

# Protocolo TCP

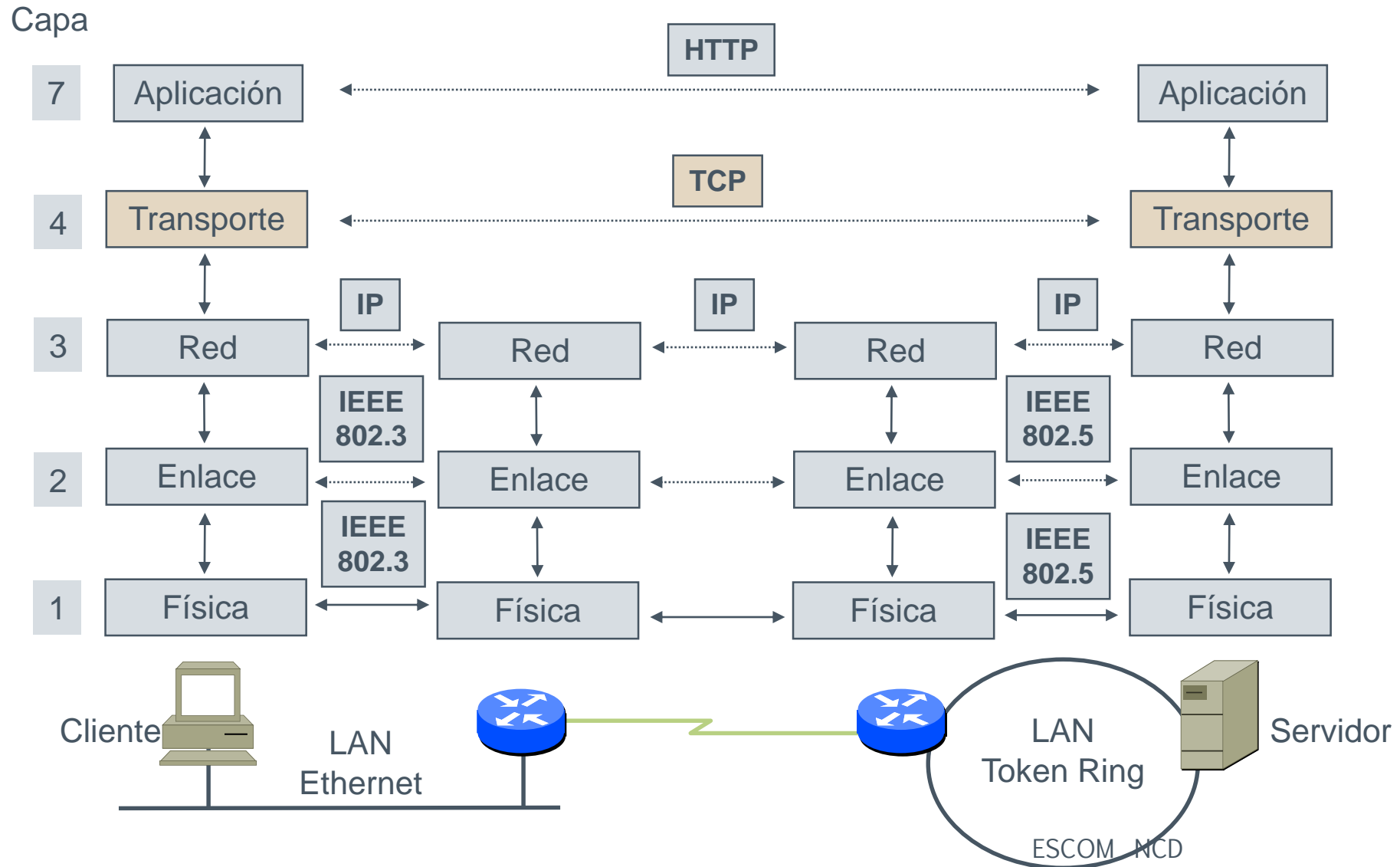
## El Protocolo de Control de Transmisión

# Introducción

Protocolo de la **capa de Transporte** que proporciona un servicio de entrega **confiable** de transferencia de datos **de extremo a extremo**.



# Acceso a un servidor Web a través de una conexión remota



# Características

## – Orientado a conexión

- › Antes de poder transferir los datos, dos procesos (local y remoto) deben negociar una conexión TCP mediante un proceso de establecimiento de conexión.
- › Las conexiones TCP se cierran formalmente mediante el proceso de finalización de conexión TCP.

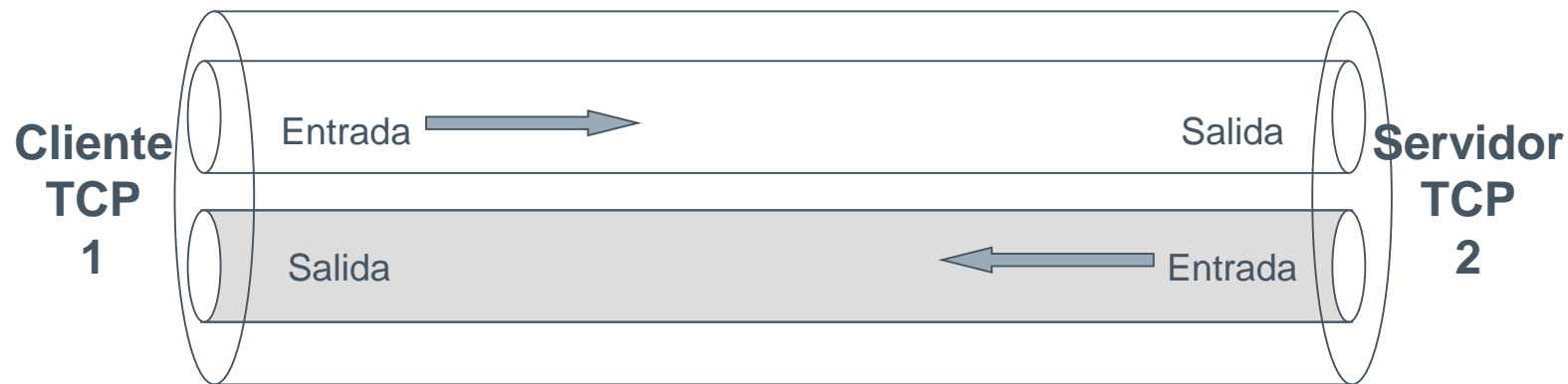
## – Control de flujo del emisor y del receptor.

- › TCP implementa control de flujo del emisor que, gradualmente, escala la cantidad de datos a la vez.
- › Para evitar que el emisor envíe datos que el receptor no puede almacenar en buffer. TCP implementa control de flujo del receptor que indica la cantidad de espacio libre en el buffer del receptor.

# Características

## – Full Dúplex

- › Para cada terminal TCP, la conexión TCP está formada por dos canales lógicos: un canal para transmitir datos (salida) y uno para recibir datos (entrada).
- › Con la tecnología adecuada de la capa de Interfaz de Red, la terminal podría transmitir y recibir datos al mismo tiempo.
- › El encabezado TCP contiene **el número de secuencia** de los datos de salida y un **reconocimiento** de los datos de entrada.



