Índice hora 2

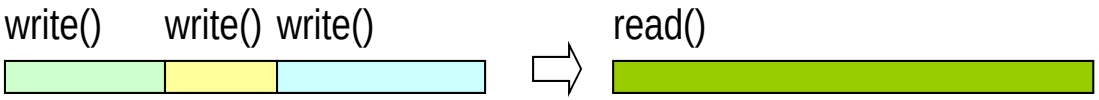
Objetivos

- Presentar el protocolo de transporte TCP
- Entender las funcionalidades ofrecidas por TCP
- Revisar en detalle la cabecera básica y opciones de TCP

4 TCP

RFC768(STD6)

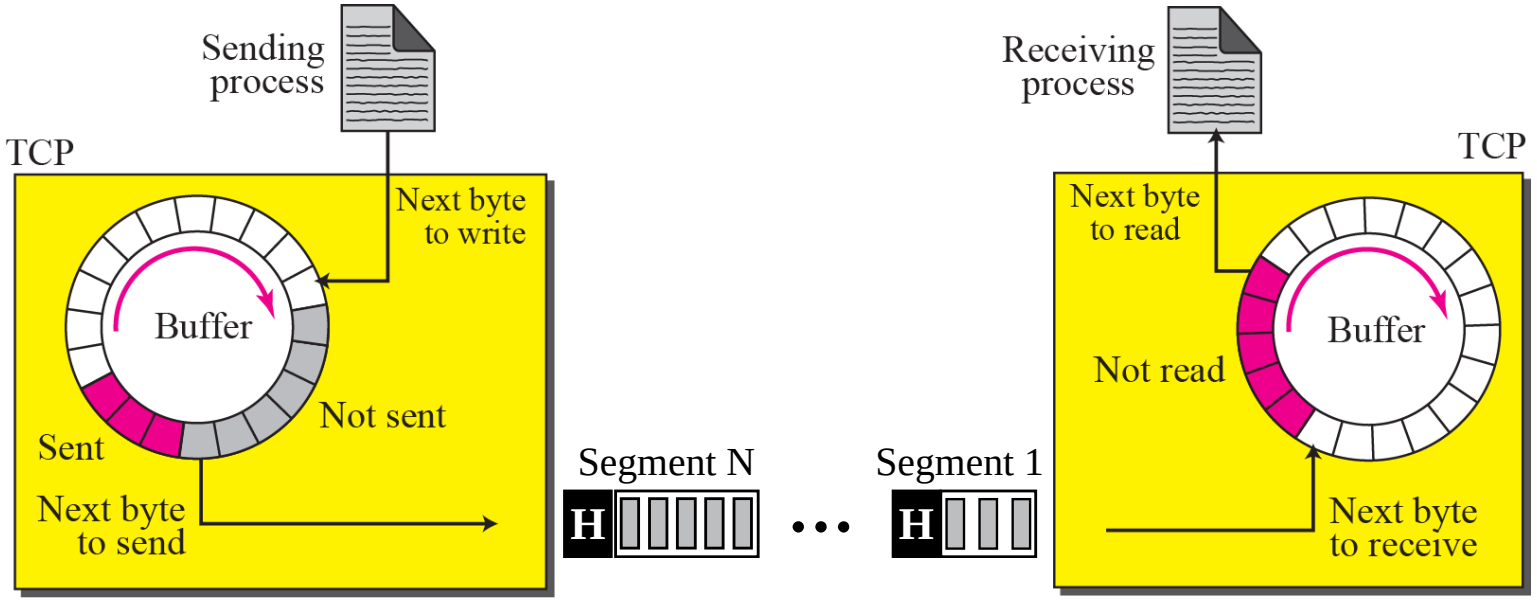
- Transmission Control Protocol.
- TCP es un protocolo de nivel de transporte encapsulado por encima de IP (campo protocol=6 de la cabecera IP).
- Ofrece un servicio orientado a conexión, de forma que se establece un circuito virtual entre extremos. Fases necesarias:
  - establecimiento de la conexión.
  - transferencia de datos.
  - finalización de la conexión.
- Permite el intercambio entre extremos de un *stream* de bytes, sin poder marcar mensajes, de forma bidireccional (full-duplex).



- Fiabilidad: garantiza la correcta recepción de los datos mediante mecanismos de confirmación y retransmisión.

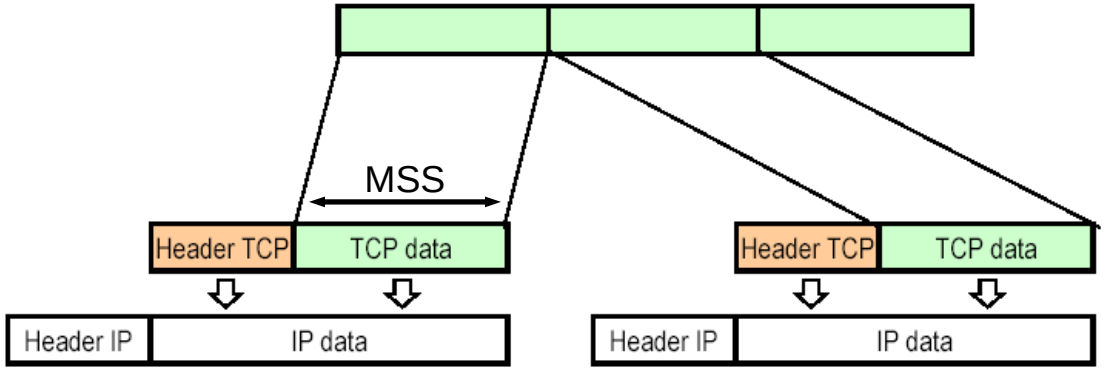
TCP

- *stream* de bytes: buffers en emisión y recepción, para cada sentido



Características TCP

- TCP divide los datos en bloques de tamaño óptimo para ser transmitidos por la red y llamados segmentos.
  - MSS (Maximum Segment Size) será el tamaño máximo de datos del segmento TCP de tamaño óptimo que estará relacionado con la MTU del camino y evitar así fragmentación.
  - Si no hay más datos para enviar, TCP podrá enviar segmentos pequeños.
  - Cada segmento se encapsula en un paquete IP manteniendo la cabecera TCP.

