ACM(S5)19

UNIVERSIDAD ABIERTA DEL ESTADO DE KRISHNA KANTA HANDIQUI
Complejo Housefed, Dispur, Guwahati - 781 006

Maestría en Aplicaciones Informáticas

COMUNICACIÓN DE DATOS Y REDES INFORMÁTICAS

CONTENIDO

Experto en la materia

Prof. Anjana Kakati Mahanta, Depto. de Ciencias de la Computación, Universidad de Gauhati Prof. Jatindra Kr. Deka, Depto. de Informática e Ingeniería, Instituto Indio de Tecnología, Guwahati

Prof. Diganta Goswami, Departamento. de Informática e Ingeniería, Instituto Indio de Tecnología, Guwahati

Coordinador de curso

Tapashi Kashyap Das, Profesor Asistente, Informática, KKHSOU Arabinda Saikia, Profesor Asistente, Informática, KKHSOU

Equipo de preparación de SLM

Unidades	Contribuyente
1	**Tapashi Kashyap Das,** KKHSOU
2	**Chakradhara Das,** Profesor (Grado de selección), Dpto. de Ingeniería Eléctrica, Politécnico de Bongaigaon, Bongaigaon, Assam
3 y 5	**Swapnanil Gogoi** Asistente Profesor, Instituto de Aprendizaje Abierto y a Distancia (IDOL), Universidad de Gauhati
6	**Bornali Gogoi**, Asistente Profesor, Depto. de Aplicaciones Informáticas (MCA), Facultad de Ingeniería de Assam, Jalukbari, Guwahati, Assam **Pritam**
4 y 7	**Medhi,** Investigador académico, Universidad de Gauhati

julio 2013

Impreso y publicado por Registrar en nombre de la Universidad Abierta Estatal Krishna Kanta Handiqui.

La universidad reconoce con agradecimiento el apoyo económico brindado por la**Consejo de Educación a Distancia, Nueva Delhi**, para la preparación de este material de estudio.

CURSO INTRODUCTORIO

Este es un curso de***Comunicación de datos y redes informáticas***.Este curso tiene como objetivo presentar al alumno los principios, el diseño, la implementación y el rendimiento de las redes informáticas y de comunicación de datos. Las redes se ocupan de la tecnología y la arquitectura de las redes de comunicación utilizadas para interconectar dispositivos de comunicación.

Este curso contiene siete unidades esenciales. La primera unidad es una unidad de introducción a las redes informáticas. Esta unidad incluye diferentes categorías de redes informáticas como LAN, MAN, WAN. El concepto de topología de red junto con sus tipos también se cubre en esta unidad. La segunda unidad está en los modelos de red. En esta unidad se describen los dos modelos ISO-OSI y TCP/IP más importantes. La tercera unidad describe la capa física, que es la capa más baja del modelo OSI. Describe diferentes medios de transmisión de hardware de red, así como métodos de transmisión de redes informáticas. La cuarta unidad está en la capa de enlace de datos. La quinta unidad se centra en la capa de red. La sexta unidad está en la capa de Transporte. La séptima unidad es la última unidad y analiza la capa de aplicación del modelo ISO-OSI.

Mientras revisa una unidad, notará algunos recuadros al costado, que se han incluido para ayudarlo a conocer algunos de los términos difíciles que no se ven. Se han incluido algunas “ACTIVIDADES” para ayudarlo a aplicar sus propios pensamientos. Nuevamente, hemos incluido algunos conceptos relevantes en “HÁGASE SABER” junto con el texto. Y, al final de cada sección, obtendrá preguntas de "COMPROBAR SU PROGRESO". Estos han sido diseñados para autoverificar su progreso de estudio. Será mejor si resuelve los problemas dados en estos recuadros inmediatamente, después de terminar de leer la sección en la que aparecen estas preguntas y luego relaciona sus respuestas con las "RESPUESTAS PARA VERIFICAR SU PROGRESO" que se encuentran al final de cada unidad.

MÁSTER EN APLICACIONES INFORMÁTICAS

Comunicación de datos y redes informáticas

PLAN DE ESTUDIOS DETALLADOS

Unidad 1: Conceptos básicos de las redes informáticas(Marcas: 15)

Red informática: definición, objetivos, estructura; Redes de Difusión y Punto a Punto; Topología de red y sus diversos tipos; Tipos de Red: LAN, MAN, WAN; LAN basadas en servidor y LAN punto a punto; Tipos de Comunicaciones: Síncronas, Asíncronas; Modos de Comunicación: Simplex, Half Duplex, Full Duplex; Protocolos y Normas.

Unidad 2: Modelos de red(Marcas: 15)

Cuestiones de diseño de la capa, jerarquía de protocolos, modelo de referencia ISO-OSI: funciones de cada capa, diversa terminología utilizada en redes informáticas, servicios orientados a la conexión y sin conexión, modelo de referencia de Internet (TCP/IP), comparación de ISO-OSI y TCP /Modelo IP

Unidad 3: Capa Física(Marcas: 15)

Señales: señales analógicas y digitales, límites de velocidad de datos, deterioro de la transmisión, mediciones de señales como rendimiento, velocidad y tiempo de propagación, longitud de onda; Transmisión digital: Codificación de líneas, codificación de bloques, muestreo, modo de transmisión; Transmisión analógica: modulación de datos digitales, módem telefónico, modulación de señales analógicas; Multiplexación: FDM, WDM, TDM; Medios de transmisión: Medios guiados, Medios no guiados, Conmutación de circuitos y Red telefónica: Conmutación de circuitos, red telefónica;

Unidad 4: Capa de enlace de datos(Marcas: 15)

Detección y Corrección de Errores: Tipo de errores, detección y corrección de errores; Protocolo y control de enlace de datos: Control de flujo y error, ARQ de parada y espera, ARQ de retroceso N, ARQ de repetición seleccionada, HDLC; Acceso punto a punto: protocolo punto a punto, pila PPP; Red de área local: Ethernet tradicional, Ethernet rápida y Gigabit; Conexión de LAN, Redes troncales y LAN virtuales: Conexión de dispositivos, Redes troncales, LAN virtuales;

Unidad 5: Capa de red(Marcas: 15)

Internetworks, direccionamiento, enrutamiento, protocolos de capa de red: ARP, IP, ICMP, IPV6, enrutamiento de unidifusión, protocolos de enrutamiento de unidifusión, enrutamiento múltiple, protocolos de enrutamiento de multidifusión;

Unidad 6: Capa de Transporte(Marcas: 15)

Entrega de proceso a proceso, diagrama de datos de usuario, protocolo de control de transmisión

Unidad 7: Capa de aplicación(Marcas: 10)

Modelo Cliente-Servidor: Modelo Cliente-Servidor, interfaz Socket; Una breve introducción a DNS, SMTP, FTP

UNIDAD-1: FUNDAMENTOS DE LA RED DE COMPUTADORAS

ESTRUCTURA DE LA UNIDAD

1.1 OBJETIVOS DE APRENDIZAJE

Después de pasar por esta unidad, podrá:

- definir la red informática con sus objetivos
- aprender sobre diferentes técnicas de conmutación
- describir los servicios orientados a la conexión y sin conexión
- aprender sobre las redes de transmisión y punto a punto
- Describir los tipos de redes informáticas.
- Describir y diferenciar las LAN de servidor y de igual a igual.
- aprender sobre topología de red
- describir los tipos de comunicaciones sincrónicas y asincrónicas
- aprender sobre los diferentes modos de comunicación
- conocer protocolos y normas.

1.2 INTRODUCCIÓN

Comenzando nuestra discusión con la definición de red de computadoras y sus objetivos, presentamos varios conceptos asociados con la comunicación de datos. Además, analizamos brevemente varias categorías de red junto con el concepto de topología de red. Hacia el final, repasamos brevemente diferentes modos de comunicación seguidos de protocolos y estándares.

1.3 RED INFORMÁTICA

Una red de computadoras es un grupo de computadoras que comparten información a través de tecnología inalámbrica o cableada. Una red informática es una colección de varias computadoras y terminales interconectados por una o más rutas de transmisión. El objetivo principal de un sistema de transmisión es la transferencia e intercambio de datos entre las computadoras y las terminales.

Los términos DTE y DCE son muy comunes en la comunicación de datos. DTE significa ***equipos terminales de datos*** y DCE significa ***equipo de comunicación de datos***. DTE se utiliza para describir la máquina del usuario final, que suele ser una computadora o terminal. La función de la comunicación.

red es interconectar los DTE para que puedan compartir recursos, intercambiar datos, brindar respaldo entre sí y permitir que los usuarios realicen su trabajo desde cualquier ubicación. El DCE se utiliza para conectar los DTE a la línea o canal de comunicaciones. También contiene una parte de un proceso de aplicación, pero la función principal del DCE sigue siendo proporcionar una interfaz al DTE con la red de comunicación.

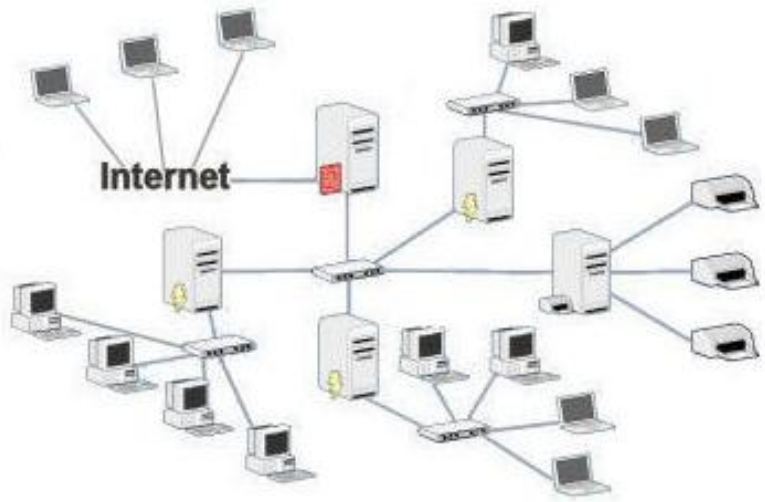


Fig. 1.1: Una red informática típica

Antes de discutir los problemas técnicos en detalle, vale la pena dedicar algún tiempo a señalar por qué la gente está interesada en las redes informáticas y para qué se pueden utilizar.

1.4 OBJETIVOS DE LAS REDES DE COMPUTADORAS

Antes de la llegada de las redes informáticas, los papeles y disquetes eran los únicos medios para compartir información. Hoy en día, las redes informáticas que comprenden una serie de computadoras, así como otros dispositivos como impresoras, escáneres, etc., se utilizan ampliamente para compartir recursos de una manera más eficiente y rápida.

En una red, ya sea grande o pequeña, todos los dispositivos están interconectados para transmitir y recibir datos entre sí. Estas conexiones generalmente se realizan no solo con cables de cobre; en cambio, la fibra óptica, las microondas, el infrarrojo y los satélites de comunicación también se utilizan para una comunicación eficiente. Los principales objetivos de estas redes se pueden resumir de la siguiente manera:

- **El intercambio de recursos**

  Usando redes informáticas, es posible compartir programas, datos y otros recursos entre varios usuarios en la red. El intercambio de recursos es independiente de la ubicación física del usuario y del recurso. Los archivos en la computadora de un usuario en particular se pueden compartir en la red o los archivos se pueden colocar en un servidor de archivos, que proporciona una ubicación central para todos los archivos que necesitan los usuarios en la red. Los usuarios también pueden compartir dispositivos como impresoras, unidades de CD-ROM y discos duros, etc. También facilita la actualización de una aplicación porque la actualización solo debe realizarse en el propio servidor. Por lo tanto, no necesitamos recursos separados para cada computadora.

- **Alta confiabilidad**

  Un segundo objetivo es proporcionar alta confiabilidad al tener fuentes alternativas de suministro. Por ejemplo, todos los archivos se pueden replicar en dos o tres máquinas, por lo que si una de ellas no está disponible, las otras copias podrían estarlo.

- **Reducción de costo**

  El intercambio de recursos reduce automáticamente los costos y, por lo tanto, se puede ahorrar dinero. Por ejemplo, suponga que hay diez usuarios y cada uno requiere una impresora. Si hubieran podido trabajar individualmente, habría que comprar diez impresoras. Si a estos diez usuarios se les permite trabajar en una red, solo dos o tres impresoras serían suficientes.

- **Medio de comunicación**

  Las redes informáticas proporcionan un poderoso medio de comunicación entre las personas que se encuentran en una ubicación geográficamente igual o diferente. Un archivo que se actualizó o modificó en una red, puede ser visto por otros usuarios en la red inmediatamente. Por lo tanto, se vuelve fácil para dos o más personas que viven lejos trabajar en un mismo proyecto dividiéndolo usando una red. Pueden escribir programas, pueden discutir o incluso pueden cambiar o modificar algunos datos usando una red mientras están lejos. De lo contrario, tendrán que esperar varios días por carta o algún otro medio. Por lo tanto, hace cooperaciones rápidas y mejora la comunicación entre humanos.

1.5 TÉCNICAS DE CAMBIO

Cuando tenemos múltiples estaciones de trabajo, surge el problema de conectarlas para una comunicación uno a uno. Una solución para este problema es establecer conexiones punto a punto entre cada par de estaciones individuales. para interconectar ***norte*** estaciones, ***n(n-1)/2*** se necesitan conexiones individuales. Por ejemplo, para conectar 10 componentes, se requiere un total de 45 conexiones individuales; para 100 componentes, se requieren 4950 conexiones. Los requisitos aumentan drásticamente a medida que aumenta el número de estaciones a interconectar, y para un gran número de estaciones, es casi imposible mantener conexiones individuales. Una solución bien conocida para este tipo de problemas es, ***traspuesta***. ¡Pensemos, cómo serían las cosas si solo pudiéramos usar nuestro teléfono para hablar con una sola persona! Por lo tanto, existen requisitos para los sistemas de conmutación para enrutar nuestras llamadas en todo el mundo. Los conmutadores se pueden colocar en la ruta de transmisión para que las estaciones no necesiten estar interconectadas directamente. En esta sección, analizaremos brevemente varias técnicas de conmutación que se utilizan actualmente.