Un circuito lógico bidireccional, full dúplex entre dos procesos

# Características

## – Fiable

- › En el **transmisor**, los datos enviados en una conexión TCP están secuenciados y se espera un reconocimiento afirmativo por parte del receptor.
- › Si no se recibe ningún reconocimiento, el segmento se transmite de nuevo.
- › En el **receptor**, los segmentos duplicados se descartan y los segmentos que llegan fuera de secuencia se colocan en la secuencia correcta.
- › Siempre se utiliza una **suma de comprobación** TCP para comprobar la *integridad* de nivel de bit del segmento TCP.

# Características

## Segmentación de datos de la capa de aplicación.

- ◆ TCP segmenta los datos obtenidos a partir del proceso de la capa de Aplicación para adaptarlos a un datagrama IP enviado por el enlace de la capa de Interfaz de Red.
- ◆ Las terminales TCP intercambian el segmento de tamaño máximo que puede recibir cada uno y ajustan el tamaño máximo del segmento TCP mediante el descubrimiento de la Unidad Máxima de Transferencia de Rutas (PMTU, Path Maximum Transmission Unit)

# Características

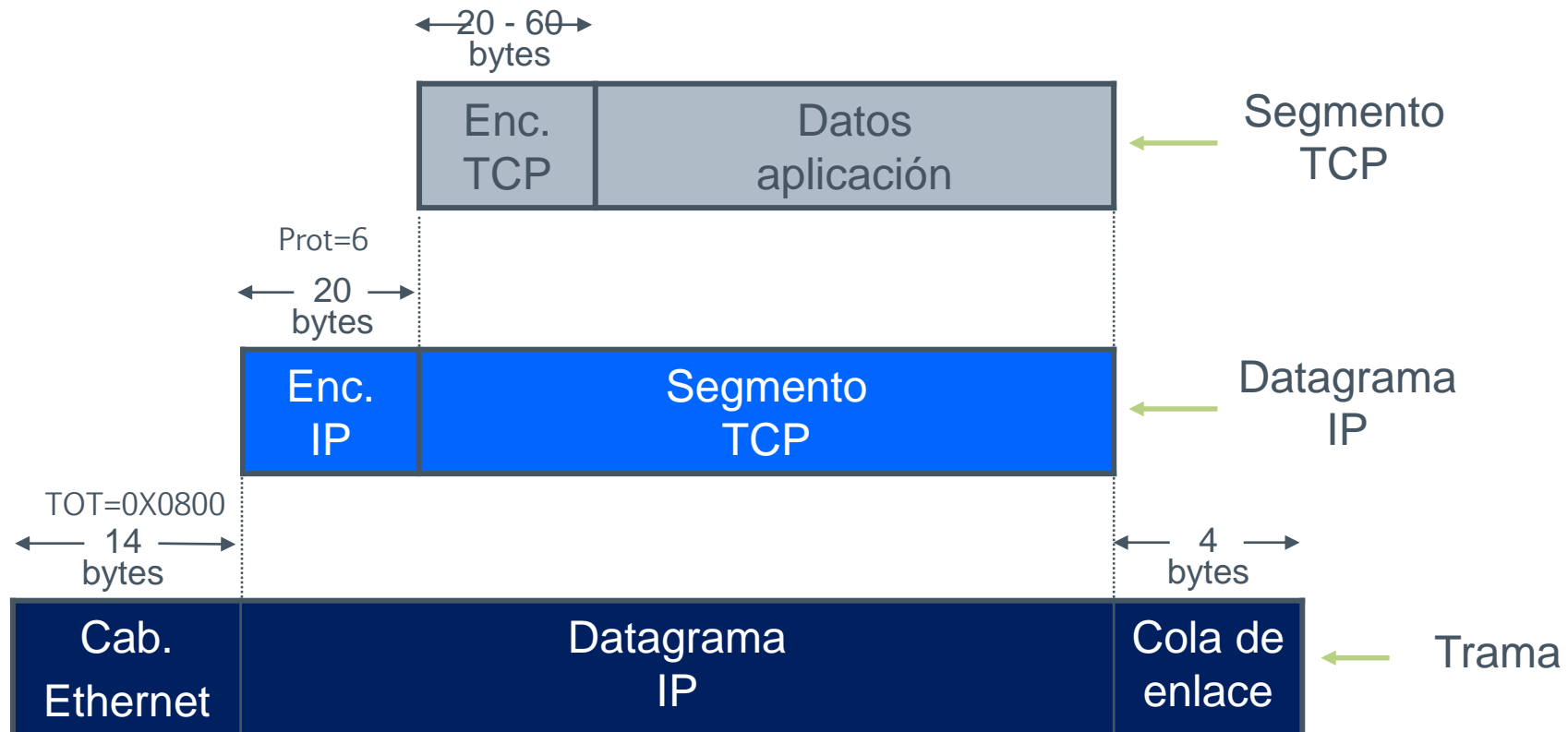
## – Entrega de uno a uno

- › Las conexiones de TCP son un circuito lógico **punto a punto** entre dos protocolos de la capa de Aplicación.
- › TCP no proporciona un servicio de uno a varios.
- › Normalmente, TCP se utiliza cuando el protocolo de la capa de Aplicación requiere un servicio de **transferencia de datos confiable** y el protocolo de Aplicación no proporciona este tipo de servicio.