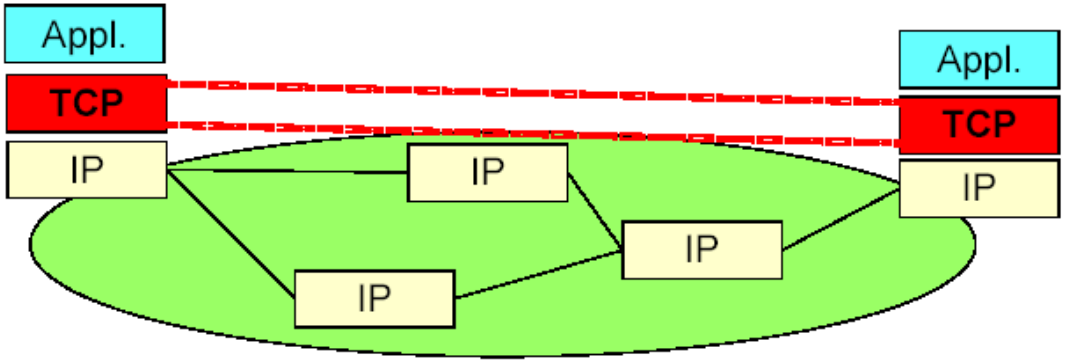


Características TCP

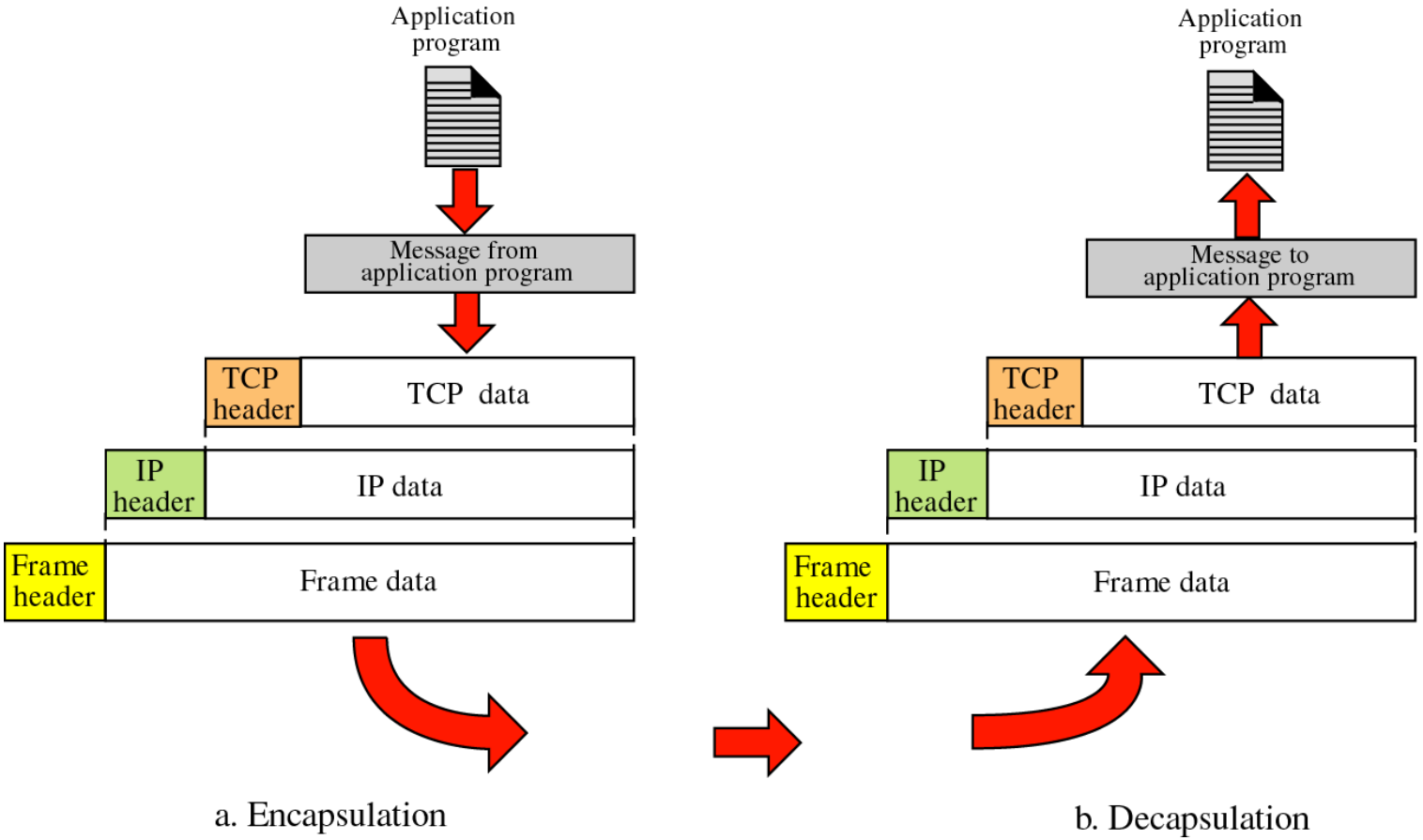
- Incorpora mecanismos de:
  - Control de flujo: evitar saturar al receptor (ventana deslizante).
  - Control de congestión: evitar saturar la red (ventana de congestión, umbral de slow-start).
- Sin embargo no incorpora (por defecto) otros mecanismos como:
  - SACK (Selective ACK): poder confirmar segmentos de datos aislados o intermedios. Es opcional.
  - NACK (Negatively ACK): mandar aviso de que un segmento de datos se ha recibido mal o no se ha recibido. No lo implementa.
- Multiplexación por puerto
  - Se denomina **conexión** a (IP_origen, puerto_origen, IP_destino, puerto_destino, protocolo) = (socket_origen, socket_destino, protocolo)
  - Estos 4 campos identifican unívocamente a todos los segmentos TCP pertenecientes a la misma conexión.