En general, se han utilizado tres métodos de conmutación: ***Cambio de circuito***, ***conmutación de mensajes*** y ***conmutación de paquetes***.

1.5.1 Conmutación de circuitos

Cambio de circuito es la tecnología de transmisión que se viene utilizando desde las primeras redes de comunicación en el siglo XIX. La idea básica de la conmutación de circuitos es establecer una ruta lógica dedicada entre dos usuarios o máquinas, antes del comienzo de la comunicación de datos. En la conmutación de circuitos, la persona que llama primero debe establecer una conexión con la persona que llama antes de cualquier comunicación. Durante el establecimiento de la conexión, los recursos se asignan entre la persona que llama y la persona que llama. La aplicación más común de la conmutación de circuitos es *teléfono* la red. El mecanismo de conmutación de circuitos se divide en tres fases: establecimiento del circuito, transferencia de datos y desconexión del circuito.

Características de la conmutación de circuitos

• Se necesita una ruta dedicada de extremo a extremo.

• La ruta de conexión debe establecerse antes del comienzo de la transmisión de datos.

• La capacidad del canal debe reservarse entre cada par de nodos, ya sea que esa capacidad se utilice o no.

• La tecnología está desarrollada para manejar datos de voz debido a su requisito clave de que no debe haber demoras en la transmisión y debe mantenerse una tasa de transmisión de señal constante.

Ventajas de la conmutación de circuitos

• Es muy adecuado para propósitos de comunicación en tiempo real.

• El servicio proporcionado por conmutación de circuitos está garantizado ya que el canal permanece dedicado a los usuarios durante toda la sesión de comunicación.

Desventajas de la conmutación de circuitos

• Cualquier sin usar ***banda ancha*** sobre el circuito asignado se desperdicia.

• La línea puede estar inactiva la mayor parte del tiempo.

• Tanto el emisor como el receptor deben funcionar con la misma velocidad. Esto limita la utilidad de la red para interconectar una variedad de terminales y computadoras anfitrionas.

Banda ancha : A comunicación canal utiliza un frecuencia específica para transmitir la electro-energía magnética que representa el datos. Transmitiendo la información requiere más de una sola frecuencia. Y para este propósito una banda de espectro alrededor la frecuencia nómica se requiere, lo que se conoce como ancho de banda de la señal

1.5.2 Cambio de mensaje

La conmutación de mensajes se refiere a una técnica de conmutación que implica la transmisión de mensajes de nodo a nodo a través de una red. En esta técnica, la computadora de origen envía primero los datos o el mensaje a la oficina de conmutación, que almacena los datos en su búfer. A continuación, busca un enlace gratuito a otra oficina de conmutación y luego envía los datos a esta oficina. Este proceso continúa hasta que los datos se envían a los equipos de destino. Debido a su principio de funcionamiento, también se conoce como *almacenamiento y reenvio*. Es decir, almacene primero (en la oficina de conmutación), reenvíe después, un salto a la vez.

Ventajas de la conmutación de mensajes

- Sin esperar la configuración de la ruta. Tan pronto como un usuario tenga datos para enviar, puede transmitirlos por el canal.
- El canal se puede utilizar por completo.

- Se puede implementar la prioridad de los mensajes.
- La transmisión se puede realizar muy fácilmente a todos los nodos de una red.

Desventajas de la conmutación de mensajes

- No apto para transmisión en tiempo real
- En el cambio de mensajes, el tamaño del mensaje no es fijo. Para mensajes muy grandes, se requiere que los nodos tengan memorias intermedias grandes, que pueden no implementarse en la práctica debido a restricciones económicas y técnicas.
- Dado que no existe una limitación de tamaño del mensaje, los mensajes largos pueden mantener el canal bloqueado durante un largo período de tiempo.

1.5.3 Conmutación de paquetes

La conmutación de paquetes es muy similar a la conmutación de mensajes. La principal diferencia es que la longitud de las unidades de datos que pueden presentarse a la red está limitada en una red de conmutación de paquetes.

Con la conmutación de mensajes, no hay límite en el tamaño del bloque, por el contrario, la conmutación de paquetes impone un límite superior estricto en el tamaño del bloque. Se especifica un tamaño fijo de paquete que se puede transmitir a través de la red. Otro punto de su diferencia con la conmutación de mensajes es que los paquetes de datos se almacenan en el disco en la conmutación de mensajes, mientras que en la conmutación de paquetes, todos los paquetes de tamaño fijo se almacenan en la memoria principal. Esto mejora el rendimiento a medida que se reduce el tiempo de acceso (tiempo necesario para acceder a un paquete de datos), por lo que se mejora el rendimiento (medida de rendimiento) de la red.

Existen dos enfoques para implementar la conmutación de paquetes. Estos son: el enfoque de datagrama y el enfoque de circuito virtual.

1.6 SERVICIO ORIENTADO A CONEXIÓN Y SIN CONEXIÓN

Sobre la base de*acuse de recibo enviado por el receptor*, hay dos técnicas distintas que se utilizan en las comunicaciones de datos para transferir datos. Estos son:

orientado a la conexión y *servicios sin conexión*. Cada servicio puede caracterizarse por una calidad de servicio. En general, un servicio confiable se implementa haciendo que el receptor acuse recibo de cada mensaje para que el remitente esté seguro de que llegó.

- ***Servicio orientado a la conexión*** sigue el modelo del sistema telefónico. Para realizar una llamada a alguien, primero tenemos que descolgar el teléfono y marcar el número. Una vez establecida la conexión, hablamos y luego colgamos.

  El servicio orientado a la conexión requiere que se establezca una conexión antes de que se pueda enviar cualquier dato. Establece enlaces virtuales entre sistemas finales a través de una red. Una vez establecida la conexión, los datos se transfieren. Tan pronto como se completa la transmisión, el usuario del servicio libera la conexión. Este procedimiento requiere un acuse de recibo específico para la información si la conexión se establece o no. Este método a menudo se denomina servicio de red "confiable". En algunos casos, cuando se establece la conexión, el remitente, el receptor y la subred negocian los parámetros que se utilizarán, como el tamaño máximo del mensaje, la calidad de servicio requerida y otros temas. Por lo general, un lado hace una propuesta y el otro lado puede aceptarla, rechazarla o hacer una propuesta como alternativa a la propuesta anterior.

- La analogía de enviar cartas y postales explica mejor la ***servicio sin conexión.*** Cada mensaje (carta) lleva una dirección de destino completa y cada uno se enruta a través del sistema independientemente de todos los demás. Normalmente, cuando se envían dos mensajes al mismo destino, el primero enviado será el primero en llegar. Pero también es posible que el primero enviado se retrase para que el segundo llegue primero. En un servicio sin conexión, no hay una configuración inicial de extremo a extremo para una sesión; cada paquete se enruta de forma independiente a su destino. El remitente simplemente comienza a enviar paquetes (llamados datagramas) al destino. Este servicio no tiene la confiabilidad del método orientado a conexión, pero es útil para transferencias de ráfagas periódicas. Ninguno de los sistemas debe mantener información de estado para los sistemas a los que envía o recibe transmisiones.

1.7 REDES DE BROADCAST Y PUNTO A PUNTO

Sobre la base de*tecnología de transmisión*, las redes informáticas se pueden dividir en dos categorías:*transmisión*y*punto a punto*.

- ***Redes de difusión***tener un único canal de comunicación que es compartido por todas las máquinas en la red. Aquí, los mensajes se dividen en*paquetes* que luego se transmiten a todas las máquinas en el canal. Un paquete enviado por cualquier máquina es recibido por todas las máquinas, pero solo esa máquina procesa el paquete para el que está destinado. Un campo de dirección (dirección de destino) dentro del paquete especifica a quién está destinado. Al recibir un paquete, una máquina verifica el campo de dirección. Si el paquete está destinado a sí mismo, lo procesa; si está destinado a alguna otra máquina, simplemente se ignora. Sin embargo, en las cadenas de transmisión, tenemos el problema de decidir quién usa el canal, si hay competencia por él. Para esto, *control de acceso medio*(MAC) se utiliza para determinar quién va después.

  Como analogía, imaginemos un escenario donde un maestro toma la asistencia en una clase en particular. Llamará a todos los números de lista en el salón de clases, todos escucharán pero solo responderá el estudiante respectivo. Difusión implica "*enviar una señal donde varias partes pueden escuchar a un solo remitente*".

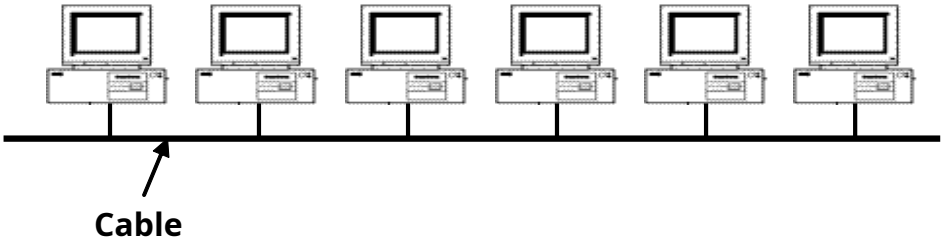


Fig.1.2: Red ABroadcast

- Nuevamente, imaginemos un escenario en el que un maestro está tomando la asistencia en una sala de exámenes. Se dirigirá a cada miembro de la sala y tomará la asistencia individualmente. Aquí, solo una persona escucha y solo responde a la llamada.***red punto a punto***es un método de comunicación donde un "punto" (persona o dispositivo o entidad) habla a otra entidad. La red punto a punto consta de muchas conexiones entre pares individuales de máquinas. paquete, en este

tipo de red, puede que tenga que visitar una o más máquinas intermedias para llegar desde el origen hasta el destino. A menudo son posibles múltiples rutas de diferentes longitudes en redes punto a punto. Por lo tanto, se requieren algoritmos de enrutamiento para determinar las mejores rutas entre las múltiples rutas disponibles.

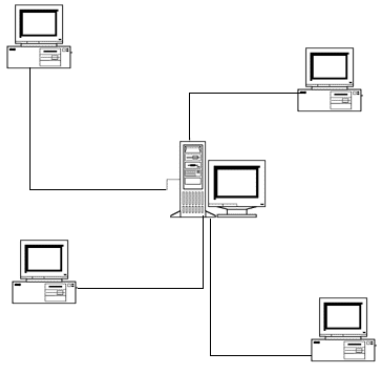


Fig. 1.3: Red punto a punto

1.8 CATEGORÍAS DE RED

Siempre que tenemos un conjunto de computadoras o dispositivos de red para conectar, hacemos las conexiones, según el diseño físico y nuestros requisitos. Sobre la base del área geográfica cubierta, las redes informáticas se pueden clasificar en las siguientes tres categorías:

- Red de área local
- Red de área metropolitana
- Red de área amplia

1.8.1 Red de área local

A***Red de área local***o**LAN**, es una red de datos de alta velocidad que cubre un área geográfica relativamente pequeña, como un edificio, un laboratorio o una escuela. Por lo general, conecta estaciones de trabajo, computadoras personales, impresoras, servidores y otros dispositivos. Las LAN ofrecen a los usuarios de computadoras muchas ventajas, incluido el acceso compartido a dispositivos y aplicaciones, el intercambio de archivos entre usuarios conectados y la comunicación entre usuarios a través del correo electrónico y otras aplicaciones. Las LAN difieren en la forma en que se conectan las computadoras (es decir, su topología), en cómo se mueve la información por la red (es decir, su tecnología de transmisión) y en su tamaño.

IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) es una organización de publicación y estándares de EE. UU. responsable de muchos estándares LAN.

dardos como la serie 802.*IEEE 802.3,* llamado popularmente *ethernet* es la LAN más popular, normalmente opera de 10 Mbps a 10 Gbps y está presente en la mayoría de las grandes organizaciones y oficinas. En una configuración LAN típica, una computadora se designa como la *servidor*. Almacena todo el software que controla la red, así como el software que pueden compartir las computadoras conectadas a la red. Las computadoras conectadas al servidor se llaman *estaciones de trabajo*. Los diversos componentes y estándares asociados con una LAN se discutirán en *Unidad 5* de este curso. Las dos implementaciones de LAN más utilizadas se muestran a continuación:

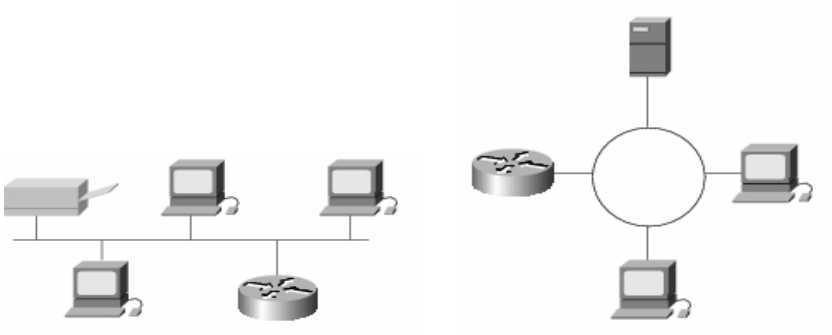


Fig. 1.4: (a) Ethernet o IEEE 802.3 (b) Token Ring o IEEE 802.5

Los protocolos LAN funcionan en las dos capas más bajas, es decir, la *físico* y el *Capa de enlace de datos* del modelo de referencia OSI. Discutiremos el concepto de capa del modelo de referencia ISO-OSI en la siguiente unidad, que es "Modelos de red".

1.8.1.1 Métodos de transmisión LAN

Las transmisiones de datos LAN se dividen en tres clasificaciones: *unidifusión*, *multidifusión*, y *transmisión*. En cada tipo de transmisión, se envía un solo paquete a uno o más nodos.

- ***Unidifusión:*** Unicast es un método de transmisión uno a uno en el que la red lleva un mensaje a un receptor, como desde un servidor a una estación de trabajo LAN. En un entorno de unidifusión, aunque varios usuarios pueden solicitar la misma información del mismo servidor al mismo tiempo, como un videoclip; se envían flujos de datos duplicados. Unicast envía flujos de datos separados a cada computadora que solicita los datos, lo que a su vez inunda la red con tráfico.