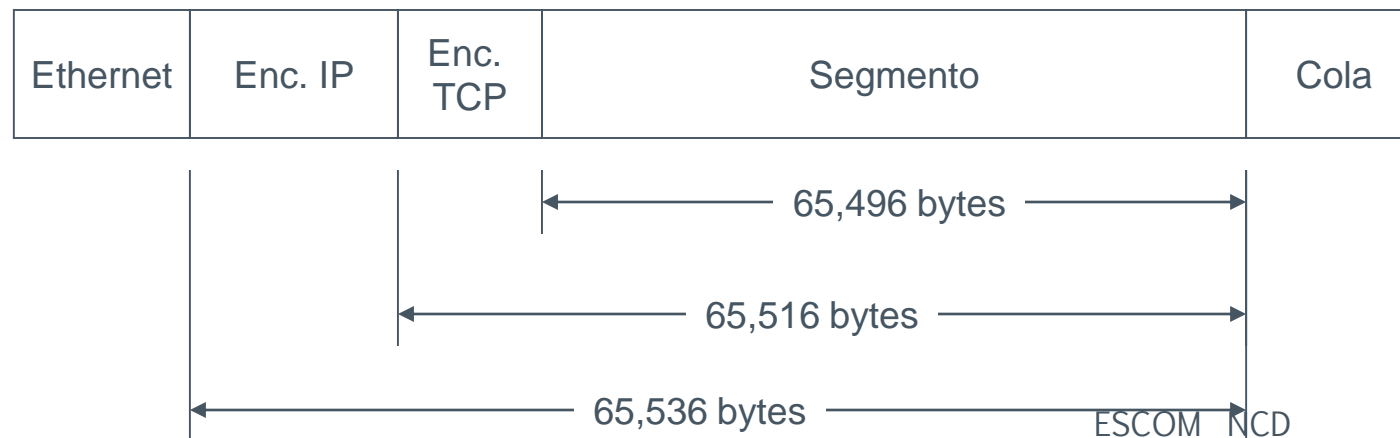
# Segmento TCP

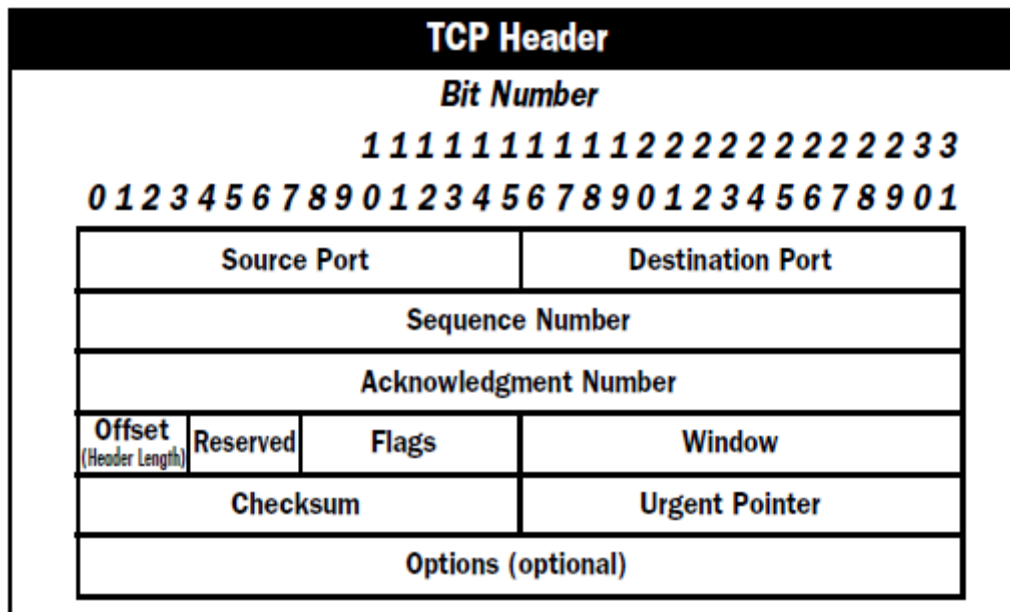
- Los segmentos TCP se envían como datagramas IP.

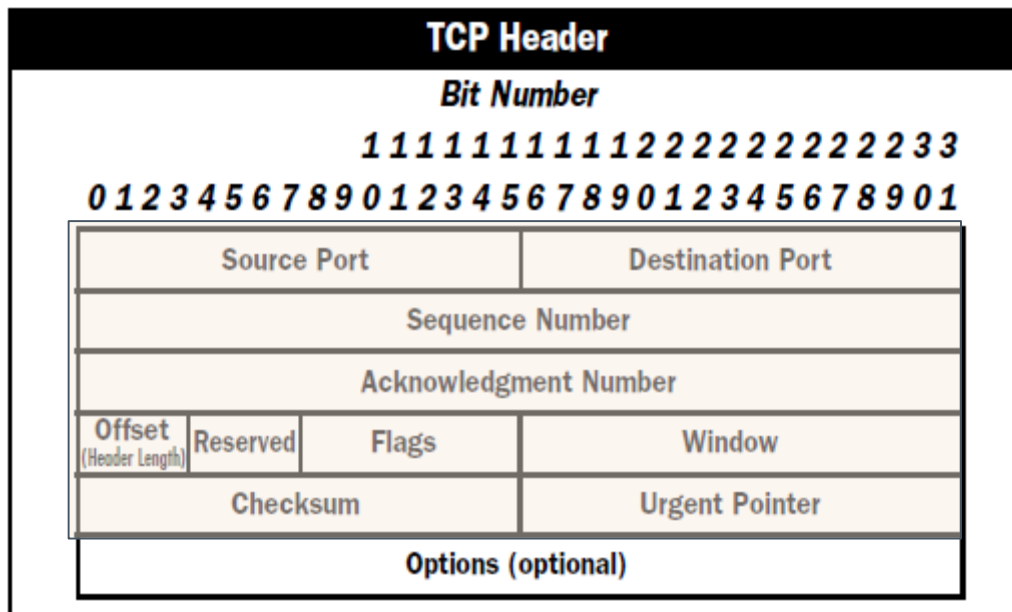


# Segmento TCP

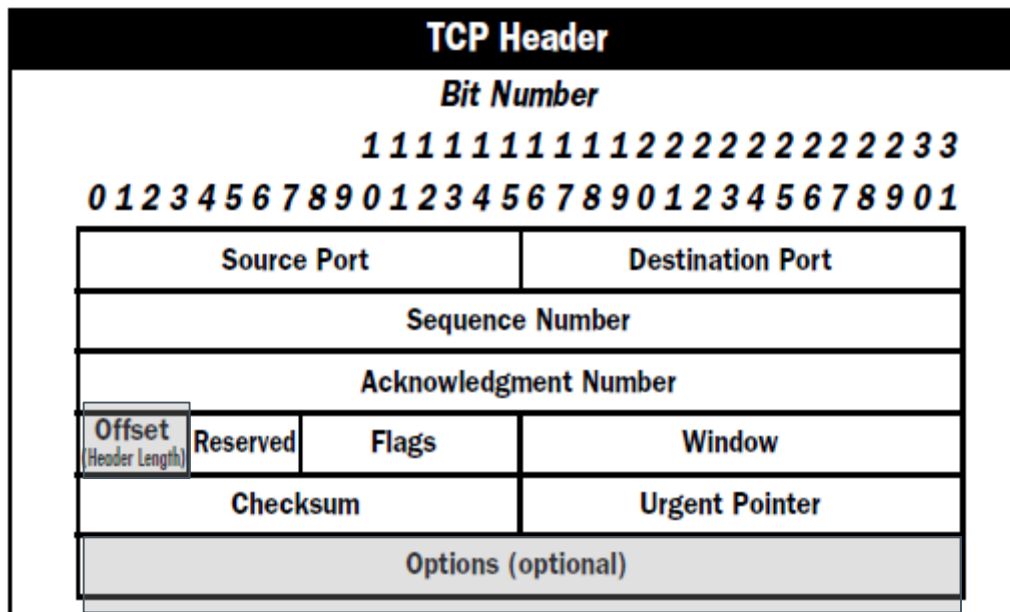
- › Un segmento TCP, esta formado por un encabezado TCP y un segmento.
- › Se encapsula con un encabezado IP con el campo protocolo con el valor 6(0x06).
- › El segmento de datos puede tener un tamaño máximo de 65,496 bytes (65,536 menos 20 de encabezado IP y 20 de encabezado TCP).