Funcionalidades TCP

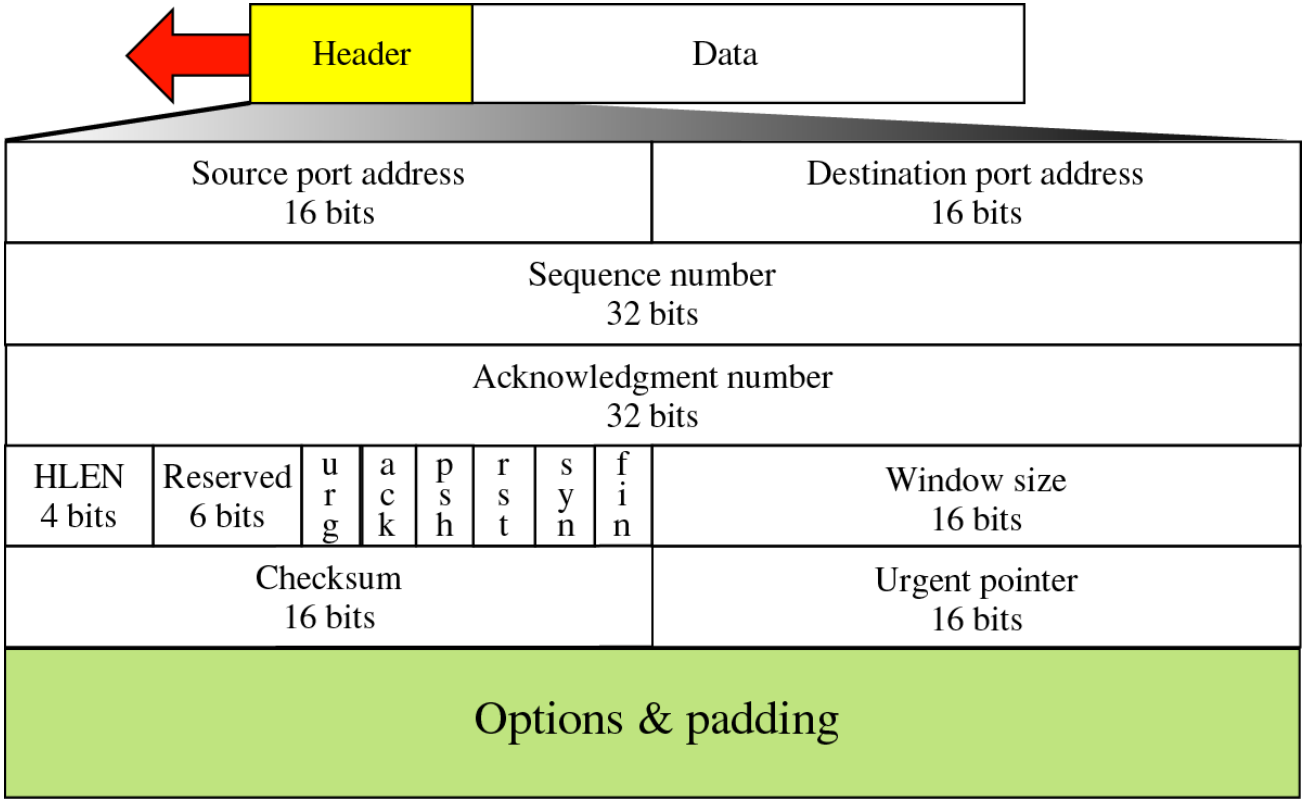
- Multiplexación de aplicaciones (puertos).
- Recuperación de errores (confirmaciones y retransmisiones).
- Reordenación (números de secuencia).
- Control de flujo.
- Control de congestión.



TCP encapsulación



4.1 Cabecera TCP

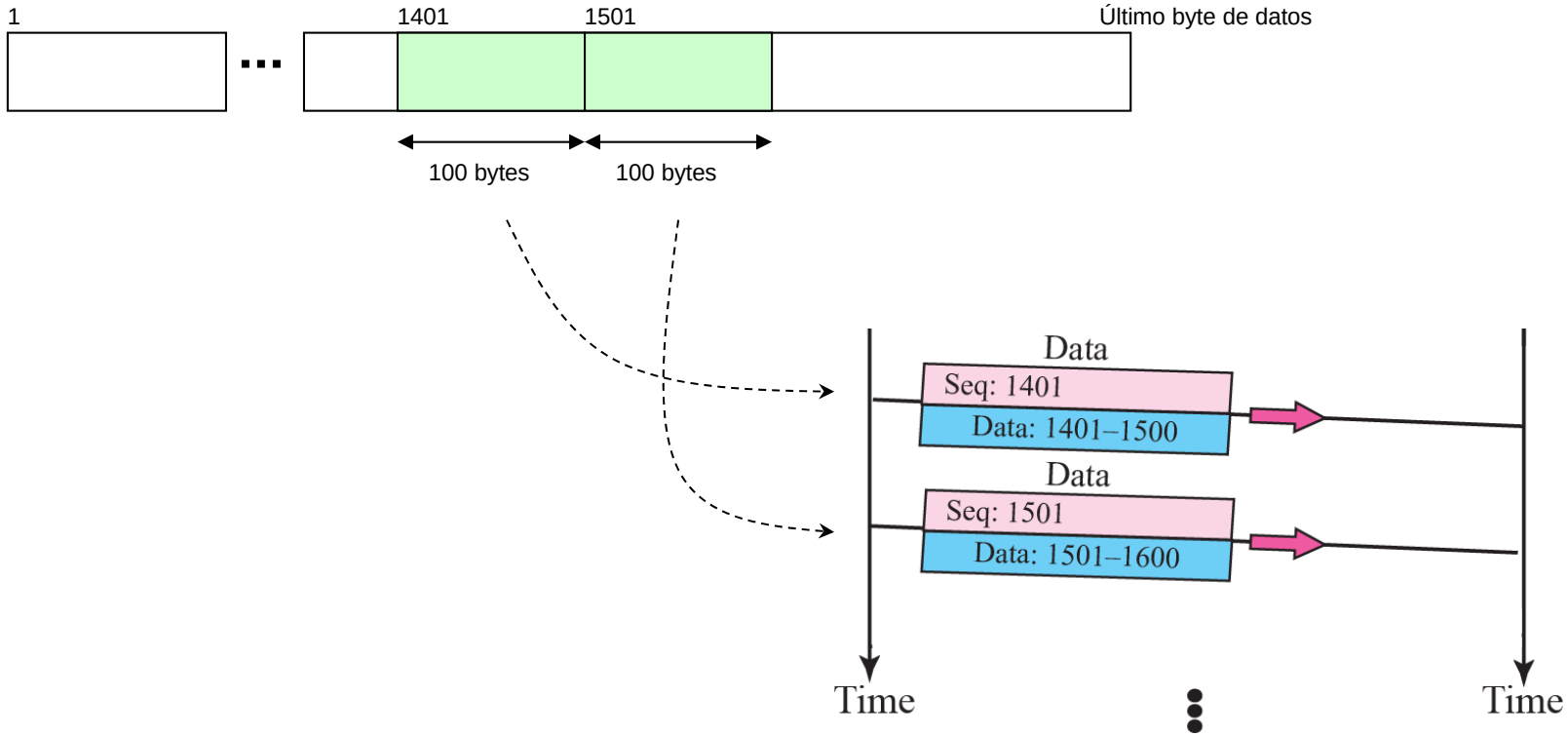


Cabecera TCP

- Puerto origen/destino (16bits)
- Numero de secuencia (32bits)
  - Identifica el byte del stream que representa el primer byte de datos de este segmento. TCP numera cada byte de datos transmitidos con un número de secuencia.
  - El contador se reinicia a 0 después de alcanzar el máximo de 2^{32}-1.
  - Al establecer la conexión, en el segmento con el flag SYN activado este campo contiene el nº de secuencia inicial (ISN, Initial Sequence Number) que la máquina elige de un contador de 32 bits que se incrementa cada 4 μs. Esto permite no confundir segmentos antiguos de una conexión pasada como de esta.
  - Este segmento de SYN consume un nº de secuencia por lo que el primer byte de datos reales comienza en ISN+1. El segmento de FIN también consumirá un nº de secuencia.

