

Tema 4 – Nivel de Red

Redes de Computadores

Grado en Ingeniería Informática en Tecnologías de
la Información

ÍNDICE

1. Introducción
2. Protocolo IP
3. Redes de datagramas y circuitos virtuales
4. Algoritmos de encaminamiento
5. Control de congestión
6. Calidad de servicio (QoS)

1.- INTRODUCCIÓN

- Encaminamiento de paquetes desde el origen hasta el destino
 - Proporciona una interfaz de red al nivel de transporte
 - No se encarga únicamente de la conexión extremo a extremo
 - Gestiona los pasos a través de los diferentes **nodos intermedios**
 - Una de sus funciones es **buscar** la **ruta** más apropiada
 - Utiliza tablas de encaminamiento
 - Almacena información sobre los destinos conocidos
 - Indica cuál es el siguiente nodo al que enviar la información para seguir con la ruta

Introducción

- Control de la congestión
 - Evitar la sobrecarga de las líneas
 - Tarea compartida con el nivel de transporte
 - Control a lo largo de la red, no solo entre extremos
- Calidad de servicio (QoS)
 - Ancho de banda
 - Retardo y variación del retardo
 - Pérdida

2.- EL PROTOCOLO IP

- **Características**

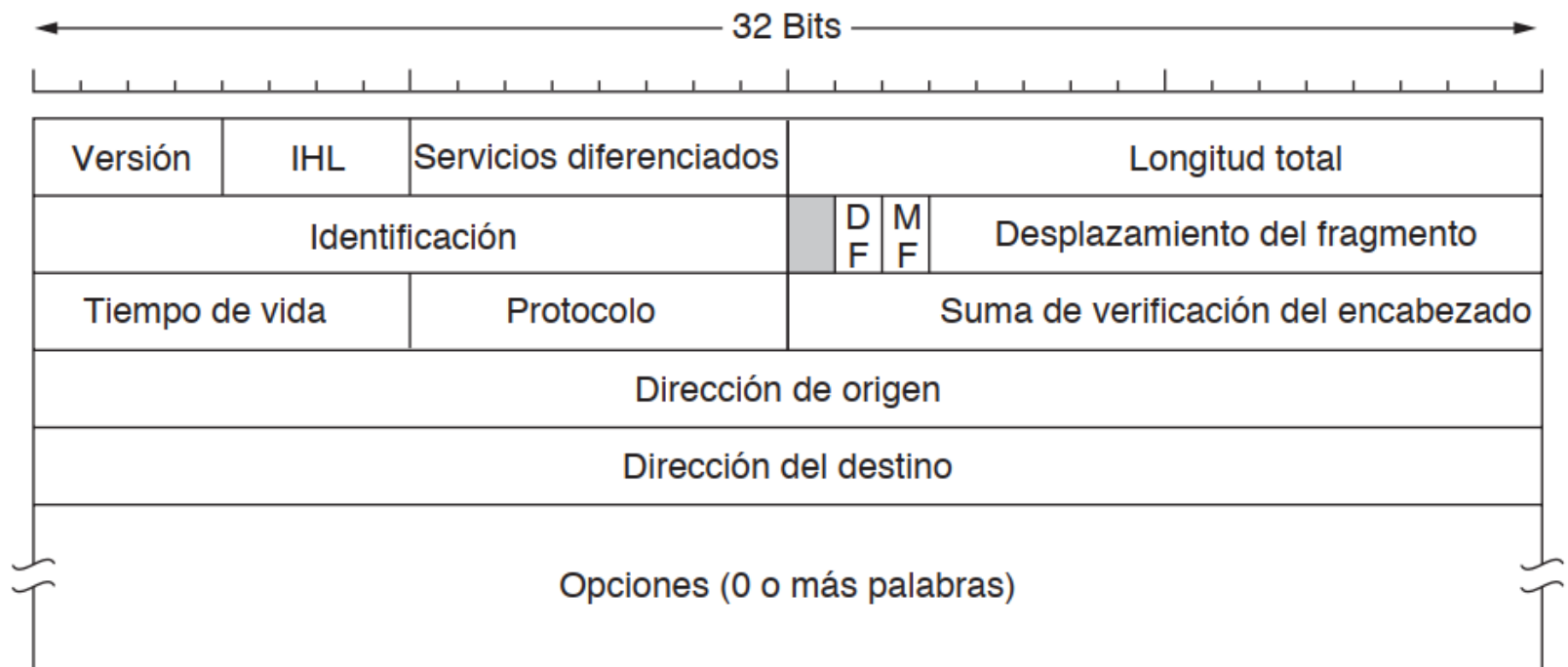
- **No orientado a conexión**: cada datagrama se trata de forma independiente, pudiendo seguir caminos distintos
- **No fiable**: es decir, cuando se produzca algún error los datagramas se perderán
- **Inseguro**: la entrega de los datagramas no está garantizada: se pueden perder, duplicar, retasar o entregarse fuera de orden
- No incluye técnicas para la gestión de errores
- La unidad básica de información se denomina **datagrama IP**

