

# Protocolos de transporte TCP y UDP

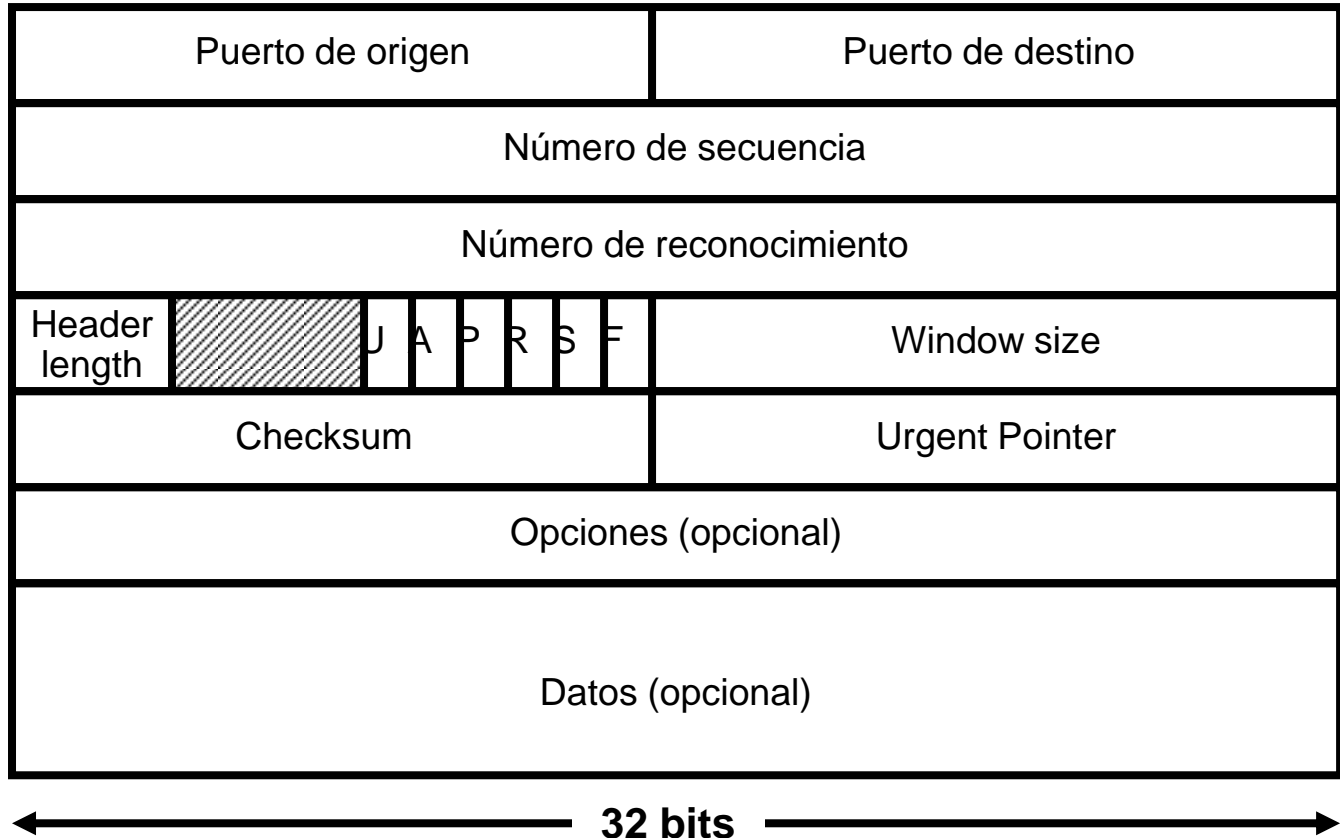
# Agenda

- **Protocolo TCP**
  - **Direccionamiento**
  - **Establecimiento y corte de conexiones**
  - **Control de flujo**
  - **Control de congestión**
- **Protocolo UDP**

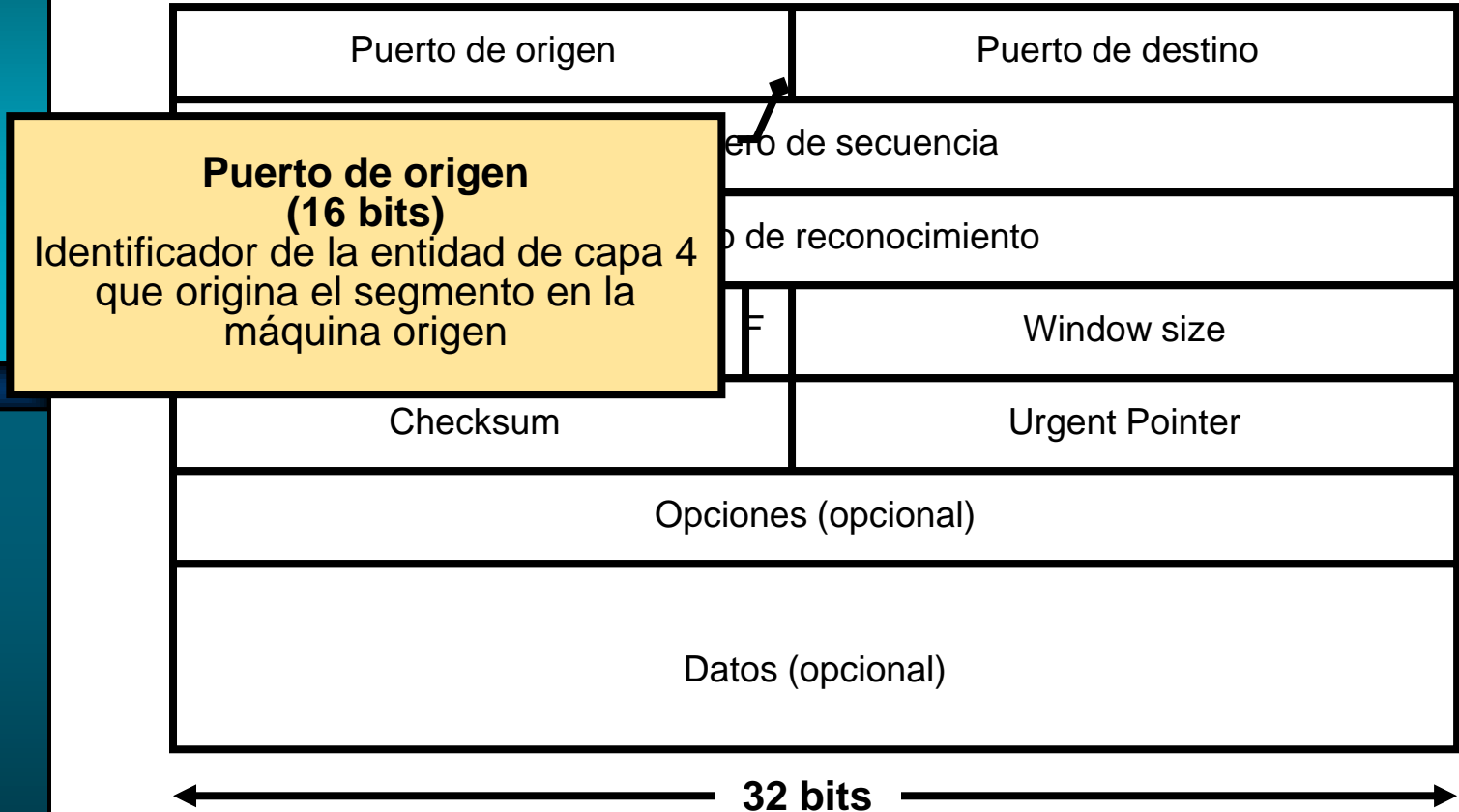
# Protocolo TCP

- **Protocolo de capa de transporte (extremo a extremo) orientado a conexión y confiable**
- **Garantiza un flujo confiable de información entre un cliente y un servidor**