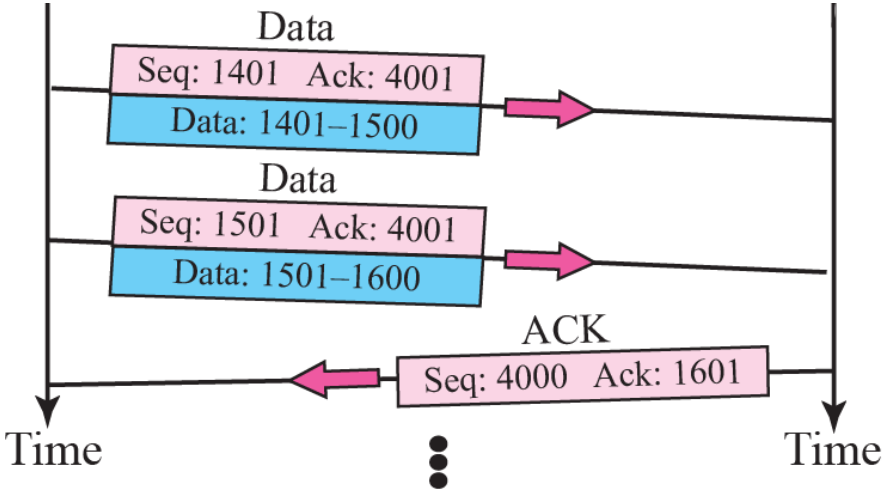
Cabecera TCP

Datos TCP - segmentación



Cabecera TCP

- Número de confirmación (32bits)
  - Este campo es válido sólo si el flag ACK está activado.
  - Contiene el siguiente número de secuencia que el emisor de la confirmación espera recibir: con ello confirma todos los bytes recibidos hasta el nº secuencia=(nºconfirmación - 1).
    - Es un ACK <u>acumulativo</u>



Cabecera TCP

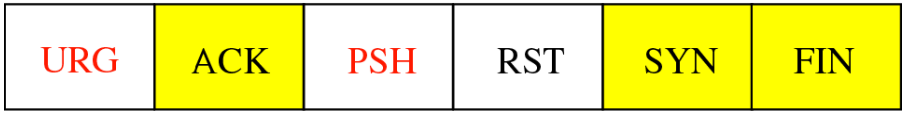
- Como los campos de confirmación están incluidos en la cabecera básica no supone un coste extra activarlo. Normalmente activado en la fase de intercambio de datos.
- Full-duplex, cada extremo debe llevar cuenta de su nº de secuencia y su nº de confirmación.

- Longitud cabecera (4bits)
  - Tamaño de la cabecera TCP en palabras de 32 bits para identificar así el tamaño variable de los campos de opciones.
- Reservados (6 bits)
  - No utilizados.

Cabecera TCP

- Flags (6bits)

URG: Urgent pointer is valid	RST: Reset the connection
ACK: Acknowledgment is valid	SYN: Synchronize sequence numbers
PSH: Request for push	FIN: Terminate the connection



  - URG: Urgent
    - Comunica al otro extremo que se han colocado datos urgentes en este segmento, y queda en manos del receptor qué hacer.
    - El campo “puntero urgente” es válido sólo ahora.
  - ACK: Acknowledgement
    - Indica que el campo “número de confirmación” es válido.
  - PSH: Push
    - Notifica al otro extremo para que pase los datos a la aplicación inmediatamente.
    - Hace referencia a datos que habría antes en el buffer receptor y a los nuevos de este segmento con PSH activado.
    - Muchas implementaciones adjuntan PSH automáticamente cuando se vacía el buffer del transmisor.

Cabecera TCP

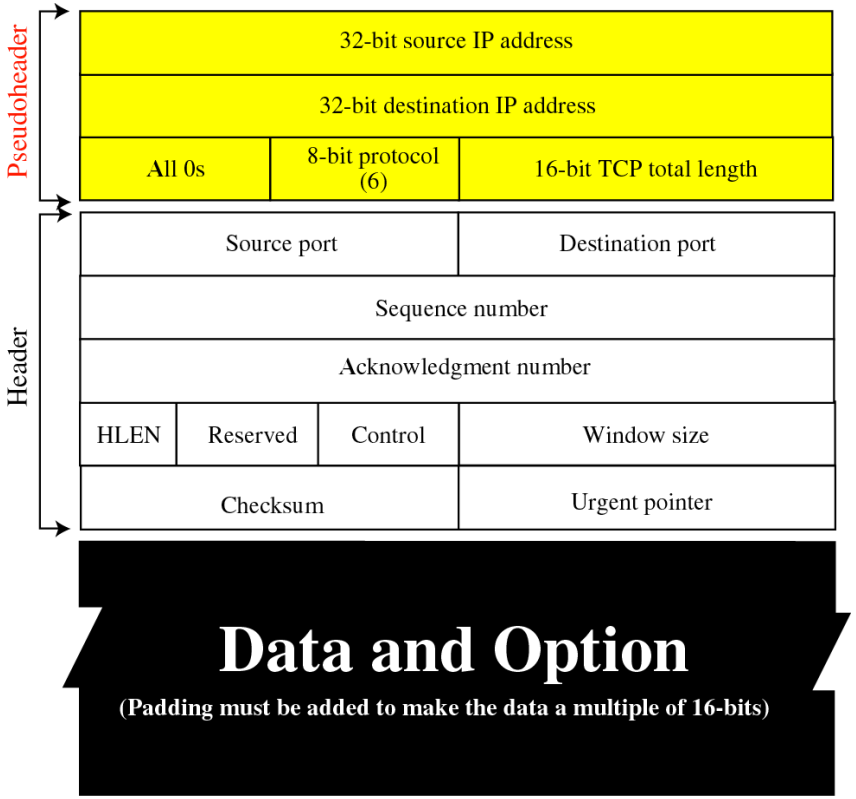
- RST: Reset
  - Cierra la conexión pasando directamente al estado “closed” (del diagrama de estados de TCP), sin pasar por la fase de finalización de conexión.
  - Todos los segmentos que hubiera en el buffer del receptor se desechan.
  - Se puede usar para:
    - Cierre brusco de la conexión (unilateral).
    - Avisar de que no existe aplicación escuchando en ese puerto.
- SYN: Synchronize
  - Sincronizar los nº de secuencia para iniciar la conexión.
  - Con SYN activo, el campo de nº de secuencia se interpreta como número de secuencia inicial (ISN).
  - Consume un nº de secuencia aunque no incluye datos.
- FIN: Finish
  - El emisor ha terminado de enviar datos.
  - Consume un nº de secuencia aunque no incluye datos.

Cabecera TCP

- Tamaño de ventana (16bits)
  - Ventana anunciada por el receptor W^R.
  - Control de flujo: indica el nº de bytes que el receptor está dispuesto a aceptar (posee buffer libre) desde el byte especificado por el campo número de confirmación.
  - Tamaño máximo de ventana 2^{16}-1=65535 bytes. Existe una opción de escalado de ventana para aumentar ese límite.
- Puntero urgente (16bits)
  - Sólo válido si el flag URG está activado.
  - Indica el offset que añadido al nº de secuencia TCP obtiene el nº de secuencia del último byte urgente.

