PAR – Unidad 7

IMPLANTACIÓN DE REDES DE ÁREA LOCAL :

CAPA DE RED

Capa de red

- La capa de red es responsable de la **entrega de paquetes de extremo a extremo** seleccionando y gestionando los saltos necesarios entre los *routers* que conectan las distintas redes intermedias
  - mientras que la capa de enlace de datos lo es de la entrega de paquetes *entre un router y el siguiente* (entre salto y salto)
- Distinguir entre:
  - internet, Internet e intranet
  - internet y subred
  - sistema final (SF) e intermedio (SI)
  - Puente, conmutador 2/3 y encaminador

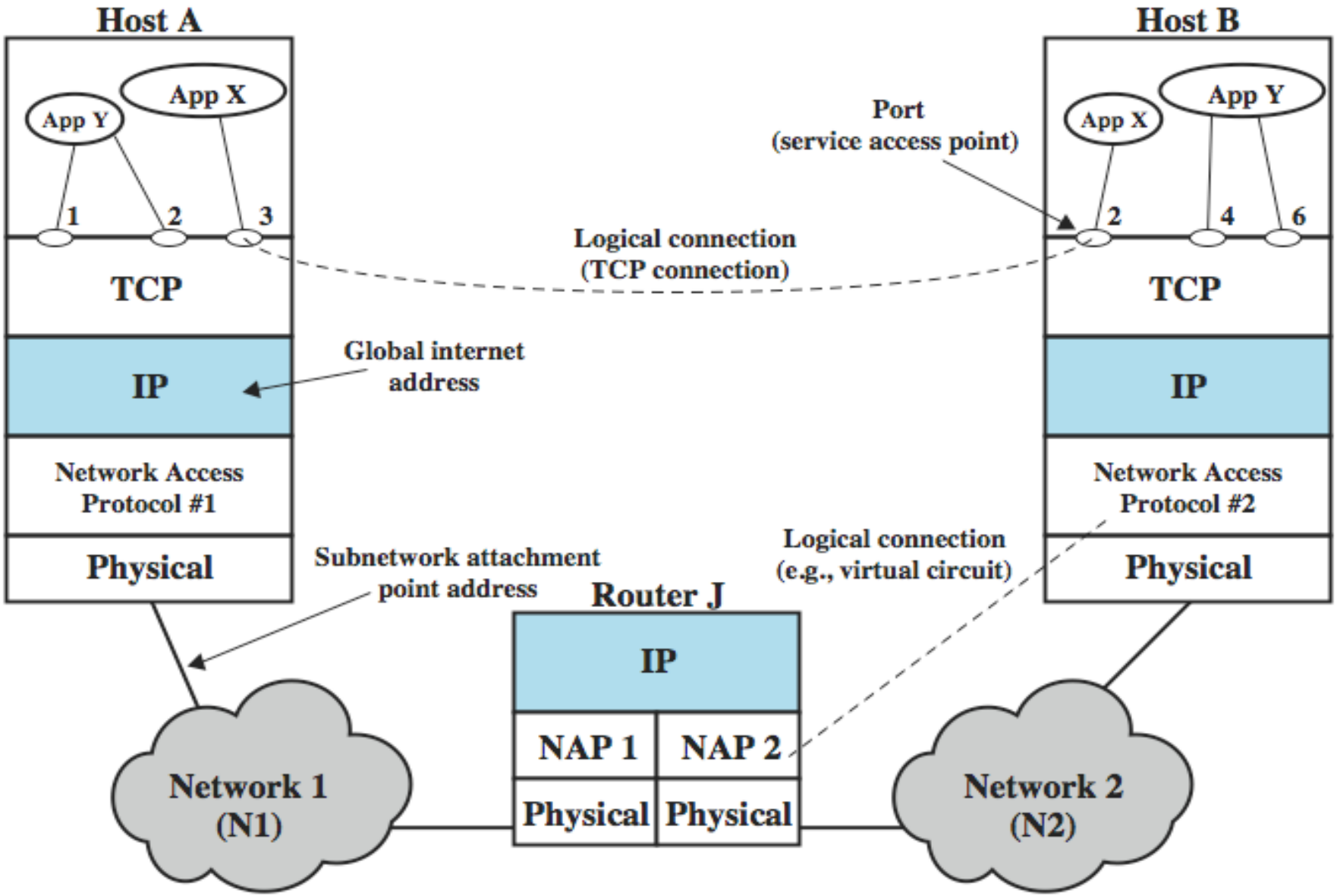
Arquitecturas de red

- Funcionamiento **no orientado a la conexión**:
  - correspondencia con redes de conmutación de paquetes
  - cada PDU se trata independientemente y se encamina desde el ES origen al ES destino a través de una serie de subredes, decidiendo cada uno de los *routers* el camino a seguir por cada PDU
  - ventajas: flexible y sin sobrecargas innecesarias
  - desventajas: no garantiza la entrega ni el orden de entrega. Por tanto, si se quiere fiabilidad, ésta es responsabilidad de las capas superiores (p.e., TCP)
- Funcionamiento **orientado a la conexión**:
  - se establece primero una conexión lógica a nivel de red entre cualquier par de nodos y se intercambian datos, siguiendo todos la misma ruta por el sistema intermedio
  - alternativa: circuitos virtuales