El protocolo IP

- **Funciones más importantes**
 - **Direccionamiento**: identificar a las máquinas mediante direcciones lógicas -> direcciones IP
 - **Encaminamiento**: analizar los datagramas para encaminarlos por los nodos intermedios desde el origen hasta el destino
 - **Fragmentación y reensamblado**: partición de los datagramas demasiado grandes para poder retransmitirlos por la red

El protocolo IP

- Cabecera IP



[1]

El protocolo IP

- **Cabecera IP**

- **Versión**: indica la versión del protocolo a la que pertenece el datagrama
- **Internet Header Length (IHL)**: tamaño total de la cabecera en palabras de 8 *bytes*. El tamaño mínimo es 5 y el máximo 15
- **Servicios diferenciados**: sirve para distinguir entre distintos tipos de servicios. Los 6 primeros bits indican la clase de servicio y los 2 restantes llevan información sobre congestión

El protocolo IP

- **Cabecera IP**

- **Longitud total**: tamaño total del datagrama incluida la cabecera. El máximo son 2^{16} bits
- **Identificación**: identifica el fragmento de un paquete recién llegado. Todos los fragmentos de un paquete tienen el mismo identificador
- **Don't Fragment (DF)**: bit utilizado para comunicar al enrutador que no fragmente el paquete. Se puede utilizar para descubrir el tamaño máximo posible en una ruta concreta
- **More Fragments (MF)**: indica que van a llegar más fragmentos de un mismo datagrama. Salvo el último, todos los fragmentos tienen este bit a 1

El protocolo IP

- **Cabecera IP**

- **Desplazamiento del fragmento**: indica a qué parte del paquete pertenece el fragmento. Los fragmentos excepto el último deben ser múltiplos de 8 bytes
- **Tiempo de vida (TtL)**: contador para limitar el tiempo de vida de un paquete
- **Protocolo**: código que indica cuál es el protocolo de la capa superior al que pasar el paquete
- **Suma de verificación**: asegura que la cabecera no ha llegado corrompida por el camino. Es necesario recalcularla en cada salto

El protocolo IP

- **Cabecera IP**
 - **Dirección de origen y destino:** contienen las direcciones IP de las interfaces de red de origen y destino
 - **Opciones:** permite añadir nuevas características no incluidas en la cabecera original
 - Seguridad: evitar enrutar paquetes a través de ciertas redes
 - Enrutamiento estricto: indicar todos los saltos desde el origen al destino
 - Registrar ruta: almacenar las IPs de todos los saltos realizados

El protocolo IP

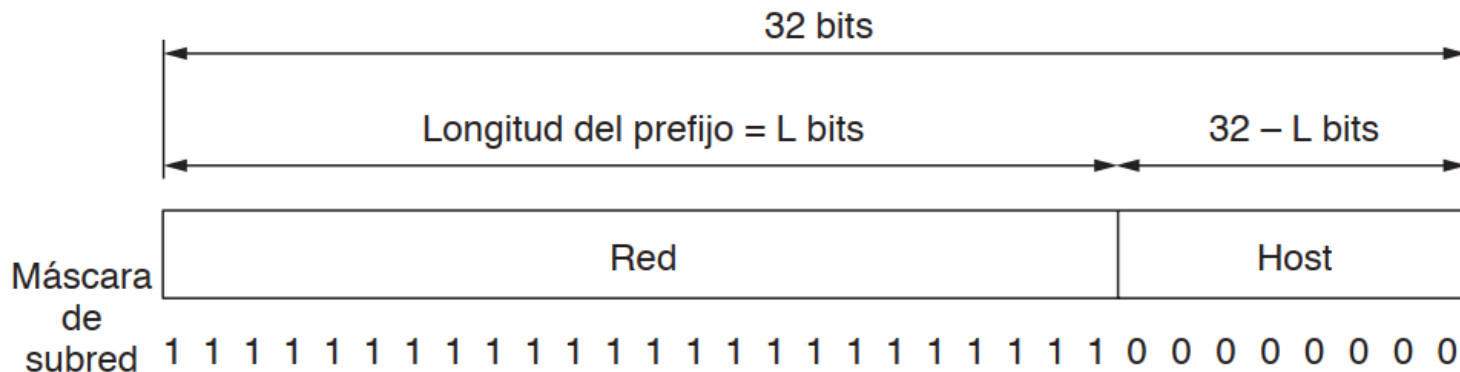
- **Direccionamiento IP**

- Todas las máquinas conectadas a internet poseen una dirección IP de 32 bits que actúa como identificador
- Los 32 bits se dividen en 4 grupos de 8 bits separados por puntos
- Cada grupo se representa con notación decimal
 - 156.35.14.2

