- Los **routers intermedios** ⟶ en principio no tienen capa de transporte y de aplicación propiamente dichas, pero para su funcionamiento (para los algoritmos de encaminamiento) necesitan **comunicarse entre sí**, para lo que que usan la **capa de transporte y de aplicación.**

SERVICIOS DE LA CAPA DE RED

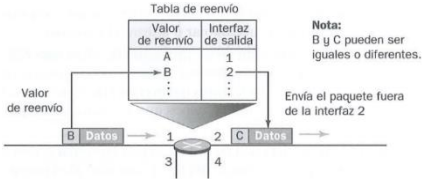
- Son el **encaminamiento**, el **reenvío**, el **control de errores**, el **control de flujo**, el **control de congestión**, la **calidad de servicio** y la **seguridad**.

ENCAMINAMIENTO

- El ENCAMINAMIENTO consiste en enviar un paquete desde el emisor al receptor siguiendo una ruta calculada por un **algoritmo de encaminamiento**.
- Los ALGORITMOS DE ENCAMINAMIENTO se encargan de **obtener la mejor ruta** posible hasta el destino y de **construir las tablas de reenvío** o encaminamiento.

REENVÍO

- El REENVÍO es la acción del router cuando recibe un paquete, consiste en hacerlo pasar a la interfaz de salida apropiada.
  El router **escoge la interfaz de salida** en base a:
  - Su tabla de reenvío.
  - La información en la cabecera del paquete.



CONTROL DE ERRORES

La capa de red realiza el control de errores mediante:

- Una **suma de comprobación** de la cabecera (como en la capa de transporte).
- El **protocolo ICMP** (protocolo de mensajes de control de internet).

CONTROL DE FLUJO

- Está implementado en algunos protocolos de nivel superior (TCP).

CONTROL DE CONGESTIÓN

- Está implementado en algunos protocolos de nivel superior (TCP).
  - ↳ Los routers se encargan de **activar ciertos bits en la cabecera** de la capa de red para indicar que está sufriendo de descarte de paquetes por congestión.

CALIDAD DE SERVICIO (QoS)

- Está implementado en algunos protocolos de nivel superior.
  - ↳ Es un tema muy importante en **nuevas aplicaciones** (como las multimedia).

SEGURIDAD

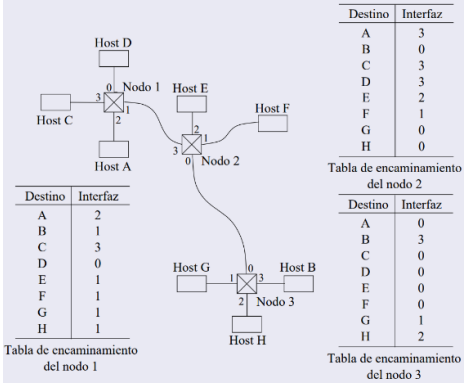
- **Inicialmente** la capa de red **no** incorporaba ningún tipo de seguridad.
- Debido a que es un tema muy importante en la actualidad, ahora se añade un nivel virtual que **cambia el servicio sin conexión por uno orientado a conexión**.

REDES DE CONMUTACIÓN DE PAQUETES

- En las REDES DE CONMUTACIÓN DE PAQUETES no se reservan recursos para cada transmisión: todos se comparten y asignan bajo demanda.
- Hay 2 tipos de redes de conmutación de paquetes: las **redes de datagramas** y las **redes de circuitos virtuales**.

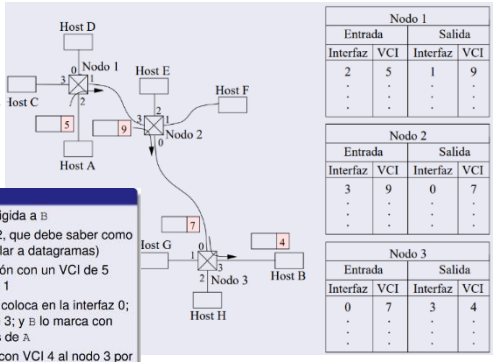
REDES DE DATAGRAMAS

- Son las usadas en la capa de red de **internet**.
- **Encaminamiento en función de destino** ⟶ los paquetes se encaminan en función de la IP destino de su cabecera. Durante el reenvío, el router examina la cabecera del paquete y lo coloca en la interfaz de salida más apropiada usando la **tabla de reenvío**.
- **Sin estado** ⟶ los routers no mantienen información de estado, así que cada paquete es tratado de manera independiente a los paquetes tratados previamente.
- **Red no fiable** ⟶ es una red de **mejor servicio posible**. Esto quiere decir que:
  - No se garantiza { que los paquetes lleguen al destino. / tiempos de llegada. / que los paquetes lleguen en orden. }
  - Aun así, es una buena solución. Se pretende encaminar el **mayor número de paquetes posibles**, aunque **algunos de ellos se pierdan o lleguen mal**.
  - ↳ Si alguna aplicación necesita fiabilidad, puesto que la capa de red no se la puede dar, en la capa de transporte usará TCP.
- **Permite la interconexión** ⟶ como la capa de red es tan simple, hace muy fácil interconectar redes que emplean tecnologías de capa de enlace diferentes.



REDES DE CIRCUITOS VIRTUALES

- Un CIRCUITO VIRTUAL (CV) es una ruta al destino planificada durante el establecimiento de la conexión.
- **Encaminamiento en función del CV** ⟶ los paquetes se encaminan en función del identificador de CV (ICV) de su cabecera. Durante el reenvío, el router examina la cabecera del paquete y lo coloca en la interfaz de salida más apropiada usando la **tabla de circuitos virtuales**.
- **Con estado** ⟶ los routers sí mantienen información de estado en la **tabla de circuitos virtuales**.



Nodo 1			
Entrada		Salida	
Interfaz	VCI	Interfaz	VCI
2	5	1	9

Nodo 2			
Entrada		Salida	
Interfaz	VCI	Interfaz	VCI
3	9	0	7

Nodo 3			
Entrada		Salida	
Interfaz	VCI	Interfaz	VCI
0	7	3	4

TABLA DE CIRCUITOS VIRTUALES

- **Cada nodo** mantiene una tabla con la siguiente información para cada entrada:
  - Interfaz de entrada del CV.
  - ICV de entrada.
  - Interfaz de salida del CV.
  - ICV de salida.
- Un paquete llega por una interfaz de entrada concreta con un determinado ICV y se coloca en la interfaz de salida indicada en la tabla con el nuevo ICV.