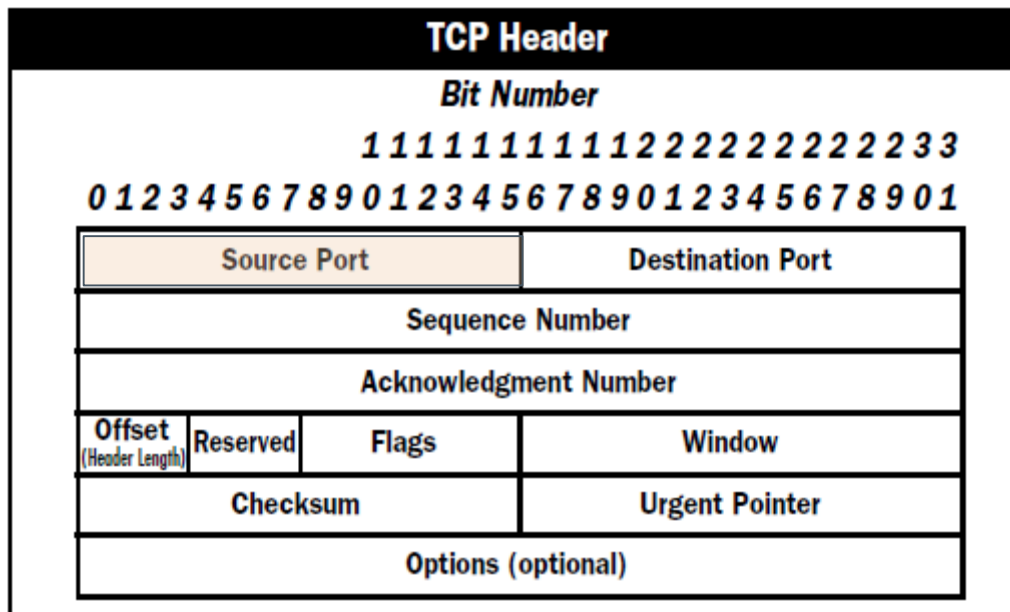
Tamaño de la cabecera sin opciones  
20 bytes



Offset o Header Length indica el número de palabras de 32 bits (4bytes) de la cabecera TCP

Valor mínimo = 5      => 5x4 = 20 bytes

Valor máximo = 15   => 15x4= 60 bytes

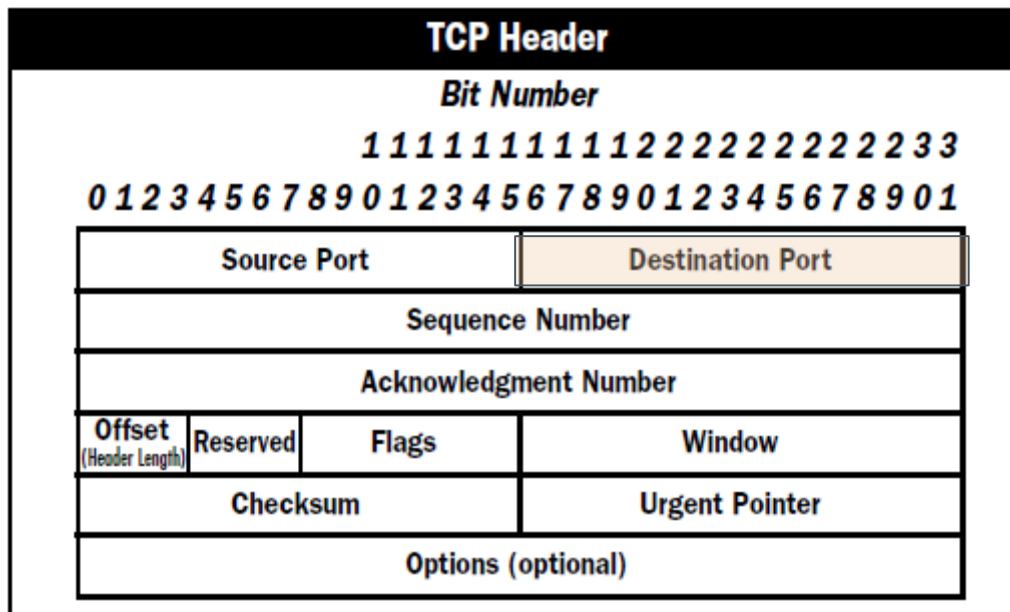


## Puerto Origen

Indica el protocolo de la capa de aplicación de origen que envía el segmento TCP.

La dirección IP de origen [IP] + el puerto origen [TCP] = Socket (conector) dirección única

Es globalmente significativa desde la que se ha enviado el segmento.



## Puerto Origen

Indica el protocolo de la capa de aplicación de origen que envía el segmento TCP.

La dirección IP de origen [IP] + el puerto origen [TCP] = Socket (conector) dirección única

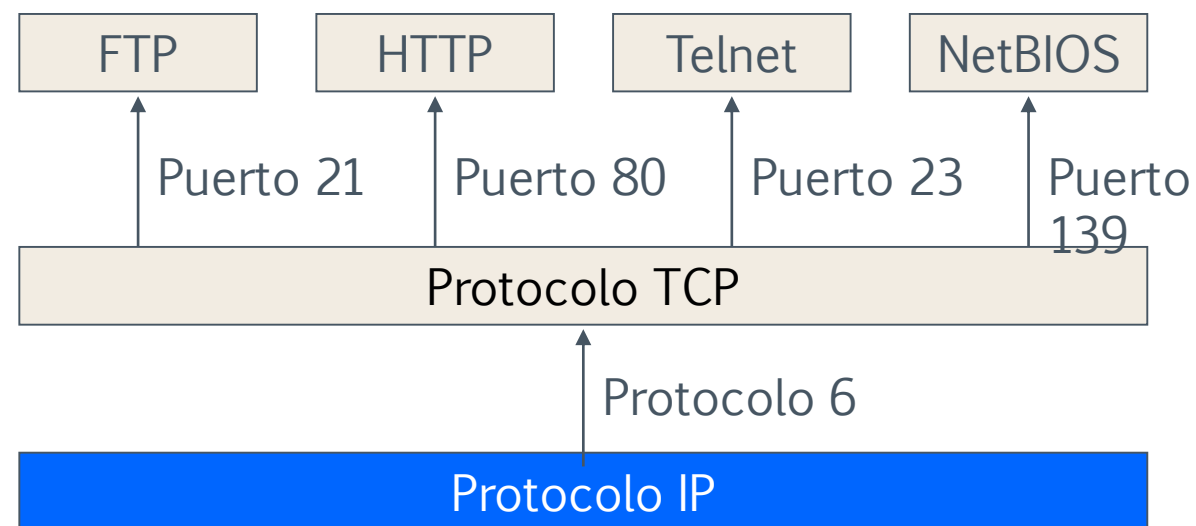
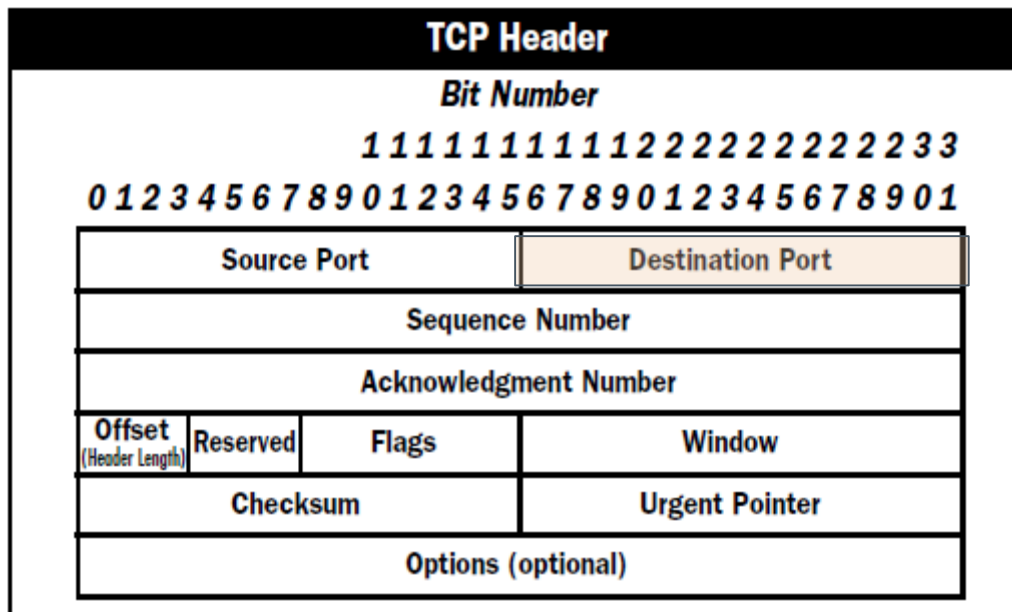
Es globalmente significativa desde la que se ha enviado el segmento.