Cuestiones de diseño

- **Encaminamiento**:
  - hay una tabla de encaminamiento en cada nodo, que guarda el siguiente *router* para cada posible destino
  - esta tabla puede ser estática o dinámica
- **Tiempo de vida** de los datagramas (TTL):
  - posibilidad de que un datagrama vague a través de la red indefinidamente por falta de coherencia entre las tablas
- **Fragmentación** y ensamblado:
  - cuando el MTU (*Maximum Transmission Unit* o tamaño máximo de datos transmisible) en la capa de enlace de la siguiente red a atravesar es menor que el tamaño del datagrama
- **Control de errores y de flujo**:
  - los *routers* pueden descartar datagramas (TTL, CRC, congestión), informar de ello (se necesitaría un identificador) e incluso regular su envío limitadamente

Direccionamiento y Multiplexión



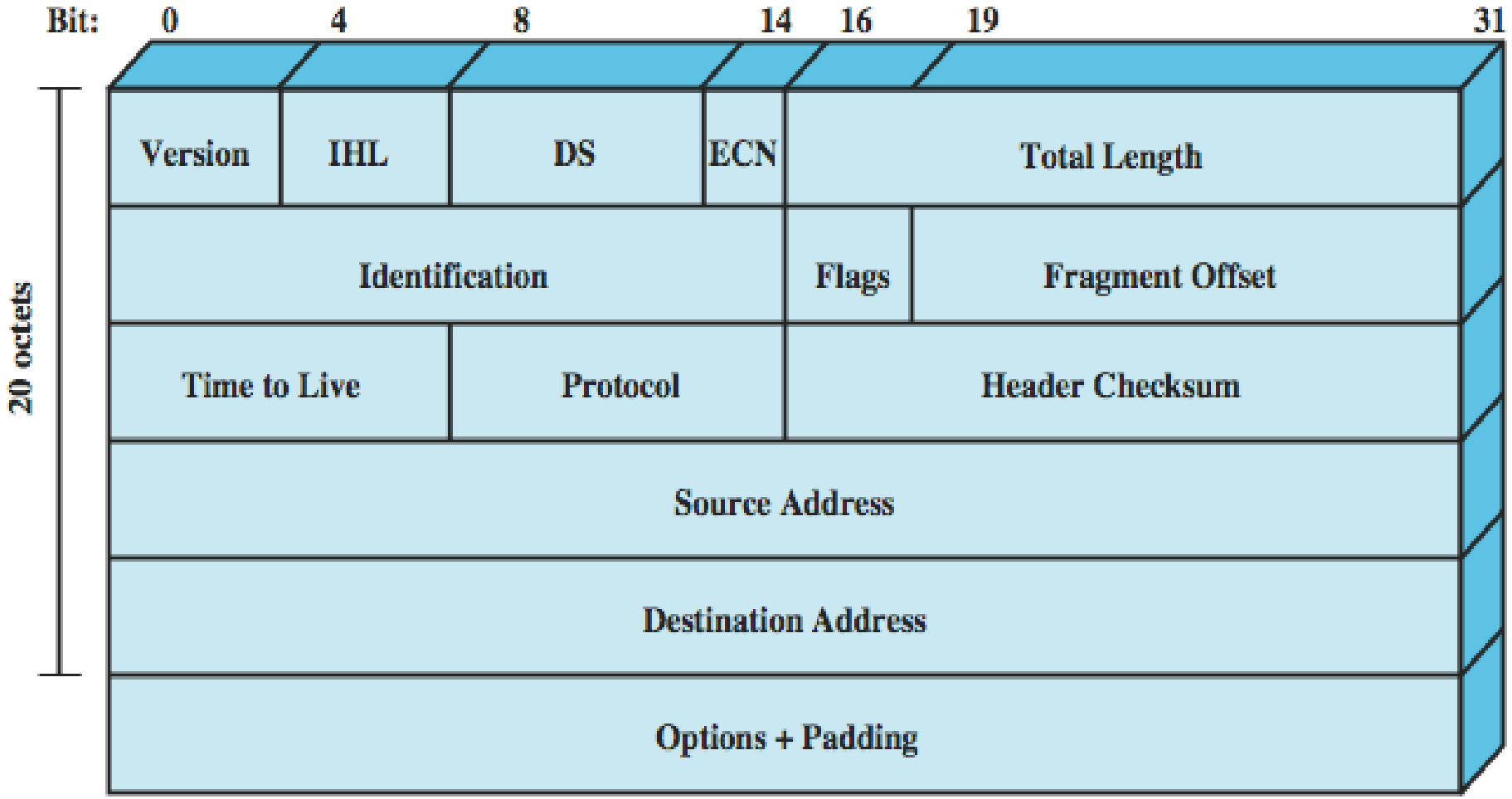
Protocolo de Internet (IP) v4

- Definido en **RFC 791** por la IETF (rev. 1122)
- Parte del conjunto de protocolos de Internet (TCP/IP)
- Se supone que debe ser sustituido por IPv6
- Como cualquier protocolo estándar, se especifica en 2 partes:
  - interfaz con la capa superior, especificando los servicios que se le proporciona [dirección vertical]
  - sintaxis y semántica de las PDU y los mecanismos asociados a su intercambio [dirección horizontal]
- No es el único protocolo de la capa de red o Internet:
  - la mayoría de los restantes son de control
  - ICMP, (R/)ARP, IGMP, IPSec, ...