Construcción de la tabla de VC

- A envía una *Petición de llamada* dirigida a B
- Esta llega al nodo 1 por la interfaz 2, que debe saber como reenviarla para que llegue a B (similar a datagramas)
- El nodo 1 decide marcar esta petición con un VCI de 5 (aleatorio), y la envía por la interfaz 1
- 2 lo recibe, lo marca con VCI 9 y lo coloca en la interfaz 0; 3 lo marca con VCI 7 y lo coloca en 3; y B lo marca con VCI 4, que identificará los paquetes de A
- B devuelve una *Llamada aceptada* con VCI 4 al nodo 3 por el interfaz 3
- El nodo 3 puede completar su entrada en la tabla (VCI salida = 4); lo mismo los nodos 2 y 1
- 1 manda el ACK a A que lo recibe con VCI 5
  - ⇒ A marca el resto de paquetes a B con VCI 5

ALGORITMOS DE ENCAMINAMIENTO

- El ALGORITMO DE ENCAMINAMIENTO es el encargado de encontrar el camino mínimo entre el origen y el destino rellenando su tabla de reenvío.
  - ↳ Cada host está conectado directamente a su router por defecto. Cuando el host emite un paquete se limita a entregárselo a ese router. Por tanto, el problema se limita a **encontrar el camino mínimo entre dos routers**.
- Encontrar el camino mínimo entre dos routers es un problema equivalente **encontrar el camino mínimo en un grafo ponderado**.
  - Los routers serán los nodos del grafo y los enlaces sus aristas.
  - Las aristas tienen asociado un coste que puede reflejar la distancia del enlace, su velocidad, su nivel de congestión, etc.

CLASIFICACIÓN DE ALGORITMOS DE ENCAMINAMIENTO

- **Globales** (o de estado de los enlaces):
  - Inicialmente cada nodo dispone de toda la información de la red (todos los nodos conectados y el coste de todos los enlaces).
  - El cálculo de los caminos mínimos para formar la tabla de reenvío puede ser realizado por cada nodo por sí sólo gracias a esta información.
- **Descentralizados** (o de vector de distancias):
  - Inicialmente cada nodo sólo conoce la distancia a los nodos vecinos y el enlace por el que se debe comenzar.
  - Los nodos intercambian la información de distancias de la que disponen con sus vecinos.
  - El cálculo de los caminos mínimos para formar la tabla de reenvío se realiza en colaboración de todos los nodos.
- **Estáticos** ⟶ sólo cambian la tabla cuando cambia la topología de la red (añadiendo nuevos nodos o enlaces) o cuando se modifican manualmente ciertos parámetros.
- **Dinámicos** ⟶ se ejecutan periódicamente de forma automática.
  - ↳ Son los usados actualmente en **Internet**.
- **Insensibles a la carga** ⟶ el coste de los enlaces no varía dinámicamente para reflejar su nivel de congestión.
  - ↳ Son los usados actualmente en **Internet**.
- **Sensibles a la carga** ⟶ el coste de los enlaces varía dinámicamente para reflejar su nivel de congestión.
  - Pueden provocar que los mensajes se queden atrapados en un ciclo.

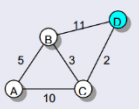
ALGORITMO DE DIJKSTRA FORWARD SEARCH (EE)

- Cada nodo N tendrá 2 listas: $Confirmado$ y $Provisional$.
  - Cada elemento de las listas indica el coste y el siguiente salto a realizar para alcanzar un determinado nodo desde N.
    - ↳ Por ejemplo: $(M, 5, L)$ indica que M se alcanza desde N con coste 5 a través de L.
  - Inicialmente la lista $Confirmado$ tendrá una única entrada para N: $(N, 0, -)$.

1. Examina el LSP (link state packet) del último nodo añadido a $Confirmado$, S.
2. Para cada vecino de S, V, calcula el su coste como $Coste(N,V) = Coste(N,S) + Coste(S,V)$.
   - Si V no está en ninguna lista ⟶ lo añade a $Provisional$ como $(V, Coste, Sigsalto)$.
   - Si V está en $Provisional$ y el coste calculado es menor que el almacenado ⟶ se actualiza el coste de su entrada en la lista.
3. Se pasa la entrada de menor coste de $Provisional$ a $Confirmado$.
4. Repetir hasta que $Provisional$ esté vacía.

CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS

- Tiene **complejidad cuadrática** $O(n^2)$.
- ✓ Se **estabiliza rápidamente**.
- ✓ **No** genera **mucho tráfico**.
- ✓ Responde rápidamente a **cambios en la topología** o fallos de nodos.
- × La cantidad de información (los LSPs) almacenada en cada nodo puede ser bastante grande, lo cual provoca **problemas de escalabilidad**.



Destino	Coste	SigSalto
A	10	C
B	5	C
C	2	C

Tabla de routing del nodo D

Paso	Confirmado	Provisional	Comentarios
1	(D,0,–)		D es el único elemento inicial de **Confirmado**
2	(D,0,–)	(B,11,B) (C,2,C)	El LSP de D dice que puede alcanzar B a coste 11, y C a coste 2. Lo pone en **Provisional**.
3	(D,0,–) (C,2,C)	(B,11,B)	Pasa el miembro de **Provisional** con menor coste (C) a **Confirmado**, y examina su LSP.
4	(D,0,–) (C,2,C)	(B,5,C) (A,12,C)	El coste de alcanzar B a través de C es 5, así que reemplaza (B,11,B) por (B,5,C). El LSP de C indica que puede alcanzar A con coste 10+ 2 a través de C.
5	(D,0,–) (C,2,C) (B,5,C)	(A,12,C)	Pasa el miembro de **Provisional** con menor coste (B) a **Confirmado**, y examina su LSP.
6	(D,0,–) (C,2,C) (B,5,C)	(A,10,C)	El LSP de B dice que puede alcanzar A a coste 5, así que cambia (A,12,C) por (A,10,C) (el coste D–B es 5 a través de C)
7	(D,0,–) (C,2,C) (B,5,C) (A,10,C)		Pasa el miembro de **Provisional** con menor coste (A) a **Confirmado**, y ya está.

ALGORITMO DESCENTRALIZADO (VD)

- Cada nodo tendrá una **tabla de distancias** en la que almacenará la información que recibe de sus vecinos.
  - Para cada nodo de la red, almacena la distancia hasta él partiendo desde cada uno de sus vecinos.
  - Inicialmente en esta tabla sólo estarán los costes de los enlaces a los nodos vecinos.