## Puerto Destino

Indica el protocolo de la capa de aplicación de destino.

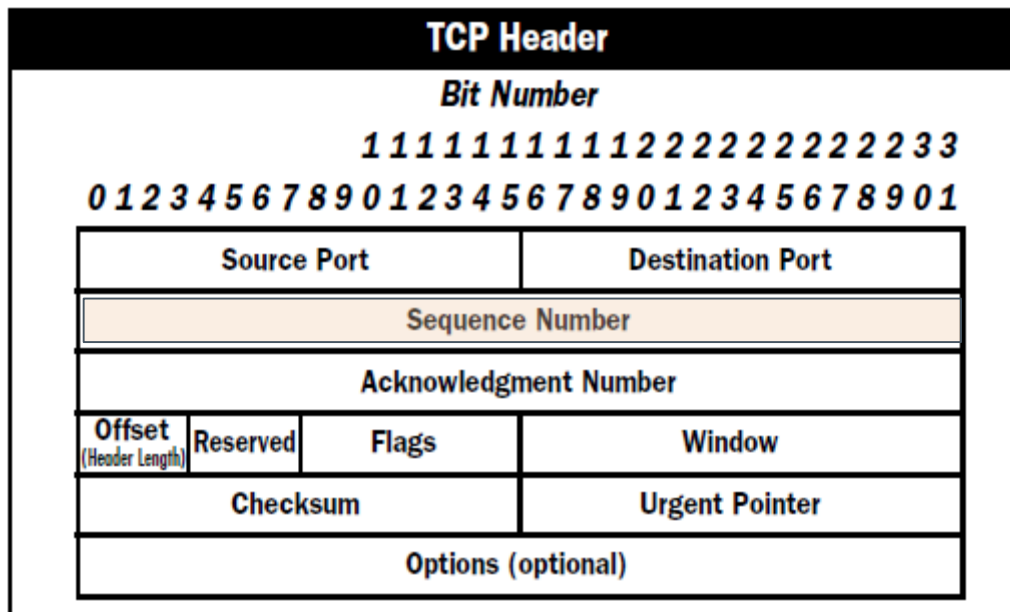
La dirección IP de destino + el puerto de destino del encabezado TCP = Conector = Dir. única

Es globalmente significativa a la que se envía el segmento.



Números de puertos asignados por la IANA (Autoridad de. Números Asignados en Internet)



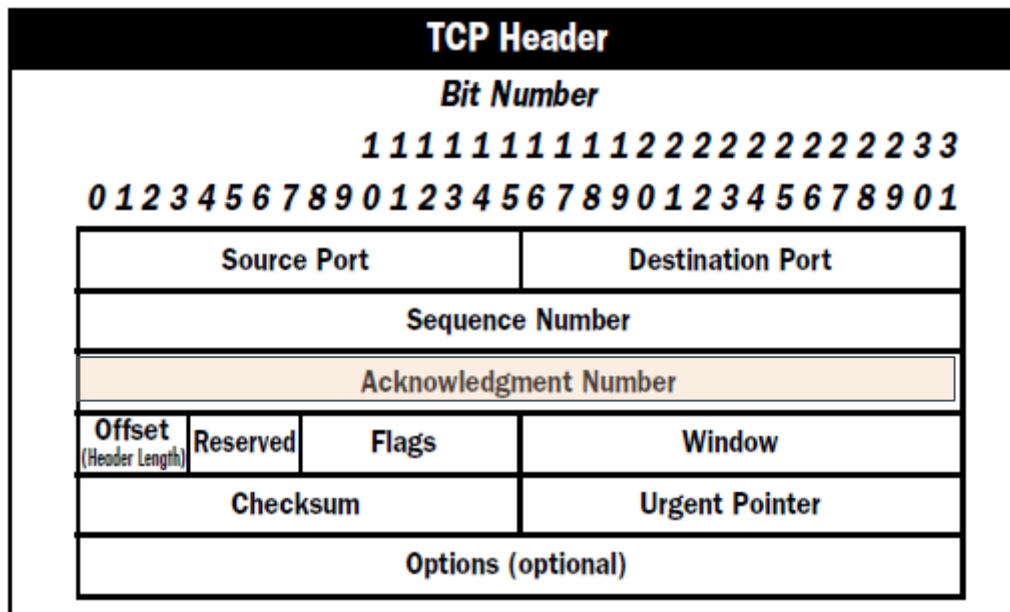


### - Número de Secuencia

Indica el número de secuencia, basado en **secuencias de bytes** de salida del **primer byte del segmento**.

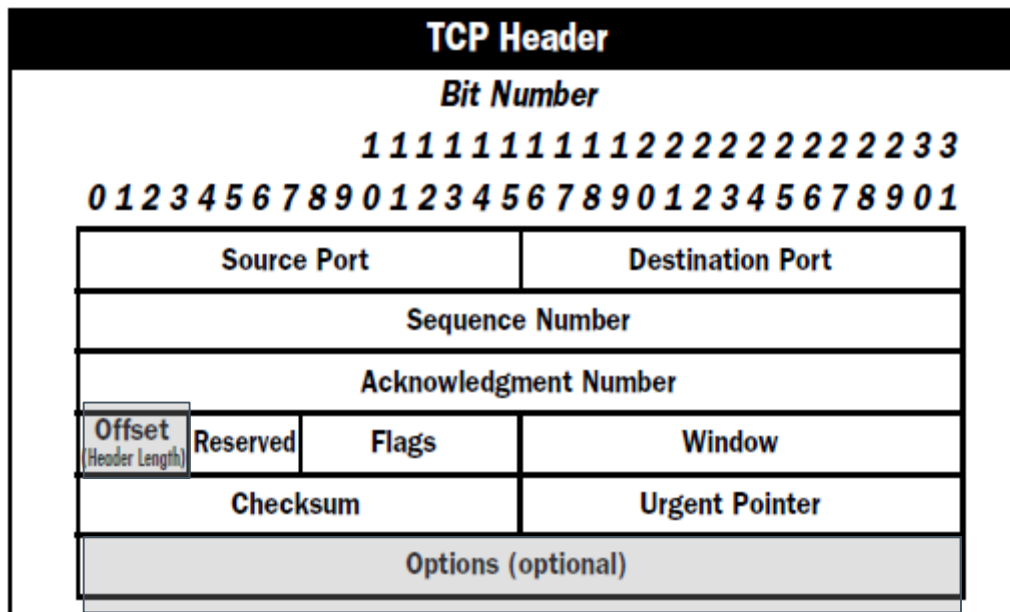
Al establecer una conexión TCP, los segmentos TCP con un valor de indicador **SYN** (Sincronización) **en 1**, define el campo Número de Secuencia con el **Numero de Secuencia Inicial** (ISN, Inicial Sequence Number).