Servicios IP

- Los servicios a proporciona a la capa superior (p.e., los que ofrece IP a TCP o UDP) se expresan en términos de primitivas (funciones) y parámetros (argumentos)
- IP proporciona **dos primitivas de servicio** en la interfaz con la capa superior:
  - **envío** (*send*): para solicitar a IP la transmisión de un PDU
  - **entrega** (*deliver*): para solicitar a IP ser avisado de la llegada de un PDU
- Los **parámetros** asociados a estas dos primitivas son:
  - dirección de origen y destino
  - protocolo, tipo de servicio, TTL, identificadores, opciones
  - longitud de los datos
  - datos

Cabecera IPv4



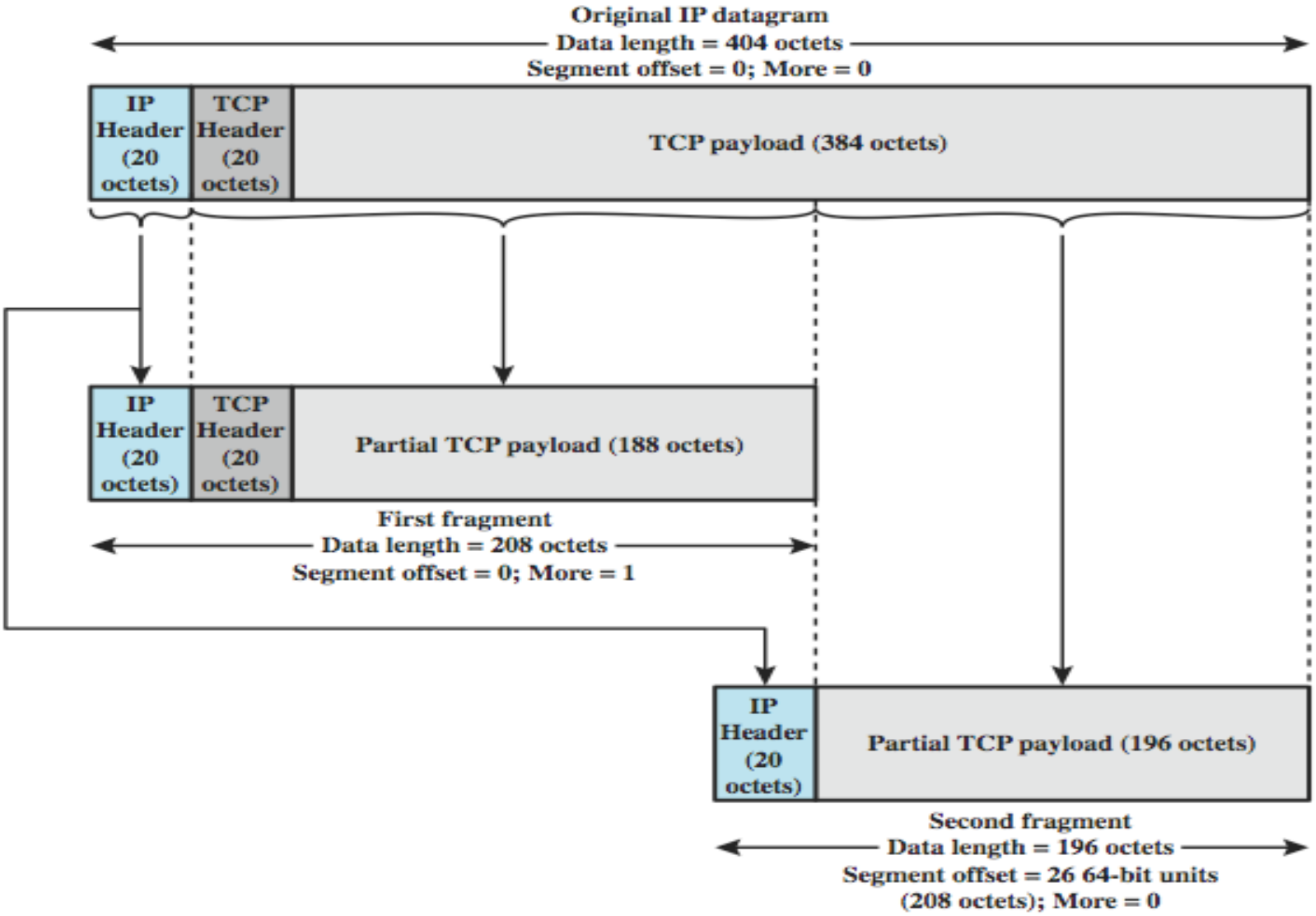
Campos de la cabecera IP (1)

- **Versión** [4b]: x04 (x06 para IPv6)
- **Longitud de cabecera de Internet** [4b] (IHL)
  - incluida las opciones
  - expresada en palabras de 32b (valor mínimo: 5 => 20B)
- **Tipo de servicio** [8b] (DS/ECN)
  - fiabilidad, prioridad, retardo y rendimiento (se usa poco)
  - DS (*Differentiated Services*) ocupa 6 bits, ECN (*Explicit Congestion Notification*) ocupa 2 bits
- **Longitud total** [16b] del datagrama en bytes
  - entre 20B y 64535B
  - todo host debe poder recibir 576B
- **Identificador** [16b] único de los fragmentos de un PDU

