INVESTIGACIÓN, CAPA DE TRANSPORTE Y CAPA DE APLICACIÓN Y SERVICIOS RED.

FUNDAMENTOS DE REDES

ISAAC VIEYRA SANTOS
MTRO. ANDRES LUNA JAIMES
UNIDAD 2
UNIVERSIDAD POLITÉCNICA DE LÁZARO CÁRDENAS

CAPA DE TRANSPORTE

La Capa de Transporte es la cuarta capa del modelo OSI y es fundamental para la comunicación entre aplicaciones en diferentes hosts. Su objetivo principal es proporcionar una comunicación lógica entre programas de aplicación que se ejecutan en diferentes dispositivos, independientemente de la red subyacente. Piensa en ella como el servicio postal que asegura que una carta (tus datos) no solo llegue a la ciudad correcta (capa de red), sino también a la persona específica (aplicación) dentro de esa ciudad.

FUNCIONES:

- **Multiplexación y Demultiplexación:**
    - **Multiplexación:** Imagina que tienes varias aplicaciones en tu computadora (navegador web, cliente de correo, juego en línea) que necesitan enviar datos a través de la misma conexión de red. La multiplexación es el proceso por el cual la capa de transporte combina los datos de estas múltiples aplicaciones en un solo flujo para enviarlos a través de la red. Es como empacar varios paquetes pequeños de diferentes remitentes en un solo camión grande.
    - **Demultiplexación:** En el lado receptor, la demultiplexación es el proceso inverso. La capa de transporte desempaqueta el flujo de datos entrante y lo dirige a la aplicación correcta basándose en la información del puerto (más sobre esto en breve). Siguiendo la analogía, el camión llega y la capa de transporte sabe a qué persona (aplicación) entregar cada paquete pequeño.
- **Segmentación de Datos:**
    - Los datos de las aplicaciones suelen ser grandes. La capa de transporte toma estos datos y los divide en unidades más pequeñas llamadas segmentos (en TCP) o datagramas (en UDP). Esto es como cortar un libro grande en páginas individuales para que sean más fáciles de manejar y enviar. Cada segmento/datagrama incluye un encabezado con información de control y el número de puerto.
- **Control de Errores (Extremo a Extremo):**
    - Esta función asegura que los datos lleguen sin errores y en el orden correcto al destino. Si un segmento se pierde, se daña o llega desordenado, la capa de transporte puede detectarlo y solicitar su retransmisión. Este control se realiza "extremo a extremo", es decir, desde la aplicación emisora hasta la aplicación receptora, no solo entre nodos intermedios de la red.
- **Control de Flujo (Extremo a Extremo):**
    - El control de flujo evita que un remitente rápido sature a un receptor lento. La capa de transporte negocia la cantidad de datos que el remitente puede enviar antes de esperar una confirmación del receptor. Esto asegura que el receptor tenga suficiente búfer para procesar los datos entrantes sin perderlos. Es como un control de velocidad entre dos personas conversando, donde uno no habla tan rápido que el otro no pueda entender.

PROTOCOLOS DE LA CAPA DE TRANSPORTE:

Los dos protocolos principales de la capa de transporte son TCP y UDP, cada uno con propósitos diferentes.

o TCP (Transmission Control Protocol):

El Protocolo de Control de Transmisión (TCP) es el protocolo más utilizado en la capa de transporte debido a su confiabilidad.

- **Orientado a Conexión:** Esto significa que TCP establece una conexión lógica entre el remitente y el receptor antes de que se envíen los datos y la mantiene durante toda la comunicación. Es como hacer una llamada telefónica: primero estableces la llamada y luego hablas.
- **Establecimiento y Liberación de Conexión (Three-way Handshake):**
    - Para establecer una conexión, TCP usa un proceso de tres pasos conocido como **"three-way handshake"** (apretón de manos de tres vías):
        1. **SYN (Synchronize):** El cliente envía un segmento SYN al servidor para iniciar la conexión.
        2. **SYN-ACK (Synchronize-Acknowledge):** El servidor recibe el SYN, lo reconoce y envía su propio SYN de vuelta al cliente.
        3. **ACK (Acknowledge):** El cliente recibe el SYN-ACK, lo reconoce y la conexión se establece.
    - Para liberar la conexión, se utiliza un proceso similar, generalmente de cuatro pasos (FIN, ACK, FIN, ACK), que asegura que ambas partes estén de acuerdo en cerrar la comunicación.
- **Ventanas Deslizantes, Control de Congestión, Retransmisiones:**
    - **Ventanas Deslizantes:** TCP usa ventanas deslizantes para el control de flujo. Permite que un remitente envíe múltiples segmentos antes de recibir un ACK, lo que mejora la eficiencia. El tamaño de la ventana se ajusta dinámicamente.
    - **Control de Congestión:** TCP detecta y responde a la congestión de la red para evitar que la red se sature. Reduce la velocidad de envío de datos cuando detecta congestión y la aumenta gradualmente cuando la congestión disminuye.
    - **Retransmisiones:** Si un segmento no es reconocido dentro de un tiempo determinado, TCP asume que se perdió y lo retransmite, asegurando la entrega confiable.
- **Puertos TCP:** Los puertos TCP son números que identifican de forma única las aplicaciones o servicios en un host. Permiten que múltiples aplicaciones compartan la misma conexión de red y que los datos se entreguen a la aplicación correcta. Por ejemplo, el puerto 80 es para HTTP (navegación web), el puerto 443 para HTTPS, y el puerto 21 para FTP.

o UDP (User Datagram Protocol):

El Protocolo de Datagramas de Usuario (UDP) es un protocolo más simple y ligero que TCP.

