La capa de red
El protocolo IP

Programación y administración de redes - Semana 7

Grado en *Ingeniería Informática*

Departamento de Informática. Universidad de Jaén

Objetivos

General

Conocer el funcionamiento del protocolo IP en términos generales, las funciones que ofrece y el formato de los datagramas de IPv4 e IPv6

Específicos

- Conocer la estructura de los datagramas de IPv4 e IPv6
- Identificar la finalidad de los campos de la cabecera de los datagramas
- Reconocer los tipos de paquete que pueden incluirse en un datagrama
- Saber cómo se controla el tiempo de vida de los datagramas
- Conocer el mecanismo de fragmentación de IP

Introducción

El protocolo IP

Es el principal protocolo de la capa de red y tiene la responsabilidad de facilitar la comunicación entre host no directamente conectados, reenviando los datagramas a través de los dispositivos de conmutación

Cómo funciona

1. En el ***host* de origen** el segmento TCP/UDP se aloja en un datagrama
2. IP se encarga de configurar los **campos de la cabecera** para iniciar la transmisión del datagrama
3. Los **elementos de conmutación**, casi siempre *router*, examinan la cabecera para reenviar el datagrama por la mejor ruta
4. Los *router* también **actualizan** algunos campos del datagrama
5. La capa de red en el ***host* de destino** extrae del datagrama el segmento y lo entrega a la capa de transporte

Tareas a cargo del protocolo IP

Responsabilidad

Para afrontar el reenvío de los datagramas hasta su destino, que es la principal responsabilidad de IP, este tiene que abordar múltiples tareas

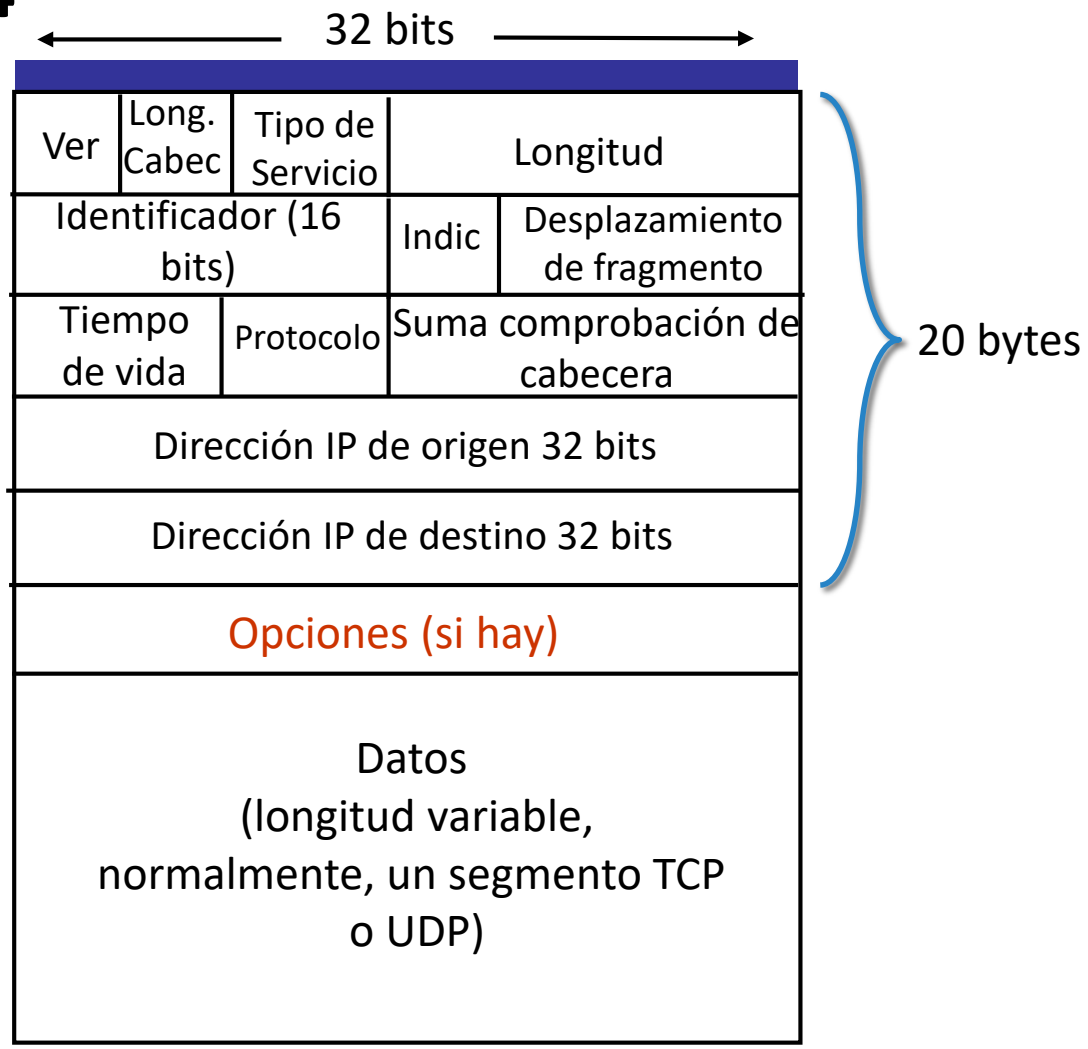
Detalles

- Determinar qué **protocolo de la capa de transporte** envía los datos para entregar el segmento a TCP o UDP en el *host* de destino
- Evitar que un datagrama pueda estar eternamente circulando por Internet si entra en **un bucle entre múltiples equipos** de conmutación
- En caso necesario, **fragmentar el datagrama** para facilitar su transferencia a través de infraestructuras de red con una MTU de menor tamaño
- **Detectar posibles errores** en la transmisión de los datagramas
- Atender otras **peticiones específicas** que el emisor haya solicitado