Campos de la cabecera IP (2)

- **Indicadores** [3b], sólo se usan dos:
  - el bit DF (*Don't Fragment*) prohíbe la fragmentación, sirve para regular el tamaño de la PDU
  - el bit MD (*More Data*) indica si quedan más fragmentos
- **Desplazamiento del fragmento** [13b]
  - indica el lugar donde se sitúa el fragmento dentro del datagrama original, medido en unidades de 64bits (fragmentos múltiplos de 8B ==> máx. 8192 fragmentos)
- **Tiempo de vida (TTL)** [8b] en segundos=saltos
- **Tipo** [8b] de protocolo de la capa superior
- **Suma de comprobación** [16b] de la cabecera
  - se comprueba y recalcula en cada salto: es el C. a uno de la suma de los C. a uno de cada palabra de 16b de la cab.

Ejemplo de fragmentación



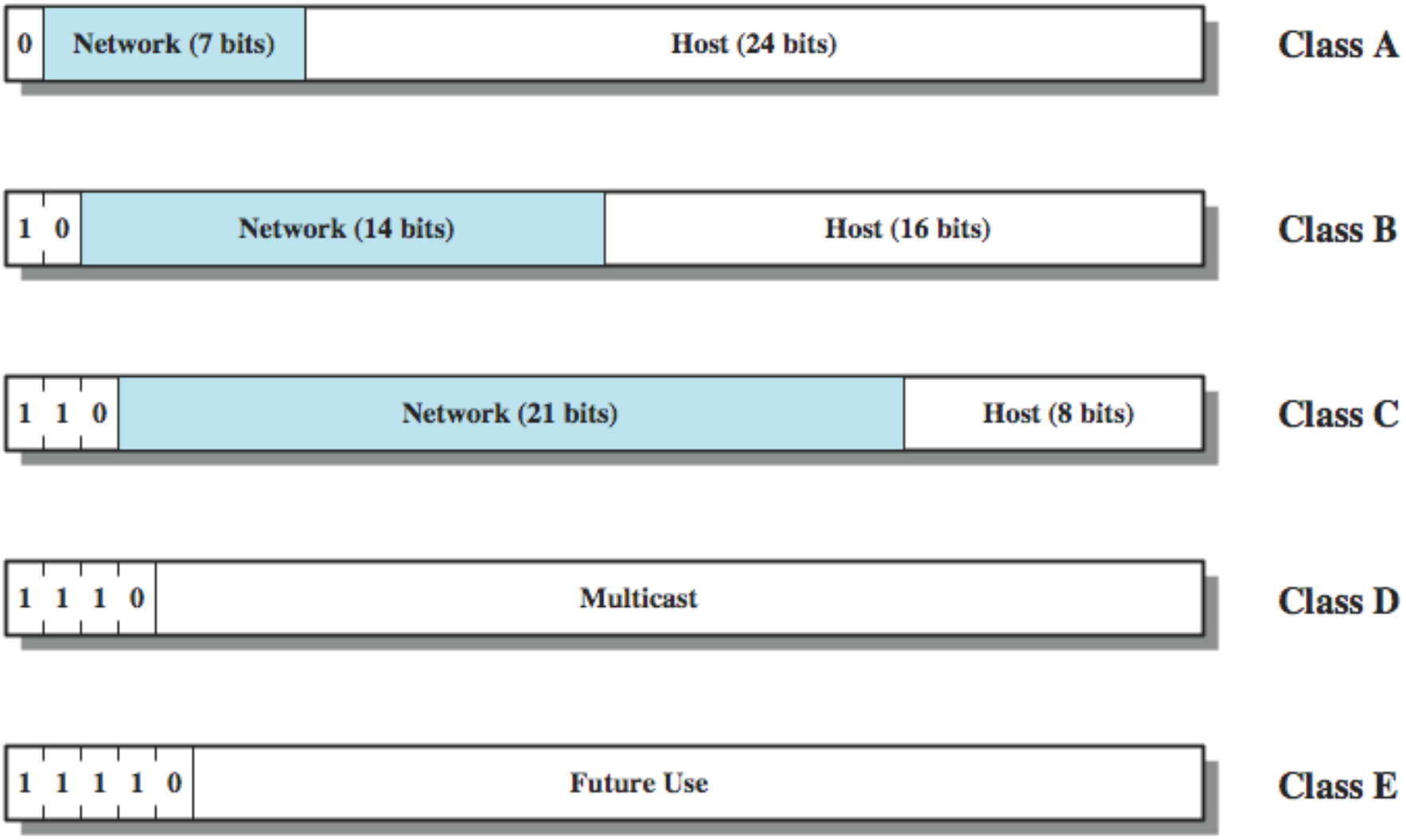
Campos de la cabecera IP (3)

- **Dirección de origen** [32b] y **de destino** [32b]
- **Opciones** [variable]
  - seguridad (mediante etiquetas)
  - encaminamiento en el origen (indica la secuencia de *routers* por el datagrama)
  - registro de la ruta (solicita a los *routers* que añadan su dirección al campo de opción)
  - identificación de la secuencia (para reservar recursos)
  - marcas de tiempo (para medir retrasos, latencia, etc.)
- **Relleno** [variable] hasta que la IHL es múltiplo de 32b
- **Datos** [variable]
  - múltiplo de 8b hasta un máximo de 65535B