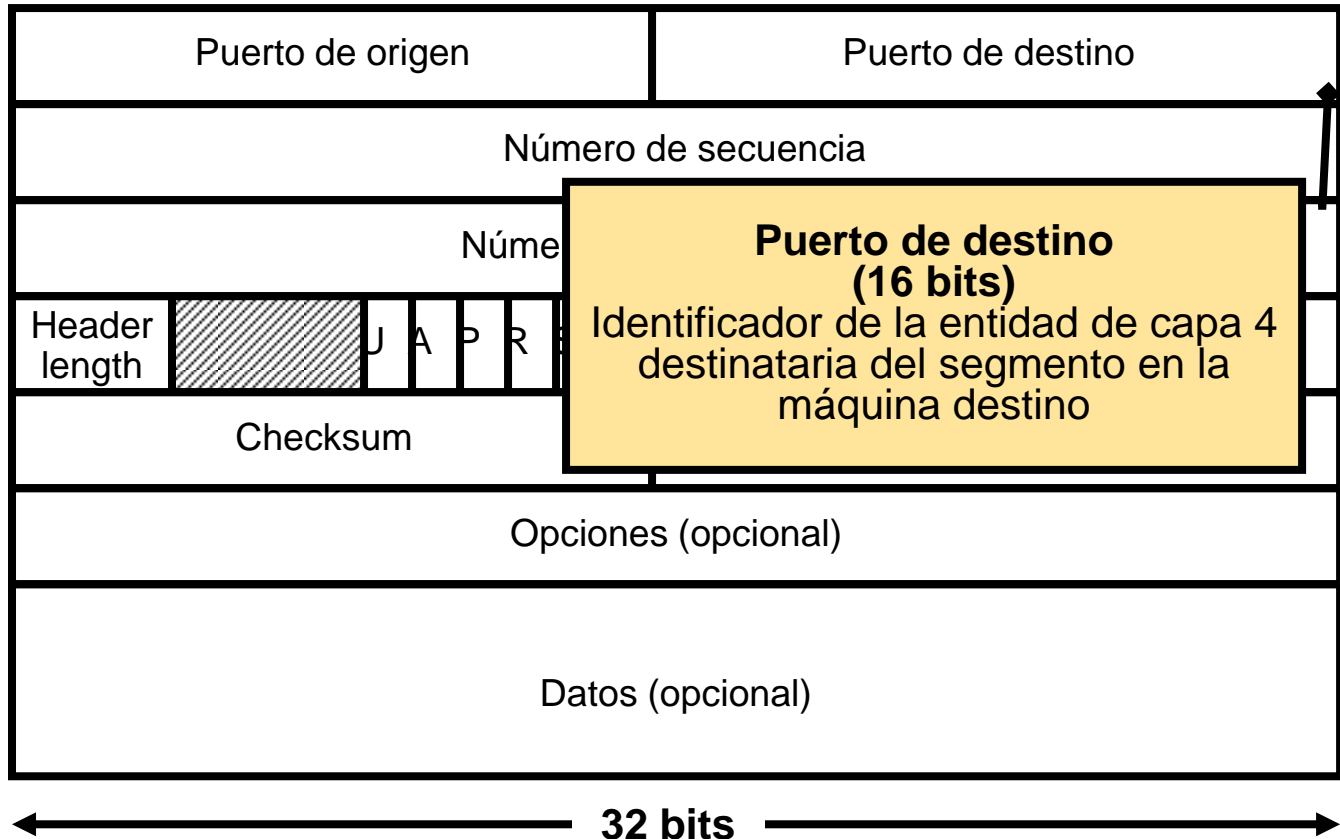
# Formato del segmento TCP



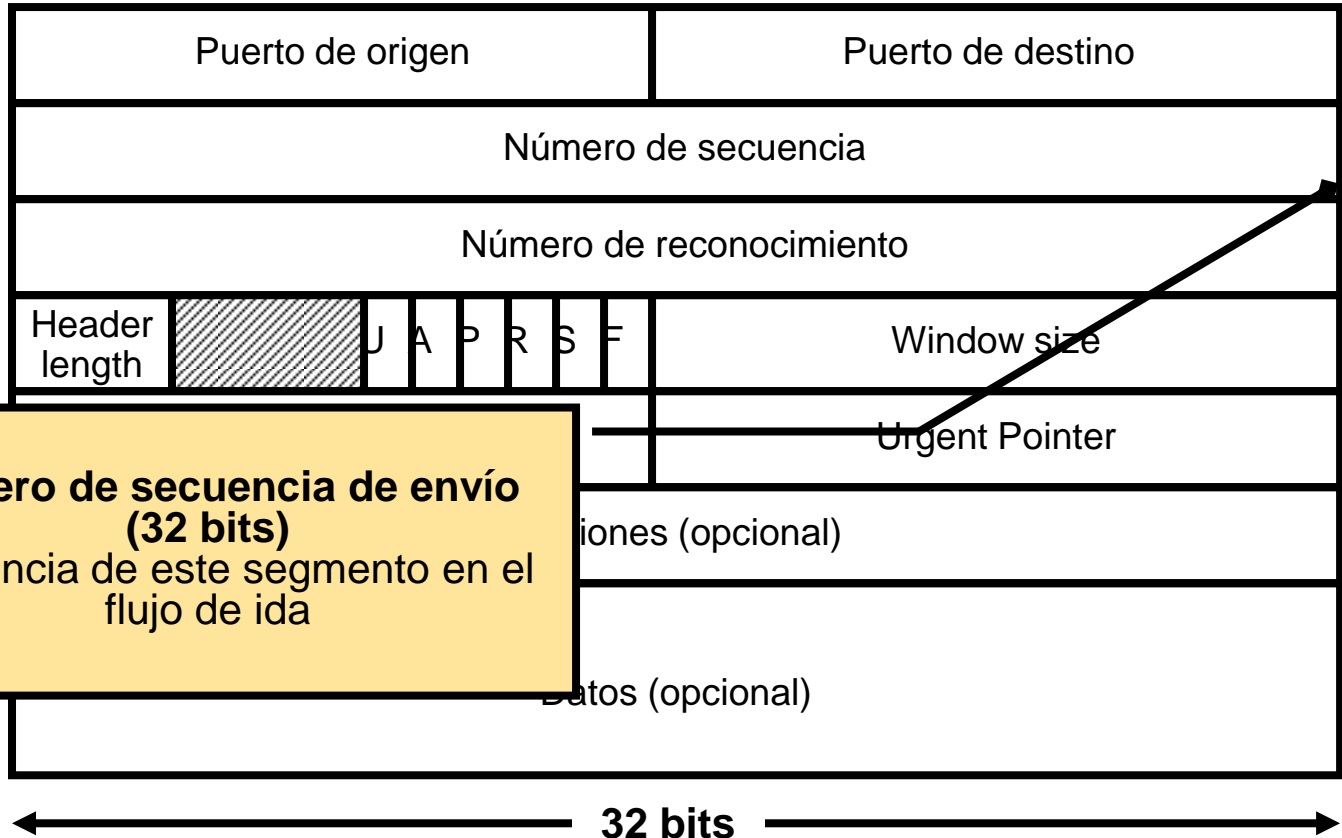
# Formato del segmento TCP



# Formato del segmento TCP

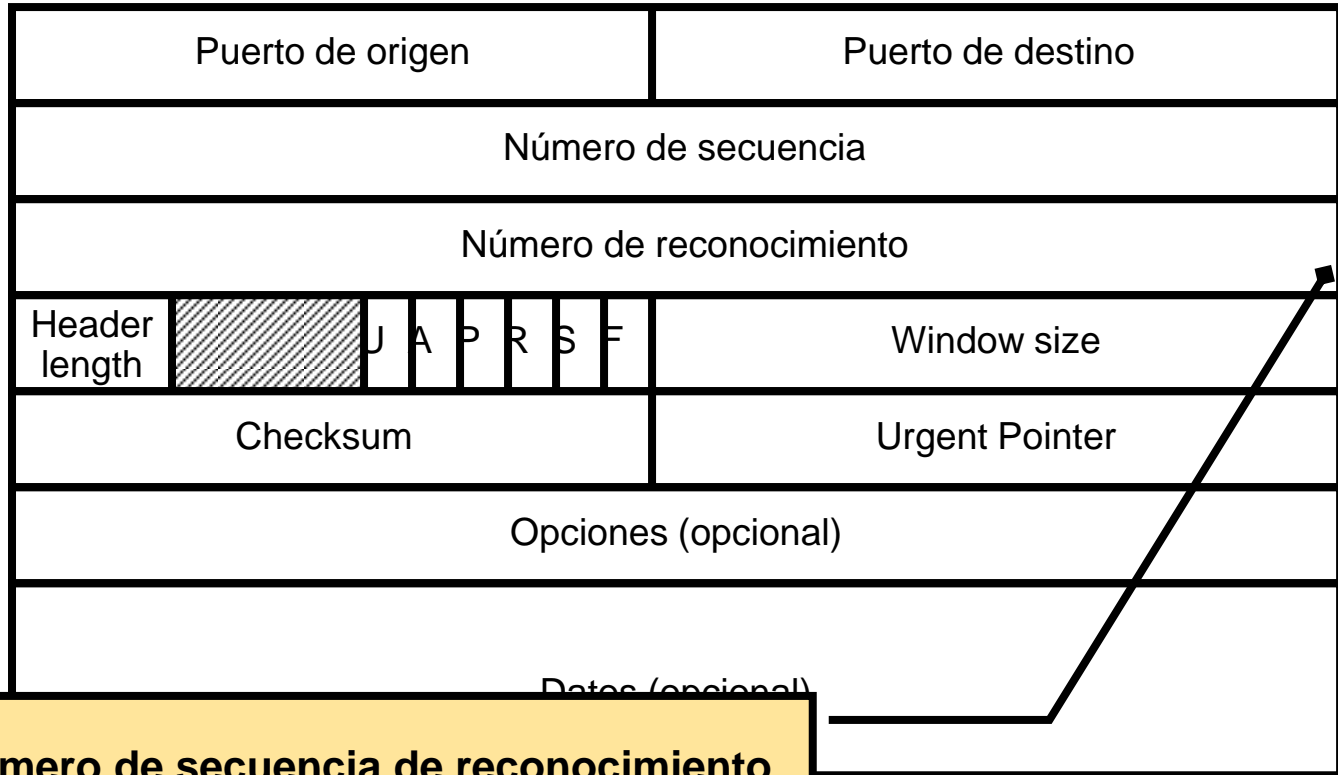


# Formato del segmento TCP



**Número de secuencia de envío (32 bits)**  
Secuencia de este segmento en el flujo de ida

# Formato del segmento TCP

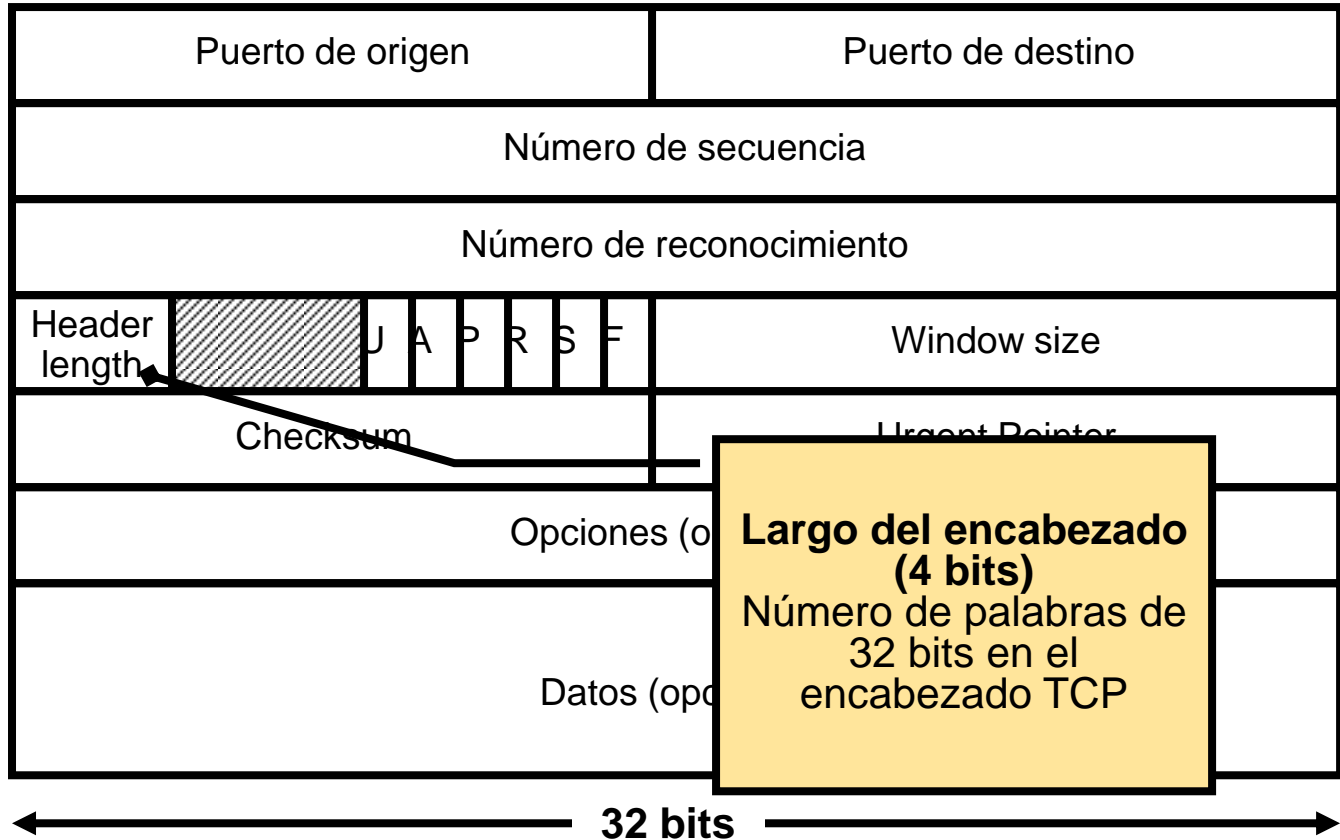


## Número de secuencia de reconocimiento (32 bits)

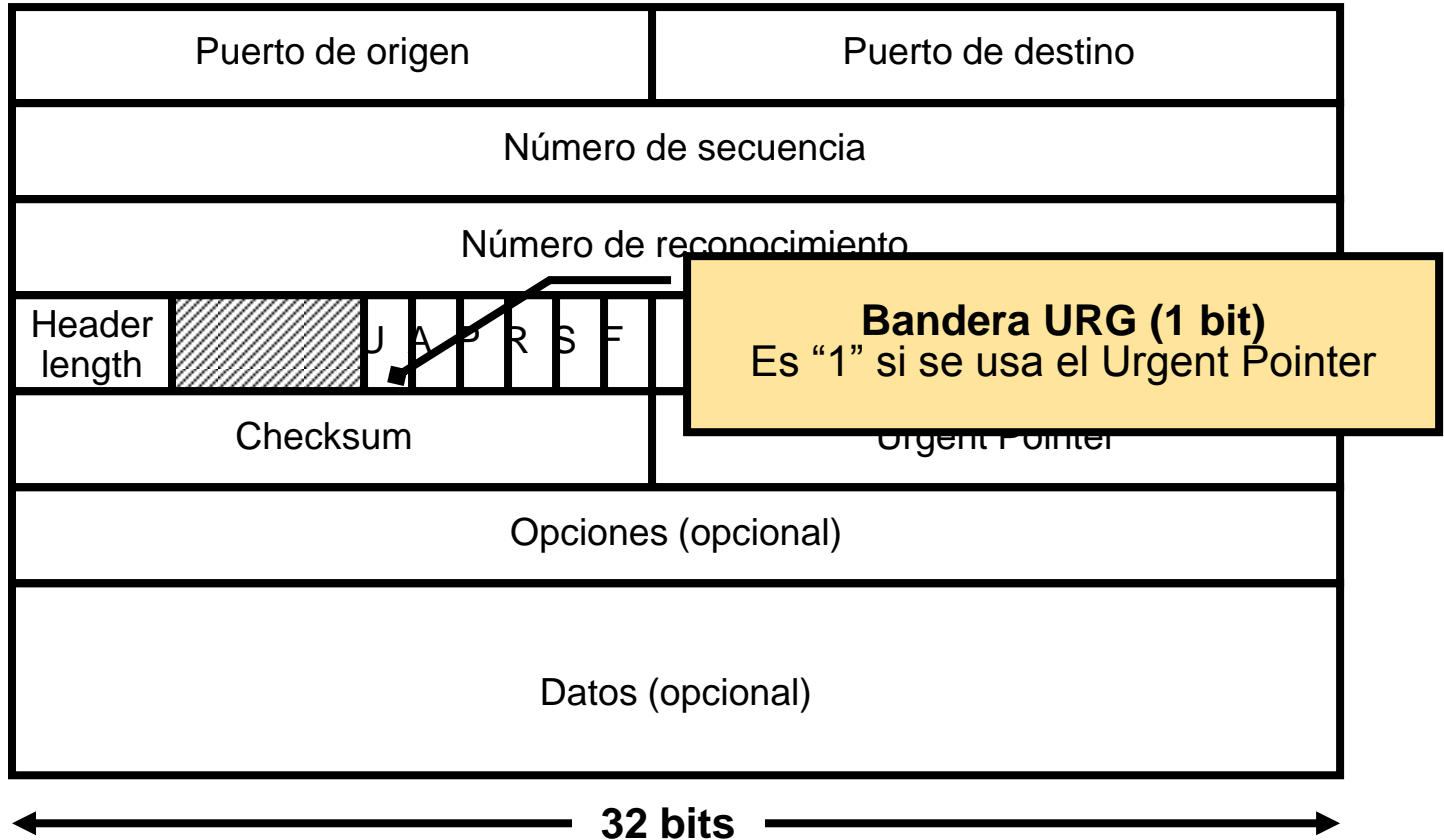
Indica al destinatario del segmento, cual es el próximo número de secuencia que el originador espera recibir en el flujo de vuelta. Es válido si la bandera de ACK está en 1