El protocolo IP

- **Direccionamiento IP**

- Toda dirección está compuesta de dos partes: una parte identifica la red y otra identifica al host
- Para saber qué parte de la dirección corresponde a la red, utilizaremos una **máscara de red**
- La máscara se indica después de la dirección IP con un número entero que corresponde al número de 1s de la máscara
 - 156.35.14.2/16



[1]

El protocolo IP

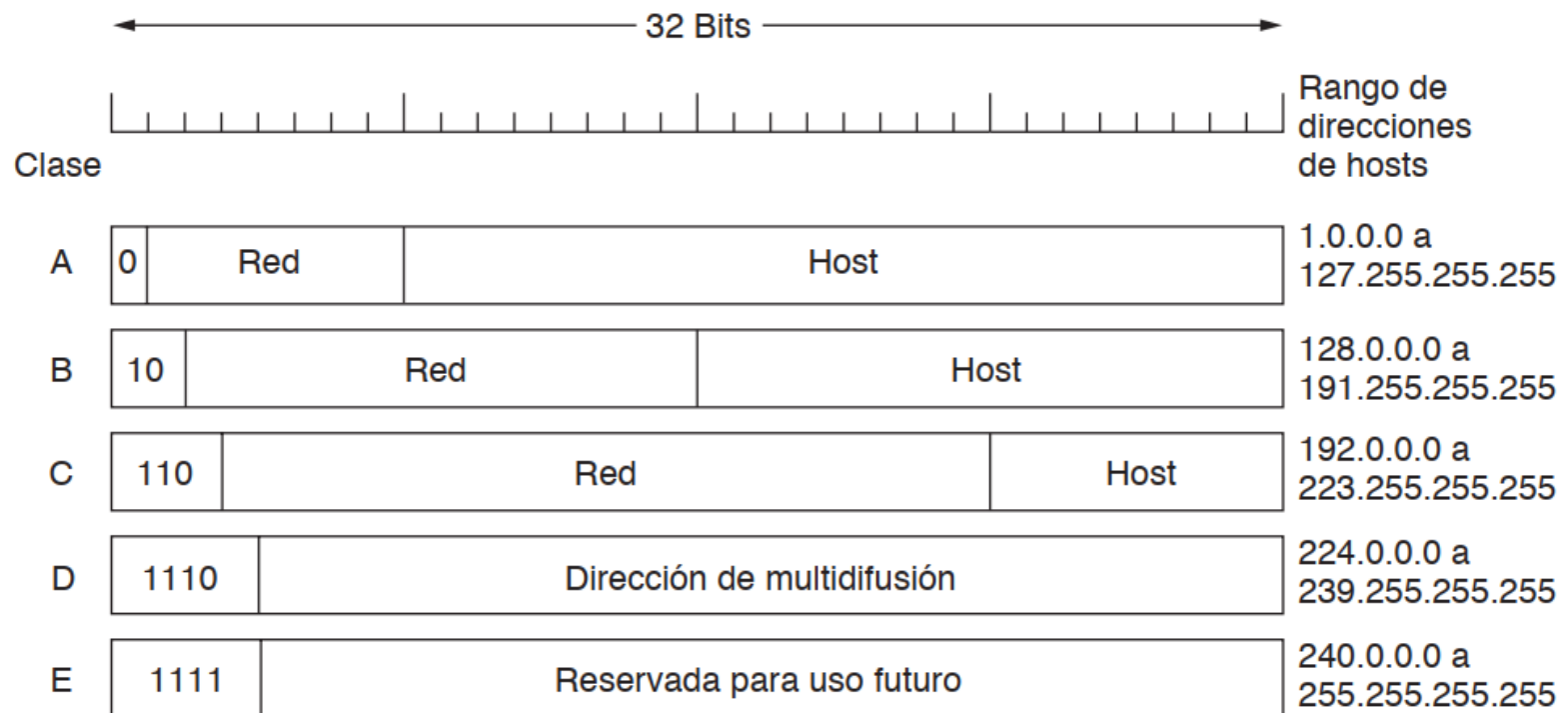
- **Direccionamiento IP**

- Tipos de direcciones

- **Unicast** (unidifusión): dirección que identifica a un interfaz de red de una máquina
 - **Multicast** (multidifusión): dirección que identifica a un grupo de interfaces de red en distintas máquinas
 - Nunca aparecerá como dirección origen
 - **Broadcast** (difusión): dirección que identifica a todas las interfaces de una determinada red
 - Nunca aparecerá como dirección origen
 - **Públicas**: direcciones visibles en Internet
 - **Privadas**: direcciones que no son visibles en Internet
 - **Estáticas**: direcciones que no cambian en cada conexión
 - **Dinámicas**: direcciones que pueden cambiar en cada conexión

El protocolo IP

- Direccionamiento IP



[1]

El protocolo IP

- **Direccionamiento IP**

- Clase A – Máscara 255.0.0.0

- Direcciones privadas: 10.0.0.0 a 10.255.255.255
 - Direcciones privadas: 127.0.0.0 a 127.255.255.25

- Clase B - Máscara 255.255.0.0

- Direcciones privadas: 172.16.0.0 a 172.31.255.255

- Clase C – Máscara 255.255.255.0

- Direcciones privadas: 192.168.0.0 a 192.168.255.255

El protocolo IP

- **Direccionamiento IP**

- Dirección base de red: dirección que identifica a la red
 - Tiene todos los bits de la parte de host a 0
 - Se obtiene realizando un AND bit a bit entre la dirección IP y de la máscara de subred
- Dirección de broadcast
 - Tiene todos los bits de la parte de host a 1
 - Se obtiene realizando un OR bit a bit entre la dirección IP y el complemento a 1 de la máscara
- El resto de direcciones libres se pueden asignar a los diferentes hosts
- Direcciones libres para n bits $\rightarrow 2^n - 2$

El protocolo IP

- **Direccionamiento IP - Ejemplo**

