PAR – Unidad 10

# ACCESO A INTERNET Y MONITORIZACIÓN

# Network Address Translation(NAT)

- Método que cambia las dir. IPs de los datagramas que pasan por un *router* para mapear un espacio de direcciones en otro
- Hay varios tipos de NAT (rfc2663) (en ES):
  - uno a uno (NAT básico) y uno a muchos (NAPT / PAT)
  - estático (uno a uno) y dinámico
  - de origen ([S]NAT o Enmascaramiento) y de destino (DNAT)
- NAPT/PAT mantiene una tabla de conexiones establecidas:
  - sustituye los campos de dirección origen o destino de la cabecera IP por la IP pública del *router* y los campos de puerto origen o destino de la cabecera TCP/UDP por un número que indica una entrada de la tabla de conexiones (max. 65535) donde se almacena las IPs y puertos junto a otra información
  - objeciones: anti-capa, anti-IP, sólo TCP/UDP, hay aplicaciones especiales (p.e.FTP, VoIP, P2P, ...), límite de conexiones
  - NAT *traversal*: RSIP, UpnP/IGD, Bonjour/NAT-PMP, ...
- En Linux se configura con *iptables -j DNAT/SNAT/MASQUERADE*
- Herramienta *conntrack* para hacer seguimiento de las conexiones