1. **Iteративamente**, los nodos comunican a sus vecinos toda la información de sus tablas.
2. Al recibir información, los nodos calculan distancias a nuevos nodos o actualizan con un valor menor la distancia a nodos ya conocidos.
   - ↳ Como siguiente salto colocarán al nodo que les envió la información.
3. Las actualizaciones continúan hasta que convergen (es decir, se producen varias iteraciones sin ninguna nueva).
4. A partir de las tablas de distancia calculadas se obtiene la tabla de reenvío.

Sea el nodo x con un vecino z, cuyo enlace tiene coste $c_{x,z}$, y que z envía $d_{z,y}$ ⇒

$D_{x,y}(z) =$ distancia de x a y a través de la salida z = $c_{x,z}$ + coste del enlace que une x con z + $d_{z,y}$ distancia de z a y

Tabla de distancias del nodo x

		distancia por z	distancia por z'
destino	y	$D_{x,y}(z)$	$D_{x,y}(z')$
	y'	$D_{x,y'}(z)$	$D_{x,y'}(z')$

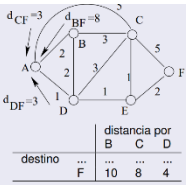
CARACTERÍSTICAS Y COMPARACIÓN CON GLOBAL

- El intercambio de información con los vecinos se realiza **periódicamente**.
- La **propagación del cambio en el coste de un enlace** se realiza en varias iteraciones:
  - Si es una disminución → los nodos se dan cuenta en seguida de que hay un camino más corto y actualizan rápidamente sus tablas.
  - Si es un aumento → las tablas tardan mucho en actualizarse porque los caminos con distancias mayores a los ya existentes son ignorados por los nodos.
    - ↳ Existen varias técnicas para solucionar este problema, como el **horizonte dividido** y el **inverso envenenado**.
- × Su **coste** depende del número de iteraciones que haga, que puede ser muy grande, por lo que es **mayor** que el global.
- × Es **menos robusto** que el global, pues si un nodo calcula mal una distancia, todos los demás usarán este valor incorrecto.

- Nodos B, C y D envían a A la distancia a F
- A calcula las distancias

$D_{A,F}(B) = c_{A,B} + d_{B,F} = 2 + 8 = 10$
$D_{A,F}(C) = c_{A,C} + d_{C,F} = 5 + 3 = 8$
$D_{A,F}(D) = c_{A,D} + d_{D,F} = 1 + 3 = 4$

- La tabla de distancias para A



destino	distancia por B	C	D
...	...	...	...
F	10	8	4

- A comunica a sus vecinos la distancia a F, $d_{A,F} = \min_z D_{A,F}(z) = 4$
- Después de ciertas iteraciones, converge

destino	distancia por B	C	D
B	2	8	3
C	5	5	3
D	4	7	1
E	5	6	2
F	7	8	4

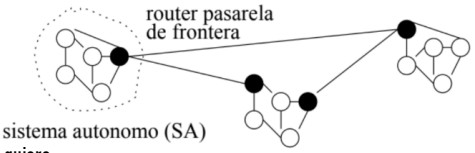
Tabla de distancias de A

destino	salida
B	B
D	D
C	D
E	D
F	D

Tabla de rutas de A

ENCAMINAMIENTO JERÁRQUICO

- Las redes grandes como Internet se dividen en regiones denominadas SISTEMAS AUTÓNOMOS (SA).
  - ↳ Las SAs suelen estar operadas por una **empresa** u **organismo**.
- Los routers de una SA sólo conocen los detalles de encaminamiento en su región, por lo que el tráfico de salida de la región se centraliza a través de los ROUTERS PASARELA DE FRONTERA.
- Así, existen dos **niveles de encaminamiento**:
  - Encaminamiento **intradominio** (en los routers de dentro del SA) → cada SA puede elegir el algoritmo que quiera.
  - Encaminamiento **interdominio** (en los routers pasarela de frontera) → todos los SAs deben usar el mismo algoritmo (BGP).



PROTOCOLOS DE ENCAMINAMIENTO EN INTERNET

- Los protocolos de encaminamiento en Internet son protocolos de la **capa de aplicación**.

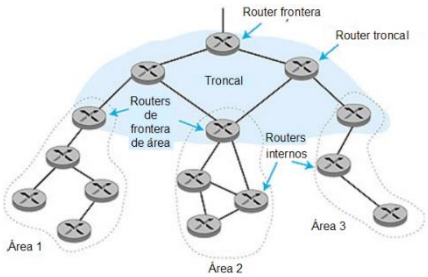
Nivel de encaminamiento	Nombre	Tipo	Protocolo de la capa de transporte/red
Protocolos de encaminamiento **intradominio**	RIP	Vector de distancias	UDP/IP
	OSPF	Estado de los enlaces	Propio/IP
Protocolos de encaminamiento **interdominio**	BGP	Vector de rutas	TCP/IP

RIP - protocolo de información de encaminamiento

- Es un protocolo de encaminamiento **intradominio** de tipo **vector de distancias** que usa UDP/IP.
- Considera que el **coste** de todos los enlaces es **1** y que la **distancia máxima** es 15, es decir, sólo se pueden encaminar paquetes a través de 15 routers intermedios.
- Los **mensajes** RIP se envían sólo a los nodos **vecinos**.
  - Mensajes de **petición** RIP → solicitan información a los vecinos.
  - Mensajes de **respuesta** RIP → contienen una lista de hasta 25 redes internas del SA.
    - Los mensajes respuesta no sólo se envían cuando se recibe una petición, también se envían **periódicamente,** cada 30s.
      Si un router no tiene noticias de un vecino en 3m, lo considerará **caído**, por lo que modificará su tabla de rutas y anunciará esta información a sus vecinos.

OSPF - primero el camino más corto

- La O de OSPF viene de *Open* y se refiere a que es un algoritmo libre, es decir, no patentado.
- Es un protocolo de encaminamiento **intradominio** de tipo **estado de los enlaces** que usa su propio protocolo de transporte.
  - ↳ Es más avanzado que el RIP y fue pensado para reemplazarlo.
- El **coste** de los enlaces es **establecido por el administrador**.
- Los **mensajes** OSPF se difunden a **todos los nodos del SA**.
  - La difusión se realiza cuando { se produce algún cambio en la red / periódicamente, al menos una vez cada 30 min }
  - Mensajes de **saludo** OSPF → se envían a los vecinos para comprobar que los enlaces funcionan.
  - Mensajes de **interrogación** OSPF → se solicita a un vecino toda su información.