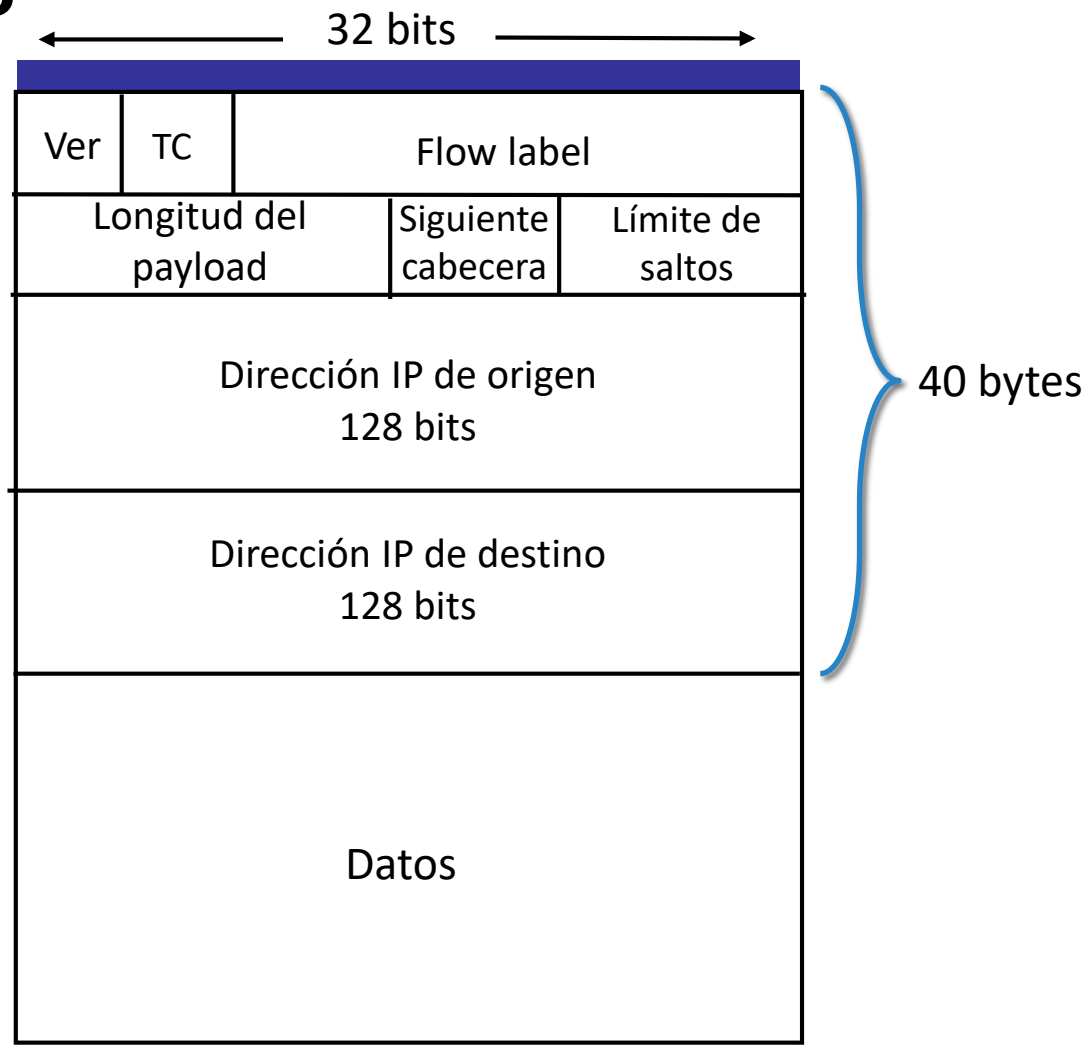
Formato del datagrama

Datagrama de IPv4 frente al de IPv6

IPv4



IPv6



Datagrama de IPv4 frente al de IPv6

Detalles

- IPv4 cuenta con una cabecera de **longitud variable** según que se incluyan o no opciones, de ahí la necesidad del campo que indica la longitud de la cabecera
- El datagrama de **IPv6 es más simple** que el de IPv4, con menos campos y sin la posibilidad de opciones, por lo que la longitud de la cabecera es fija
- La cabecera de **IPv6 tiene mayor tamaño** que la de IPv4 porque las direcciones IP de origen y destino son de 128 bits en lugar de 32 bits
- Los primeros 4 bits de ambas cabeceras identifican la **versión de IP**, que será 4 o 6, determinando así qué protocolo ha de procesar el datagrama al recibirlo

Actividad – Cabeceras IP y estructuras de datos

Estructuras de datos asociadas a las cabeceras IP

- Ejecuta en tu máquina virtual el comando `sudo apt install gcc` a fin de instalar los archivos de cabecera de C/C++ necesarios para desarrollar aplicaciones Linux
- En el directorio `/usr/include/netinet` encontrarás los archivos `ip.h` e `ip6.h`
- Examina dichos archivos (los puedes abrir con el editor nano) y responde a estas cuestiones:
  - ¿Qué tipo de dato se usa para el campo `ttl` de la cabecera de IPv4?
  - ¿Cuál es el tipo de datos para las direcciones IP de origen y destino? ¿Qué diferencia hay entre IPv4 e IPV6?
  - Examina las constantes que se definen a lo largo del archivo de cabecera `ip.h` para saber cuál es el valor por defecto que se asigna al campo TTL. ¿Cuál es el nombre de la constante que almacena dicho valor?
  - ¿Cuál es el MSS por defecto para IPv4 según el archivo de cabecera `ip.h`?

```
usuario@par:~$
usuario@par:~$ sudo apt install gcc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  dns-root-data dnsmasq-base libbluetooth3 libdbusmenu-glib4 libdbusmenu-gtk3-4 libllvm11 libndp0
  libnm0 libteamdctl0 net-tools ppp pptp-linux squashfs-tools xul-ext-ubufox
Utilice «sudo apt autoremove» para eliminarlos.
Paquetes sugeridos:
  gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc
Se instalarán los siguientes paquetes NUEVOS:
  gcc
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 5.212 B de archivos.
Se utilizarán 51,2 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu hirsute/main amd64 gcc amd64 4:10.3.0-1ubuntu1 [5.212 B]
Descargados 5.212 B en 0s (46,9 kB/s)
Seleccionando el paquete gcc previamente no seleccionado.
(Leyendo la base de datos ... 257323 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../gcc_4%3a10.3.0-1ubuntu1_amd64.deb ...
Desempaquetando gcc (4:10.3.0-1ubuntu1) ...
Configurando gcc (4:10.3.0-1ubuntu1) ...
Procesando disparadores para man-db (2.9.4-2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.
usuario@par:~$
usuario@par:~$ nano /usr/include/netinet/ip.h
```

Funcionalidad común

Resumen

IPv4 e IPv6 tienen el mismo objetivo, por lo que comparten varias de las funcionalidades que se esperan del servicio de red

Funcionalidades

- Reenvían el datagrama hacia su destino, para lo cual precisan una **dirección de destino y también una de origen** (para la respuesta). Lo único que cambia es el tamaño de las direcciones, según estudiamos la pasada semana
- Ambos cuentan con un campo que permite conocer la **longitud del *payload*** o paquete de datos a transferir
- Con el campo **Protocolo / Next hop header** (8 bits) se indica el protocolo al que pertenece el paquete de datos enviado. Códigos habituales son 6 para TCP o 17 para UDP
- El campo **Tipo de servicio** / **TC (Traffic Class)** ocupa 8 bits que denotan la solicitud de un servicio que se ajusta a unas características concretas
- Con el campo **Tiempo de vida / Hop limit** (8 bits) se determina el número máximo de saltos antes de que se descarte el datagrama si no llega a su destino