# Funcionamiento del NAT

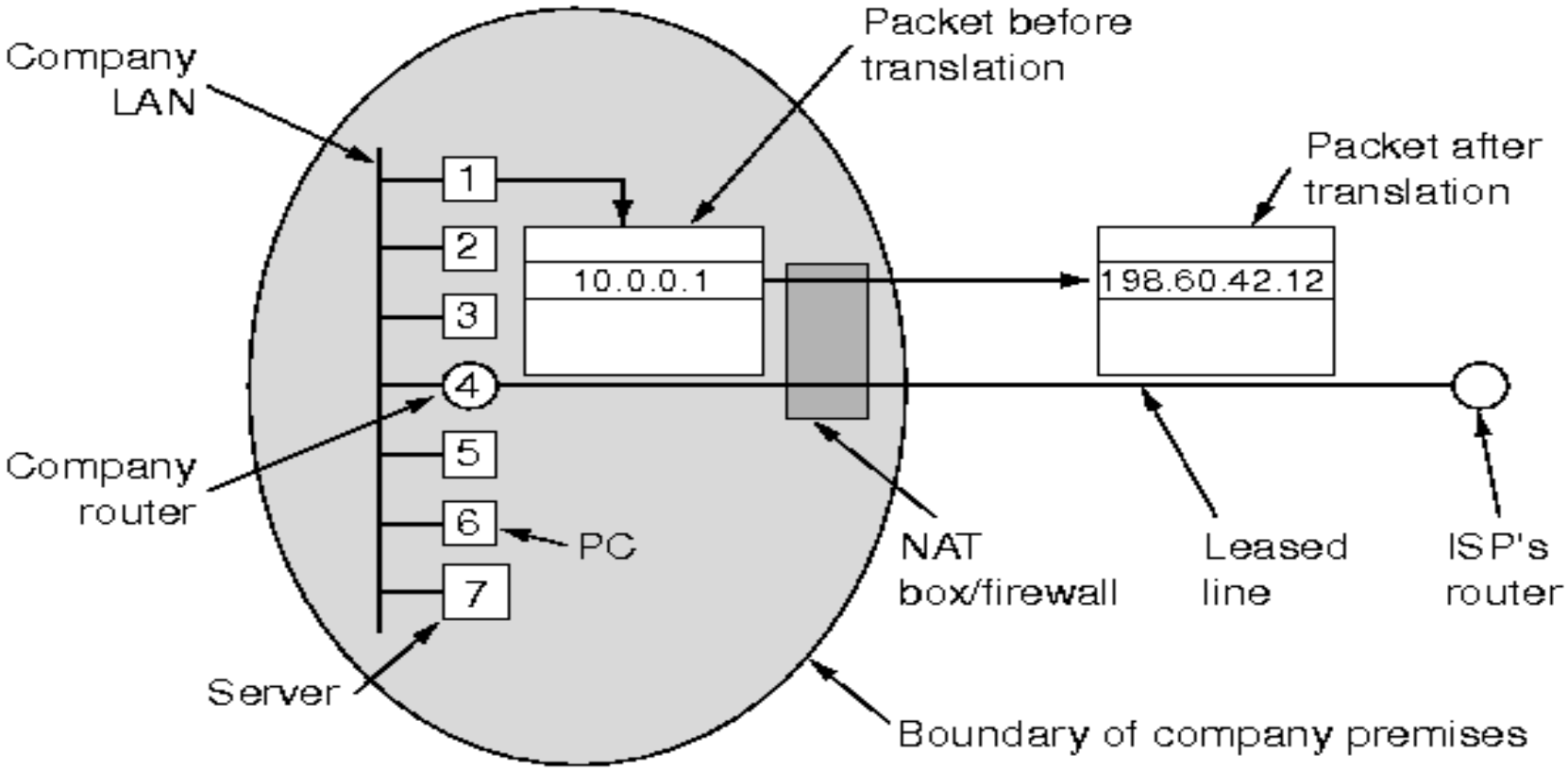
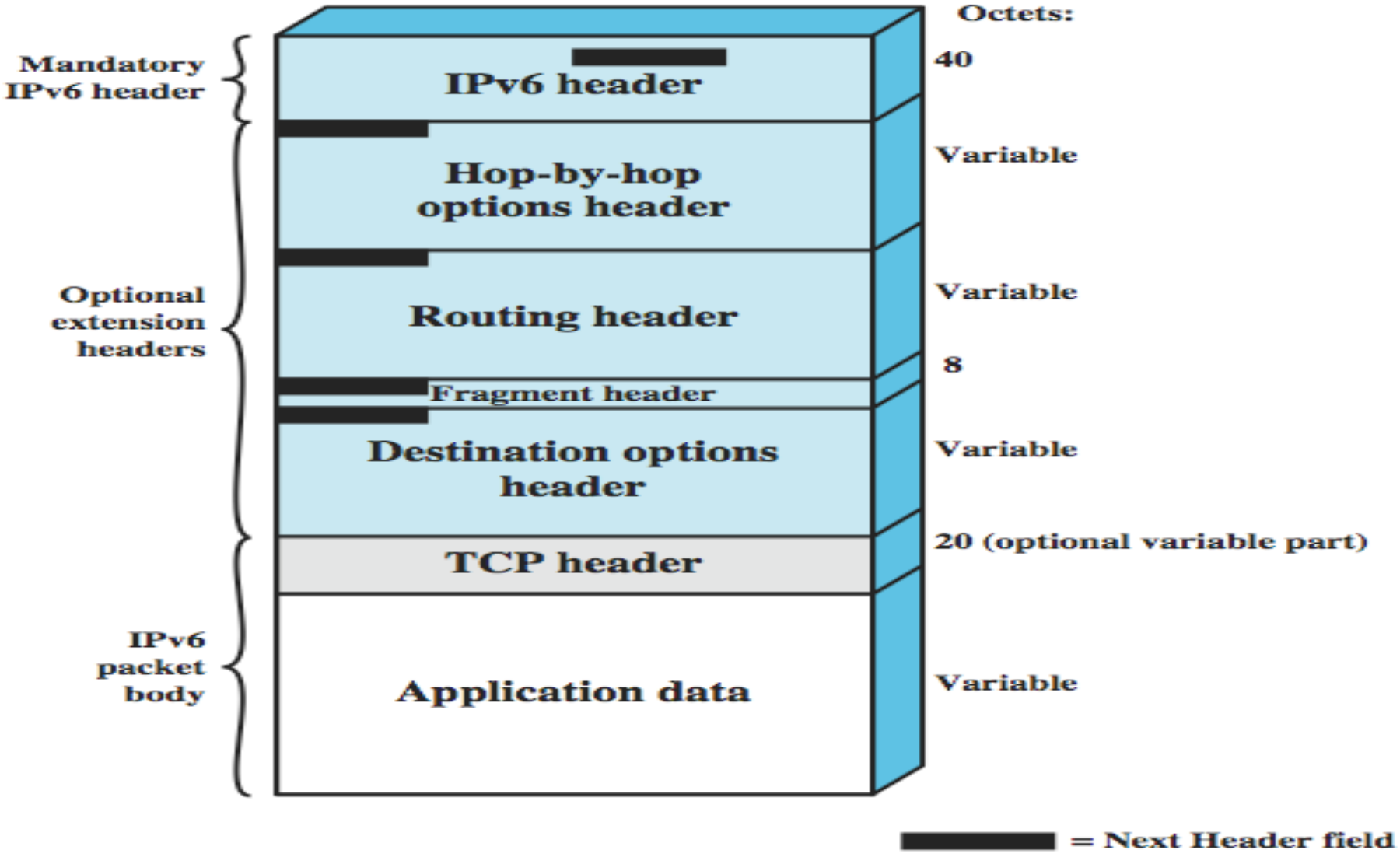



Fig. 5-60. Placement and operation of a NAT box.