Esto indica que **el primer octeto** de la secuencia de bytes **de salida** enviado en la conexión es **ISN+1**.



### - Número de Asentimiento o reconocimiento

Indica el número de secuencia del siguiente byte que el receptor espera recibir.  
(Similar a un ACK)



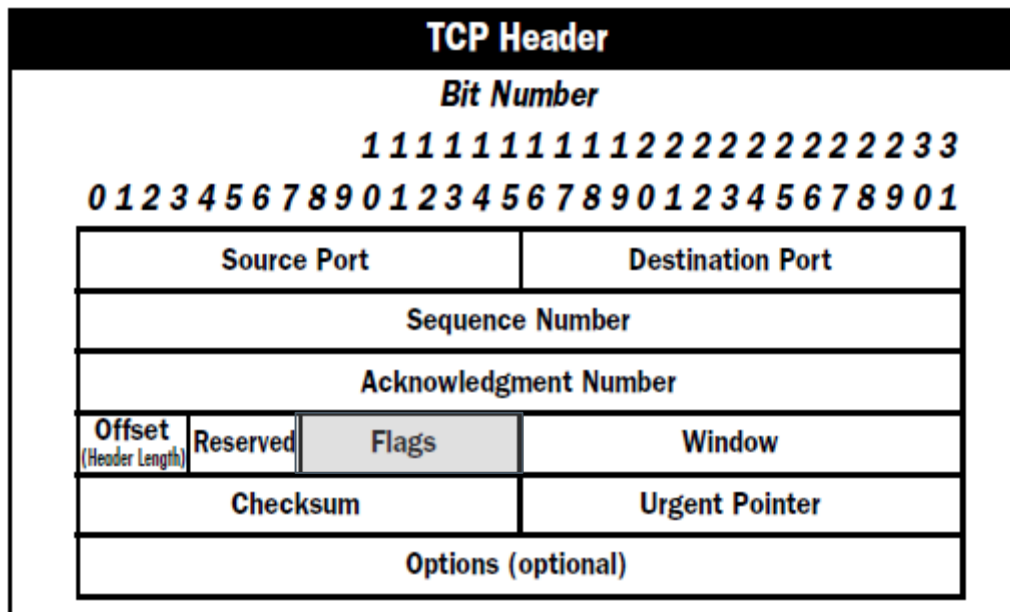
Offset o Header Length (4 bits)

indica el número de palabras de 32 bits (4bytes) de la cabecera TCP

Valor mínimo = 5    => 5x4 = 20 bytes

Valor máximo = 15   => 15x4= 60 bytes

6 BITS RESERVADOS 0's



## - Banderas

|   |   |     |     |     |     |     |     |
|---|---|-----|-----|-----|-----|-----|-----|
| 0 | 0 | URG | ACK | PSH | RST | SYN | FIN |
|---|---|-----|-----|-----|-----|-----|-----|

URG- Urgente.- Indica que el segmento TCP contiene datos urgentes.

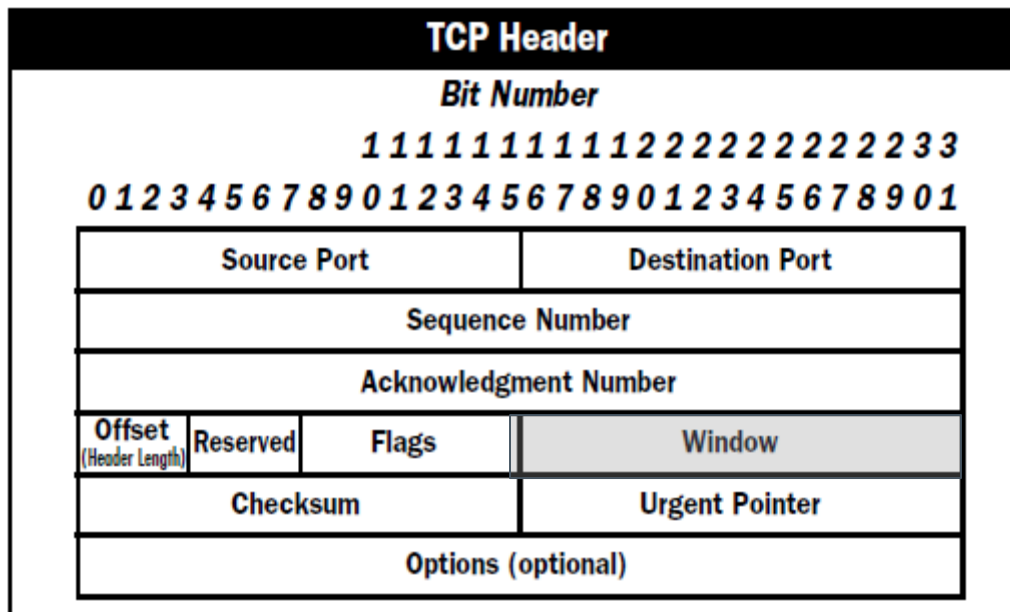
ACK.- Asentimiento Este campo contiene el valor siguiente octeto esperado en la conexión.

PSH.- Push .- Indica que el contenido del búfer de recepción de TCP se debería trasladar al protocolo de la capa de aplicación con lo que se liberará inmediatamente.

RST.- Reset Reinició de la conexión.- Indica que la conexión debe abortar.

SYN .- Sincronizar.- Indica que el segmento contiene el número de secuencia de sincronización.

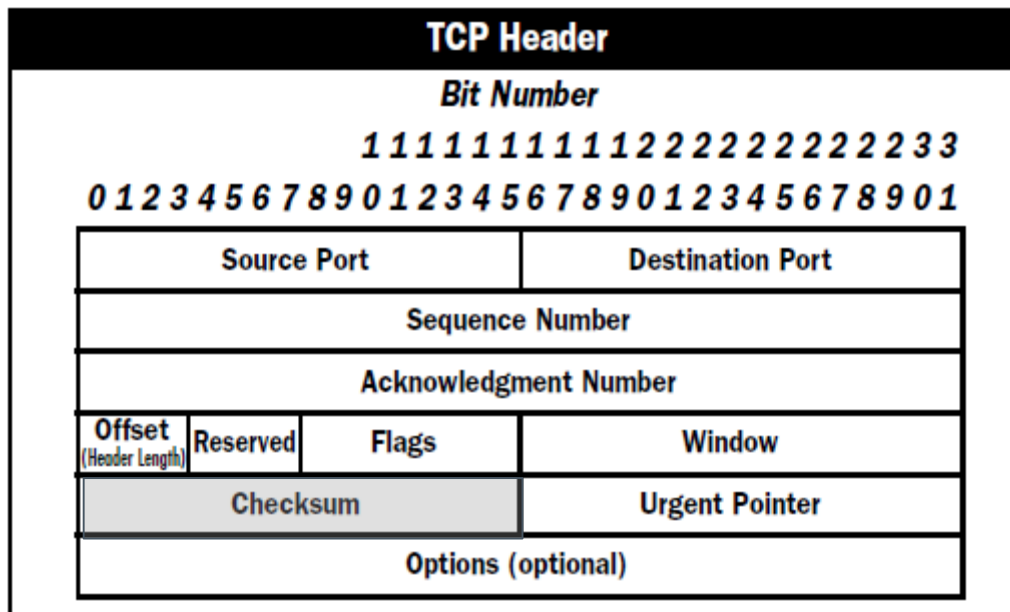
FIN.- Finalización del envío de datos.



