

UT1: Revision conceptos Redes

Servicios en Red - 2º Curso CFGM SMR

Repaso de conceptos previos.

- ▶ Introducción.
 - ▶ Modelo OSI.
 - ▶ Protocolos.
 - ▶ IP.
 - ▶ Máscaras.
 - ▶ Formato CIDR.
 - ▶ Dirección de red.
 - ▶ Dirección de broadcast.
 - ▶ Subredes.
 - ▶ Arquitectura cliente-servidor.
-



Introducción.

- ▶ La finalidad de una red de ordenadores es que los usuarios puedan hacer un mejor uso de la misma mejorando de este modo el rendimiento global de la organización.
- ▶ Ejemplos de algunos servicios son:
 - ▶ Transferencia de archivos.
 - ▶ Correo y mensajería instantánea.
 - ▶ Conexión remota a equipos.
 - ▶ Acceso a información en servidores web.
 - ▶ Transferencia de audio y video.



Introducción.

- ▶ Los servicios de red se instalan y funcionan sobre una red, así que es necesario tener un conocimiento profundo de la misma para poder trabajar con ellos.
- ▶ Los conceptos básicos de redes se obtuvieron en el módulo de Redes Locales, en I°.
- ▶ En esta unidad vamos a repasar los conceptos más importantes y que usaremos durante este módulo.



Introducción.

- ▶ Para que dos ordenadores puedan intercambiar información, ¿qué tareas se deben realizar?
 - Formatos de los datos.
 - Comprimir los datos.
 - Conocer el destino.
 - Enviar varios datos a la vez.
 - ¿Y si hay congestión?
 - Conocer el origen.
 - Asegurarse de que ha llegado.
 - Sincronización.
 - Controlar que no hay errores.
 - Proporcionar

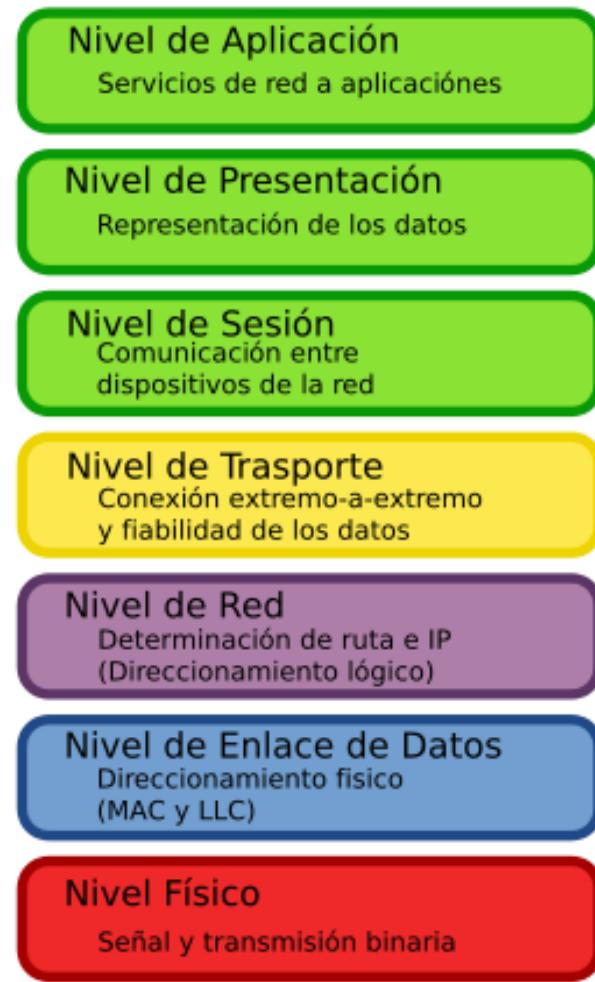


Introducción.

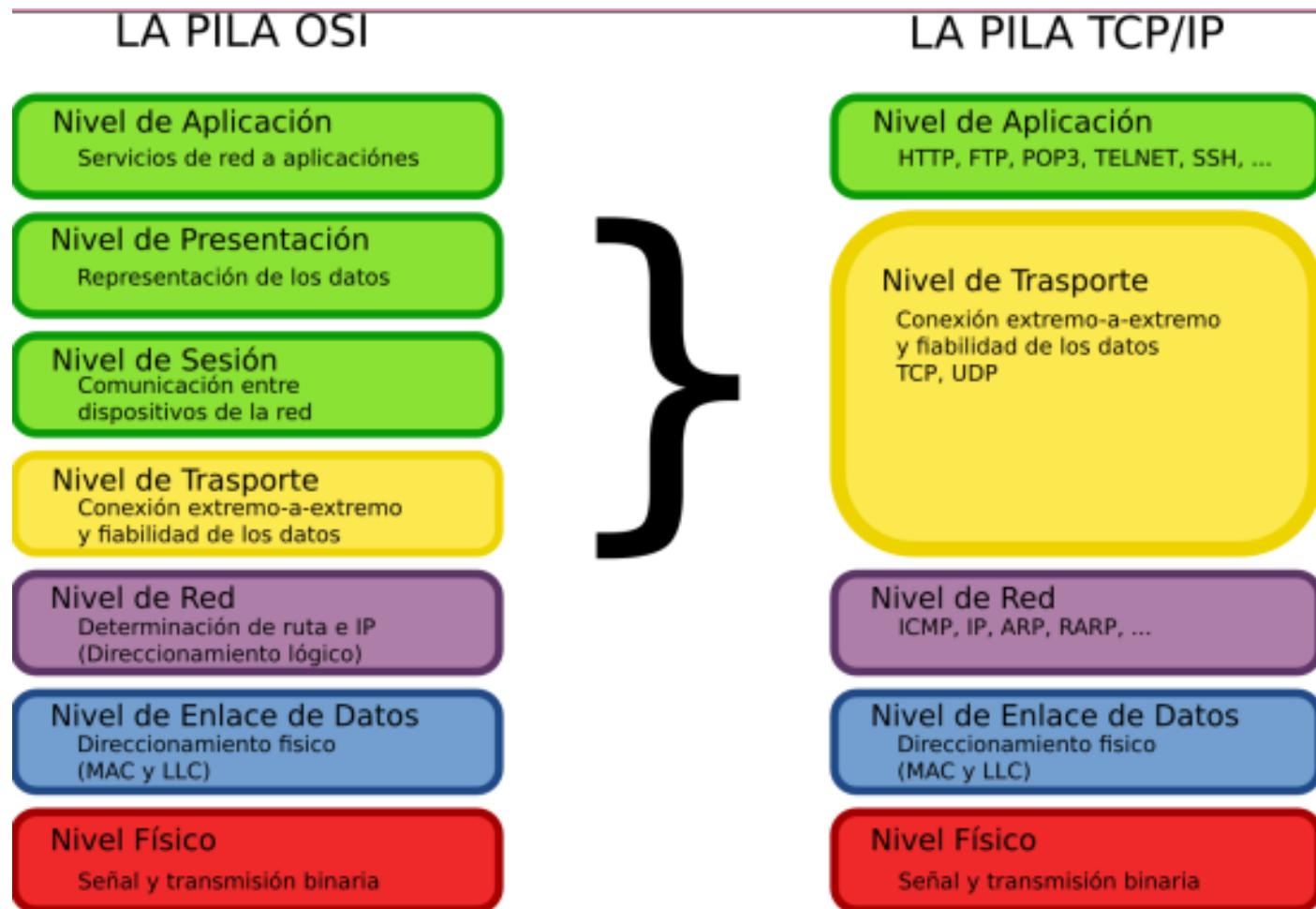
- ▶ La solución al problema de conectar dos ordenadores es partir el problema en partes.
- ▶ Se organizarán las tareas en niveles y cada nivel realizará alguna de las tareas indicadas.
- ▶ ¿Con qué criterio se establecen esos niveles?
- ▶ El modelo OSI es una posibilidad.
- ▶ Este modelo establece 7 niveles, y en cada uno existe una entidad responsable de realizar una o varias tareas para llevar a cabo la comunicación



El modelo OSI.



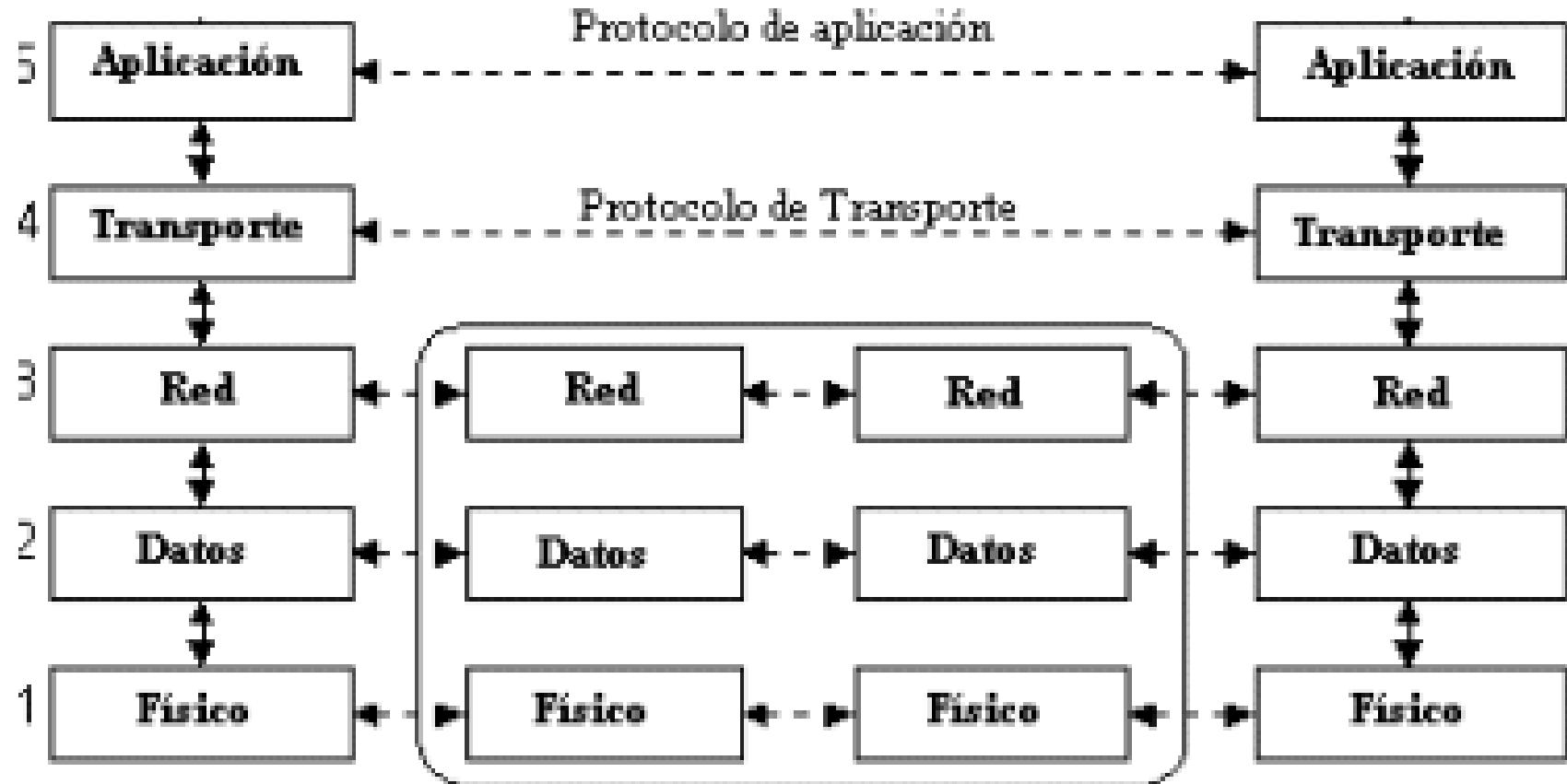
El modelo OSI vs TCP/IP.



El modelo OSI.



Redes TCP/IP.



Protocolos.

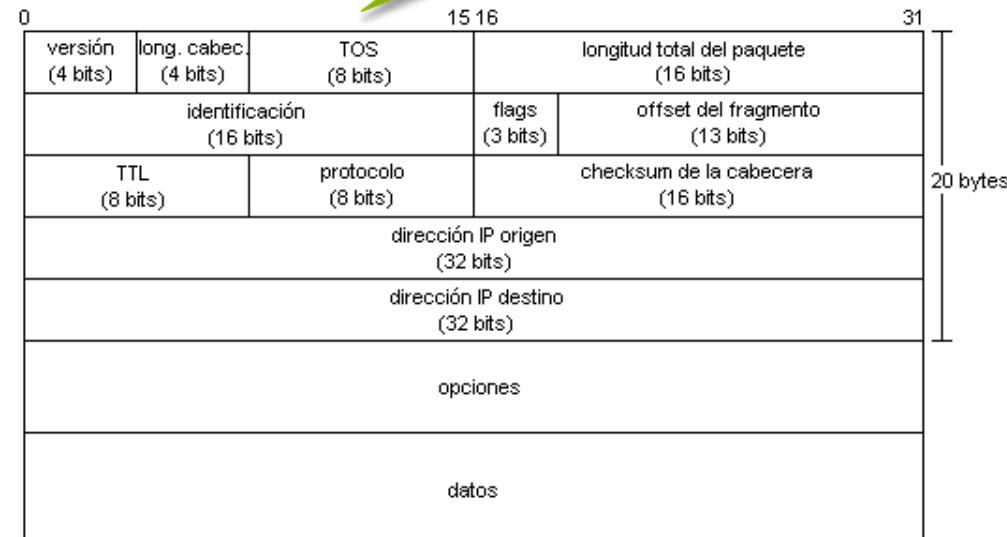
- ▶ Conjunto de reglas que utilizan dos computadores para intercambiar información.
- ▶ Existen diferentes protocolos según el nivel en el que nos encontremos dentro de una determinada arquitectura de red.
- ▶ Ejemplo:
 - ▶ Protocolos de nivel físico: USB, 802.11x, RS-232, 100BaseT.
 - ▶ Protocolo de nivel de enlace: Ethernet
 - ▶ Protocolos de nivel de red: IP, ICMP, RIP, OSPF.
 - ▶ Protocolos de nivel de transporte: TCP, UDP.



IP

- ▶ IP Internet Protocol o Protocolo de Internet.
- ▶ Protocolo de la arquitectura TCP/IP correspondiente al nivel de red o internet.
- ▶ Hay dos versiones: IPv4 e IPv6.
- ▶ Funciones:
 - Direccionamiento.
 - Formato de los paquetes.
 - Encaminamiento.