- ID de red 156.35.0.0 -> Clase B

- Una sola red, con espacio para $2^{16} - 2$ hosts

- ¿Y si se quiere tener tres subredes?

- Se puede dividir la red anterior en subredes más pequeñas
 - Se utilizarán parte de los bits de host para identificar las nuevas subredes
 - El tamaño de la máscara puede ser variable y dependerá del tamaño que se le quiera dar a la red – *Variable Length Subnet Mask (VLSM)*

El protocolo IP

- **Direccionamiento IP - Ejemplo**

- ID de red 156.35.0.0 – Máscara 255.255.0.0

- ID red -> 10011100 00100011 00000000 00000000

- Mask -> 11111111 11111111 00000000 00000000

- Para crear k subredes se necesitan n bits, que cumplan la ecuación $2^n \geq k$ -> Se necesitan 2 bits

- Mask -> 11111111 11111111 11000000 00000000

El protocolo IP

- **Direccionamiento IP - Ejemplo**

- Habrá que hacer combinaciones con el 17º y 18º bit para generar los nuevos ID de cada subred

Mask -> 11111111 11111111 11000000 00000000

ID red 1 -> 10011100 00100011 00000000 00000000

ID red 2 -> 10011100 00100011 01000000 00000000

ID red 3 -> 10011100 00100011 10000000 00000000

- ¿Y la combinación 11?
- ¿Cuántos equipos entran en cada subred?
- ¿Sobrarán IPs?

El protocolo IP

- **Direccionamiento IP – Ejemplo 2**
 - ID de red 192.168.1.0/24
 - Tres subredes, una con 100 equipos y las otras dos con 40
 - Solución: Utilizar dos bits de la parte de *host* para crear tres nuevas subredes
 - ¿Cuántos equipos entran en cada subred?
 - 6 bits libres para direccionar *hosts*
 - $2^6 - 2 = 62$ direcciones posibles
 - No se puede dividir utilizando máscara de tamaño fijo
- > Hay que utilizar máscaras de tamaño variable (*VSLM*)

El protocolo IP

- **Direccionamiento IP – Ejemplo 2**

- ID de red 192.168.1.0/24
- Número n de bits que se necesita para direccionar los k *hosts* de cada subred:
 - $2^n - 2 \geq 100 \rightarrow n = 7$
 - $2^n - 2 \geq 40 \rightarrow n = 6$
- Cada una de las nuevas subredes tendrá una máscara que se adapte al número de bits que se necesitan para la parte de *host*
 - Subred 100 PCs \rightarrow Máscara de $32 - 7 = 25$ bits
 - Subredes de 40 PCs \rightarrow Máscara de $32 - 6 = 26$ bits

