



Capa de Transporte



Capa de transporte

Capa 4 del modelo OSI: Transporte

Su función básica es aceptar los datos enviados por las capas superiores, dividirlos en pequeñas partes si es necesario, y pasarlos a la capa de red.

Segmenta los datos originados en el host emisor y los reensambla en una corriente de datos dentro del sistema del host receptor.

Suministra un servicio de transporte de datos que aísla las capas superiores de los detalles de implementación del transporte.

Se encarga de multiplexar la comunicación entre diferentes aplicaciones, utiliza puertos lógicos para determinar con qué aplicación se identifican los datos.

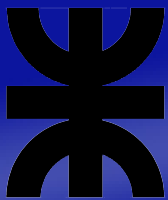


Capa de transporte

Capa de transporte

En el modelo TCP/IP, esta capa se diseñó para que pares de nodos (origen, destino) lleven a cabo una conversación, es similar a la capa de transporte del modelo OSI.

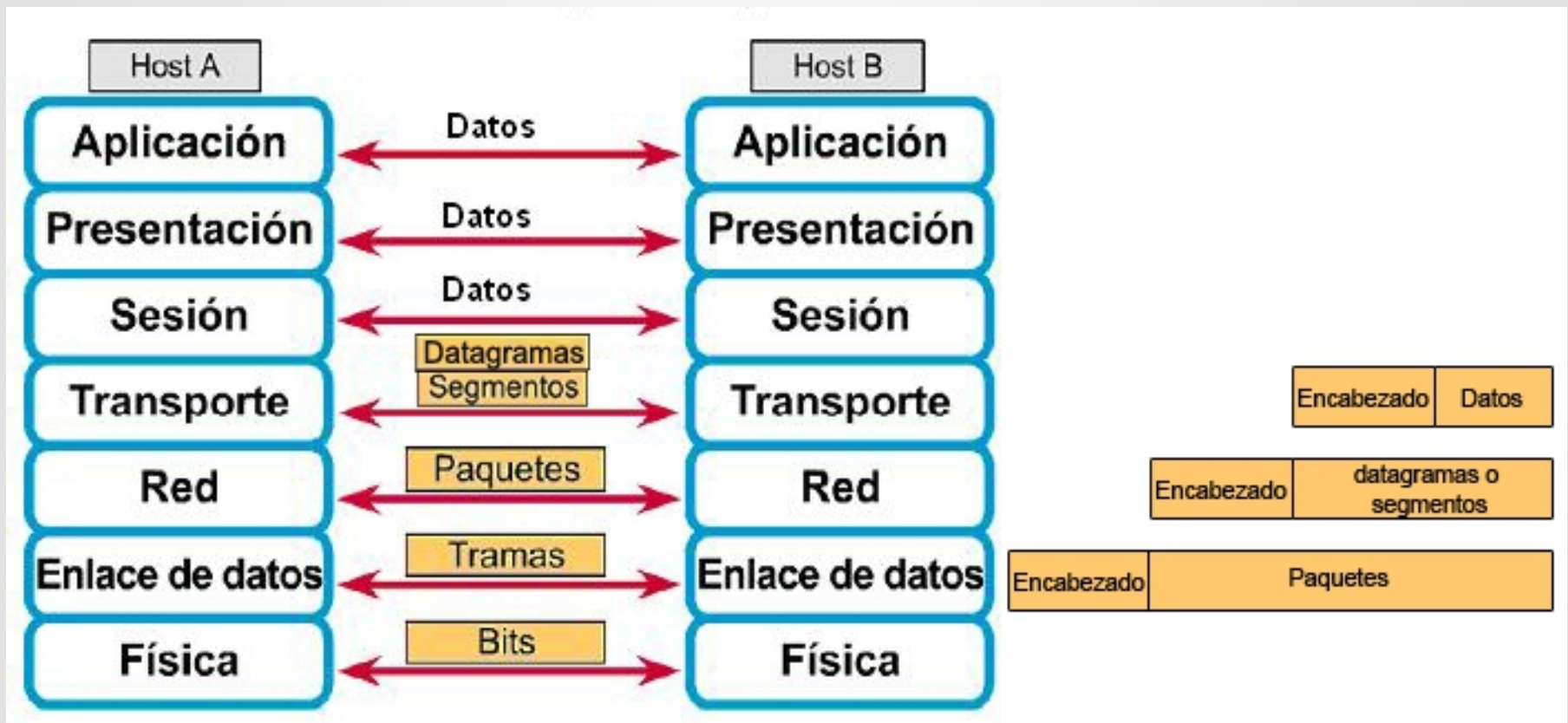
Modelo OSI		Modelo TCP/IP
7	Aplicación	Aplicación
6	Presentación	
5	Sesión	
4	Transporte	Transporte
3	Red	Internet
2	Enlace de Datos	Acceso a la red
1	Física	



Capa de transporte

Capa de transporte

Recibe el flujo de datos desde la aplicación y lo divide en pequeñas piezas denominadas segmentos o datagramas, según el protocolo utilizado.