Cabecera TCP

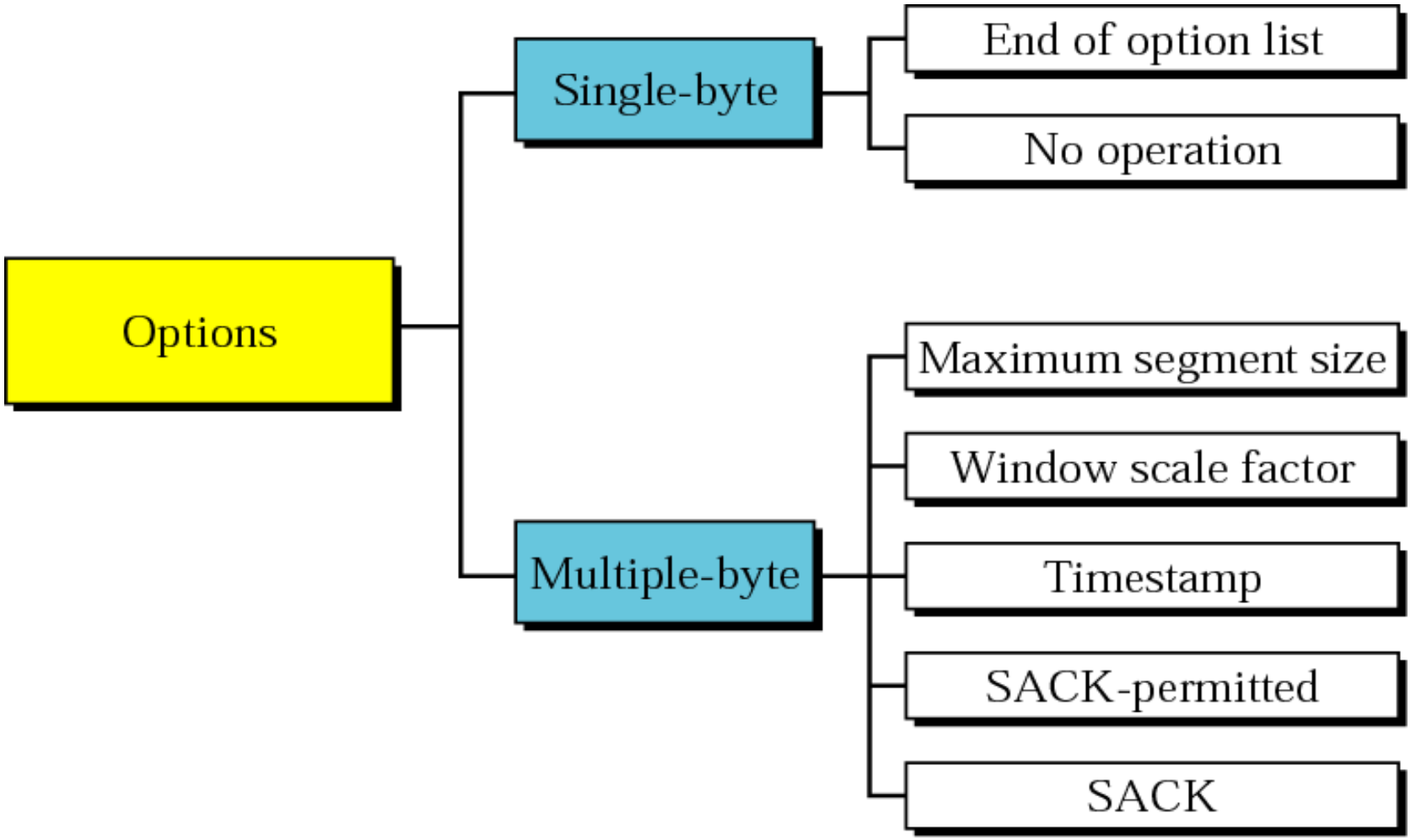
- Checksum (16bits)
  - Se calcula como en IP en palabras de 16 bits aplicado sobre:
    - Cabecera TCP
    - Datos TCP
    - Seudocabecera TCP
      - IP origen
      - IP destino
      - Protocolo
      - Longitud TCP



Cabecera TCP

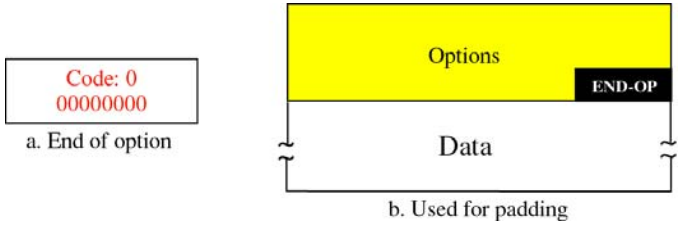
- Datos TCP (tamaño variable)
  - Opcional, no se incluye en paquetes de:
    - SYN
    - FIN

4.2 Opciones cabecera TCP



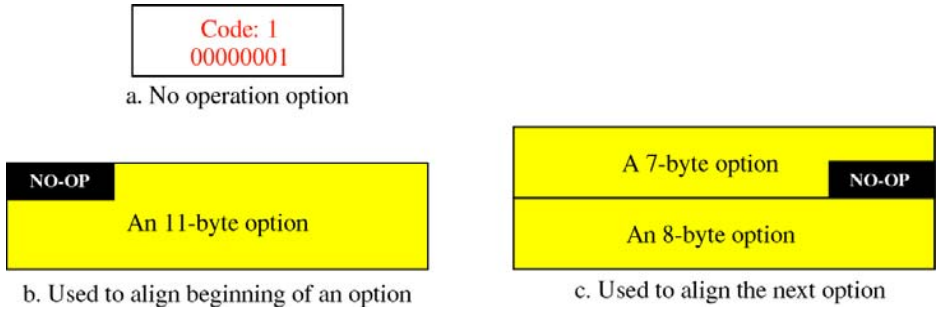
4.2.1 Opción: Fin de opciones

- Indica que:
  - No siguen más opciones.
  - Lo restante de esa palabra de 32 bits es basura.
  - Los datos comienzan en la siguiente palabra de 32 bits.



4.2.2 Opción: No operación

- Padding de la cabecera TCP a palabras de 4 bytes (exigido por campo longitud).
- Los campos de opciones de 2 o más bytes deben alinearse a palabras de 4 bytes.



4.2.3 Opción: Máximo tamaño de segmento

- Maximum Segment Size (MSS).
- Normalmente se incluye en el primer segmento intercambiado en la conexión, el de SYN activado.
- Especifica el tamaño de segmento máximo (campo de datos) que el receptor desea recibir para evitar fragmentación. Se calcula como:

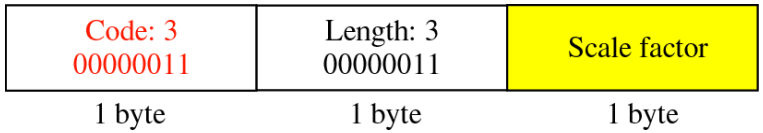
$$MSS=MTU_{camino} - Long.Cab.IP - Long.Cab.TCP$$

Ej: Ethernet MSS=1500-20-20=1460

- Para calcular la MTU del camino se aplica un mecanismo de descubrimiento.
  - En su defecto se usa la MTU del interfaz directamente conectado.
  - Si se desconoce también éste se toma por defecto 536 (576-20-20).