El protocolo IP

- **Direccionamiento IP – Ejemplo 2**

- ¿Cuáles serán los nuevos IDs de las subredes?
- Hay que dividir el ID original en redes más pequeñas:
 - Máscara 24 bits (original)
 - ID: 11000000 10101000 00000001 00000000
 - Máscara 25 bits
 - ID subred 1: 11000000 10101000 00000001 00000000
 - ID libres: 11000000 10101000 00000001 10000000
 - Máscara 26 bits
 - ID subred 2: 11000000 10101000 00000001 10000000
 - ID subred 3: 11000000 10101000 00000001 11000000

El protocolo IP

- **Direccionamiento IP**
 - Obtención de direcciones
 - Métodos
 - Configuración **manual**
 - Protocolo de autoconfiguración → **DHCP**
 - Parámetros necesarios para configurar una máquina
 - Dirección IP
 - Máscara de red
 - Dirección IP del router de salida
 - Dirección IP de servidores DNS

El protocolo IP

- **Direccionamiento IP**

- **Protocolo DHCP**

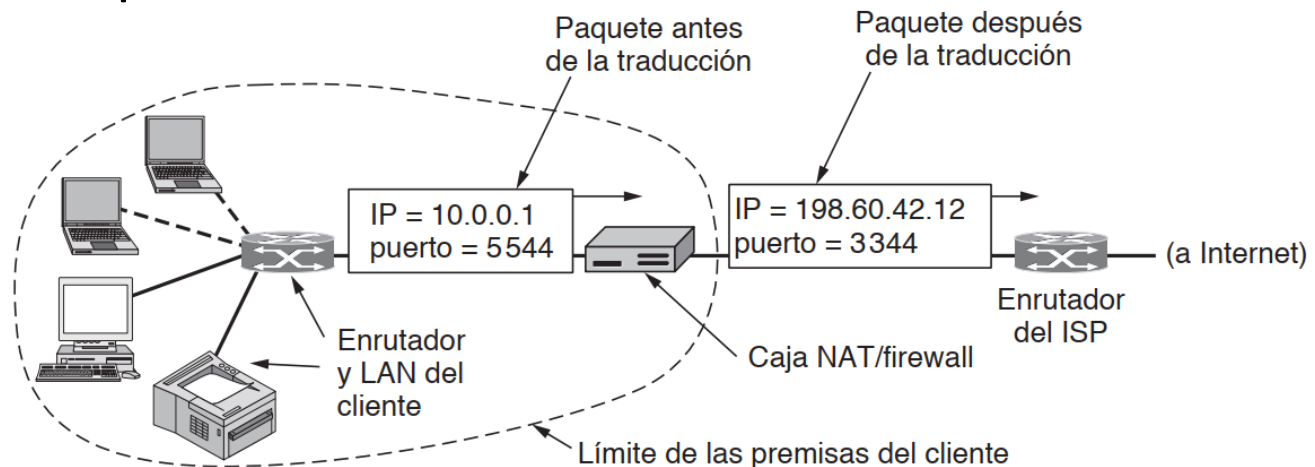
- Protocolo para obtener parámetros de **configuración automática** desde la red (por ejemplo, dirección IP)
 - Basado en el modelo **cliente-servidor**
 - El cliente contacta con un servidor DHCP para obtener sus parámetros
 - Asignación de dirección IP
 - Apropiaada a la red o subred a la que se conecta el cliente
 - No asignada a otra máquina

El protocolo IP

- **Direccionamiento IP**

- ***Network Address Translation (NAT)***

- Gestiona la posible escasez de direcciones IP dentro de una subred
 - Utiliza el campo de puerto de las cabeceras de nivel de transporte



[1]