- **No Orientado a Conexión (Sin Estado):** A diferencia de TCP, UDP no establece una conexión antes de enviar datos. Simplemente envía los datagramas al destino sin ninguna garantía de entrega o de orden. Es como enviar una postal: la envías y esperas que llegue, pero no sabes si lo hizo.
- **Funcionamiento:** UDP es un protocolo "best-effort" (mejor esfuerzo). Envía los datos lo más rápido posible, sin preocuparse por la pérdida, la duplicación o el orden de los paquetes. No tiene mecanismos de control de errores, control de flujo o control de congestión.
- **Cuándo utilizar UDP:** UDP es ideal para aplicaciones donde la velocidad y la baja latencia son más importantes que la fiabilidad absoluta. Ejemplos incluyen:
    - **Transmisión de vídeo y audio en tiempo real (streaming):** Si se pierde un fotograma o un paquete de audio, no es crítico retransmitirlo, ya que el siguiente paquete ya está en camino y el usuario apenas lo notará.
    - **Juegos en línea:** La latencia es crucial. Pequeñas pérdidas de paquetes son preferibles a la demora causada por las retransmisiones de TCP.
    - **Consultas DNS (Domain Name System):** Las respuestas DNS suelen ser pequeñas y rápidas.
- **Puertos UDP:** Al igual que TCP, UDP utiliza puertos para dirigir los datagramas a la aplicación correcta en el host. Por ejemplo, el puerto 53 es para DNS y el puerto 68 para DHCP.

CONCEPTOS RELACIONADOS:

- **Socket:**
    - Un socket es un punto final de una comunicación bidireccional en una red. Es una combinación única de una dirección IP y un número de puerto. Un socket permite que un programa envíe y reciba datos a través de una red. Imagina que tu aplicación quiere hablar con otra: tu aplicación crea un socket, que es como un "enchufe" virtual con una dirección específica para que la comunicación pueda ocurrir. Cada conexión TCP o UDP tiene dos sockets, uno en cada extremo.
- **Números de Puerto (Tipos):**
    - Los puertos son números de 16 bits (0 a 65535) que permiten a la capa de transporte distinguir entre diferentes aplicaciones o servicios que se ejecutan en el mismo host.
    - Números de Puerto Conocidos (Well-Known Ports): Son puertos del 0 al 1023. Están reservados para servicios de red comunes y estándar, asignados por la IANA (Internet Assigned Numbers Authority). Ejemplos:
        - **20/21:** FTP (File Transfer Protocol)
        - **23:** Telnet

        - **25:** SMTP (Simple Mail Transfer Protocol)
        - **53:** DNS (Domain Name System)
        - **80:** HTTP (Hypertext Transfer Protocol)
        - **443:** HTTPS (HTTP Secure)
    - Números de Puerto Registrados (Registered Ports): Van del 1024 al 49151. Pueden ser registrados por empresas o aplicaciones para sus propios servicios. Por ejemplo, bases de datos o aplicaciones propietarias a menudo utilizan puertos registrados.
    - Números de Puerto Dinámicos/Privados (Dynamic/Private Ports): Van del 49152 al 65535. Son puertos que se asignan dinámicamente por el sistema operativo cuando un cliente inicia una conexión saliente. Se utilizan para conexiones efímeras y se liberan después de que la comunicación termina.

CAPA DE APLICACIÓN Y SERVICIOS RED

La Capa de Aplicación es la séptima y última capa del modelo OSI, y es la más cercana al usuario. Actúa como la interfaz entre las aplicaciones de software que usamos día a día y la red subyacente. Su función principal es proporcionar los servicios de red que las aplicaciones necesitan para comunicarse y operar. Piensa en ella como el idioma que tus programas usan para hablar con el internet, traduciendo tus solicitudes (como abrir una página web o enviar un correo) en algo que la red pueda entender y viceversa.

FUNCIONES DE LA CAPA DE APLICACION:

- **Interfaz con las aplicaciones de usuario:** La Capa de Aplicación es donde los usuarios interactúan directamente con la red. Facilita que programas como navegadores web, clientes de correo electrónico, aplicaciones de mensajería instantánea y juegos en línea puedan acceder a los recursos y servicios de la red. Es el punto donde tus clics, escritos y comandos se convierten en datos que viajan por la red, y donde los datos que llegan de la red se presentan de forma comprensible para ti.
- **Provisión de servicios de red:** Esta capa ofrece un conjunto de funciones y protocolos que permiten a las aplicaciones realizar tareas específicas en la red. Estos servicios incluyen la transferencia de archivos, la navegación web, el envío y recepción de correo electrónico, la resolución de nombres de dominio y muchas otras operaciones que requieren comunicación a través de una red. En esencia, la capa de aplicación define cómo las aplicaciones formatean, transmiten y reciben los datos para lograr sus objetivos.

