

8	1.117775	192.168.0.100	74.125.65.100	HTTP	877 GET /site/carojash/ HTTP/1.1
9	1.319190	74.125.65.100	192.168.0.100	TCP	54 80_2818 [ACK] Seq=1 Ack=824 Win=6584 Len=0
10	1.768058	74.125.65.100	192.168.0.100	TCP	442 [TCP segment of a reassembled PDU]

▶ Frame 6: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)  
 ▶ Ethernet II, Src: Belkin\_40:36:38 (00:17:3f:40:36:38), Dst: IntelCor\_4b:09:ca (00:15:00:4b:09:ca)  
 ▶ Internet Protocol Version 4, Src: 74.125.65.100 (74.125.65.100), Dst: 192.168.0.100 (192.168.0.100)

▼ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 2818 (2818), Seq: 0, Ack: 1, Len: 0

Source Port: 80 (80)  
 Destination Port: 2818 (2818)  
 [Stream index: 0]  
 [TCP Segment Len: 0]  
 Sequence number: 0 (relative sequence number)  
 Acknowledgment number: 1 (relative ack number)  
 Header Length: 28 bytes

▶ ... 0000 0001 0010 = Flags: 0x012 (SYN, ACK)

Window size value: 5720  
 [Calculated window size: 5720]  
 ▶ Checksum: 0x94dc [validation disabled]  
 Urgent pointer: 0  
 ▶ Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted  
 ▶ [SEQ/ACK analysis]

4b 09 ca 00 17 3f 40 36 38 08 00 45 00  
 06 9c 1a 4a 7d 41 64 c0 a8  
 54 7e 70 12



**UAEM** | Universidad Autónoma  
 del Estado de México

**Centro Universitario UAEM Zumpango**  
**Ingeniería en Computación**

**Protocolos de red**  
**M.T.I. Carlos Alberto Rojas Hernández**





## Identificación de la Unidad de Aprendizaje (UA)

Nombre UA:

**Protocolos de red (L41041)**

Total de horas a la semana: **5**

Créditos: **9**

Carácter de la UA: Obligatoria

Modalidad: **Presencial**

UA Antecedente: **Transmisión de datos**

UA Consecuente: **Ninguna**





# Protocolos de red

## Propósito de la unidad de aprendizaje:

Comprender de manera teórica y práctica los conceptos del modelo de capas de las redes de telecomunicaciones, los protocolos de red involucrados en las primeras capas de los modelos de referencia y caracterizar las principales tecnologías de las redes de área local





# Introducción

La presente unidad de aprendizaje enfatiza el modelo de referencia OSI y TCP/IP y se concentra en la comprensión y practica de los protocolos de las capas de enlace de datos, red y transporte. Explora las diferentes tecnologías heredadas, actuales y emergentes usadas en las redes de área local (LAN) y redes de área personal (PAN). Las capas de red y transporte se concentran en el estudio de los protocolos más ampliamente usados en Internet.



## Unidad de competencia VI

Comprender protocolos de la capa de transporte TCP/UDP.



## Unidad de competencia VI

6.1 Concepto de puerto.

6.2 Concepto de paquetes TCP.

6.3 Concepto de paquetes UDP.



## 6.1 Concepto de puerto.

La capa de transporte se encarga de manejar los datos y proporcionar la confiabilidad necesaria en el transporte de los mismos.

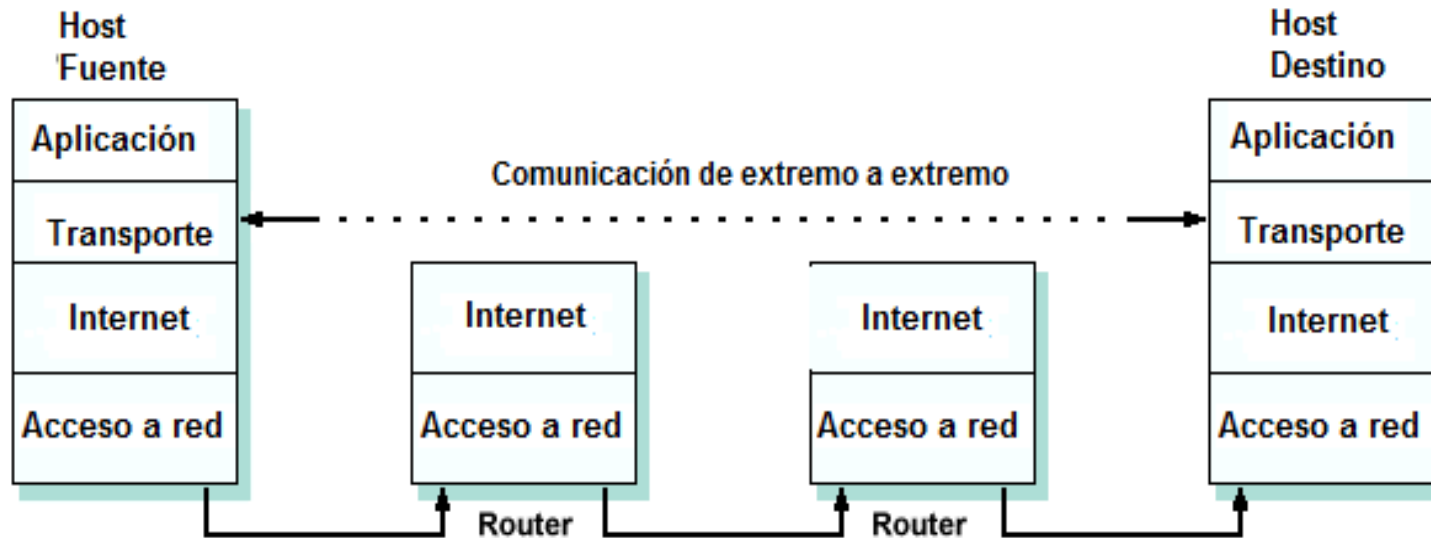
Coincide el nivel de transporte del modelo OSI y del modelo TCP / IP.