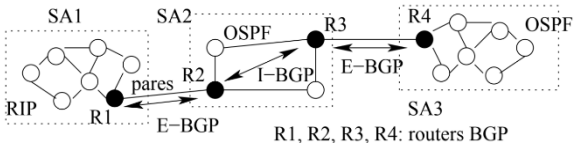


CARACTERÍSTICAS

- **Seguridad** → OSPF implementa su propio protocolo de transporte.
  - ↳ Todos los mensajes enviados desde un router OSPF van **autentificados**, por tanto, sólo se tendrán en cuenta los **routers fiables**, el resto se ignoran.
- **Múltiples caminos de un mismo coste** → permite repartir el tráfico con el mismo destino a través de caminos alternativos.
- **Soporte de jerarquía** → permite que cada SA se divida en áreas. Cada área ejecuta OSPF sobre sus routers y los mensajes no salen de ella.
  - Las áreas se comunican entre sí a través de los **routers de frontera de área**: cuando un paquete tiene un destino fuera del área se encamina primero a este router.
    - ↳ Los routers de frontera de área están interconectados entre sí en un **área troncal** dentro del SA.
  - Para sacar los paquetes fuera del SA, se usa un **router de pasarela de frontera**.

BGP – protocolo de pasarela de frontera

- Es el encaminamiento **interdominio** estándar en **Internet,** de tipo **descentralizado** y que usa TCP/IP.
- Su funcionamiento es **similar al de VD** pero intercambiando rutas completas, por lo que es mejor denominarlo **vector de rutas**.
- Los routers pasarela de frontera se comunican entre sí para intercambiar rutas. Hay dos **tipos de comunicaciones**:
  - **External-BGP** → se usa entre routers BGP vecinos, denominados **pares BGP.**
  - **Internal-BGP** → se usa entre routers BGP del mismo SA, denominados **vecinos lógicos** pues, aunque no tengan ninguna conexión directa con un enlace, se supone que tienen una conexión lógica usando los routers del SA.
- Cada SA se identifica por un **número de sistema autónomo** único (ASN).
- Los administradores de los SAs pueden seleccionar sus **políticas de encaminamiento**.
  - Por ejemplo, pueden decidir vetar cierto proveedor de internet haciendo que sus routers BGP no anuncien ninguna ruta a los routers BGP de la otra compañía).



Ejemplo de mensaje en RIP (vector de distancias):

red destino	métrica
x	4

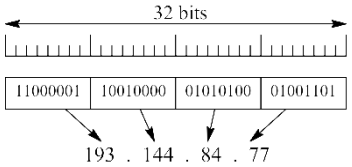
Ejemplo de mensaje en BGP (vector de rutas):

red destino	ruta
x	SA1/SA2/SA3/S4

IP: PROTOCOLO DE INTERNET

- Los componentes de la capa de red en internet son:
  - El protocolo de red **IP,** que define { el formato de las direcciones. / el formato de las cabeceras de los datagramas. / las acciones que realizan los routers en función de la cabecera de los datagramas.
  - Los **protocolos de encaminamiento** que determinan la ruta que deben seguir los datagramas.
  - El **protocolo ICMP** usado para control de errores.
- El **protocolo de Internet** (IP) es un protocolo de redes de conmutación de paquetes de **datagramas**.

DIRECCIONAMIENTO IPv4



- Las **direcciones IPv4** son números de 4 bytes, generalmente representados en notación decimal y separados por puntos. Se dividen en 2 secciones:
  - Identificador de **red** (común a todos los nodos de una red).
  - Identificador de **interfaz** del nodo dentro de la red.
- Cada nodo de una red tiene **una dirección IP por interfaz**.
  - Una INTERFAZ es la unión de un nodo con un enlace.
- **Direcciones especiales reservadas**:
  - **Identificación de red** → campo de estación a 0s.
  - **Dirección de broadcast a una red** → campo de estación a 1s.
  - 0.0.0.0 → ninguna dirección en concreto (ruta por defecto en tablas de reenvío)
  - 127.0.0.0 – 127.255.255.255 → la propia red (dirección de *loopback*, se suele usar la 127.0.0.1).
  - 240.0.0.0 – 255.255.255.254 → reservadas para uso futuro.
  - 255.255.255.255 → dirección de broadcast a toda la red.

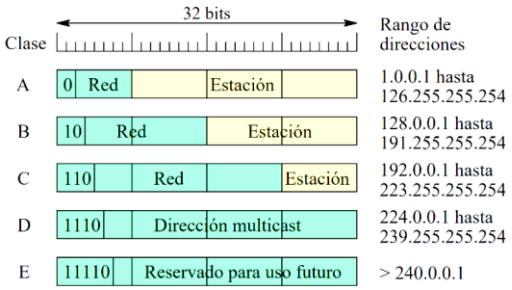
SUBREDES

- Una red con muchas estaciones se puede dividir en **subredes** de manera que { cada subred se gestione de manera independiente dentro de la red. / todas las subredes actúen como una única red de cara a exterior.
  - Se utiliza parte del campo estación para delimitar la subred.
- La **máscara de red** permite indicar a qué subred pertenece una estación.
  Una máscara $/n$ indica que se usan los n MSB de la dirección para especificar la red y subred.
  - Se puede representar con n bits puestos a 1 y $32 - n$ a 0.

DIRECCIONES CON CLASE

- Inicialmente, las direcciones de red se dividieron en **clases**:
  - El objetivo de las clases era asignar a grandes organizaciones una red de clase A, a las medianas una de clase B y a las pequeñas una de clase C.
    Las redes de clase D se usarían para multicast (envío a más de un dispositivo) y las de clase E estaban reservadas para uso futuro.

Clase	Primer byte	Segundo byte	Tercer byte	Cuarto Byte	Rango de direcciones
A	0 | Red	Estación			1.0.0.1 – 126.255.255.254
B	10 | Red		Estación		128.0.0.1 – 191.255.255.254
C	110 | Red			Estación	192.0.0.1 – 223.255.255.254
D	1110 | Dirección multicast				224.0.0.1 – 239.255.255.254
E	11110 | Reservado para uso futuro				240.0.0.1 – 255.255.255.254



- Un campo de red o estación no puede estar todo a 1s o todo a 0s, así que existen:
  - $A \rightarrow 2^7 - 2 = 127$ redes con $2^{24} - 2 \approx 16M$ estaciones cada una.
  - $B \rightarrow 2^{14} \approx 16K$ redes con $2^{16} - 2 \approx 16K$ estaciones cada una.
  - $C \rightarrow 2^{24} \approx 2M$ redes con $2^8 - 2 = 254$ estaciones cada una.

