
INSTITUTO TECNOLÓGICO DE SALINA CRUZ

**CATEDRÁTICO:**

ROMAN NAJERA SUSANA MONICA

**ACTIVIDAD:**

REPORTE DEL CAPITULO 4

**MATERIA:**

FUNDAMENTOS DE REDES

**ALUMNO:**

ZARATE LÓPEZ LEONARDO

**CARRERA:**

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

**SEMESTRE:** 5 **GRUPO:** E

SALINA CRUZ OAXACA, A 28 DE OCTUBRE DEL 2014.

**CAPITULO 4 Capa de Transporte del modelo OSI**

**Funciones de la capa de Transporte**

La capa de Transporte permite la segmentación de datos y brinda el control necesario para reensamblar las partes dentro de los distintos streams de comunicación. Las responsabilidades principales que debe cumplir son:

* seguimiento de la comunicación individual entre aplicaciones en los hosts origen y destino,
* segmentación de datos y gestión de cada porción,
* reensamble de segmentos en flujos de datos de aplicación, e
* identificación de las diferentes aplicaciones.





La segmentación de los datos, que cumple con los protocolos de la capa de Transporte, proporciona los medios para enviar y recibir datos cuando se ejecutan varias aplicaciones de manera concurrente en una computadora. Sin segmentación, sólo una aplicación, la corriente de vídeo por ejemplo, podría recibir datos. No se podrían recibir correos electrónicos, chats ni mensajes instantáneos ni visualizar páginas Web y ver un vídeo al mismo tiempo.

En la capa de Transporte, cada conjunto de secciones en particular que fluyen desde una aplicación de origen a una de destino se conoce como conversación.

Para identificar todos los segmentos de datos, la capa de Transporte agrega un encabezado a la sección que contiene datos binarios. Este encabezado contiene campos de bits. Son los valores de estos campos los que permiten que los distintos protocolos de la capa de Transporte lleven a cabo las diversas funciones.

Las funciones principales especificadas por todos los protocolos de la capa de Transporte incluyen:

Segmentación y reensamblaje: La mayoría de las redes poseen una limitación en cuanto a la cantidad de datos que pueden incluirse en una única PDU (Unidad de datos del protocolo). La capa de Transporte divide los datos de aplicación en bloques de datos de un tamaño adecuado. En el destino, la capa de Transporte reensambla los datos antes de enviarlos a la aplicación o servicio de destino.

Multiplexación de conversaciones: Pueden existir varias aplicaciones o servicios ejecutándose en cada host de la red. A cada una de estas aplicaciones o servicios se les asigna una dirección conocida como puerto para que la capa de Transporte pueda determinar con qué aplicación o servicio se identifican los datos.

Además de utilizar la información contenida en los encabezados para las funciones básicas de segmentación y

Reensamblaje de datos, algunos protocolos de la capa de Transporte proveen:

* Conversaciones orientadas a la conexión,
* Entrega confiable,
* Reconstrucción ordenada de datos, y
* Control del flujo.

Cabe recordar que la función principal de la capa de Transporte es administrar los datos de aplicación para las conversaciones entre hosts. Sin embargo, las diferentes aplicaciones tienen diferentes requerimientos para sus datos y, por lo tanto, se han desarrollado diferentes protocolos de Transporte para satisfacer estos requerimientos.

Un protocolo de la capa de Transporte puede implementar un método para asegurar la entrega confiable de los datos. En términos de redes, confiabilidad significa asegurar que cada sección de datos que envía el origen llegue al destino. En la capa de Transporte, las tres operaciones básicas de confiabilidad son:

* seguimiento de datos transmitidos,
* acuse de recibo de los datos recibidos, y
* retransmisión de cualquier dato sin acuse de recibo.

Esto requiere que los procesos de la capa de Transporte de origen mantengan el seguimiento de todas las porciones de datos de cada conversación y retransmitan cualquiera de los datos que no dieron acuse de recibo por el destino. La capa de Transporte del host de recepción también debe rastrear los datos a medida que se reciben y reconocer la recepción de los datos.

**TCP y UDP**

Los dos protocolos más comunes de la capa de Transporte del conjunto de protocolos TCP/IP son el Protocolo de control de transmisión (TCP) y el Protocolos de datagramas de usuario (UDP). Ambos protocolos gestionan la comunicación de múltiples aplicaciones. Las diferencias entre ellos son las funciones específicas que cada uno implementa.