- ***multidifusión:*** La multidifusión es un método de transmisión de uno a muchos en el que la red lleva un mensaje a múltiples receptores al mismo tiempo. La multidifusión es similar a la transmisión, excepto que la multidifusión significa enviar a un grupo específico, mientras que la transmisión implica enviar a todos, ya sea que quieran el tráfico o no. Al enviar grandes cantidades de datos, la multidifusión ahorra un ancho de banda de red considerable porque la mayor parte de los datos se envía solo una vez. Los datos viajan desde su origen a través de las principales redes troncales y luego se multiplican o distribuyen en puntos de conmutación más cercanos a los usuarios finales. Esto es más eficiente que un sistema de unidifusión, en el que los datos se copian y reenvían a cada destinatario.
- ***Transmisión :*** El concepto de transmisión ya se trata en la sección anterior. *Transmisión* es un método de transmisión de uno a todos en el que la red lleva un paquete a todos los dispositivos al mismo tiempo, pero una máquina en particular para la que está destinado el paquete lo acepta.

1.8.1.2 LAN punto a punto y LAN basada en servidor

En una LAN, esperamos compartir archivos, programas o impresoras, todo sin ser particularmente conscientes de dónde se encuentran realmente los recursos físicos que estamos utilizando. Las LAN que brindan este tipo de servicios generalmente se configuran como "*de igual a igual*" o "*Servidor de cliente*" LAN, o tal vez como una combinación de los dos.

• LAN punto a punto

Todas las máquinas en una LAN punto a punto son iguales. Siempre que los propietarios del archivo den permiso, se puede acceder a un archivo en la máquina A desde la máquina B, y viceversa. Las LAN punto a punto no requieren que ninguna máquina sea un servidor dedicado de alto rendimiento; el servicio de una LAN de igual a igual suele ser más barato por este motivo. Las LAN de igual a igual funcionan bien cuando solo se conecta una pequeña cantidad de máquinas. Pero a medida que crece el tamaño de la LAN, los servicios peer-to-peer pueden volverse bastante desorganizados. Para servir a todos sus pares, cada máquina en la LAN

debe ser lo suficientemente potente y por esta razón aumenta el costo. Para LAN más grandes, la arquitectura de LAN de cliente-servidor dedicada se vuelve más rentable.

Ventajas

Menos gasto inicial–Sin necesidad de un servidor dedicado.

Configuración–Es posible que solo sea necesario reconfigurar un sistema operativo existente (como Windows XP) de la máquina para operaciones de igual a igual.

Desventajas

descentralizado–Sin depósito central para archivos y aplicaciones.

Seguridad–No proporciona la seguridad disponible en un cliente/servidor.

• LAN Cliente-Servidor

Una LAN cliente-servidor consta de una o más máquinas servidor en las que residen archivos y programas compartidos y muchas máquinas cliente donde las personas realizan su tarea. Las máquinas del servidor LAN generalmente tienen una configuración más alta y son más rápidas porque deben servir a muchos usuarios, mientras que las máquinas cliente solo necesitan ser lo suficientemente rápidas para que las use una persona a la vez. Las impresoras compartidas se conectan directamente a un servidor, a un servidor de impresión (una computadora especializada conectada a la red), o a una computadora personal en la red que actúa como servidor de impresión.

Ventajas

centralizado–En el caso de la arquitectura cliente-servidor, los recursos y la seguridad de los datos se controlan a través del servidor.

Escalabilidad–Cualquiera o todos los elementos se pueden reemplazar individualmente a medida que aumentan las necesidades.

Flexibilidad–La nueva tecnología se puede integrar fácilmente en el sistema.

Accesibilidad–Se puede acceder al servidor de forma remota y a través de múltiples plataformas.

Desventajas

Gastos–Requiere inversión inicial en servidor dedicado.

Mantenimiento: las redes grandes requerirán personal para garantizar un funcionamiento eficiente.

Dependencia–Cuando el servidor se cae, las operaciones cesarán en toda la red.

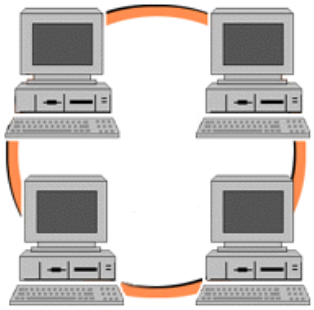


Fig. 1.5: De igual a igual

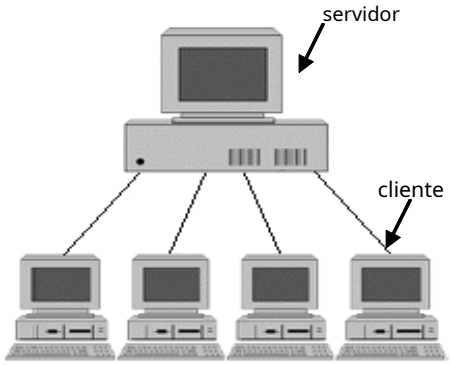


Fig. 1.6: Cliente-Servidor

1.8.2 Red de Área Metropolitana

A***red de área metropolitana***,o**HOMBRE**, es básicamente una versión más grande de una LAN y normalmente usa una tecnología similar. Puede cubrir un grupo de oficinas corporativas cercanas o una ciudad y puede ser privado o público.

Un MAN a menudo proporciona conexiones eficientes ya que tiene capacidades de transmisión de alta velocidad que utiliza algún tipo de componentes de telecomunicaciones para manejar la transmisión de larga distancia. Un ejemplo muy común de MAN es la red de televisión por cable. Otro ejemplo importante es el acceso inalámbrico a Internet de alta velocidad, que ha sido estandarizado como*IEEE 802.16.*

1.8.3 Red de área amplia

A***red de área amplia***o**PÁLIDO**es una red informática que cubre múltiples áreas de distancia, que pueden extenderse por todo el mundo. Las WAN a menudo conectan varias redes más pequeñas, como las redes de área local (LAN) o las redes de área metropolitana (MAN).

Por lo general, una WAN consta de varios nodos de conmutación interconectados. Una transmisión desde cualquier dispositivo se enruta a través de estos nodos internos al dispositivo de destino especificado. Estos nodos no están relacionados con el contenido de los datos; más bien, su propósito es proporcionar una función de conmutación que moverá los datos de un nodo a otro hasta que lleguen a su destino. Tradicionalmente, las WAN se han implementado utilizando *Cambio de circuito* o *conmutación de paquetes* tecnologías Más recientemente, Frame Relay y *Cajero automático* (*Modo de Transferencia Asíncrona*) las redes han asumido papeles importantes. Frame Relay proporciona tasas de datos más altas, costos más bajos, manejo eficiente de la transmisión de datos en ráfagas con menos gastos. Los cajeros automáticos ofrecen más ancho de banda a los usuarios finales a un costo menor.

los ***Internet*** es el ejemplo más conocido de una WAN. Algunos segmentos de Internet también son WAN en sí mismos. Internet es un sistema de redes enlazadas que tienen un alcance mundial y facilitan servicios de comunicación de datos tales como *inicio de sesión remoto*, *transferencia de archivos*, *Email*, los *World Wide Web* etc. Con el aumento de la demanda de conectividad, Internet se ha convertido en una vía de comunicación para millones de usuarios. Internet se restringió inicialmente a las instituciones militares y académicas, pero ahora es un conducto completo para todas y cada una de las formas de información y comercio. Los sitios web de Internet ahora brindan recursos personales, educativos, políticos y económicos a todos los rincones del planeta. La última unidad (Unidad 6) de este curso nos dará el concepto de Internet y varios servicios proporcionados por Internet.

CONSULTA TU PROGRESO-1

1. Indique verdadero o falso:

i) Las redes de difusión comparten un único canal de comunicación.

ii) Para LAN más grandes, la arquitectura de LAN cliente-servidor es menos rentable en comparación con las LAN de igual a igual.

iii) La WAN más popular del mundo es Internet.

iv) La red de televisión por cable es un ejemplo de MAN.

1.9 TOPOLOGÍA DE RED

En esta sección discutiremos cómo las computadoras y otros dispositivos están conectados en una red. Mientras discutimos, podemos encontrarnos con el nombre de varios dispositivos de interconexión de redes como concentrador, conmutador, enrutador, etc.; la función y el funcionamiento de cada dispositivo se tratarán en la Unidad 4 de este curso.

En redes informáticas, ***topología*** se refiere al diseño de los dispositivos conectados. Las topologías de red pueden ser *físico* o *lógico*. ***Topología física*** significa el diseño físico de una red, incluidos los dispositivos, la ubicación y la instalación de cables. Define cómo los sistemas están conectados físicamente. En la actualidad, se utilizan varias topologías físicas para las redes. Algunos de los comunes incluyen el *autobús*, *anillo*, *estrella*, *árbol* y *malla*. Se pueden construir redes más complejas como *híbridos* de dos o más de las topologías básicas anteriores. los ***topología lógica*** define cómo se comunican los sistemas a través de las topologías físicas. Los dos tipos más comunes de topologías lógicas son ***transmisión*** y ***paso de fichas***.

Cuando decidimos qué topología debemos elegir para nuestra red, hay algunos puntos básicos que debemos tener en cuenta. Los factores que deciden qué topología debemos elegir son:

- ***Costo***: El costo es un factor que juega un papel importante para la decisión de la topología. Si queremos crear una red para 4-5 computadoras, no debemos gastar mucho en red.
- ***Escalabilidad***: ¿Cuál es el tamaño de la red que necesitamos hacer y es posible en ese tipo de topología?
- ***Capacidad de ancho de banda***: La velocidad requerida de la red que puede ser atendida por cualquier topología particular.
- ***Facilidad de instalación***: ¿Es fácil instalar la red utilizando la topología seleccionada?
- ***Facilidad de localización de averías y mantenimiento.***: Si tenemos una red, definitivamente también tendremos el problema; así será fácil para el administrador de la red identificar el problema y dar la solución con facilidad y en el menor tiempo posible.

1.9.1 Topología de bus

Una red de topología de bus consta de un solo cable largo al que se conectan todas las computadoras y otros dispositivos. Cualquier nodo conectado al bus puede enviar señales por el cable a todos los nodos de la red; eso significa que un autobús es un medio de transmisión. Cuando más de un nodo comienza a enviar datos a través del bus, se mezclan entre sí y los datos enviados se convierten en basura. Se llama *colisión.* Para evitar la colisión, debe haber algún acuerdo entre los nodos para que cuando una computadora comience a enviar datos, otras se abstengan de enviar datos. Para garantizar una comunicación de datos correcta, ambos extremos del cable están terminados por un dispositivo especial llamado terminador final.

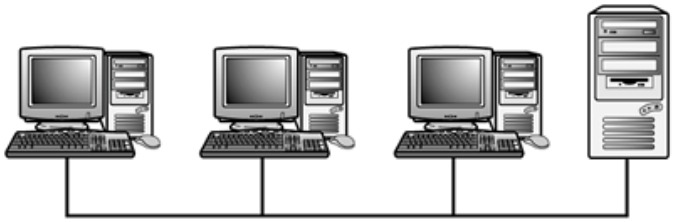


Fig.1.7: Topología ABus

Las topologías de bus Ethernet son relativamente fáciles de instalar y no requieren mucho cableado en comparación con las alternativas. 10Base-2 (delgado) y 10Base-5 (grueso) eran opciones populares de cableado Ethernet para topologías de bus. Algunas ventajas y desventajas de la topología de bus se enumeran a continuación:

Ventajas

• La topología de bus es económica en la instalación

• Requiere menos cable que otras topologías

• Bueno para redes más pequeñas que no requieren mayor velocidad.

• Fácil de agregar sistemas a la red.

Desventaja

- Tecnología obsoleta. La topología de bus se usó en los primeros días de las redes porque era económica de usar y relativamente fácil de configurar.

- Si falla el cable troncal, toda la red queda efectivamente inutilizable.
- Inmanejable en una red grande. Si se agregan más de unas pocas docenas de computadoras a un bus de red, es probable que surjan problemas de rendimiento.
- Difícil de solucionar.

1.9.2 Topología de anillo

En la topología de anillo, las computadoras están conectadas entre sí de tal manera que forman un circuito cerrado. En la práctica, un cable conecta la primera computadora a la segunda computadora, otro cable conecta la segunda computadora a la tercera y así sucesivamente hasta que la última computadora se vuelve a conectar a la primera para completar el ciclo. Cabe señalar que es posible que la topología no parezca físicamente un círculo. El anillo significa que las computadoras están conectadas entre sí en un anillo lógico. Los cables de interconexión pueden adoptar cualquier forma en la práctica. Todos los mensajes viajan a través de un anillo en la misma dirección (ya sea "en el sentido de las agujas del reloj" o "en sentido contrario a las agujas del reloj").

En la topología de anillo, las computadoras que se comunican deben seguir algún acuerdo entre ellas para evitar colisiones, ya que también es una red de transmisión. La principal diferencia entre el bus y el anillo es que la topología de anillo no requiere terminación. porque los sistemas son

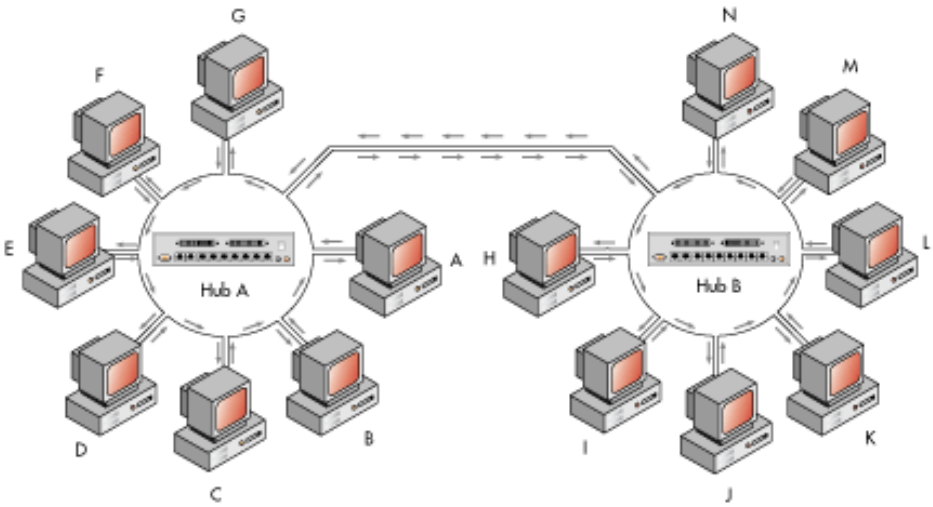


Fig. 1.8: Una topología de doble anillo

como ocurre con la topología de bus. Una falla en cualquier cable o dispositivo rompe el bucle y puede acabar con toda la red.