Las funciones de la capa de transporte son:



## 6.1 Concepto de puerto.

Se encarga del transporte de los datos extremo a extremo (host a host)

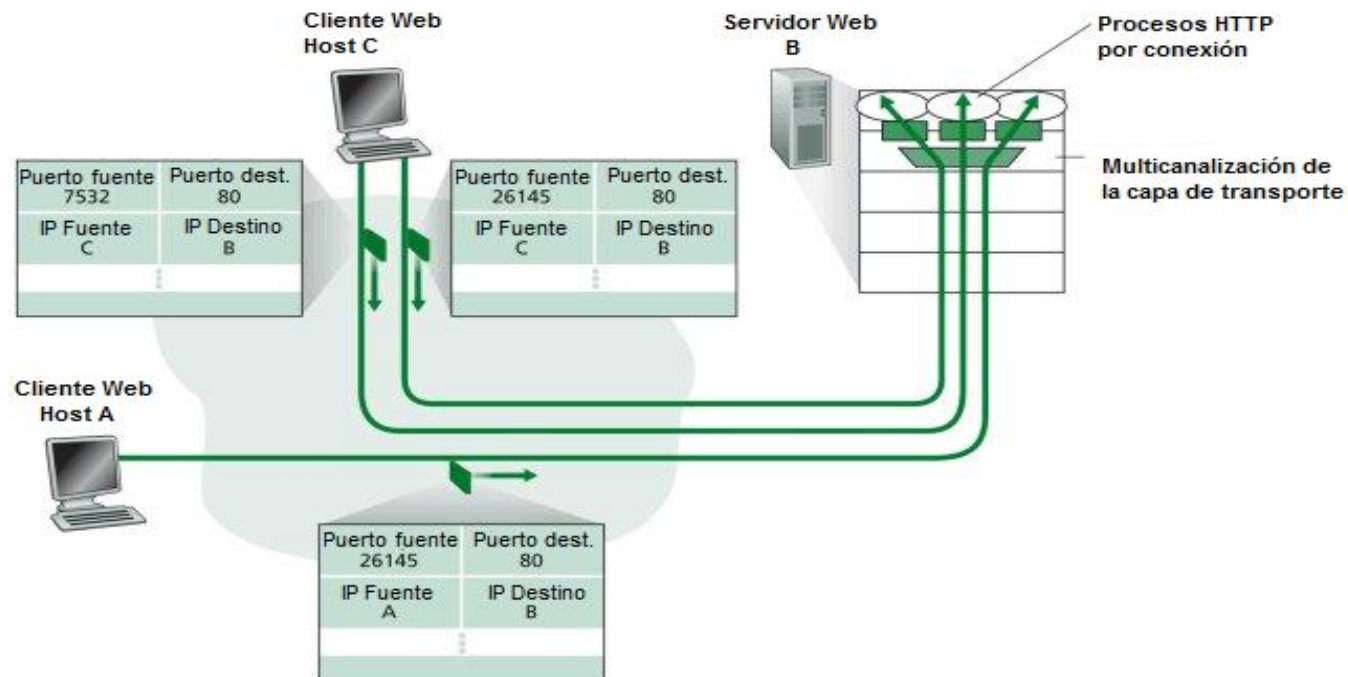






## 6.1 Concepto de puerto.

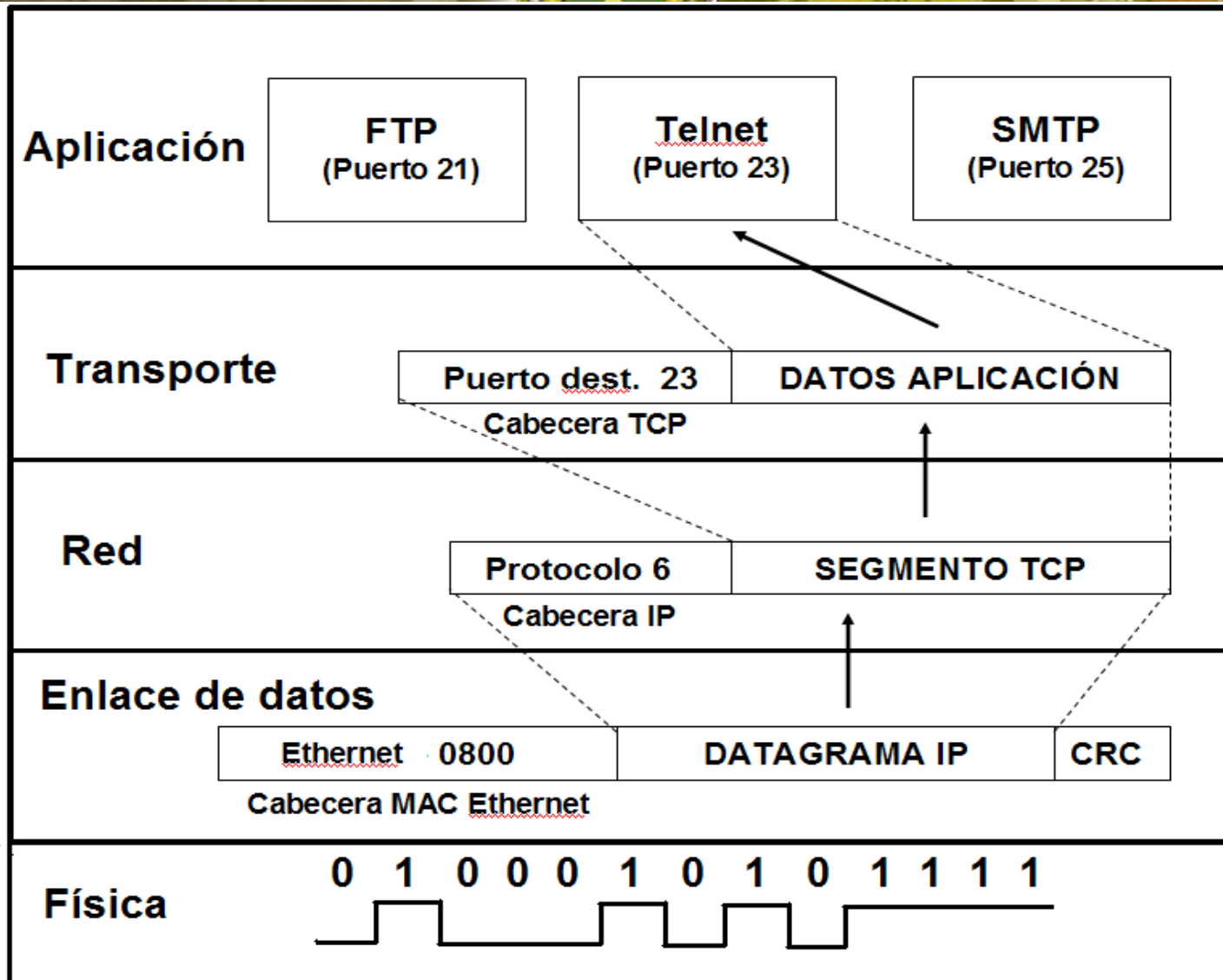
Multicanaliza el tráfico de diversas instancias (procesos) del nivel de aplicación





## 6.1 Concepto de puerto.

La unidad de transferencia de información a nivel de transporte es la TPDU (Transport Protocol Data Unit) o segmento.





## 6.1 Concepto de puerto.

La multicanalización se realiza mediante el puerto (origen o destino) que puede valer de 0 a 65535 (16 bits) y tiene significado local en el host.

Los puertos 0 a 1023 están reservados para servidores “bien conocidos” (“well known ports”)





## 6.1 Concepto de puerto.

La combinación de dirección IP del host y el puerto identifica el “socket”