Formato paquete IPv4



Dirección IPv4

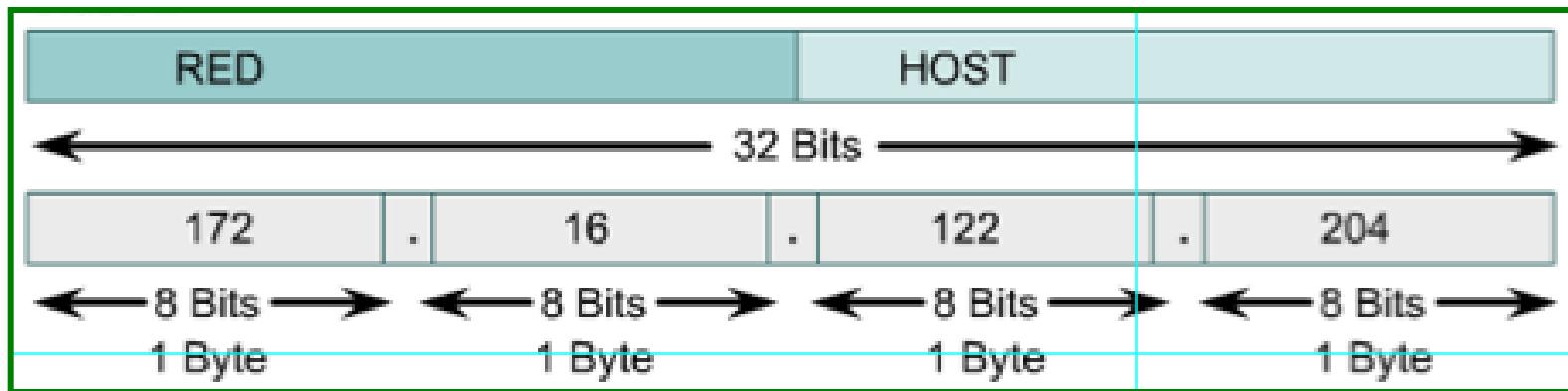
- ▶ Identificador de un equipo dentro de la red.
- ▶ Recordar que hay una IP por cada tarjeta de red, no por cada equipo.
- ▶ Formato:
 - ▶ conjunto de 32 bits
 - ▶ agrupados de 8 en 8 y separados por puntos
 - ▶ Escritos en decimal.
 - ▶ Ejemplo: 192.168.1.1
 - ▶ Cada decimal vale como máximo 255.



Direcciones IPv4.

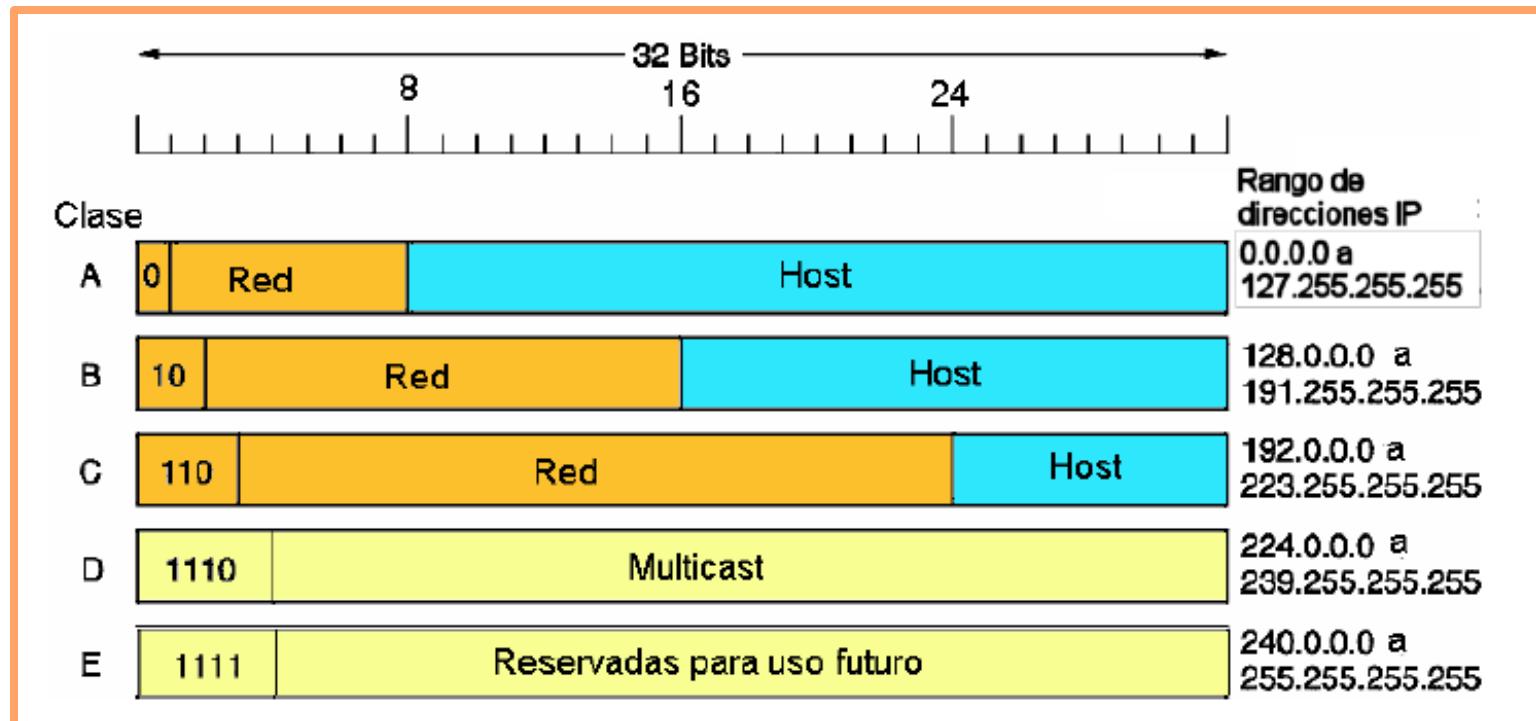
► Una IP tiene dos campos:

- ▶ Identificador de red o **netid**:
 - Identifica la red en la que se encuentra nuestro host.
 - Todas las máquinas de nuestra red tienen el mismo netid.
- ▶ Identificador del host o **hostid**.
 - Identifica a la máquina dentro de la red.
 - Tiene un valor distinto para cada máquina de la red.



Clases de redes IPv4.

- ▶ Existen muchas direcciones IPv4 y se organizan en clases.
- ▶ Hay 5 clases de direcciones: A, B, C, D y E.
- ▶ La clase a la que pertenece una dirección la da el **netid**.



Máscara de red.

- ▶ Secuencia de 32 bits formada por una serie de 1s seguidos de 0s : 255.255.0.0.
- ▶ La usan los routers para calcular la dirección de red o subred a la que pertenece una IP.
- ▶ Se construye poniendo un 1 por cada bit del **netid** y un 0 por cada bit del **hostid**.

CLASE	MÁSCARA
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0



Dirección de red.

- ▶ Se utiliza para identificar a toda la red (nunca es destino).
- ▶ La dirección de red se obtiene de realizar un AND lógico entre la dirección IP del host y la máscara.
- ▶ O también poniendo a 0 los bits del host.
- ▶ El AND lógico se calcula como sigue:

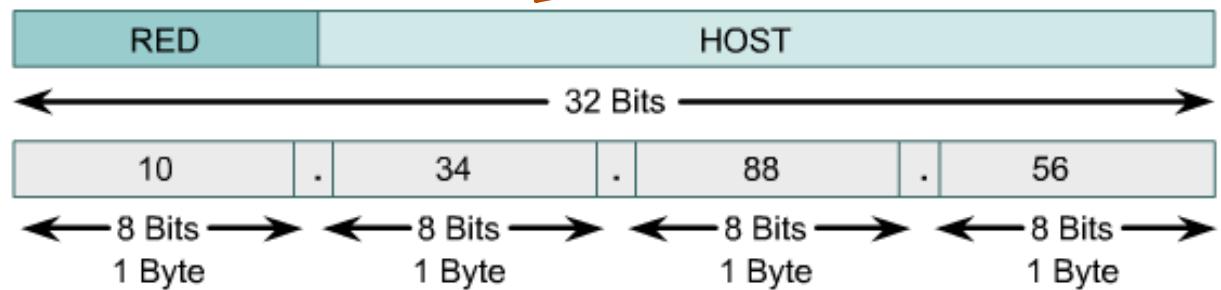
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



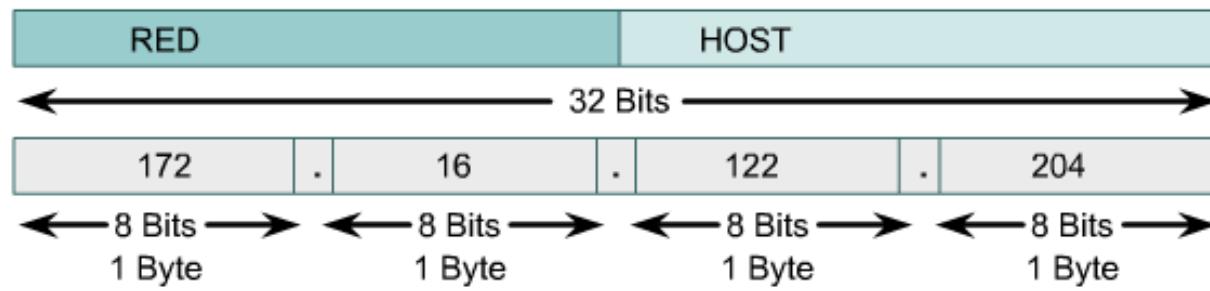
Dirección de red.

La dirección de red sería:
10.0.0.0

Clase A



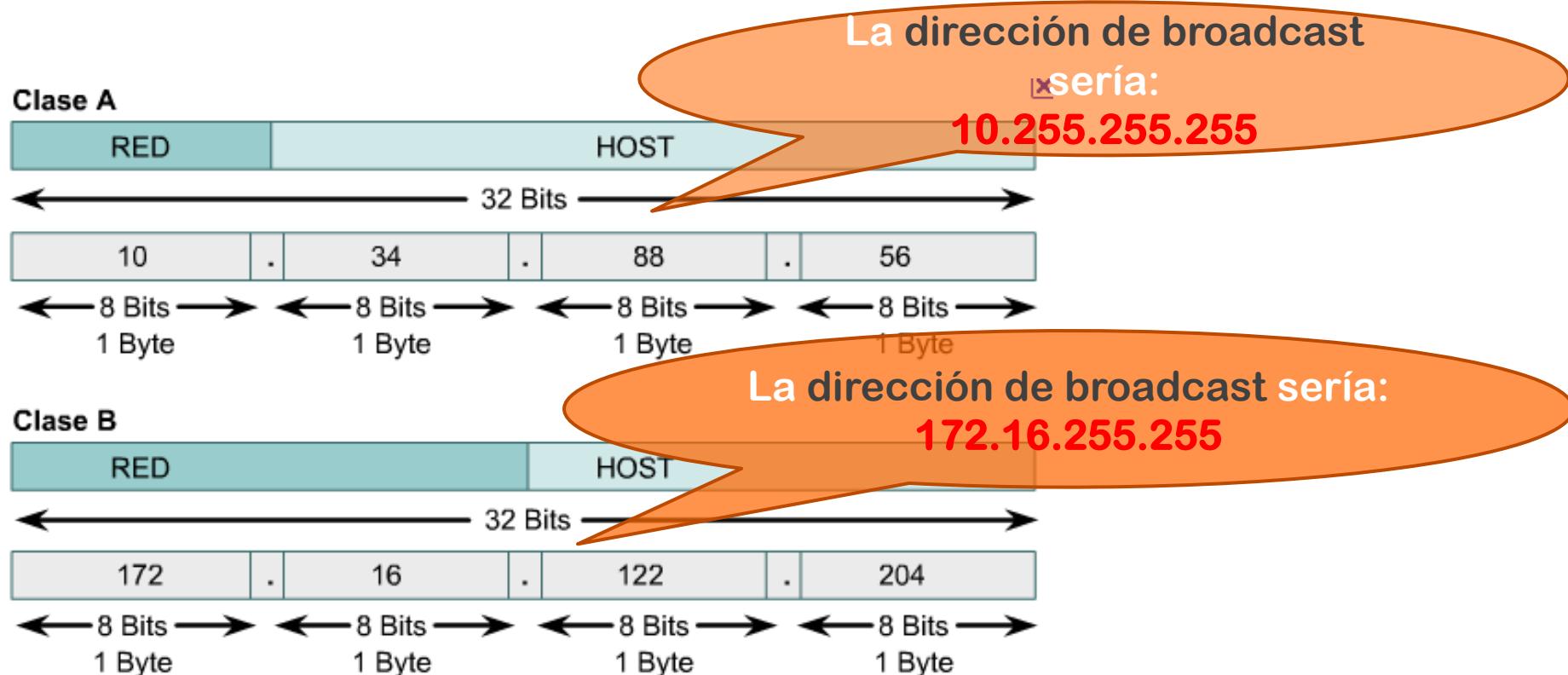
Clase B



La dirección de red sería:
172.16.0.0

Dirección de broadcast.

- ▶ Se utiliza para enviar datos a toda la red (siempre es destino).
- ▶ Se obtiene poniendo a 1 todos los bits del host.



Ejemplo:

- ▶ Dada la dirección 192.168.1.1
 - ▶ ¿Cuál es la dirección de red?
 - ▶ ¿Cuál es la dirección de broadcast?
 - ▶ ¿Cuántas máquinas pueden conectarse en la red con dirección 192.168.1.0?
 - ▶ ¿Cuál es la dirección de la primera máquina?
 - ▶ ¿Cuál es la dirección de la última máquina?



Número de IPs en una red.

- ▶ ¿Cuántos ordenadores se podrían conectar en la red 192.168.1.1?

La parte de host sería el último byte así que los host serían:

192.168.1.00000000 192.168.1.0	192.168.1.00000101 192.168.1.5	192.168.1.00000100 192.168.1.10
192.168.1.00000001 192.168.1.1	192.168.1.00000110 192.168.1.6
192.168.1.00000010 192.168.1.2	192.168.1.00000111 192.168.1.7	192.168.1.11111101 192.168.1.253
192.168.1.00000011 192.168.1.3	192.168.1.00001001 192.168.1.8	192.168.1.11111110 192.168.1.254

Salvo la dirección de red y la de broadcast, el resto de direcciones pueden usarse como direcciones de máquinas en la red 192.168.1.0.

La primera dirección: 192.168.1.1

La última dirección: 192.168.1.254

192.168.1.11111111 192.168.1.255

Esta es la dirección de broadcast.

Cómo calcular el número de máquinas.

$2^{n \circ}$ de hosts - 2

192.168.1.xxxxxxxx

El número total de máquinas es $2^8 = 256$ porque hay 8 bits de host. Hay que restar 2: la dirección de red y la de broadcast.



Direcciones IP especiales (no asignables).

Dirección de red	Todos los bits del identificador de host tienen el valor 0
Máscara de subred	Todos los bits del identificador de red tienen el valor 1 y todos los bits del identificador de host tienen el valor 0.
Broadcast de una red distante (indirecto)	Todos los bits del identificador de host tienen el valor 1
Broadcast local o directo	Todos los bits valen 1 (255.255.255.255)
Host de esta red	Todos los bits de la parte de red valen 0.
Este host	0.0.0.0
loopback	Red 127.0.0.0. Dirección de red ficticia local a cada host que se utiliza para pruebas de los protocolos TCP/IP
localhost	127.0.0.1. Se refiere al propio host.



Tipos de IP.

- ▶ **PÚBLICAS:** presentes en Internet.
- ▶ **PRIVADAS:** no están presentes en Internet.
- ▶ **ESTÁTICAS:** no cambian.
- ▶ **DINÁMICAS:** cambian en cada conexión.
Asignadas por un servidor DHCP.