Los dos principales tipos de red que utilizan la topología en anillo son:

I) *Interfaz de datos distribuidos por fibra* (FDDI) donde una gran red de alta velocidad utiliza cables de fibra óptica en una topología de anillo físico.

ii) *Redes Token-Ring* que utilizan topología de anillo lógico.

Ventajas

• No hay colisión de datos ya que los datos viajan en una sola dirección.

• Más fácil encontrar fallas. Si algún punto se rompe, podemos rastrearlo fácilmente.

• No se requiere terminador.

Desventajas

• La topología de anillo requiere más cable que la topología de bus

• Una interrupción en el anillo provocará la caída de toda la red.

• La adición o eliminación de cualquier nodo puede afectar a toda la red.

1.9.3 Topología en estrella

En el *estrella* topología, todas las computadoras y otros dispositivos de red se conectan a un dispositivo central (controlador) llamado ***centro*** como se muestra en la figura.1.9. Cada dispositivo conectado requiere un solo cable para conectarse al concentrador. Un concentrador normalmente acepta datos de una computadora emisora y los entrega a la computadora a la que se dirigen los datos. Por lo tanto, una red en estrella no es una red de transmisión, sino una red punto a punto.

El uso de un cable separado para conectarse al concentrador permite expandir la red sin interrumpir la red. Debido a que cada computadora usa un cable separado para conectarse al concentrador, la falla de una conexión de red afecta solo a la preocupación de una sola máquina. Las otras computadoras pueden continuar funcionando normalmente.

Fast Ethernet (100Base-TX y 100Base-FX) en una topología en estrella es la LAN más utilizada en la actualidad. *Ethernet 10Base-T* y *Gigabit Ethernet 1000Base-TX* también utiliza topología en estrella.

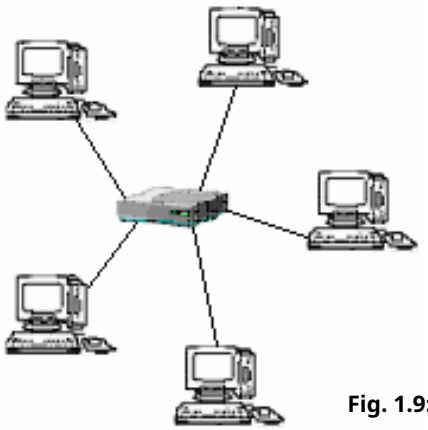


Fig. 1.9: Topología en estrella

Ventajas:

• Las redes en estrella se expanden fácilmente sin interrupción de la red.

• Fácil de agregar/quitar dispositivos a/de la red

• Una interrupción no provoca la caída de toda la red. La falla del cable afecta a un solo usuario.

• Fácil de solucionar y aislar problemas.

• Gestión centralizada ampliamente utilizada

Desventajas:

• Los costos suelen ser más altos que con las redes de bus o anillo.

• Requiere más cable que la mayoría de las otras topologías.

• Si el concentrador falla, ningún dispositivo conectado a él podrá acceder a la red.

En la siguiente tabla se muestran varias redes con sus tipos de cable, topologías:

Tipo de red	Estándar	Tipo de cable	Topología
ethernet	10Base-2	Delgado (RG-58)	Autobús
ethernet	10Base-5	Coaxial grueso (RG-59)	Autobús
ethernet	10Base-T	CAT3 o CAT5 UTP	Estrella
Ethernet rápida	100Base-TX	UTP CAT5	Estrella
Gibabit ethernet	1000BaseTX	CAT5, 5e o UTP	Estrella
Token Ring	todos	UTP o STP	Anillo lógico

Tabla 1.1: Tipos y topologías de cables de red

1.9.4 Topología de malla

La topología de malla incorpora un diseño de red único en el que cada computadora en la red se conecta entre sí, creando una conexión punto a punto entre cada dispositivo en la red. La topología Amesh se usa cuando no puede haber absolutamente ninguna interrupción en las comunicaciones, por ejemplo, los sistemas de control de una planta de energía nuclear. El propósito de la topología de malla es proporcionar un alto nivel de redundancia. Si falla un cable de red, computadoras u otros componentes, los datos siempre tienen una ruta alternativa para llegar a su destino.

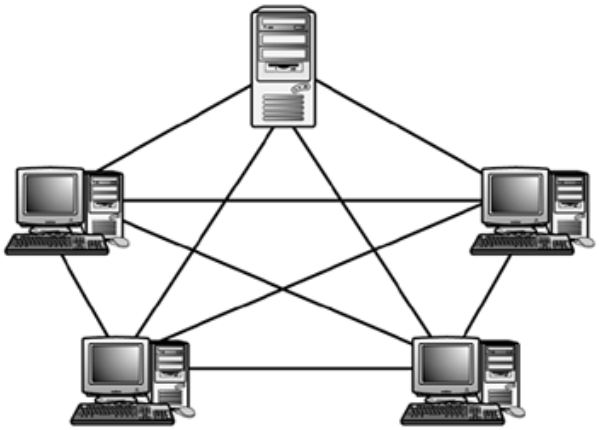


Fig. 1.10: Una topología de malla

Una red de malla totalmente conectada tiene n(n-1)/2 cables para vincular 'n' dispositivos. Por lo tanto, cada dispositivo en la red debe tener puertos de entrada/salida (E/S) 'n-1'. Por ejemplo, en la figura (Fig. 1.10), tenemos cinco sistemas que requieren 10 cables para crear una red de malla. Esta topología se usa principalmente en entornos donde la alta disponibilidad supera los costos asociados con esta cantidad de interconexión. Podemos ver en la figura que el cableado para una red mallada puede ser muy complicado. Además, la solución de problemas de un cable defectuoso puede ser complicada. Debido a esto, la topología de malla rara vez se usa.

ventajas:

• Proporciona caminos alternativos entre dispositivos en la red.

• La red se puede ampliar sin interrumpir a los usuarios actuales.

Desventajas:

• Es costoso porque se requiere una gran cantidad de cableado.

• El enrutamiento del tráfico de la red puede ser difícil debido a todas las diferentes rutas posibles entre los nodos.

• Es muy costoso de cablear.

También existen redes de malla parcial donde algunos de los nodos están conectados con todos los demás, pero otros solo están conectados con los nodos con los que intercambian la mayor cantidad de datos.

1.9.5 Topología de árbol

Una topología de árbol combina las características de las topologías de bus y estrella. Consiste en grupos de estaciones de trabajo configuradas en estrella conectadas a un cable troncal de bus lineal (fig. 1.11). Este enfoque híbrido de bus/estrella admite la expansión futura de la red mucho mejor que un bus (limitado en la cantidad de dispositivos debido al tráfico de transmisión que genera) o una estrella (limitada por la cantidad de puntos de conexión del concentrador) solo.

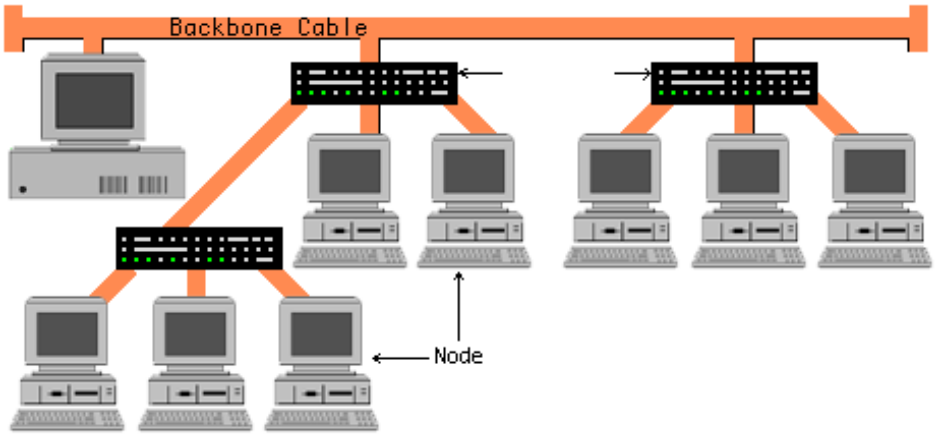


Fig. 1.11: Una topología de árbol

Ventajas de la topología de árbol

• La red es fácil de ampliar simplemente agregando otra sucursal.

• El aislamiento de fallas es relativamente fácil.

Desventajas de la topología de árbol

• Si el cable principal se rompe, toda la red se cae.

• Más difícil de configurar y cablear que otras topologías.

• Si algún concentrador falla, todas las ramas de ese concentrador fallan.

CONSULTA TU PROGRESO-2

1. Complete los espacios en blanco:

I)_____ define la disposición física o lógica de los enlaces en una red.

ii) En_____ topología, cada dispositivo tiene un enlace punto a punto dedicado a cada otro dispositivo.

iii)_____ topología es la combinación de diferentes tipos de topologías.

iv) En_____ topología, cada dispositivo tiene un enlace punto a punto dedicado solo a un controlador central.

v)_____ La topología proporciona rutas alternativas entre los dispositivos de la red.

vi) La topología en anillo requiere _____ cables que la topología en bus.

vii) La topología utilizada en Ethernet es la topología _____.

viii) La topología _____ utiliza un controlador central o concentrador.

2. ¿Qué topología utiliza 10Base-5 y 10Base-2 Ethernet?

3. ¿Qué topología utiliza FDDI?

1.10 TIPOS DE TRANSMISIÓN

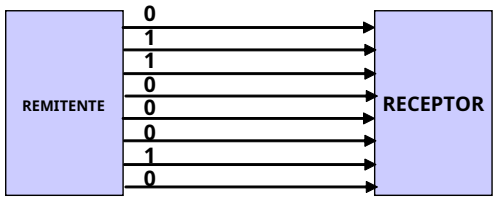
La transmisión digital es el envío de información a través de un medio físico de comunicación en forma de señales digitales. Por lo tanto, las señales analógicas deben digitalizarse primero antes de transmitirse. Sin embargo, como la información digital no se puede enviar directamente en forma de 0 y 1, debe codificarse en forma de una señal con dos estados, por ejemplo:

- dos niveles de tensión con respecto a tierra
- la diferencia de voltaje entre dos cables
- la presencia/ausencia de corriente en un cable
- la presencia/ausencia de luz

La transmisión de datos digitales puede ocurrir en dos modos básicos:***paralela*** y ***de serie.*** Los datos dentro de un sistema informático se transmiten en modo paralelo en buses con el ancho del bus paralelo adaptado al tamaño de palabra del sistema informático. Los datos entre sistemas informáticos generalmente se transmiten en modo serie de bits. En consecuencia, es necesario realizar una conversión de paralelo a serie en una interfaz de computadora cuando se envían datos desde un sistema informático a una red y una conversión de serie a paralelo en una interfaz de computadora cuando se recibe información de una red. El tipo de modo de transmisión utilizado también puede depender de la distancia y la velocidad de datos requerida.

1.10.1 Transmisión en paralelo

transmisión en paralelo comunica bits simultáneamente a través de múltiples líneas (cables, canales de frecuencia); normalmente, el total consta de uno o más bytes a la vez. Los dispositivos en paralelo tienen un bus de datos más ancho que los dispositivos en serie y, por lo tanto, pueden transferir datos en palabras de uno o más bytes a la vez. Como resultado, hay una aceleración en la velocidad de transmisión de bits en paralelo con respecto a la velocidad de transmisión en serie. La temporización para la transmisión paralela la proporciona una señal de sincronización constante enviada a través de un cable separado dentro del cable paralelo; por lo tanto, la transmisión en paralelo se considera síncrona. Las computadoras generalmente están conectadas a impresoras y unidades de disco externas a través de interfaces paralelas, puertos y buses.



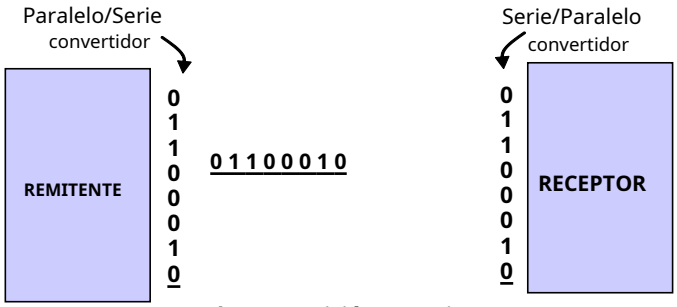
(a) Transmisión en paralelo;
Los ocho bits se envían a través de ocho cables a la vez.

Figura 1.12

1.10.2 Transmisión en serie

La transmisión en serie envía un bit a la vez a través de una sola línea de transmisión. El costo del hardware de comunicación se reduce considerablemente ya que solo se requiere un solo cable o canal para la transmisión de bits en serie, lo que también reduce la velocidad de transmisión. Las líneas telefónicas usan transmisión en serie para datos digitales, por lo que los módems se conectan a la computadora a través de un puerto en serie. A ***puerto de serie*** es un enchufe en

una computadora utilizada para conectar la interfaz serial a una línea o bus serial. A ***Interfaz de serie*** es un canal de datos que transfiere datos digitales en serie; por lo general, se implementa como una tarjeta que se conecta a una ranura de expansión en la placa base de una computadora. Las interfaces seriales tienen múltiples líneas, pero solo una se usa para datos. un externo ***autobús serie*** transporta datos en serie a cualquier dispositivo conectado a él, por ejemplo, Ethernet. La transmisión en serie puede ser *sincrónico* o *asincrónico*.



(b) Transmisión en serie;
Los ocho bits se envían uno tras otro a través de un solo cable.