Una conexión TCP queda especificada por los dos sockets que se comunican





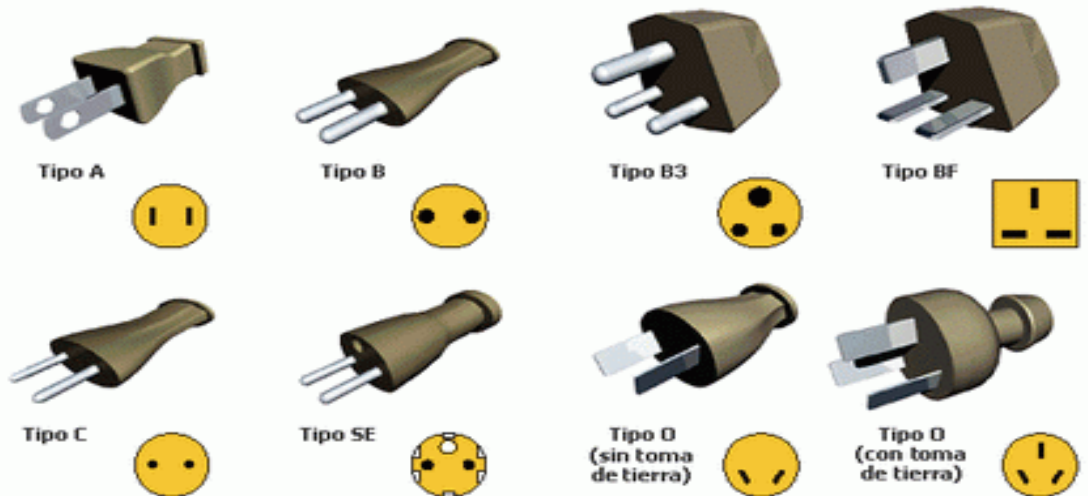
## 6.1 Concepto de puerto.

La capa de transporte utiliza un puerto para identificar aplicaciones específicas o protocolos dentro de cada dispositivo (host), de esta forma cada proceso que usa la red en una computadora utiliza además de la dirección IP y un puerto de protocolo.



## 6.1 Concepto de puerto.

El socket es una representación abstracta de un extremo final en la comunicación entre dos procesos.





## 6.1 Concepto de puerto.

La conexión de red entre 2 programas o procesos comprende lo siguiente:

- Una dirección local u origen, que identifica el dispositivo que enviará los paquetes de datos.
- Un puerto de protocolo local u origen que especifica el proceso que envía los mensajes.