- Clase A → 10.0.0.0
- Clase B → 172.16.0.0
- Clase C → 193.144.84.0

- Clase A → 10.255.255.255
- Clase B → 172.16.255.255
- Clase C → 193.144.84.255

DIRECCIONES SIN CLASE

- En 1993 se suprimen las clases, estableciéndose las **direcciones CIDR** (enrutamiento entre dominios sin clases).
- En estas direcciones se puede elegir cuántos bits forman el campo de red y cuántos el de estación.
  - Esto se indica mediante la máscara de red.

- Dirección clase C 193.168.17.0/27 (o máscara 255.255.255.224)
  - Los 24 primeros bits indican la red (192.168.17)
  - Los 3 siguientes la subred
  - Los 5 últimos la posición de la estación en la subred
  - Tenemos $2^3 = 8$ subredes, con $2^5 - 2 = 30$ estaciones por subred
  - En total, podemos direccionar $8 \times 30 = 240$ estaciones (254 en clase C sin máscara)

Nº de subred	Dir. base	Dir. broadcast
0	193.168.17.0	193.168.17.31
1	193.168.17.32	193.168.17.63
2	193.168.17.64	193.168.17.95
3	193.168.17.96	193.168.17.127
4	193.168.17.128	193.168.17.159
5	193.168.17.160	193.168.17.191
6	193.168.17.192	193.168.17.223
7	193.168.17.224	193.168.17.255

- Dirección 193.168.17.133/27
  - ¿A qué subred pertenece?

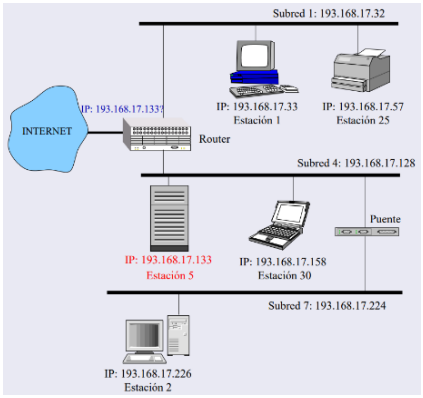
193.168.17.133 → 11000001.10101000.00010001.10000101 ⊗
255.255.255.224 → 11111111.11111111.11111111.11100000

11000001.10101000.00010001.10000000
Red — Subred 4

  - ¿Qué posición ocupa en la subred?

193.168.17.133 → 11000001.10101000.00010001.10000101 ⊗
255.255.255.224 → 00000000.00000000.00000000.00011111

00000000.00000000.00000000.00000101
Estación 5



- Ejemplo: 193.168.173.253/18
  - Nº de red: 11000001.10101000.10000000.00000000 = 193.168.128.0
  - Broadcast: 11000001.10101000.10111111.11111111 = 193.168.191.255
  - Nº estación: 11000001.10101000.10101101.11111101 = estación nº 11773
  - Nº total de estaciones: $2^{14} - 2 = 16382$

193.144.48.0/20 en dos subredes				
193	144	0011-0-000	0	subred 193.144.48.0/21
193	144	0011-1-000	0	subred 193.144.56.0/21

193.144.48.0/20 en 8 subredes				
193	144	0011-000-0	0	subred 193.144.48.0/23
193	144	0011-001-0	0	subred 193.144.50.0/23
...				
193	144	0011-111-0	0	subred 193.144.62.0/23

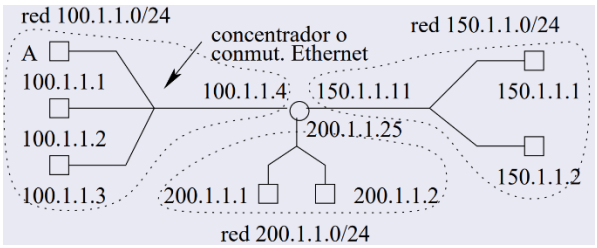
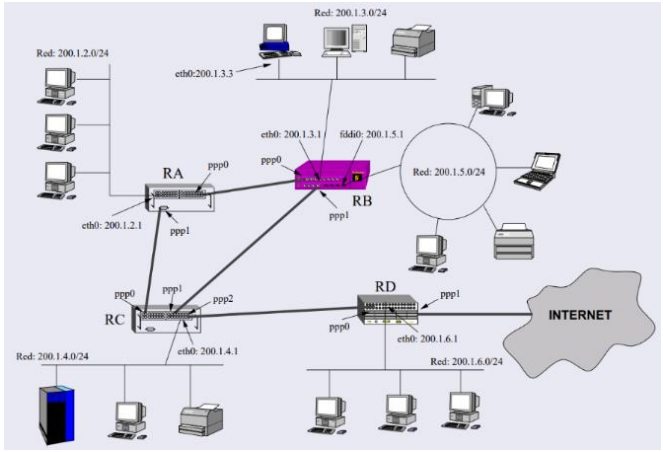


Tabla de rutas del host A			
destino	interfaz	gateway	métrica
100.1.1.0/24	eth0 (=100.1.1.1)	*	0
150.1.1.0/24	eth0 (=100.1.1.1)	100.1.1.4	1
200.1.1.0/24	eth0 (=100.1.1.1)	100.1.1.4	1

Tabla de rutas del router			
destino	interfaz	gateway	métrica
100.1.1.0/24	eth0 (=100.1.1.4)	*	0
150.1.1.0/24	eth1 (=150.1.1.11)	*	0
200.1.1.0/24	eth2 (=200.1.1.25)	*	0



ROUTER A

Red destino	Gateway	Interfaz
200.1.2.0	200.1.2.1	eth0
200.1.3.0	*	ppp0
200.1.4.0	*	ppp1
200.1.5.0	*	ppp0
200.1.6.0	*	ppp1
0.0.0.0	*	ppp1

ROUTER B

Red destino	Gateway	Interfaz
200.1.2.0	*	ppp0
200.1.3.0	200.1.3.1	eth0
200.1.4.0	*	ppp1
200.1.5.0	200.1.5.1	fddi0
200.1.6.0	*	ppp1
0.0.0.0	*	ppp1

ROUTER C

Red destino	Gateway	Interfaz
200.1.2.0	*	ppp0
200.1.3.0	*	ppp1
200.1.4.0	200.1.4.1	eth0
200.1.5.0	*	ppp1
200.1.6.0	*	ppp2
0.0.0.0	*	ppp2

ROUTER D

Red destino	Gateway	Interfaz
200.1.2.0	*	ppp0
200.1.3.0	*	ppp0
200.1.4.0	*	ppp0
200.1.5.0	*	ppp0
200.1.6.0	200.1.6.1	eth0
0.0.0.0	*	ppp1

Tabla de reenvío del host 200.1.3.3		
Red destino	Gateway	Interfaz
127.0.0.0/8	*	lo
200.1.3.0/24	*	eth0
0.0.0.0/0 (por defecto)	200.1.3.1	eth0

COINCIDENCIA DEL PREFIJO MÁS LARGO

- Para saber a qué red de la tabla de reenvío de un router se envía un paquete se realiza un and de la **dirección de destino** con **cada máscara de la tabla** buscando que se produzca alguna coincidencia de prefijo (el resultado coincida con el identificador red de la máscara usada).
- Si hay varias coincidencias se toma la que tenga la **máscara más grande** (el mayor prefijo).