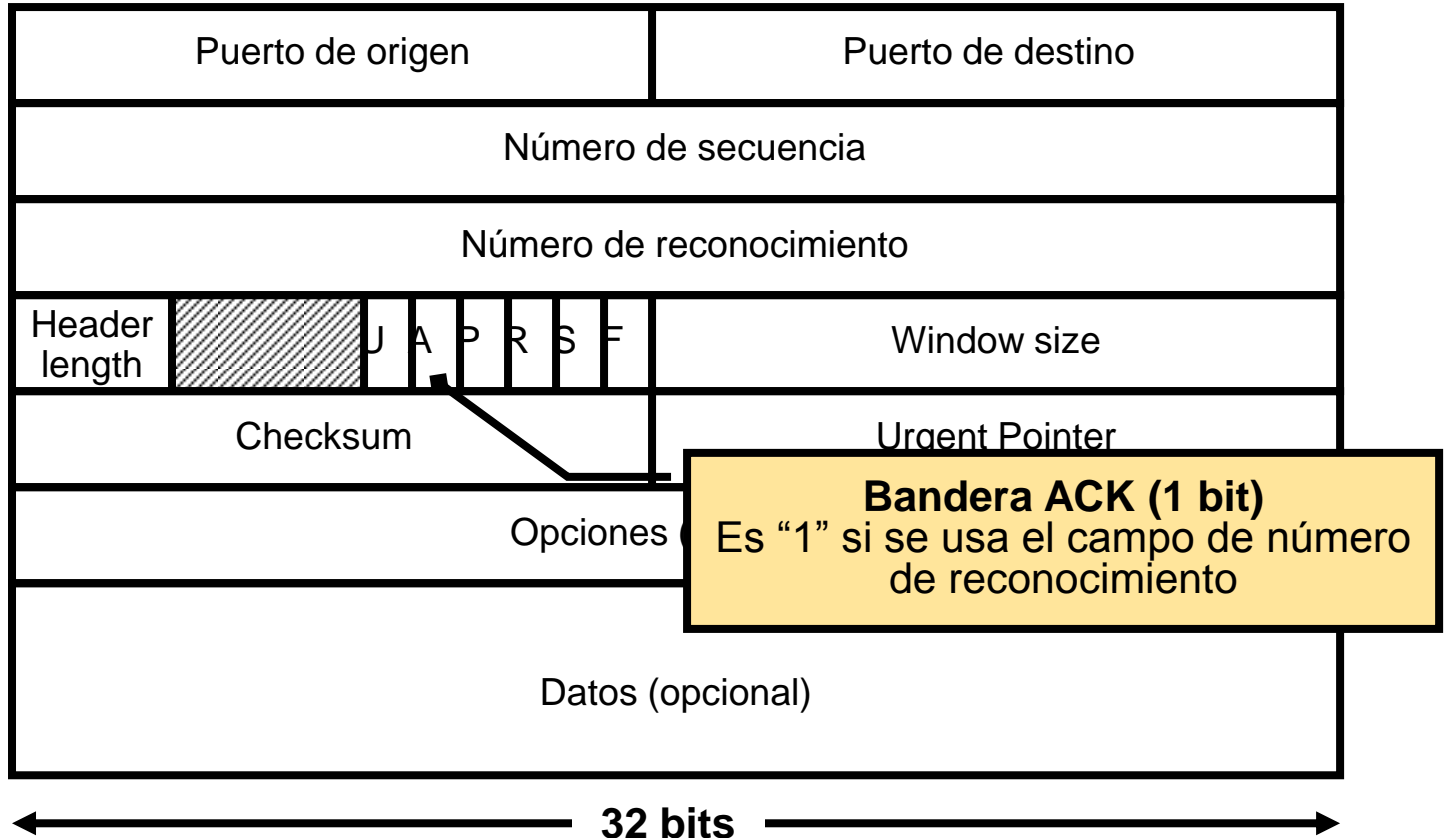
# Formato del segmento TCP



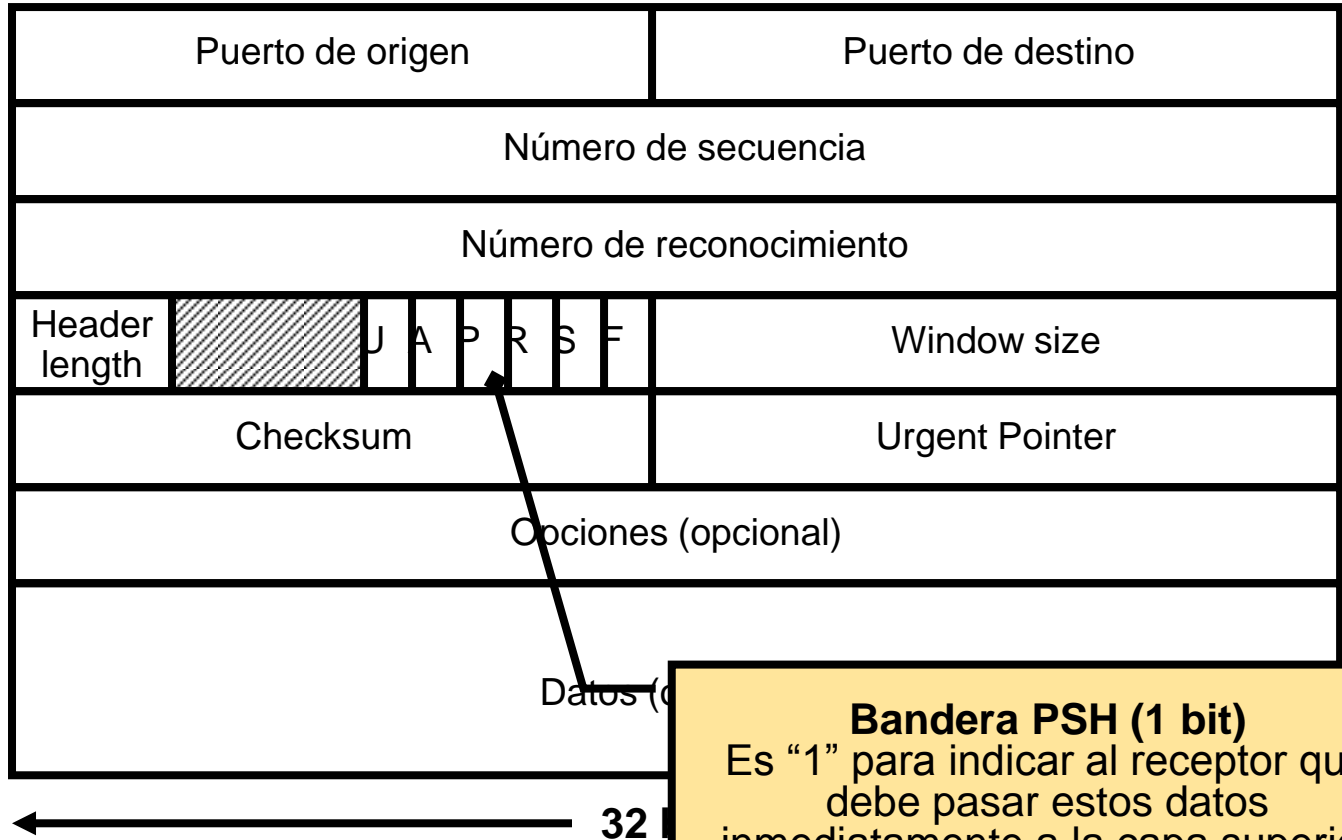
# Formato del segmento TCP



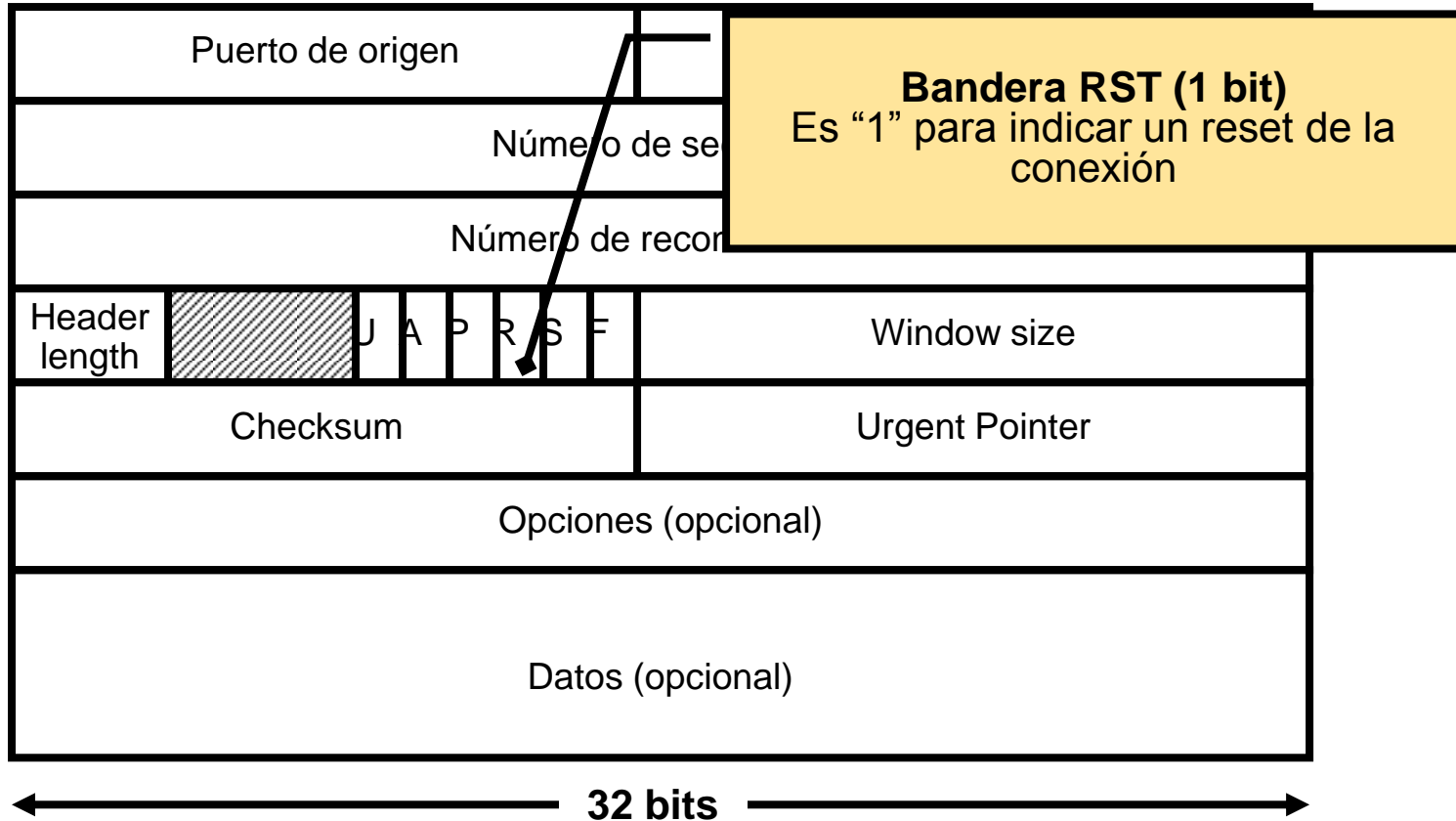
# Formato del segmento TCP



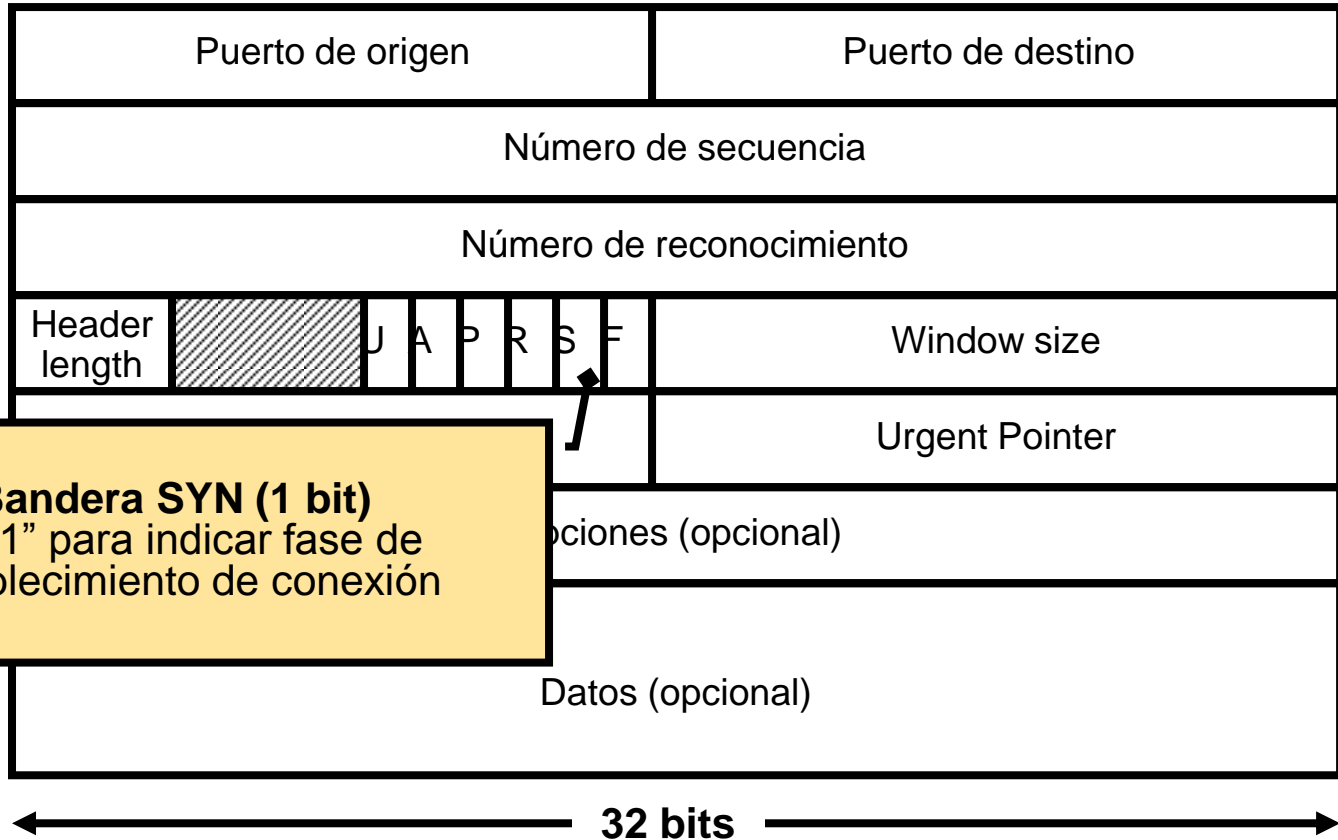
# Formato del segmento TCP



# Formato del segmento TCP

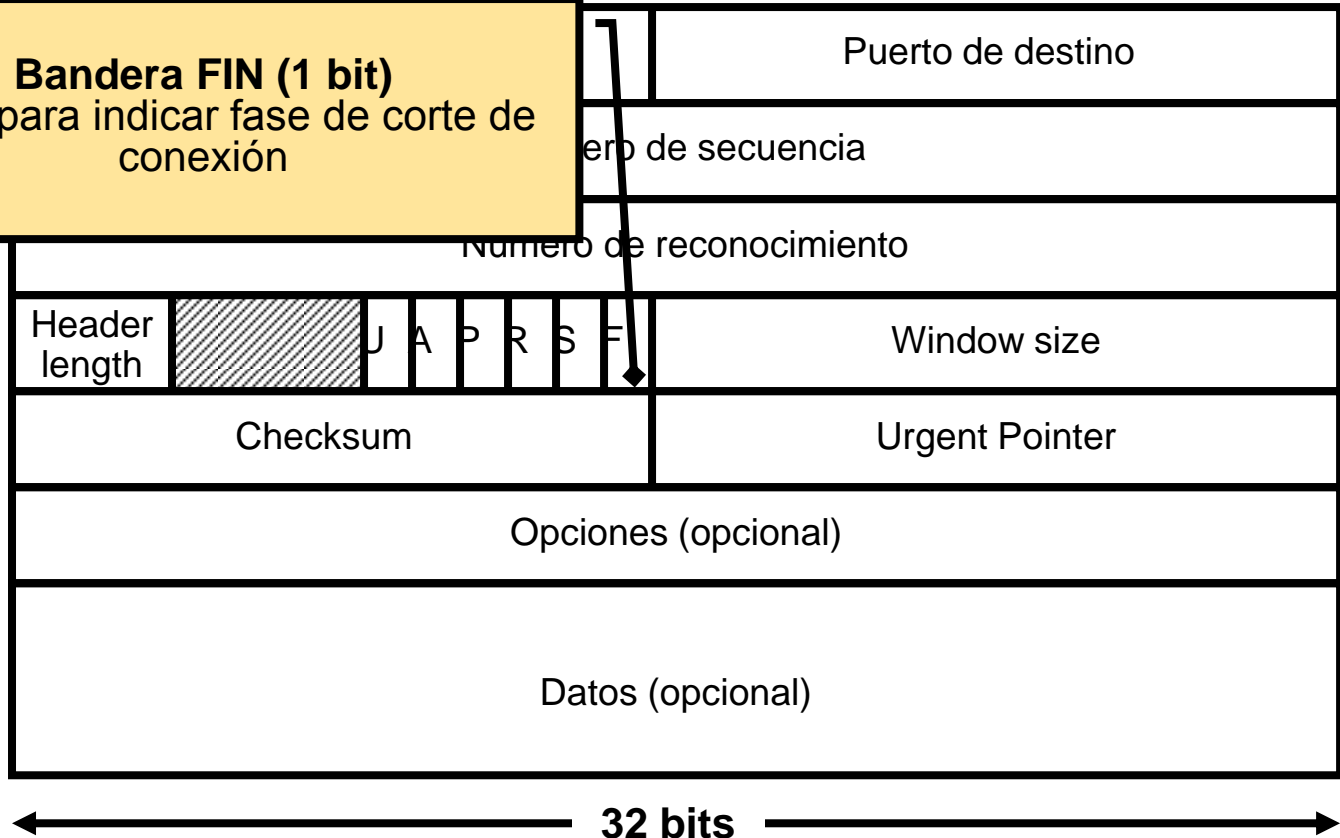


# Formato del segmento TCP

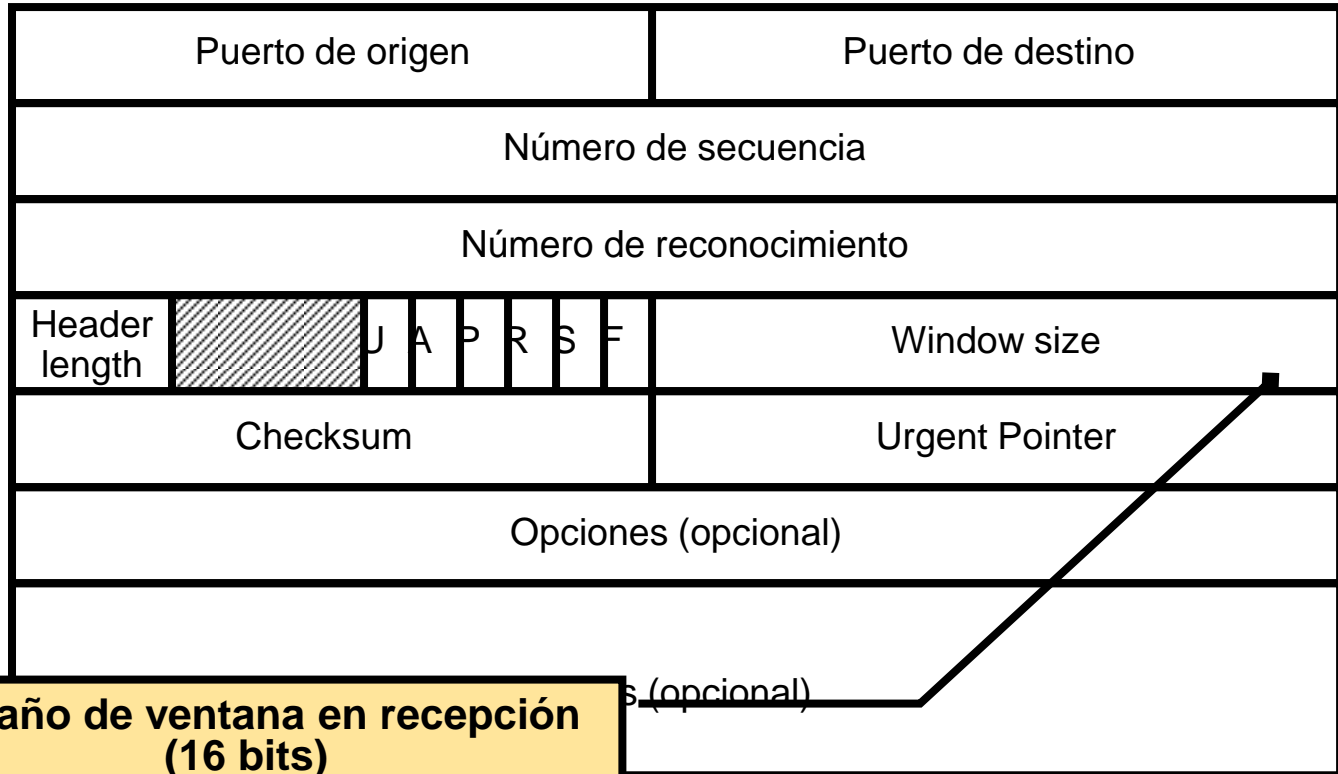


# Formato del segmento TCP

**Bandera FIN (1 bit)**  
Es "1" para indicar fase de corte de conexión



# Formato del segmento TCP

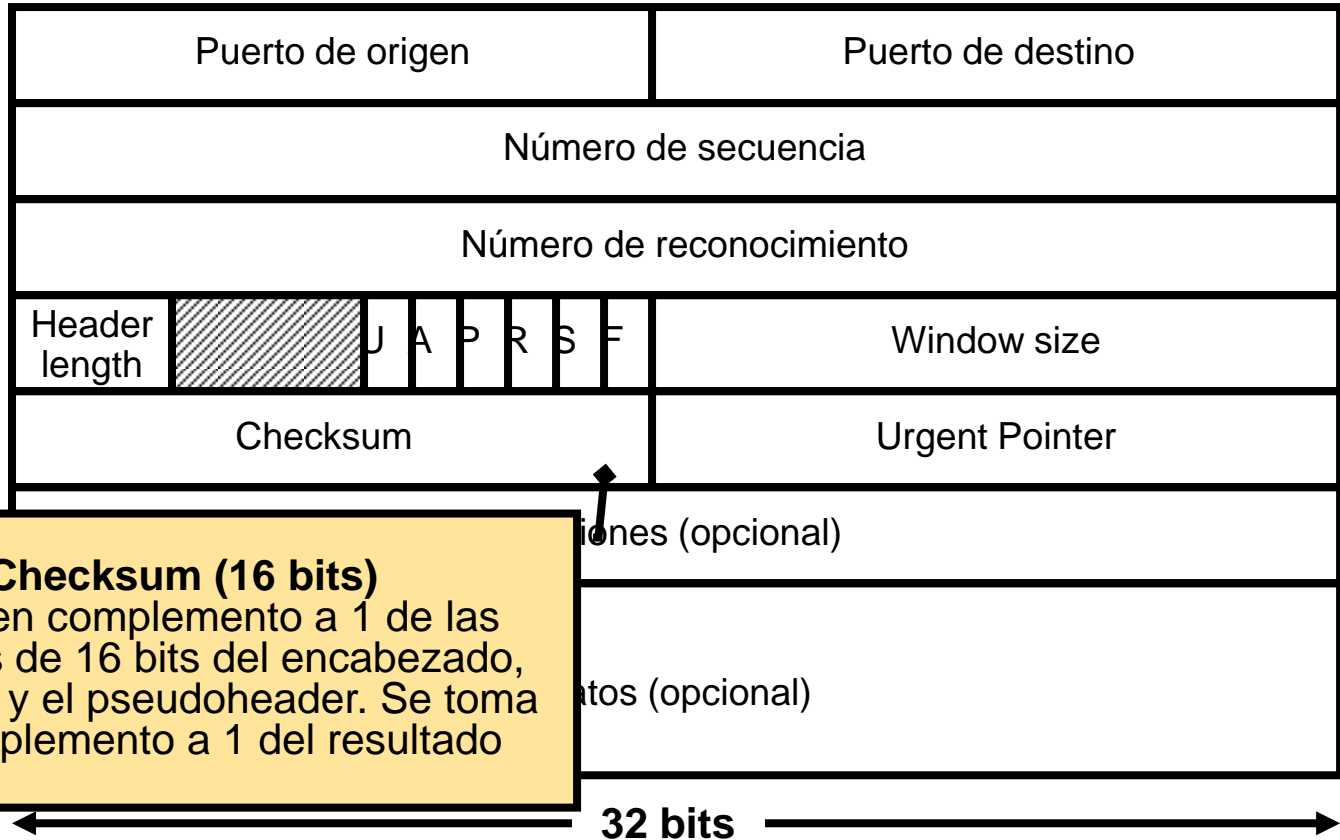


**Tamaño de ventana en recepción (16 bits)**  
Indica la cantidad máxima de bytes que se aceptarán en el flujo de vuelta (control de flujo)





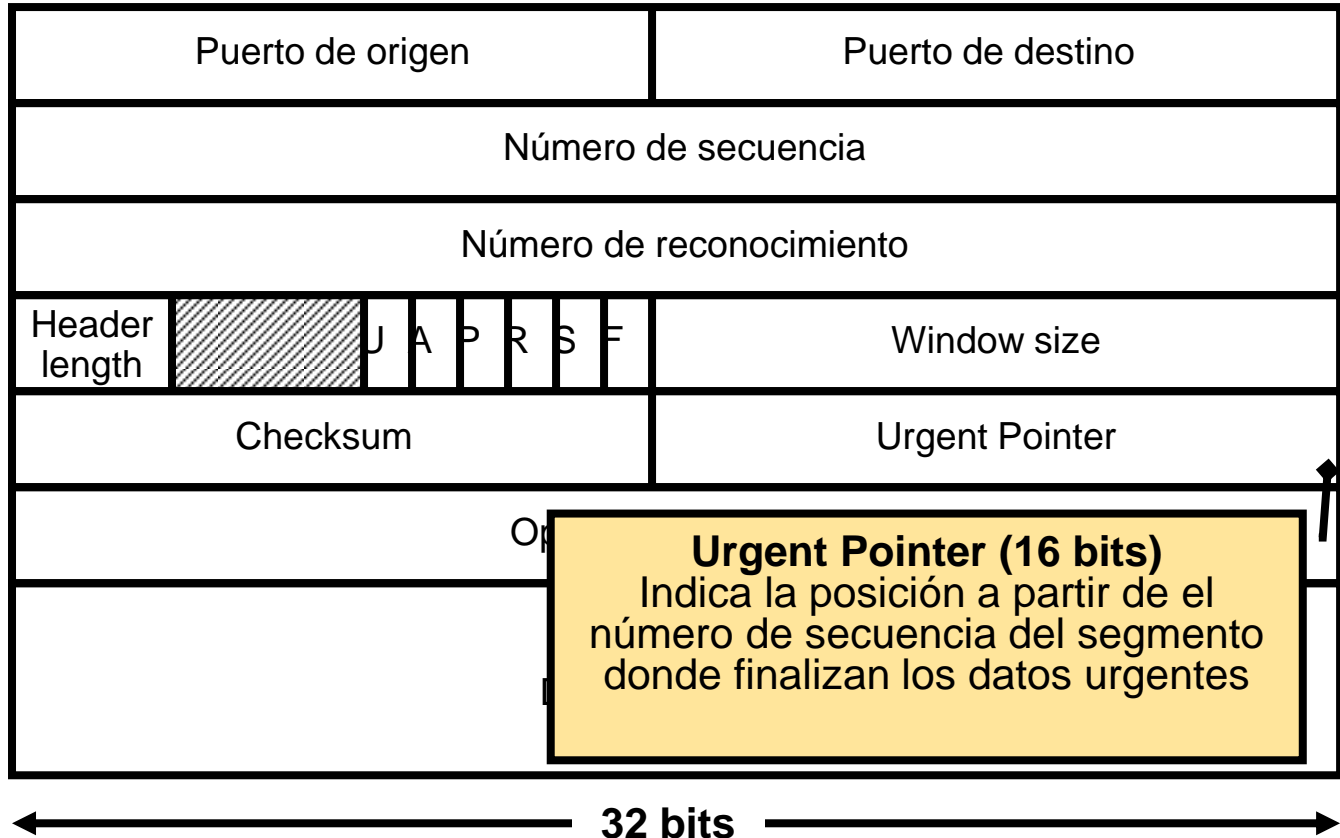
# Formato del segmento TCP



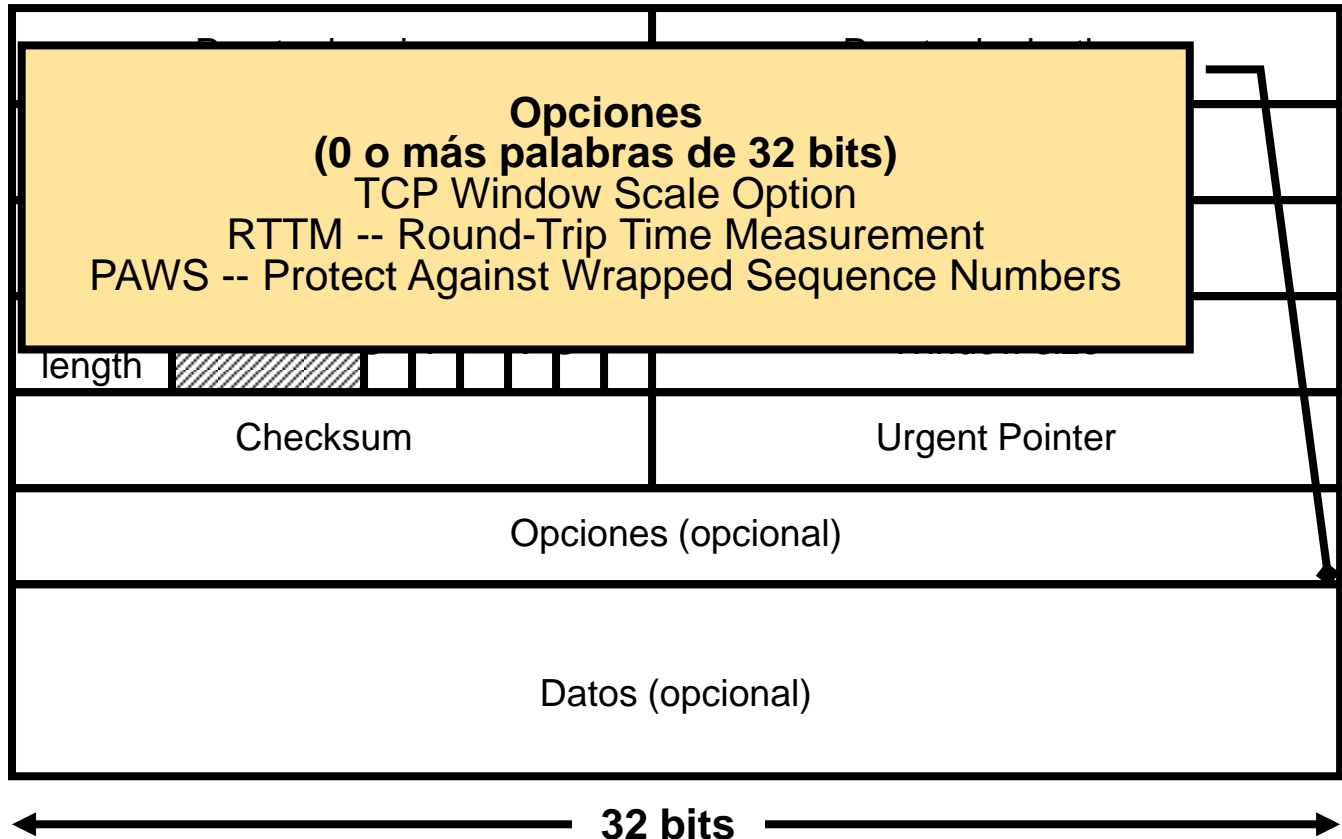
## Checksum (16 bits)

Suma en complemento a 1 de las palabras de 16 bits del encabezado, los datos y el pseudoheader. Se toma el complemento a 1 del resultado