201.45.65.71:1001





## 6.1 Concepto de puerto.

La conexión de red entre 2 programas o procesos comprende lo siguiente:

- Una dirección remota o destino, que identifica el dispositivo o computadora remota.
- Un puerto de protocolo remoto o destino que identifica y recibirá el proceso o programa destino.

121.1.27.44:1001



## 6.1 Concepto de puerto.

La conexión de red entre 2 programas o procesos comprende lo siguiente:

- Un protocolo que especifica cómo los programas transfieren datos a través de la red.

`http:// 23.45.67.104:80`

`ftp:// 199.65.6.100:21`

`34.4.21.224:23`

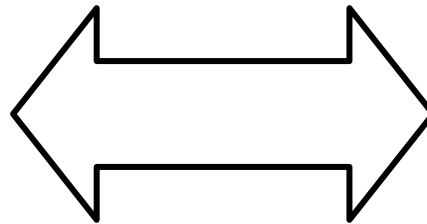


## 6.1 Concepto de puerto.

Para que ocurra la comunicación a través de una interfase de socket, un programa necesita un socket en cada extremo de la conexión de red.



23.45.67.104:80

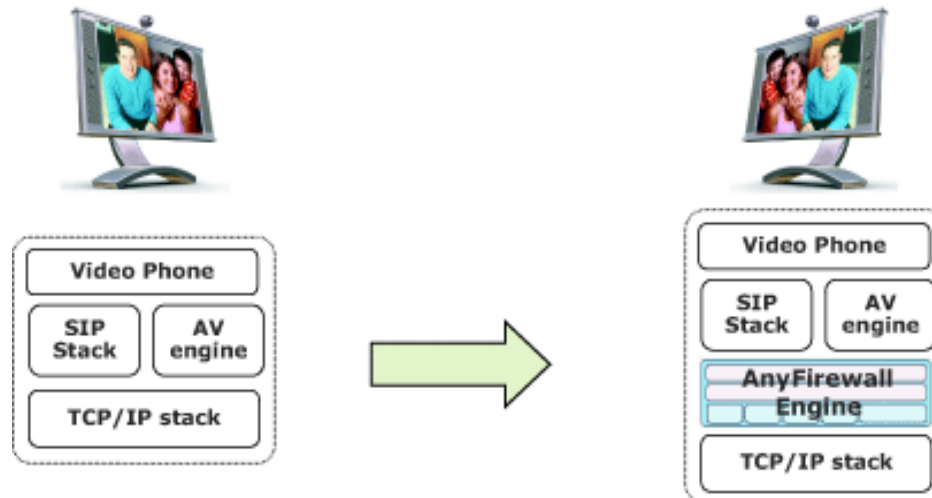


112.4.116.109:80



## 6.1 Concepto de puerto.

La interfase de socket es un grupo de funciones de software o rutinas (API: Application Programming Interface) que permiten a un programador desarrollar aplicaciones para usarse en una red TCP/IP.





## 6.1 Concepto de puerto.

La interfase de socket usa el método de:

**Open  $\Rightarrow$  Read/Write  $\Rightarrow$  Close**

1. Abre (Open) primero una comunicación de red.
2. Leer y escribe (Read/Write) datos a través de la conexión (transferencia de datos).
3. Cierra (Close) la conexión.



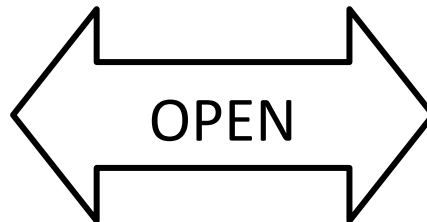
## 6.1 Concepto de puerto.

La interfase de socket usa el método de:

Open  $\Rightarrow$  Read/Write  $\Rightarrow$  Close



23.45.67.104:80



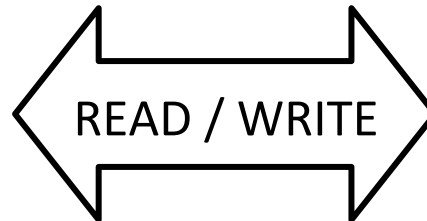
112.4.116.109:80



## 6.1 Concepto de puerto.

La interfase de socket usa el método de:

Open  $\Rightarrow$  Read/Write  $\Rightarrow$  Close



23.45.67.104:80

112.4.116.109:80



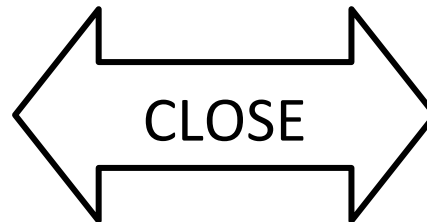
## 6.1 Concepto de puerto.