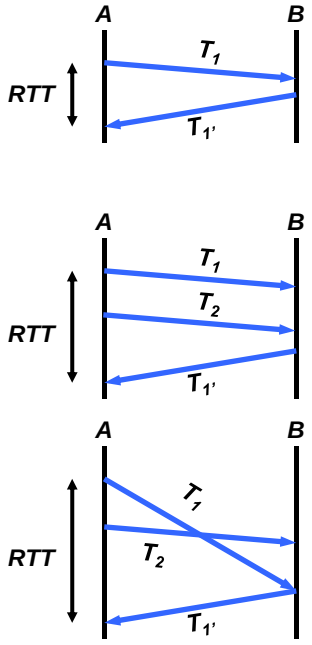
4.2.4 Opción: Escalado de ventana

- Permite incrementar la capacidad del tamaño de ventana de 16 a 30 bits sin tocar el campo correspondiente de la cabecera TCP.
- Factor escalado:
  - ventana final = ventana real * $2^{\text{factor escalado}}$
  - min=0: no escalado.
  - max=14: máximo valor de ventana final= $2^{16} * 2^{14}$
- Esta opción se manda sólo en el paquete de SYN y han de mandarlo ambos extremos aunque los factores de escalado puedan ser distintos en cada sentido.
- Si un extremo no manda la opción se supone que no la soporta y no se usa aunque el otro extremo la haya anunciado.
- El factor de escalado lo elige TCP automáticamente en función del tamaño de buffer disponible.

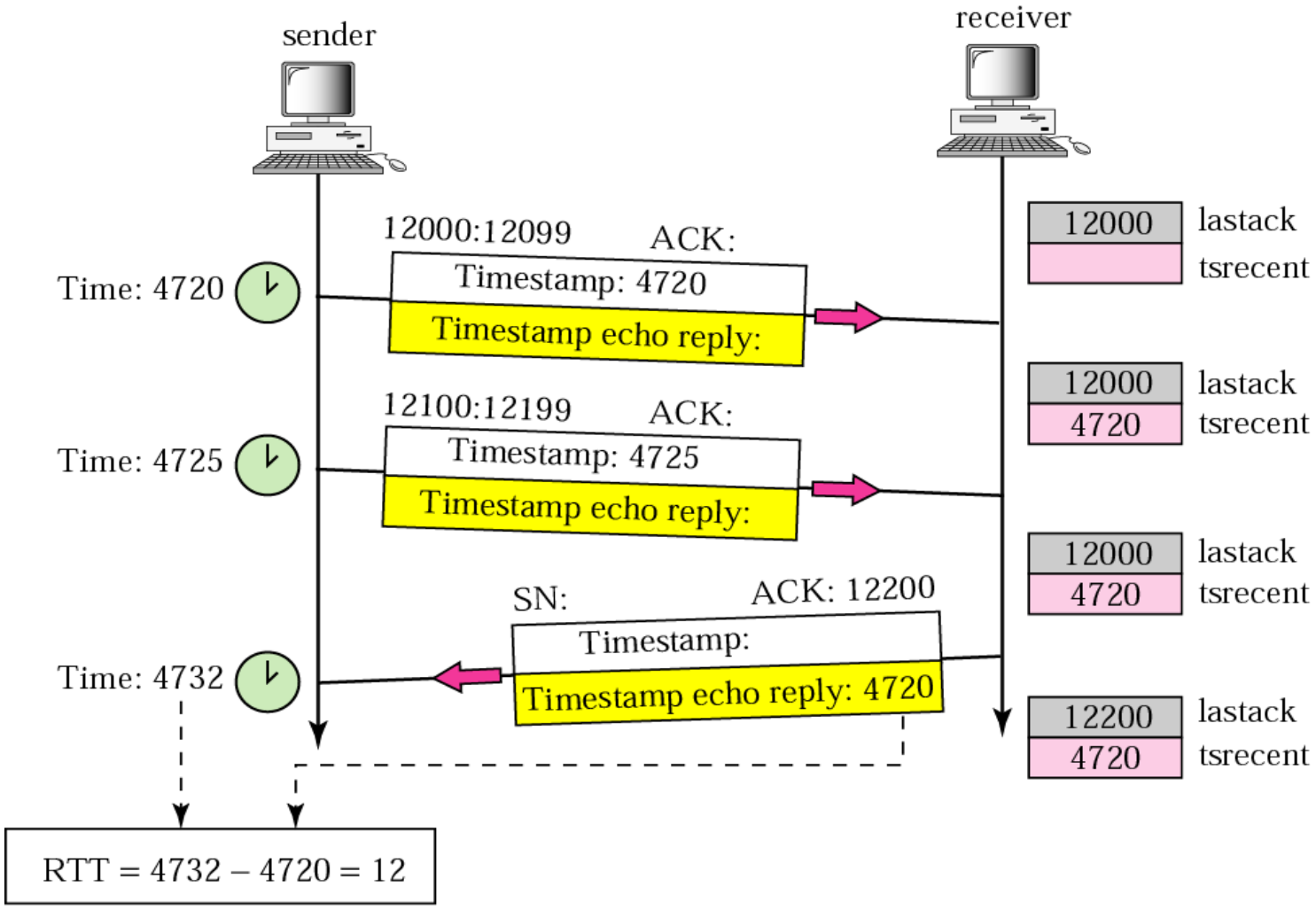


4.2.5 Opción: Timestamp

- Permite al emisor adjuntar un timestamp (en cualquier unidad porque se retorna al emisor) en cada segmento que después el receptor copiará en el paquete de confirmación, permitiendo al emisor calcular el RTT por cada ACK recibido.
- Debe enviarse en los SYNs iniciales para que luego se adjunte en todos los demás paquetes.
- Casos especiales:
  - Si se recibe un ACK por 2 segmentos mandados, se incluye copia del timestamp del primero de esos segmentos.
  - Si se reciben fuera de secuencia, el ACK adjunta copia del timestamp del primero que debería haber llegado aunque hayan llegado después en desorden.