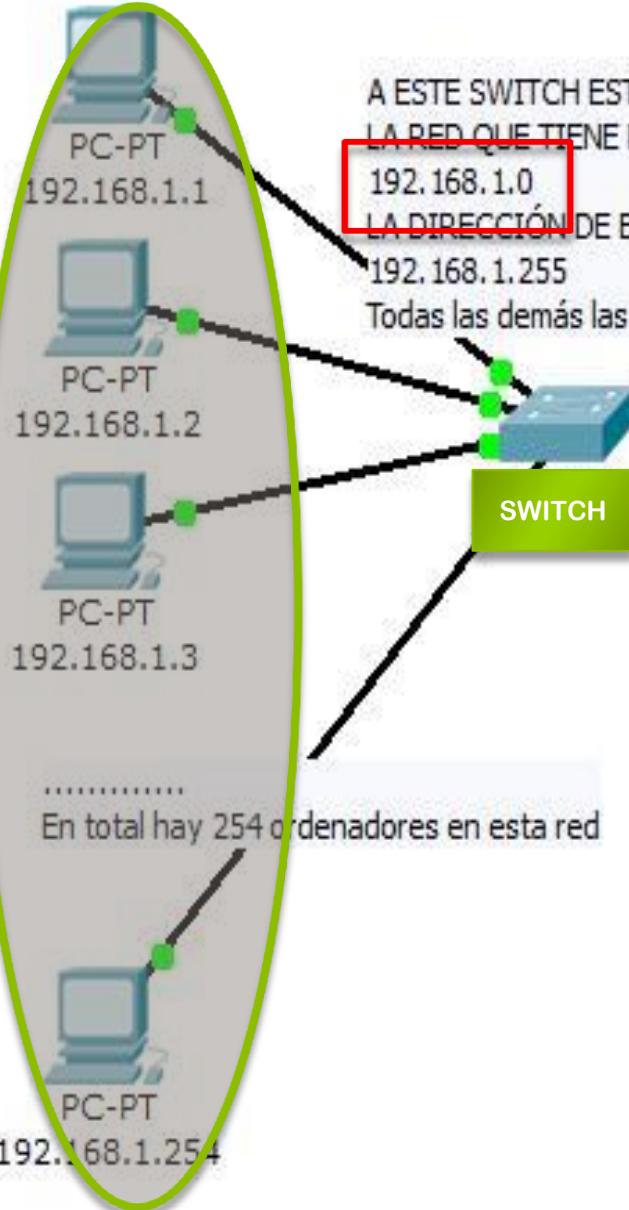
Direcciones IP privadas.

- ▶ Las direcciones privadas no se pueden utilizar para conectarse a Internet.
- ▶ Dentro de cada clase, hay rangos de direcciones que no son asignadas en Internet.
- ▶ Permite conectar a Internet muchos hosts usando pocas direcciones públicas. (NAT)

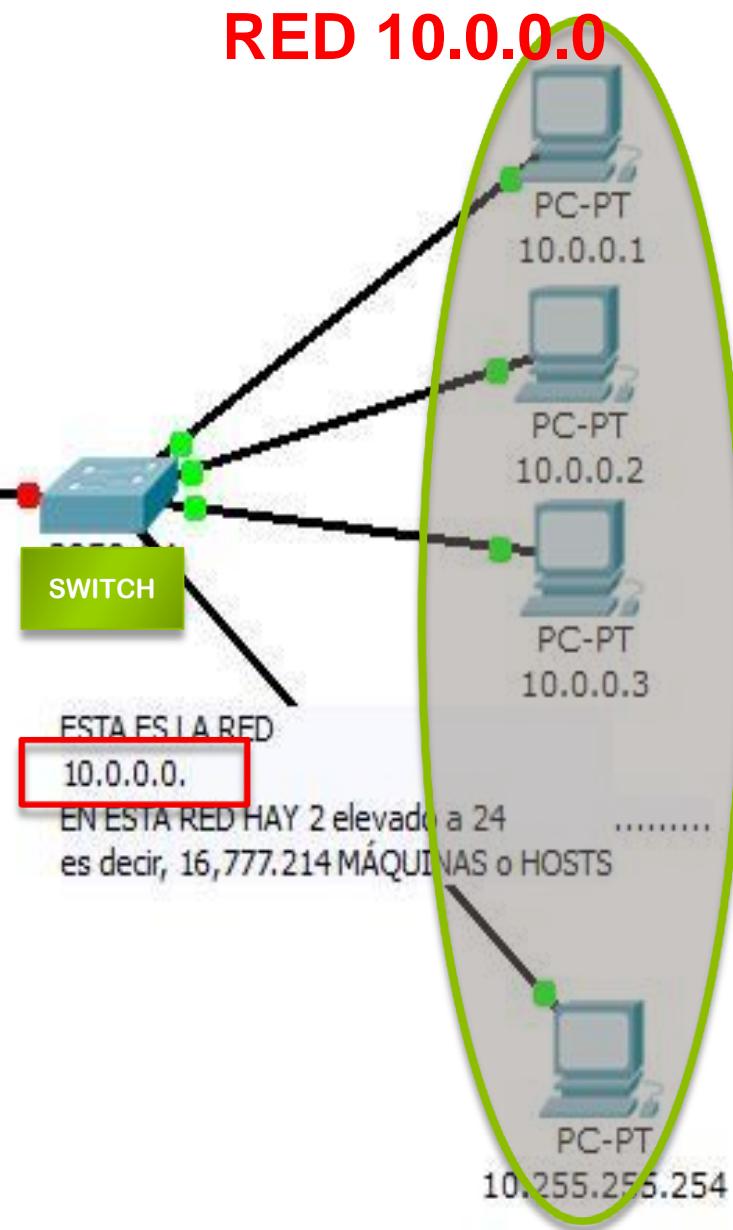
- Clase A: 10.0.0.0 a 10.255.255.255
 - Clase B: 172.16.0.0 a 172.31.255.255
 - Clase C: 192.168.0.0 a 192.168.255.255



RED 192.168.1.0

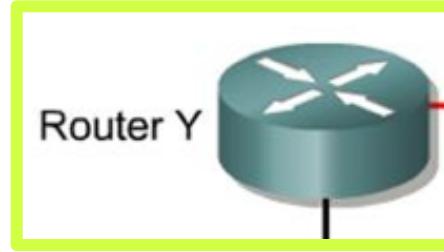
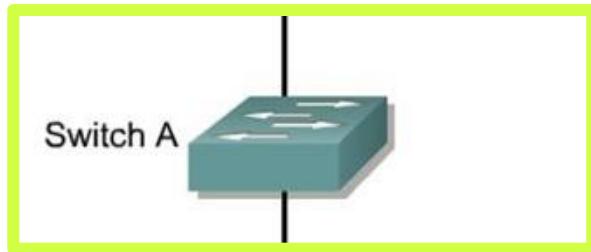


RED 10.0.0.0



Antes de seguir ...

- ▶ **Routers** o encaminadores son dispositivos de nivel 3 que encaminan paquetes IP entre redes (interconectan redes).
- ▶ **Switches** o conmutadores son dispositivos de nivel 2 que envían tramas entre máquinas de la misma red (construyen físicamente la red).



Inconvenientes del direccionamiento basado en clases.

► Uso ineficiente del espacio de direcciones.

Imaginemos que una empresa necesita asignar 5000 direcciones IP. Su ISP, le debería asignar una IP de una red de clase B.

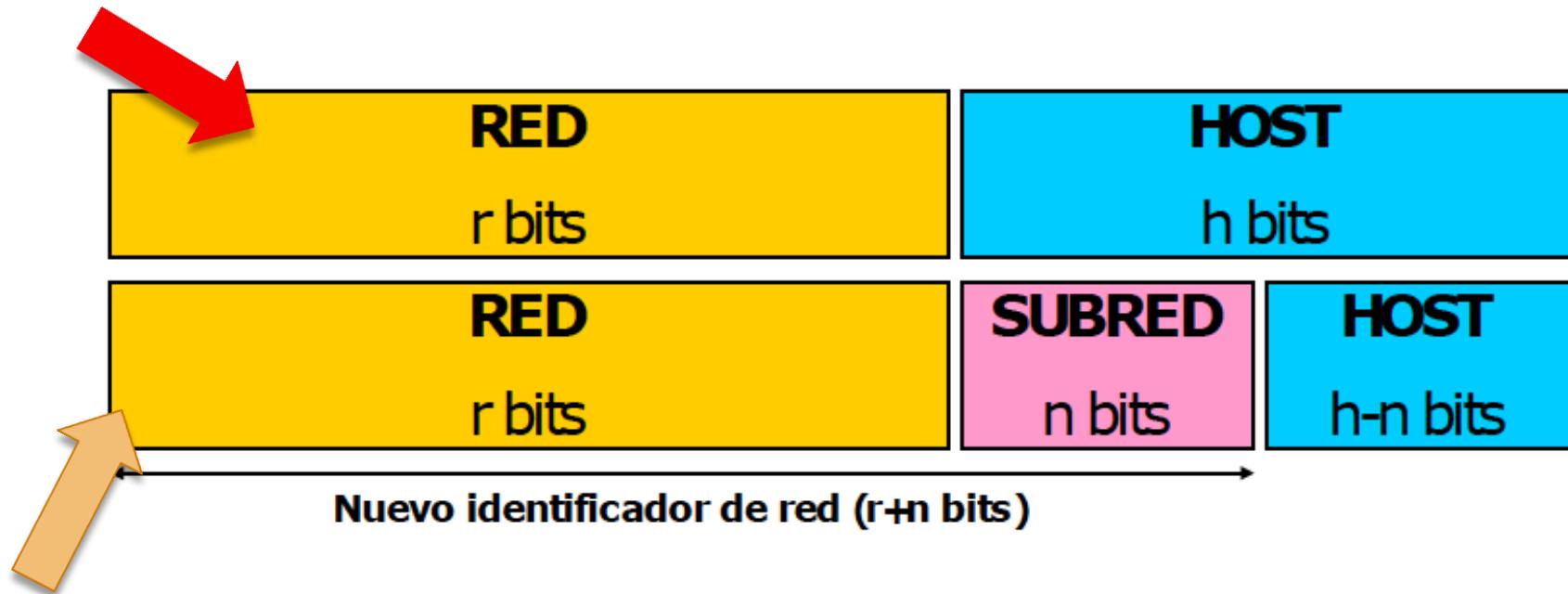
150.16.xxxxxxx.xxxxxxx

$2^{16}-2=65534$ direcciones IP le dan, así que se desperdician
 $65534-5000=60535$ DESPERDICIADAS!!!!



Subnetting.

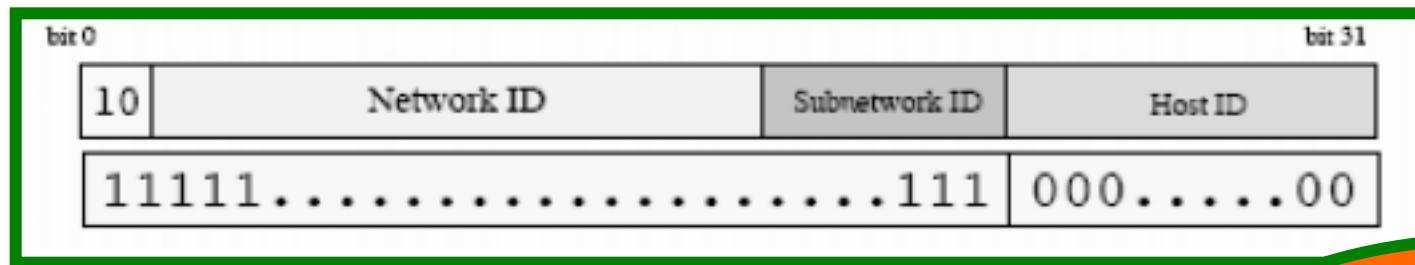
Sin subnetting una IP tiene **dos** partes.



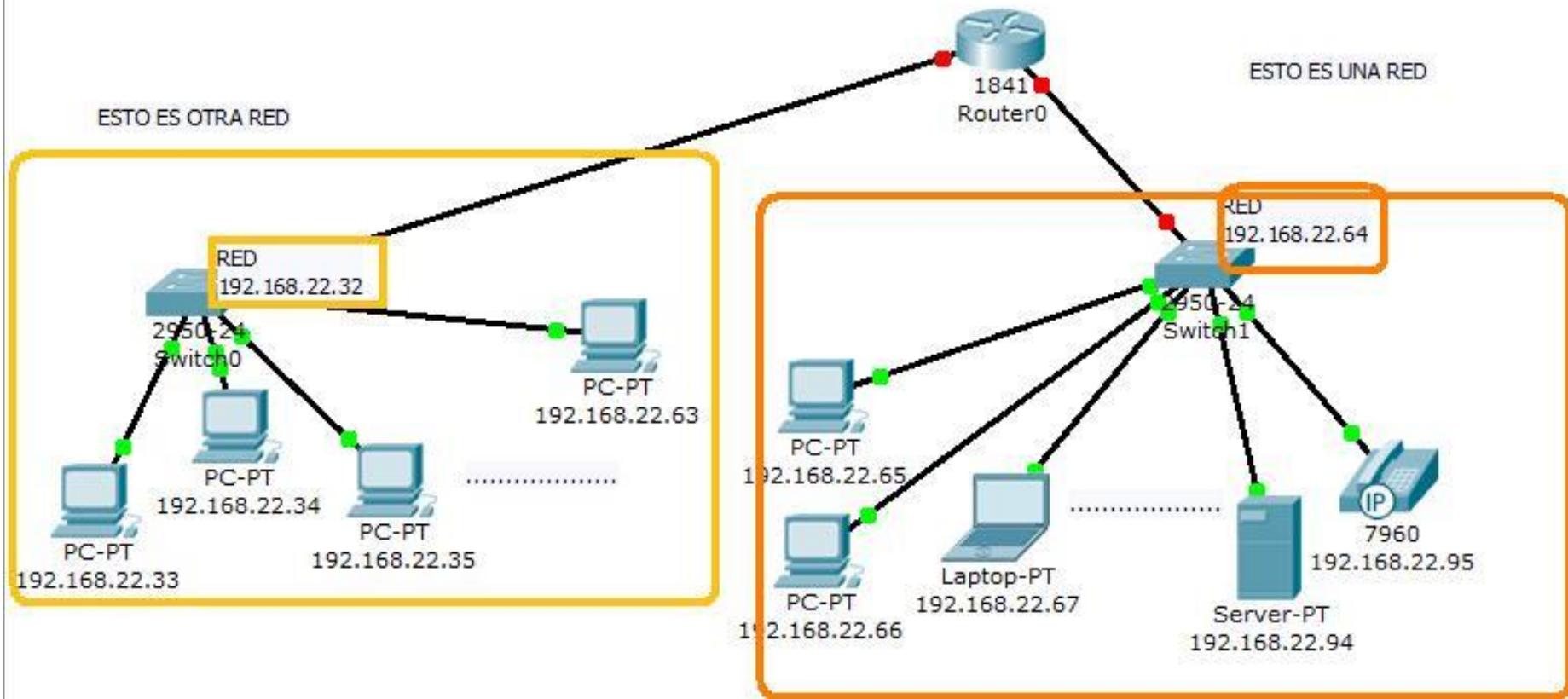
Con subnetting una IP tiene **tres** partes.

Cómo identificar la red.

- ▶ Para saber como se divide la parte de host en subred y host se utilizan las máscaras.
- ▶ Recordad, que la máscara pone a 1 todos los bits de red y *además ahora pondrá a 1 los bits de subred*.