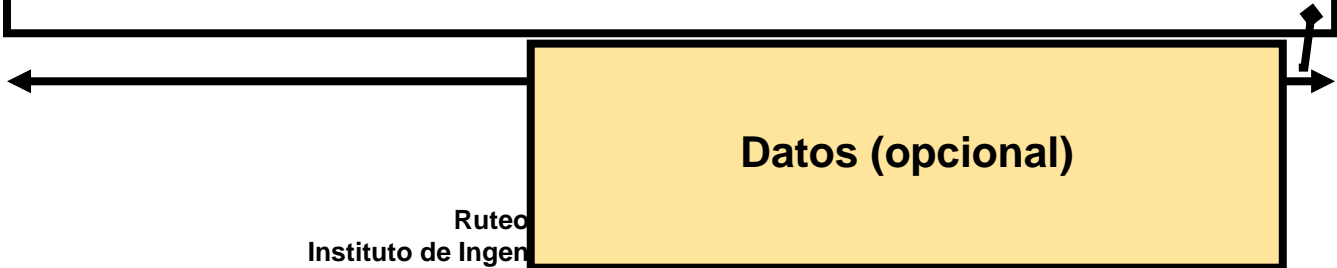
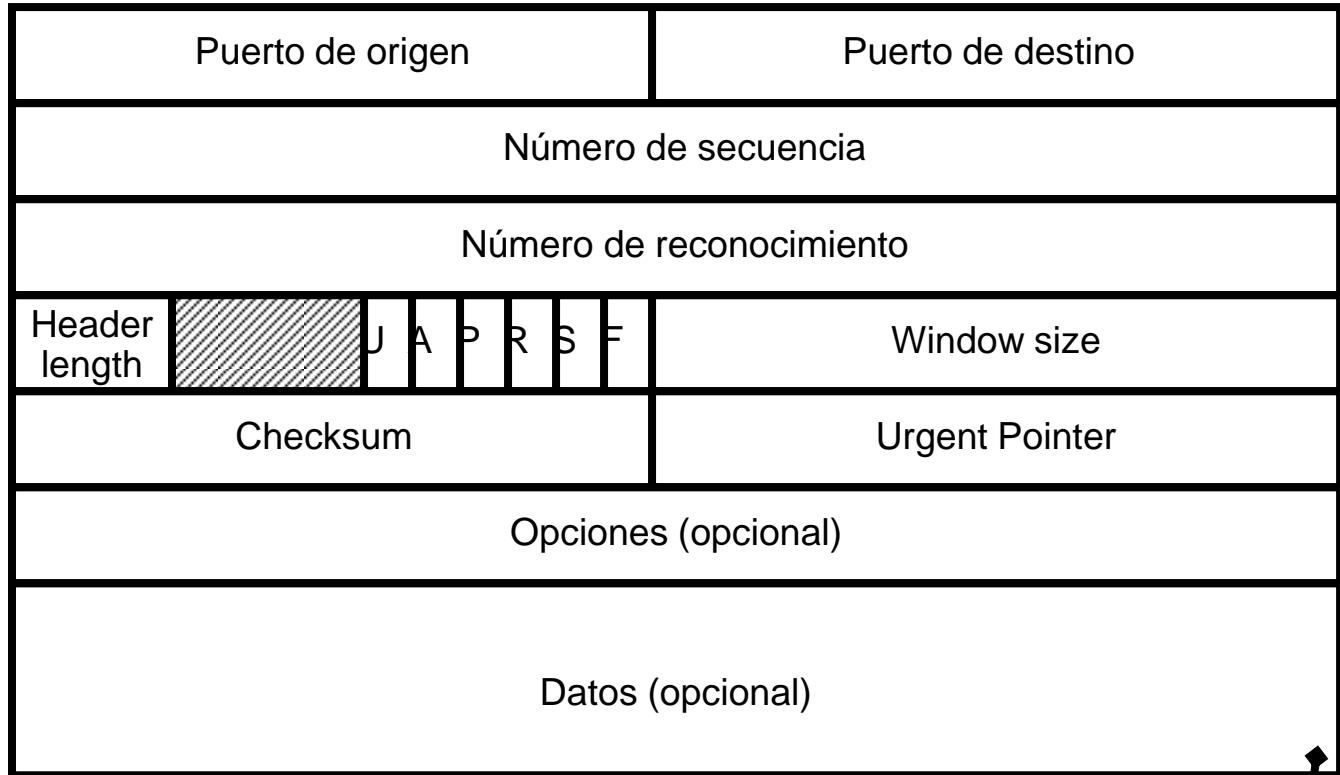
# Formato del segmento TCP