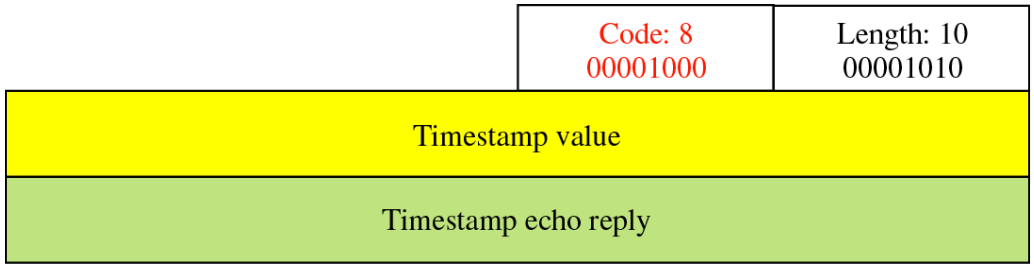


Opción: Timestamp



Opción: Timestamp

- Formato cabecera

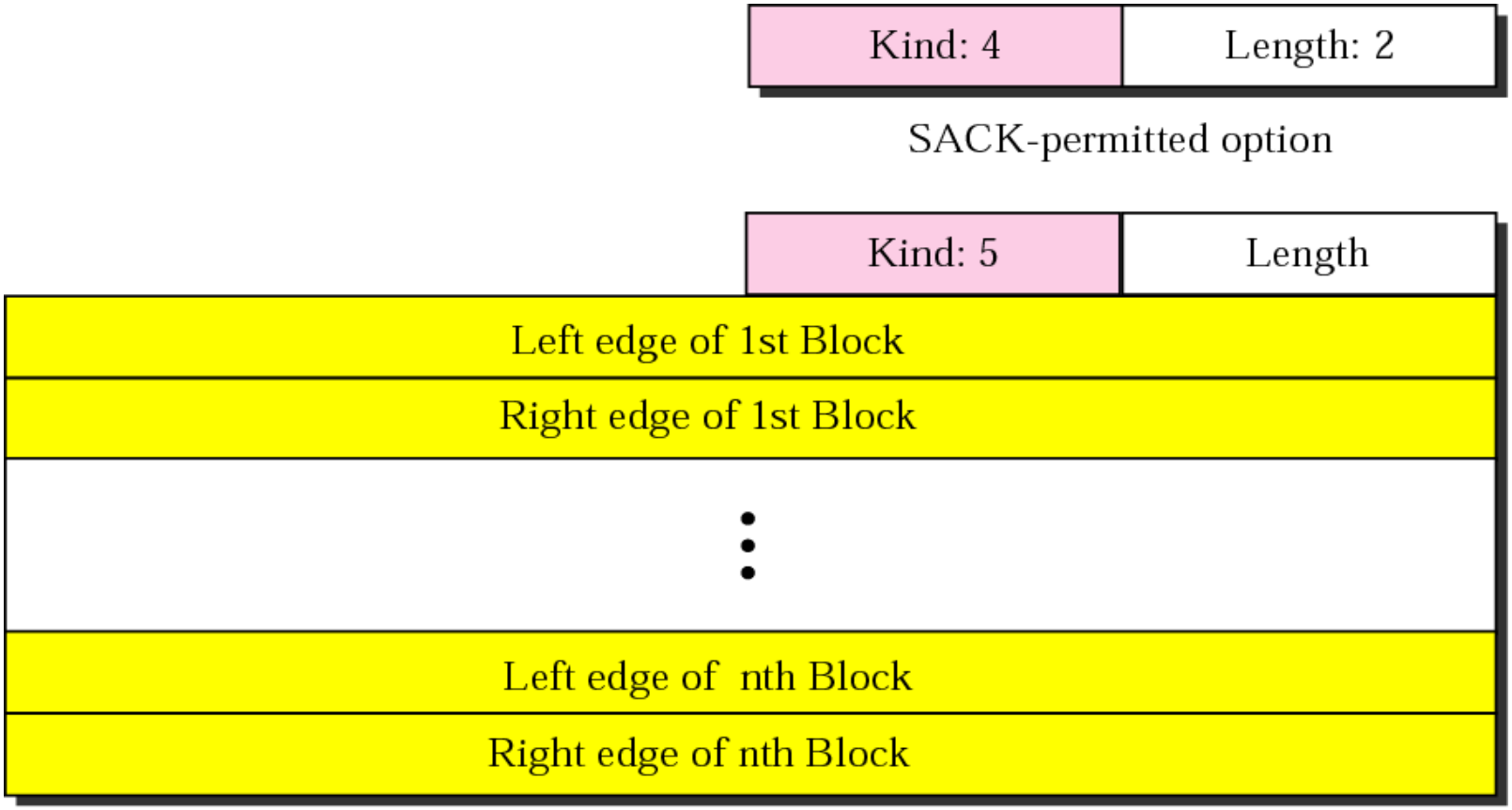


- PAWS: Protection Against Wrapped Sequence Numbers
  - En una red de muy alta velocidad puede ocurrir que al repetirse cíclicamente los números de secuencia (contador 32 bits, 2^{32}, 4,3 GB) se reciba en un momento un segmento con un nº de secuencia dentro de los esperados pero que en realidad es antiguo (pertenece al ciclo de contador anterior y no ha agotado su TTL).
  - Si se incluye la opción TCP de timestamp, este timestamp será menor en el segmento antiguo con respecto al esperado por lo que se ignorará. El timestamp no se resetea por lo que no tiene ese problema.

4.2.6 Opción: SACK

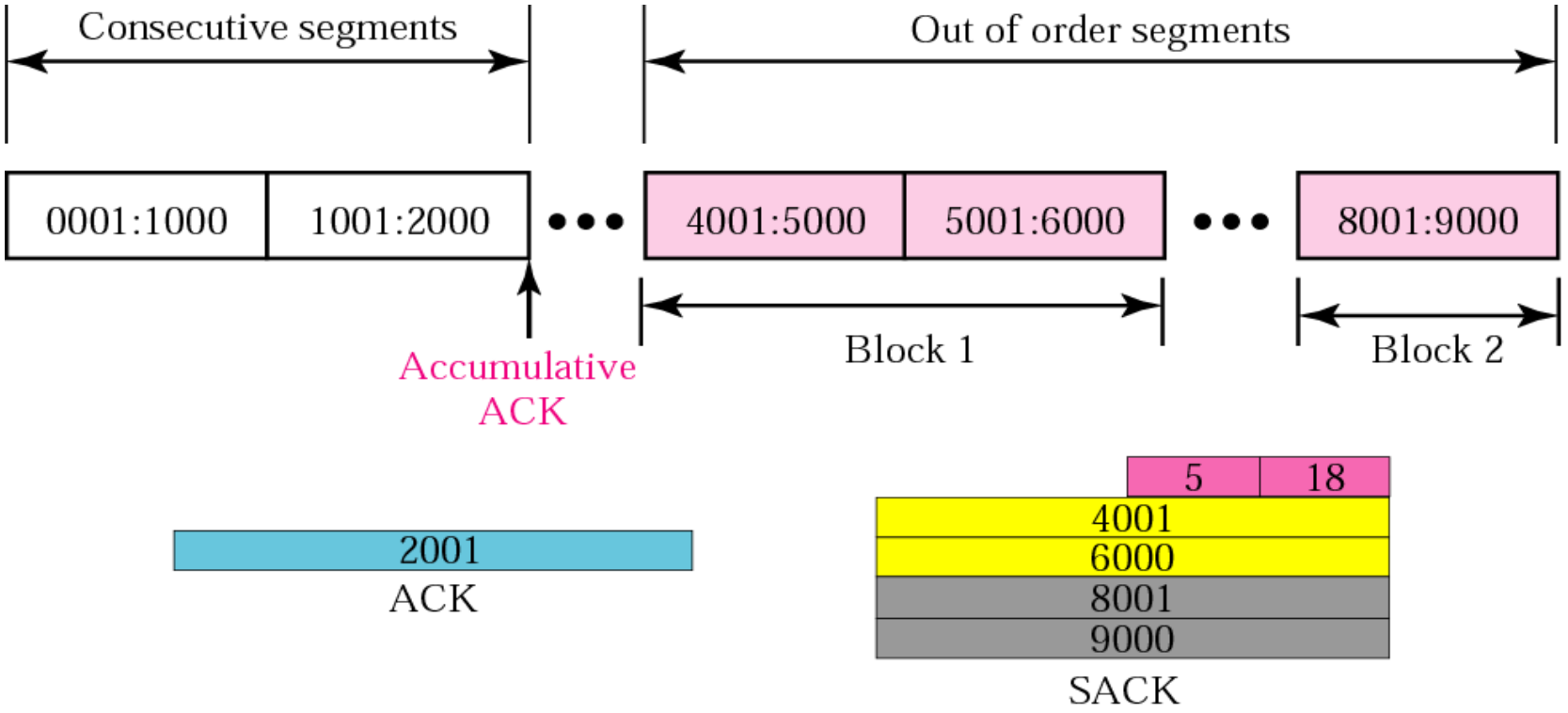
- SACK (Selective Acknowledgment)
- Permite confirmar al receptor aquellos segmentos recibidos sin que sea necesario que sean consecutivos y sin huecos como el ACK acumulativo.
- Se consigue evitar que el emisor envíe segmentos duplicados que ya se recibieron anteriormente.
- Se implementa en 2 opciones:
  - SACK-permitted: únicamente se utiliza en la fase de establecimiento cuando se envía el SYN y el SYN+ACK, para avisar al otro extremo que se pretende utilizar SACK. Ambos deben señalizar su uso para que tenga efecto.
  - SACK option: de tamaño variable, se utiliza en la fase de transferencia. Consiste en un listado de bloques que han llegado fuera de orden, con 2 números de 32 bits, que indican respectivamente el comienzo y el final del bloque recibido (inclusives).