Ahora una
máscara puede
ser: 255.255.128.0



Problema 1

Nº de subredes útiles necesarias **14**

Nº de hosts útiles necesarios **14**

Dirección de Red **192.10.10.0**

Esto es lo
que hay
que
calcular

Clase	<u>C</u>
Máscara de Subred (por defecto)	<u>255 . 255 . 255 . 0</u>
Máscara de Subred (adaptada)	<u>255 . 255 . 255 . 240</u>
Nº total de subredes	<u>16</u>
Nº de subredes útiles	<u>14</u>
Nº total de direcciones de host	<u>16</u>
Nº de direcciones útiles	<u>14</u>
Nº de bits cogidos	<u>4</u>

Muestre aquí su forma de proceder para el Problema 1:

Número de subredes	256	128	64	32	16	8	4	2	-	Número de hosts
	-	2	4	8	16	32	64	128	256	
	128	64	32	16	8	4	2	1	-	Valores binarios
192 . 10 . 10 . 0	0	0	0	0	0	0	0	0	0	

$$\begin{array}{r} 128 \\ \text{Sumar los valores binarios} \\ \text{de los nPs a la izquierda de la línea} \\ \text{para crear la máscara de subred.} \\ 64 \\ 32 \\ +16 \\ \hline 240 \end{array}$$

$$\begin{array}{r} 16 \\ -2 \\ \hline 14 \end{array}$$

Observar el número total de hosts.
Restar 2 para obtener el nº de hosts direccionables.

$$\begin{array}{r} 16 \\ -2 \\ \hline 14 \end{array}$$

Restar 2 al nº total de subredes
para obtener el nº de subredes
válidas.

Problema 2

Nº de subredes útiles necesarias 1000

Nº de hosts útiles necesarios 60

Dirección de Red 165.100.0.0

Esto es lo
que hay
que
calcular

Clase	B
Máscara de Subred (por defecto)	255 . 255 . 0 . 0
Máscara de Subred (adaptada)	255 . 255 . 255 . 192
Nº total de subredes	1.024
Nº de subredes útiles	1.022
Nº total de direcciones de host	64
Nº de direcciones útiles	62
Nº de bits cogidos	10

Muestre aqui su forma de proceder para el Problema 2:

Problema 3

Dirección de Red **148.75.0.0 /26**

1/28 Indica el número total de bits usados para la parte de red y subred de la dirección. El resto de bits son de la parte de host de la dirección.

Esto es lo
que hay
que
calcular

Clase	B
Máscara de Subred (por defecto)	255 . 255 . 0 . 0
Máscara de Subred (adaptada)	255 . 255 . 255 . 192
Nº total de subredes	1,024
Nº de subredes útiles	1,022
Nº total de direcciones de host	64
Nº de direcciones útiles	62
Nº de bits cogidos	10

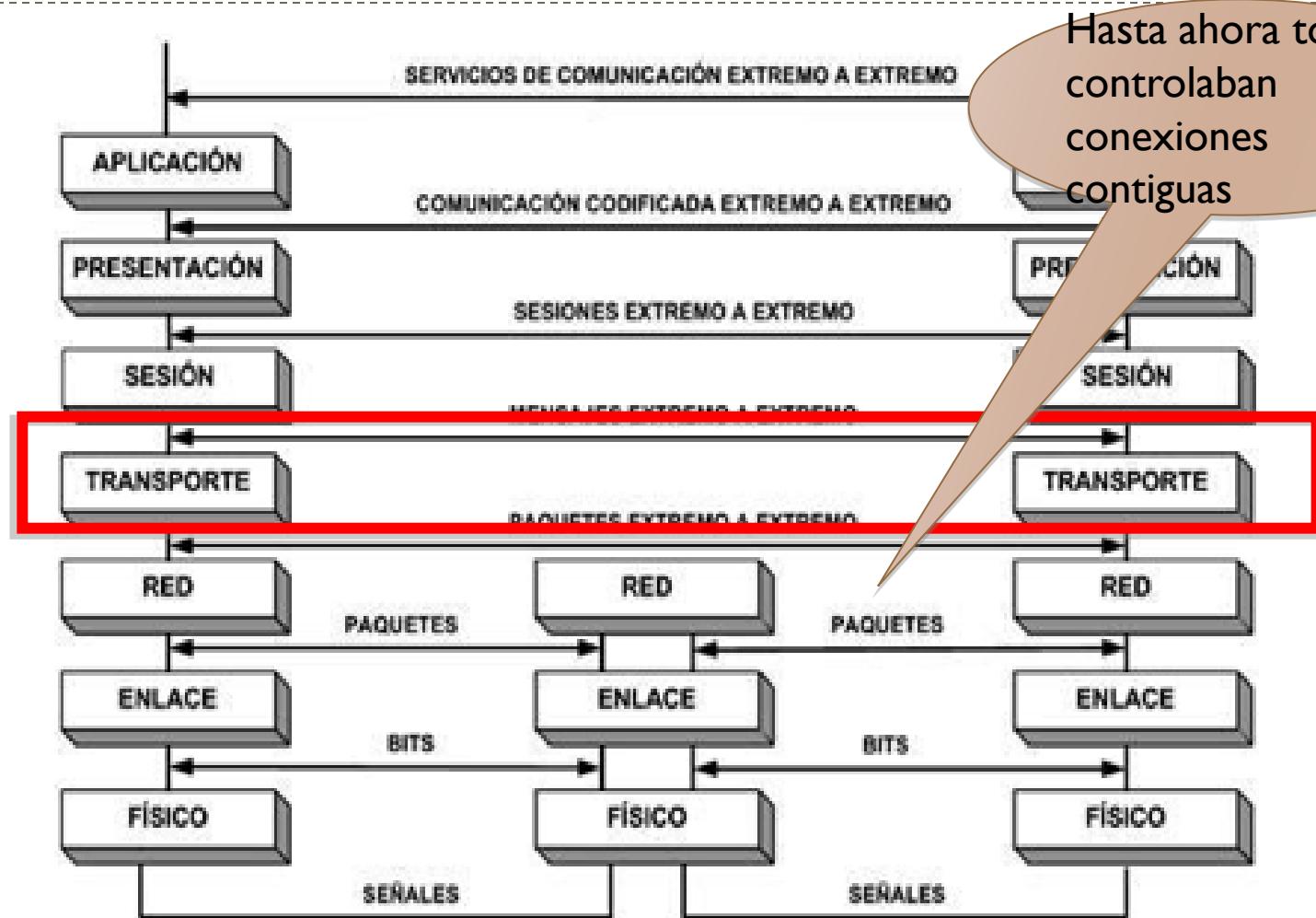
Muestre aquí su forma de proceder para el Problema 3:

El nivel de transporte.

- ▶ Será el encargado de llevar los datos de **extremo a extremo**, de la **aplicación origen a la destino**.
- ▶ Para ello realiza **conexiones lógicas entre los extremos** (**sesiones**), estableciendo un flujo de datos entre los extremos.
- ▶ Cogerá los datos del **nivel de aplicación** y los entregará al **nivel de red**.
- ▶ Dividirá los datos en **segmentos** y los enviará al receptor **en el mismo orden**.
- ▶ Dentro de la conexión, podrá realizar **control de errores** (extremo a extremo) y **control de flujo**.



El nivel de transporte.



Direccionamiento.

- ▶ La identificación de los usuarios (extremos) se realiza mediante:
 - ▶ **Dirección IP + número del puerto.**
 - ▶ TCP denomina **socket** a esta combinación.
- ▶ Un puerto representa a un usuario particular del servicio de transporte.
- ▶ El número de puerto es valor de 16 bits que se incluye en la cabecera del protocolo de nivel de transporte (**desde 1 hasta 65.536**)



Números de puerto.

- ▶ Los números de puertos se asignan como sigue:

Rango de números de puerto	Grupo de puertos
De 0 a 1023	Puertos bien conocidos (Contacto)
De 1024 a 49151	Puertos registrados
De 49152 a 65535	Puertos privados y/o dinámicos

numeros de puertos origen (valor > 1023).

- ▶ **Los host destino:** utilizan los puertos para seleccionar la aplicación adecuada.

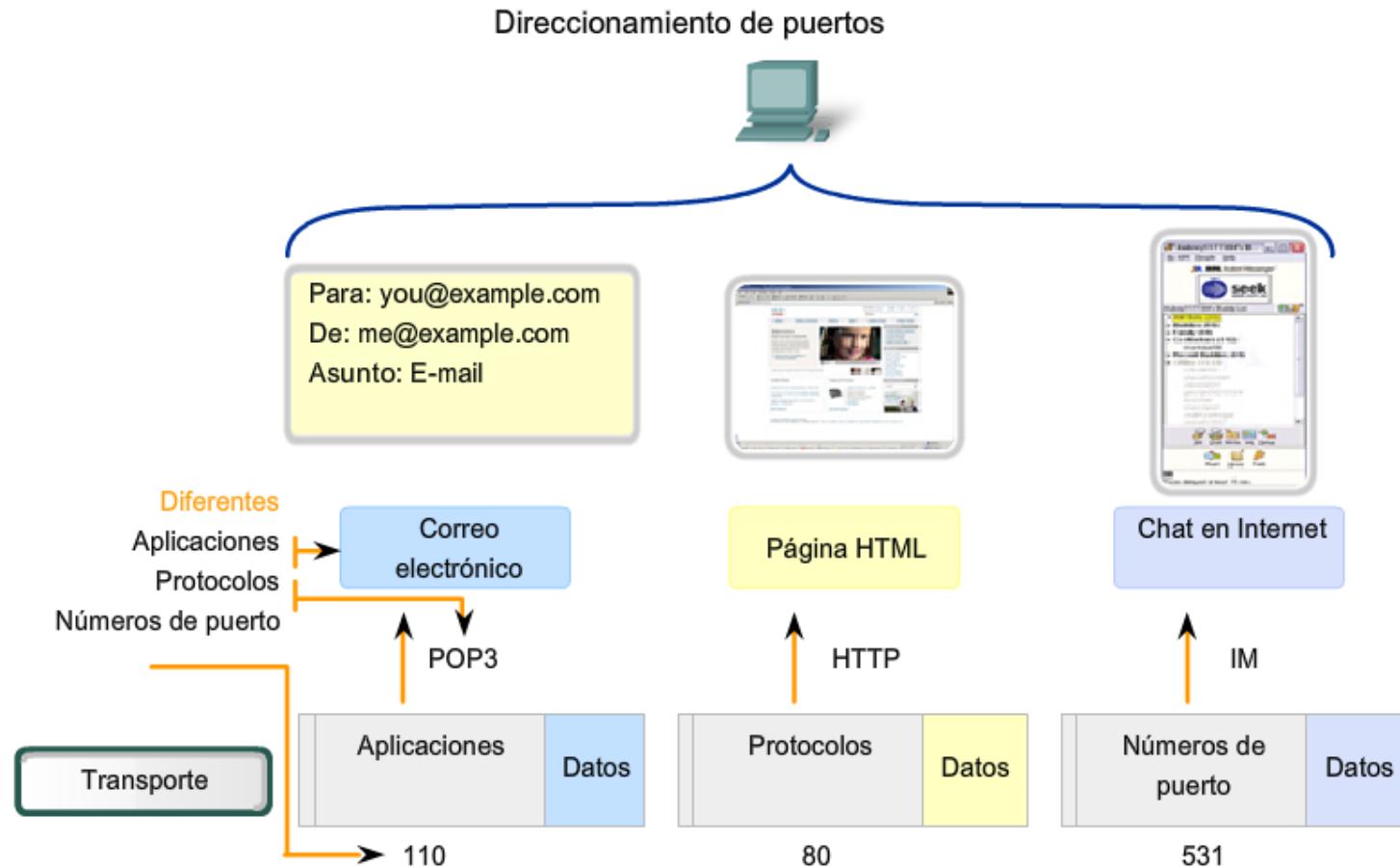


Puertos bien conocidos.

- ▶ Los números de puerto que corresponden a aplicaciones bien conocidas son:

Número de puerto	Aplicación
20	Ftp
21	Ftp
22	Ssh
23	Telnet
25	Ssmtp (correo entrante)
53	DNS
69	TFTP
80	Web
110	Pop3 (correo saliente)

Direccionamiento en el nivel de transporte.



Los datos de las distintas aplicaciones se dirigen a la aplicación correcta, ya que cada aplicación tiene un número de puerto único.



Arquitectura cliente-servidor.

- ▶ Para la comunicación de aplicaciones a través de una red se definen tres paradigmas:
- ▶ **Modelo cliente/servidor:** en él se distingue un proceso cliente y un proceso servidor.
- ▶ **Modelo entre pares o P2P:** todos los nodos de la red son responsables por igual en la comunicación de las aplicaciones y no existe ningún elemento que centralice la comunicación.
- ▶ **Modelo híbrido:** combinación de los dos anteriores y donde el servidor no presta el servicio como tal, sino que generalmente pone en contacto a los clientes para que se comuniquen entre sí.



Arquitectura cliente-servidor.

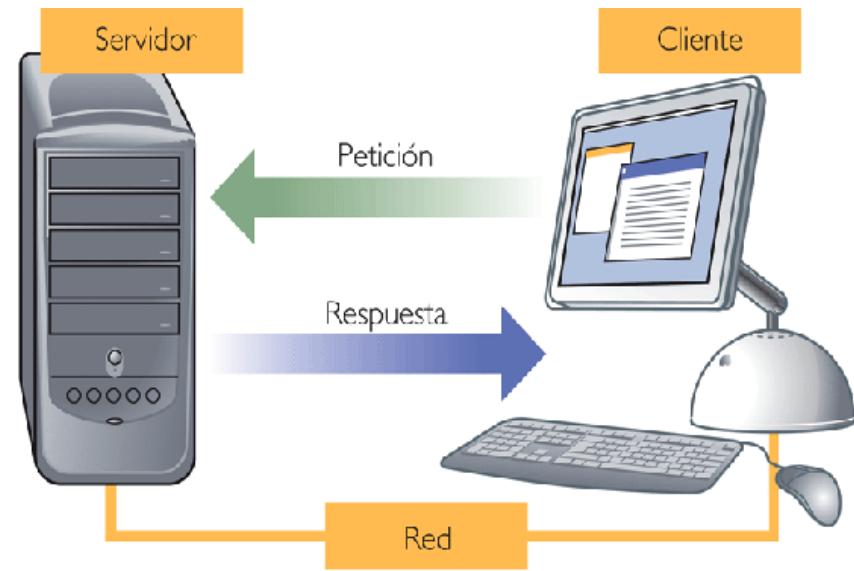
- ▶ La comunicación se produce a través de dos procesos que interactúan entre sí:

- ▶ Servidor:

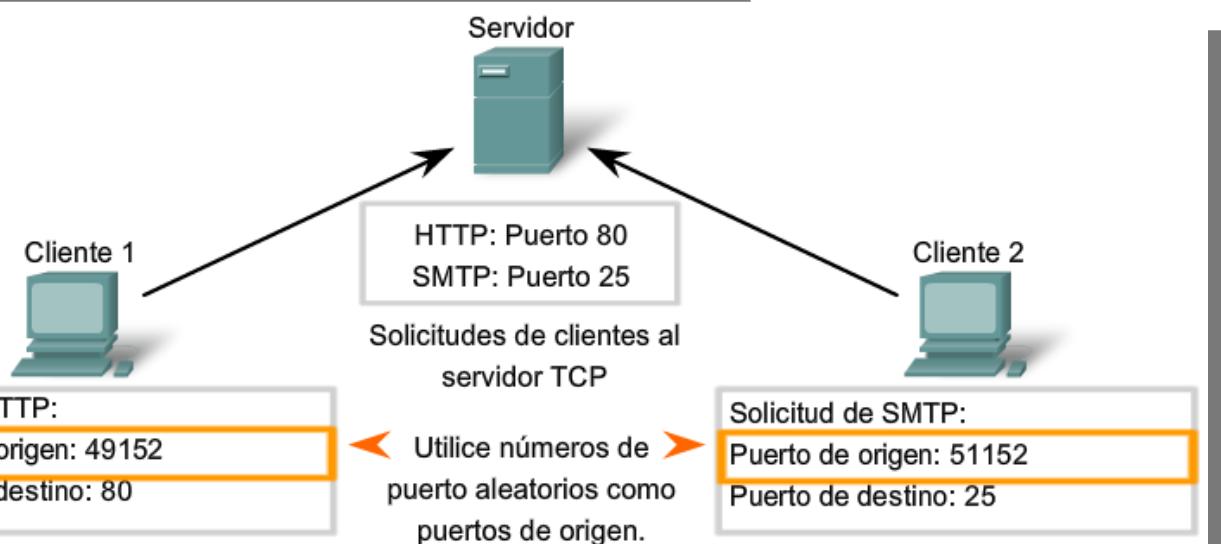
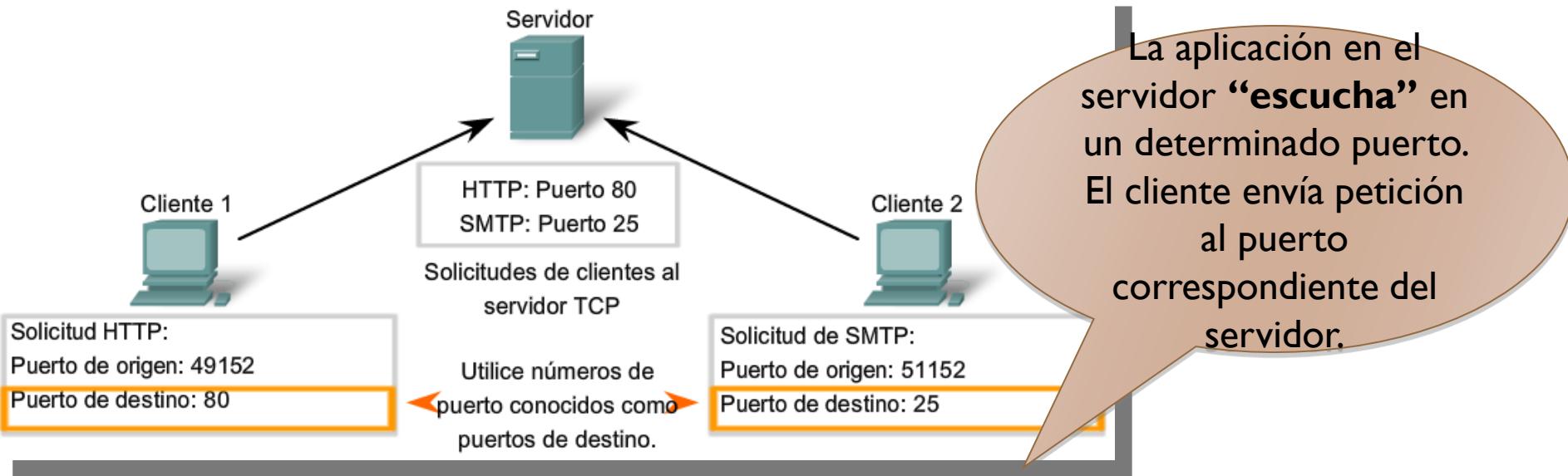
- Proporciona servicios a los clientes.
 - Espera que algún cliente se conecte a él con una petición.

- ▶ Cliente:

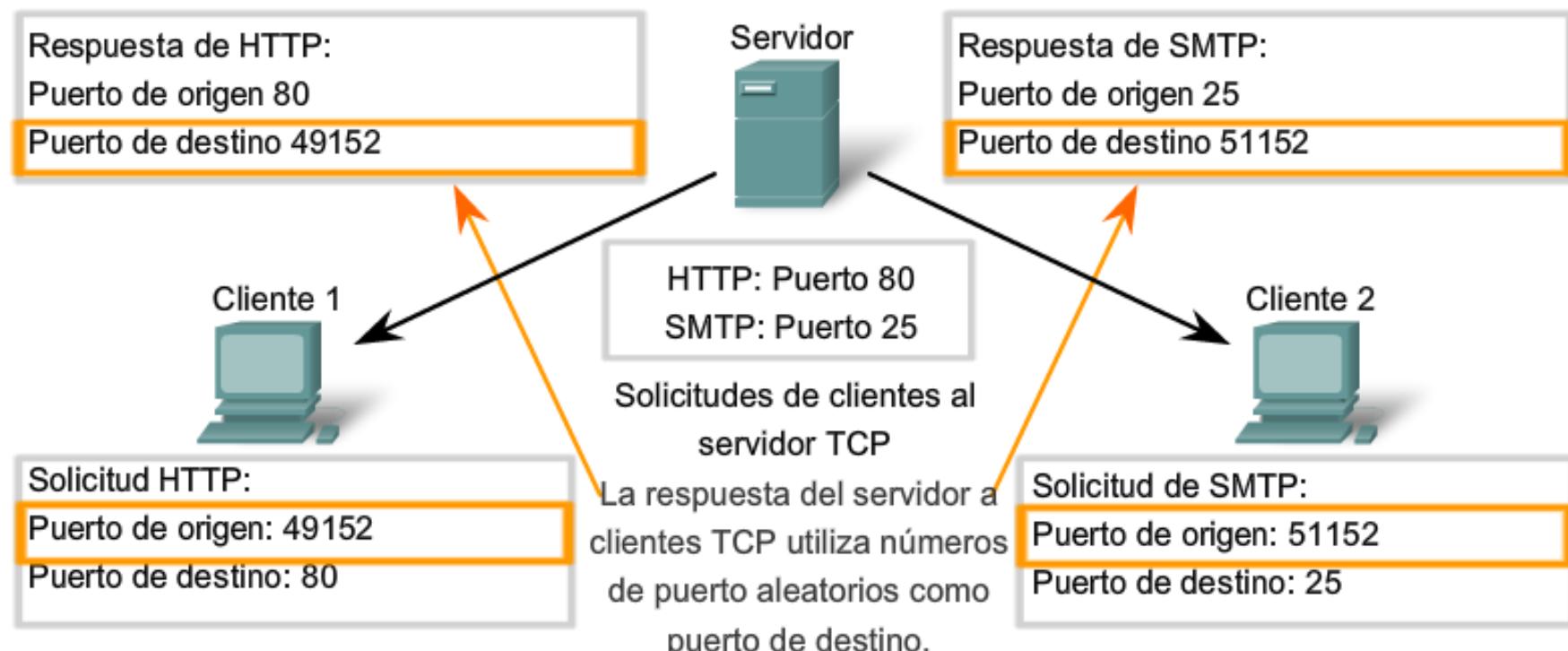
- Se conecta al servidor para obtener un servicio.



Arquitectura cliente-servidor.



Arquitectura cliente-servidor.



Servicios orientados y no orientados a conexión.

Cuando dos extremos se comunican pueden pasar dos cosas:

► **Que se establezca una sesión.**

- Que se establezca la comunicación en tres pasos.
 1. Inicio de la conexión.
 2. Envío de los datos.
 3. Cierre de la conexión.
- Ejemplo: conexión telefónica. Antes de poder hablar (enviar datos) se debe marcar y el otro extremo debe contestar.

**Protocolos Orientados
a conexión.**



Servicios orientados y no orientados a conexión.

- ▶ Que no se establezca una sesión, **sin sesión**:
 - ▶ Que se envíen los datos sin más. Estos pueden ser de dos tipos:
 - ▶ **“Envía y reza”**: envío los datos sin avisar y sin preocuparme de si llegan.
 - ▶ **Envíos con acuse de recibo**: envío los datos sin avisar pero al menos espero un acuse de recibo.

Protocolos no orientados a conexión.



TCP: Transmission Control Protocol.

- ▶ **Protocolo de nivel de transporte** usado en internet.
- ▶ Es un protocolo **orientado a conexión**.
- ▶ Esto significa que una **conexión** TCP se realiza **en tres fases**:
 1. **Establecimiento** de la conexión.
 2. **Envío** de los datos.
 3. **Desconexión** o terminación de la conexión.
- ▶ Se encarga de establecer un flujo de bytes entre extremos que TCP dividirá en **segmentos**.
- ▶ Un segmento estará formado por:
 - Una cabecera
 - Una porción de datos de usuario.
 - Algunos segmentos pueden no llevar datos de usuario. Ejemplo: segmentos usados para establecer o liberar conexiones.
 - Los segmentos viajan contenidos en paquetes IP (**encapsulamiento**).



TCP: características.

► **TCP es un protocolo fiable:**

- ▶ Los segmentos están formados una cantidad de bytes.
- ▶ Cada byte se enumera con un **número de secuencia (SN)**.
- ▶ Envía **ACK** cuando ha recibido un cierto número de segmentos correctos.
- ▶ Inicia un timer cada vez que envía un segmento. Si expira, se retransmite.
- ▶ Los números de secuencia servirán para que el receptor ordene los datos.

► **Control de flujo:** La técnica anterior (envío de ACKs) sirve evitar que el receptor se sature (**ventana de transmisión**).

► **Control de errores:** mediante checksum.

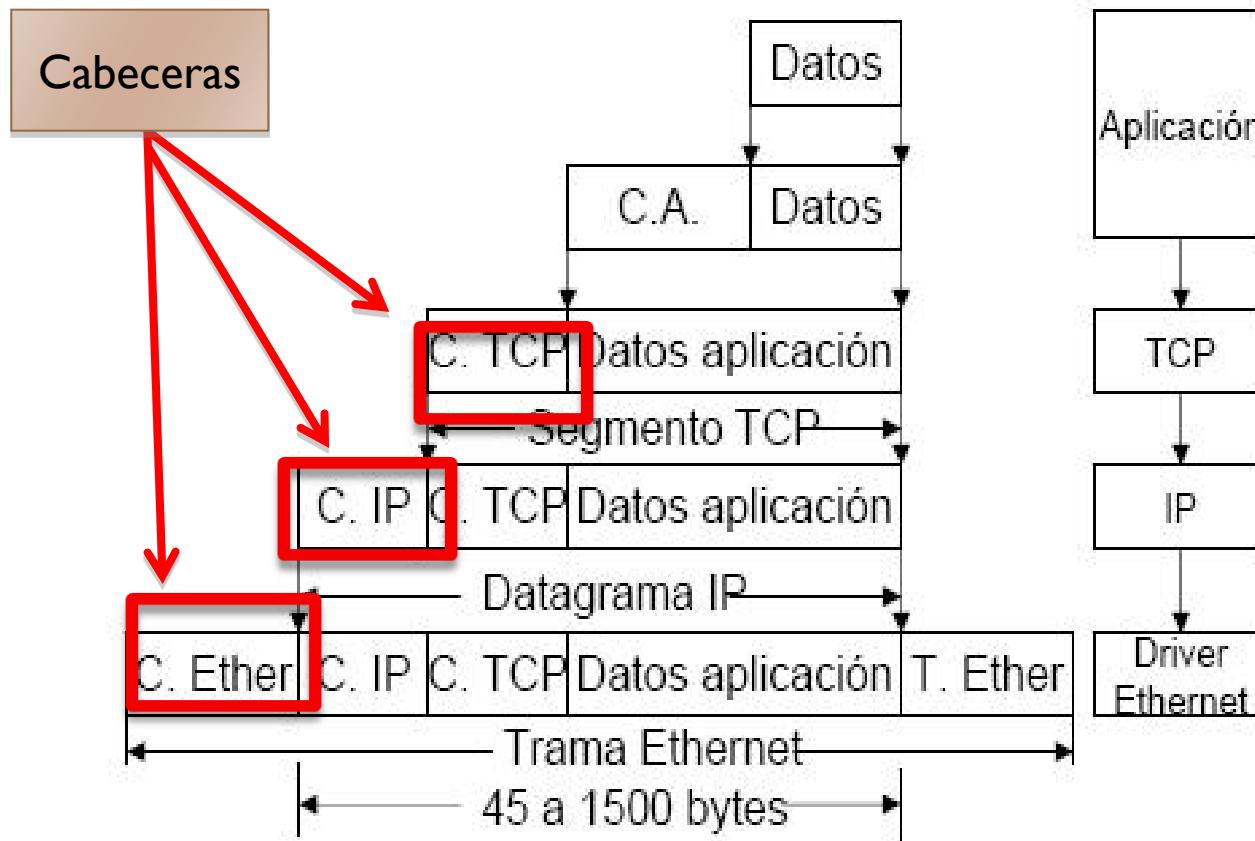
► **Todo lo anterior se realizaba también en nivel de enlace.**

- ▶ Nivel de transporte: proporciona servicios *extremo a extremo*.
- ▶ Nivel de enlace: entre dos conexiones consecutivas.

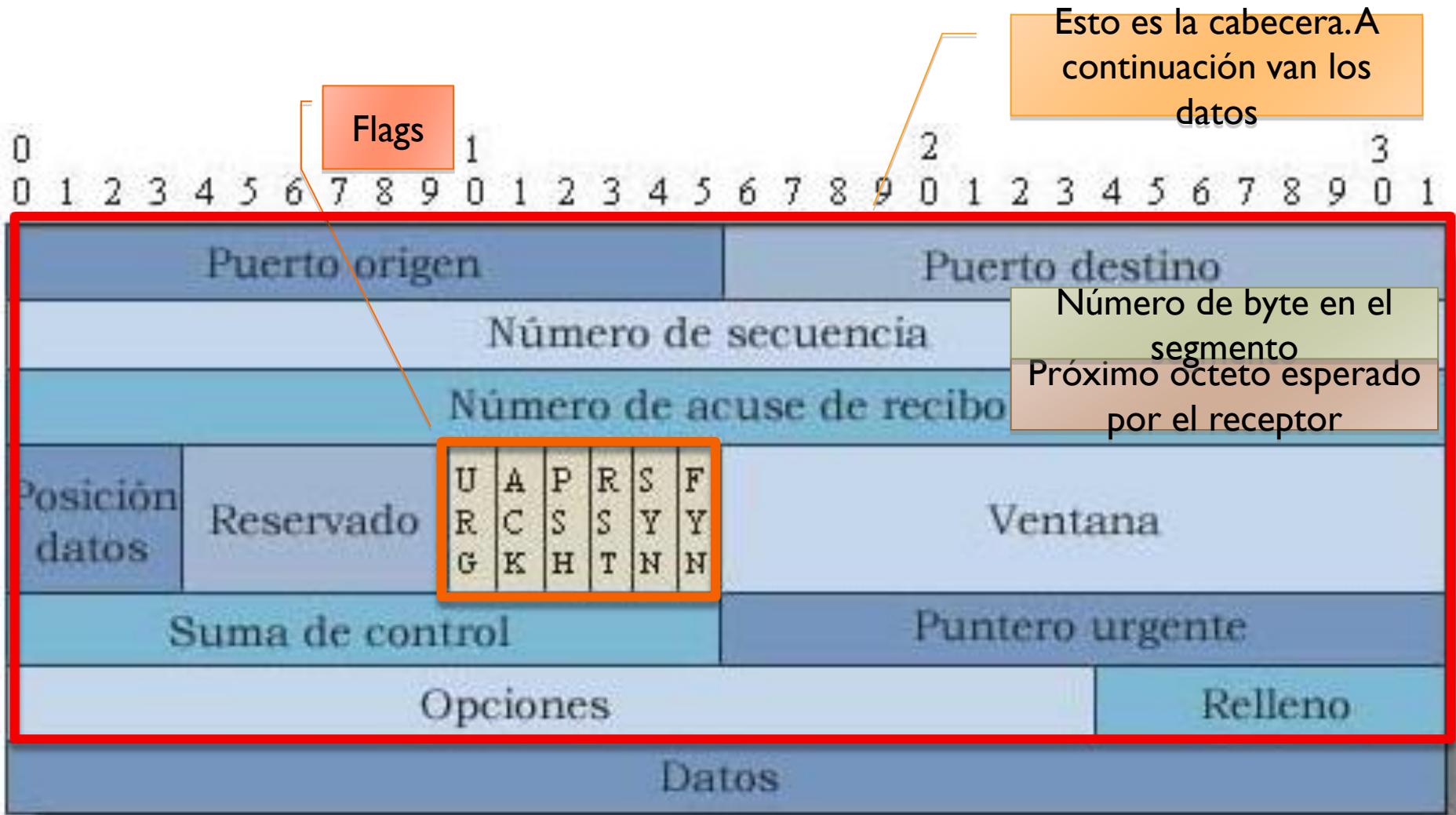


Cabecera TCP.