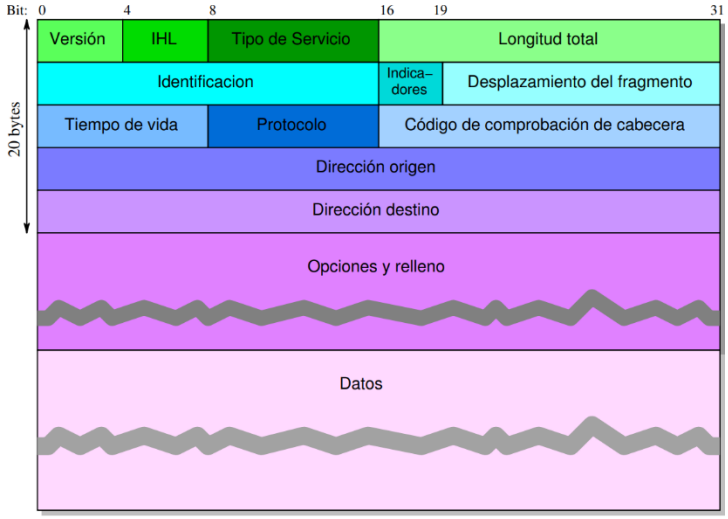
Tabla de reenvío de un router			
destino	interfaz	gateway	métrica
...	...	...	...
194.24.0.0/21	int_i	*	i saltos
194.24.8.0/22	int_j	*	j saltos
194.24.16.0/20	int_k	*	k saltos
...	...	...	...
0.0.0.0/0 (por defecto)	int_x	*	x saltos

- ¿Qué ocurre cuando le llega un paquete a 194.24.17.4?
  - Coincidencia de prefijo: se realiza un AND con cada máscara hasta que se produzca la coincidencia de prefijo
    - Con 194.24.0.0/21
      194. 24.00010001.00000100 = 194.24.17.4
      255.255.11111000.00000000 = 255.255.248.0
      194. 24.00010000.00000000 = **194.24.16.0**
      No coincide con la dirección base de la red (194.24.0.0)
    - Con 194.24.16.0/20
      194. 24.00010001.00000100 = 194.24.17.4
      255.255.11110000.00000000 = 255.255.240.0
      194. 24.00010000.00000000 = **194.24.16.0**
      Sí coincide con la dirección base de la red (194.24.16.0)
      $\Longrightarrow$ se envía por la interfaz correspondiente
    - Si hubiese otra coincidencia con un prefijo más largo (máscara más grande), se reenviaría por la interfaz asociada a esa entrada

AGREGACIÓN DE RUTAS

- La **agregación de rutas** es un proceso que realizan los routers en el que toman un bloque CIDR (un grupo de direcciones contiguas) y las anuncian resumidas en una única dirección de red común a todas ellas.
- Para obtener la **dirección base** de esta red se comprueba que todas las direcciones son contiguas y se les realiza un and.
- Para obtener la **máscara de red** se toma la longitud la parte común de todas ellas.
- Así se optimiza el encaminamiento de las grandes redes corporativas, pues las **tablas de reenvío** de los routers **mantienen menos entradas**

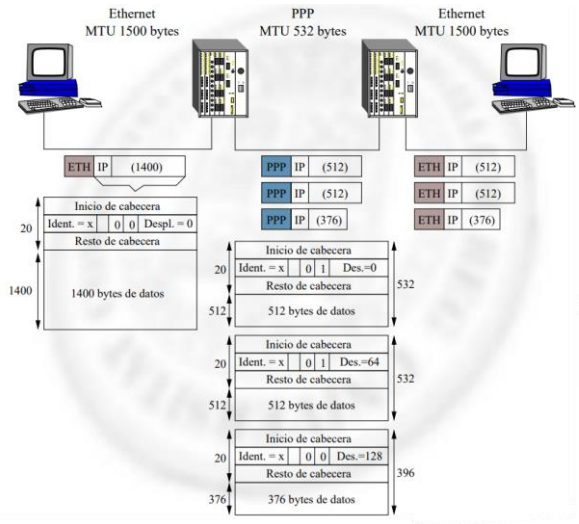
FORMATO DEL DATAGRAMA IPv4

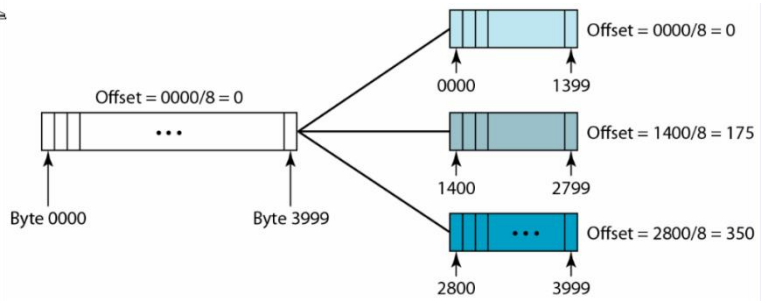


- **Versión** ⟶ IPv4/IPv6.
- **IHL** ⟶ longitud de la cabecera en palabras de 32 bits.
- **Tipo de servicio** ⟶ para diferenciar distintos tipos de datagramas IP.
  - ↳ Incluye los bits ECN.
- **Longitud total** ⟶ longitud total del datagrama (cabecera + datos) con todos sus fragmentos incluidos en bytes.
- **Identificación, indicadores** y **desplazamiento del fragmento** ⟶ para fragmentación.
- **Tiempo de vida** ⟶ comienza en cierto valor y se resta 1 cada vez que el datagrama es reenviado. Cuando llegue a 0 el datagrama se descarta.
- **Protocolo** ⟶ protocolo de la capa de transporte al cual se le pasará el datagrama.
- **Suma de comprobación de cabecera** ⟶ se procede como en la capa de transporte, pero considerando sólo la cabecera y palabras de 32 bits.
- **Dirección origen** y **destino**.
- **Opciones** ⟶ para ampliar la cabecera (p. ej. especificar la ruta desde origen)
- **Datos** ⟶ segmento de la capa de transporte.