La interfase de socket usa el método de:

Open  $\Rightarrow$  Read/Write  $\Rightarrow$  Close



23.45.67.104:80



112.4.116.109:80





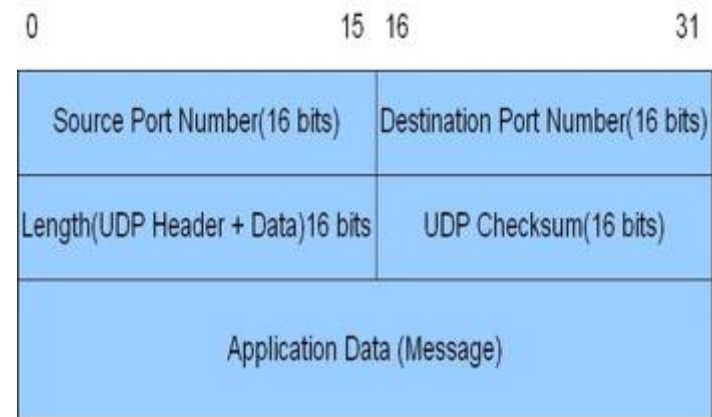
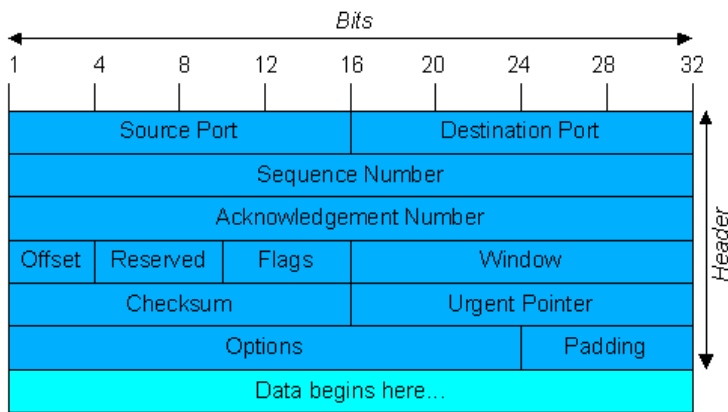
## 6.1 Concepto de puerto

La interfase de socket permite a los programas de usuario utilizar protocolos orientados a conexión y no orientados a conexión, sin embargo el programa de aplicación realiza pasos separados para crear un socket y para conectar el socket a su dispositivo destino.

La creación de un socket significa la asignación de espacio de almacenamiento para la estructura de datos del socket



# 6.1 Concepto de puerto



```

37  /*
38   * Udp protocol header.
39   * Per RFC 768, September, 1981.
40   */
41  struct udphdr {
42      u_int16_t  uh_sport;    /* source port */
43      u_int16_t  uh_dport;    /* destination port */
44      u_int16_t  uh_ulen;    /* udp length */
45      u_int16_t  uh_sum;     /* udp checksum */
46  };
47

```





## 6.1 Concepto de puerto

La capa de transporte en Internet utiliza dos protocolos, uno de ellos está completamente orientado a conexión TCP (Transmission Control Protocol) y el otro no orientado a conexión UDP (User Datagram Protocol).

¿Cuál de los 2 es el más adecuado para enviar un correo electrónico ?

¿Cuál de los 2 es el más adecuado para realizar una transmisión de radio a través de Internet ?



## 6.2 TCP - Transmission Control Protocol (Protocolo de control de transmisión)

Es el protocolo más utilizado en la capa de transporte, ya que por sus características se dice que es el contrapeso del protocolo de Internet (IP), para que la información a través de Internet llegue de la mejor manera posible, algunas de sus características son:



## 6.2 TCP - Transmission Control Protocol

### Características

- El IP mueve los paquetes de datos a granel, mientras TCP se encarga del flujo y asegura que los datos estén correctos.
- Es un protocolo orientado a conexión.
- Las TPDU de TCP se llaman segmentos.
- Establece y termina conexiones



## 6.2 TCP - Transmission Control Protocol

### Características

- Gestionar los buffers y ejercer control de flujo de forma eficiente
- Multicanalización el nivel de aplicación (puerto) e intercambiar datos con las aplicaciones.
- La multicanalización se realiza mediante el puerto (origen o destino) que puede valer de 0 a 65535 (16 bits) y tiene significado local en el host.



## 6.2 TCP - Transmission Control Protocol

### Características

- Una conexión TCP queda especificada por los dos sockets que se comunican
- Controlar errores, retransmitir segmentos perdidos o erróneos y eliminar duplicados
- Efectuar control de congestión



## 6.2 TCP - Transmission Control Protocol

### Características

- Es más confiable que UDP ya que mantiene una conexión entre el receptor y el emisor, asegurando una entrega correcta de datos.
- Proporciona una transmisión confiable de datos mediante detección y corrección de datos extremo a extremo.





## 6.2 TCP - Transmission Control Protocol

### Características

- Retransmite los datos no recibidos por el nodo destino
- Ofrece garantía contra datos duplicados
- Es utilizado por Telnet, FTP, SMTP, y POP.



## 6.2 TCP - Transmission Control Protocol

Como ya se menciona en el UDP, al utilizar datagramas IP es necesario, añadir una cabecera y una pseudocabecera al principio del datagrama, la cual permite a TCP comprobar que IP no se ha equivocado en la entrega del datagrama.

