

Redes II

TCP



luis marrone

- Orientado a conexión.
- Confiable.
- Ordenado.
- Byte Stream.
- Ventana.
- Control de flujo y congestión.

TCP – Estructura

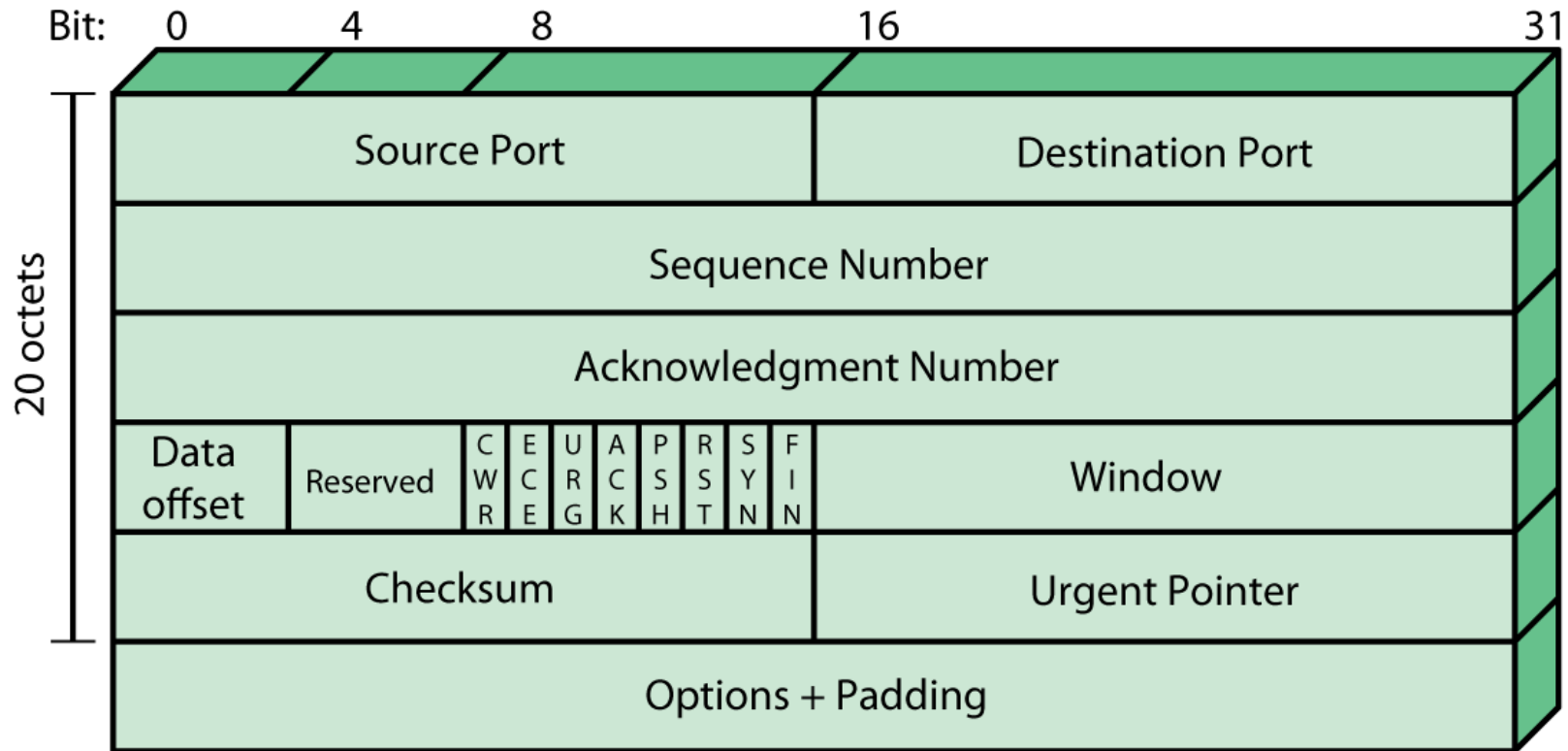
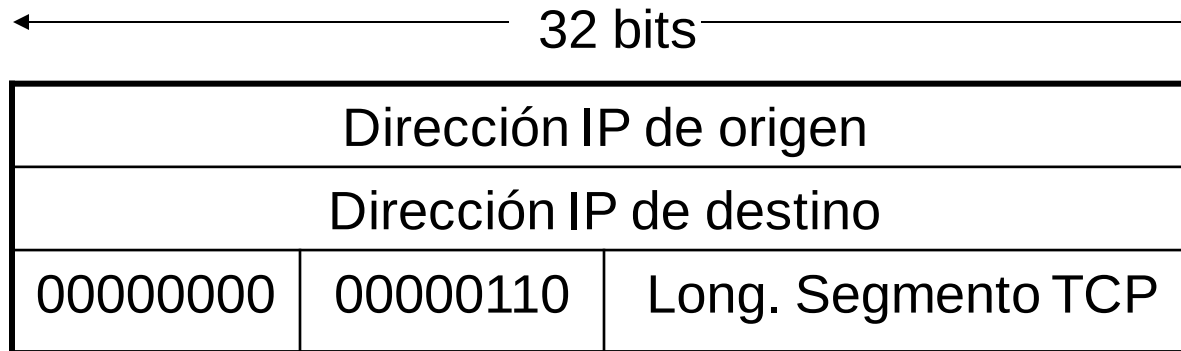