Configuración de cabecera IP - Origen

Inicialización

El host de origen es el encargado de inicializar todos los campos de la cabecera del datagrama IP

- Se asigna la versión al campo **Ver** (4 bits): 4 o 6
- Según el servicio de transporte que haya enviado el segmento, se da al campo **Protocolo** (8 bits) el valor que corresponda
- Se introduce el *payload* y se asigna su **longitud** al campo correspondiente (8 bits)
- Se almacenan las **direcciones IP** de origen y de destino
- Se da un valor inicial al campo **Tiempo de vida** (8 bits)
- En IPv4 se dan valores por defecto a los campos **Identificador**, **Indicadores** y **Desplazamiento** si fuese preciso fragmentar el datagrama
- En IPv4 se calcula la longitud de la cabecera y se asigna a **Long. Cabec.** (4 bits) en función de que haya o no opciones adicionales

Configuración de cabecera IP - Router

Actualización

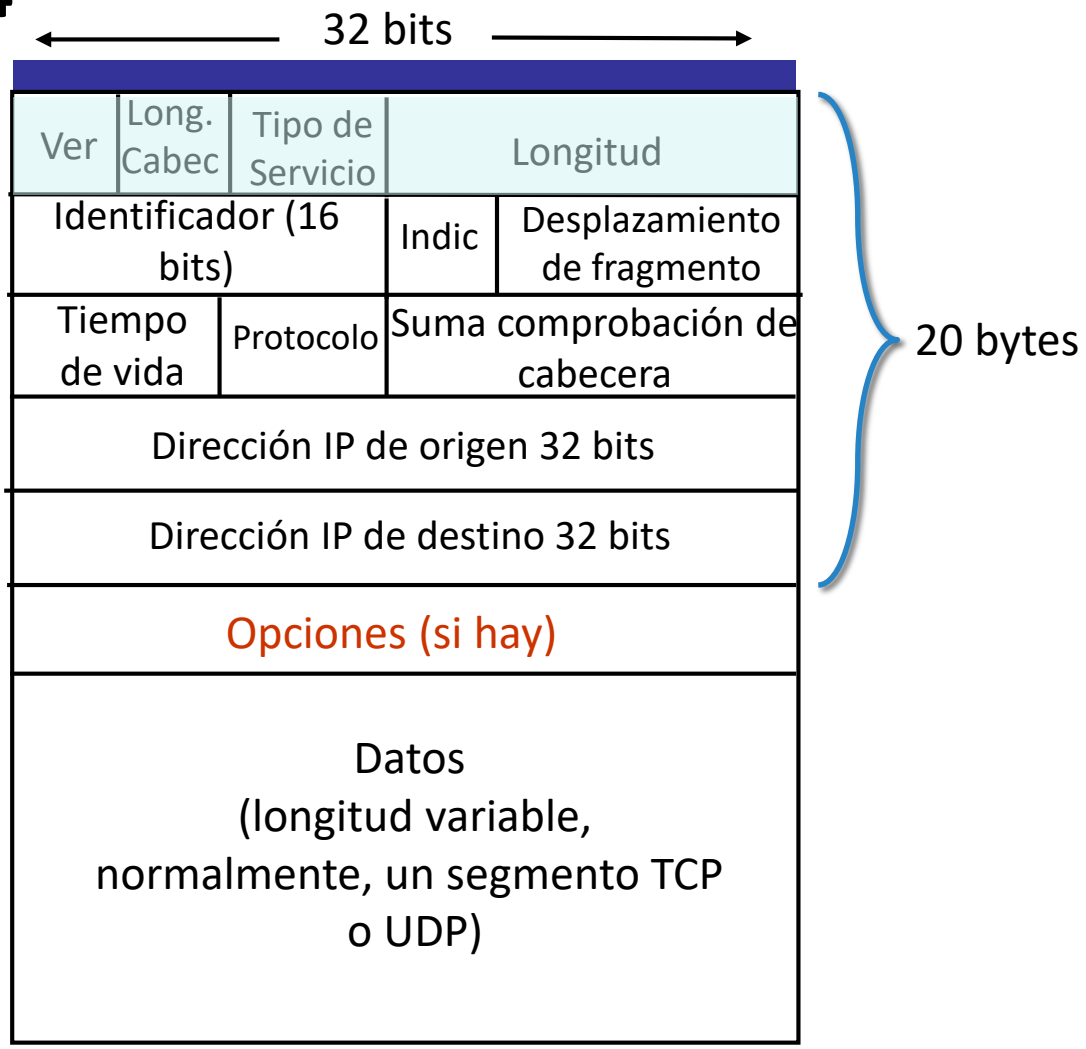
La capa de red de los equipos de conmutación obtienen el datagrama desde la capa de enlace, examinan la dirección de destino para reenviarlo por la ruta adecuada e introducen cambios en algunos campos de la cabecera

- Se toma el campo **Tiempo de vida** y se reduce en una unidad
- Si el campo **Tiempo de vida** es cero, el datagrama se descarta en lugar de ser reenviado
- Si el datagrama se descarta se envía un paquete de **notificación de error** al origen

Formato del datagrama
Detalles de IPv4

Detalles del datagrama de IPv4

IPv4



- **Versión** (4 bits): número de versión del protocolo IP del datagrama, 4 para IPv4 y 6 para IPv6
- **Longitud de cabecera** (4 bits): necesario ya que el número de opciones es variable. Cuando no hay opciones este valor es 5 (20 bytes)
- **Tipo de servicio** (8 bits): calidad de servicio de transmisión, poco usado. Composición:

0	1	2	3	4	5	6	7
Prioridad			D	T	R	C	N.U.

  - Los tres primeros indican prioridad del datagrama.
  - Bit 3 act. (*D-Delay*): Mínimo retardo.
  - Bit 4 act. (*T-Throughput*):Máximo rendimiento
  - Bit 5 act. (*R-Reliability*):Máxima fiabilidad
  - Bit 6 act. (*C-Monetary Cost*): Mínimo coste eco.