## Ventana

Indica el número de bytes que hay disponible en el buffer receptor

El receptor TCP indica al emisor cuantos datos puede enviar aún que se puedan guardar en el buffer.



## Checksum

1.- Formar la pseudocabecera TCP

Dirección IP Origen [IP] 4 bytes

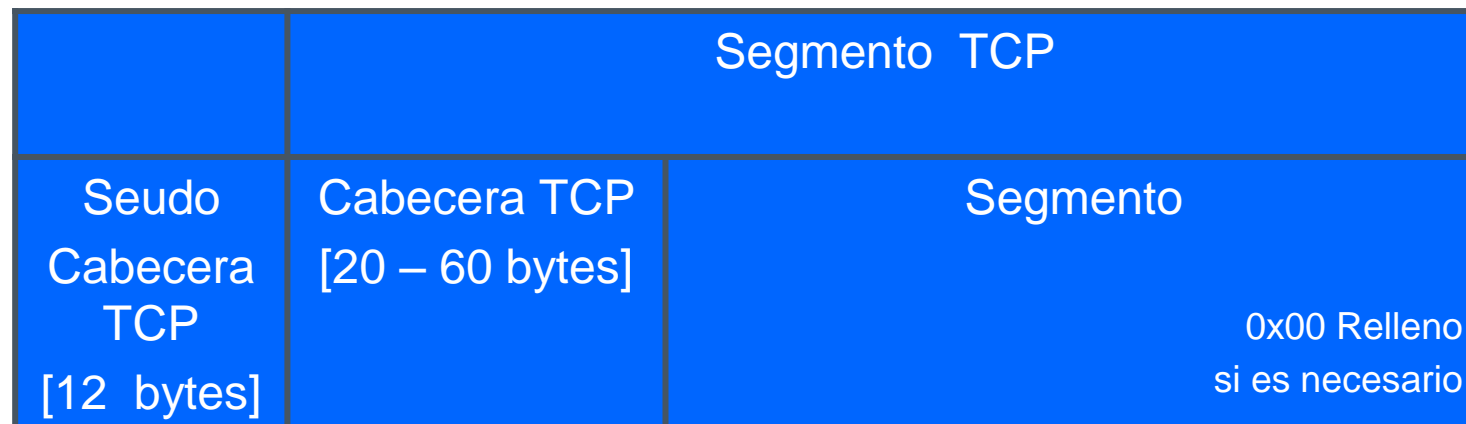
Dirección IP Destino [IP] 4 bytes

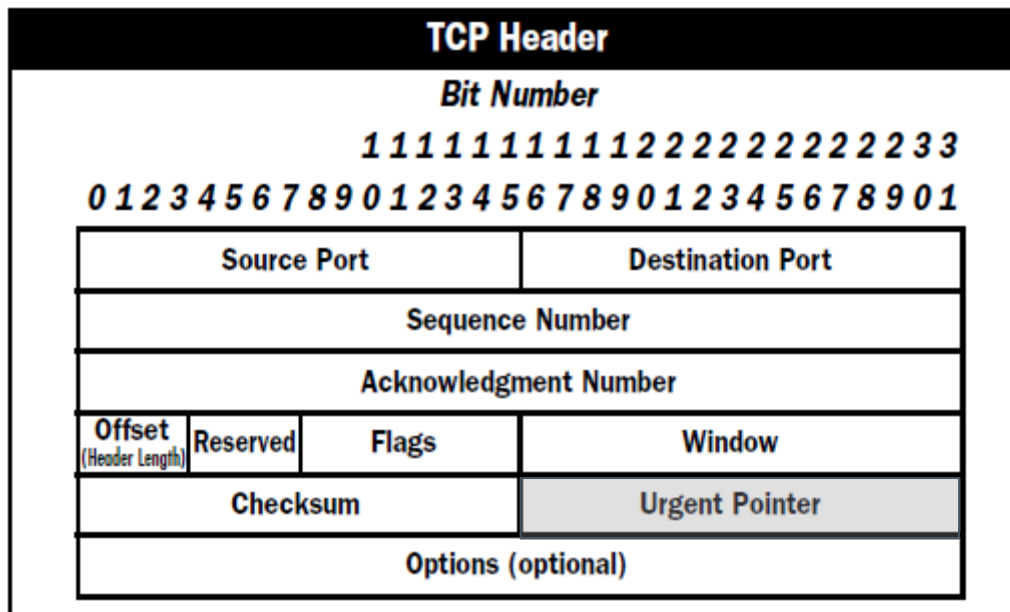
No usado = 0X00 1 byte

Protocolo= 0X06 [IP] 1 byte

Tamaño en bytes [TCP] hexa 2 bytes

Cabecera TCP sin opciones tiene tamaño=20bytes  
=0x0014

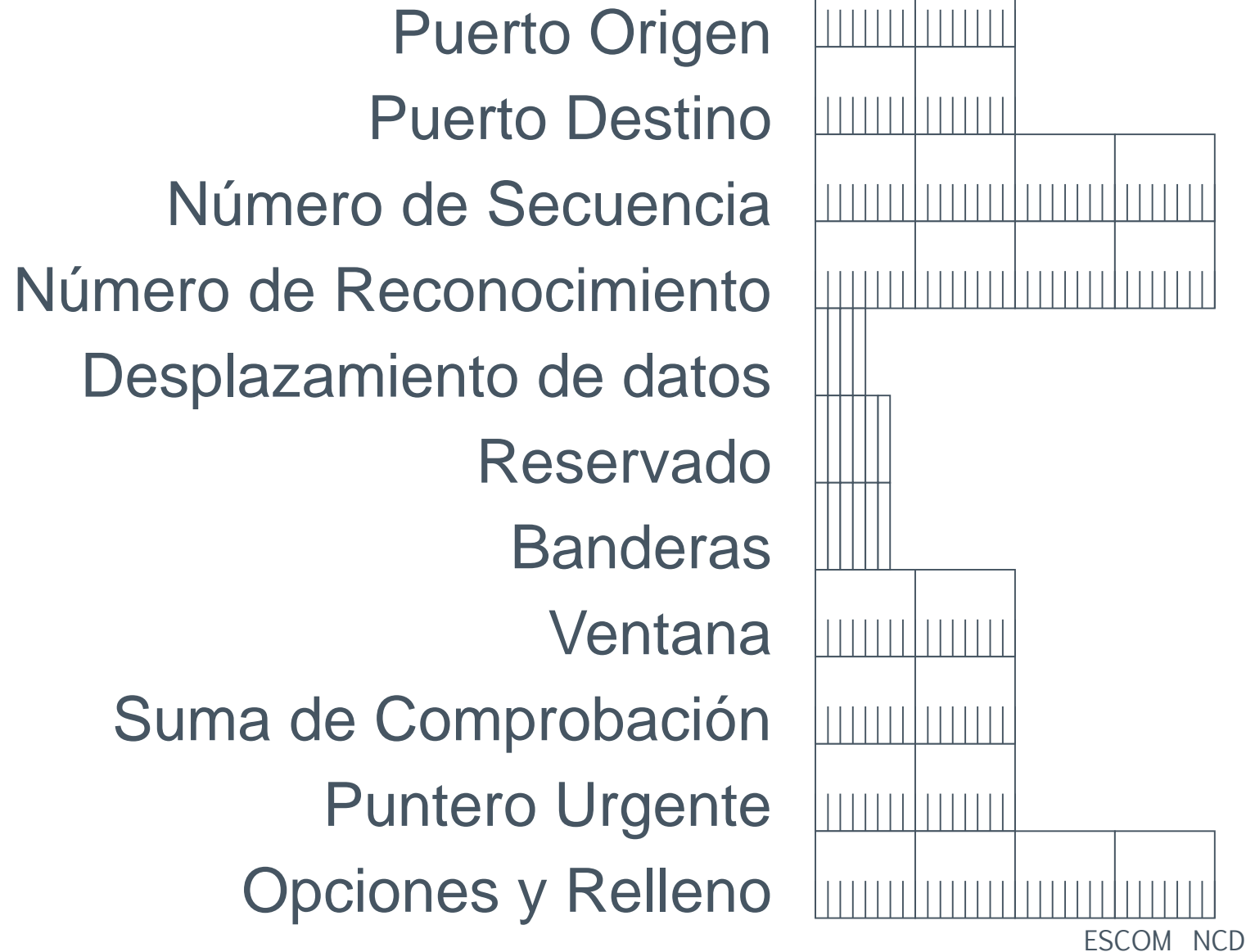




## Puntero Urgente

Indica la ubicación de los datos urgentes en el segmento

# Encabezado TCP





## ■ Trama TCP

TOT= 0X0800  
Sigue IP

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 14 | d1 | c2 | 38 | be | 00 | 18 | e7 | 33 | 3d | c3 | 08 | 00 | 45 | 00 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

00 30 94 71 40 00 80 06 f9 8c c0 a8 02 3c 4a 7d  
5f 68 10 52 00 50 03 c7 5a a1 00 00 00 00 70 02  
40 00 67 4b 00 00 02 04 05 b4 01 01 04 02

## ■ Checksum Trama TCP

IP sin opciones  
Prot= 0x06  
TCP

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 14 | d1 | c2 | 38 | be | 00 | 18 | e7 | 33 | 3d | c3 | 08 | 00 | 45 | 00 |
| 00 | 30 | 94 | 71 | 40 | 00 | 80 | 06 | f9 | 8c | c0 | a8 | 02 | 3c | 4a | 7d |
| 5f | 68 | 10 | 52 | 00 | 50 | 03 | c7 | 5a | a1 | 00 | 00 | 00 | 00 | 70 | 02 |
| 40 | 00 | 67 | 4b | 00 | 00 | 02 | 04 | 05 | b4 | 01 | 01 | 04 | 02 |    |    |

## ■ Checksum Trama TCP

Offset= 7 x 4 = 28 bytes

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 14 | d1 | c2 | 38 | be | 00 | 18 | e7 | 33 | 3d | c3 | 08 | 00 | 45 | 00 |
| 00 | 30 | 94 | 71 | 40 | 00 | 80 | 06 | f9 | 8c | c0 | a8 | 02 | 3c | 4a | 7d |
| 5f | 68 | 10 | 52 | 00 | 50 | 03 | c7 | 5a | a1 | 00 | 00 | 00 | 00 | 70 | 02 |
| 40 | 00 | 67 | 4b | 00 | 00 | 02 | 04 | 05 | b4 | 01 | 01 | 04 | 02 |    |    |

## ■ Checksum Trama TCP

Trama TCP con 8 bytes de opciones

Offset= 7 x 4 = 28 bytes

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 14 | d1 | c2 | 38 | be | 00 | 18 | e7 | 33 | 3d | c3 | 08 | 00 | 45 | 00 |