Figure 22.10 TCP Header

La pseudocabecera TCP



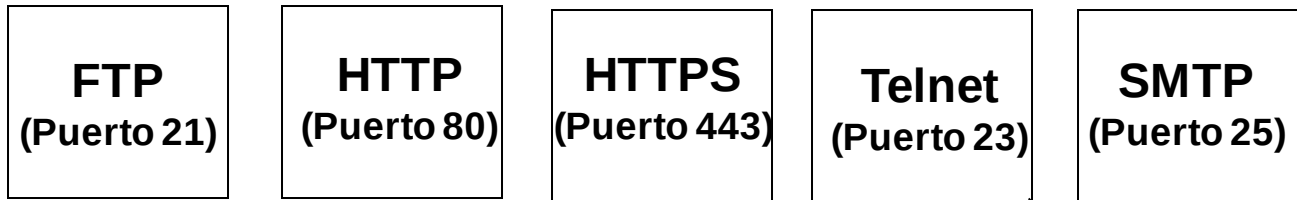
Se añade al principio del segmento solo para el cálculo del Checksum, **no se envía**. Permite a TCP comprobar que IP no se ha equivocado (ni le ha engañado) en la entrega del segmento.

El valor $110_2 = 6_{10}$ indica que el protocolo de transporte es TCP

Multiplexación

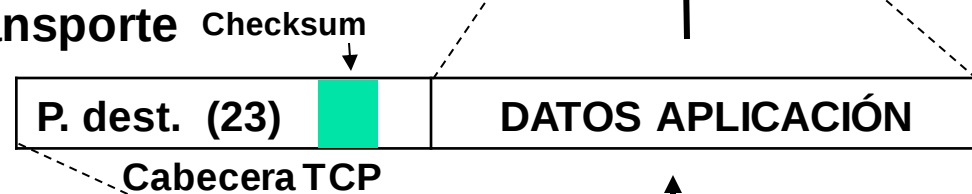
Múltiples instancias
(una o varias por
protocolo)

Nivel de aplicación



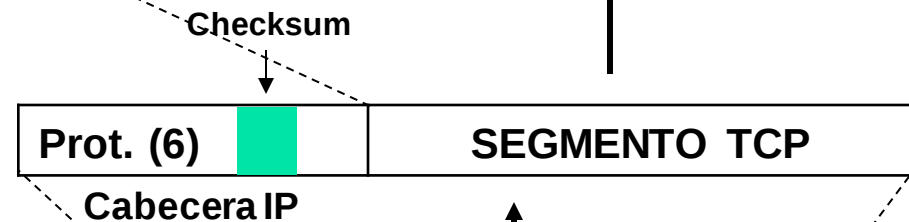
Dos instancias
(TCP y UDP)

Nivel de transporte



Una instancia IP
(puede haber otros
protocolos)