- **Longitud** (16 bits): longitud total de todo el datagrama en bytes. Típica entre 500 y 1460 bytes

Actividad - Tipos de servicio

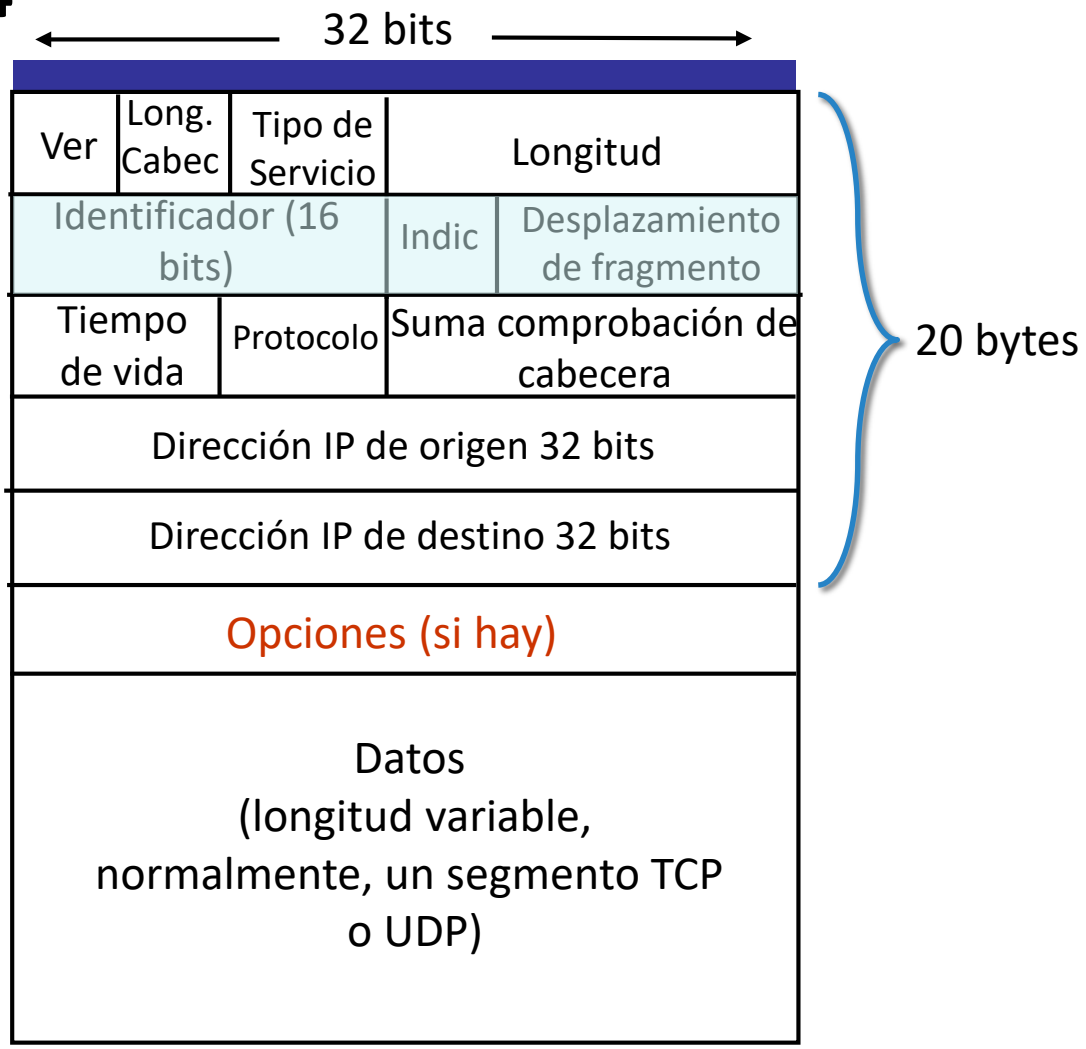
Indicadores TOS

- Mediante la utilidad hping3, que usábamos en actividades de semanas previas, es posible configurar el TOS (*Type-Of-Service*) solicitado. Lo habitual es activar únicamente un bit del campo TOS
- Consulta la ayuda de hping3 y, en particular, de la opción --tos, para hacer pruebas con distintos valores y responder lo siguiente:
  - ¿Qué representa cada uno de los valores de --tos?
  - ¿Hay diferencias en el tiempo de respuesta según el valor asignado a TOS?
  - ¿Cambia el campo TTL en la respuesta según el valor asignado a TOS?
  - En general, ¿crees que tiene alguna incidencia en la comunicación cambiar los bits de TOS?

```
[usuario@servidor ~]$ hping --tos help
tos help:
        TOS Name                Hex Value          Typical Uses

      Minimum Delay               10               ftp, telnet
      Maximum Throughput          08               ftp-data
      Maximum Reliability         04               snmp
      Minimum Cost                02               nntp
[usuario@servidor ~]$ sudo hping3 -c 5 -S -p 80 --tos 10 google.es
HPING google.es (enp0s3 172.217.168.163): S set, 40 headers + 0 data bytes
len=46 ip=172.217.168.163 ttl=59 id=62881 sport=80 flags=SA seq=0 win=65535 rtt=25.1 ms
len=46 ip=172.217.168.163 ttl=59 id=22832 sport=80 flags=SA seq=1 win=65535 rtt=30.0 ms
len=46 ip=172.217.168.163 ttl=60 id=27571 sport=80 flags=SA seq=2 win=65535 rtt=24.8 ms
len=46 ip=172.217.168.163 ttl=59 id=50676 sport=80 flags=SA seq=3 win=65535 rtt=24.8 ms
len=46 ip=172.217.168.163 ttl=60 id=5601 sport=80 flags=SA seq=4 win=65535 rtt=27.7 ms

--- google.es hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 24.8/26.5/30.0 ms
[usuario@servidor ~]$ sudo hping3 -c 5 -S -p 80 --tos 8 google.es
HPING google.es (enp0s3 172.217.168.163): S set, 40 headers + 0 data bytes
len=46 ip=172.217.168.163 ttl=59 id=26436 sport=80 flags=SA seq=0 win=65535 rtt=29.1 ms
len=46 ip=172.217.168.163 ttl=60 id=31315 sport=80 flags=SA seq=1 win=65535 rtt=28.7 ms
len=46 ip=172.217.168.163 ttl=60 id=21627 sport=80 flags=SA seq=2 win=65535 rtt=29.6 ms
len=46 ip=172.217.168.163 ttl=59 id=34774 sport=80 flags=SA seq=3 win=65535 rtt=30.3 ms
len=46 ip=172.217.168.163 ttl=60 id=55136 sport=80 flags=SA seq=4 win=65535 rtt=25.1 ms

--- google.es hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 25.1/28.5/30.3 ms
[usuario@servidor ~]$
```