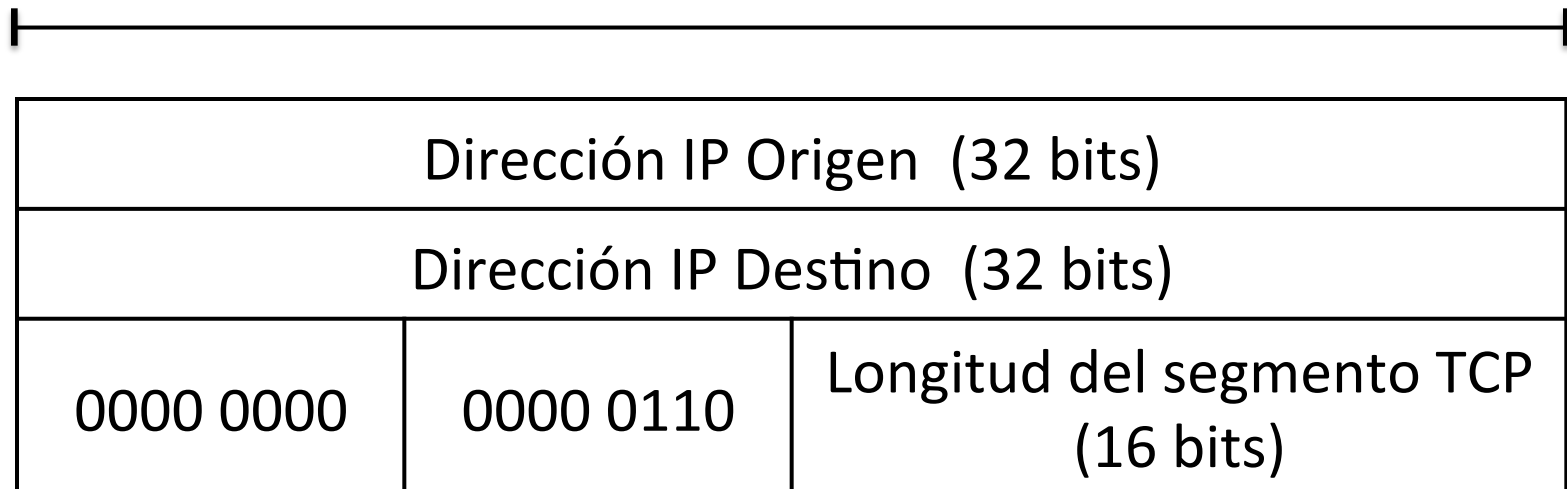


## 6.2 TCP - Transmission Control Protocol

### Encabezado

### Pseudocabecera

**32 bits**





## 6.2 TCP - Transmission Control Protocol

### Encabezado

### Cabecera

Puerto origen (16 bits)				Puerto destino (16 bits)				
Número de secuencia (32 bits)								
Número de confirmación de recepción (32 bits)								
Longitud de cabecera (6 bits)	Reservado (4 bits)	U R G	A C K	P S H	R S T	S Y N	F I N	Tamaño de ventana (16 bits)
Checksum (16 bits)				Puntero de datos urgente (16 bits)				
Opciones / Relleno								



## 6.2 TCP - Transmission Control Protocol

### Conexión y desconexión de TCP.

El protocolo TCP al ser un protocolo orientado a conexión tiene que inicializar la conexión, transferir datos y terminar la conexión, por lo que necesita algunos comandos para comunicarse con el dispositivo al que se desea enviar información, a estos comando se les denominan primitivas.



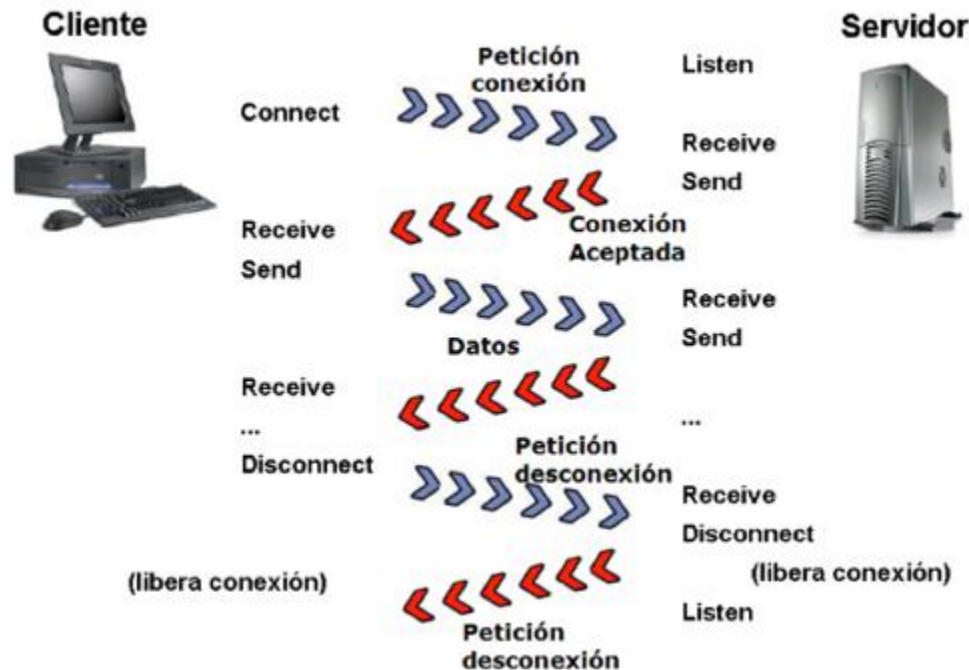
## 6.2 TCP - Transmission Control Protocol

Primitiva	Significado
SOCKET	Crear un nuevo punto terminal de comunicación
LISTEN	Anunciar la disposición para aceptar conexiones
ACCEPT	Conectar y bloquear hasta que otra solicitud llegue
CONNECT	Intenta establecer la conexión
SEND	Enviar datos a través de la conexión
RECEIVE	Recibir datos a través de la conexión
DISCONNECT	Anuncia la desconexión
CLOSE	Cerrar la conexión



## 6.2 TCP - Transmission Control Protocol

Un procedimiento de conexión simple, e incluso se puede decir que empírico es el siguiente:





## 6.2 TCP - Transmission Control Protocol

El esquema anterior no contempla los problemas de pérdida de datagramas o que pueden llegar duplicados (Se pierde la confirmación de un segmento, por lo que el emisor lo reenvía) y podrían causar problemas, como que una sesión entera se duplique. Para evitarlo se utiliza el saludo a tres vías (conexión a tres vías), un procedimiento de conexión más elaborado que evita los problemas debidos a duplicados





## 6.2 TCP - Transmission Control Protocol

En este procedimiento se identifica cada intento de conexión mediante un número diferente. El cliente elige un número a modo de clave para la comunicación en sentido de ida y el servidor otro para el sentido de vuelta. Estos dos números actúan como identificadores de cada intento de conexión y lo protegen de segmentos retrasados que pudieran aparecer fruto de conexiones anteriores.