Encapsulación



Cabecera TCP.



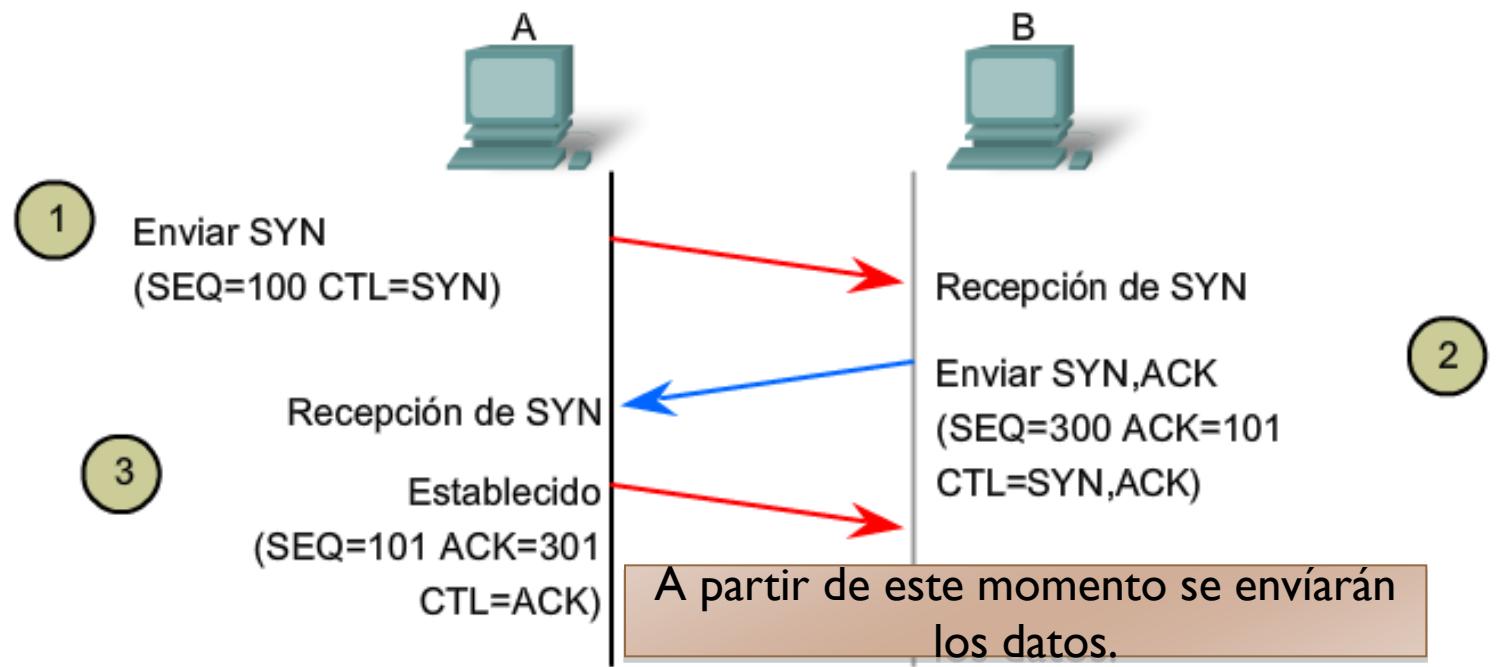
Flags en la cabecera TCP

1. **URG** El puntero urgente (*urgent pointer*) es válido.
2. **ACK** el número de confirmación (*acknowledgment number*) es válido.
3. **PSH** El receptor debe pasar estos datos a la aplicación tan pronto como sea posible.
4. **RST** Conexión perdida (*reset*).
5. **SYN** Sincronizar números de secuencia para iniciar una conexión.
6. **FIN** El remitente ha terminado el envío de datos.



TCP: Inicio de sesión.

- ▶ **Saludo a tres bandas.** Para establecer una conexión TCP se siguen los pasos siguientes:



CTL = Qué bits de control en el encabezado TCP están establecidos en 1

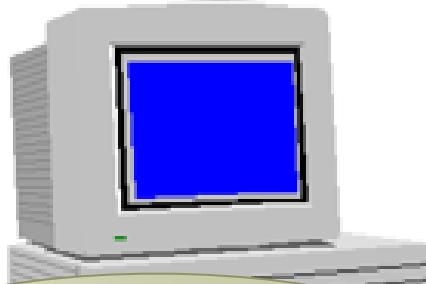
A envía la respuesta de ACK a B.

TCP: Inicio de sesión.

Los dos extremos envían en qué número de secuencia empiezan a numerar para que el contrario pueda ordenar los bytes.

En el ACK envían cual es el siguiente que espera.

Mi primer n° de secuencia es “a”.



Vale, pues el siguiente que me debes mandar es , $b+1$.

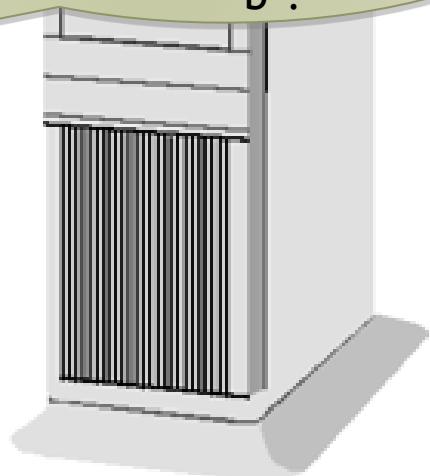
HOST CONFIABLE A

SYN sec = a

SYN sec = b, ACK a+1

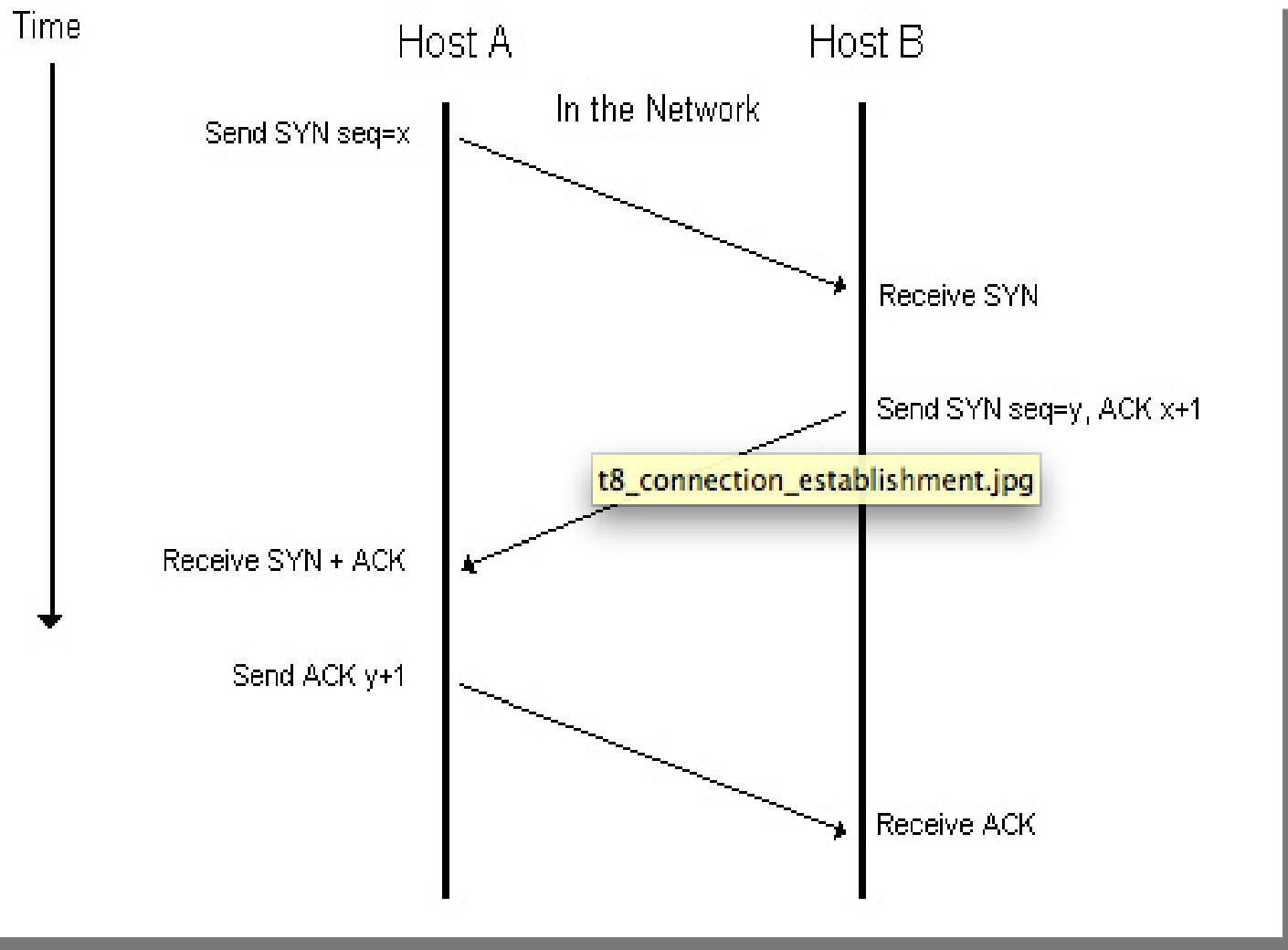
ACK b+1

I. Vale, pues mándame ahora el “ $a+1$ ”.
2. 2. El mío, mi primer n° de secuencia es “b”.



HOST DESTINO B

TCP: Inicio de sesión.



TCP: Inicio de sesión.

SYN: sec=0,ACK

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
5	0.261859	192.168.1.40	209.85.146.101	TCP	57310 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=3 TSV=439666246 TSER=0
6	0.350440	209.85.146.101	192.168.1.40	TCP	http > 57310 [SYN, ACK] Seq=0 Ack=1 Win=5672 Len=0 MSS=1430 TSV=224565236 TSER=0
7	0.350526	192.168.1.40	209.85.146.101	TCP	57310 > http [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSV=439666247 TSER=224565236
8	0.350948	192.168.1.40	209.85.146.101	HTTP	GET /complete/search?client=chrome&hl=en-US&q=h HTTP/1.1
9	0.448671	209.85.146.101	192.168.1.40	TCP	http > 57310 [ACK] Seq=1 Ack=597 Win=6912 Len=0 TSV=224565334 TSER=439666247

Frame 5 (78 bytes on wire, 78 bytes captured)
Ethernet II, Src: Apple_ae:fa:2a (00:1e:c2:ae:fa:2a), Dst: XaviTech_24:1a:a9 (e0:91:53:24:1a:a9)
Internet Protocol, Src: 192.168.1.40 (192.168.1.40), Dst: 209.85.146.101 (209.85.146.101)
Transmission Control Protocol, Src Port: 57310 (57310), Dst Port: http (80), Seq: 0, Len: 0

Source port: 57310 (57310)
Destination port: http (80)
[Stream index: 2]
Sequence number: 0 (relative sequence number)
Header length: 44 bytes

Flags: 0x02 (SYN)
0... = Congestion Window Reduced (CWR): Not set
.0... = ECN-Echo: Not set
..0. = Urgent: Not set
...0 = Acknowledgement: Not set
.... 0... = Push: Not set
.... .0.. = Reset: Not set

.... .1. = Syn: Set
.... .0 = Fin: Not set

0000 e0 91 53 24 1a a9 00 1e c2 ae fa 2a 08 00 45 00 .S\$.... .*.E.
0010 00 40 91 64 40 00 40 06 83 c8 c0 a8 01 28 d1 55 .@.d@.a.U
0020 92 65 df de 00 50 61 f4 d4 5f 00 00 00 00 b0 02 .e...Pa.
0030 ff ff 1a 76 00 00 02 04 05 b4 01 03 03 03 01 01v....
0040 08 0a 1a 34 c6 46 00 00 00 00 04 02 00 004.F.

Ethernet (eth), 14 bytes Packets: 1228 Displayed: 1228 Marked: 0 Profile: Default

TCP: Inicio de sesión.

SYN sec=0,ACK: I

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: **tcp** Expression... Clear Apply

No..	Time	Source	Destination	Protocol	Info
5	0.261859	192.168.1.40	209.85.146.101	TCP	57310 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=3 TSV=439666246 TSER=0
6	0.350440	209.85.146.101	192.168.1.40	TCP	http > 57310 [SYN, ACK] Seq=0 Ack=1 Win=5672 Len=0 MSS=1430 TSV=224565236 TSER=0
7	0.350526	192.168.1.40	209.85.146.101	TCP	57310 > http [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSV=439666247 TSER=224565236
8	0.350948	192.168.1.40	209.85.146.101	HTTP	GET /complete/search?client=chrome&hl=en-US&q=h HTTP/1.1
9	0.448671	209.85.146.101	192.168.1.40	TCP	http > 57310 [ACK] Seq=1 Ack=597 Win=6912 Len=0 TSV=224565334 TSER=439666247
10	0.472202	209.85.146.101	192.168.1.40	HTTP	HTTP/1.1 200 OK (text/javascript)

Frame 6 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: XaviTech_24:1a:a9 (e0:91:53:24:1a:a9), Dst: Apple_ae:fa:2a (00:1e:c2:ae:fa:2a)
Internet Protocol, Src: 209.85.146.101 (209.85.146.101), Dst: 192.168.1.40 (192.168.1.40)
Transmission Control Protocol, Src Port: http (80), Dst Port: 57310 (57310), Seq: 0, Ack: 1, Len: 0

Source port: http (80)
Destination port: 57310 (57310)
[Stream index: 2]
Sequence number: 0 (relative sequence number)
Acknowledgement number: 1 (relative ack number)
Header length: 40 bytes

Flags: 0x12 (SYN, ACK)
Window size: 5672
Checksum: 0xb5aa [validation disabled]
Options: (20 bytes)
[SEQ/ACK analysis]

TCP: Inicio de sesión.