Detalles del datagrama de IPv4

IPv4



- **Identificador** (16 bits): identificador único asignado a cada datagrama enviado en transmisión. Si se fragmenta, todos los paquetes mantienen la identificación
- **Indicadores** (3 bits):
  - Bit 1: no se usa
  - Bit 2: (*Don't fragment*) indica que no se fragmente el datagrama
  - Bit 3: (*More fragments*) indica si hay (1) más fragmentos o no (0)
- **Desplazamiento de fragmento** (13 bits):
  - Se usa para indicar a partir de qué número de byte del datagrama original hay que introducir los bytes que vienen con este fragmento

Actividad - Fragmentación del datagrama IPv4

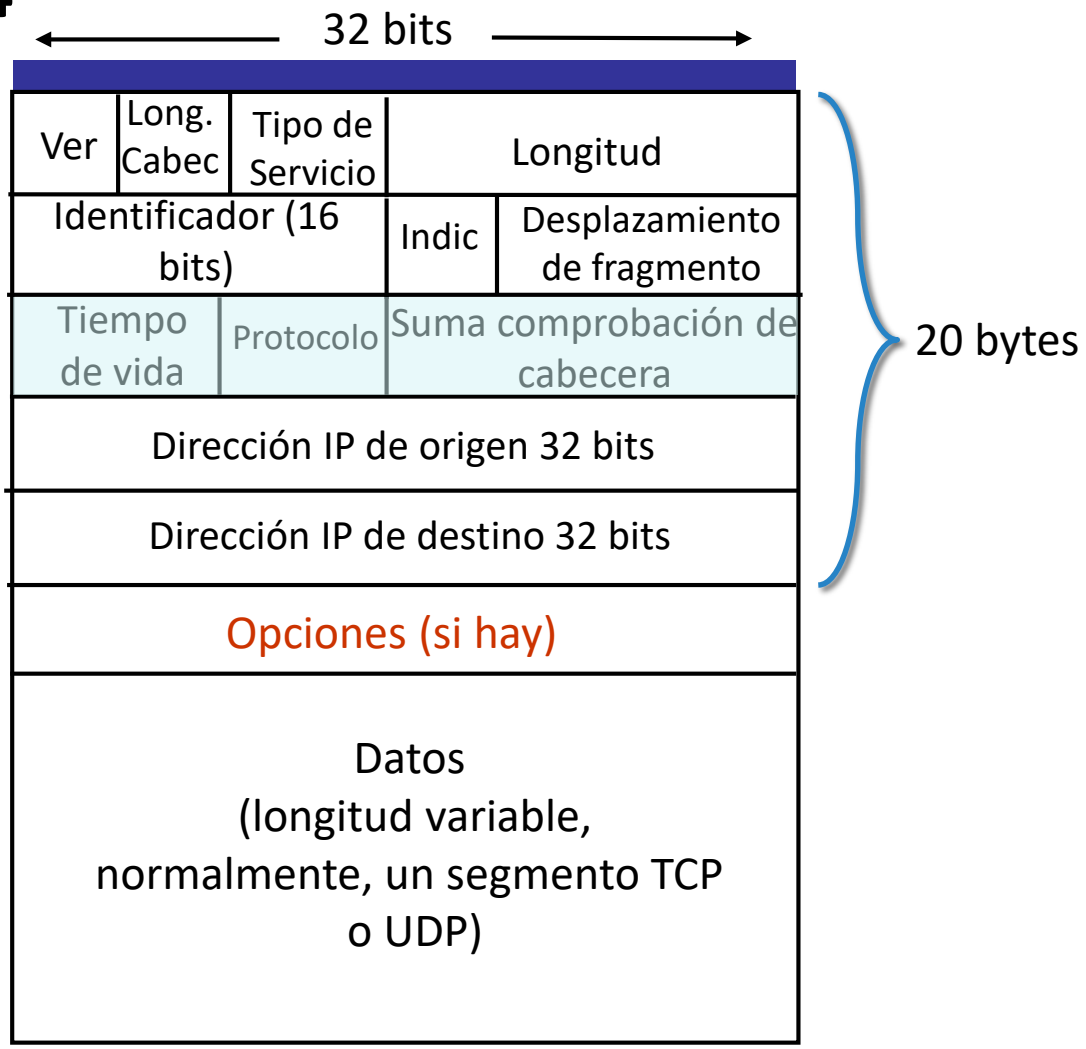
Fragmentación y MTU

- El comando `ping` nos permite enviar un paquete a otro *host* y solicitar eco, de forma que se obtenga como respuesta un paquete con el mismo contenido
- Esta utilidad cuenta con múltiples opciones que puedes consultar con `man ping`, entre ellas:
  - `-M do`: impide que se fragmenten los datagramas
  - `-s tam`: fija la longitud del paquete de datos
  - `-c veces`: indica el número de repeticiones
- Usando este comando prueba en tu equipo a enviar paquetes de diferentes tamaños, siempre deshabilitando la fragmentación, descubriendo así cuál es la longitud máxima del paquete de datos que puedes enviar
- Cuáles son el MSS y MTU según esas pruebas

```
usuario@par:~$
usuario@par:~$ ping -M do -s 1460 -c 3 ujaen.es
PING ujaen.es (150.214.170.66) 1460(1488) bytes of data.
1468 bytes from wwwr.ujaen.es (150.214.170.66): icmp_seq=1 ttl=61 time=1.72 ms
1468 bytes from wwwr.ujaen.es (150.214.170.66): icmp_seq=2 ttl=61 time=1.61 ms
1468 bytes from wwwr.ujaen.es (150.214.170.66): icmp_seq=3 ttl=61 time=1.74 ms

--- ujaen.es ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.613/1.689/1.736/0.054 ms
usuario@par:~$
usuario@par:~$ ping -M do -s 1480 -c 3 ujaen.es
PING ujaen.es (150.214.170.66) 1480(1508) bytes of data.
ping: local error: message too long, mtu=1500
ping: local error: message too long, mtu=1500
ping: local error: message too long, mtu=1500

--- ujaen.es ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2030ms

usuario@par:~$ _
```

Detalles del datagrama de IPv4

IPv4



- **Tiempo de vida** (8 bits): (TTL, *Time To Live*), controla que un datagrama no quede eternamente circulando por la red. El emisor le asigna un valor X que va reduciéndose en 1 por cada router por el que se pasa hacia el destino