Direcciones IP

- Cada interfaz de red tiene asignada una dirección IP, que está formada por, al menos, dos partes:
  - un identificador de red (*netid*)
  - un identificador de *host* (*hostid*)
- Se han utilizado tres esquemas de asignación de direcciones IP diferentes:
  - direcciones basadas en clases (A, B, C, D y E)
  - subredes (añade un identificador: red/subred/*host*)
  - direcciones sin clases (usado actualmente)
- Para evitar el agotamiento de las direcciones IP:
  - traducción de direcciones de red (NAT)
  - IPv6

Direcciones IPv4 con Clase



Clases de direcciones (ICANN)

- **Clase A**: 1-127.[0.0.0] : 16.777.214 *hosts*/red
  - 0.0.0.0 = *default* ;
  - 10.0.0.0-10.255.255.255 direcciones privadas
  - 127.[0.0.0] reservada para *loopback*
- **Clase B**: 128.0.-191.255.[0.0] : 65.534 *hosts*/red
  - 172.16.0.0-172.31.255.255 direcciones privadas
- **Clase C**: 192.0.0.-223.255.255.[0] : 254 *hosts*/red
  - 192.168.0.0-192.168.255.255 direcciones privadas
- **Clase D**: 224-239.255.255.255: grupos de multidifusión
- **Clase E**: 240-255.255.255.255: reservadas
  - 255.255.255.255 = difusión en la red local
  - red.255. ... = difusión en la red indicada

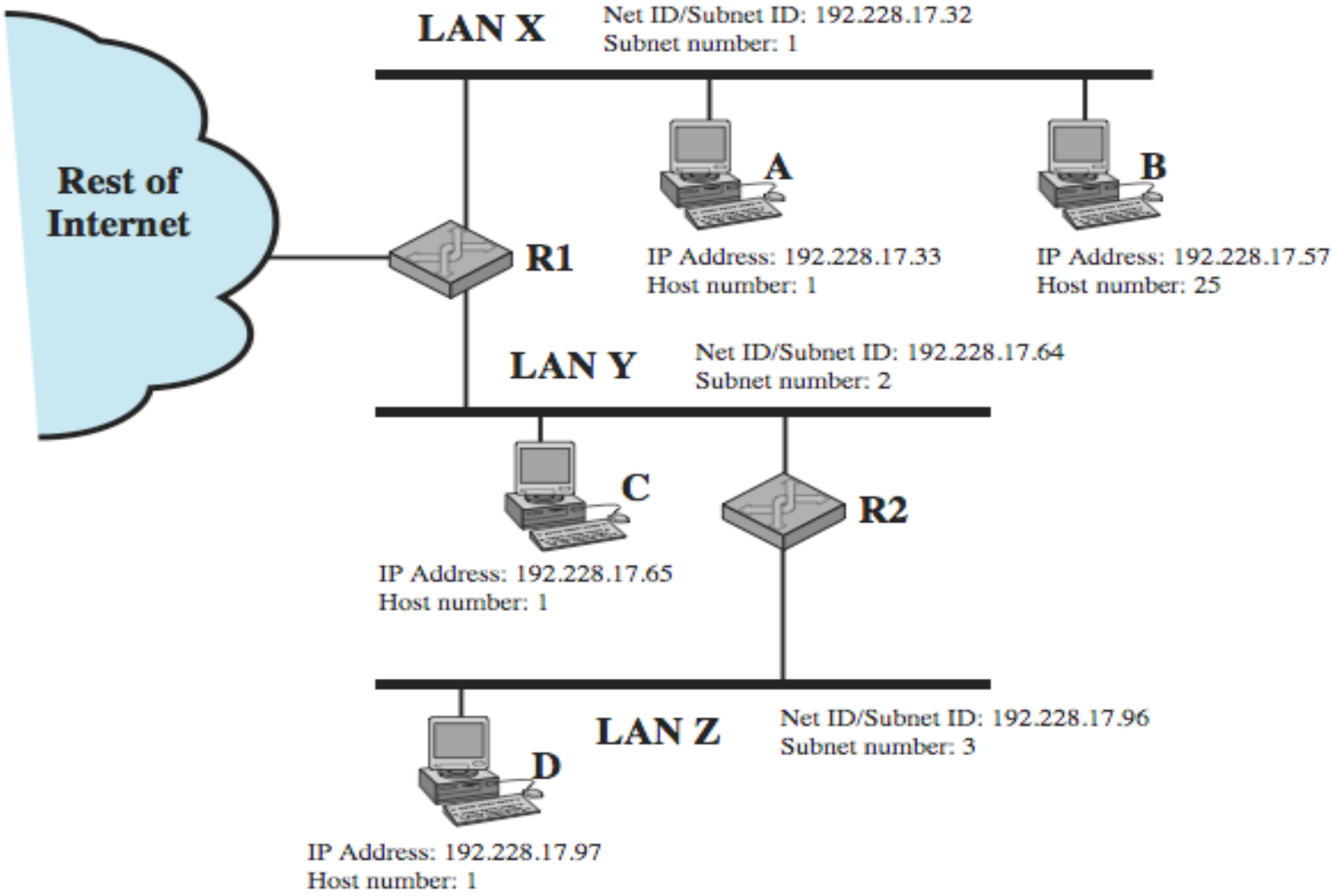
Subredes y Máscaras

- En una red grande, dividida en muchas LANs, el rígido esquema anterior causa problemas a la hora manejar el conjunto de direcciones asignado
  - un router por cada LAN conectada a Internet !?
- El concepto de **subred** sirve para separar las funciones de encaminamiento de una organización particular de las de la interred global:
  - el sito es para el resto de Internet una única red (*netid*)
- A cada LAN se le asigna un número de subred, que forma parte del identificador de host (*idhost*)
- Los routers locales se encarga del encaminamiento dentro de las subredes de la red
- La **máscara de subred** define los límites entre la parte de subred y la de host dentro de una subred dada