Nivel de red

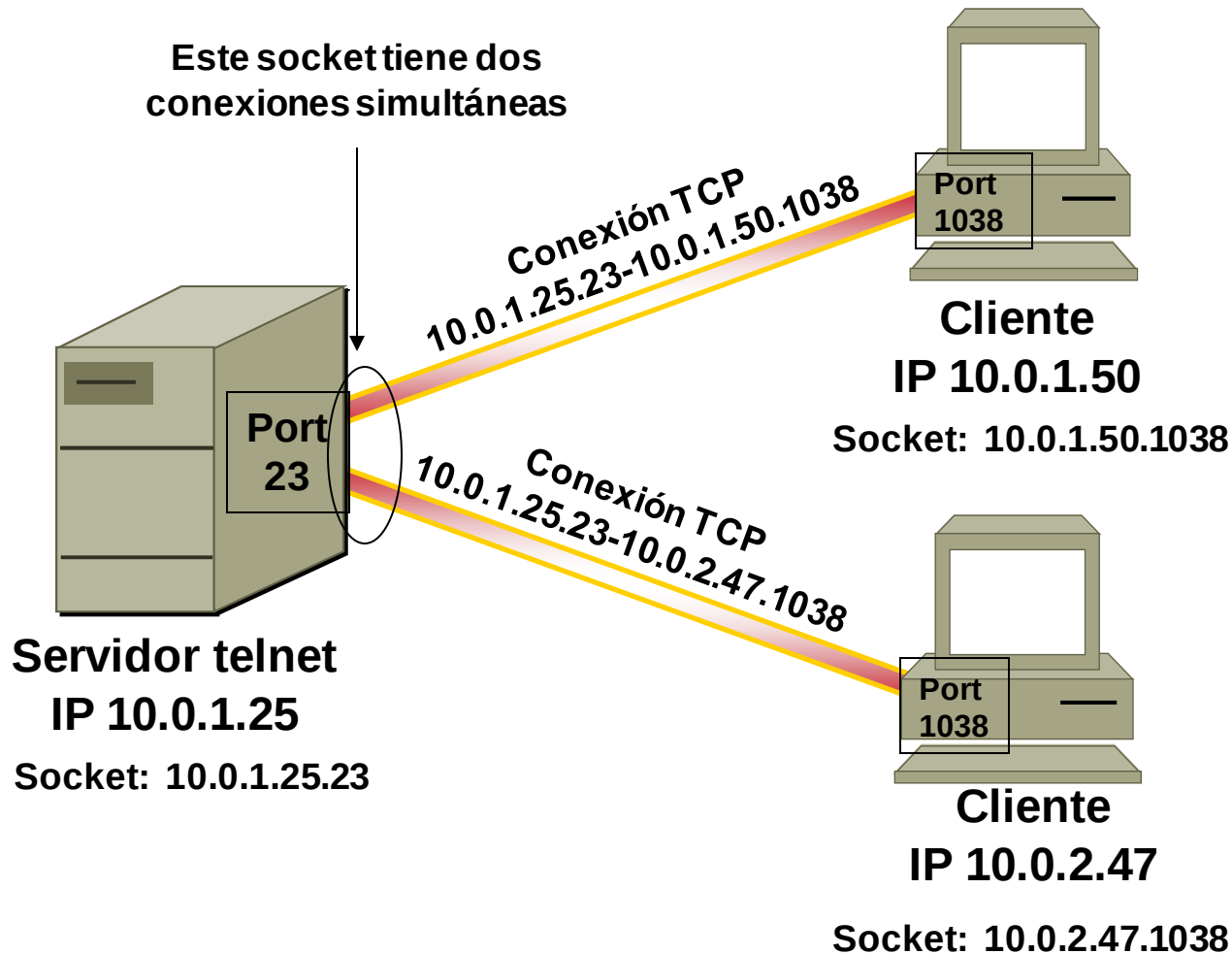


Múltiples instancias
(una por interfaz)

Nivel de enlace

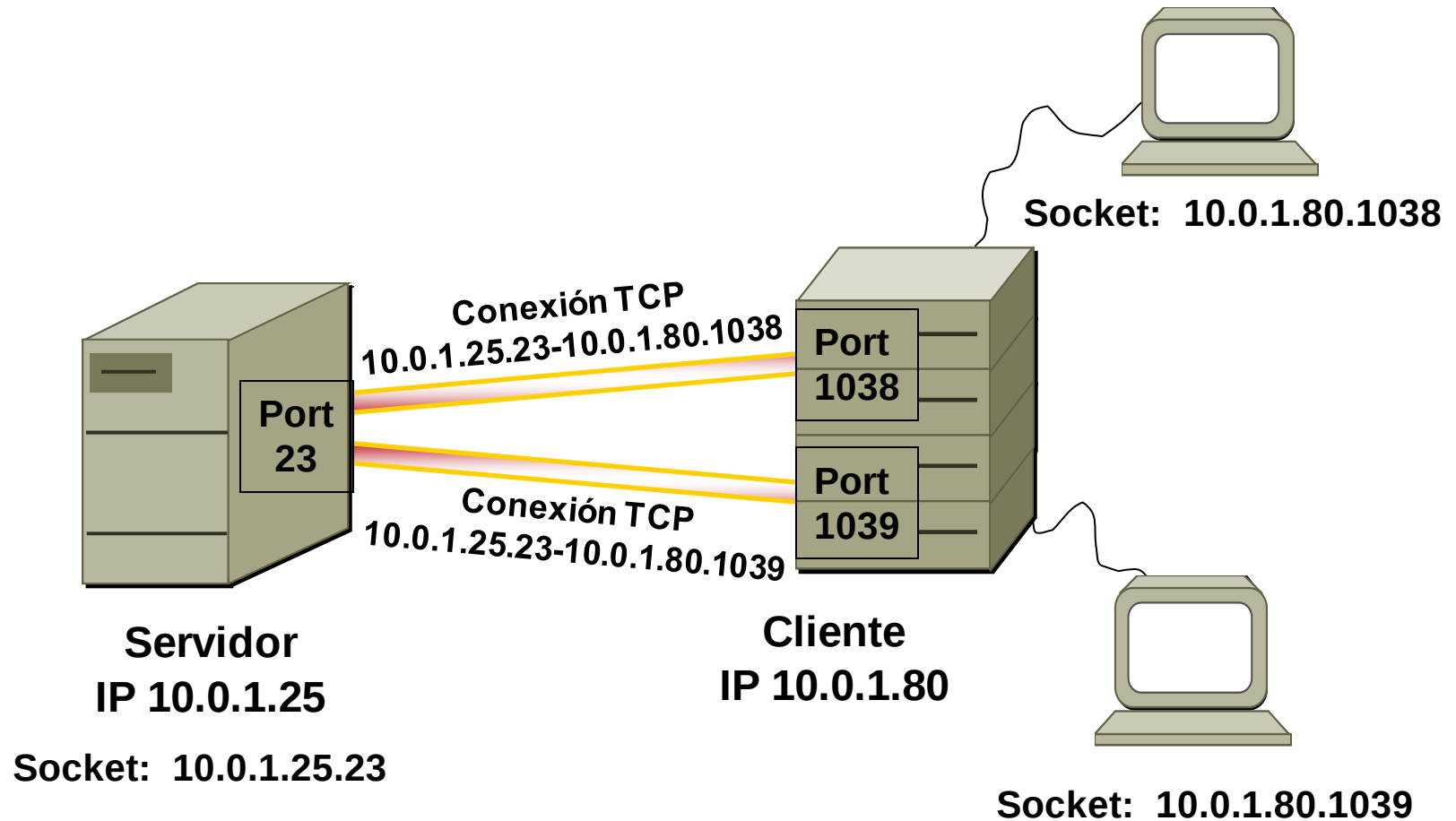


Dos conexiones TCP a un mismo socket desde dos sockets con el mismo número de puerto