PROTOCOLOS DE LA CAPA DE APLICACIÓN:

- **HTTP/HTTPS (Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure): Navegación web.**
    - **HTTP** es el protocolo fundamental para la World Wide Web. Permite la transferencia de documentos HTML, imágenes, videos y otros recursos a través de internet. Es un protocolo sin estado, lo que significa que cada solicitud del cliente al servidor es independiente de las anteriores.
    - **HTTPS** es la versión segura de HTTP. Utiliza SSL/TLS (Secure Sockets Layer/Transport Layer Security) para cifrar la comunicación entre el navegador y el servidor web, protegiendo la información sensible como contraseñas y datos bancarios. Es esencial para transacciones seguras en línea.
- **DNS (Domain Name System): Resolución de nombres de dominio.**
    - El **DNS** es como la "guía telefónica" de internet. Traduce los nombres de dominio legibles para humanos (ej., `google.com`) en direcciones IP numéricas (ej., `172.217.160.142`) que las computadoras usan para identificar y localizar servidores en la red. Sin DNS, tendrías que recordar la dirección IP de cada sitio web que quisieras visitar.
- **FTP/SFTP/TFTP (File Transfer Protocol): Transferencia de archivos.**

    - **FTP** es un protocolo estándar para la transferencia de archivos entre un cliente y un servidor en una red. Permite subir y descargar archivos, así como gestionar directorios. Es un protocolo antiguo que utiliza conexiones separadas para control y datos.
    - **SFTP (SSH File Transfer Protocol)** es una extensión del protocolo SSH que proporciona una transferencia de archivos segura. A diferencia de FTP, SFTP cifra tanto los datos como los comandos, ofreciendo mayor seguridad.
    - **TFTP (Trivial File Transfer Protocol)** es una versión más simple y ligera de FTP. Se utiliza para transferencias de archivos muy básicas, a menudo en el arranque de dispositivos de red (como routers) o para actualizaciones de firmware, ya que no ofrece seguridad ni autenticación.
- **SMTP/POP3/IMAP (Simple Mail Transfer Protocol): Correo electrónico.**
    - **SMTP (Simple Mail Transfer Protocol)** es el protocolo principal utilizado para **enviar correos electrónicos** desde un cliente de correo a un servidor de correo, y entre servidores de correo. Es el encargado de la "entrega" del correo.
    - **POP3 (Post Office Protocol version 3)** es un protocolo que permite a los clientes de correo descargar correos electrónicos del servidor a su dispositivo local. Por defecto, los correos se eliminan del servidor una vez descargados.
    - **IMAP (Internet Message Access Protocol)** es otro protocolo para acceder a correos electrónicos en un servidor. A diferencia de POP3, IMAP permite a los usuarios gestionar sus correos directamente en el servidor, sincronizando el estado (leído, no leído, borrador) en múltiples dispositivos. Los correos permanecen en el servidor hasta que el usuario los elimina explícitamente.
- **SSH (Secure Shell): Acceso remoto seguro.**
    - **SSH** es un protocolo de red criptográfico que permite a los usuarios acceder de forma segura a computadoras remotas a través de una red no segura. Proporciona una interfaz de línea de comandos cifrada, así como la capacidad de tunelizar otros protocolos, lo que lo hace muy utilizado por administradores de sistemas para gestionar servidores de forma remota.
- **Telnet (breve mención de su inseguridad):**
    - **Telnet** es un protocolo de red más antiguo que SSH, diseñado para el acceso remoto a la línea de comandos de un dispositivo. Sin embargo, su principal desventaja y la razón por la que ha sido ampliamente reemplazado por SSH es su inseguridad: Telnet transmite los datos, incluidas las contraseñas, en texto plano (sin cifrar). Esto lo hace vulnerable a la interceptación y el robo de credenciales. No se recomienda su uso en redes no seguras.
- **SNMP (Simple Network Management Protocol): Gestión de redes.**
    - **SNMP** es un protocolo estándar para la gestión y monitoreo de dispositivos de red (routers, switches, servidores, impresoras, etc.). Permite a los administradores de red recopilar información sobre el estado de los dispositivos, detectar fallos y configurar parámetros de red de forma remota.

- **DHCP (Dynamic Host Configuration Protocol):**
    - Aunque el DHCP se mencionó en la Capa de Red (porque asigna direcciones IP, que son parte de esa capa), sus operaciones y la interacción con los

clientes se consideran a menudo servicios de la Capa de Aplicación. El DHCP permite a los dispositivos en una red obtener automáticamente configuraciones de red, como direcciones IP, máscaras de subred, puertas de enlace predeterminadas y servidores DNS, al conectarse a la red. Esto simplifica enormemente la administración de la red, eliminando la necesidad de configurar manualmente cada dispositivo.