Cálculo de la máscara

- Ejemplo: host destino 192.228.17.57 (/255.255.255.224)
  - Dirección IP : **192.228.17.57**
    - **11000000.11110100.00010001.00111001**
  - Máscara de subred: 255.255.255.224
    - **11111111.11111111.11111111.111**00000
    - al ser de clase C:
      - bits a 1: **red**(clase C)+**subred**; bits a 0: host
  - AND bit a bit entre dirección y máscara
    - **11000000.11110100.00010001.001**00000
    - 192.228.17.32 = netid+subid
    - *netid*: 192.228.17.0 / *hostid*: 57
      - subid: 1 (de 0-7) / hostid: 25 (de 1-30, 0 de red/31 de difusión)
    - el router hace el AND a la dirección IP de destino y coge la red de la tabla de encaminamiento según el resultado

Enrutado usando subredes



Direcciones sin clase (CIDR)

- **Classless Inter-Domain Routing** (RFC 1519):
  - generalización del concepto de máscara de red:
    - el *netid* puede ocupar cualquier nº de bits (no sólo 8, 16, o 24), expresándose: w.x.y.z/n (n=nº de bits de la red)
  - si una organización solicita una dirección de red de 1000 *hosts*, se le podría asignar un bloque de 1024, en vez de uno de 65533 (clase B)
  - el encaminamiento se hace más complicado:
    - AND entre la dirección de destino y las máscaras de red guardadas en la tabla de encaminamiento del *router*
    - se selecciona la red con cuya máscara la coincidencia es mayor (puede coincidir con varias entradas de la tabla)
    - p.e., para el destino 200.64.17.46 y éstas dos entradas en la tabla: 200.64.0.0/16 y 200.64.17.0/24; se elige la 2ª

Address Resolution Protocol (ARP)

- Se necesita una dirección MAC para enviar una trama a un *host* de una LAN (dominio difusión), esta se podría obtener:
  - de forma manual
  - incluida en la dirección de red
  - directorio central
  - preguntando al resto de hosts del dominio (~ ARP)
- ARP (RFC 826) proporciona un mapeo dinámico entre direcciones IP y MACs
  - el origen difunde un petición ARP
  - el destino contesta con una respuesta ARP
  - los direcciones se almacenan en una tabla durante unos pocos minutos (revalidables)

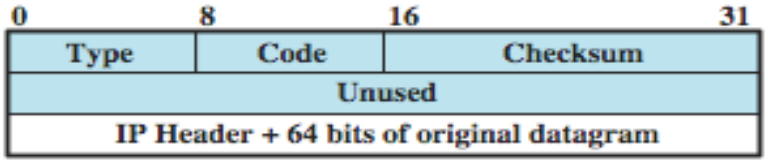
Configuración dinámica de hosts

- Necesidad de automatizar la configuración de los *hosts* de una red: dirección IP, *router*, servidor DNS, ...
- *Reverse* ARP (RARP) (RFC 903)
  - muy simple sólo proporciona la dirección IP
- BOOTP (RFC 951, 1048 y 1084)
  - usa UDP
  - sólo proporciones direcciones estáticas (hay que relacionar explícitamente las MACs y las IPs)
- DHCP (RFC 2131, 2132)
  - usa UDP
  - proporciona direcciones estáticas y dinámicas (que se “arriendan” (*lease*) temporalmente y hay que renovar)

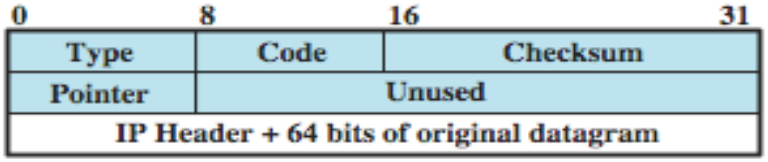
ICMP

- *Internet Control Message Protocol* (RFC 792)
- Transferencia de mensajes de control entre *routers* y *hosts*, retroalimentación sobre problemas:
  - datagrama que no puede alcanzar su destino
  - *router* no puede almacenar un datagrama
  - hay una ruta más corta, etc...
- Se encapsula en un datagrama IP (tipo = x01)
- El formato de su cabecera lo componen:
  - **tipo** [8b] de mensaje ICMP
  - **código** [8b] o parámetros cortos del mensaje
  - **suma de comprobación** [16b] del mensaje completo
  - **parámetros** [32b] más largos