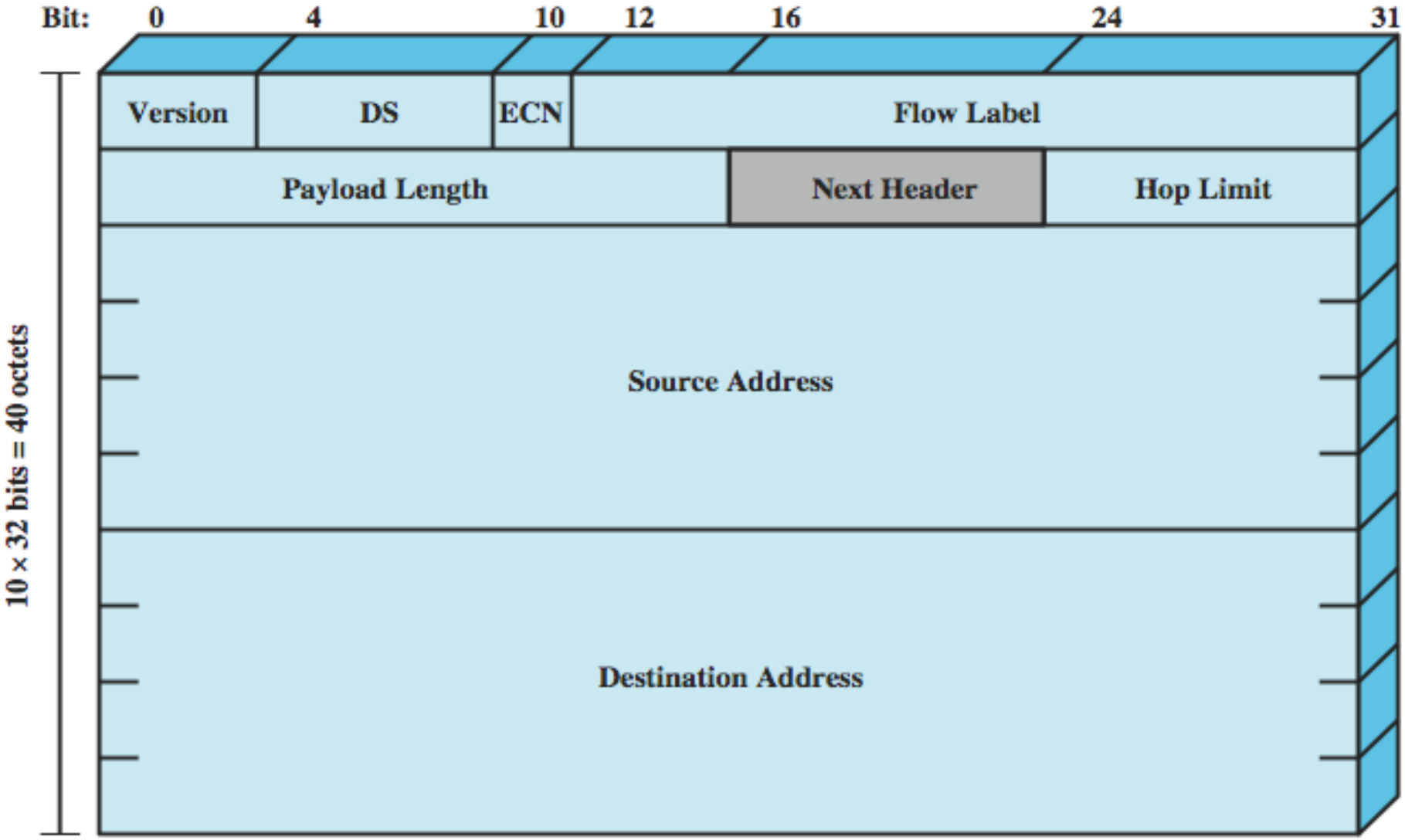
# IP versión 6

- RFC 1752, 2460, 2373, ...
- Razones para el cambio:
  - agotamiento del espacio de direcciones de la versión 4
  - nuevos tipos de servicios y mejor integración con los antiguos
- 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
- Mecanismos de transición:
  - Teredo (miredo): para nodos que se conectan mediante NAT
  - 6to4: para nodos con IP pública