Figura 1.13

La tecnología de transmisión en serie se utiliza cada vez más para la transmisión de datos digitales. Una gran cantidad de redes de comunicaciones actualizadas aplican la transmisión en serie. Las numerosas aplicaciones incluyen redes informáticas para comunicaciones de oficina, automatización de edificios y fabricación y, por último, Internet. La transmisión de un flujo de bits de un dispositivo a otro a través de un medio de transmisión implica una gran cooperación y acuerdo entre las dos partes (emisor y receptor). Uno de los requisitos más fundamentales es ***sincronización***. El receptor debe conocer la velocidad a la que se reciben los bits para que pueda muestrear el medio a intervalos apropiados para determinar el valor de cada bit recibido. Para lograr la sincronización deseada, existen dos enfoques. ***Sincrónico*** y ***transmisiones asíncronas*** son dos métodos diferentes de sincronización de transmisión.

1.10.2.1 Transmisión asíncrona

La comunicación se llama *asincrónico* si el transmisor y el receptor no necesitan sincronizarse antes de cada transmisión. A

el remitente puede esperar un tiempo arbitrario entre transmisiones y el receptor debe estar listo para recibir los datos cuando lleguen. La mayoría de los dispositivos seriales de PC, como mouse, teclados y módems, son asíncronos.

Como su nombre lo indica, la comunicación asíncrona se realiza entre dos o más dispositivos que funcionan con relojes independientes. Por lo tanto, no hay garantía de que cuando el Punto A comience a transmitir, el Punto B comenzará a recibir, o que el Punto B continuará muestreando a la tasa de transmisión del Punto A. En la transmisión asíncrona, los datos se transmiten un carácter a la vez, donde cada carácter tiene una longitud de cinco a ocho bits. El tiempo o la sincronización solo deben mantenerse dentro de cada carácter; el receptor tiene la oportunidad de resincronizar al comienzo de cada nuevo carácter. La siguiente figura (Fig. 1.14) muestra lo que sucede cuando los relojes de transmisión difieren significativamente. En la figura 1.14 (a), vemos que el receptor muestrea en el punto medio de cada bit de los datos entrantes. En la figura 1.14 (b), el reloj del receptor es demasiado lento; lo que hace que se salte el bit 4 y, como resultado, los datos se corrompen. Para combatir este tipo de problema de temporización, la comunicación asíncrona requiere que se agreguen bits adicionales alrededor de los datos reales para mantener la integridad de la señal. Los bits del carácter se transmiten comenzando por el bit menos significativo.

Los datos transmitidos de forma asincrónica van precedidos de un***bit de inicio***que indica al receptor que un carácter está a punto de comenzar. El final de un carácter es seguido por un***poco de parada***, que le dice al receptor que el carácter ha llegado a su fin, que debe comenzar a buscar el siguiente bit de inicio y que debe ignorar cualquier bit que reciba antes de obtener el bit de inicio. Para evitar confusiones con otros bits, el bit de inicio tiene el doble de tamaño que cualquier otro bit en la transmisión. Para garantizar la integridad de los datos, un***bit de paridad***a menudo se agrega entre el último bit de datos y el bit de parada. El transmisor establece el bit de paridad de modo que el número total de unos en el carácter, incluido el bit de paridad, sea par (paridad par) o impar (paridad impar), según la convención que se utilice. El receptor utiliza este

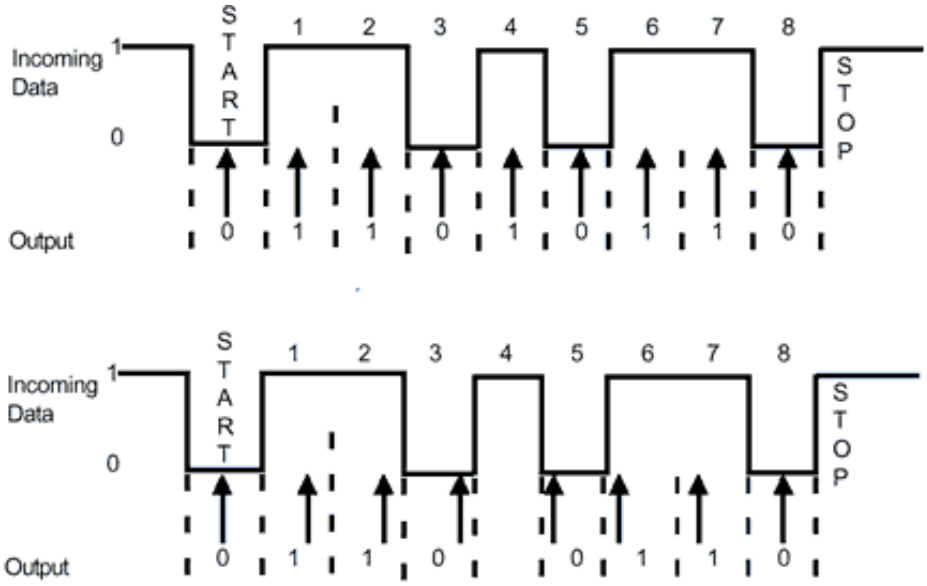


Fig. 1.14: Muestreo de datos asíncrono

bit para la detección de errores. El bit de paridad asegura que los datos a recibir estén compuestos por el mismo número de bits en el mismo orden en que fueron enviados.

1.10.2.2 Transmisión síncrona

El término síncrono se utiliza para describir un método de transferencia de datos en el que un flujo continuo de señales de datos va acompañado de señales de temporización (generadas por un reloj electrónico) para garantizar que el transmisor y el receptor estén sincronizados entre sí. Estos tipos de conexiones se utilizan cuando se deben transferir grandes cantidades de datos muy rápidamente de una ubicación a otra. La velocidad de la conexión síncrona se logra transfiriendo datos en grandes bloques en lugar de caracteres individuales. Un bloque de bits se transmite en un flujo constante sin bits de inicio y parada. Los datos o la información se mueven de un lugar a otro en instantes de tiempo que se miden contra la señal de reloj que se está utilizando. Esta señal generalmente se compone de una o más formas de onda rectangulares de alta frecuencia, generadas por un circuito de reloj de propósito especial.

Los ejemplos típicos de señales sincrónicas incluyen la transferencia y recuperación de información de direcciones dentro de una computadora mediante el uso de un *bus de direcciones*. Por ejemplo, cuando un procesador coloca una dirección en el bus de direcciones, la mantendrá allí durante un período de tiempo específico. Dentro de este intervalo, un dispositivo particular dentro de la computadora se identificará como el que está siendo direccionado y reconocerá el comienzo de una operación relacionada con esa dirección.

En la transmisión síncrona, cada bloque comienza con un patrón de bits de preámbulo y generalmente termina con un patrón de bits de postámbulo. Además, hay algunos otros bits que transmiten información de control. Los datos con el preámbulo, el epílogo y la información de control se denominan como***marco***. En la figura (1.15) se muestra un formato de trama general para transmisión síncrona. Por lo general, la trama comienza con un preámbulo denominado bandera, que tiene una longitud de 8 bits.

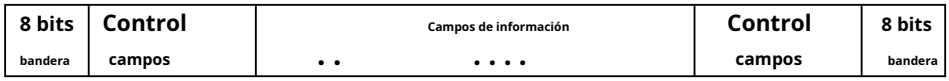


Fig. 1.15: Formato de cuadro síncrono

La misma bandera se utiliza como postámbulo. El receptor busca la aparición del patrón de bandera para señalar el comienzo de un cuadro. Esto es seguido por una cantidad de campos de control, luego un campo de datos, más campos de control y finalmente se repite la bandera.

1.11 MODOS DE COMUNICACIÓN

Los modos de transmisión significan la forma en que se logra una comunicación entre dos dispositivos vinculados. El dispositivo que envía datos o información se llama emisor y el que recibe la información se llama receptor. Un canal puede admitir comunicación unidireccional o comunicación bidireccional a la vez. Según la forma de enlace de comunicación, los modos de transmisión se pueden clasificar en:*símplex*,*medio duplex*y*duplex completo*.

- ***símplex:***A*símplex*La conexión es una conexión en la que los datos fluyen en una sola dirección, desde el transmisor hasta el receptor. Este tipo de conexión es útil si los datos no necesitan fluir en ambos

direcciones, por ejemplo, de nuestro ordenador a la impresora o del ratón a nuestro ordenador. Permite la comunicación unidireccional de datos a través de la red, utilizando todo el ancho de banda del cable para la señal de transmisión.

- ***Medio duplex :*** A*medio duplex*El sistema proporciona comunicación en ambas direcciones, pero solo en una dirección a la vez (no simultáneamente). Por lo general, una vez que una parte comienza a recibir una señal, debe esperar a que el transmisor deje de transmitir antes de responder. Este tipo de conexión permite tener comunicaciones bidireccionales utilizando toda la capacidad de la línea. Una buena analogía para un sistema semidúplex sería una carretera de un solo carril con controladores de tránsito en cada extremo. El tráfico puede fluir en ambos sentidos, pero solo en un sentido a la vez, regulado por los controladores de tráfico. Muchas redes están configuradas para comunicación semidúplex. Un ejemplo adecuado de un sistema semidúplex es uno de dos

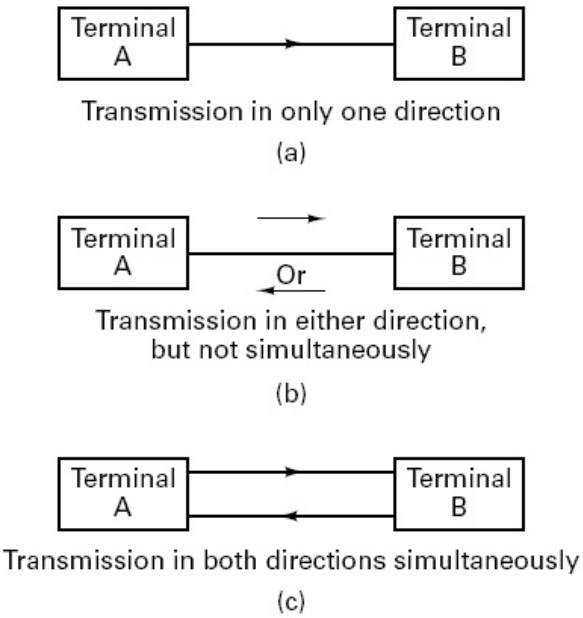


Fig.1.16 (a) Modo simplex (b) Modo semidúplex y

(c) Modo dúplex completo

- ***Duplex completo*:**El modo de transmisión preferido para la comunicación de red es el*duplex completo*modo. A*duplex completo*, permite la comunicación en ambas direcciones simultáneamente. Cada extremo de la línea puede transmitir y recibir al mismo tiempo, lo que significa que el ancho de banda se divide en dos para cada dirección de transmisión de datos si se utiliza el mismo medio de transmisión para ambas direcciones de transmisión.

sion Una buena analogía para un *duplex completo* El sistema sería una carretera de dos carriles con un carril para cada dirección. Por ejemplo, las redes telefónicas son full-duplex, ya que permiten que ambos interlocutores hablen y se escuchen al mismo tiempo.

1.12 PROTOCOLOS Y NORMAS

Se requieren muchos tipos diferentes de protocolos y estándares de red para garantizar que una computadora pueda comunicarse con otra computadora ubicada en el escritorio de al lado o en cualquier lugar del mundo.

Un protocolo de red es un conjunto de reglas que rigen las comunicaciones entre computadoras y otros dispositivos en una red. Estas reglas incluyen pautas que regulan las características de una red, como el método de acceso, las topologías físicas permitidas, los tipos de cableado y la velocidad de transferencia de datos. Algunos protocolos también admiten el reconocimiento de mensajes y la compresión de datos diseñados para una comunicación de red confiable y/o de alto rendimiento. Se han desarrollado cientos de protocolos de redes informáticas diferentes, cada uno diseñado para propósitos y entornos específicos.

El OSI de siete capas (*Sistemas abiertos de interconexión*), creado por la ISO (*Organización de Estándares Internacionales*), define los entornos de interconexión de redes. Proporciona una descripción de cómo interactúan el software y el hardware para permitir la comunicación entre computadoras. Un *interfaz* separa cada capa de las que están por encima y por debajo de ella; cada capa proporciona servicios a la capa directamente encima de ella. Aprenderemos el papel y la funcionalidad de estas capas con algo de detalle en la Unidad 2.

CONSULTA TU PROGRESO-3

1. Elige la respuesta correcta:

i) En modo simplex de transmisión

a) La transmisión de datos es unidireccional.

b) Los datos solo se pueden transmitir a distancias pequeñas

c) El formato de datos es simple

re. Ninguna de las anteriores

ii) En transmisión de datos semidúplex

a) Los datos pueden ser transmitidos en una sola dirección

b) Los datos se pueden transmitir en ambas direcciones

c) Los datos se pueden transmitir en ambas direcciones simultáneamente

re. Ninguna de las anteriores

iii) En_____transmisión, un bit de inicio y un bit de parada forman un byte de carácter.

a) asíncrono b) síncrono

c) paralelo d) ninguno de estos

iv)_____comunica bits simultáneamente a través de múltiples líneas.

a) transmisión en serie b) comunicación síncrona

c) comunicación asíncrona

d) transmisión paralela

1.13 RESUMAMOS

Esta unidad proporciona el concepto básico de redes informáticas.**Redes** es la conexión de computadoras (no necesariamente a grandes distancias) para que puedan comunicarse, compartiendo hardware y software, uniendo así el poder de procesamiento. En esta unidad también se presentan varios conceptos tales como servicios orientados a conexión y sin conexión, redes de difusión y punto a punto, etc. También estamos familiarizados con los diferentes tipos de redes informáticas con sus ventajas y desventajas relativas. También proporciona el concepto de topologías de red. Aparte de esto, los tipos de modos de transmisión y comunicación también se tratan hacia el final. En esta unidad también se incluye una breve introducción al protocolo y al estándar.

1.14 RESPUESTAS PARA COMPROBAR TU PROGRESO

CONSULTA TU PROGRESO-1

i) Verdadero ii) Falso iii) Verdadero iv) Verdadero

CONSULTA TU PROGRESO-2

1. i) Topología ii) malla iii) híbrido iv) estrella
   v) Malla vi) más vii) autobús viii) estrella

2. Topología de bus 3. Topología en anillo

CONSULTA TU PROGRESO-3

1. i) a) La transmisión de datos es unidireccional
   ii) b) Los datos se pueden transmitir en ambas direcciones
   iii) a) asíncrono iv) d) transmisión en paralelo

1.15 LECTURAS ADICIONALES

1. *"Redes informáticas"*, Andrew S. Tanenbaum, Prentice Hall India.
2. *"Datos y Comunicaciones Informáticas"*, William Stallings, Pearson Prentice Hall.

