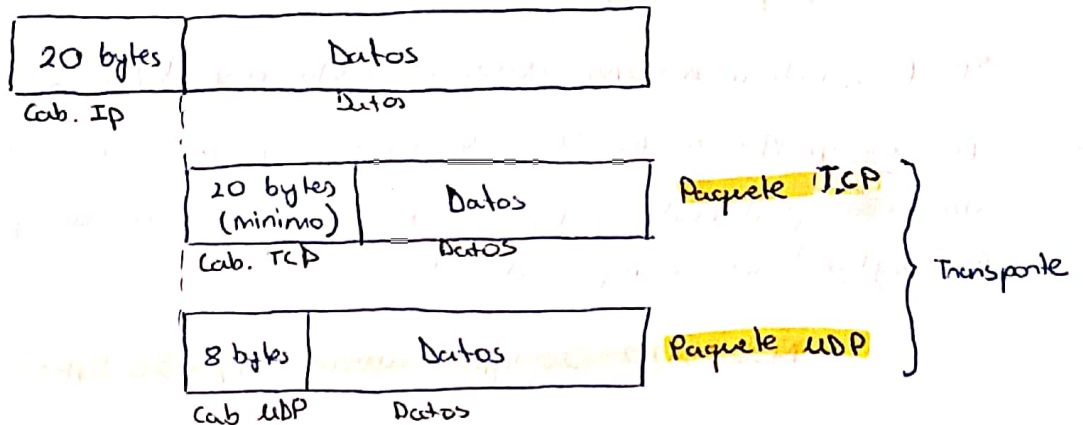


## Protocolos de transporte

### TCP y UDP

TCP: orientado a conexión, muy fiable y bastante tráfico adicional en red  
UDP: no orientado a conexión, poco fiable y con poca carga adicional.

### Encapsulación



### • Cabecera TCP:

- Puerto fuente y puerto destino: 16 bits correspondiente a los identificadores de los puertos.

- N° Secuencia: primer byte del campo de datos

- N° ACK: bytes del campo de datos próximo paquete

- Len: (4 bits), → reservado

- Flags: 6 bits con significado propio:

• ACK → (vale 1) indica que el N° ACK es válido y debe interpretarse

• PSH (push) → (vale 1) la capa de transporte debe pasar los datos a capa aplicación sin esperar más paquetes

• RST → Rechazo en la ~~secuencia~~ de conexión. (falla inicio conexión o rechazan paquetes no válidos)

• SYN (synchronise) → solicitan establecimiento de una conexión

• FIN (Finish) → solicitan liberación de una conexión.

- Ventana: n° de bytes que el emisor puede recibir en su buffer. si vale 0 no puede recibir datos (sin contar los flags)

- SUM: Suma de verificación aplicada a cabecera y datos TCP

- Puerto urgente: info urgente desplazada en el campo de datos y que se ha de enviar directamente a la capa de aplicación  
- Opciones: Permite campos adicionales.

## • Cabecera UDP:

- Puerto fuente y puerto destino: 16 bits correspondientes a los puertos de nivel de transporte.
- Longitud: n: total de bytes (cabecera + datos) antes de ser fragmentado en paquetes IP
- SVT: suma de verificación de cabecera y datos UDP

## Segmentación paquetes TCP

MSS (cantidad máxima datos en cada paquete) es un valor variable que se ajusta a la MTU de ambos equipos que intentan establecer conexión pudiendo evitar la fragmentación de los paquetes reduciendo los bytes del campo 'datos'.

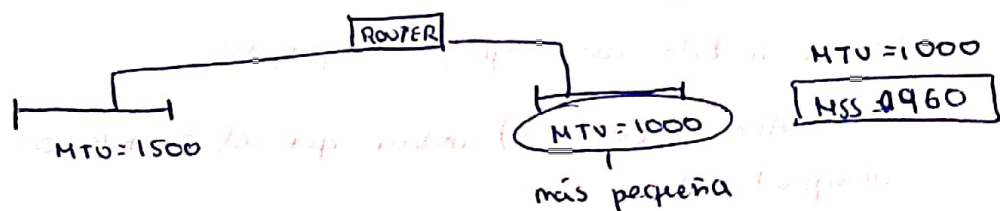
$$MSS = MTU - 20 \text{ bytes (cabecera TCP)} - 20 \text{ bytes (cabecera IP)}$$

Este valor es informado por cada extremo de la comunicación durante el establecimiento de la conexión (campo opciones paquetes SYN).

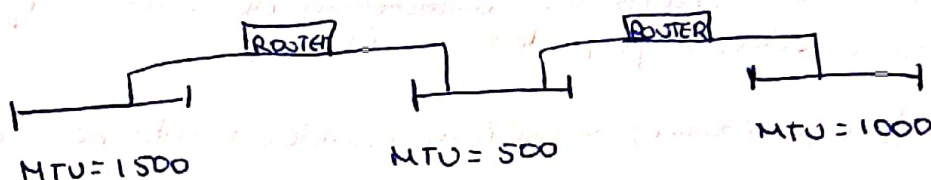
## Norma RFC 1191

Objetivo: Evitar que los datagramas IP que contienen paquetes TCP sean fragmentados.