# Formato del segmento TCP

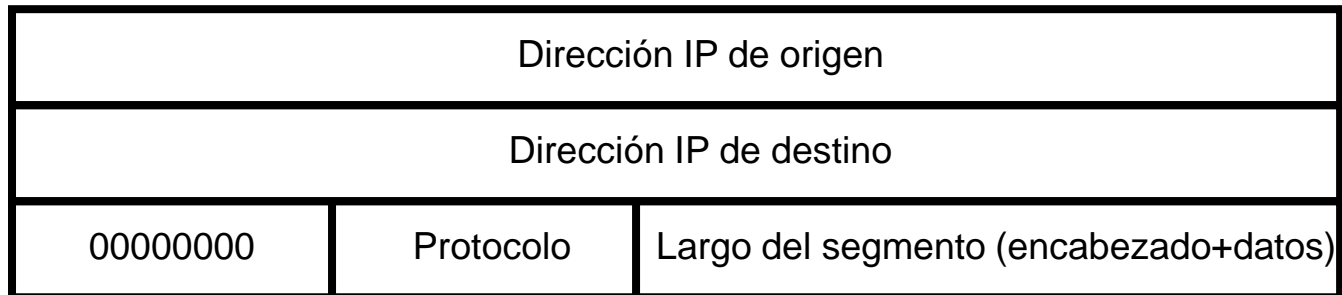


# Formato del segmento TCP



# Pseudo header

- Se utiliza en el cálculo del checksum de TCP
- Viola la independencia de capas



← 32 bits →

# Establecimiento de conexión

Originador



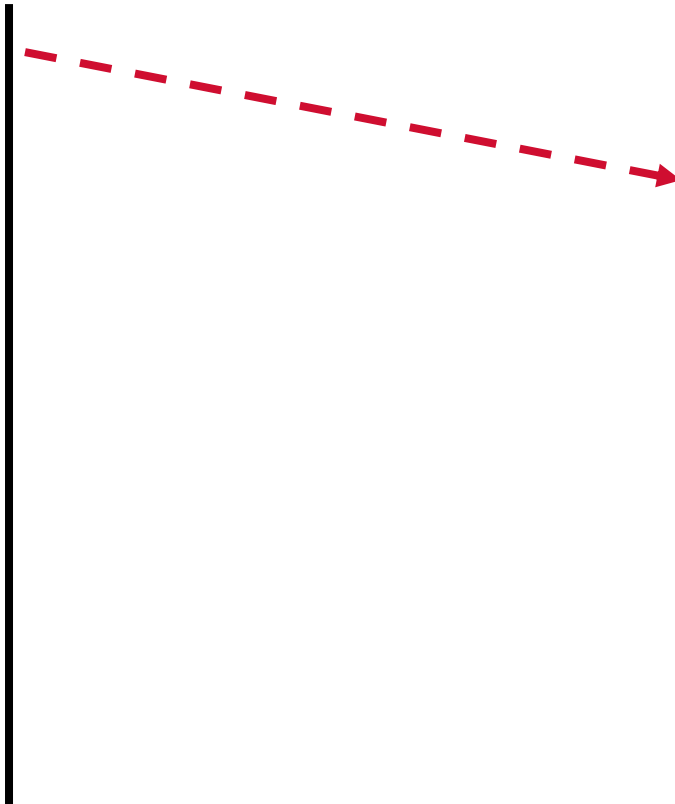
Destinatario



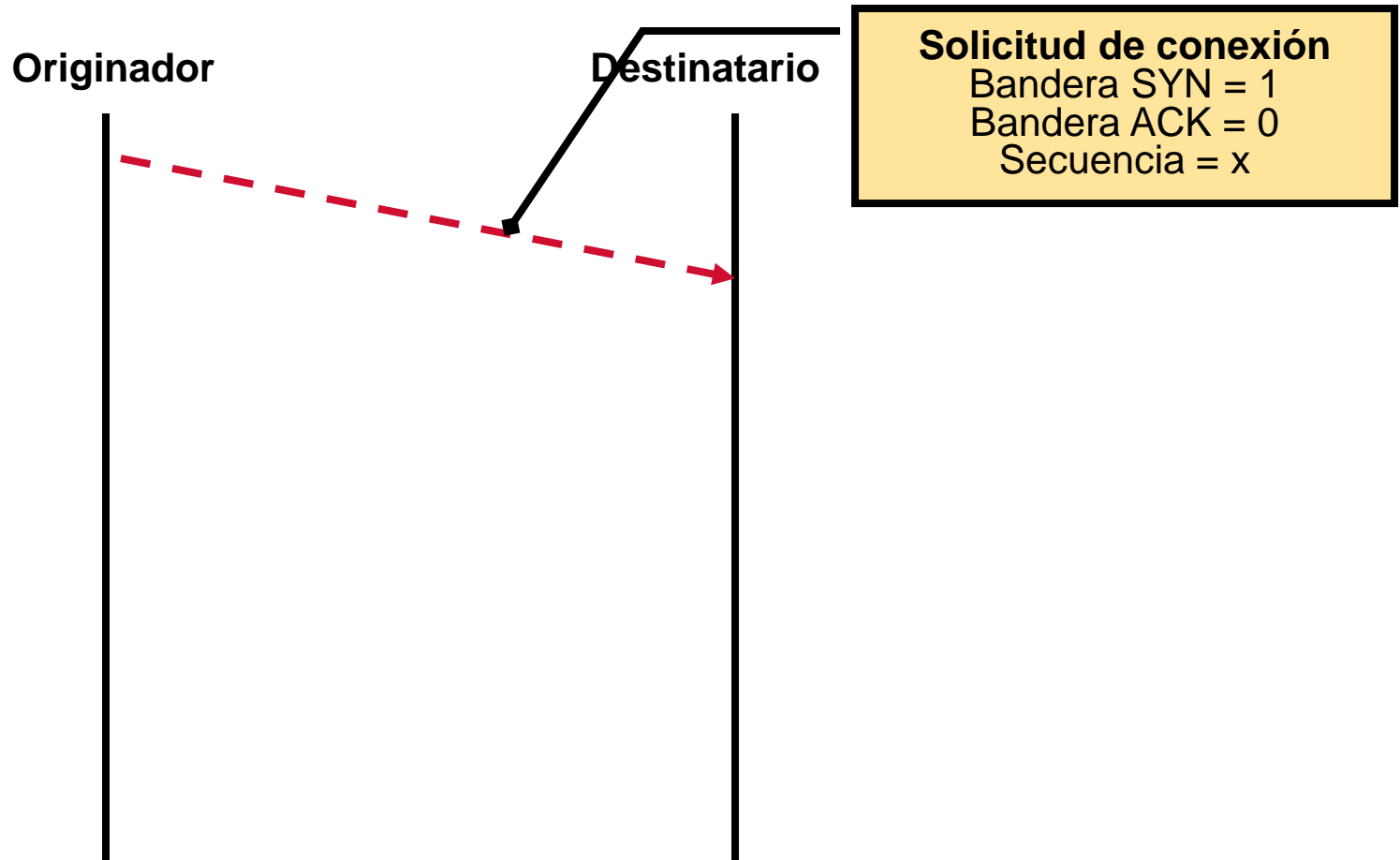
# Establecimiento de conexión

Originador

Destinatario

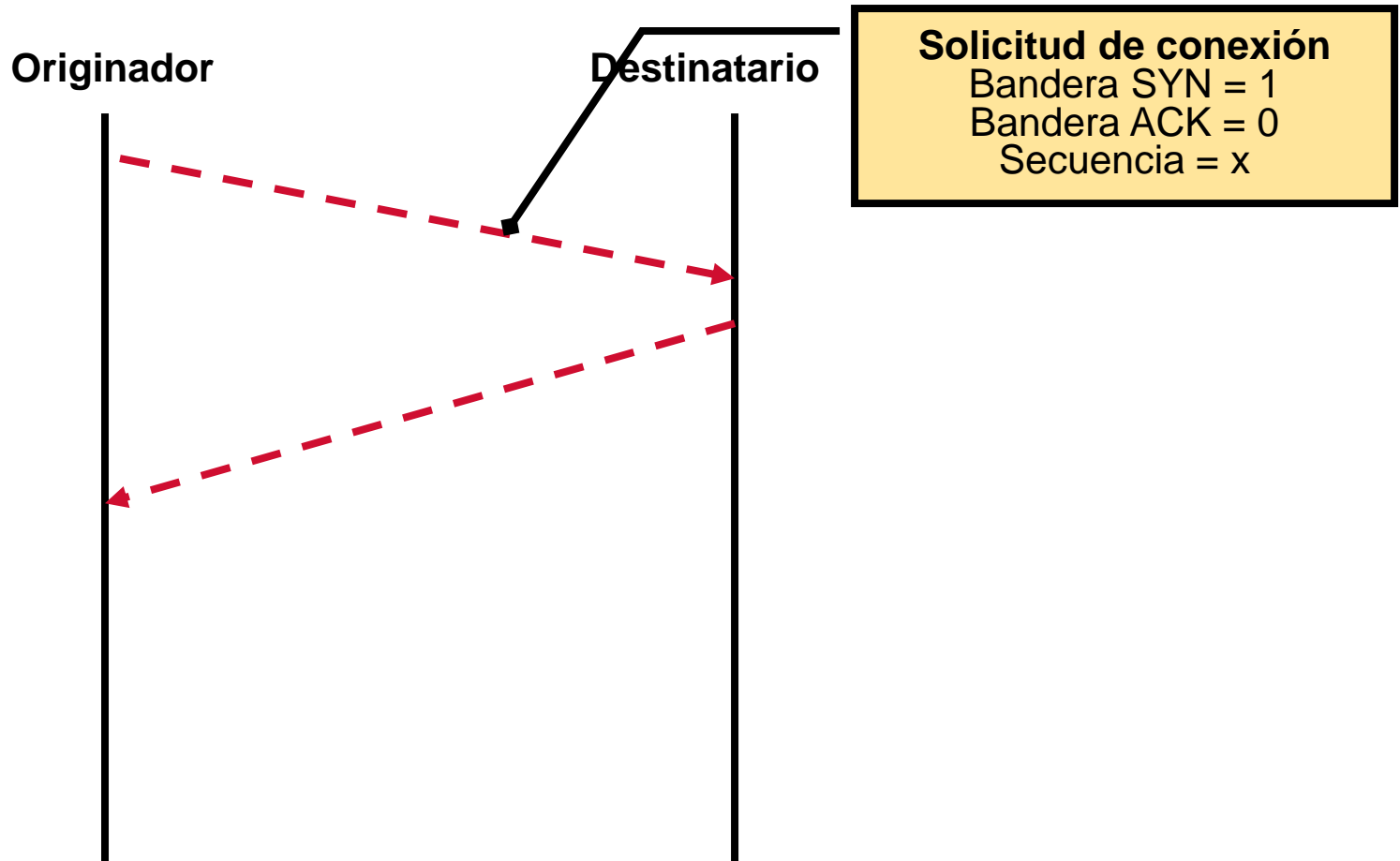


# Establecimiento de conexión

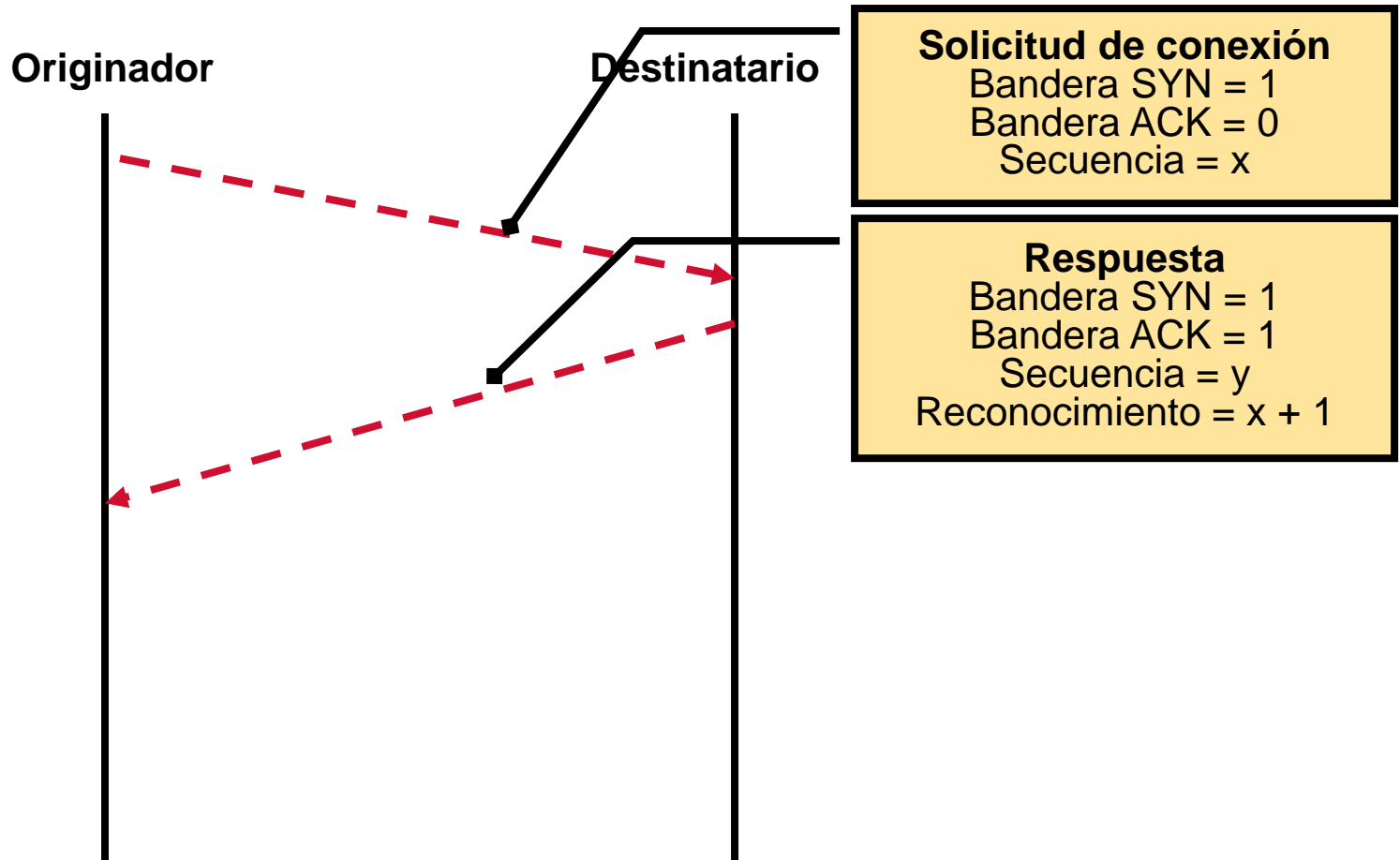




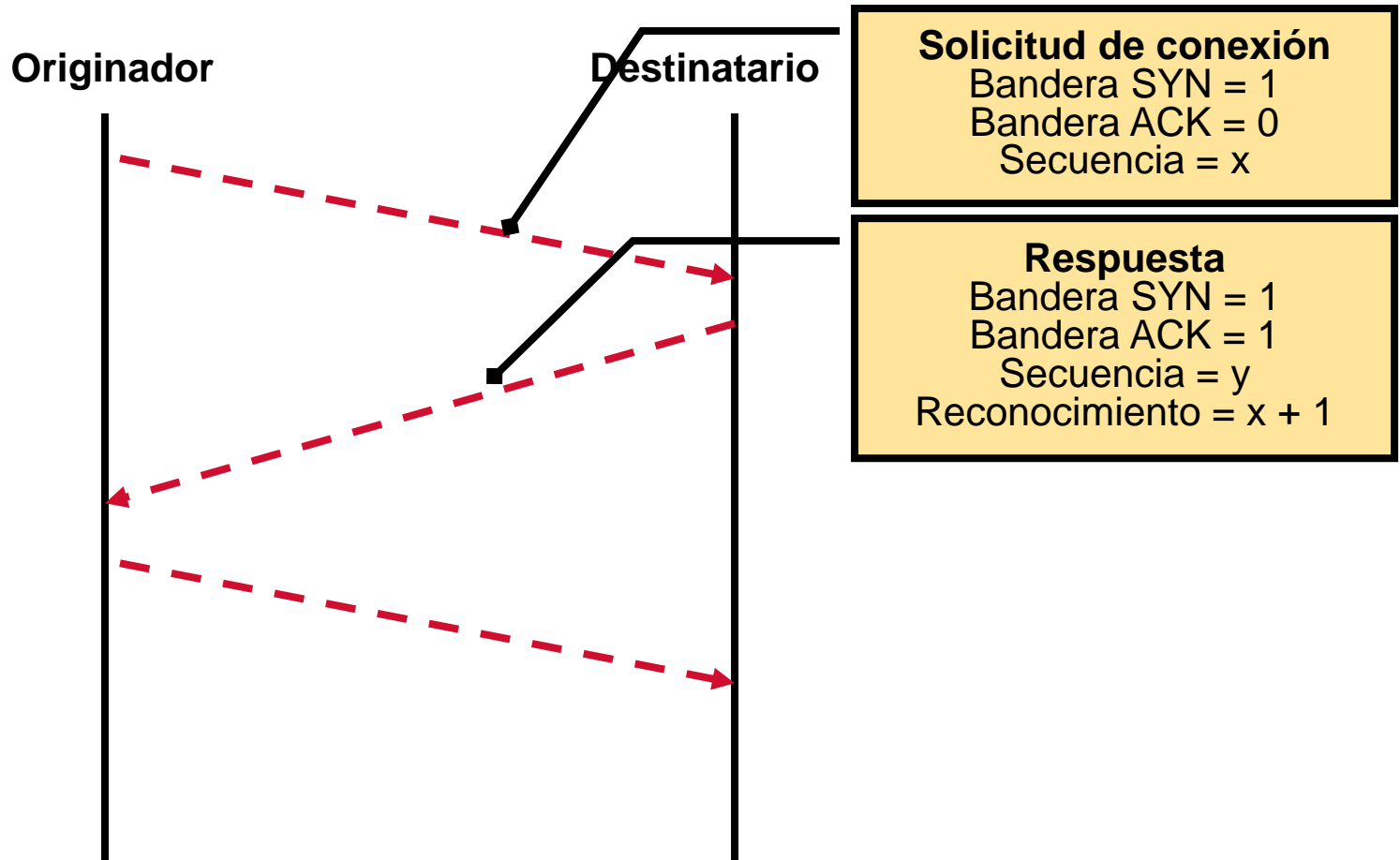
# Establecimiento de conexión



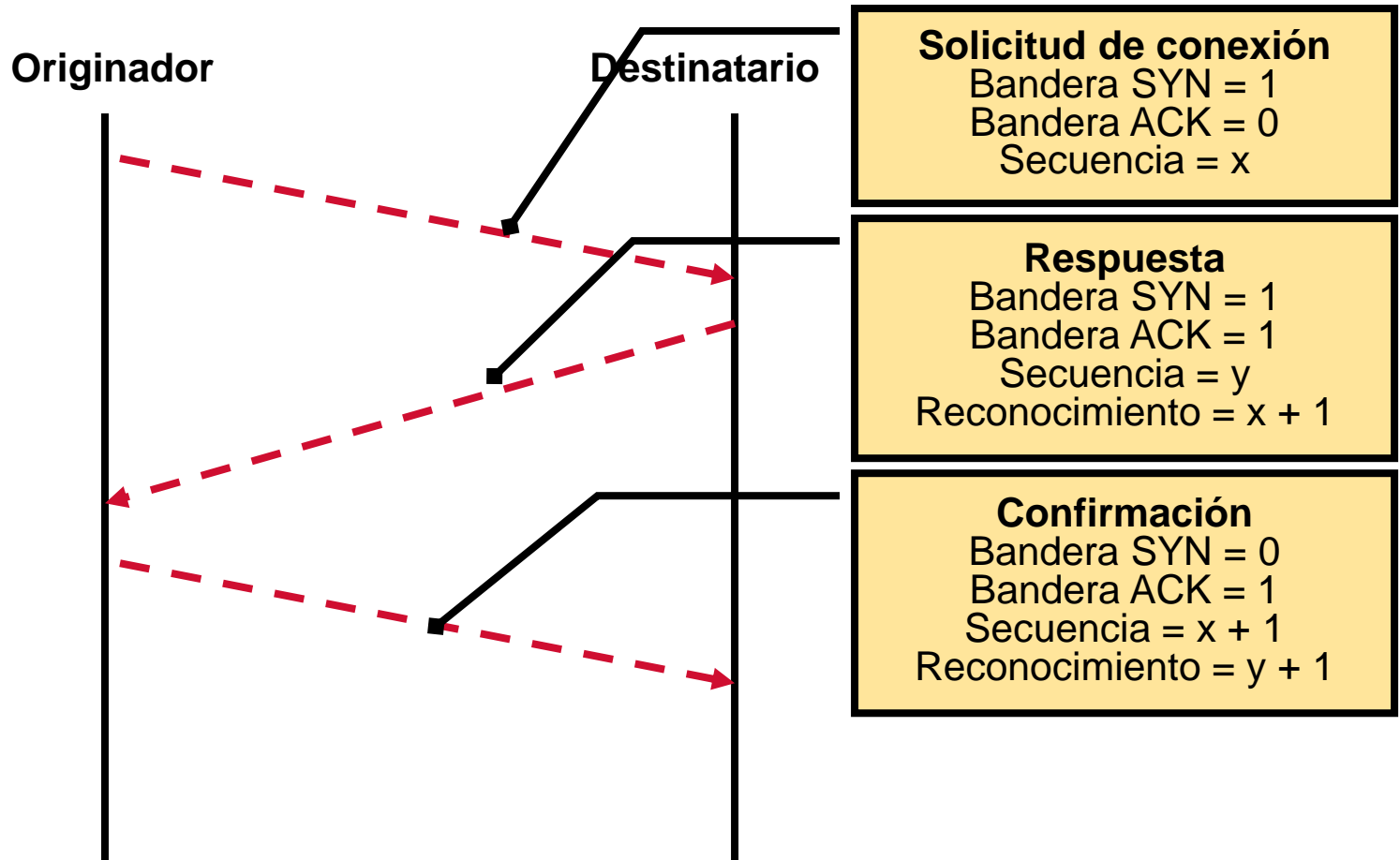
# Establecimiento de conexión



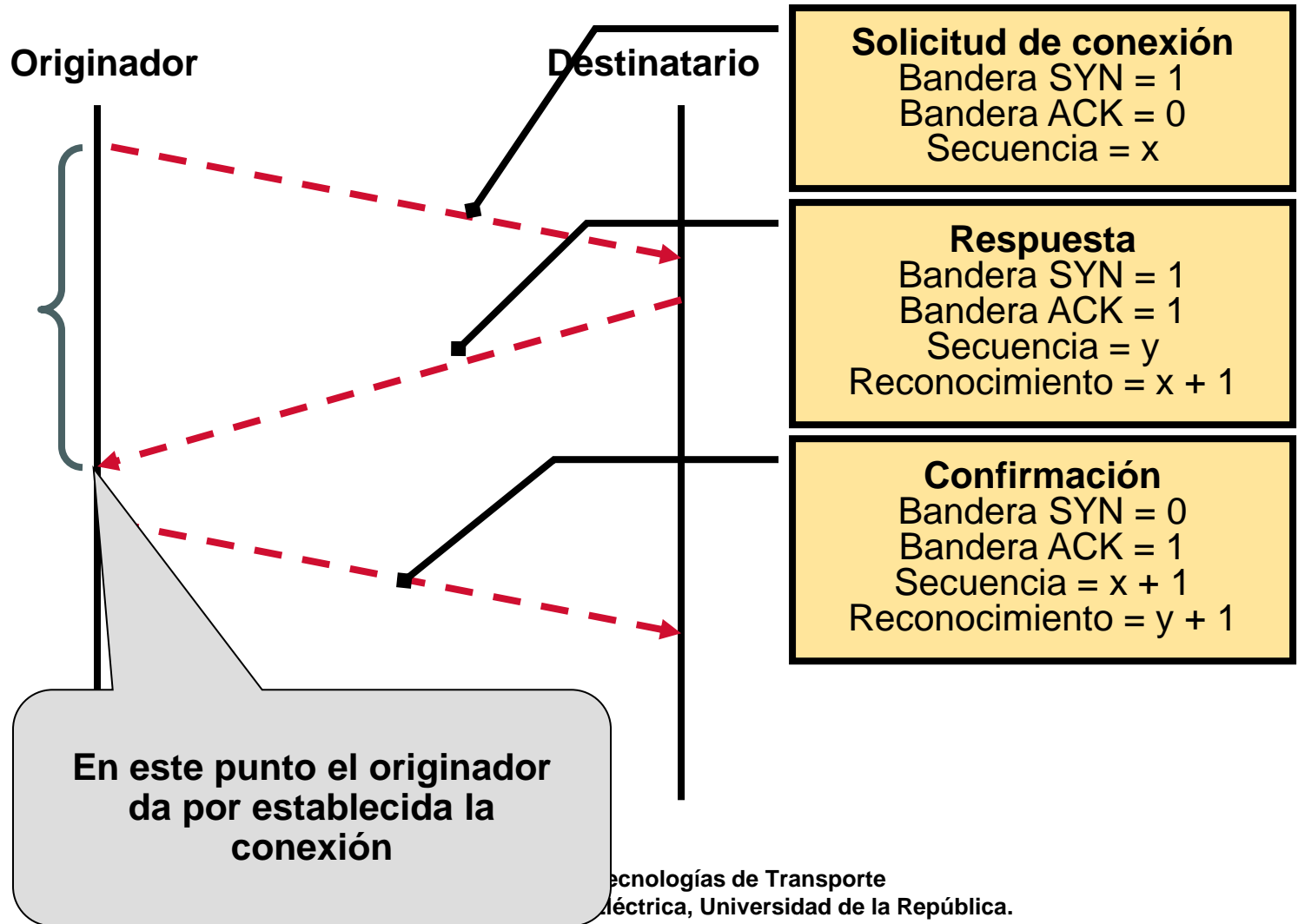
# Establecimiento de conexión



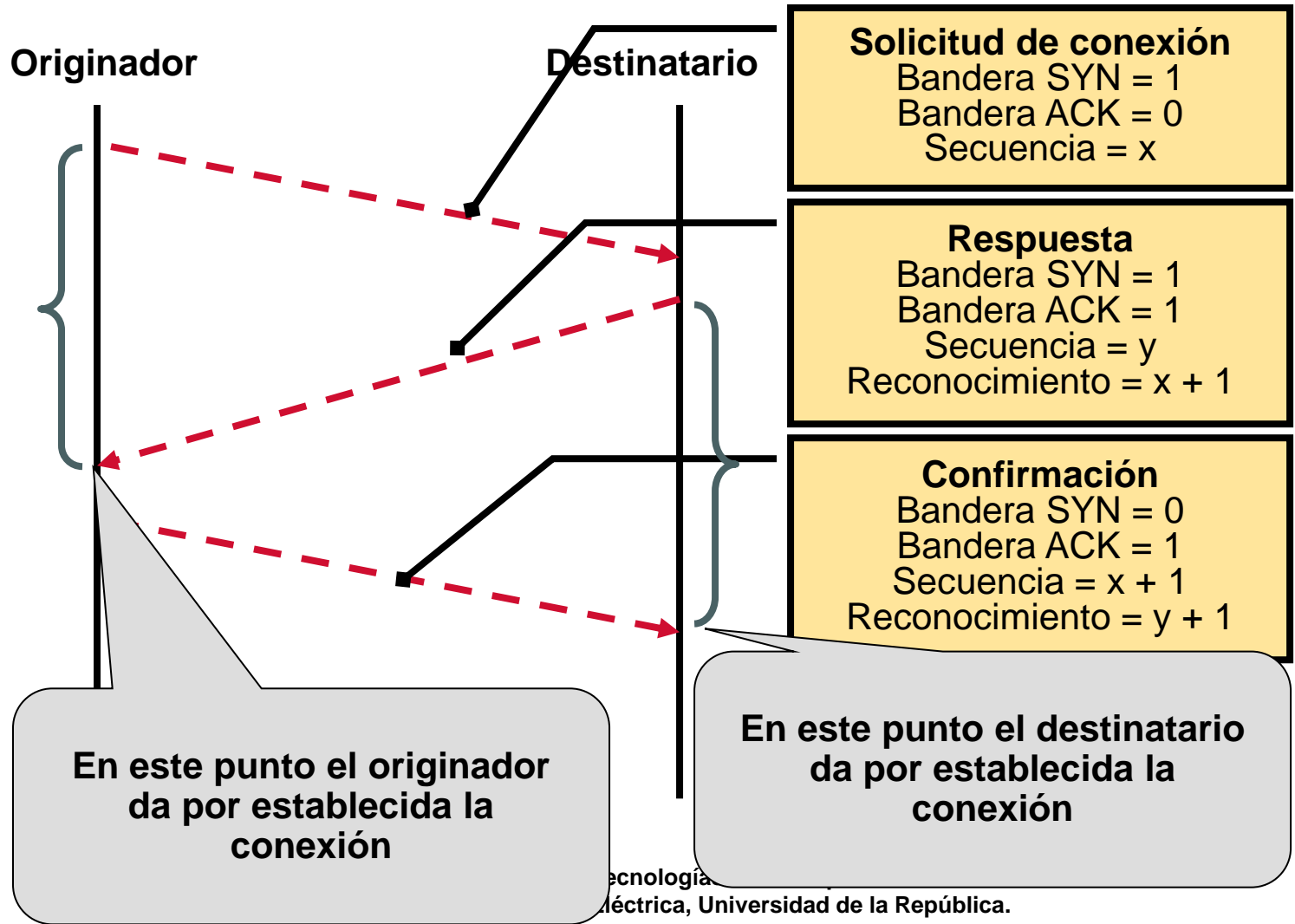
# Establecimiento de conexión



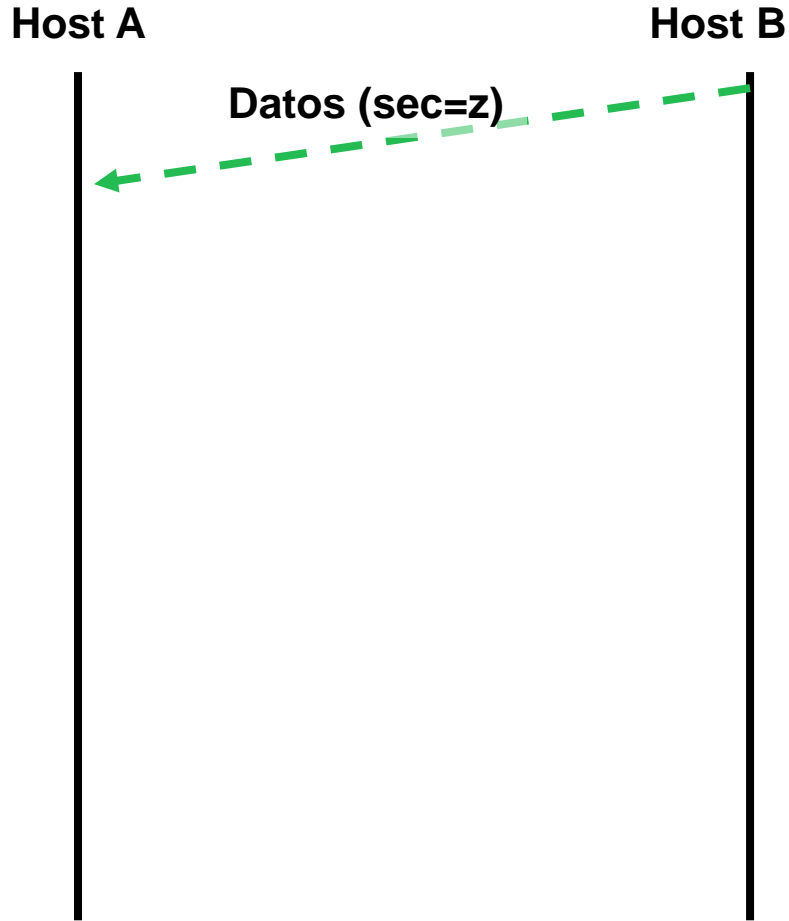
# Establecimiento de conexión



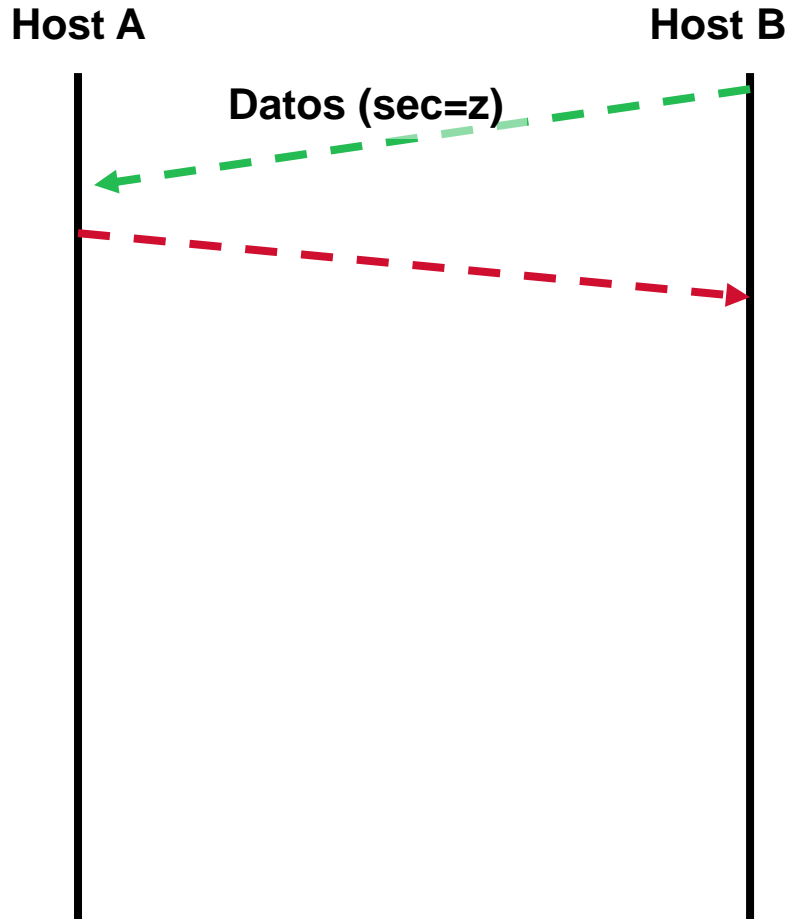
# Establecimiento de conexión



# Corte de conexión

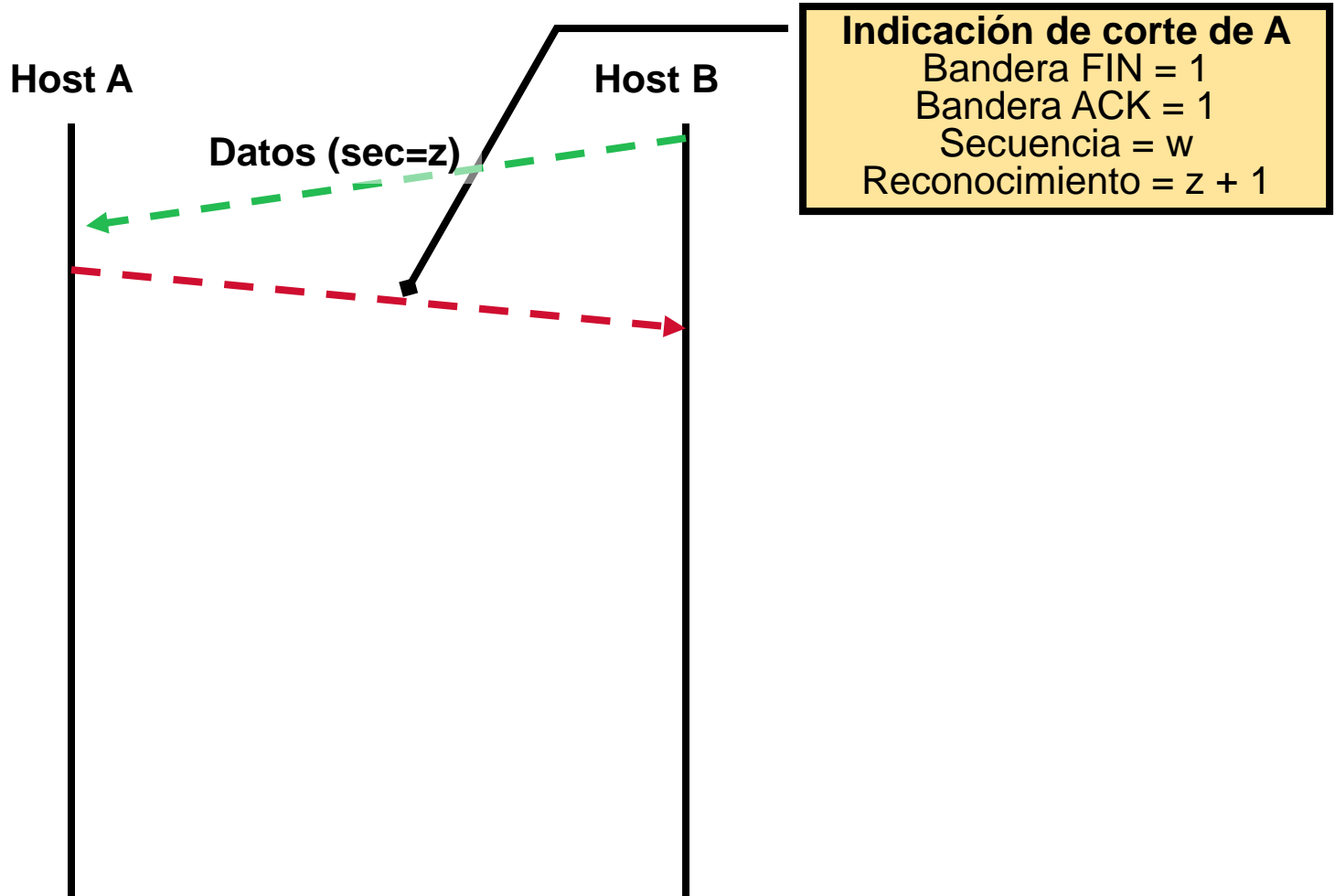