1.16 PREGUNTAS MODELO

1. Definir red informática. ¿Cuáles son los objetivos de las redes informáticas?
2. ¿Cuál es la diferencia entre conmutación de circuitos y conmutación de paquetes?
3. Explique el concepto de conmutación de mensajes.
4. ¿Cuáles son las ventajas y desventajas de la conmutación de circuitos?
5. Comparar y contrastar diferentes topologías de red.
6. Enumerar las ventajas y desventajas relativas de las topologías de bus, anillo y estrella.
7. Diferenciar entre redes sin conexión y redes orientadas a la conexión.
8. ¿Cuáles son los diferentes modos de comunicación? Distinguir el modo de transmisión símplex, semidúplex y dúplex completo.
9. Distinguir entre LAN punto a punto y LAN cliente-servidor.
10. Describa brevemente las diversas características de la transmisión en serie y en paralelo.

UNIDAD-2: MODELOS DE RED

ESTRUCTURA DE LA UNIDAD

2.1 OBJETIVOS DE APRENDIZAJE

Después de pasar por esta unidad, podrá:

- describir los problemas de diseño de la arquitectura de red.
- describir las funciones de varias capas de red.
- definir varias terminologías de red.
- distinguir entre servicios orientados a la conexión y servicios sin conexión.
- comparar entre modelo OSI y TCP/IP.

2.2 INTRODUCCIÓN

El tema de la red informática y su estructura se puede entender mejor estudiando su software de comunicación subyacente y el hardware relacionado. Históricamente, las primeras redes informáticas se diseñaron para funcionar en hardware que luego resultó ser fatal. Conduce a la inflexibilidad de las redes informáticas. El desarrollo posterior de las redes informáticas se basó

principalmente en software altamente estructurado implementado en hardware en constante cambio. El software de comunicación en una red informática es un software muy pensado que está organizado como software en capas. La estructura interna del software en capas se analiza en esta unidad.

2.3 PROBLEMAS DE DISEÑO DE LA CAPA

Antes de proceder a aprender la estructura en capas del software de red y varios aspectos de la red informática, debemos familiarizarnos con los diferentes problemas de diseño de la capa.

-Para intercambiar datos entre dos procesos que se ejecutan en dos computadoras, debe haber alguna técnica de direccionamiento para la comunicación correcta.

-La comunicación puede ser unidireccional (comunicación simplex)) o en ambas direcciones alternativamente una tras otra (comunicación half-duplex) o bidireccional (comunicación full-duplex).

-El control de errores también es un tema importante porque los canales de comunicación físicos no son perfectos, sino más bien propensos a errores. Ahora ambas partes comunicantes deben acordar qué código de detección y corrección de errores usar. El receptor también debe informar al remitente qué mensaje se recibe correctamente y cuál no.

-Durante la transferencia, los mensajes a menudo se dividen en partes y es posible que no todas las partes se transfieran en el orden correcto. Esta pérdida de secuenciación debe ser atendida por la preocupación del protocolo.

-Debe haber una disposición para evitar que un emisor rápido inunde a un receptor lento.

-La preocupación de la capa puede decidir utilizar la misma conexión para varias conversaciones no relacionadas mediante la técnica de multiplexación y demultiplexación.

-Cuando hay muchos caminos entre el emisor y el receptor, se debe tomar una decisión sobre qué ruta tomar. La decisión puede ser estática o dinámica.

2.4 JERARQUÍA DE PROTOCOLO

A partir de los problemas de diseño discutidos anteriormente, es evidente que el diseño de la red informática es un trabajo complejo. Para reducir la complejidad del diseño, la mayoría de las redes informáticas se organizan como una pila de capas, una encima de la otra, tomando el servicio que ofrece la capa inferior. En esta estructura, una capa inferior brinda ciertos servicios a la capa que se encuentra justo encima de ella, al mismo tiempo que protege los detalles de la implementación de esos servicios de la capa superior.

En la organización en capas, una capa en particular en una máquina habla con una capa similar en otra máquina. Durante la conversación, siguen algunas reglas y convenciones. Estas reglas y convenciones se conocen colectivamente como**protocolo**. Aquí hemos utilizado el término**mirar**lo que significa entidades que comprenden la capa correspondiente en diferentes máquinas. Entonces, son los pares los que hablan usando su propio protocolo.

En la Fig. 2.1 se muestra una red de cinco capas. Aquí podemos ver una flecha punteada entre pares y una flecha continua entre dos capas adyacentes. La flecha punteada indica esa capa*norte*en una máquina se comunica con la capa*norte*en

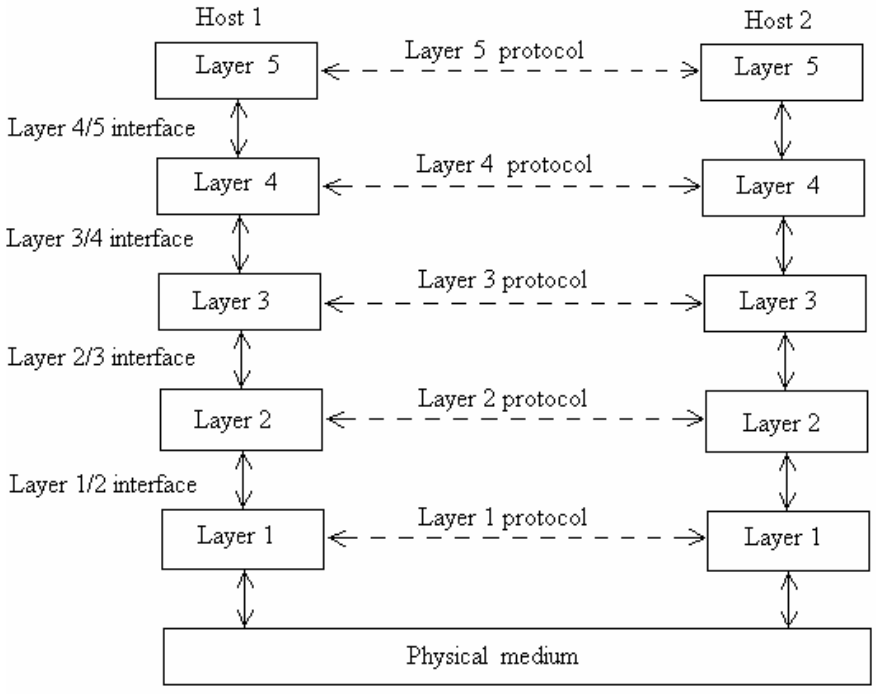


Fig. 2.1: Capas, protocolos e interfaces

Cada capa se comunica con la capa del mismo nivel en la otra máquina. Pero todas estas comunicaciones son virtuales ya que ningún dato se mueve realmente a través de estas líneas punteadas; más bien, los datos se mueven a través de líneas de flecha sólidas. En el proceso de comunicación, cada capa entrega sus datos a la capa inmediatamente inferior. De esta manera, los datos se mueven hacia abajo hasta la capa más baja, debajo de la cual se encuentra el medio físico a través del cual se lleva a cabo la comunicación real. Esta es la imagen en la máquina de envío. La imagen es justo opuesta en la máquina receptora donde los datos se mueven hacia arriba después de que ingresan a la computadora desde el medio físico.

En la Fig. 2.1 cada capa obtiene**Servicio**de la capa justo debajo de ella. Una capa realiza algunas operaciones primitivas para dar el servicio requerido por la capa superior inmediata. La transacción entre dos capas adyacentes se lleva a cabo según una guía predefinida donde cada capa realiza una colección específica de funciones. Entonces, existe un**interfaz**entre cada par de capas adyacentes a través de las cuales intercambian datos e información y, en el proceso, la capa superior obtiene el servicio de la capa inferior. Mientras la interfaz no cambie, se puede cambiar el hardware subyacente en cualquier momento sin afectar las operaciones de la red. Como por ejemplo, los cables de cobre subterráneos se reemplazan gradualmente por cables de fibra óptica, pero no obstaculizan las operaciones de la red existente porque las interfaces entre las capas permanecen sin cambios.

Las capas y sus protocolos se conocen colectivamente como red.**arquitectura**. En la arquitectura de red, cada capa tiene su propio protocolo. Los protocolos utilizados por todas las capas de un sistema de red se denominan**pila de protocolos**. Pila significa algo dispuesto uno encima de otro, como una pila de ladrillos o algo similar. En una arquitectura de red en capas, las capas se organizan de abajo hacia arriba, una sobre otra, por lo que sus respectivos protocolos también se encuentran uno sobre otro. Es por eso que las pilas de protocolo de nombre. En el mismo sentido, la jerarquía de protocolos indica que el protocolo de cualquier capa está jerárquicamente por encima del protocolo de la capa inmediatamente inferior.

2.5 MODELO DE REFERENCIA ISO-OSI

El modelo de referencia OSI se muestra en la Fig. 2.2. La Organización Internacional de Normalización (ISO) desarrolló una propuesta para un modelo de red y la re-

El modelo resultante se conoce como modelo de referencia ISO - OSI. Este modelo tiene siete capas y las capas son:

i) Capa de aplicación

ii) Capa de presentación

iii) Capa de sesión

iv) Capa de transporte

v) Capa de red

vi) Capa de enlace de datos

vii) Capa física

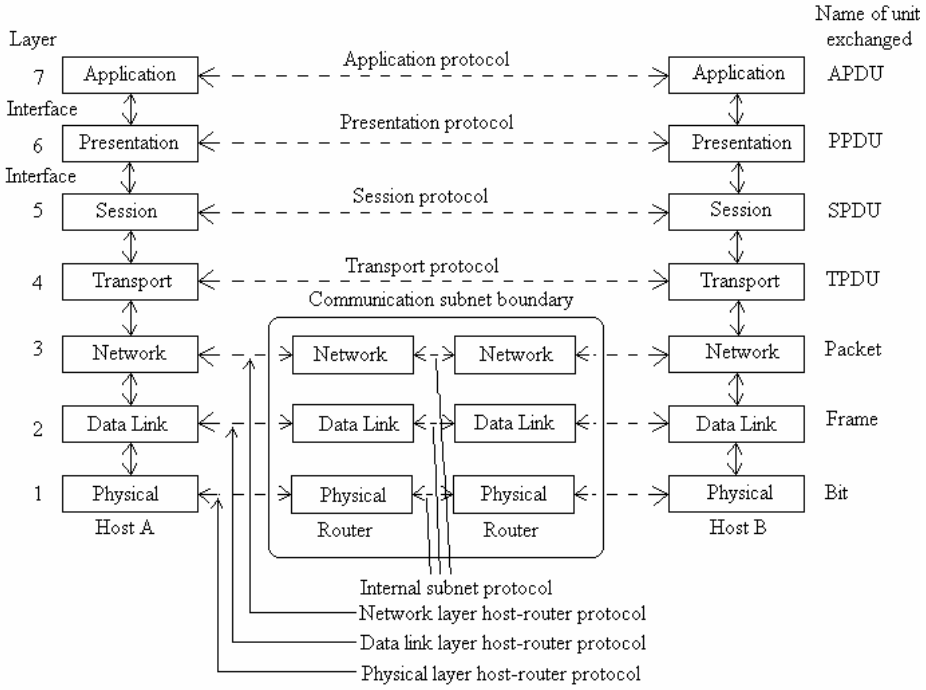


Fig. 2.1: Capas, protocolos e interfaces

(i) La capa física: La capa física es responsable de transmitir bits sin procesar a través del canal de comunicación. Esta capa es para asegurar cómo enviar 1 bit desde la computadora emisora como 1 bit a la computadora receptora; no como 0 bits. La capa física también se ocupa de los problemas de cuántos bits por segundo se transmitirán, qué nivel de voltaje se usará para representar el 1 y el 0, si la transmisión será unidireccional o bidireccional, cómo se establecerá la conexión inicial y cómo se terminará al final. , cuántos pines tiene el conector de red y qué pin es para qué, etc. Por lo tanto, los problemas de diseño de la capa física son principalmente mecánicos, eléctricos y orientados a procedimientos.

(ii) La capa de enlace de datos: La capa de enlace de datos utilizó la instalación de transmisión sin procesar y la transforma en una instalación aparentemente libre de errores para ser utilizada por la capa de red. Esta capa divide los datos de entrada en marcos insertando el límite de marco apropiado, transmite los marcos secuencialmente y procesa el acuse de recibo enviado por la computadora receptora. Si una trama es completamente destruida por una ráfaga de ruido, es deber de la capa de enlace de datos retransmitirla desde la máquina de origen. La capa de enlace de datos también garantiza que un remitente rápido no inunde a un receptor lento enviando datos a una velocidad superior a la que puede manejar el receptor. Esto se llama control de flujo. En una red de transmisión, es deber de la subcapa MAC (control de acceso al medio) de la capa de enlace de datos decidir quién accederá al medio de transmisión en un momento determinado.

(iii) La capa de red: La capa de red controla el funcionamiento de la subred. La capa es para determinar cómo se enrutan los paquetes desde el origen hasta el destino. El enrutamiento puede ser estático o dinámico según la carga de tráfico y la disponibilidad del canal.

Demasiados paquetes pueden causar congestión (atascos de tráfico) y el control de dicha congestión también es un deber de la capa de red.

La operación de la subred requiere un costo, por lo tanto, alguna función de contabilidad también está integrada en la capa de red. Cuando un paquete cruza la frontera nacional, la capa de red debe tratar otros aspectos de la contabilidad.

Los paquetes tienen que viajar entre redes heterogéneas que se ejecutan en diferentes plataformas utilizando diferentes protocolos de red y la capa de red también es responsable de resolver todos los problemas que surgen de tales situaciones.