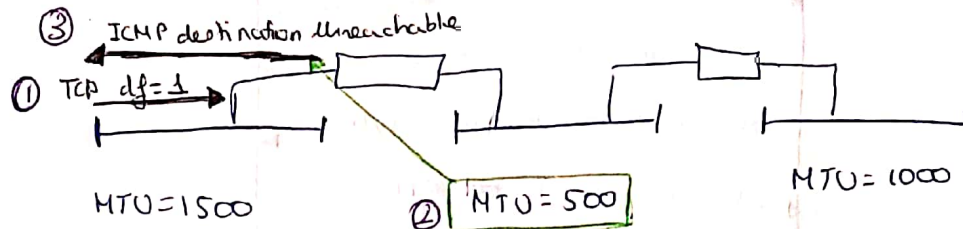
Cuando 2 equipos establecen conexión TCP, determinan el valor MSS al menor de 2 estaciones:



Sin embargo, si existe una red intermedia con un MTU aún menor al de ambos extremos:



si se fragmentasen se reduciría la fluidez de la conexión. Para ello, la norma establece que en los extremos intercambien paquetes TCP en los que en la cabecera IP esté **activo el bit don't fragment** y que pueda **interpretar paquetes ICMP fragmentation needed and the bit don't fragment was set**.

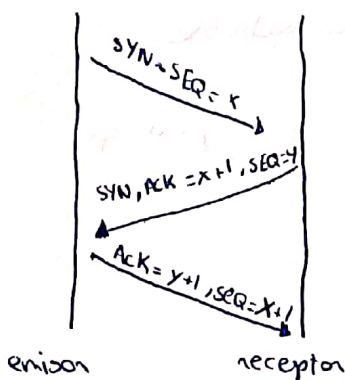


- ① Manda paquete TCP con bit don't fragment = 1 para no ser fragmentado
- ② No puede pasar por la red pero cuando el MTU = 500
- ③ El router responde con un ICMP destination unreachable con código fragmentation needed and bit don't fragment was set. En la cabecera ICMP se indica el valor de la MTU (campo Next Hop MTU).
- ④ La estación origen determina el nuevo MSS de la conexión y reenvía el paquete TCP que no llegó al destino (nuevo valor MSS)

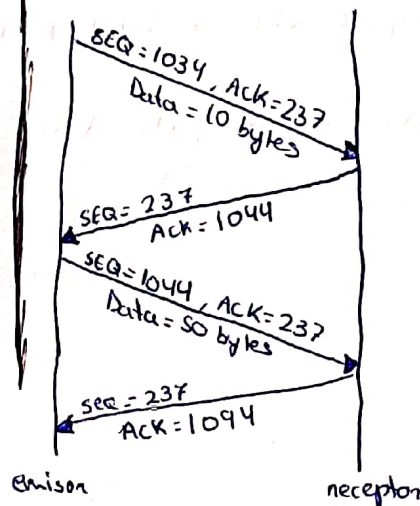
### Control del flujo con TCP

Número de secuencia (SEQ)  
Número de acuse de recibo (ACK)

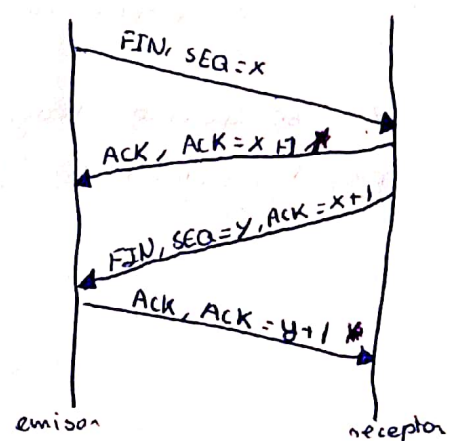
#### Apertura de conexión



#### Transmisión de datos



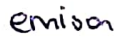
#### Cierre de conexión



\* Confirmaciones de cierre.



Calc:



receptor

## Observaciones

\* Cuando se intenta acceder a un puerto no soportado puede ocurrir:

- se envía un paquete TCP  $\rightarrow$  RESET / RST
- se envía un paquete UDP  $\rightarrow$  ICMP Port unreachable

\* Si se quiere fragmentar un paquete UDP se hará igual que los paquetes ICMP (ver 2)

## Cadenencia eficaz

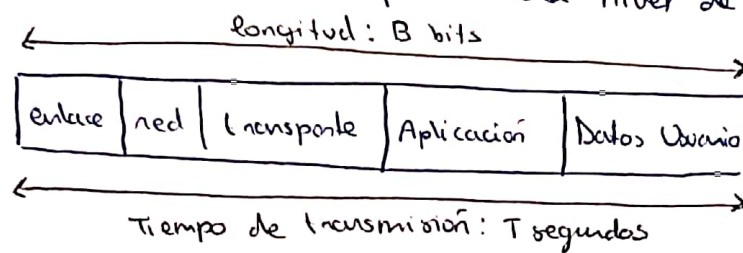
velocidad real en bits por segundo de envío de datos, inferior a la  $V_t$  (velocidad de transmisión)

$$C_e = \frac{n}{T_o} = (\text{bps})$$

$n = n^\circ$  de bits transmitidos

$T_o =$  tiempo en s que tarda un paquete en llegar.

Se puede calcular solo para protocolos de comunicación que en los que no se hayan producido errores ni reenvíos de paquetes. La cadencia eficaz es distinta para cada nivel de arquitectura:



• Para cada nivel:

$$C_e - \text{enlace} = (B - \text{lon. cab. enlace}) / T = \text{bps}$$

$$C_e - \text{red} = (B - \text{lon. cab. enlace} - \text{lon. cab. red}) / T = \text{bps}$$

$$C_e - \text{transp} = (B - \text{lon. cab. enl} - \text{lon. cab. red} - \text{lon. cab. transp}) / T = \text{bps}$$

## Velocidad de transmisión

$$V_t = B / T = \text{bps}$$