# PDU IPv6


Octets:
Mandatory IPv6 header
IPv6 header
40
Optional extension headers
Hop-by-hop options header
Variable
Routing header
Variable
Fragment header
8
Destination options header
Variable
TCP header
20 (optional variable part)
IPv6 packet body
Application data
Variable
= Next Header field

# Cabecera IP v6



# Campos de la cabecera IPv6

- **Versión** [4b]: 6
- **Clase de tráfico** (DS + ECN) [8b]:
  - DS para distinguir distintas prioridades de tráfico
  - ECN para control de congestión de red
- **Etiqueta de flujo** [20b]:
  - permite el establecimiento de seudoconexiones, de forma que puedan reservarse recursos para éstas
- **Longitud de carga útil** [16b]:
  - tamaño de los datos en bytes
- **Encabezado siguiente** [8b]:
  - tipo de extensión de la cabecera ipv6 o tipo de la cabecera en la carga útil (compartida con ipv4)
- **Límite de saltos** [8b]: TTL
- **Dirección de origen** [128b] y **de destino** [128b]
- **Extensiones de la cabecera**

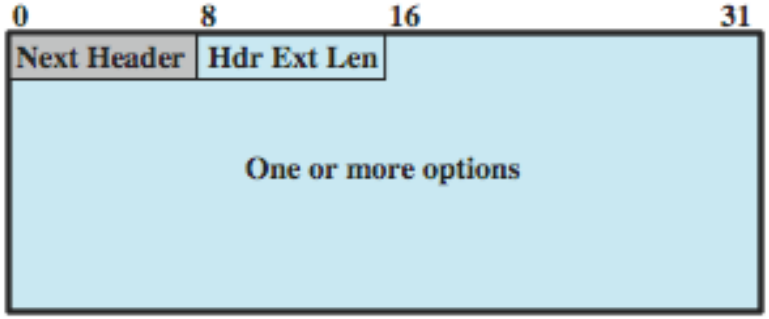
# Direcciones IPv6

- Cada interfaz suele tener varias direcciones
- Se escriben como 8 grupos separados por ':' de 4 dígitos hexadecimales(/nº bits de máscara de red) [rfc2373-es]:
  - los ceros a la izquierda pueden omitirse
  - 1 ó + grupos de 4 0s seguidos se sustituye por '::'  (una única vez)
  - las direcciones IPv4 pueden escribirse p.e. ::150.18.1.67
- Tres tipos de direcciones, según el destino:
  - ***unicast***: una única dirección
  - ***anycast***: al mejor nodo (más cercano) de un conjunto de direcciones
  - ***multicast***: un conjunto (8pref.+4flags+4scope + 112group) ff00::/8
- Tipos de direcciones,  según sean válidas en el ámbito:
  - de ***host*** (~ *loopback*)  ::1/128
  - ***link-local*** (~ no enrutable;~ 169.254.0.0)  fe80::/10
  - ***unique-local address*** (ant. s*ite-local;*  ~ privada) fc00::/7
  - **global** (~ pública)  2000::/3 (el ISP asigna una red de /48 a cada usuario y este la divide en subredes de /64: red:48b+sub:16b+h:64b)
- Identificador de interfaz: EUI-64 Modificado (derivado de la MAC), DHCPv6, aleatorio o manual

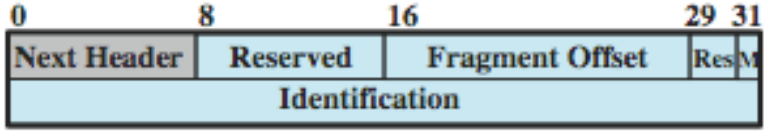
# Cambios con respecto a IPv4

- Se eliminan los siguientes campos:
  - IHL, porque la cabecera Ipv6 tiene longitud fija de 40B
  - tipo de protocolo, porque va en el último campo de encabezado siguiente
  - los relacionados con la fragmentación: se usa una extensión de cabecera y nunca se fragmenta en los routers
  - suma de verificación, porque se confía en la capa de enlace y de transporte y en la fiabilidad de la capa física
- Se establece el encabezado de extensión (opcional), organizado como una tupla: tipo [1B], longitud [1B], valor
- Se fragmenta en el origen nunca en los enrutadores
- Nuevo protocolo: ICMPv6 que reúne ICMP, IGMP y ARP
- Configuración automática sin estado (dhcp) de direcciones *local-link*: *Neighbor Discovery Protocol* (sustituto y extensión de ARP usando ICMPv6)