FRAGMENTACIÓN

- Como se dijo antes, IP está pensado para conectar redes muy distintas.
  - Distintas redes manejan distintos tamaños de trama, por lo que el tamaño máximo de un datagrama IP que viaje por cierta red no puede superar su **MTU** (unidad máxima de transmisión), que será la cantidad máxima de datos de una trama que puede manejar.
- Por tanto, en algunos casos los routers tendrán que fragmentar los datagramas para que atraviesen ciertas redes.
  - Para identificar un fragmento se usan 3 campos de la cabecera:
    - **Identificador** ⟶ número del datagrama (todos los fragmentos de un datagrama tienen el mismo identificador).
    - **Indicadores** (3 bits) $\begin{cases} \text{un 0.} \\ NF \longrightarrow \text{activo para indicar que no se puede fragmentar el datagrama, si no cabe en la red a la que se tiene que enviar, se desechará.} \\ MF \longrightarrow \text{activo en todos los fragmentos de un datagrama, excepto el último.} \end{cases}$
    - **Desplazamiento** ⟶ posición del fragmento dentro del datagrama original. Se calcula dividiendo entre 8 la posición del primer byte del fragmento en los datos del datagrama original. Así, el primero byte del fragmento tenderá que ser múltiplo de 8.
- Los fragmentos no se vuelven a **unir** hasta llegar al **destino**, donde IP se encarga de recuperar todos los fragmentos de un segmento y unirlos.
- Se procura que **TCP y UDP generen segmentos pequeños** para evitar fragmentar siempre que sea posible, ya que introduce complicaciones adicionales.





SEGURIDAD EN IPv4

- Rastreo de paquetes (packet sniffing) ⟶ ataque pasivo, no modifica el contenido.
- Modificación de paquetes ⟶ el receptor cree que viene del remitente original.
- Falsificación de IP (spoofing) ⟶ suplantación de identidad.
  - Para solucionar todo esto a IP se le añadió **IPSec**, que crea un **servicio orientado a conexión** entre dos entidades. Los servicios que proporciona son:
    - Definición de **algoritmos** y **claves** entre emisor y receptor.
    - **Cifrado** de paquetes (soluciona el packet sniffing).
    - **Integridad** de los datos (soluciona la modificación).
    - **Autenticación** en el origen (soluciona el spoofing).

PROBLEMAS DE IPv4

- Las direcciones de IPv4 se están **agotando**.
- Además, se necesita $\begin{cases} \textbf{simplificar} \text{ el protocolo para que los routers sean más eficientes.} \\ \text{proporcionar mayor } \textbf{seguridad} \text{ a la red.} \end{cases}$
  - Como respuesta, surge **IPv6**.

DIRECCIONAMIENTO IPv6

- Las **direcciones IPv6** son números de 16 bytes, generalmente representados en 8 campos de 2 bytes cada uno en notación hexadecimal separados por puntos.
  - **No** tienen **clases**.
  - Los **grupos de cuatro 0s** se abrevian con ::.
  - Las **direcciones IPv4** se pueden escribir en este formato.
  - Proporcionan servicios en **tiempo real**.
  - Proporcionan servicios de **autenticación** y **seguridad**.

```
47CD:0000:0000:0000:0000:0000:A456:0124
⇔ 47CD::A456:0124

::FFFF:193.144.84.77
```

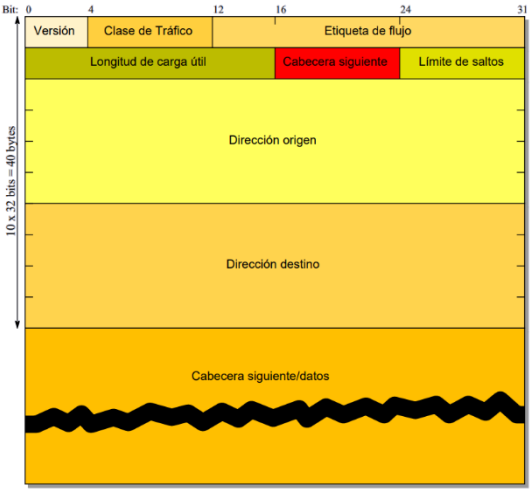
- Se usan **máscaras** para diferenciar la parte de red y de host de la misma manera que se hacía en IPv4.

dirección:	3ffe:ffff:100:1:2:3:4:5/48
máscara:	ffff:ffff:ffff:0000:0000:0000:0000:0000
red:	3ffe:ffff:0100:0000:0000:0000:0000:0000

- Permiten envío:
  - **Unicast** (a un host).
  - **Multicast** (a un conjunto de hosts) → la dirección empieza por FF.
  - **Anycast** (a un host cualquiera de la red) → se toma la dirección unicast con 0s en el campo de host.
  - **No** existe el **broadcast** (a todos los hosts de la red), se usa el multicast en su lugar.

CABECERA IPv6

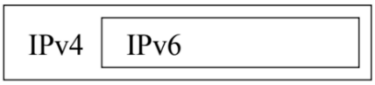
- La cabecera IPv6 pese a ser **más larga** (40 bytes) que la IPv4, es **más sencilla**, pues tiene menos campos (tan sólo 8).



- Los campos **eliminados** son:
  - **Opciones** → se indican en cabeceras adicionales.
  - **Longitud de cabecera** → todas son de 40 bytes.
  - **Fragmentación** → ya no se fragmentan datagramas.
    - Si un datagrama es demasiado grande para una red, se descarta y se devuelve un mensaje ICMP para que el emisor envíe los datos en segmentos más pequeños.
  - **Suma de comprobación** → para ahorrar procesamiento en el router. La comprobación de errores se delega en las capas de transporte o enlace.
- Los **nuevos** campos son:
  - **Clase de tráfico** y **etiqueta de flujo** → relacionados con la QoS.
  - **Longitud de carga útil** → longitud del datagrama (sólo de los datos) en bytes.
  - **Cabecera siguiente** → como el campo de protocolo de IPv4.
    Si hay opciones, especifica que hay otra cabecera con opciones.
  - **Límite de saltos** → como el campo de TTL de IPv4.