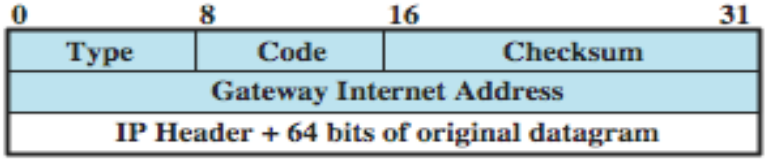
Formatos ICMP



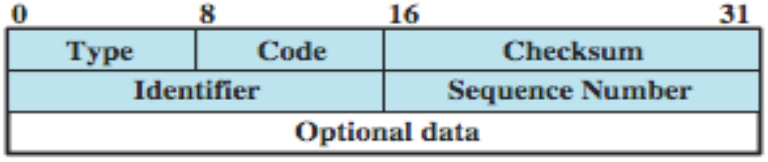
(a) Destination Unreachable; Time Exceeded; Source Quench



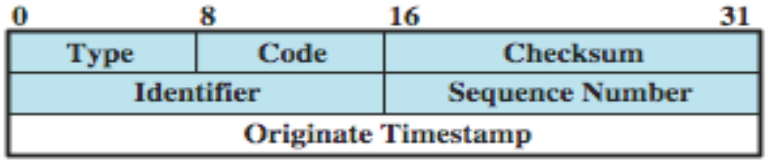
(b) Parameter Problem



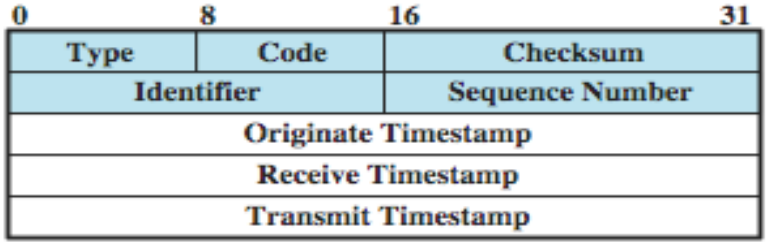
(c) Redirect



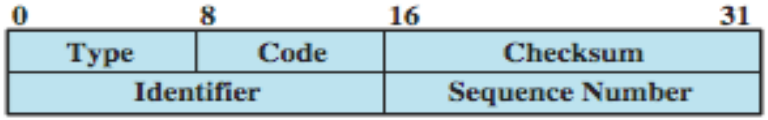
(d) Echo, Echo Reply



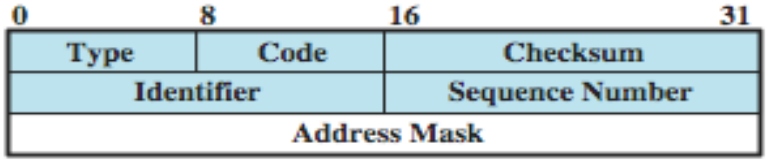
(e) Timestamp



(f) Timestamp Reply



(g) Address Mask Request



(h) Address Mask Reply

Mensajes ICMP comunes

- Destino inalcanzable
- Tiempo excedido
- Problema con los parámetros u opciones
- Ralentización de origen (rudimentario control de flujo)
- Redirección (porque hay una ruta mejor)
- Petición de eco y respuesta
- Marca de tiempo y respuesta con marca de tiempo
- Petición de máscara de la red local y respuesta
- ... hasta 40

Otros protocolos relacionados

- SNMP (*Simple Network Management P.*) (RFC 1052)
  - monitorización y reconfiguración centralizada de redes a partir de la información enviada desde los nodos. Seguirá en la unidad 9
- DNS (*Domain Name System*) (RFC 1034, 1035)
  - facilita el manejo de las direcciones de red organizando un sistema de nombres de forma jerárquica (por dominios) y que hace corresponder con las direcciones IP numéricas
- IGMP (*Internet Group Management P.*) (RFC 1112)
  - usa una dirección de clase D para crear un grupo de multi-difusión (los paquetes enviados le llegan a todo el grupo) al que se pueden unir *hosts* de distintas redes enviando una petición de suscripción a su *router*(IGMP-*aware*) local
  - hay direcciones permanentes (224.0.0.1 = *hosts* locales 224.0.0.2 = *routers* locales, ...) y temporales

Protocolos de Enrutamiento

- Los *routers* reciben y reenvían paquetes
- Toman decisiones basadas en el conocimiento de la topología y las condiciones del tráfico (retrasos)
- El uso de enrutamiento estático está limitados a redes pequeñas y sujetas a pocos cambios
- En el resto se usan algoritmos de enrutamiento dinámico
- Se distingue entre:
  - información de enrutamiento: sobre topología y retrasos
  - algoritmos de enrutamiento: toman las decisiones de enrutamiento basadas en dicha información.
- Seguirá en la unidad 8