(iv) La capa de transporte: La capa de transporte acepta datos de la capa de sesión, los divide en unidades más pequeñas si es necesario, las entrega a la capa de red y garantiza que todas las piezas se entreguen correctamente al receptor. Las funciones anteriores deben realizarse de manera eficiente y de tal manera que no afecte la capa superior en caso de que haya algún cambio en el hardware.

Normalmente, la capa de transporte crea una conexión individual para cada sesión. Si se requiere un alto rendimiento, la capa de transporte puede ser

Establecer múltiples conexiones de red, dividiendo los datos entre conexiones individuales, mejorando así el rendimiento.

Para reducir costes, la capa de transporte también puede multiplexar varias conexiones de transporte en la conexión de red. Sin embargo, la capa de sesión no debe ver conexiones múltiples o multiplexación.

La capa de transporte también determina el tipo de servicio dado al usuario de la red. La conexión punto a punto sin errores es el servicio de capa de transporte más popular donde los mensajes o bytes se entregan en el orden en que se enviaron. Otro tipo de servicio de transporte es el transporte de mensajes aislados sin garantía de entrega. La capa de transporte es una verdadera capa final entre el origen y el destino.

La capa de transporte también realiza el control de flujo.

(v) La capa de sesión: Esta capa ofrece facilidad a diferentes usuarios en diferentes computadoras para establecer sesiones entre ellos. Una sesión permite a un usuario iniciar sesión de forma remota en una máquina distante y transferir archivos entre las dos máquinas. La capa de sesión realiza la gestión de tokens para proporcionar una comunicación unidireccional. También proporciona un servicio llamado sincronización.

(vi) La capa de presentación: Esta capa realiza el trabajo de presentación de datos siguiendo las reglas de sintaxis y semántica. Antes de presentar los datos al usuario, transforma los datos en su forma aceptable. Por ejemplo, supongamos que escribimos un nombre como Sri Nilimoy Choudhury, mientras que se escribirá como Choudhury, Sr. Nilimoy en Europa o EE. UU. Escribimos la fecha como dd/mm/aaaa, la moneda como Rs. mientras que en el país occidental se escribe como mm/dd/yyyy y $, etc. Por lo tanto, estas conversiones las realiza la capa de presentación.

(vii) La capa de aplicación: Esta capa es la capa más cercana a todos los usuarios de la red. Ofrece una variedad de protocolos que se necesitan comúnmente.
Ayuda a transferir archivos. Los diferentes sistemas de archivos tienen un significado diferente en diferentes máquinas con diferentes formatos de datos, etc. Cuando los archivos se transfieren de una máquina a otra con un sistema de archivos diferente, la capa de aplicación toma las medidas necesarias para resolver las anomalías.

CONSULTA TU PROGRESO-1

1. ¿Qué es*Servicio*y la interfaz?
2. ¿Qué entiende por arquitectura de red?
3. ¿Cuántas capas hay en el modelo ISO-OSI?
4. ¿Cuáles son las funciones principales de la capa de enlace de datos?

2.6 TERMINOLOGÍA

Cuando estudiamos el tema de la red informática, nos encontramos con muchas terminologías asociadas con el tema. Ya nos hemos encontrado con algunas terminologías como**par, interfaz, protocolo, servicio**etc. Algunos otros se describen brevemente a continuación:

-**Entidad :**Los componentes activos en cada capa se denominan entidades. La entidad puede ser un proceso de software o hardware como un dispositivo inteligente de entrada/salida, etc.

-**Entidad par:**Las entidades en la misma capa que se ejecutan en diferentes máquinas se denominan entidades pares.

-**Proveedor de servicios y usuario del servicio:**En arquitectura en capas, capa *norte*proporciona un servicio que es utilizado por la capa*n+1*. Aquí capa*norte*es proveedor de servicios y capa*n+1*es usuario del servicio.

-**SAP (Puntos de acceso al servicio):**Una capa*norte*ofrece servicios a la capa*n+1* en un lugar que se llama puntos de acceso al servicio. Cada SAP tiene una dirección única para su identificación.

2.7 SERVICIO ORIENTADO A CONEXIÓN Y SIN CONEXIÓN

Una capa puede ofrecer dos tipos diferentes de servicios a la capa inmediatamente superior: servicio orientado a conexión y servicio sin conexión.

En el servicio orientado a la conexión, el usuario del servicio establece una conexión al principio, se comunica a través de la conexión y finalmente libera la conexión. Es similar a un sistema telefónico.

En el servicio sin conexión, no se establece ninguna conexión de antemano. En cambio, como en un sistema postal, cada mensaje lleva su dirección de destino completa y cada uno se enruta de forma independiente a su destino. Si un mensaje grande se divide en pedazos y se envía al mismo destino, entonces, en el servicio sin conexión, en algún momento puede suceder que la primera parte llegue al destino después de las últimas partes. Eso significa que en el extremo receptor el orden de entrega puede no ser el mismo que el orden de transmisión. En el servicio orientado a la conexión, esto nunca sucede.

2.8 EL MODELO DE REFERENCIA DE INTERNET (TCP/IP)

El modelo de arquitectura de red de Internet, también conocido como modelo de referencia TCP/IP, se desarrolló mucho antes que el modelo de referencia OSI. Se desarrolló a partir de la red de investigación del Departamento de Defensa de los EE. UU. (DoD): ARPANET. Finalmente, ARPANET conectó muchas universidades y otras organizaciones gubernamentales. Durante el proceso de interconexión de todos estos a través de las líneas telefónicas existentes, el enlace satelital y la radio, el protocolo utilizado para ARPANET tuvo problemas y, por lo tanto, se necesitaba una nueva arquitectura para superar eso. La arquitectura así surgida se conoció como INTERNET o Modelo de Referencia TCP/IP. El modelo tiene cuatro capas, a saber:

1. La capa de host a red.
2. La capa de Internet.
3. La capa de transporte.
4. La capa de aplicación.

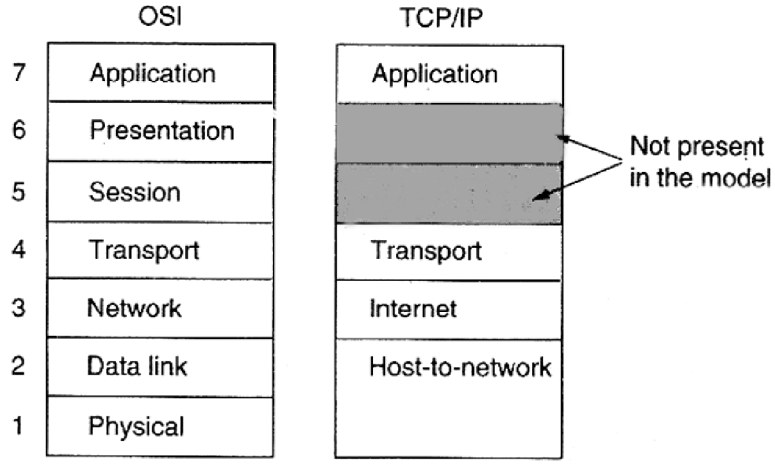


Fig. 2.3 El modelo de referencia TCP/IP

1. **La capa de host a red:**En el modelo de referencia TCP/IP, la capa inferior no está claramente definida. Sin embargo, podemos considerar que lo que sea que esté debajo de la capa de Internet es la capa inferior en el modelo TCP/IP.
   En el modelo, el host debe conectarse a la red mediante algún protocolo para poder enviar paquetes IP a través de él.

2. **La capa de Internet:**El Departamento de Defensa planeó configurar su inter-red de tal manera que tuviera que sobrevivir incluso si un enlace en particular falla en una probable guerra. Por lo tanto, este requisito condujo a una red de conmutación de paquetes sin conexión en lugar de una red de conmutación de circuitos orientada a la conexión. La capa de Internet está diseñada para cumplir con el objetivo de la arquitectura de tal manera que se convirtió en el eje de toda la arquitectura. El trabajo de esta capa es bombear los paquetes desde la máquina host a cualquier red y ayudar a que los paquetes lleguen al destino de forma independiente. En este estilo de comunicación, los paquetes pueden llegar en un orden diferente al que fueron enviados. Esta capa tiene su propio protocolo llamado**IP (Protocolo de Internet)**y un formato de paquete especificado. El enrutamiento de paquetes y la evitación de la congestión son los principales problemas en esta capa. Esta capa es similar a la capa de red en el modelo de referencia OSI.

3. **La capa de transporte:**La capa inmediatamente superior a la capa de Internet se denomina capa de transporte. Esta capa se comunica con la capa del mismo nivel en la otra máquina. Tiene dos protocolos a saber**TCP y UDP.**

   TCP es un protocolo confiable orientado a la conexión que garantiza la entrega del flujo de bytes desde la máquina de origen a la máquina de destino. Esta capa divide el flujo de bytes recibido de la capa superior en mensajes y los pasa a la capa de Internet. En la máquina receptora, el TCP vuelve a ensamblar estos mensajes y los pasa a la capa superior. El control de flujo también es otro trabajo de esta capa para evitar que un remitente rápido inunde a un receptor lento.

   UDP es otro protocolo utilizado por esta capa que ofrece un servicio sin conexión no confiable. Aquí, a diferencia de TCP, no se realiza secuenciación ni control de flujo. Cuando la entrega rápida es más importante que la entrega precisa, se utiliza UDP. También es ampliamente utilizado en clientes de una sola vez.

comunicación tipo servidor. Las transmisiones de correo electrónico, video o sonido son algunas de estas aplicaciones.

4. **La capa de aplicación:**En el modelo OSI, hay capas de sesión y presentación por encima de la capa de transporte. En el modelo TCP/IP estas dos capas están ausentes. Entonces, además de la capa de transporte, la capa de aplicación está presente en el modelo de referencia TCP/IP. Todos los protocolos de alto nivel están presentes aquí. TELNET, FTP, SMTP son algunas de las primeras inclusiones. Posteriormente, se agregan a esta capa otros protocolos como DNS, NNTP, HTTP.

2.9 COMPARACIÓN DEL MODELO DE REFERENCIA ISO-OSI Y TCP/IP

No hay mucha diferencia entre el modelo de referencia OSI y TCP/IP. En ambos se utiliza el concepto de pila independiente de protocolos. La función de las capas también es más o menos la misma. En ambos modelos, comenzando de abajo hacia arriba hasta la capa de transporte, la función de las capas es proporcionar un servicio de transporte independiente de la red de extremo a extremo para los procesos de comunicación. Las capas por encima de la capa de transporte están orientadas a la aplicación.

Por otro lado, hay algunas diferencias entre los dos modelos. En el modelo de referencia OSI, tres conceptos son distintos. Estos son -

1. Servicios,
2. Interfaces,
3. Protocolos.

los*Servicio*dice para qué sirve una capa en particular a la capa que está justo encima de ella, no cómo accede a estos servicios la capa superior o cómo los proporciona la capa. los*Interfaz*dice cómo la capa superior accede a los servicios provistos por la capa justo debajo de ella. Tampoco dice cómo se proporcionan. los*protocolos* utilizado entre las capas del mismo nivel es el conjunto de reglas acordadas por ambas capas para realizar el trabajo. Los protocolos se pueden cambiar sin afectar el software en las capas superiores.

En el modelo TCP/IP, no existe tal distinción entre Servicio*, interfaz y protocolo*. Por lo tanto, en el modelo de referencia TCP/IP los protocolos no son

tan oculto como en el modelo de referencia OSI. A medida que avanza la tecnología, el protocolo entre una capa de pares se puede reemplazar fácilmente en el modelo OSI que en el modelo TCP/IP.

En el modelo OSI, la estructura en capas se pensó antes de que se inventaran los protocolos. Los diseñadores eran nuevos en la tecnología de red y, por lo tanto, en alguna capa se debe proporcionar una subcapa para acomodar algún nuevo modo de comunicación. Por ejemplo, en la capa de enlace de datos, la subcapa MAC se introdujo más tarde cuando la comunicación de transmisión comenzó a abordar los problemas de asignación de canales. En cambio en el modelo TCP/IP, primero venían los protocolos y luego el modelo. Por lo tanto, no hubo problema para ajustar los protocolos al modelo.

Los dos modelos también se diferencian en el número de capas; en el modelo OSI hay siete capas si el modelo TCP/IP tiene cuatro capas. Ambos tienen capas de (inter)red, transporte y aplicación en común, pero las otras capas no son iguales.

En el modelo OSI, la capa de red admite ambos *orientado a la conexión y* comunicación sin conexión y la capa de transporte solo admite *orientado a la conexión* comunicación. Por el contrario, en el modelo TCP/IP la capa de red solo admite *sin conexión* comunicación, pero la capa de transporte admite tanto el modo, es decir *sin conexión* y *orientado a la conexión* comunicación.

CONSULTA TU PROGRESO-2

1. ¿Cuáles son los dos servicios que puede ofrecer una capa a la capa superior?
2. ¿Cuántas capas hay en el modelo de referencia TCP/IP?
3. ¿Qué modelo de arquitectura de red se desarrolló anteriormente: OSI o TCP/IP?
4. ¿Qué dos capas del modelo OSI están ausentes en el modelo TCP/IP?

2.10 RESUMAMOS

-Debe haber alguna técnica de direccionamiento para una comunicación fluida entre dos computadoras.

-Comunicación simplex significa comunicación unidireccional.

-En la comunicación semidúplex, la comunicación es bidireccional alternativamente; No al mismo tiempo.

-La comunicación full-duplex significa comunicación bidireccional en todo momento.

-La mayoría de las redes informáticas están organizadas como una pila de capas, una encima de otra, tomando el servicio que ofrece la capa inferior.

-Protocolo significa un conjunto de reglas seguidas por dos pares durante la comunicación.

-Entidad: Los componentes activos en cada capa se denominan entidades. La entidad puede ser un proceso de software o hardware como un dispositivo inteligente de entrada/salida, etc.

-Entidad de pares: las entidades en la misma capa que se ejecutan en diferentes máquinas se denominan entidades de pares.

-Proveedor de servicios y usuario del servicio: en la arquitectura en capas, la capa*norte* proporciona un servicio que es utilizado por la capa*n+1*. Aquí capa*norte*es proveedor de servicios y capa*n+1*es usuario del servicio.