**Protocolo de datagramas de usuario (UDP)**

UDP es un protocolo simple, sin conexión, descrito en la RFC 768. Cuenta con la ventaja de proveer la entrega de datos sin utilizar muchos recursos. Las porciones de comunicación en UDP se llaman datagramas. Este protocolo de la capa de Transporte envía estos datagramas como "mejor intento". Entre las aplicaciones que utilizan UDP se incluyen:

* sistema de nombres de dominios (DNS),
* streaming de vídeo, y
* Voz sobre IP (VoIP).

**Protocolo de control de transmisión (TCP)**

TCP es un protocolo orientado a la conexión, descrito en la RFC 793. TCP incurre en el uso adicional de recursos para agregar funciones. Las funciones adicionales especificadas por TCP están en el mismo orden de entrega, son de entrega confiable y de control de flujo. Cada segmento de TCP posee 20 bytes de carga en el encabezado, que encapsulan los datos de la capa de Aplicación, mientras que cada segmento UDP sólo posee 8 bytes de carga.

Las aplicaciones que utilizan TCP son:

* exploradores Web,
* e-mail, y
* transferencia de archivos

Existen distintos tipos de números de puerto:

**Puertos bien conocidos** (Números del 0 al 1 023): estos números se reservan para servicios y aplicaciones. Por lo general, se utilizan para aplicaciones como HTTP (servidor Web), POP3/SMTP (servidor de e-mail) y Telnet. Al definir estos puertos conocidos para las aplicaciones del servidor, las aplicaciones del cliente pueden ser programadas para solicitar una conexión a un puerto específico y su servicio asociado.

**Puertos Registrados** (Números 1024 al 49151): estos números de puertos están asignados a procesos o aplicaciones del usuario. Estos procesos son principalmente aplicaciones individuales que el usuario elige instalar en lugar de aplicaciones comunes que recibiría un puerto bien conocido.

**Puertos dinámicos o privados** (Números del 49 152 al 65 535): también conocidos como puertos efímeros, suelen asignarse de manera dinámica a aplicaciones de cliente cuando se inicia una conexión.

**Utilización de los dos protocolos TCP y UDP**

Algunas aplicaciones pueden utilizar los dos protocolos: TCP y UDP. Por ejemplo, el bajo gasto de UDP permite que DNS atienda rápidamente varias solicitudes de clientes. Sin embargo, a veces el envío de la información solicitada puede requerir la confiabilidad de TCP. En este caso, el número 53 de puerto conocido es utilizado por ambos protocolos con este servicio.

**La diferencia clave entre TCP y UDP es la confiabilidad**

La confiabilidad de la comunicación TCP se lleva a cabo utilizando sesiones orientadas a la conexión. Antes de que un host que utiliza TCP envíe datos a otro host, la capa de Transporte inicia un proceso para crear una conexión con el destino. Esta conexión permite el rastreo de una sesión o stream de comunicación entre los hosts. Este proceso asegura que cada host tenga conocimiento de la comunicación y se prepare.

Para entender el proceso de enlace de tres vías, es importante observar los distintos valores que intercambian los dos hosts. Dentro del encabezado del segmento TCP, existen seis campos de 1 bit que contienen información de control utilizada para gestionar los procesos de TCP. Estos campos son los siguientes:

* **URG**: Urgente campo de señalizador significativo,
* **ACK**: Campo significativo de acuse de recibo,
* **PSH**: Función de empuje,
* **RST**: Reconfiguración de la conexión,
* **SYN**: Sincronizar números de secuencia,
* **FIN**: No hay más datos desde el emisor.

A estos campos se los denomina señaladores porque el valor de uno de estos campos es sólo de 1 bit, entonces tiene sólo dos valores: 1 ó 0. Si el valor del bit se establece en 1, indica la información de control que contiene el segmento.

**Terminación de la sesión TCP**

Para cerrar la conexión se debe establecer el señalizador de control FIN (Finalizar) en el encabezado del segmento.

Para finalizar todas las sesiones TCP de una vía, se utiliza un enlace de dos vías, que consta de un segmento FIN y un segmento ACK.