Práctica 3 - IP, UDP e ICMP

nivel transporte

UDP = datagramas UDP con datos útiles

ICMP = protocolo para notificar errores y realizar tareas de operación relacionadas con el nivel IP
↳ EJ: mensajes ICMP

- **IP (Internet Protocol)**: protocolo de nivel de red que permite enviar info extremo a extremo a lo largo de varios saltos.

Formato:

- **Versión** (4 bits): indica la versión de IP. En nuestro caso siempre será 4.
- **IHL** (4 bits): longitud de la cabecera IP. *Expresado en palabras de 4B
  *Para obtener el tamaño total en Bytes hay que multiplicar IHL por 4.
  $Tam_{mín}$ = 20B $Tam_{máx}$ = 60B
- **Type of Service** (1B): indica el tipo de tráfico que transporta este datagrama
  En nuestro caso siempre es 0
- **Total Length** (2B): longitud total (en B) del datagrama actual.
  Incluye: cabecera + payload (que va detrás de la cabecera)
- **Identification** (2B): identificador del datagrama IP = IPID
  Este campo es útil cuando hay fragmentación IP.
- **Flags** (3 bits): banderas IP
  - Bit 1 (Reservado): siempre a 0
  - Bit 2 (DF): bandera que indica que no debe fragmentarse el datagrama.
    En nuestro caso siempre es 0.
  - Bit 3 (MF): bandera que indica que vienen + fragmentos tras el datagrama actual
    Si se fragmentan todos los fragmentos: último fragmento = -1
- **Offset** (13 bits): indica (en caso de fragmentación) el offset de los datos contenidos en el datagrama actual respecto al total de datos sin fragmentar.
  - Expresado en palabras de 8B. → para obtener el valor real multiplicar por 8
- **Time to Live** (1B): nº máx. saltos IP que puede realizar el datagrama actual antes de ser descartado. Valor: — por defecto = 64
  Cada vez que un paquete atraviesa un salto a nivel IP: decrementa en 1
  y cuando llega a 0: datagrama actual se descarta
- **Protocol** (1B): protocolo de nivel superior encapsulado en el payload del datagrama.
  - 1 → ICMP
  - 6 → TCP
  - 17 → UDP
- **Header Checksum** (2B): suma de verificación calculada sobre la cabecera IP para detectar errores o modificaciones de la cabecera IP durante el envío de datos. Cuando recibimos un datagrama → si: checksum = errónea descartado

- Dir. IP origen (4B): dir. IP del emisor del datagrama actual
- Dir. IP destino (4B): dir. IP del receptor del datagrama
- Opciones (Tamaño variable): opciones que aportan funcionalidades adicionales.
  Tam = múltiplo de 4. $Tam_{mín}$ = 0 B $Tam_{máx}$ = 40 B

$Tam_{datagrama}$: $Tam_{máx}$ que puede enviarse a nivel IP = (cabecera + datos) definido por la MTU (= maximum transmission unit)
↑ depende del protocolo de nivel inferior (en nuestro caso Ethernet)
Ej valor MTU = 1500 B

- Cuando los datos a enviar no caben en un único datagrama IP → fragmentación
  Cada fragmento se envía junto a su propia cabecera IP,
  → valor diferente en Total Length, MF, Offset y checksum

- UDP: protocolo de nivel de transporte no fiable y no orientado a conexión que permite enviar datagramas. UDP se usa normalm. para implementar por encima protocolos de nivel de aplicación no orientados a flujo o que tienen restricciones de tiempo real y en los cuales el concepto de retransmisión no tiene sentido. Ej: protocolos de nivel de aplicación DNS, DHCP, RTP usan a UDP como nivel de transporte

- Campos cabecera:
  Source Port (2B): puerto origen del datagrama UDP
  Destination Port (2B): " destino
  Length (2B): $Tam_{datagrama\ UDP}$ = cabecera + datos (B)
  Checksum (2B): suma de comprobación sobre la pseudocabecera IP. Su uso es opcional
  ↑ 0

para comprobar la conectividad a nivel IP con equipos y estimar su RTT ← Round-Trip Time

- ICMP: protocolo de control, diagnóstico y notificación de errores relacionado estrechamente con IP. Aunque funciona por encima de IP, no es un protocolo de transporte ya que no se usa para intercambiar info de extremo a extremo entre sistemas o aplicaciones.

- Cabecera: parte fija (32 bits) + parte variable que depende del tipo de mensaje
  - Type (1B): tipo mensaje ICMP. 8: Echo Request; 0: Echo Reply
  - Code (1B): subtipo mensaje ICMP. 0 para ambos
  - Checksum (2B): suma de comprobación. Tiene que ser par (sino se añade un B a 0)
  - Identifier (2B): relaciona Echo Request con Echo Reply
  - Sequence Number (2B): n-secuencia para correlar preguntas y respuestas ↓ incrementa en 1 por cada request

ICMP Echo Request / Reply

- Dir. IP origen (4B): dir. IP del emisor del datagrama actual
- Dir. IP destino (4B): dir. IP del receptor del datagrama
- Opciones (Tamaño variable): opciones que aportan funcionalidades adicionales.
  Tam = múltiplo de 4. $Tam_{mín}$ = 0 B $Tam_{máx}$ = 40 B

$Tam_{datagrama}$: $Tam_{máx}$ que puede enviarse a nivel IP = (cabecera + datos) definido por la MTU (= maximum transmission unit)
↑ depende del protocolo de nivel inferior (en nuestro caso Ethernet)
Ej valor MTU = 1500 B

- Cuando los datos a enviar no caben en un único datagrama IP → fragmentación
  Cada fragmento se envía junto a su propia cabecera IP,
  ↳ valor diferente en Total Length, MF, Offset y checksum

- UDP: protocolo de nivel de transporte no fiable y no orientado a conexión que permite enviar datagramas. UDP se usa normalm. para implementar por encima protocolos de nivel de aplicación no orientados a flujo o que tienen restricciones de tiempo real y en los cuales el concepto de retransmisión no tiene sentido. Ej: protocolos de nivel de aplicación DNS, DHCP, RTP usan a UDP como nivel de transporte

- Campos cabecera:
  Source Port (2B): puerto origen del datagrama UDP
  Destination Port (2B): " destino
  Length (2B): $Tam_{datagrama\ UDP}$ = cabecera + datos (B)
  Checksum (2B): suma de comprobación sobre la pseudocabecera IP. Su uso es opcional
  ↑ 0

para comprobar la conectividad a nivel IP con equipos y estimar su RTT (Round-Trip Time)

- ICMP: protocolo de control, diagnóstico y notificación de errores relacionado estrechamente con IP. Aunque funciona por encima de IP, no es un protocolo de transporte pq no se usa para intercambiar info de extremo a extremo entre sistemas o aplicaciones.

necesario en! por cada request

- Cabecera: parte fija (32 bits) + parte variable que depende del tipo de mensaje
  - Secuencia Number (2B) → relacionar para conocer fragmentos y respuesta
  - Type (1B): tipo mensaje ICMP. 8: Echo Request, 0: Echo Reply
  - Code (1B): subtipo mensaje ICMP. 0 para ambos
  - Checksum (2B): suma de comprobación. Tiene que ser par (sino se añade un B a 0)
  - Identifier (2B): relaciona Echo Request con Echo Reply

ICMP Echo → Request / Reply

ip.py

checksum(msg)

getMTU(interface)

getNetmask(interface)

getDefaultGW(interface)

process_IP_datagram(us, header, data, srcMac

registerIPProtocol(callback, protocol)

initIP(interface, opts=None)

sendIPDatagram(dstIP, data, protocol)

udp.py