## 6.2 TCP - Transmission Control Protocol

- El cliente elige para cada intento de conexión un número único. El número elegido lo incluye en la petición de conexión que envía al servidor.



## 6.2 TCP - Transmission Control Protocol

- El servidor, cuando recibe la petición, elige otro número único y envía una respuesta al cliente indicándoselo.

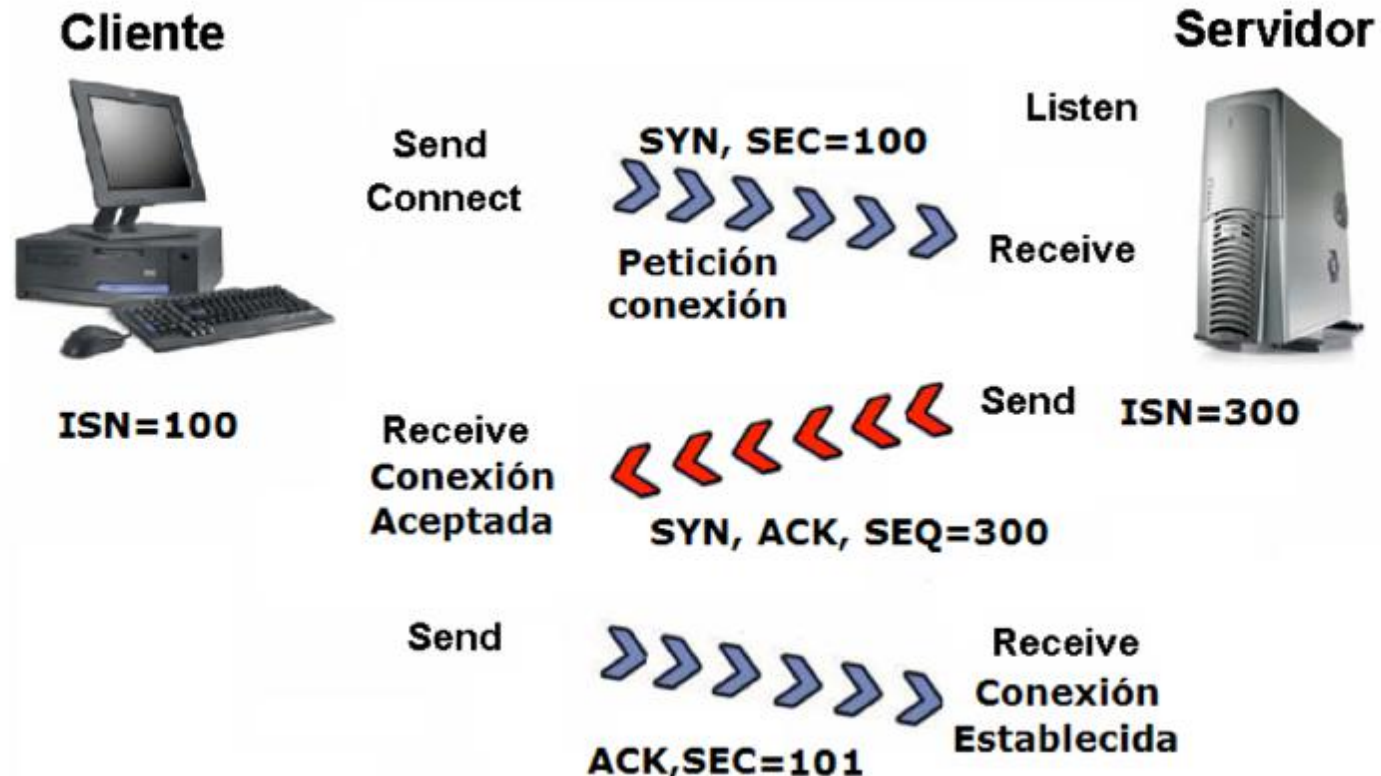


## 6.2 TCP - Transmission Control Protocol

- El cliente al recibir la respuesta considera establecida la conexión. A continuación envía un tercer mensaje en el que confirma recepción del anterior. El servidor considera establecida la conexión cuando él recibe este tercer mensaje.



## 6.2 TCP - Transmission Control Protocol





## 6.3 UDP - User Datagram Protocol ( Protocolo de datagramas de usuario).

- Es un protocolo no confiable y no orientado a conexión para la entrega de mensajes discretos.
- Utiliza datagramas para enviar datos.
- Proporciona el mecanismo primario que utilizan los programas de aplicación para enviar datagramas a otros programas de aplicación.