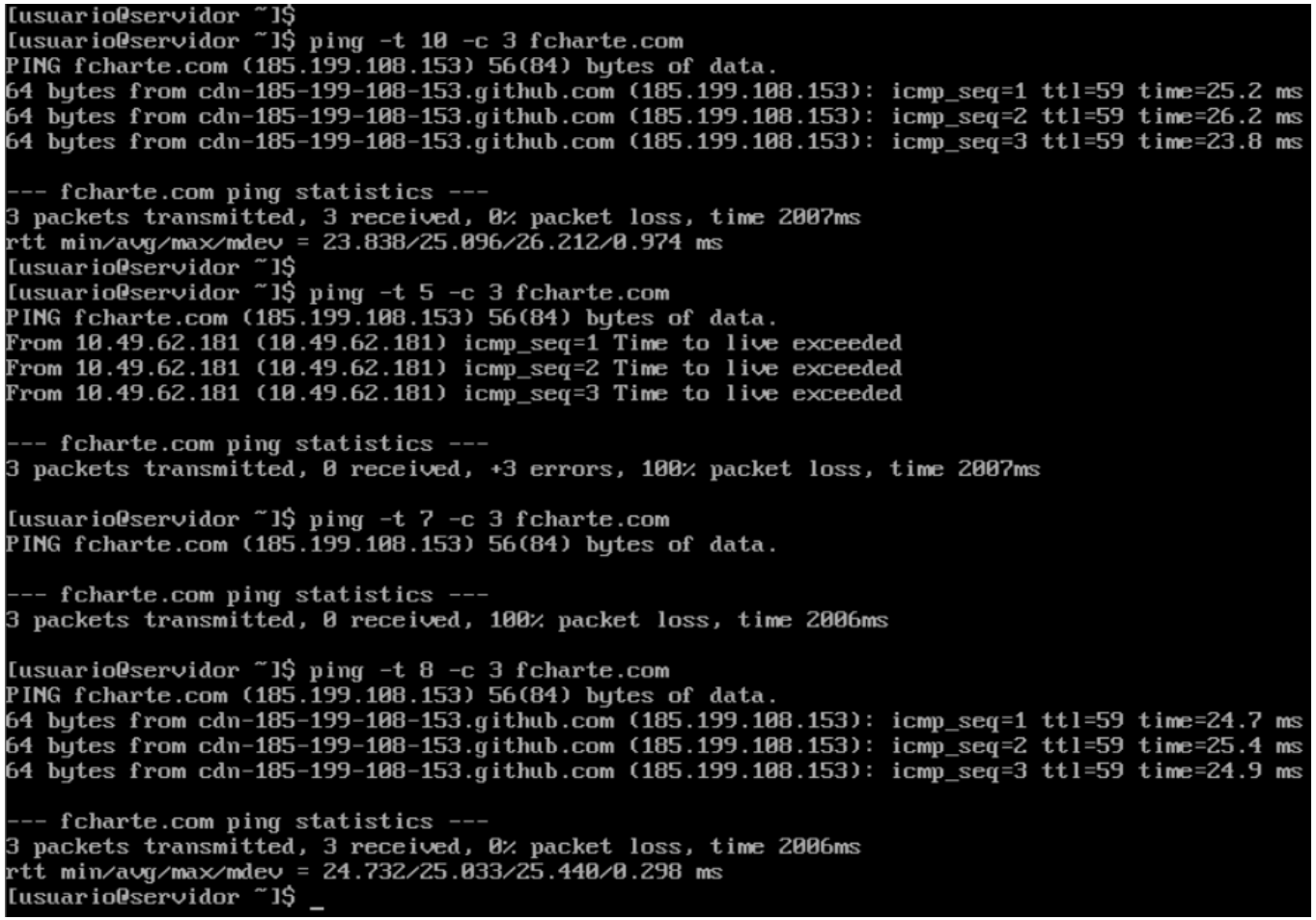
- **Protocolo** (8 bits): utilizado en los equipos extremos de la transmisión. Indica el protocolo de la capa de transporte al que va destinado el datagrama. Por e.: TCP: 6, UDP: 17

- **Suma de comprobación de cabecera** (16 bits): detección de errores de cabecera, se calcula tomando los bits de la cabecera en palabras de 16 bits y sumándolas con aritmética en complemento a 1. Normalmente se desechan los datagramas erróneos.