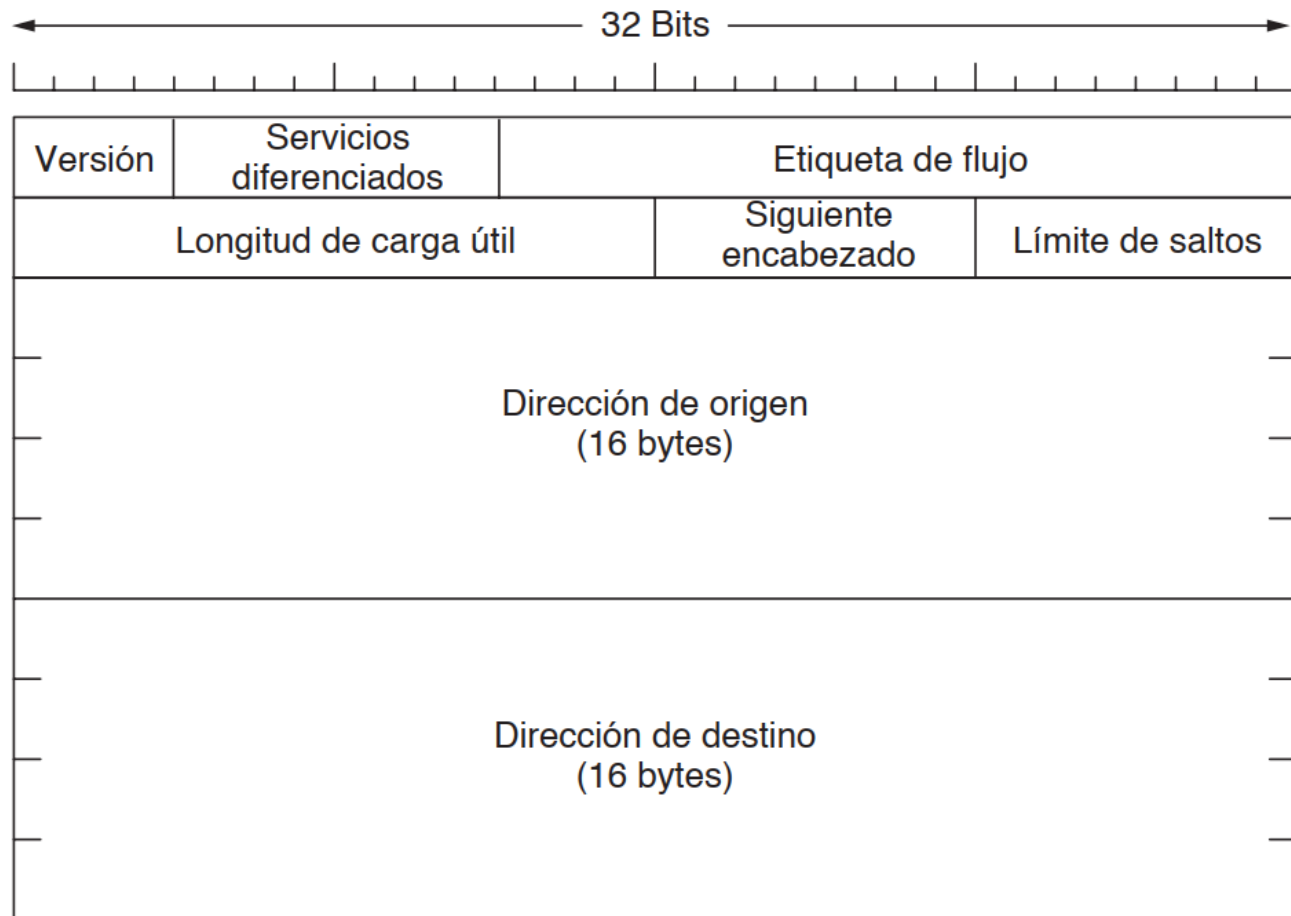
El protocolo IP

- **IPv6**

- Implementa mejoras sustanciales sobre IPv4
- Aumenta de forma notable la cantidad de direcciones existente en Internet 2^{32} vs 2^{128}
- Es compatible hacia atrás, pero IPv4 no es plenamente compatible con él
- No muy ampliamente extendido, pese a que fue desarrollado hace aproximadamente 20 años
- Utiliza direcciones agrupadas en 8 bytes y escritas en hexadecimal:
 - 8000:0000:0000:0000:0123:4567:89AB:CDEF

El protocolo IP

- IPv6



El protocolo IP

- **Segmentación y reensamblado**
 - Las redes individuales pueden especificar tamaños máximos de paquetes diferentes.
 - Debemos conocer: MTU (*Maximum Transfer Unit*) del nivel de enlace
 - Segmentar si el tamaño del datagrama es mayor que el MTU:
 - Se generan varios datagramas
 - El tamaño de los datagramas no es fijo
 - La segmentación puede producirse en el emisor o por culpa de un router intermedio
 - El nuevo datagrama contiene un offset que indica la porción de datos enviado en este paquete en relación al paquete original