ACK: 1

File Edit View Go Capture Analyze Statistics Telephony Tools Help

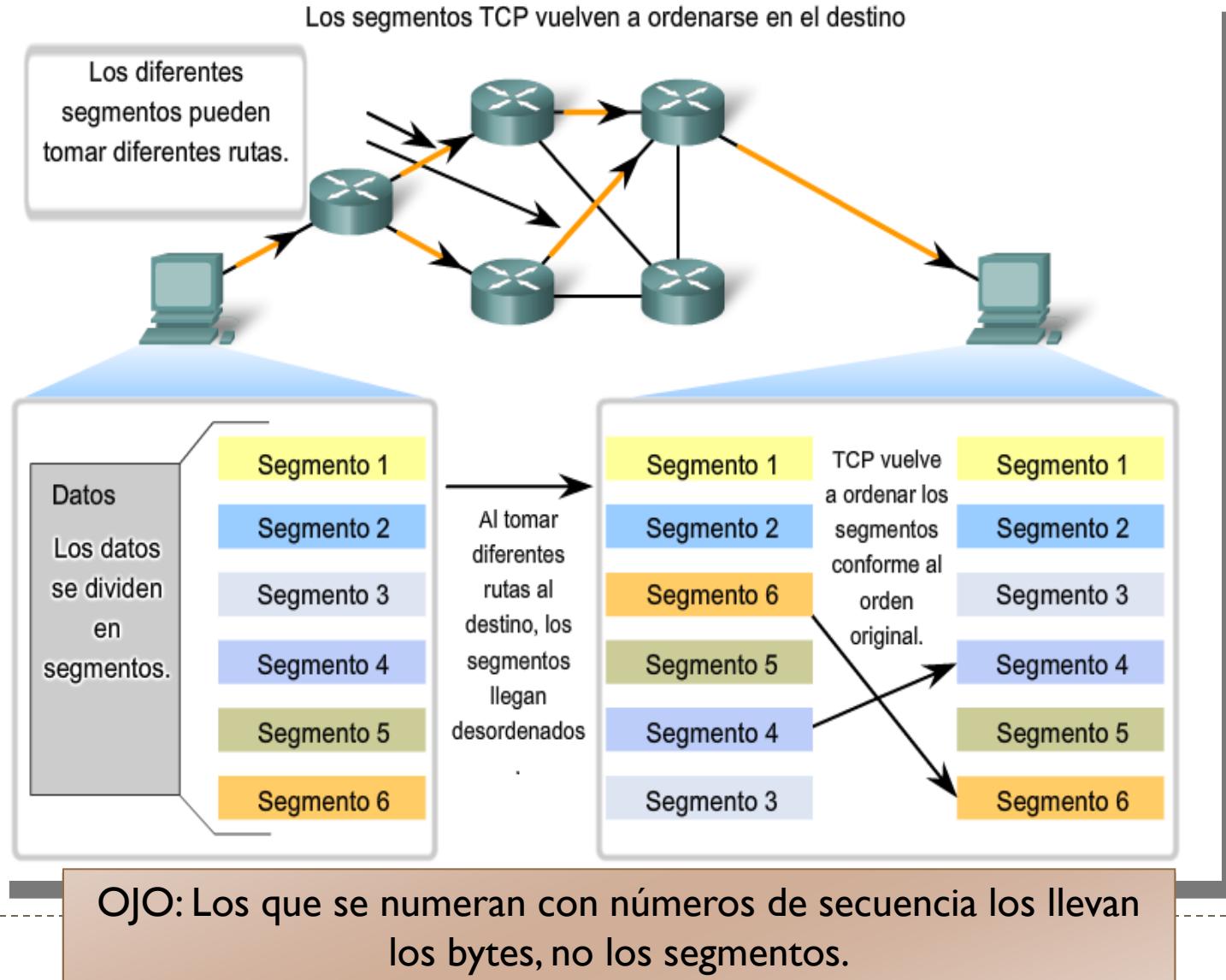
Filter: **tcp** Expression... Clear Apply

No..	Time	Source	Destination	Protocol	Info
5	0.261859	192.168.1.40	209.85.146.101	TCP	57310 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=3 TSV=439666246 TSER=0
6	0.350440	209.85.146.101	192.168.1.40	TCP	http > 57310 [SYN, ACK] Seq=0 Ack=1 Win=5672 Len=0 MSS=1430 TSV=224565236 TSER=0
7	0.350526	192.168.1.40	209.85.146.101	TCP	57310 > http [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSV=439666247 TSER=224565236
8	0.350948	192.168.1.40	209.85.146.101	HTTP	GET /complete/search?client=chrome&hl=en-US&q=h HTTP/1.1
9	0.448671	209.85.146.101	192.168.1.40	TCP	http > 57310 [ACK] Seq=1 Ack=597 Win=6912 Len=0 TSV=224565334 TSER=439666247
10	0.472292	209.85.146.101	192.168.1.40	HTTP	HTTP/1.1 200 OK (text/javascript)

▷ Frame 7 (66 bytes on wire, 66 bytes captured)
▷ Ethernet II, Src: Apple_ae:fa:2a (00:1e:c2:ae:fa:2a), Dst: XaviTech_24:1a:a9 (e0:91:53:24:1a:a9)
▷ Internet Protocol, Src: 192.168.1.40 (192.168.1.40), Dst: 209.85.146.101 (209.85.146.101)
▽ Transmission Control Protocol, Src Port: 57310 (57310), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0

Source port: 57310 (57310)
Destination port: http (80)
[Stream index: 2]
Sequence number: 1 (relative sequence number)
Acknowledgement number: 1 (relative ack number)
Header length: 32 bytes
▷ Flags: 0x10 (ACK)
Window size: 524280 (scaled)
▷ Checksum: 0xfa7e [validation disabled]
▷ Options: (12 bytes)
▷ [SEQ/ACK analysis]

TCP: Envío de datos.



TCP: Envío de datos.

Te envío la Sec=1,
espero de ti el 597

Screenshot of Wireshark showing a network capture of a TCP connection. The packet list shows several HTTP and TCP packets. An orange callout bubble is positioned above the packet list, containing the text "Te envío la Sec=1, espero de ti el 597".

No..	Time	Source	Destination	Protocol	Info
8	0.350948	192.168.1.40	209.85.146.101	HTTP	GET /complete/search?client=chrome&hl=en-US&q=%20 HTTP/1.1
9	0.448671	209.85.146.101	192.168.1.40	TCP	http > 57310 [ACK] Seq=1 Ack=597 Win=6912 Len=0 TSV=224565334 TSER=439666247
10	0.472292	209.85.146.101	192.168.1.40	HTTP	HTTP/1.1 200 OK (text/javascript)
11	0.472369	192.168.1.40	209.85.146.101	TCP	57310 > http [ACK] Seq=597 Ack=402 Win=524256 Len=0 TSV=439666248 TSER=22456535
12	1.796061	192.168.1.40	209.85.146.101	HTTP	GET /complete/search?client=chrome&hl=en-US&q=http%3A%2F%2Fwww.facebook.com HTT
13	1.918671	209.85.146.101	192.168.1.40	HTTP	HTTP/1.1 200 OK (text/javascript)

Frame 9 (66 bytes on wire, 66 bytes captured)
Ethernet II, Src: XaviTech_24:1a:a9 (e0:91:53:24:1a:a9), Dst: Apple_ae:fa:2a (00:1e:c2:ae:fa:2a)
Internet Protocol, Src: 209.85.146.101 (209.85.146.101), Dst: 192.168.1.40 (192.168.1.40)
Transmission Control Protocol, Src Port: http (80), Dst Port: 57310 (57310), Seq: 1, Ack: 597, Len: 0
Source port: http (80)
Destination port: 57310 (57310)
[Stream index: 2]
Sequence number: 1 (relative sequence number)
Acknowledgement number: 597 (relative ack number)
Header length: 32 bytes
Flags: 0x10 (ACK)
Window size: 6912 (scaled)
Checksum: 0xf75c [validation disabled]
Options: (12 bytes)
[SEQ/ACK analysis]

TCP: Envío de datos.

Te envío la
Sec=597 espero de
ti el 402

No.	Time	Source	Destination	Protocol	Info
10 0.472229		209.85.146.101	192.168.1.40	HTTP	HTTP/1.1 200 OK (text/javascript)
11 0.472369		192.168.1.40	209.85.146.101	TCP	57310 > http [ACK] Seq=597 Ack=402 Win=524256 Len=0 TSV=439666248 TSER=22456535
12 1.796061		192.168.1.40	209.85.146.101	HTTP	GET /complete/search?client=chrome&hl=en-US&q=http%3A%2F%2Fwww.facebook.com HTT
13 1.918671		209.85.146.101	192.168.1.40	HTTP	HTTP/1.1 200 OK (text/javascript)
14 1.918799		192.168.1.40	209.85.146.101	TCP	57310 > http [ACK] Seq=1221 Ack=842 Win=524216 Len=0 TSV=439666263 TSER=2245668
16 2.197539		192.168.1.40	209.85.146.101	HTTP	GET /complete/search?client=chrome&hl=en-US&q=http%3A%2F%2Fwww.facebook.co HTT

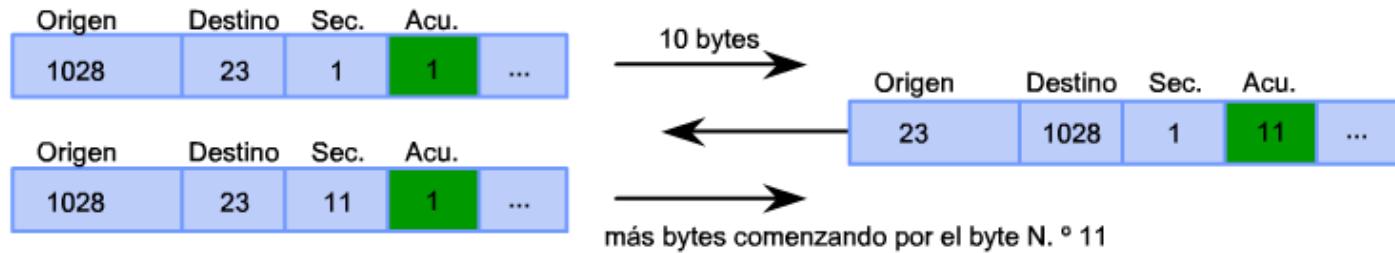
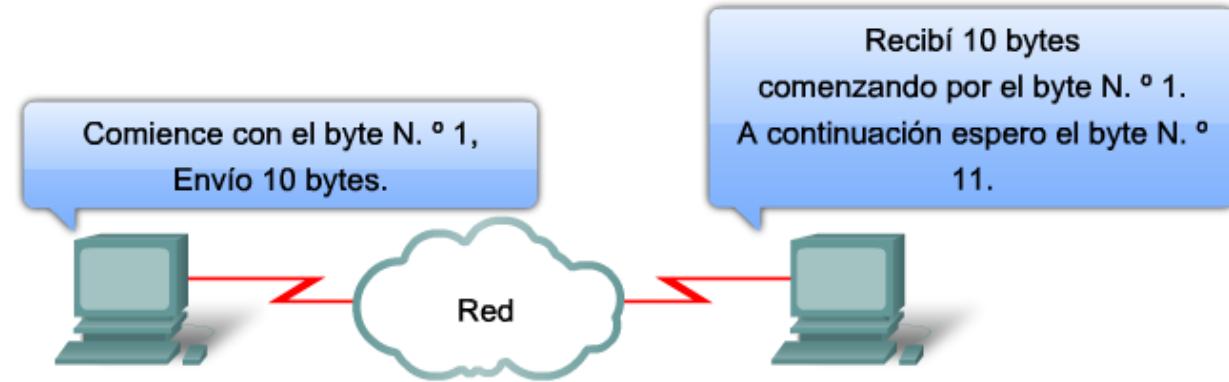
Frame 11 (66 bytes on wire, 66 bytes captured)
Ethernet II, Src: Apple_ae:fa:2a (00:1e:c2:ae:fa:2a), Dst: XaviTech_24:1a:a9 (e0:91:53:24:1a:a9)
Internet Protocol, Src: 192.168.1.40 (192.168.1.40), Dst: 209.85.146.101 (209.85.146.101)
Transmission Control Protocol, Src Port: 57310 (57310), Dst Port: http (80), Seq: 597, Ack: 402, Len: 0

Source port: 57310 (57310)
Destination port: http (80)
[Stream index: 2]
Sequence number: 597 (relative sequence number)
Acknowledgement number: 402 (relative ack number)
Header length: 32 bytes
Flags: 0x10 (ACK)
Window size: 524256 (scaled)
Checksum: 0xf622 [validation disabled]
Options: (12 bytes)
[SEQ/ACK analysis]

TCP: Envío de datos.

Para ver en Wireshark cuántos bytes enviamos, hay que verlo en el campo LEN mensaje inmediatamente anterior. Eso explica la variación de los números de secuencia.

Puerto de origen	Puerto de destino	Número de secuencia	Números de acuse de recibo	...
------------------	-------------------	---------------------	----------------------------	-----



TCP: Envío de datos.

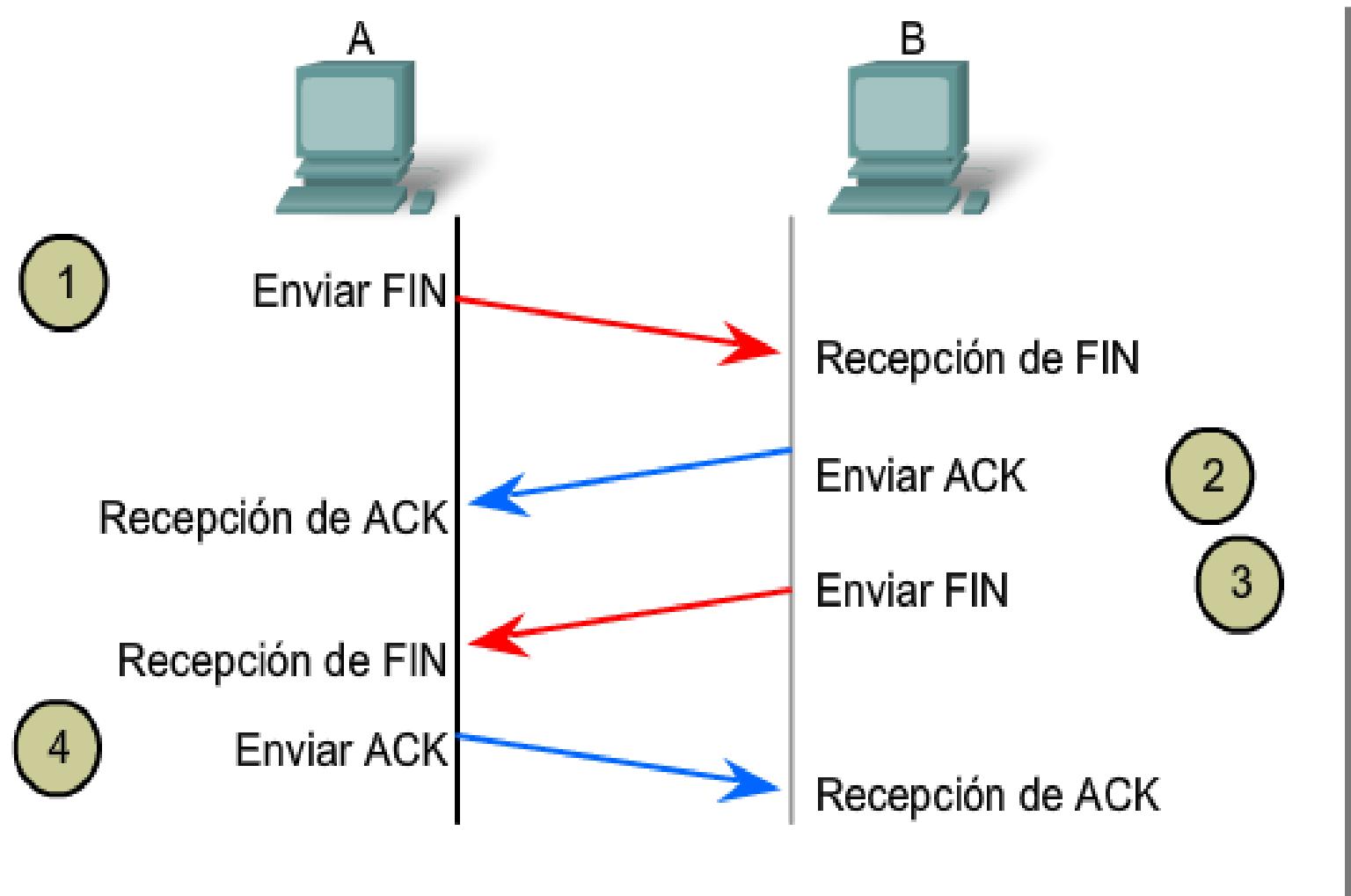
La trama entre las dos anteriores, envió 401 bytes.