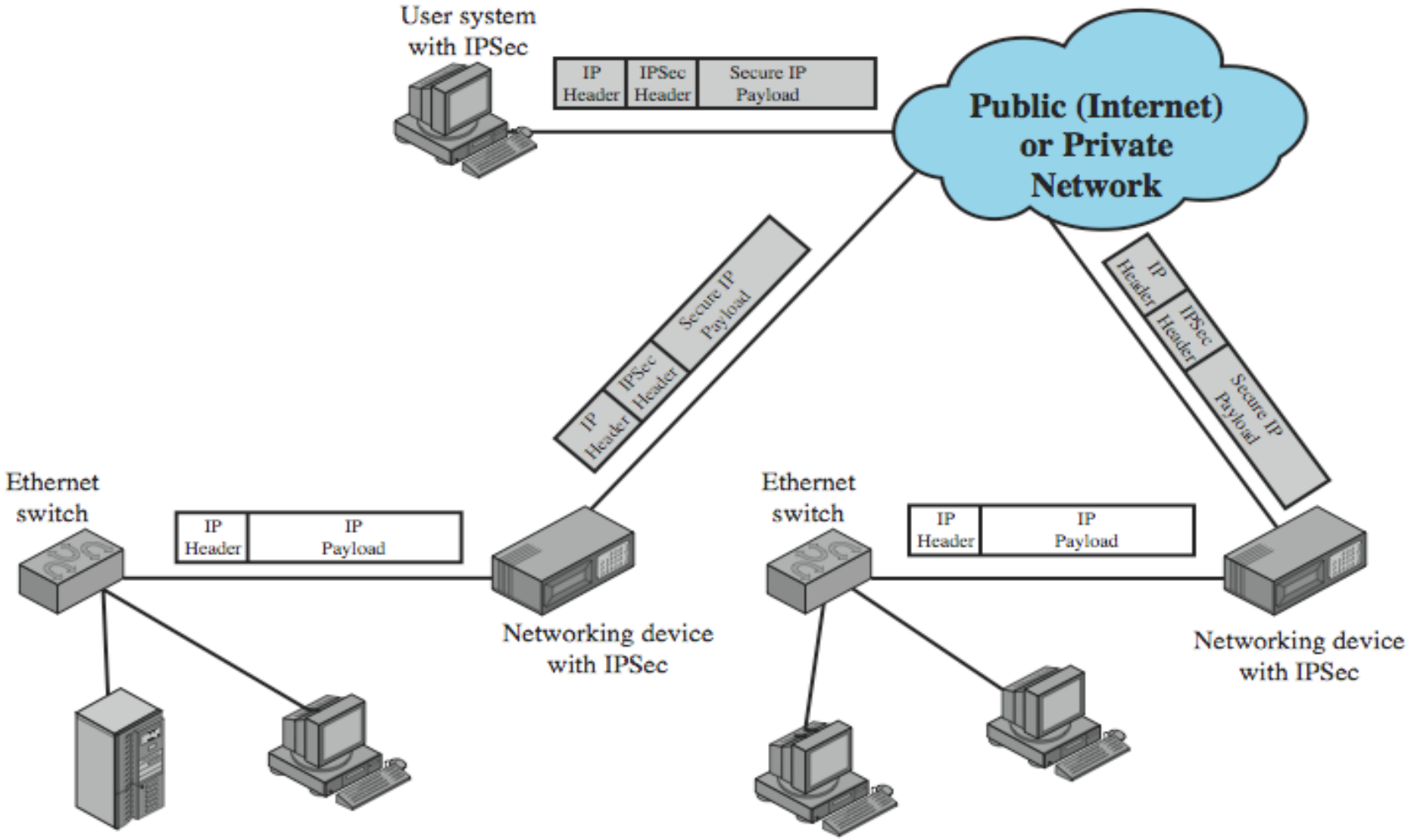
IP versión 6

- RFC 1752, 2460, 2373, ...
- Razones para el cambio:
  - agotamiento del espacio de direcciones de la versión 4
  - demanda de nuevos tipos de servicios
- Mejoras:
  - direcciones de 128 bits y más flexibles:
    - 66,7 trillones ($6{,}67 \times 10^{19}$) direcciones/cm^2 de la Tierra
  - mecanismo de opciones más eficaz
  - los paquetes se pueden etiquetar para identificar un determinado flujo de datos y poder tratarlos de forma específica
- Seguirá en la unidad 9

Redes Privadas Virtuales (VPN)

- Conjunto de *hosts* interconectados usando una red insegura
  - p.e. conectando las LANs de una organización a través de Internet
- Se usa encriptación y protocolos especiales para proporcionar seguridad:
  - para evitar robo de información y el acceso a usuarios no autorizados
- Las soluciones propietarias son problemáticas, por esto se desarrolló el estándar IPSec (RFC 1636):
  - es una extensión al protocolo IP que añade cifrado fuerte para permitir servicios de autenticación y, de esta manera, asegurar las comunicaciones a través de dicho protocolo. Inicialmente fue desarrollado para usarse con el nuevo estándar IPv6, aunque posteriormente se adaptó a IPv4

Esquema IPSec



Filtrado de paquetes

- Proceso que permite controlar y manipular los paquetes que atraviesan la pila de protocolos en función de su contenido, tanto en la capa de red, como de transporte o de enlace (o de aplicación)
- Suele estar íntimamente ligado al software de red del sistema operativo:
  - p.e., en el núcleo Linux actual se dispone de *Netfilter*, que es un conjunto de 'ganchos' (*hooks*) a los que se pueden 'enganchar' funciones que filtren y manipulen los paquetes de red durante su tránsito a través de la pila de protocolos
  - así mismo se nos proporciona herramientas como *iptables* para manejar dichos 'ganchos'.
  - también se puede filtrar tramas en la capa de enlace con herramientas como *ebtables*
  - Es importante que sea *stateful* (que siga las conexiones)