El protocolo IP

- **Segmentación y reensamblado**
 - ¿Dónde reensamblar?
 - En el destino
 - Los fragmentos sólo se pueden hacer más pequeños
 - En los nodos intermedios
 - Uso de un gran espacio de almacenamiento
 - Todos los fragmentos de los datagramas deben pasar a través del mismo dispositivo de encaminamiento
 - En IP se reensambla en el destino

El protocolo IP

- **Segmentación y reensamblado**
 - Fallos en el reensamblado
 - Si se pierden paquetes, se descarta el reensamblaje
 - ¿Cuánto tiempo es necesario esperar para reensamblar?
 - Tiempo de vida para el reensamblado: asignar un tiempo al recibir el primer segmento. Si el tiempo expira sin completar el reensamblaje se descarta
 - Tiempo de vida del datagrama: si expira el tiempo de vida de un fragmento, se descarta el reensamblaje

3.- REDES DE DATAGRAMAS Y CIRCUITOS VIRTUALES

- Redes de datagramas
 - Proporcionan un servicio de red no orientado a conexión
 - Cada paquete puede seguir una ruta distinta a lo largo de la red
- Redes de circuitos virtuales
 - Proporcionan un servicio de red orientado a conexión
 - Los paquetes siguen el mismo circuito virtual a lo largo de toda la red

Redes de datagramas y circuitos virtuales

- **Redes de datagramas**
 - No se determina una ruta anticipadamente
 - Cada datagrama se **encamina independientemente** con lo que puede seguir rutas diferentes
 - Los nodos mantienen una **tabla de encaminamiento** con las redes conocidas para encaminar los paquetes en función de su red de destino
 - El datagrama deberá contener la **dirección completa** del destinatario y del origen

Redes de datagramas y circuitos virtuales

- Redes de datagramas

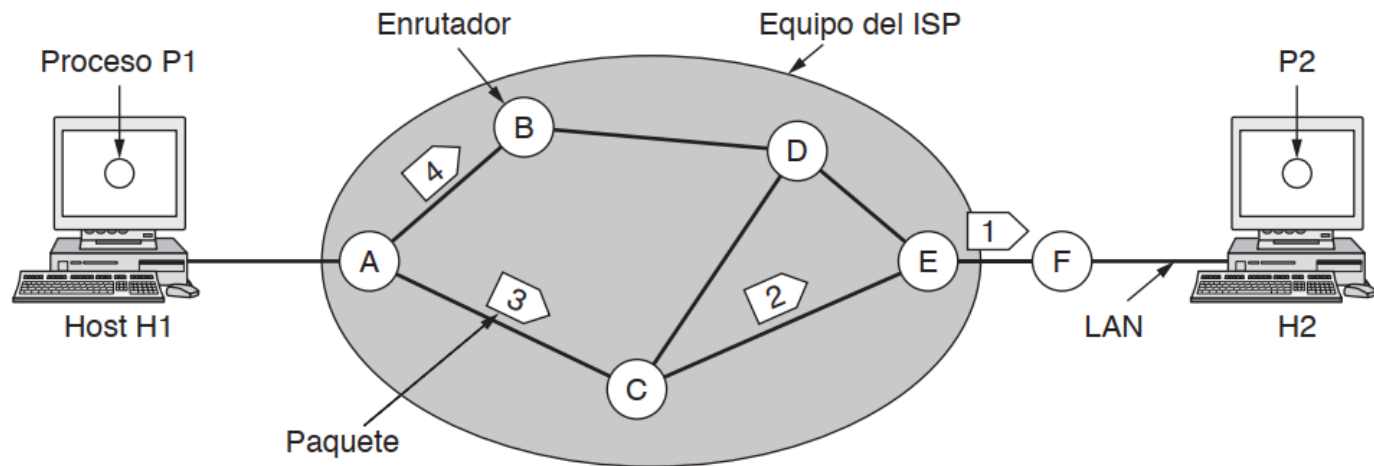


Tabla de A (al principio)

A	-
B	B
C	C
D	B
E	C
F	C

Destino Línea

Tabla de A (más tarde)

A	-
B	B
C	C
D	B
E	B
F	B

Tabla de C

A	A
B	A
C	-
D	E
E	E
F	E

Tabla de E

A	C
B	D
C	C
D	D
E	-
F	F

[1]

Redes de datagramas y circuitos virtuales

- **Redes de circuitos virtuales**
 - Establecen una **ruta predeterminada** al inicio de la conexión – Se almacenan los siguientes nodos en una tabla
 - Los nodos no necesitan calcular la ruta cada vez que llega un paquete, ya la tienen almacenada
 - Todos los paquetes circulan por la misma ruta
 - Una vez que se termina la conexión, se liberan los recursos