9 0.44486/1	209.85.146.101	192.168.1.40	TCP	57310 > http [ACK] Seq=597 Ack=402 Win=524256 Len=0 TSV=439666248 TSER=224565344	TSER=439666247
10 0.472292	209.85.146.101	192.168.1.40	HTTP	GET /complete/search?client=chrome&hl=en-US&q=http%3A%2F%2Fwww.facebook.com	HTTP/1.1 200 OK (text/javascript)
11 0.472369	192.168.1.40	209.85.146.101	TCP	57310 > http [ACK] Seq=597 Ack=402 Win=524256 Len=0 TSV=439666248 TSER=22456535	
12 1.796061	192.168.1.40	209.85.146.101	HTTP	GET /complete/search?client=chrome&hl=en-US&q=http%3A%2F%2Fwww.facebook.com	HTTP/1.1 200 OK (text/javascript)
13 1.918671	209.85.146.101	192.168.1.40	HTTP	HTTP/1.1 200 OK (text/javascript)	
14 1.918799	192.168.1.40	209.85.146.101	TCP	57310 > http [ACK] Seq=1221 Ack=842 Win=524216 Len=0 TSV=439666263 TSER=2245668	

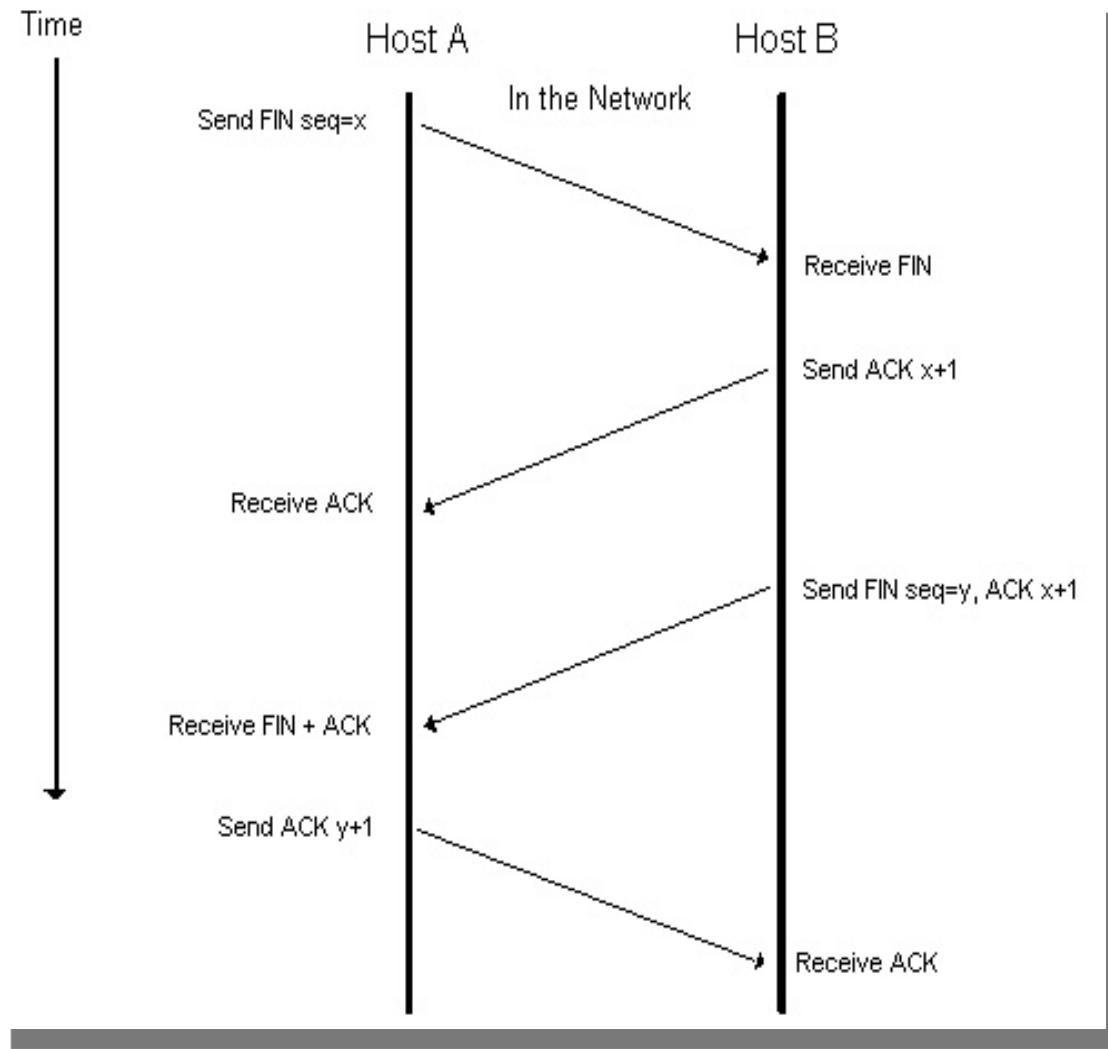
Frame 10 (467 bytes on wire, 467 bytes captured)
Ethernet II, Src: XaviTech_24:1a:a9 (e0:91:53:24:1a:a9), Dst: Apple_ae:fa:2a (00:1e:c2:ae:fa:2a)
Internet Protocol, Src: 209.85.146.101 (209.85.146.101), Dst: 192.168.1.40 (192.168.1.40)
Transmission Control Protocol, Src Port: http (80), Dst Port: 57310 (57310), Seq: 1, Ack: 597, Len: 467

Source port: http (80)
Destination port: 57310 (57310)
[Stream index: 2]
Sequence number: 1 (relative sequence number)
[Next sequence number: 402 (relative sequence number)]
Acknowledgement number: 597 (relative ack number)
Header length: 32 bytes
Flags: 0x18 (PSH, ACK)
Window size: 6912 (scaled)
Checksum: 0xe9fd [validation disabled]
Options: (12 bytes)
[SEQ/ACK analysis]
Hypertext Transfer Protocol
Line-based text data: text/javascript

TCP: Fin de sesión.



TCP: Fin de sesión.



TCP: Estados de los puertos

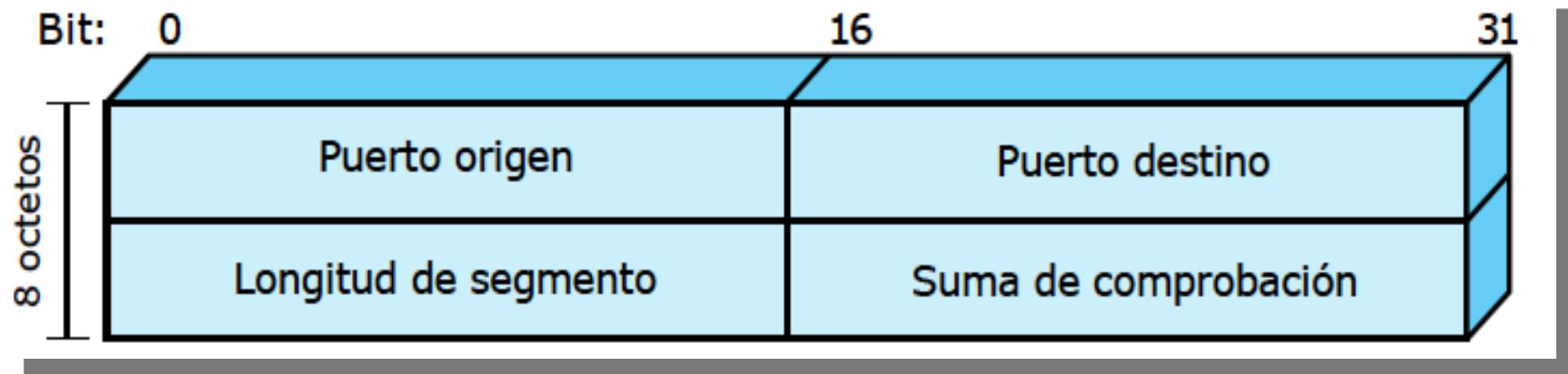
Estado	Descripción
CLOSED	No hay conexiones activas o pendientes
LISTEN	El servidor espera peticiones de conexión
SYN RCVD	Se ha recibido una petición de conexión; esperar ACK
SYN SENT	La aplicación ha iniciado la apertura de una conexión
ESTABLISHED	Estado de transferencia de datos normal
FIN WAIT 1	La aplicación ha indicado que ha terminado
FIN WAIT 2	El otro extremo está de acuerdo en liberar la conexión
TIMED WAIT	Espera a que llegue ACK de la petición de cierre
CLOSING	Ambos extremos han intentado cerrar simultáneamente
CLOSE WAIT	El otro extremo ha iniciado la liberación de la conexión
LAST ACK	Espera a que llegue ACK de la petición de cierre

UDP: User Datagram Protocol.

- ▶ Protocolo de nivel de transporte usado en internet.
- ▶ **Sin conexión:**
 - permite enviar datos sin haber establecido una conexión previa.
- ▶ **No fiable:**
 - Tiene la responsabilidad de enviar los datos, pero no la obligación de verificar la entrega de los mismos.
 - Los paquetes pueden llegar desordenados o duplicados.
- ▶ Es un protocolo de nivel de transporte no fiable y no orientado a conexión.
- ▶ Ventaja:
 - Velocidad: al no enviar ACK, se envía menos cantidad de datos lo que agiliza la transferencia.



UDP: Cabecera IP.



UDP: Aplicaciones.

- ▶ UDP se utiliza en los casos en los que la sobrecarga que supone el establecimiento y cierre de la conexión no está justificado.
- ▶ Algunas aplicaciones que usan UDP:
 - ▶ Aplicaciones de tiempo real (RTP).
 - ▶ Sistemas de denominación de nombres (DNS).
 - ▶ Protocolo de administración de redes (SNMP).
 - ▶ Protocolo de configuración dinámica (DHCP).
 - ▶ Protocolo de transferencia trivial de ficheros (TFTP).



El comando netstat.

- ▶ **Netstat** es una herramienta de red que informa sobre:
 - ▶ Tablas de enrutamiento: **netstat -r**
 - ▶ Estadísticas de las interfaces: **netstat -i**
 - ▶ Conexiones establecidas con una máquina:
 - ▶ **netstat**
 - ▶ **netstat -a**
 - ▶ **netstat -u**
 - ▶ **netstat -t**
- ▶ Esta utilidad está disponible en los sistemas operativos: Linux, Windows, Mac OS, ...



Netstat

Protocolo	Puerto local	Puerto destino	Estado de la conexión
TCP	afernani-2daec3:1033	localhost:27015	ESTABLISHED
TCP	afernani-2daec3:1100	localhost:39000	ESTABLISHED
TCP	afernani-2daec3:1101	localhost:39000	ESTABLISHED
TCP	afernani-2daec3:27015	localhost:1033	ESTABLISHED
TCP	afernani-2daec3:39000	localhost:1100	ESTABLISHED
TCP	afernani-2daec3:39000	localhost:1101	ESTABLISHED
TCP	afernani-2daec3:1143	wy-in-f17.1e100.net:https	TIME_WAIT
TCP	afernani-2daec3:1150	wy-in-f187.1e100.net:smtp	TIME_WAIT
TCP	afernani-2daec3:1156	wy-in-f109.1e100.net:587	TIME_WAIT
TCP	afernani-2daec3:1159	ww-in-f109.1e100.net:995	TIME_WAIT

Host remoto

Puertos bien conocidos sustituidos por el nombre de la aplicación.

Netstat

```
C:\Documents and Settings\afernan1>netstat -a
```

Conexiones activas

Proto	Dirección local	Dirección remota	Estado
TCP	afernan1-2daec3:ftp	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:epmap	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:microsoft-ds	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:38000	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:39000	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:1031	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:1033	localhost:27015	ESTABLISHED
TCP	afernan1-2daec3:1100	localhost:39000	ESTABLISHED
TCP	afernan1-2daec3:1101	localhost:39000	ESTABLISHED
TCP	afernan1-2daec3:5354	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:14147	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:27015	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:27015	localhost:1033	ESTABLISHED
TCP	afernan1-2daec3:39000	localhost:1100	ESTABLISHED
TCP	afernan1-2daec3:39000	localhost:1101	ESTABLISHED
TCP	afernan1-2daec3:netbios-ssn	afernan1-2daec3:0	LISTENING
TCP	afernan1-2daec3:1156	wy-in-f109.1e100.net:587	TIME_WAIT
TCP	afernan1-2daec3:1159	wy-in-f109.1e100.net:995	TIME_WAIT
TCP	afernan1-2daec3:1161	xglobe.dmarc.sii.atlanticmetro.net:http	CLOSE_WAIT
UDP	afernan1-2daec3:microsoft-ds	***	
UDP	afernan1-2daec3:isakmp	***	
UDP	afernan1-2daec3:1027	***	
UDP	afernan1-2daec3:4500	***	
UDP	afernan1-2daec3:ntp	***	
UDP	afernan1-2daec3:1025	***	
UDP	afernan1-2daec3:1026	***	
UDP	afernan1-2daec3:1034	***	
UDP	afernan1-2daec3:1035	***	
UDP	afernan1-2daec3:1040	***	
UDP	afernan1-2daec3:1093	***	
UDP	afernan1-2daec3:1102	***	
UDP	afernan1-2daec3:1900	***	
UDP	afernan1-2daec3:ntp	***	
UDP	afernan1-2daec3:netbios-ns	***	
UDP	afernan1-2daec3:netbios-dgm	***	
UDP	afernan1-2daec3:1900	***	
UDP	afernan1-2daec3:5353	***	