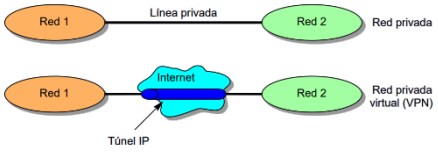
TRANSICIÓN DE IPv4 A IPv6

- El cambio tiene que ser **gradual** y durante un tiempo **coexistirán las dos**.
- Para lograr esto se usará la **tunelización**, que consiste en encapsular los paquetes IPv6 dentro de paquetes IPv4 añadiéndoles una cabecera IPv4 redundante cuando tengan que pasar por routers que no soportan IPv6. Esta cabecera se eliminará cuando lleguen a una zona donde los routers sí puedan manejar IPv6.



VPN – Virtual Public Network

- Las **redes privadas virtuales** son redes de una organización que usa la red pública para comunicarse de forma segura, como si fuese una red privada.
- Para conseguir esto usan sistemas de **encriptación** y **autenticación** como:
  - **IPSec**.
  - **Túneles IP**:
    1. R_1 cifra el datagrama (incluyendo la cabecera) usando la clave pública de R_2.
    2. R_1 le añade una nueva cabecera al datagrama con destino R_2.
    3. R_2 descifra los datos del datagrama recibido, que incluyen su cabecera real, que usará para enviarlo al destino.



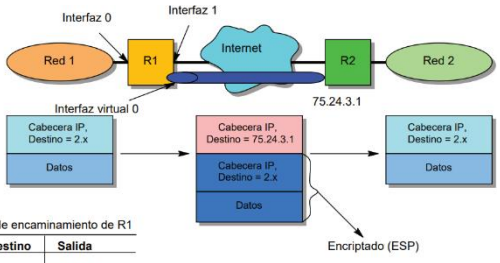


Tabla de encaminamiento de R1

Red destino	Salida
1	Interfaz 0
2	Interfaz virtual 0
Por defecto	Interfaz 1

ICMP: PROTOCOLO DE MENSAJES DE CONTROL DE INTERNET

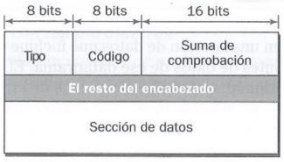
- El ICMP se usa para que los hosts y routers puedan informarse sobre **errores** o sobre el **estado** de la red.
- Es un protocolo de la **capa de transporte** pues, aunque se suele considerar parte de IP, sus mensajes se transportan en datagramas IP.
  - Por tanto, no se garantiza la entrega de los mensajes.

MENSAJES ICMP

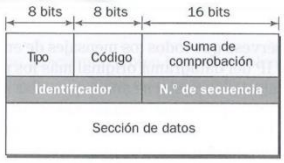
- Los mensajes ICMP se componen por:
  - **Tipo** (8 bits) → tecleados por los usuarios.
  - **Código** (8 bits) → devueltas por el servidor.
  - **Suma de comprobación** (16 bits).
  - **Datos**.
- Hay dos tipos de mensajes ICMP:
  - Mensajes de **informe de error** → informan al origen de errores durante el procesamiento.
    - En el campo datos llevan la **cabecera** y los **8 primeros bytes del datagrama** que causó el envío del mensaje ICMP
  - Mensajes de **consulta** → sondean la actividad de routers y hosts.
    - Se usan para obtener el **RTT** entre dos dispositivos o averiguar si sus **relojes** están **sincronizados**.
    - Declarados obsoletos por el IETF: solicitud de información y repetición (ARP), solicitud y respuesta de máscara de dirección (DHCP), solicitud y publicidad de routers (DHCP).

Valores de tipo y código

Mensajes de error	Mensajes de consulta
03: Destino inalcanzable (códigos 0 a 15) también si NF = 1 y no cabe en la Red	08 y 00: Solicitud de eco y respuesta (solo código 0) → para Ping y traceroute
04: Silenciar origen (solo código 0) limita los datagramas enviados para evitar la congest	13 y 14: Solicitud y respuesta de marca de tiempo (solo código 0) → para medir retardos
05: Redirección (códigos 0 a 3) si el nodo se da cuenta de que hay un mejor camino	
11: Tiempo excedido (códigos 0 y 1) TTL llega a 0	
12: Problema con parámetros (códigos 0 y 1) valor ilegal en la cabecera	



Mensajes de error



Mensajes de consulta

DHCP: PROTOCOLO DE CONFIGURACIÓN DINÁMICA DE LOS HOSTS

- La asignación de direcciones IP se puede realizar estáticamente, cuando es el administrador del equipo quien la establece, o dinámicamente, cuando se usa DHCP.
  - Se usa {cuando los ISPs no tienen suficientes direcciones para todos los abonados / en las redes inalámbricas.

1. El host envía un **mensaje de descubrimiento de servidor DHCP** $DHCPDISCOVER$ → la IP origen será $0.0.0.0$ y la destino $255.255.255.255$.
2. El servidor envía un **mensaje de ofrecimiento DHCP** → contendrá una (o varias) dirección IP, gateway por defecto, servidores DNS, máscara de red, etc.
3. El host envía un **mensaje de petición DHCP** → si el servidor envió varias ofertas, el cliente solicita una.
4. El servidor envía un mensaje **ACK DHCP** → confirma la solicitud.

▸ El host pasará de tener la IP por defecto $(0.0.0.0)$ a la asignada por DCHP durante cierto periodo de tiempo o hasta que se apague.

NAT: TRADUCCIÓN DE DIRECCIONES DE RED

- El NAT es un sistema que permite usar la misma IP válida en varios ordenadores.

Tiene dos componentes fundamentales.

- Las **direcciones sin conexión a internet**, que son direcciones especiales para uso de redes privadas que sólo tienen sentido para los dispositivos de esa red.
  - Clase A → $10.0.0.0/8$ $(10.0.0.0 - 10.255.255.255)$.
  - Clase B → $172.16.0.0/12$ $(127.16.0.0 - 172.31.255.255)$.
  - Clase C → $192.168.0.0/160$ $(192.168.0.0 - 192.168.255.255)$.
  - Cada una de estas direcciones está asignada a muchísimos dispositivos en todo el mundo
  - Los routers de **internet ignorarán** los paquetes con estas direcciones.
- El **servidor NAT**, que conectará todos los dispositivos de la red interna con el exterior.
  - Tiene dos interfaces con dos IPs distintas {una con la IP válida para conectarse con el exterior / una con la IP privada para conectarse con la red interna.
  1. Los dispositivos de la red interna usarán como gateway la IP privada del servidor NAT.
  2. El servidor NAT encaminará los paquetes que le pasen cambiando la IP y puerto origen por su IP válida y un puerto que esté libre.
  3. Tras cada redirección, tendrá que almacenar una entrada en su tabla de traducciones NAT para poder saber a qué maquina de la red privada debe enviar la respuesta.
- Puede combinarse con el **filtrado de paquetes**.