Actividad - Comprobación del TTL

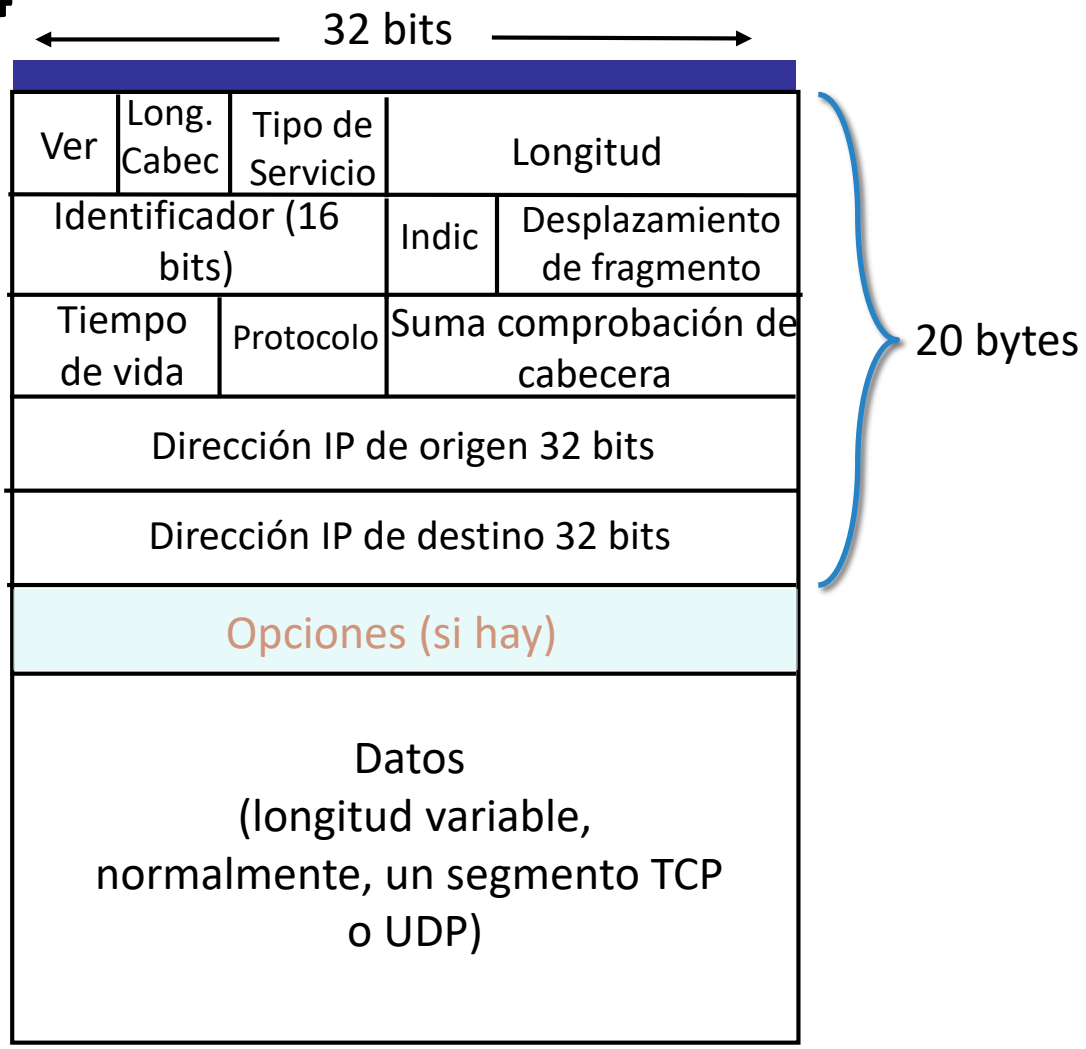
Número de saltos

- El mismo comando `ping` también permite, mediante la opción `-t`, fijar el valor inicial del campo TTL
- Usa esa opción con distintos valores y contesta las siguientes cuestiones:
  - ¿Cuál es el mínimo valor de TTL para obtener respuesta de `google.com`?
  - Cuando el valor inicial dado a TTL es inferior al anterior, ¿quién devuelve el paquete con el que se notifica el error?
  - Asumiendo que los paquetes de respuesta de eco pasan por los mismos equipos de conmutación que los de petición, ¿cuál es el valor inicial que está asignando `google.com` al campo TTL del datagrama?

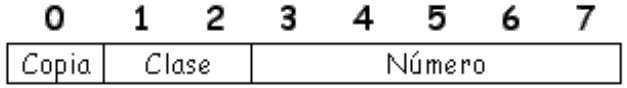


Detalles del datagrama de IPv4

IPv4



- **Opciones**: no se suelen usar para no sobrecargar la cabecera ni a los routers.

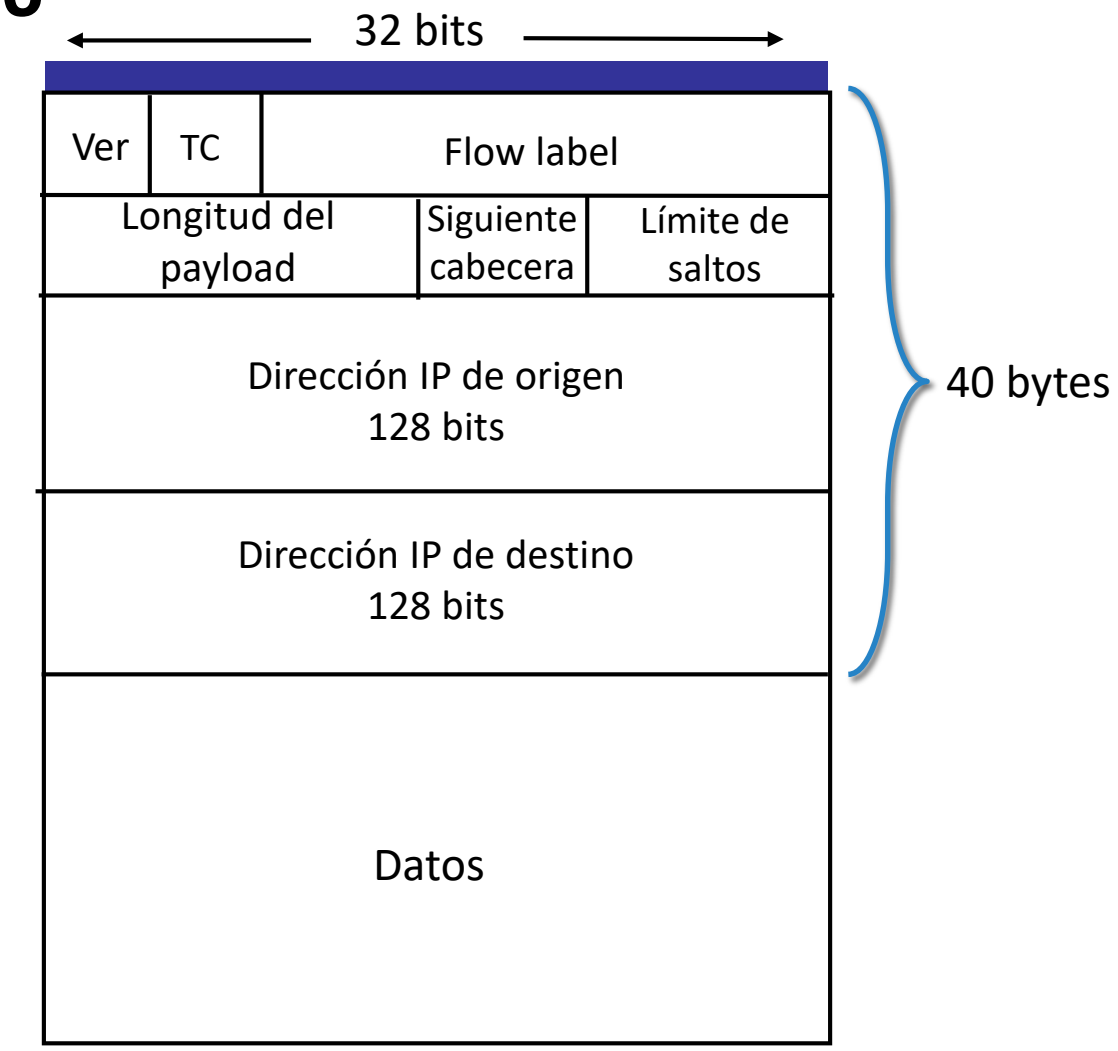


  - **Clase**: o tipo de opción
    - 0: control de red.
    - 2: depuración y test
    - 1 y 3: reservado para uso futuro
  - **Número**: identificador de opción
  - **Copia**: si debe copiarse a los fragmentos
- Las opciones más destacables son:
  - **Registro de ruta**: sirve para "grabar" en el datagrama la ruta por la que ha pasado. Los routers deberán ir rellenando este campo
  - **Encaminamiento desde el origen**: se indica el camino exacto que debe seguir un datagrama. El router no mira tablas de enrutamiento sino que mira en este campo donde se debe enviar
  - **Marca de tiempo**: Igual que la primera opción pero el router, además de grabar su dirección debe grabar el tiempo de paso del datagrama