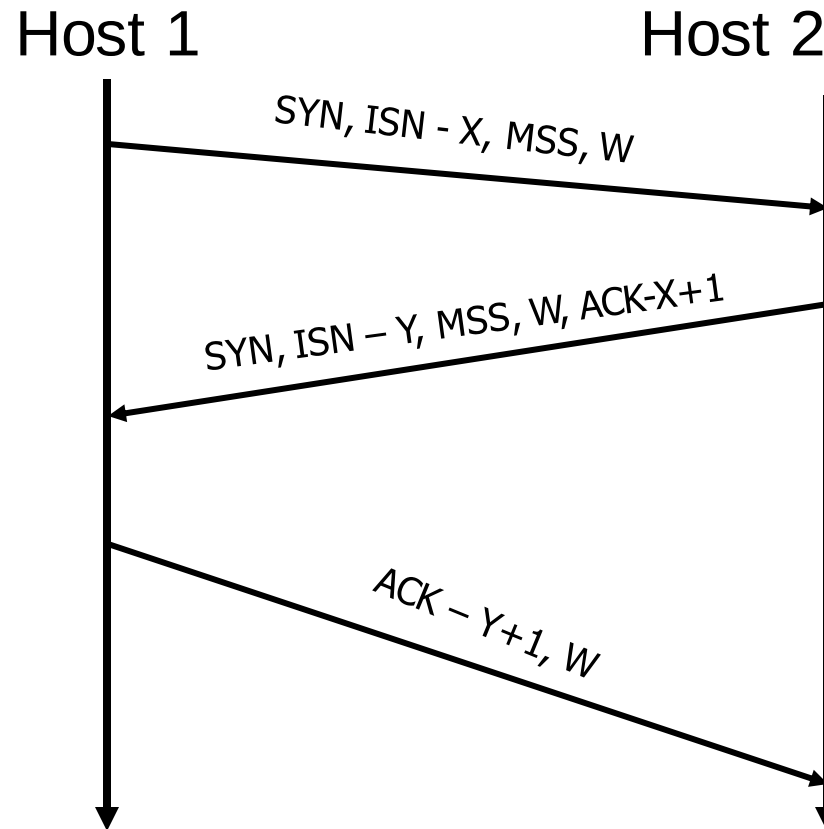
Dos conexiones TCP a un mismo socket desde dos sockets con la misma dirección IP

LINTI - UNLP



TCP – Establecimiento de la conexión

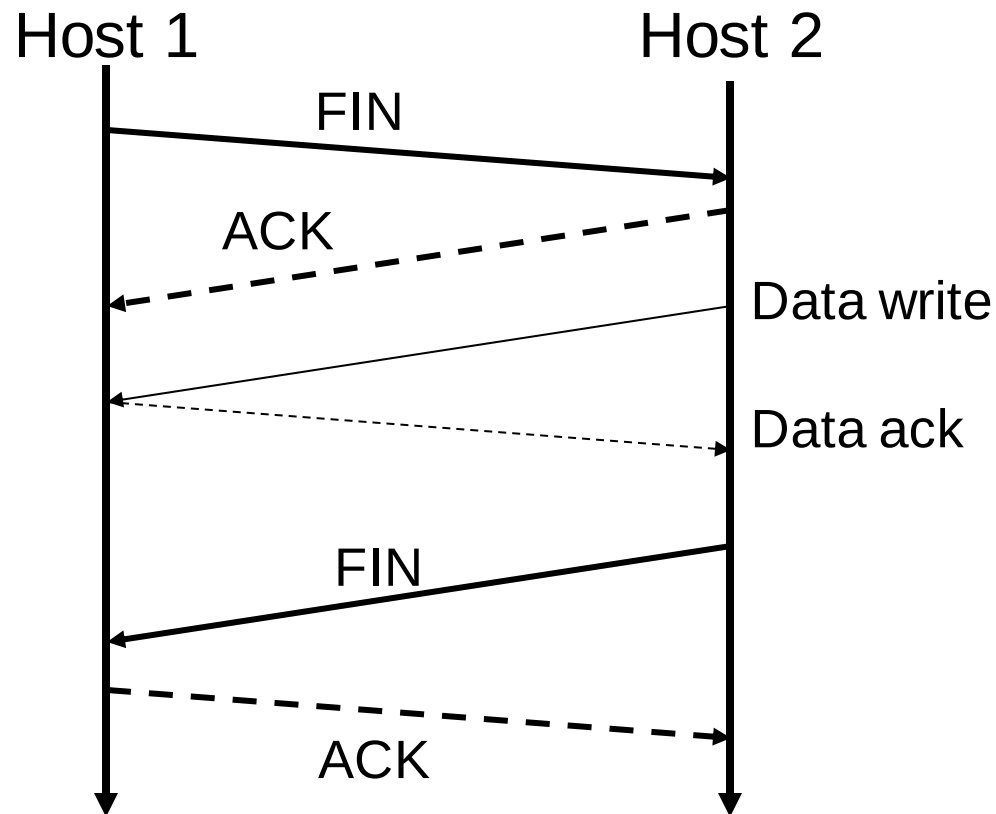
- Se intercambian 3 segmentos.
- Se envía el MSS.