| 00 | 30 | 94 | 71 | 40 | 00 | 80 | 06 | f9 | 8c | c0 | a8 | 02 | 3c | 4a | 7d |
| 5f | 68 | 10 | 52 | 00 | 50 | 03 | c7 | 5a | a1 | 00 | 00 | 00 | 00 | 70 | 02 |
| 40 | 00 | 67 | 4b | 00 | 00 | 02 | 04 | 05 | b4 | 01 | 01 | 04 | 02 |    |    |

## ■ Checksum Trama TCP

Offset= 7 x 4 = 28 bytes

|           |           |    |    |    |    |    |    |    |    |           |    |    |    |           |           |
|-----------|-----------|----|----|----|----|----|----|----|----|-----------|----|----|----|-----------|-----------|
| 00        | 14        | d1 | c2 | 38 | be | 00 | 18 | e7 | 33 | 3d        | c3 | 08 | 00 | 45        | 00        |
| 00        | 30        | 94 | 71 | 40 | 00 | 80 | 06 | f9 | 8c | <u>c0</u> | a8 | 02 | 3c | <u>4a</u> | <u>7d</u> |
| <u>5f</u> | <u>68</u> | 10 | 52 | 00 | 50 | 03 | c7 | 5a | a1 | 00        | 00 | 00 | 00 | 70        | 02        |
| 40        | 00        | 67 | 4b | 00 | 00 | 02 | 04 | 05 | b4 | 01        | 01 | 04 | 02 |           |           |

Checksum Correcto

Classroom:

Investigar como se calcula el CHECKSUM de TCP

Nota: En la cabecera IP el checksum se calcula tomando en cuenta todos los bytes que conforman a la cabecera IP

En el protocolo TCP no es así

Se debe formar la seudocabecera + cabecera TCP

(en otras palabras, se toman ciertos valores de otra cabecera y se ponen con ciertos valores de la cabecera TCP.

Eso es lo que tendrás que investigar.

Con media cuartilla a mano en tu cuaderno es suficiente

# Ejercicio TCP

```
00 01 f4 43 c9 19 00 18 e7 33 3d c3 08 00 45 00
00 28 f6 18 40 00 80 06 6b a4 94 cc 19 f5 40 e9
a9 68 08 3a 00 50 42 fe d8 4a 6a 66 ac c8 50 10
42 0e 00 00 00 00
```

Analizar Toda la trama a mano (todos los campos)  
Cabecera Ethernet, Cabecera IP,

Para la cabecera TCP  
Puerto origen y destino

Se escribe su valor en decimal : Num. Secuencia, Num  
ACK, Ventana , Puntero Urgente  
Indicar si hay alguna bandera prendida

Verificar checksum (calcularlo de acuerdo a tu investigación  
y mostrar paso a paso)

Opciones sólo ponerlas en hexadecimal

Recuerda formar la  
seudocabecera +  
cabecera TCP