Opción: SACK


Kind: 4
Length: 2
SACK-permitted option
Kind: 5
Length
Left edge of 1st Block
Right edge of 1st Block
Left edge of nth Block
Right edge of nth Block
SACK option

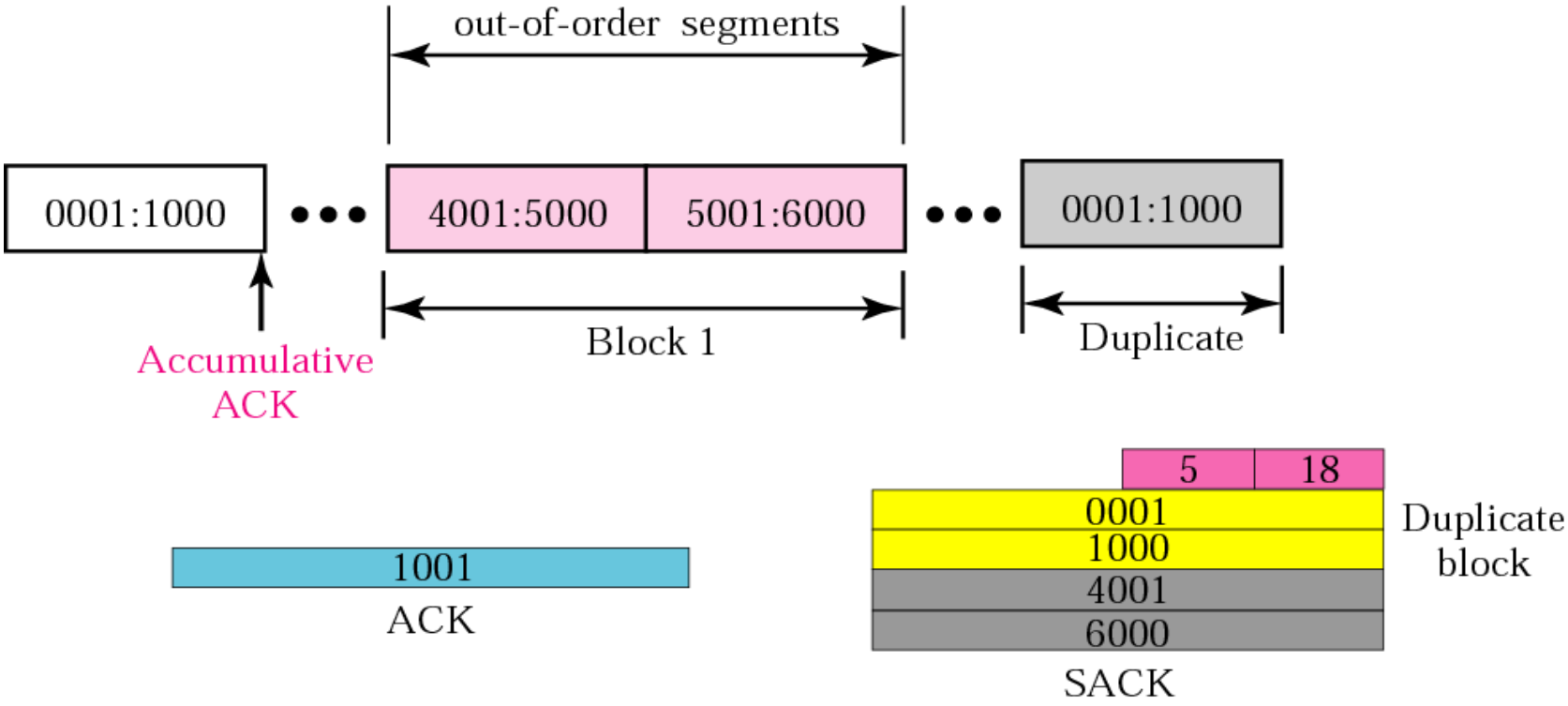
Opción: SACK

- Ejemplo, segmentos 3, 4 y 5 fuera de orden



Opción: SACK

- Ejemplo, segmentos 4 duplicado



Resumen

- TCP ofrece un servicio orientado a conexión.
  - De intercambio entre extremos de un *stream* de bytes..
  - Dividiendo los datos en bloques de tamaño óptimo: MSS.
  - Fiable.
  - Con control de flujo.
  - Con control de congestión.
- La cabecera TCP incorpora multitud de campos para soportar la complejidad del protocolo.
- Usa un esquema de fiabilidad basado en ACK acumulativos
- Una conexión TCP es bidireccional, con campos de número de secuencia y confirmación independientes en cada sentido

Referencias

- [Forouzan]
  - Capítulo 15, “Transmission Control Protocol (TCP)”, secciones 15.1-15.3, 15.11
- [Stevens]
  - Capítulo 17, “TCP: Transmission Control Protocol”