## 6.3 UDP - User Datagram Protocol

### Características

- No retransmite datos no recibido.
- No hay control de flujo.
- No hay control de congestión.
- No hay conexión ó desconexión



## 6.3 UDP - User Datagram Protocol

### Características

- Es utilizado por: TFTP, SNMP, DNS y BOOTP
- La aplicación es en tiempo real.
- Se usa UDP cuando la entrega rápida es más importante que la entrega garantizada.
- Especificado en RFC 768





## 6.3 UDP - User Datagram Protocol

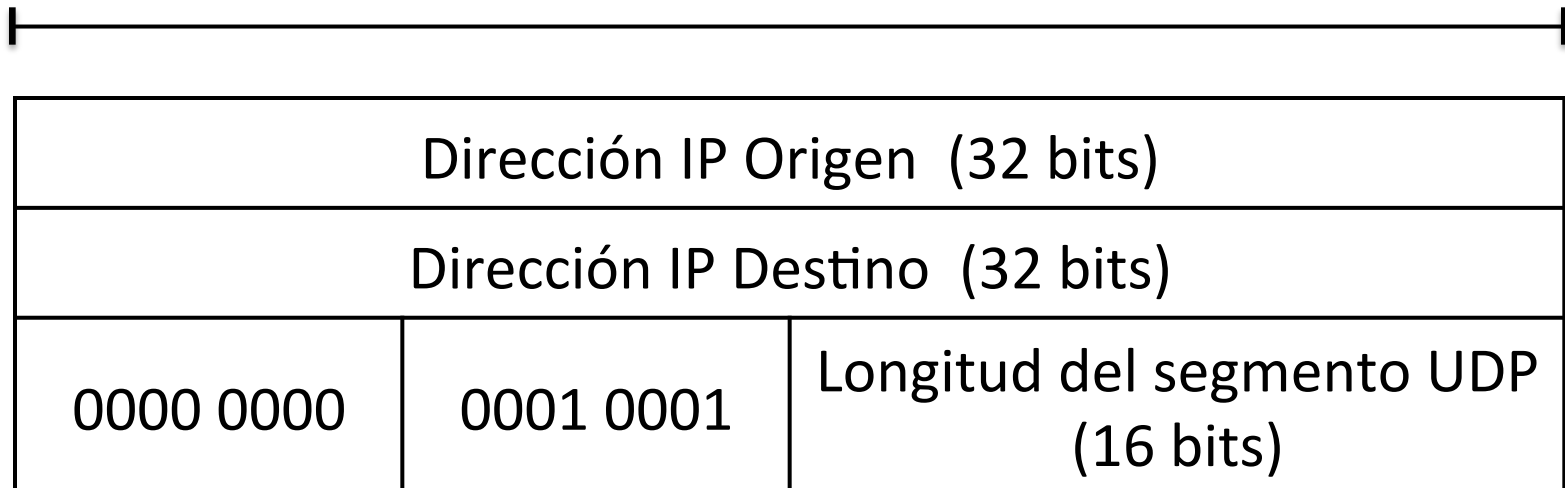
Al utilizar datagramas IP es necesario, añadir **una cabecera y una pseudocabecera** (la cual tiene el mismo formato para UDP como para TCP como veremos enseguida) al principio del datagrama, el formato de ambos es el siguiente:



## 6.3 UDP - User Datagram Protocol

Encabezado  
Pseudocabecera

**32 bits**





## 6.3 UDP - User Datagram Protocol

Encabezado  
Cabecera

**32 bits**

<hr/>	
Puerto Origen (16 bits)	Puerto Destino (16 bits)
Longitud segmento UDP (16 bits)	Checksum (16 bits)



## Referencias bibliográficas

[1] Tanenbaum, Andrew S. , "Redes de computadoras", Quinta Edición, Editorial Pearson Educación de México, 2011, ISBN 9786073208178.

[2] Stallings, William, "Data and computer communications", Novena Edición, Pearson Prentice Hall, 2011, ISBN 9780131392050

[3] Kurose, James F., Ross, Keith W., " Redes de computadoras. Un enfoque descendente" Quinta edición, Pearson Educación, 2010, ISBN 9788478291199



## Referencias bibliográficas

[4] Parkhurst, Bill, “Routing first-step ”, Cisco Press, 2005, ISBN 978-1587201226

[5] Natalia Olifer, Victor Olifer, "Redes de computadoras : principios, tecnologías y protocolos para el diseño de redes" Primera edición, McGraw Hill, 2009, 9789701072493



## Enlaces Web

[A] <http://authors.phptr.com/tanenbaumcn4/> , página del libro Redes de computadoras de Andrew S. Tanenbaum.

[B] <http://labredes.esimez.ipn.mx/> , pagina laboratorio de redes de ESIME Zacatenco, Instituto Politécnico Nacional.

[C] [http://www.econ.uba.ar/www/departamentos/sistemas/plan97/tecn\\_informac/briano/seoane/tp/2002\\_1/redes.h](http://www.econ.uba.ar/www/departamentos/sistemas/plan97/tecn_informac/briano/seoane/tp/2002_1/redes.h) Materia: Tecnología de la Información Dr. José Seoane Departamento de Sistemas Universidad de Buenos Aires, Argentina.

[D] <http://informatica.uv.es/iiguia/AER/Tema4.pdf>, Materia: Aplicaciones específicas de red, Jose J. Samperio, departamento de Informática, Universidad de Valencia, España.





# HUMANISMO QUE TRANSFORMA

---

[www.uaemex.mx](http://www.uaemex.mx)