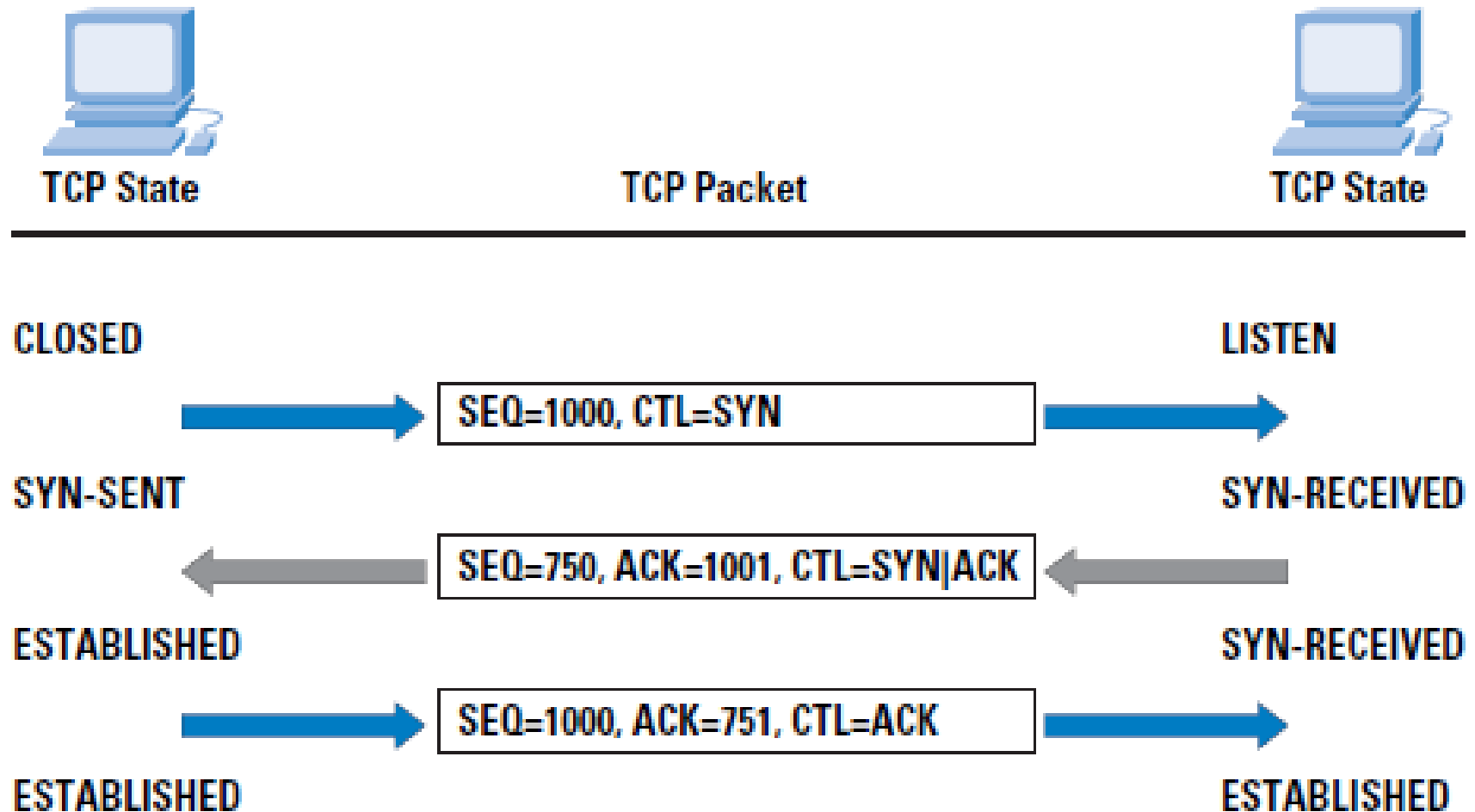
- Máxima longitud del segmento.
- Default = 536 bytes.
- Diferente para cada sentido de la conexión.
- Vinculado al MTU.