Formato del datagrama
Detalles de IPv6

Detalles del datagrama de IPv6

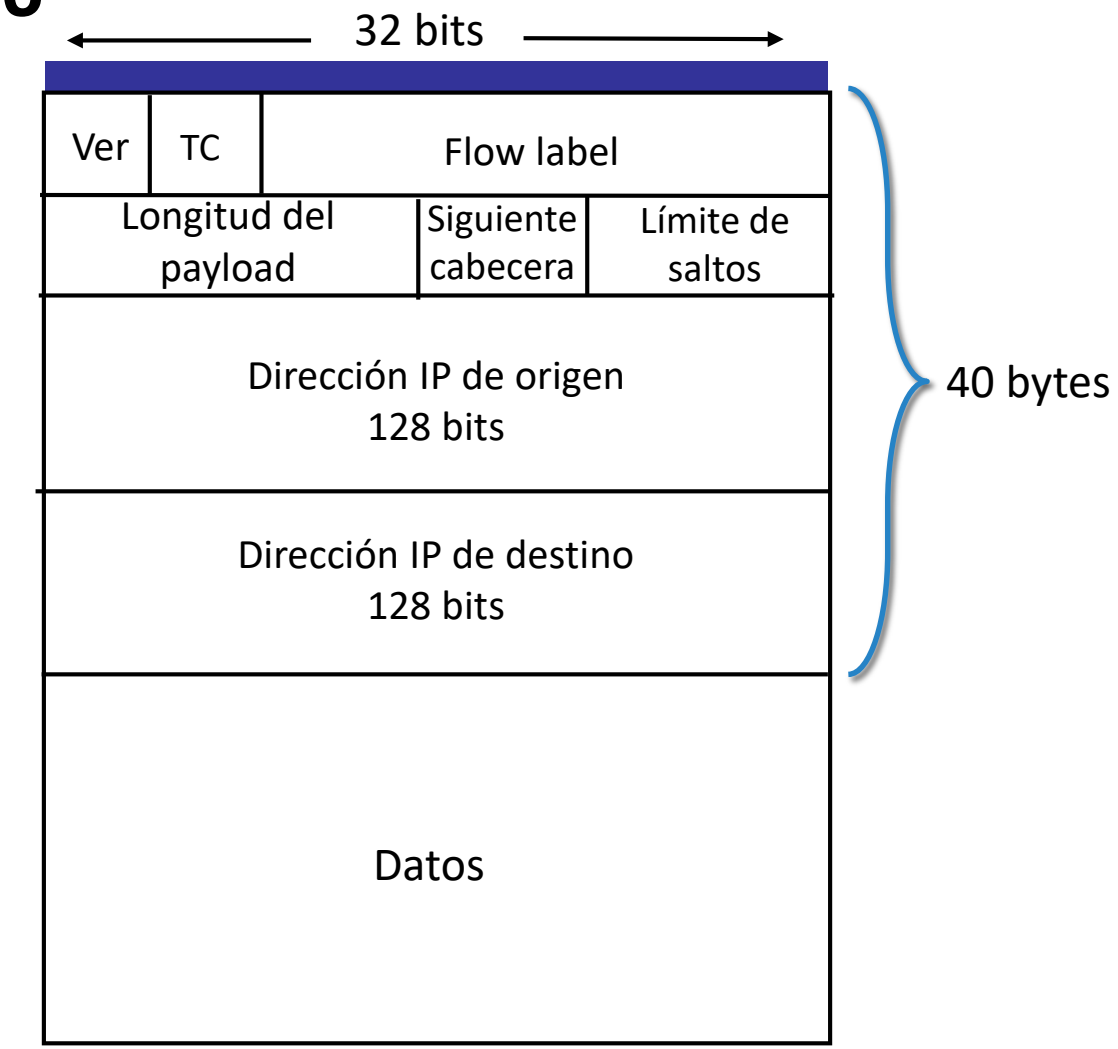
IPv6



- **Ver** (4 bits): versión (como en IPv4)
- **Traffic class** (8 bits): clases de tráfico o prioridades
- **Flow label** (20 bits): sirve para etiquetar paquetes que pertenezcan a un mismo flujo (*stream*), por ejemplo, de audio o vídeo
- **Payload lenght** (16 bits): longitud del campo de datos
- **Next header** (8 bits): identificación de la cabecera del paquete que se está encapsulando
- **Hop limit** (8 bits): igual a TTL en IPv4

Detalles del datagrama de IPv6

IPv6



- **Next header** (8 bits): introduce el concepto de cabeceras encadenadas, ampliando las posibilidades respecto a IPv4:
  - 4 - IPv4
  - 6 - TCP
  - 17 - UDP
  - 41 - IPv6
  - 44 - Cabecera de fragmentación
  - 0 - Opciones *hop-by-hop* (https://tools.ietf.org/html/rfc2460#page-11)

Un datagrama IPv6 puede contar con **múltiples cabeceras**, dependiendo de las necesidades, evitando así el uso de los campos menos útiles de IPv4 en la cabecera estándar

Cuestiones clave de este tema

Cuestiones clave

Qué deberías saber

Al inicio de este tema se planteaban unos objetivos específicos que deberían permitirte ***responder a las siguientes cuestiones*** *clave*

Cuestiones

- ¿Cómo se encapsulan los segmentos TCP/UDP en un datagrama IP?
- ¿Cuál es la finalidad de los campos más importantes en la cabecera de los datagramas IPv4 e IPv6?
- ¿Qué trabajo realizan los equipos de interconexión sobre el datagrama a medida que lo reenvían?
- ¿Cómo se controla que un datagrama no quede circulando demasiado tiempo sin llegar a su destino?
- ¿Cuál es el mecanismo que permite enviar paquetes de datos mayores que el fijado por el MTU?

Material adicional

Descripción

Para ampliar tus conocimientos sobre los contenidos de esta semana te recomendamos que consultes los recursos indicados a continuación.

Recursos

- **Capítulo 4** La capa de red: el plano de datos, del libro Redes de computadoras 7ED disponible en formato digital en la BUJA (recuerda identificarte para poder acceder a leerlo desde tu navegador), concretamente desde la **sección 4.3** en adelante
- **IP datagram encapsulation and format** en el recurso electrónico The TCP/IP Guide, donde encontrarás información sobre los campos del datagrama y un esquema general del encapsulamiento en IPv4
- **IP datagram size, MTU and fragmentation** en el recurso electrónico The TCP/IP Guide para conocer en detalle el proceso de fragmentación y reensamblado de datagramas