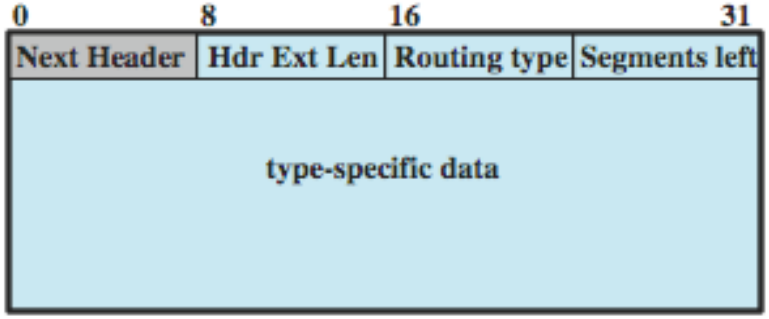
# Extensiones de la cabecera IPv6



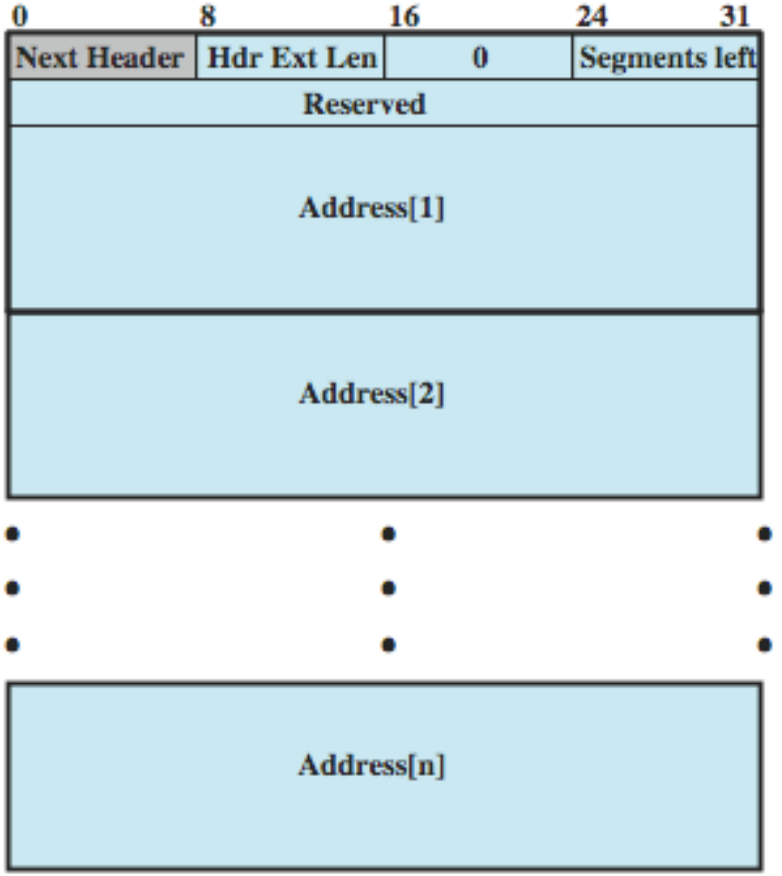

(a) Hop-by-hop options header; destination options header




(b) Fragment header




(c) Generic routing header




(d) Type 0 routing header

# Monitorización de redes: SNMP

- *Simple Network Management Protocol* es un protocolo de la capa de aplicación que facilita el intercambio de información entre dispositivos de red (RFC 1157,3410,...): SNMPv1/v2/v3
- Los dispositivos que normalmente soportan SNMP incluyen routers, switches, servidores, pcs, impresoras, bastidores de módem y muchos más.
- Permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas y planear su crecimiento
- Distingue entre los **agentes** que residen en los **dispositivos administrados** y los **sistemas administradores** (NMS)
- Y cuatro operaciones SNMP básicas: lectura, escritura, notificación (Trap) y transversal (InformRequest)
- El agente SNMP recibe peticiones  UDP en el puerto 161 y el administrador recibe notificaciones en el puerto 162
- SNMPv3 aporta seguridad criptográfica, + escalabilidad , ...

# Bases de Información de Gestión (MIB)

- Los agentes SNMP exponen los datos de gestión en los sistemas administrados como variables, organizadas en jerarquías y se describen en MIB
- El objeto administrado *atInput* se identifica por 1.3.6.1.4.1.9.3.3.1
  - o iso.identified-organization.dod.internet.private.enterprise.cisco.temporary.AppleTalk.atInput