# Corte de conexión

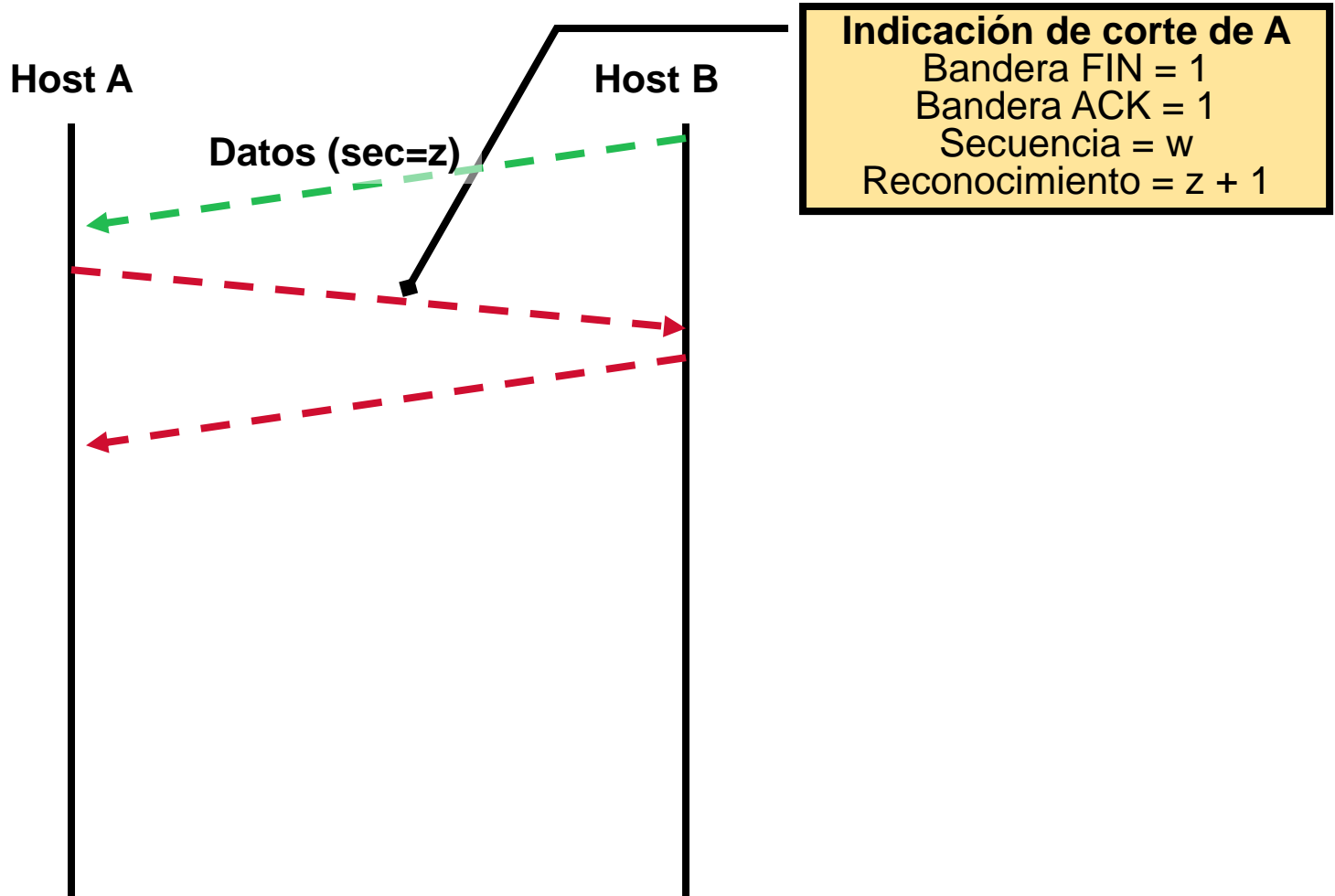




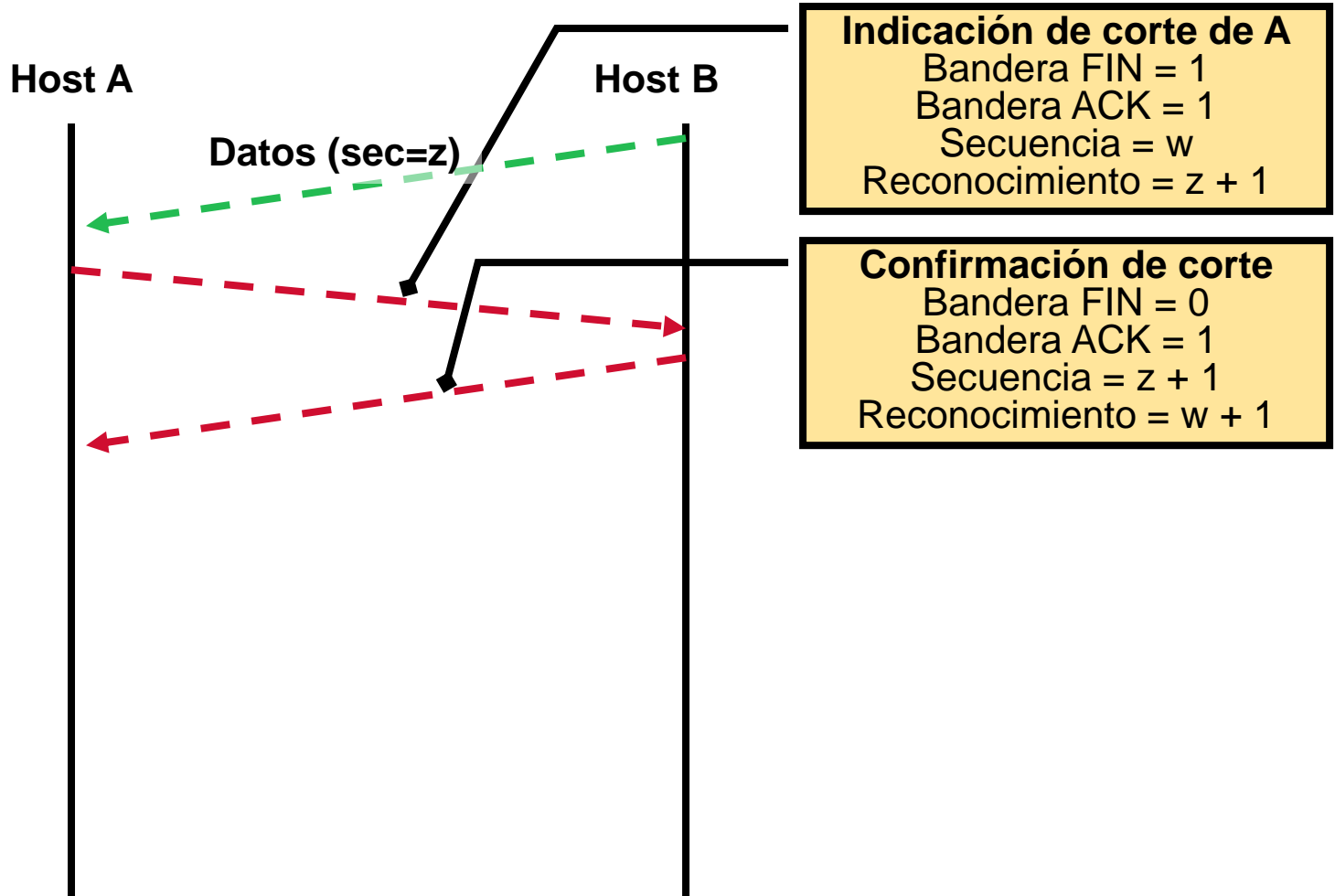
# Corte de conexión



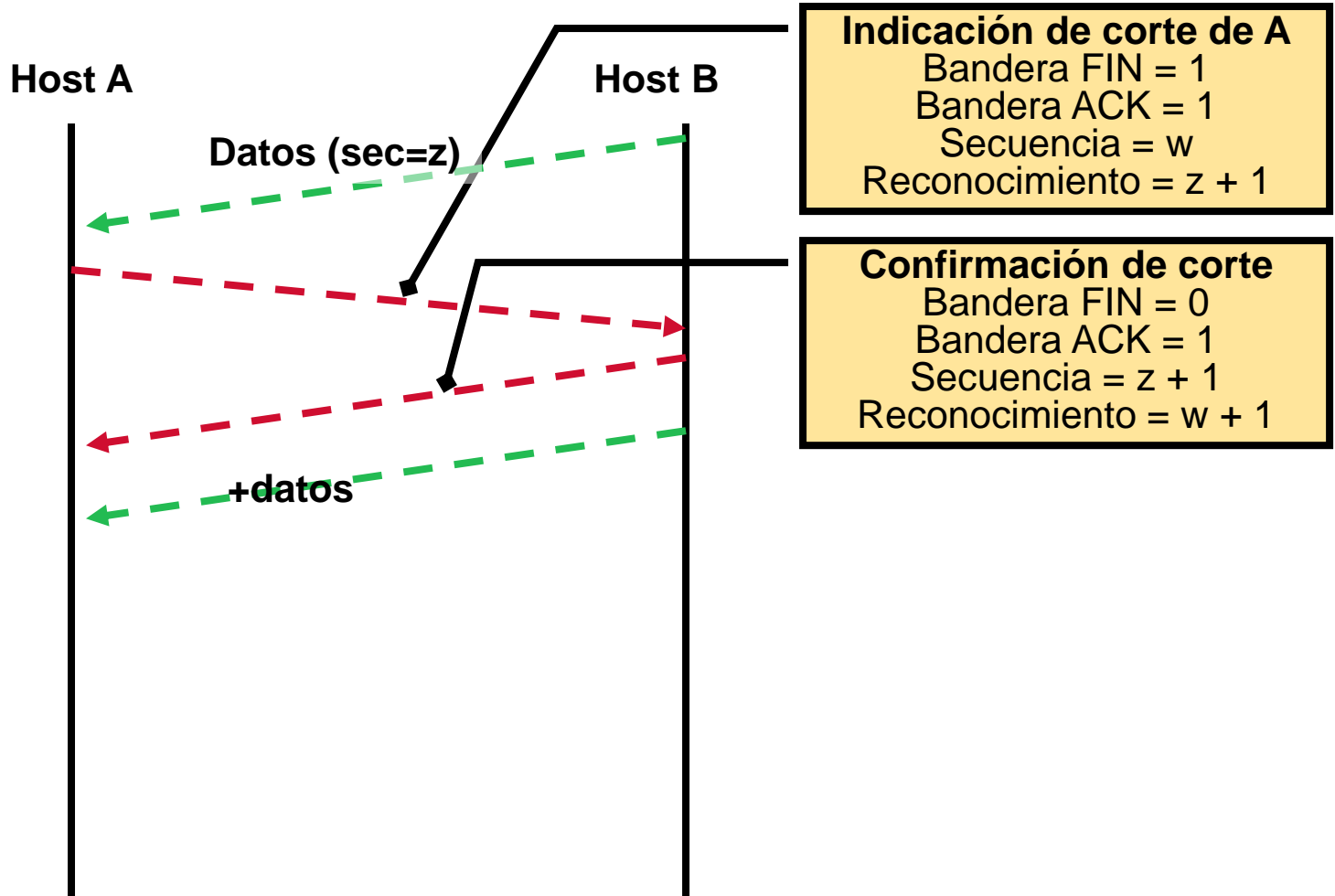
# Corte de conexión



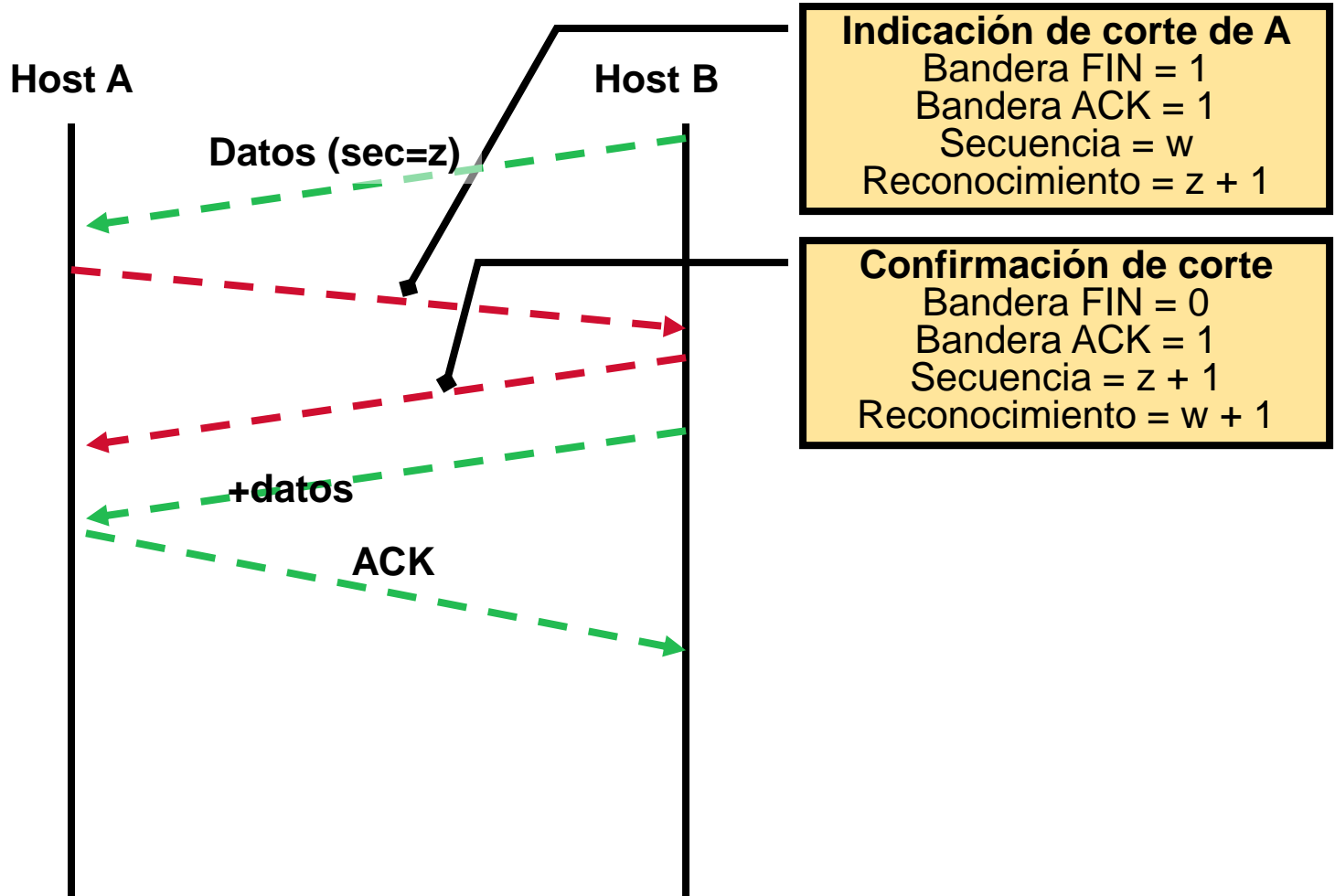
# Corte de conexión



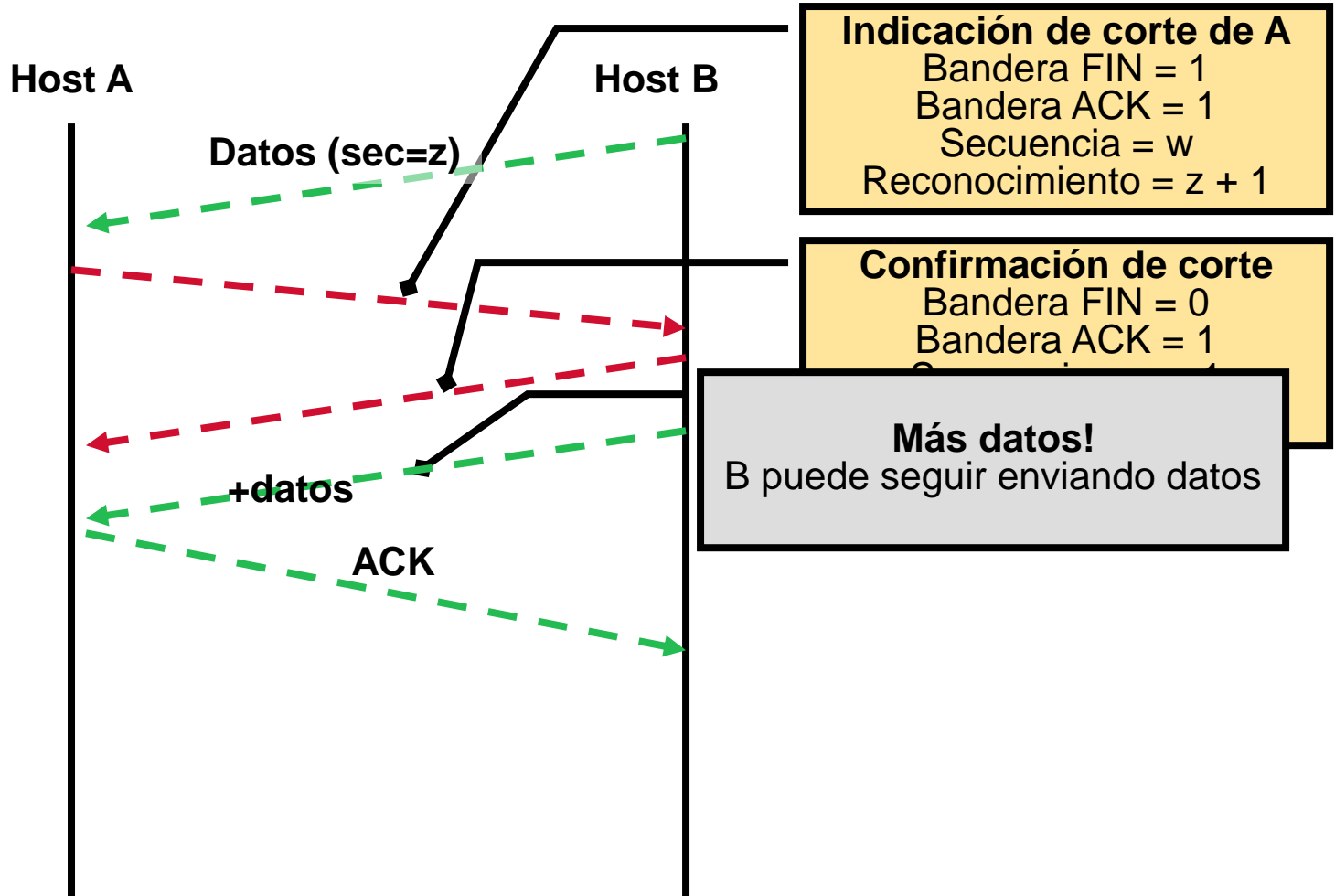
# Corte de conexión



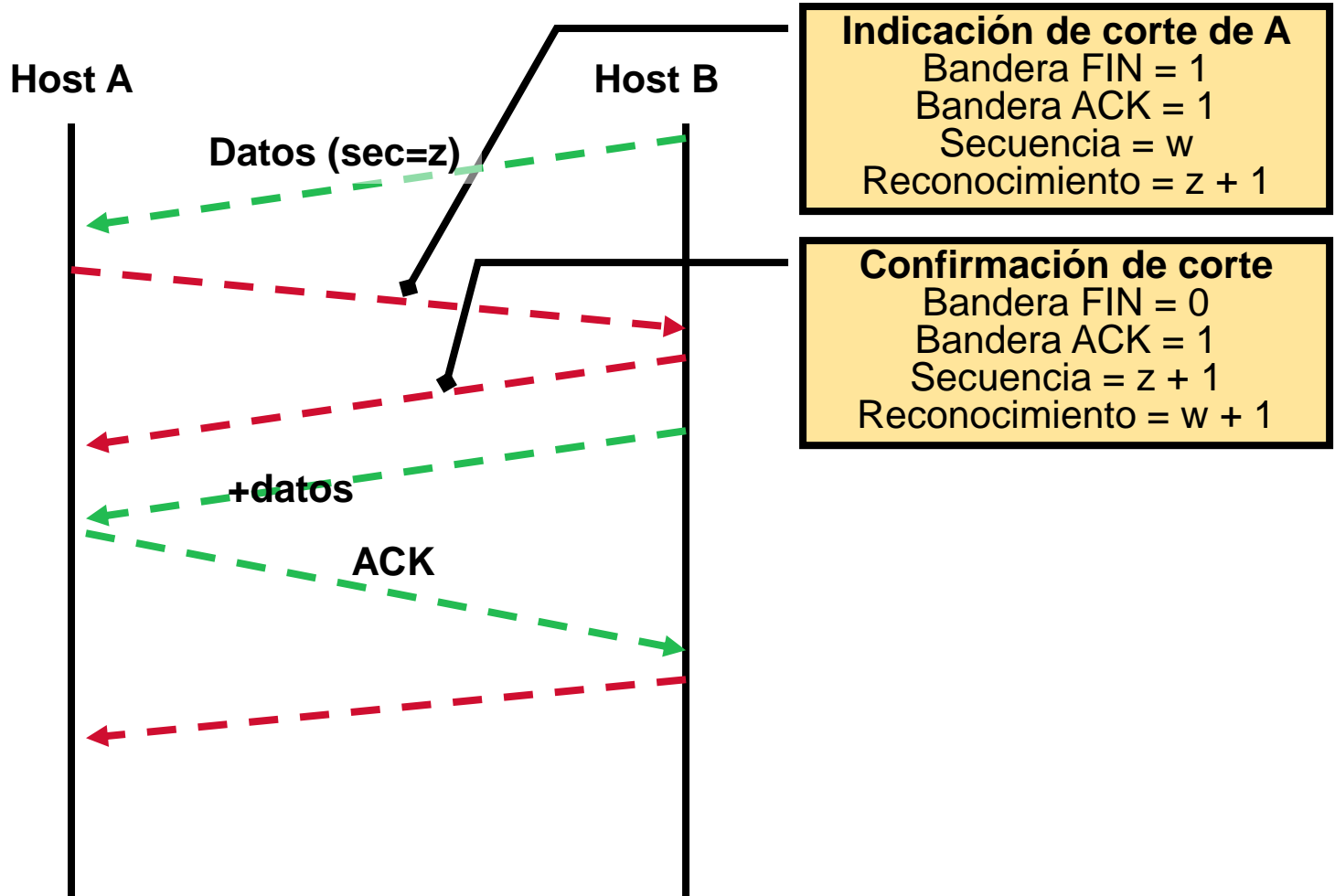
# Corte de conexión



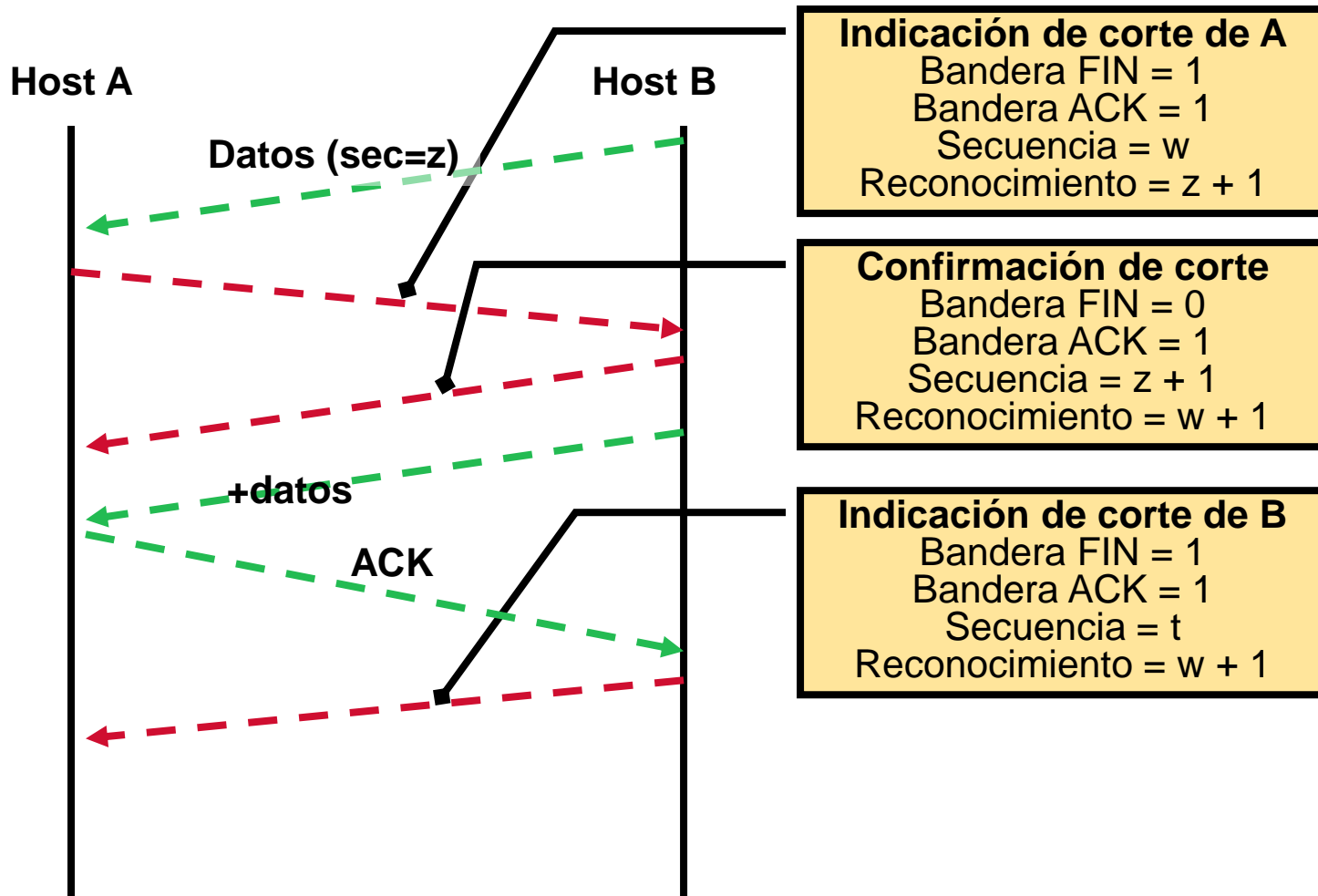
# Corte de conexión



# Corte de conexión

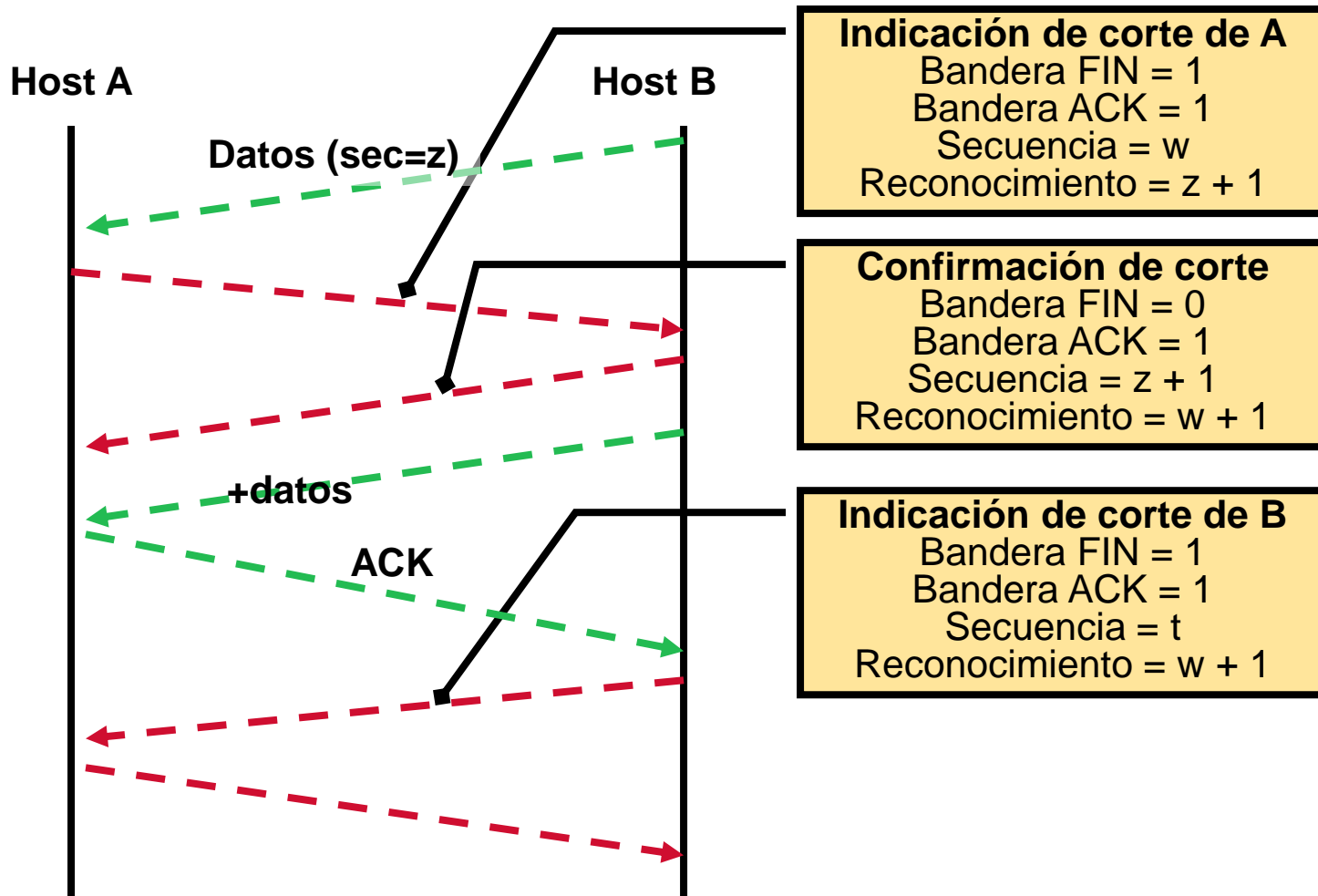


# Corte de conexión

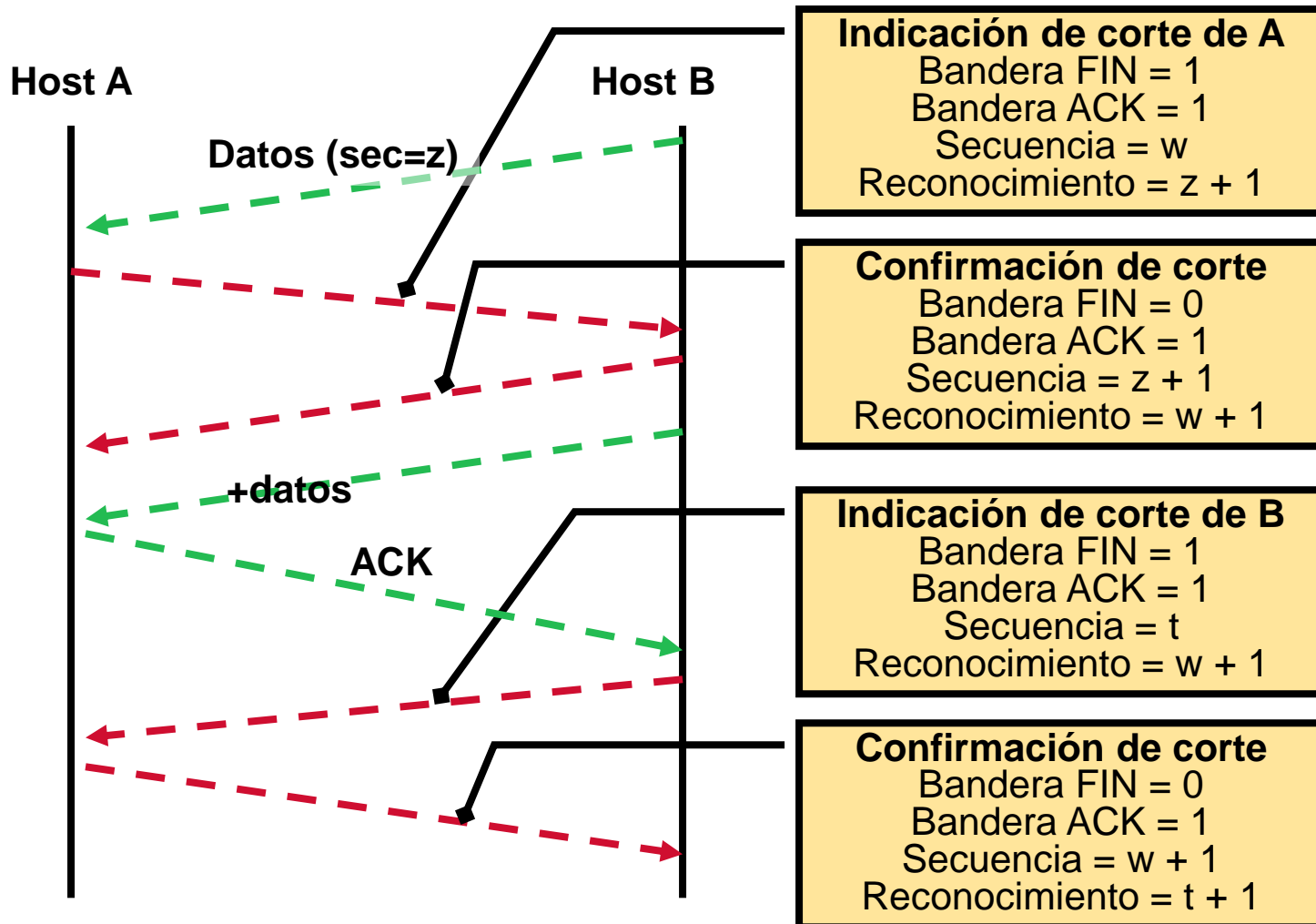




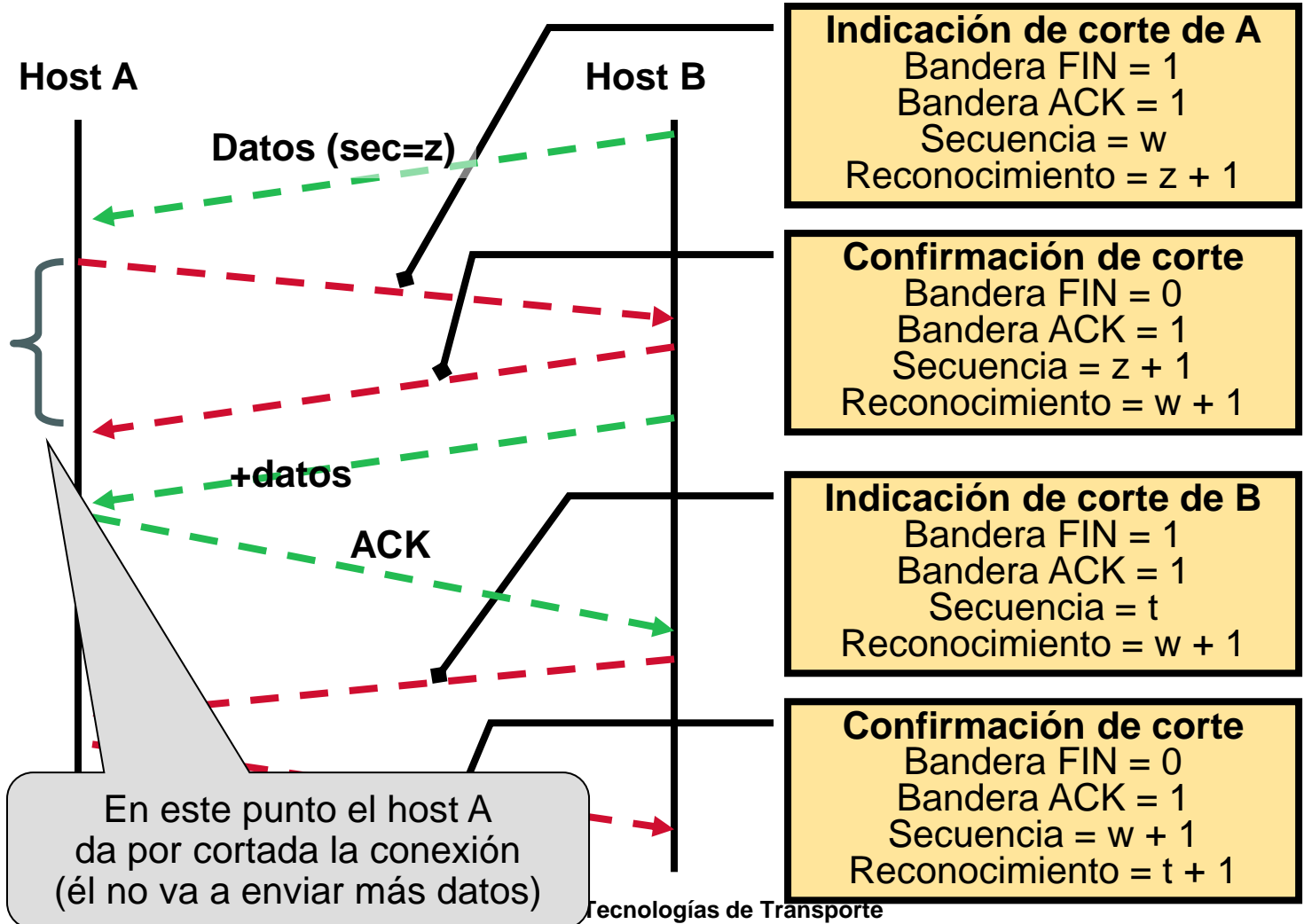
# Corte de conexión



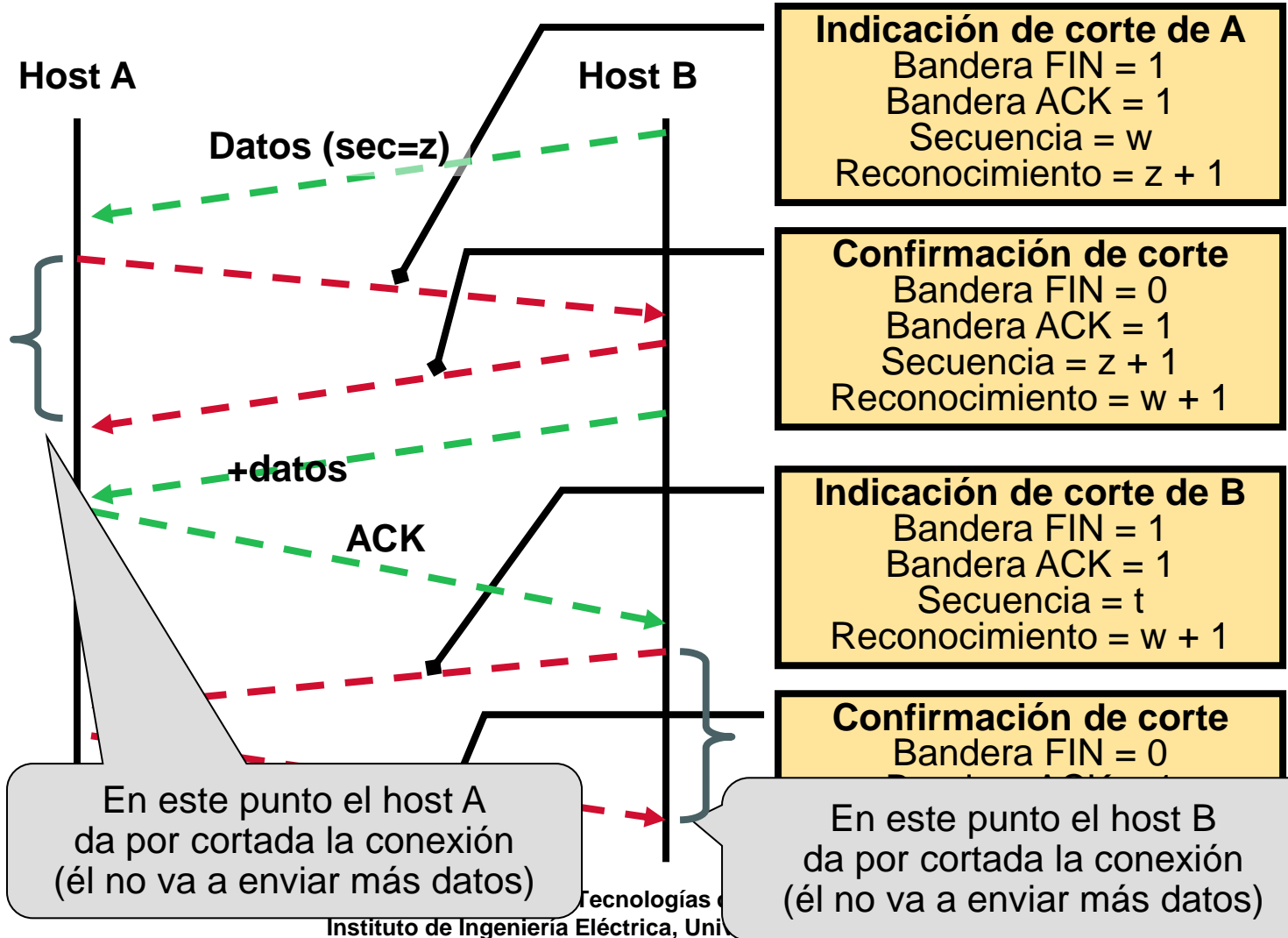
# Corte de conexión



# Corte de conexión



# Corte de conexión



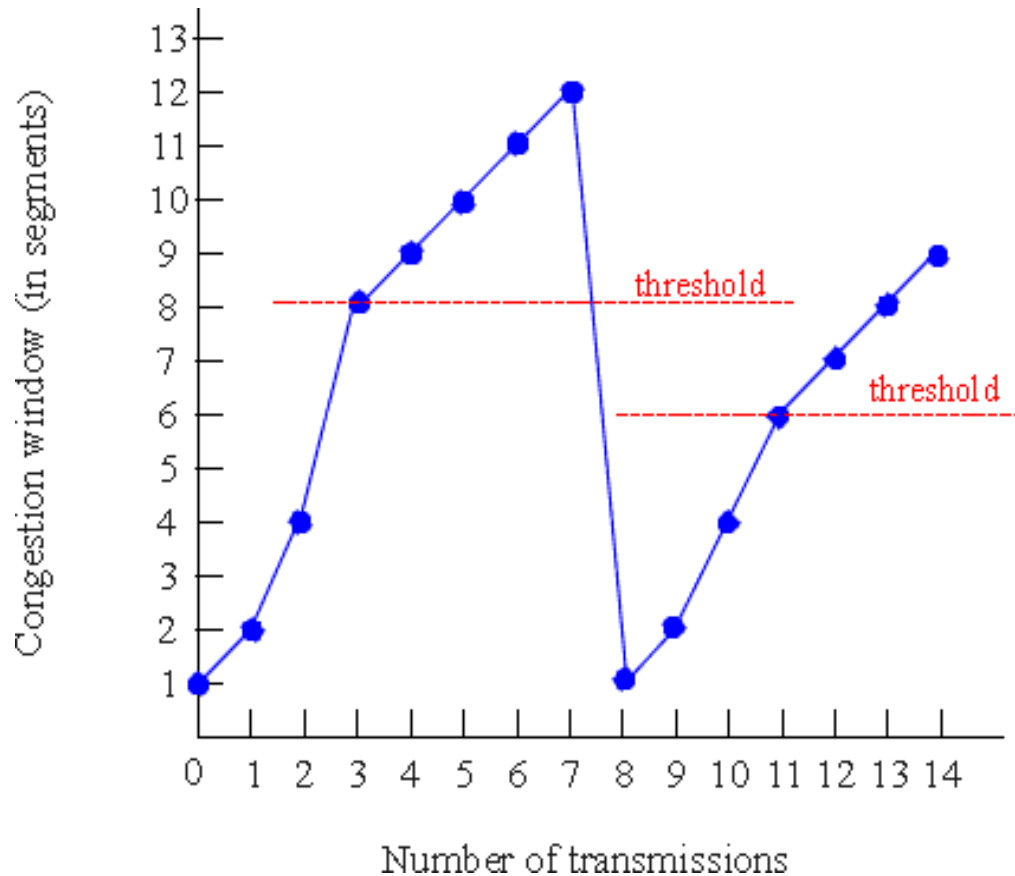
# Control de flujo

- **Se utiliza un protocolo de ventana variable**
- **Cada segmento tiene una indicación del tamaño de la ventana disponible en recepción**