Redes de datagramas y circuitos virtuales

- Redes de circuitos virtuales

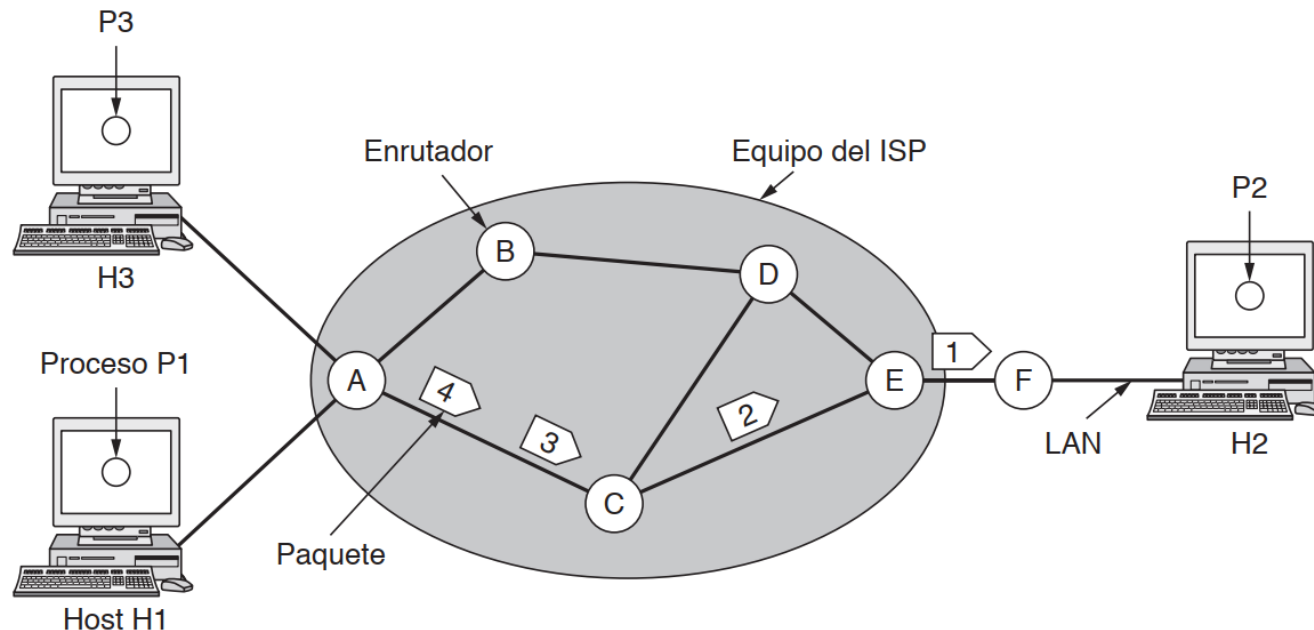


Tabla de A

H1	1	C	1
H3	1	C	2

Dentro Fuera

Tabla de C

A	1	E	1
A	2	E	2

Tabla de E

C	1	F	1
C	2	F	2

[1]

Redes de datagramas y circuitos virtuales

Asunto	Red de datagramas	Red de circuitos virtuales
Configuración del circuito	No necesaria	Requerida
Direccionamiento	Cada paquete contiene la dirección de origen y de destino completas	Cada paquete contiene un número de CV corto
Información de estado	Los enrutadores no contienen información de estado sobre las conexiones	Cada CV requiere espacio de tabla del enrutador por cada conexión
Enrutamiento	Cada paquete se enruta de manera independiente	La ruta se elige cuando se establece el CV; todos los paquetes siguen esa ruta
Efecto de fallas del enrutador	Ninguno, excepto para paquetes perdidos durante una caída	Terminan todos los CVs que pasaron por el enrutador defectuoso
Calidad del servicio	Difícil	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV
Control de congestión	Difícil	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV

Redes de datagramas y circuitos virtuales

- **Ventajas datagramas**

- No consume recursos en el establecimiento de la conexión
- Tiempo nulo de establecimiento de conexión
- Robustez ante caídas en la red
- Más difícil de interceptar los mensajes completos
- No necesita almacenar la información sobre múltiples circuitos virtuales

Redes de datagramas y circuitos virtuales

- **Ventajas circuitos virtuales**
 - No se genera sobrecarga por el cálculo de las rutas en cada nodo
 - Paquetes más ligeros al no utilizar direcciones completas
 - Mayor facilidad para proporcionar un servicio fiable y con buena calidad