TCP – Fin de la conexión

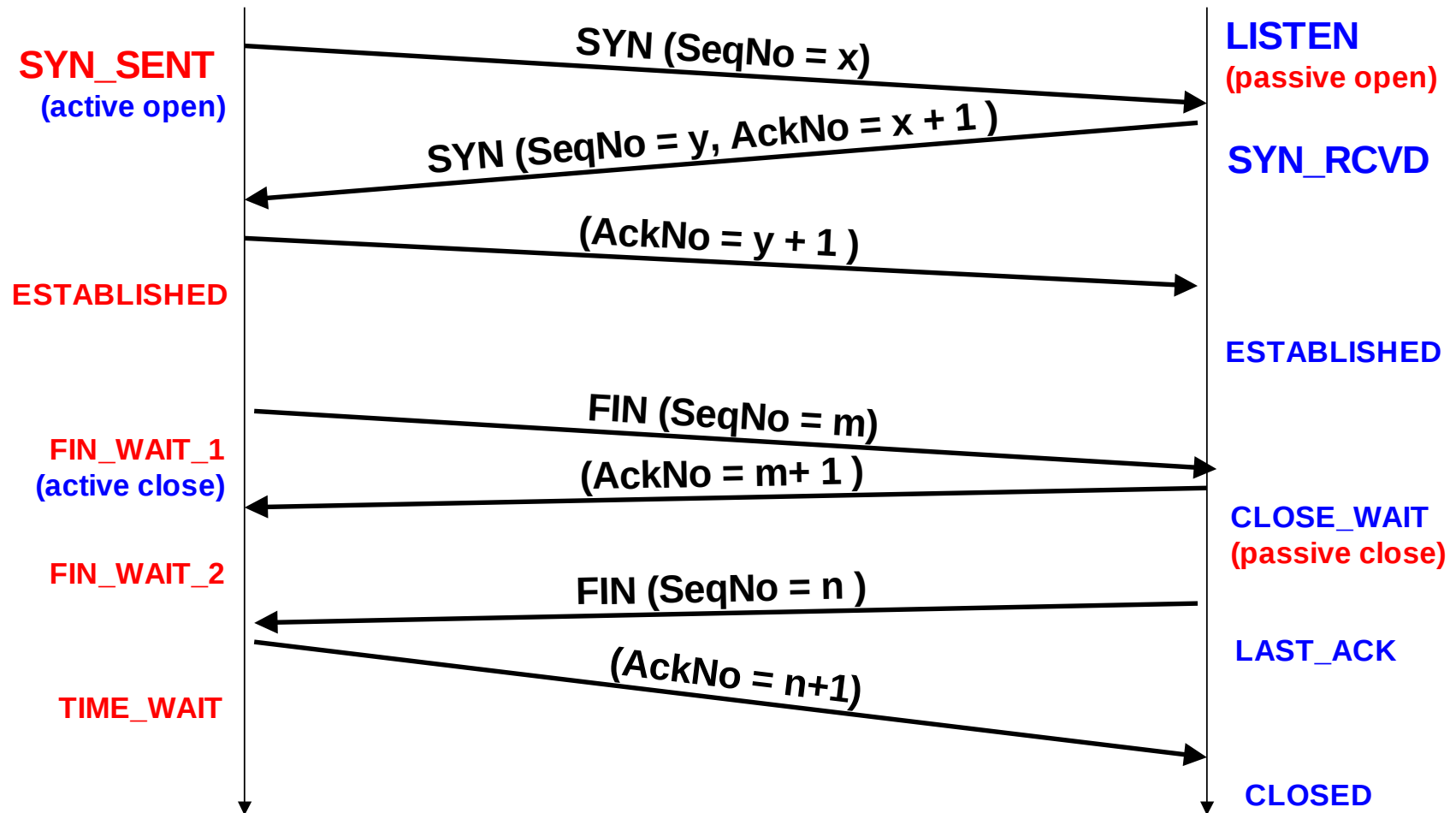
- Se intercambian 4 segmentos



SYN



SYN – FIN



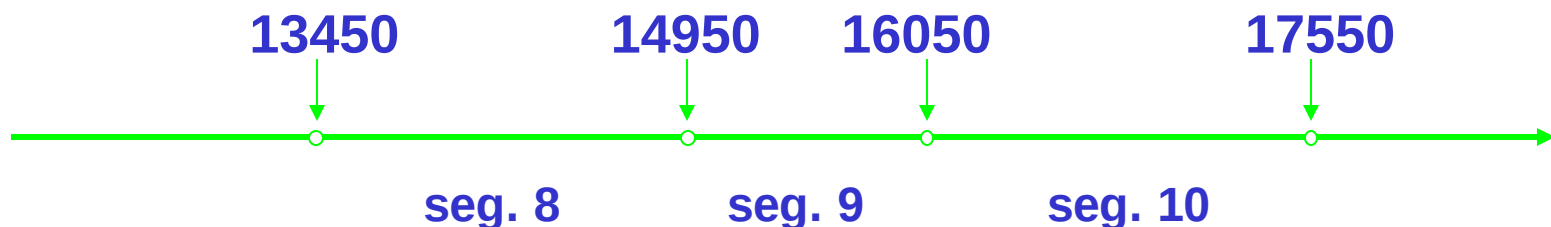
```
U:\>netstat -a -n -p TCP
```

Active Connections

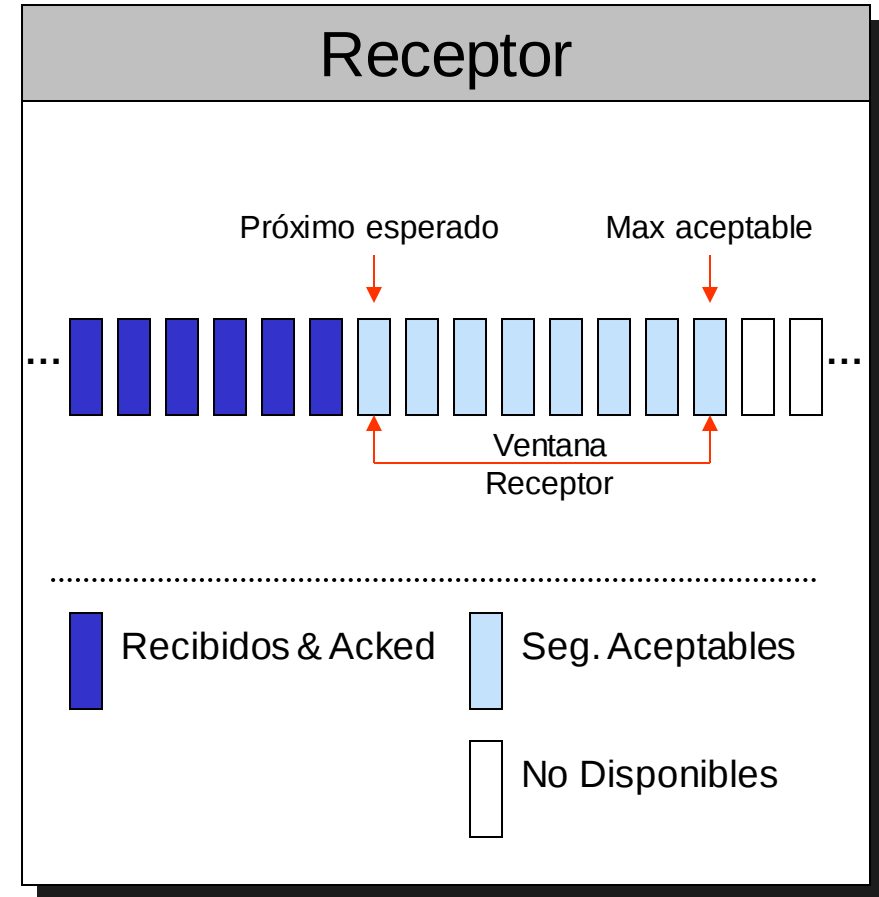
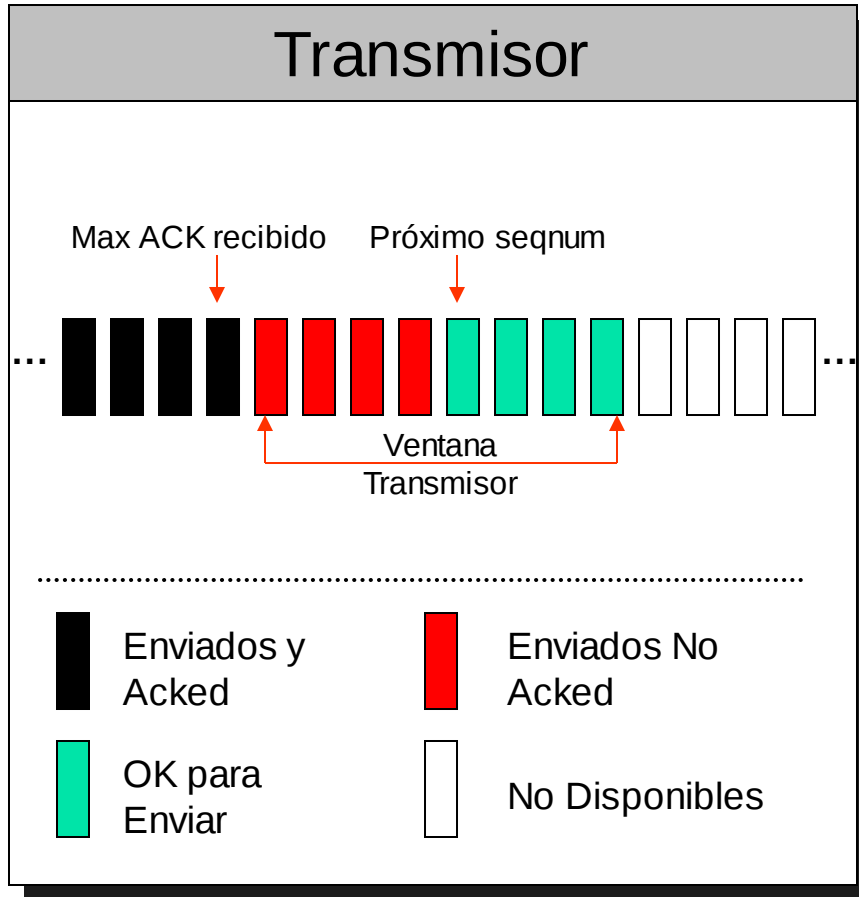
Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:12345	0.0.0.0:0	LISTENING
TCP	10.10.2.38:139	0.0.0.0:0	LISTENING
TCP	10.10.2.38:1132	0.0.0.0:0	LISTENING
TCP	10.10.2.38:1132	10.10.6.170:139	ESTABLISHED
TCP	10.10.2.38:1200	10.10.6.99:1027	ESTABLISHED
TCP	10.10.2.38:1203	10.10.6.105:1306	ESTABLISHED
TCP	10.10.2.38:1207	0.0.0.0:0	LISTENING
TCP	10.10.2.38:1207	10.10.3.228:139	ESTABLISHED

TCP – Número de secuencia

- Número de 32 bits.
- Valor inicial fijado en el establecimiento de la conexión, (ISN).
- TCP desagrega el stream de bytes en segmentos.
 - Cada segmento tiene asociado el número de secuencia.
 - Indica su ubicación en el stream de bytes.