- **El transmisor no puede utilizar un tamaño en su ventana de transmisión mayor a la capacidad indicada por el tamaño de la ventana del receptor**

# Control de congestión

- **Para controlar la congestión hay que tener realimentación de la red**
- **TCP supone que si el reconocimiento de un segmento no llega dentro del tiempo establecido, se debe a que la red está congestionada**
- **En esta hipótesis, la ventana del transmisor se ajusta ante la ocurrencia de timeouts en los reconocimientos**

# Evolución de la ventana de transmisión



Origen de la imagen: [http://www.postech.ac.kr/cse/hpc/research/webcache/book/transport\\_layer/congestion.html](http://www.postech.ac.kr/cse/hpc/research/webcache/book/transport_layer/congestion.html)

# Algoritmos

- **Slow start**

Arranque exponencial

- **Congestion avoidance**

Fase lineal

- **Fast retransmit y Fast recovery**

Cuando se detectan 3 ACKs repetidos se asume pérdida de segmento y no congestión

Se retransmite sin esperar el vencimiento del timeout

Se sigue con congestion avoidance y no slow start



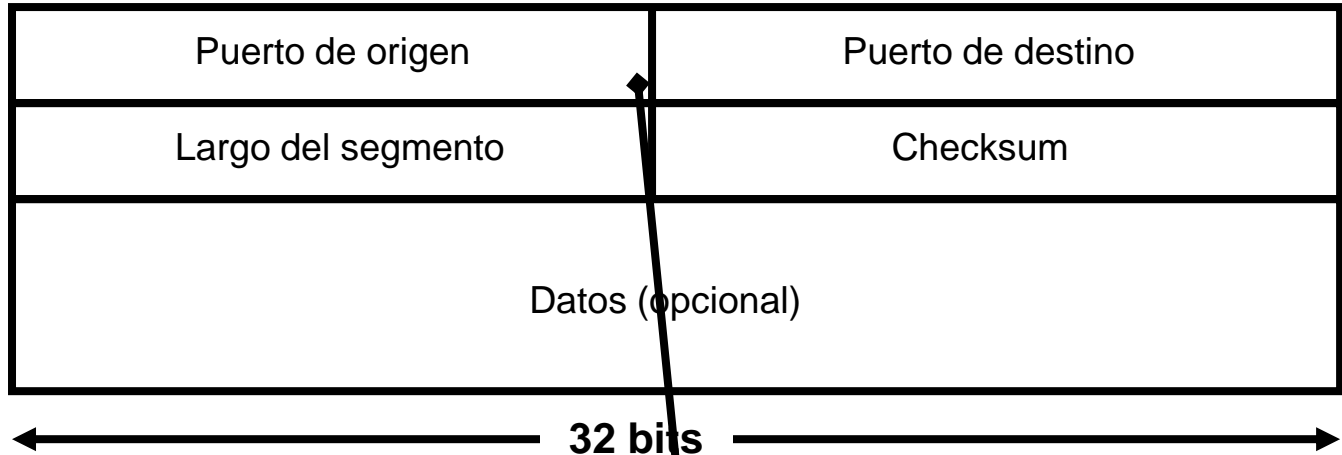
# Protocolo UDP

- **Protocolo de transporte no orientado a conexión y no confiable**
- **La única información de capa de transporte que requiere son los puertos de origen y destino**

# Formato del segmento UDP



# Formato del segmento UDP



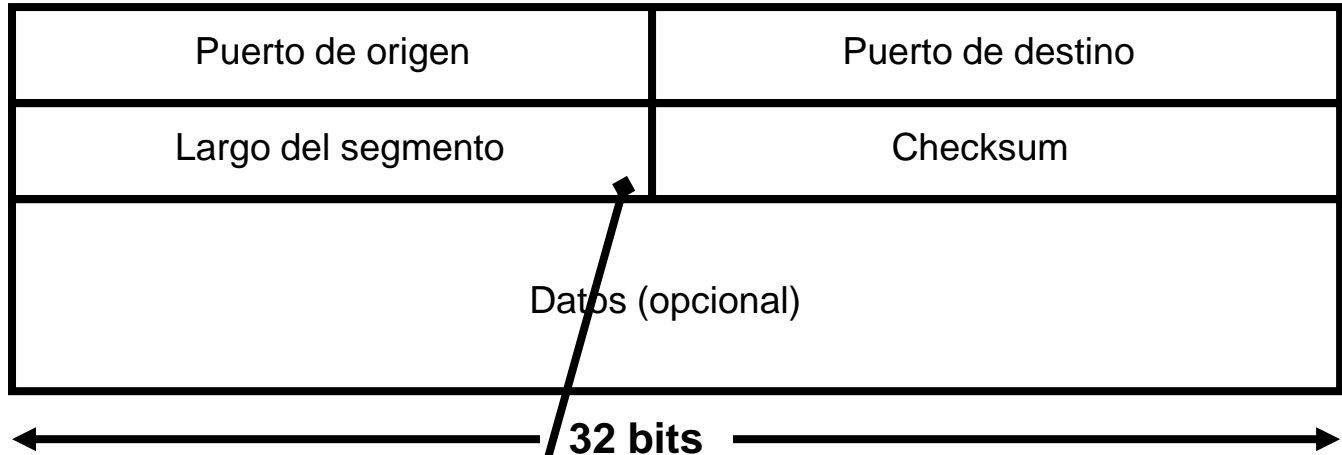
**Puerto de origen  
(16 bits)**  
Identificador de la entidad de capa 4  
que origina el segmento en la  
máquina origen

# Formato del segmento UDP



**Puerto de destino  
(16 bits)**  
Identificador de la entidad de capa 4  
destinataria del segmento en la  
máquina destino

# Formato del segmento UDP



**Largo del segmento  
(16 bits)**

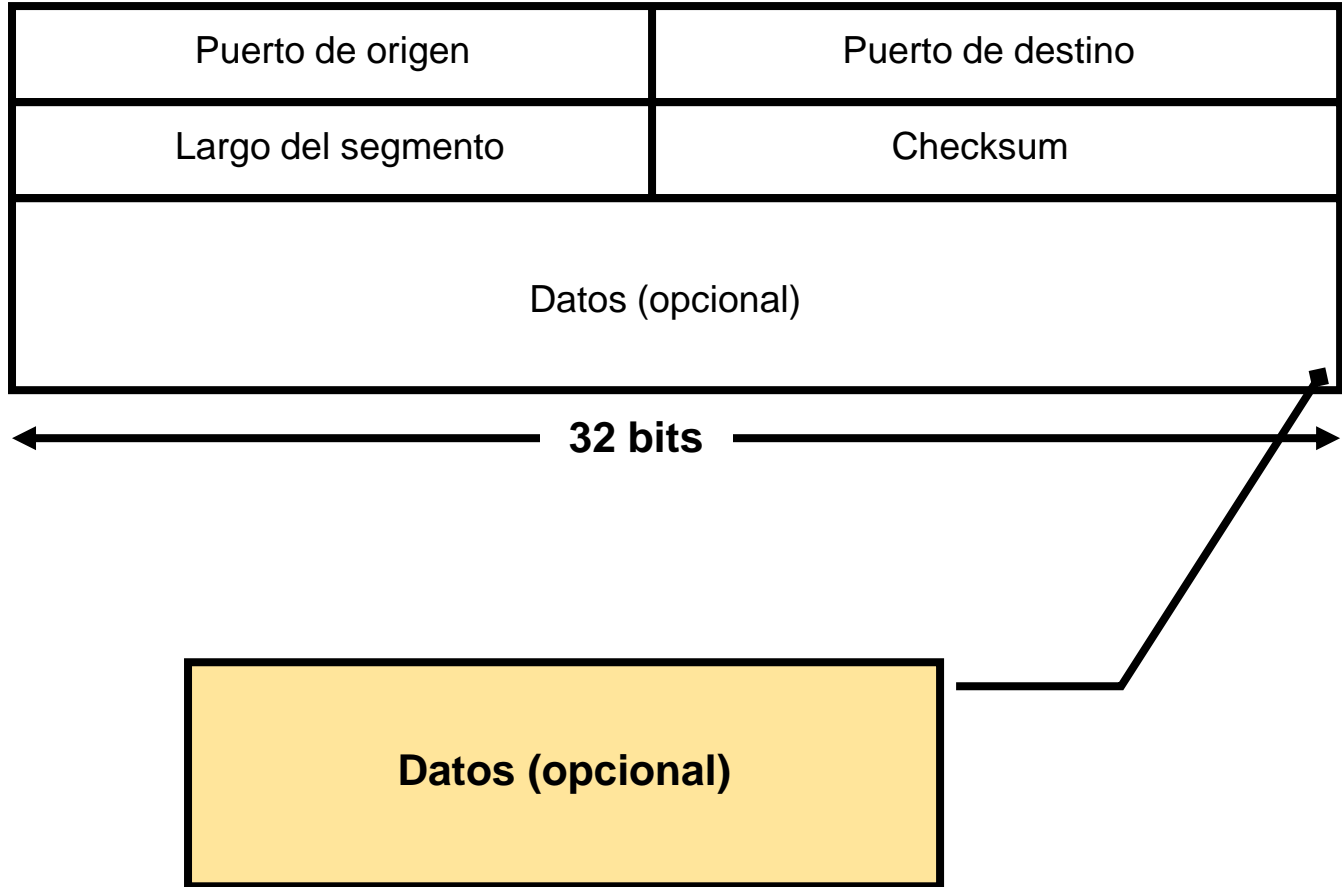
Cantidad de bytes del  
segmento incluyendo  
encabezado y datos

# Formato del segmento UDP



**Checksum (16 bits)**  
Suma en complemento a 1 de las palabras de 16 bits del encabezado, los datos y el pseudoheader. Se toma el complemento a 1 del resultado

# Formato del segmento UDP



# Formato del segmento UDP