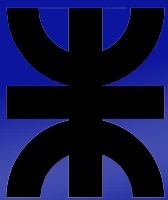
Capa de transporte

Puerto

La capa de transporte utiliza puertos para permitir la comunicación entre aplicaciones.

El campo de puerto tiene una longitud de 16 bits, por lo que el rango de valores válidos va de 0 a 65.535.

- El puerto **0** está reservado, pero es un valor permitido como puerto origen si el proceso emisor no espera recibir mensajes como respuesta.
- Los puertos **1 a 1023** se llaman puertos "bien conocidos" y en sistemas operativos tipo Unix enlazar con uno de estos puertos requiere acceso como superusuario.
- Los puertos **1024 a 49.151** son puertos registrados. Pueden ser usados por cualquier aplicación (procesos o programas que ejecutan los usuarios)
- Los puertos **49.152 a 65.535** son puertos efímeros y son utilizados como puertos temporales, sobre todo por los clientes al comunicarse con los servidores.



Capa de transporte

Ejemplo de puertos bien conocidos

Aplicación	FTP	Telnet	SMTP	DNS	TFTP	HTTP	POP3
Puerto	20/21	23	25	53	69	80	110
Protocolo	TCP / UDP						

FTP: File Transfer Protocol (20: datos, 21: control)

Telnet: acceso a terminal remoto

SMTP: Simple Mail Transfer Protocol

DNS: Domain Name System

TFTP: Trivial File Transfer Protocol

HTTP: Hyper Text Transfer Protocol

POP3: Post Office Protocol



Capa de transporte

Capa de transporte

Los protocolos utilizados en la capa de transporte:

TCP (*TransmissionControl Protocol*): Protocolo de capa de transporte orientado a la conexión (PDU=segmentos)

UDP (*UserDatagramProtocol*): Protocolo de capa de transporte no-orientado a la conexión (PDU=datagramas)

UDP no tiene confirmación de entrega, ni control de flujo, por lo que no son funciones de la capa de transporte.



Capa de transporte

UDP

Es un protocolo del nivel de transporte basado en el intercambio de datagramas.

Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

UDP sólo añade multiplexado de aplicación y suma de verificación de la cabecera y la carga útil. Cualquier tipo de garantías para la transmisión de la información deben ser implementadas en capas superiores.

Proporciona una sencilla interfaz entre la capa de red y la capa de aplicación.

Está documentado en el RFC 768 de la IETF.



Capa de transporte

UDP

Desventajas:

- No posee confirmación de entrega.
- No posee control de flujo.

Ventajas:

- No introduce retardos para establecer una conexión, no mantiene estado de conexión alguno y no realiza seguimiento de estos parámetros. Así, un servidor dedicado a una aplicación particular puede soportar más clientes activos cuando la aplicación corre sobre UDP en lugar de sobre TCP.
- Bajo coste de información, el encabezado de UDP es de solo 8 bytes



Capa de transporte

UDP

Lo utilizan aplicaciones como:

- NFS (Network File System)
- RCP: comando para copiar ficheros remotos
- BOOTP (**B**ootstrap **P**rotocol)
- TFTP:(Trivial file transfer Protocol) Protocolo de transferencia de archivos trivial
- SNMP: (Simple Network Management Protocol) Protocolo Simple de Administración de Red
- DHCP: (Dynamic Host Configuration Protocol) protocolo de configuración dinámica de host
- DNS: (Domain Name System o DNS) sistema de nombres de dominio
- La tecnología VoIP: (Voz sobre Protocolo de Internet) lo utiliza como protocolo de transporte
- Se lo utiliza para la transmisión de audio y video en tiempo real, donde no es posible realizar retransmisiones por los estrictos requisitos de retardo que se tiene en estos casos



Capa de transporte

Encabezado UDP

Puerto origen (2 bytes)	Puerto destino (2 bytes)
Longitud total (2 bytes)	Suma de verificación (2 bytes)
Datos (longitud variable)	

Puerto de origen: es el número de puerto relacionado con la aplicación del remitente del segmento UDP. Este campo representa una dirección de respuesta para el destinatario. Por lo tanto, este campo es opcional. Esto significa que si el puerto de origen no está especificado, los 16 bits de este campo se pondrán en cero. En este caso, el destinatario no podrá responder (lo cual no es estrictamente necesario, en particular para mensajes unidireccionales)

Puerto de destino: este campo contiene el puerto correspondiente a la aplicación del equipo receptor al que se envía.



Capa de transporte

Encabezado UDP

Puerto origen (2 bytes)	Puerto destino (2 bytes)
Longitud total (2 bytes)	Suma de verificación (2 bytes)
Datos (longitud variable)	

Longitud: este campo especifica la longitud total del segmento, con el encabezado incluido (la mínima longitud es 8 bytes, la del encabezado). Sin embargo, el encabezado tiene una longitud de 4 x 16 bits (que es 8 x 8 bits), por lo tanto la longitud del campo es necesariamente superior o igual a 8 bytes.

Suma de verificación: es una suma de comprobación realizada de manera tal que permita controlar la integridad del segmento.



Bibliografía

Tanenbaum, Andrew S. *Redes de computadoras, 5ta Edición*. Prentice Hall. 2011. Capítulo 6.1, 6.4