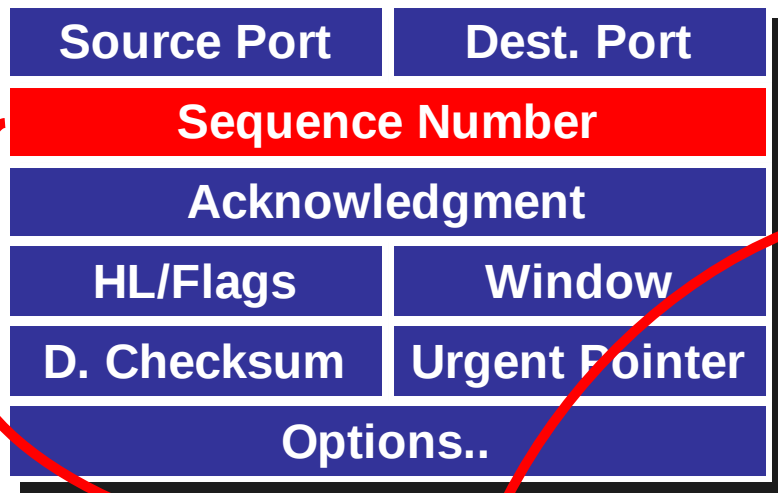


TCP – Ventana deslizante

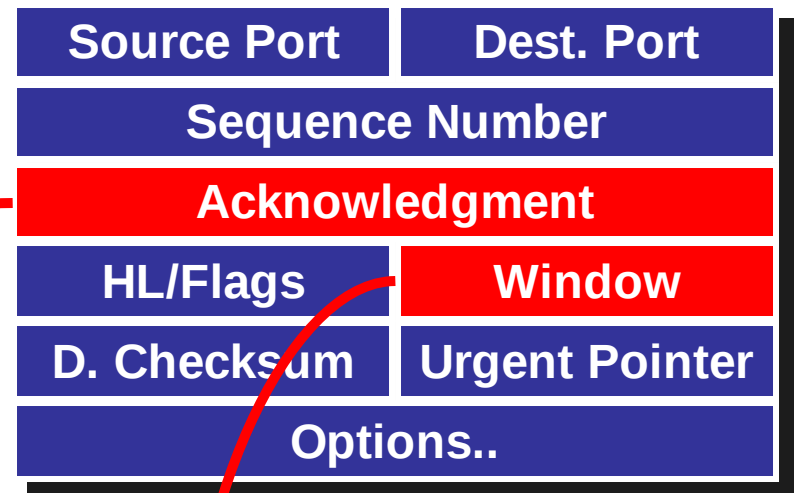


TCP – Ventana deslizable

Segmento enviado



Segmento recibido



acknowledged enviados a enviar

Timeout y RTT

■ Problema:

- El RTT varía sustancialmente
- Demasiado largo => subutilización
- Demasiado corto => retransmisiones inútiles.

■ Solución:

- Timeout adaptable, estimando el RTT.

Estimando el RTT

- Chebyshev's Theorem
- $\text{Max RTT} = \text{Avg RTT} + k * \text{Desv}$
- $\text{Avg RTT} = (1-x) * \text{Avg RTT} + x * \text{Muestra RTT}$, $x = 0.1$
- Probabilidad de error $< 1/(k^2)$
- Válida para toda distribución de muestras.

$$\text{Timeout} = \text{AverageRTT} + 4 * \text{Deviation}$$

Fijando el Timeout

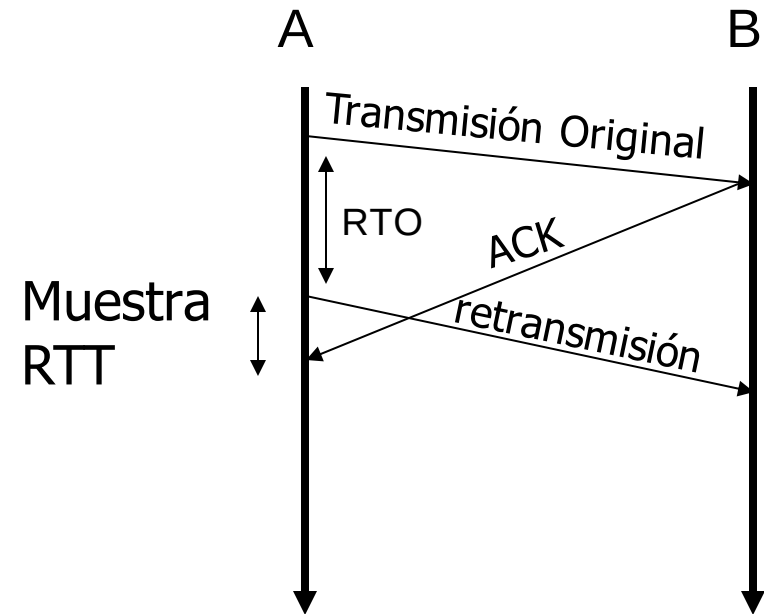
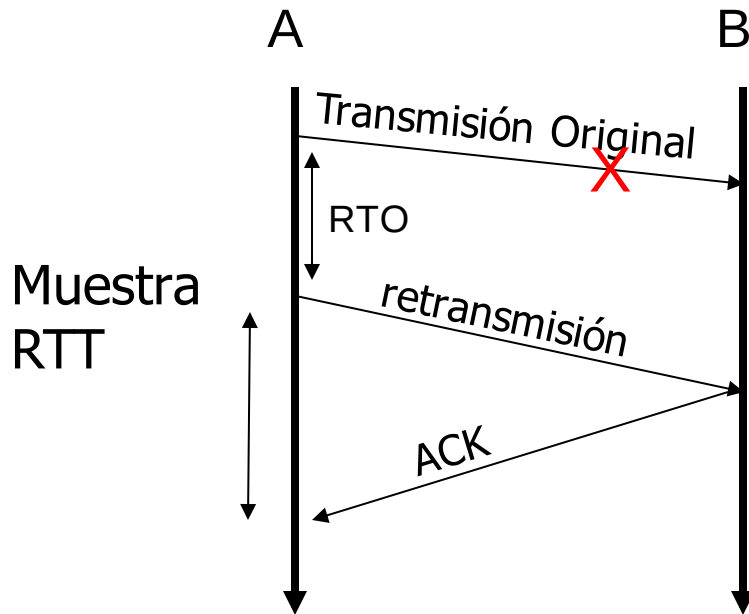
$$\text{Timeout} = \text{Avg RTT} + 4 * \text{Desv}$$

$$\text{Desv} = (1-x) * \text{Desv} +$$

$$x * | \text{Muestra RTT} - \text{Avg RTT} |$$

- Se suele fijar en múltiplos de 200,500,1000mseg.

Ambigüedades



Estimador de Karn

- Ante una retransmisión:
 - No actualizar los RTT
 - Timer backoff
 - $RTO = 2 * RTO$
- Volver a estimar después de una transmisión exitosa.
- Las muestras del RTT se toman en base a la opción de Time Stamp de TCP.