-SAP (Service Access Points): Una capa*norte*ofrece servicios a la capa*n+1* en un lugar que se llama puntos de acceso al servicio. Cada SAP tiene una dirección única para su identificación.

-Las capas y sus protocolos se conocen colectivamente como arquitectura de red.

-El modelo OSI es una arquitectura de red de siete capas.

-Una capa puede ofrecer dos tipos diferentes de servicios a la capa inmediatamente superior: servicio orientado a conexión y servicio sin conexión.

-En el modelo TCP/IP de arquitectura de red, hay cuatro capas.

-La mayoría de las redes informáticas se basan en el modelo TCP/IP.

2.11 RESPUESTAS PARA COMPROBAR TU PROGRESO

CONSULTA TU PROGRESO-1

1. Una capa realiza algunas operaciones primitivas para cumplir con el requisito de la capa inmediatamente superior. Esta responsabilidad de la capa inferior se llama servicio.

   Por otro lado, la interfaz es la unión entre dos capas adyacentes a través de las cuales la capa inferior proporciona los servicios a la capa superior.

2. Las capas y sus protocolos se conocen colectivamente como red. **arquitectura**.
3. Siete capas.
4. Encuadre, control de flujo, control de errores y control de acceso.

CONSULTA TU PROGRESO-2

1. Servicio orientado a la conexión y servicio sin conexión.
2. Cuatro capas.
3. El modelo de referencia TCP/IP se desarrolló antes que el modelo OSI.
4. Capa de sesión y capa de presentación.

2.12 LECTURAS ADICIONALES

1. Tanenbaum, Andrew, Redes informáticas, PHI.
2. Forouzan Behrouz A., Tata Mcgraw Hill.
3. Norton Peter, Guía completa para la creación de redes.
4. Comer, E. Douglas, Narayanan, MS, Redes informáticas e Internet con aplicaciones de Internet, Pearson Education.

2.13 PREGUNTAS MODELO

1. ¿Cuáles son los problemas de diseño de la arquitectura en capas de una red informática?
2. ¿Cuál es la principal diferencia entre la comunicación sin conexión y la orientada a la conexión?

3. Describir el modelo de referencia ISO-OSI de la red informática.

4. Definir: entidad par, protocolo, pila de protocolos, SAP.

5. ¿Cuál de las capas OSI maneja cada uno de los siguientes:

   a) Descomposición del flujo de bits transmitido en tramas.

   b) Determinar qué ruta a través de la subred usar.

6. Describir las funciones de varias capas del modelo TCP/IP.

7. ¿Cuáles son las responsabilidades de la capa de red en el modelo TCP/IP?

8. ¿Cuáles son las responsabilidades de la capa de transporte en el modelo TCP/IP?

9. ¿Cómo se correlacionan las capas del modelo TCP/IP con las capas del modelo OSI?

10. Haga coincidir lo siguiente con una o más capas del modelo OSI:

    un. Control de flujo

    B. Determinación de ruta

    C. Proporciona acceso para el usuario final.

    D. Interfaz a los medios de transmisión.

UNIDAD - 3 CAPA FÍSICA

ESTRUCTURA DE LA UNIDAD

3.1 OBJETIVOS DE APRENDIZAJE

Después de pasar por esta unidad, podrá:

- aprender sobre el concepto básico de la capa física
- aprender el concepto de señales analógicas y digitales
- describir la transmisión digital y analógica
- aprender sobre multiplexación y diferentes técnicas de multiplexación
- describir diferentes medios de transmisión de señales
- Aprenda sobre la conmutación de circuitos y la red telefónica.

3.2 INTRODUCCIÓN A LA CAPA FÍSICA

En la unidad 2 hemos aprendido acerca de diferentes modelos de red. En esta unidad vamos a hablar sobre la capa física, que es la capa más baja del modelo OSI. Consiste en diferentes medios de transmisión de hardware de red y métodos de transmisión de red informática. La capa física proporciona servicios para la capa de enlace de datos. Aquí se mantiene la transmisión de bits sin procesar a través de cualquier medio de transmisión de hardware. Proporciona una interfaz eléctrica, mecánica y de procedimiento a la red. El principal problema de diseño en esta capa es asegurarse de que si se envía 1 bit desde la fuente, debe recibirse en el otro extremo como 1 bit.

3.3 SEÑALES

Una de las principales funciones de la capa física es mover datos en forma de señales electromagnéticas a través de un medio de transmisión. Tanto los datos como la señal electromagnética pueden ser de forma analógica o digital.

3.3.1 SEÑALES ANALÓGICAS Y DIGITALES

Una señal analógica es una onda electromagnética que varía continuamente para la cual la característica variable en el tiempo de la señal es una representación de alguna otra cantidad variable en el tiempo que puede propagarse a través de una variedad de medios, dependiendo del espectro. Por ejemplo, cuando alguien habla, se crea una onda analógica en el aire. Esto puede ser capturado por un micrófono y convertido a una señal analógica. Una señal analógica tiene infinitos niveles de intensidad durante un período de tiempo.

Por otro lado, una señal digital es una señal física que tiene una secuencia de pulsos de voltaje que pueden transmitirse a través de un medio de transmisión. Los datos se almacenan en la memoria de la computadora en forma de 0 y 1 que se pueden convertir en señales digitales.

Las ventajas de la señalización digital sobre la señalización analógica son:

1. La señalización digital es generalmente más barata que la señalización analógica.
2. La señalización digital se ve menos afectada por la interferencia de ruido que la señalización analógica.

La desventaja de la señalización digital es que las señales digitales sufren más atenuación que las señales analógicas.

Señales periódicas y no periódicas:

Tanto las señales analógicas como las digitales pueden tener forma periódica o no periódica.

Una señal periódica completa un patrón dentro de un marco de tiempo medible, llamado período, y repite ese patrón en períodos idénticos posteriores. La finalización de un patrón completo se denomina ciclo. Una señal no periódica cambia sin exhibir un patrón o ciclo que se repita con el tiempo.

Las señales analógicas periódicas se pueden clasificar en simples o compuestas. Una señal analógica periódica simple, una onda sinusoidal, no se puede descomponer en señales más simples. Una señal analógica periódica compuesta se compone de múltiples ondas sinusoidales.

La forma más fundamental de una señal analógica periódica es la onda sinusoidal. Una onda sinusoidal se puede representar mediante tres parámetros: la amplitud máxima, la frecuencia y la fase.

Ahora, el valor absoluto de la intensidad más alta que es proporcional a la energía de una señal se llama amplitud máxima. En el caso de señales eléctricas, la amplitud máxima se mide en voltios.

La cantidad de tiempo requerida en segundos por una señal para completar 1 ciclo se denomina período y el número de períodos en un segundo se denomina frecuencia. Entonces el período es el inverso de la frecuencia.

La fase se refiere a la posición de la forma de onda relativa al tiempo 0. La fase se mide en grados o radianes. Un cambio de fase de 360° corresponde a un cambio de un período completo y un cambio de fase de 180° corresponde a un cambio de la mitad de un período.

La longitud de onda es otra característica de una señal que viaja a través de un medio de transmisión. La longitud de onda se puede calcular si conocemos la velocidad de propagación y el período de la señal con las siguientes fórmulas.

Longitud de onda = velocidad de propagación x período

Señales compuestas:

Ahora, en la comunicación de datos, requerimos enviar señales compuestas que están hechas de muchas ondas sinusoidales simples porque una onda sinusoidal de una sola frecuencia no es útil en la comunicación de datos. Según el análisis de Fourier, cualquier señal compuesta es una combinación de ondas sinusoidales simples con diferentes frecuencias, amplitudes y fases. Una señal compuesta periódica es una colección de una serie de señales con frecuencias discretas y un

La señal compuesta no periódica es una combinación de ondas sinusoidales con frecuencias continuas.

Ahora, el ancho de banda de una señal compuesta se refiere al rango de frecuencias que contiene. Por ejemplo, si una señal compuesta contiene frecuencias entre 1000 y 7000, su ancho de banda es 7000-1000, es decir, 4000.

Tasa de bits:

El término tasa de bits se usa para señales digitales que da la cantidad de bits enviados en 1 segundo (bits por segundo) a través de un canal de comunicación.

Longitud de bits:

La longitud de bits de una señal digital es la distancia que ocupa un bit en el medio de transmisión, que se puede calcular mediante las siguientes fórmulas.

Longitud de bit = velocidad de propagación X duración de bit

3.3.2 LÍMITES DE VELOCIDAD DE DATOS

En el caso de la comunicación de datos, la rapidez con la que se pueden enviar los datos a través de un canal en bits por segundo se denomina tasa de datos. Ahora bien, hay tres factores de los que depende la tasa de datos:

1. El ancho de banda
2. El nivel de las señales
3. La calidad del canal

En el caso de un canal sin ruido, según Nyquist, la tasa de bits máxima teórica se puede calcular mediante las siguientes fórmulas:

Tasa de bits = 2 registros XBX_2norte

Aquí, B es el ancho de banda del canal, N es el número de niveles de señal y la tasa de bits se calcula en bits por segundo.

Pero en el caso de nuestras comunicaciones de datos del mundo real, no es posible tener un canal sin ruido. Claude Shannon desarrolló una fórmula en 1944 para calcular la tasa de datos teórica más alta para un canal ruidoso. Esta fórmula se llama la capacidad de Shannon.

Capacidad = registro BX_2(1+SNR)

Aquí, B es el ancho de banda del canal, SNR es la relación señal/ruido y la capacidad es la tasa de bits del canal en bits por segundo.

Rendimiento:

El rendimiento es la cantidad real de datos en bits por segundo que se puede enviar a través de un canal de comunicación. Por lo tanto, es diferente del ancho de banda, ya que el ancho de banda da la capacidad total de un canal en caso de transmisión de datos en bits por segundo. En general, el rendimiento siempre es menor que el ancho de banda de un canal de comunicación.

Latencia:

La latencia o retraso es la cantidad de tiempo necesario para que un mensaje completo llegue al destino desde el momento en que se envía el primer bit al origen. La latencia se calcula mediante las siguientes fórmulas:

Latencia = tiempo de propagación + tiempo de transmisión + tiempo de cola + retraso de procesamiento

Tiempo de propagación:

El tiempo de propagación es la cantidad de tiempo necesaria para que un bit viaje desde el origen hasta el destino. El tiempo de propagación se calcula dividiendo la distancia por la velocidad de propagación.

Tiempo de propagación = Distancia / Velocidad de propagación

La velocidad de propagación de las señales electromagnéticas depende de dos factores que son el medio de comunicación y la frecuencia de la señal.

Tiempo de transmisión:

En las comunicaciones de datos, el tiempo necesario para que llegue un mensaje completo del remitente al receptor se denomina tiempo de transmisión. En otras palabras, podemos decir que el tiempo entre el primer bit que sale del remitente y el último bit que llega al receptor de un mensaje se denomina tiempo de transmisión. El tiempo de transmisión de un mensaje depende del tamaño del mensaje y del ancho de banda del canal.

Tiempo de transmisión = Tamaño del mensaje / Ancho de

banda **Tiempo de cola:**

El tiempo de cola es el tiempo requerido para que cada dispositivo intermedio o final retenga el mensaje antes de que pueda ser procesado. El tiempo de cola cambia con la cantidad de carga disponible en la red de comunicación. Un dispositivo intermedio o dispositivo final como un enrutador procesa los mensajes uno por uno en algún orden. Si hay muchos mensajes, cada mensaje tendrá que esperar. Entonces, si aumenta la carga en la red, aumenta el tiempo de espera.

3.3.3 DETERIORO DE LA TRANSMISIÓN

Después de viajar a través de algunos medios de transmisión, las señales originales en la fuente se cambian en el destino. Esto significa que se produce un deterioro de la señal. Hay tres factores para el deterioro de la señal que son la atenuación, la distorsión y el ruido.

Atenuación:

La atenuación significa una pérdida de energía. Cuando se envía una señal de origen a destino a través de un canal o medio de comunicación, pierde parte de su energía debido a la resistencia presente en el medio de comunicación y parte de la energía eléctrica de la señal se convierte en calor. Entonces, esta atenuación provoca un deterioro de la señal. Los amplificadores se utilizan para amplificar las señales y minimizar la atenuación de la señal. El decibelio (dB) es la unidad para definir la fuerza relativa de dos señales o una señal en dos puntos diferentes. El decibelio es negativo si se atenúa una señal y positivo si se amplifica una señal.

$$dB = 10 \log_{10} \frac{S_{pags}}{S_q}$$

Variables S_{pags}y $S_q$son las potencias de una señal en los puntos p y q, respectivamente.

Distorsión:
Cuando una señal cambia su forma o forma original en el momento del movimiento desde la fuente hasta el destino final, se denomina distorsión de la señal. En el caso de una señal compuesta, cada onda sinusoidal tiene su propia velocidad de propagación a través del medio de comunicación y su propio retraso en llegar al destino. Entonces, el retraso de diferentes ondas sinusoidales tiene diferencias, lo que significa que los componentes de la señal en el receptor tienen fases diferentes de los componentes en el remitente. Entonces, la forma de la señal compuesta se cambia en el destino final.

Ruido:

El ruido es una señal no deseada que forma parte de cualquier señal grabada. Así que es otra causa de deterioro. Hay diferentes tipos de ruido, como el ruido térmico, el ruido inducido, la diafonía y el ruido de impulso que pueden corromper la señal original. El movimiento aleatorio de los electrones en un cable crea una señal extra llamada ruido térmico. Los ruidos inducidos se crean a partir de fuentes como motores y electrodomésticos. La diafonía es el efecto de la antena emisora sobre la antena receptora. El ruido de impulso es una señal con alta energía en muy poco tiempo que proviene de algunas fuentes como líneas eléctricas, rayos, etc.

Ahora la relación señal a ruido se define de la siguiente manera:

SNR = potencia de señal promedio / potencia de ruido promedio

SNR se puede describir en unidades de decibelios de la siguiente

manera: $SNR_{dB}=10\log_{10}SNR$

3.4 TRANSMISIÓN DIGITAL

En una red informática, la información debe convertirse en una señal digital o una señal analógica para la transmisión de una máquina a otra. Ahora, la señal digital se puede transmitir utilizando uno de los dos enfoques diferentes, que son la transmisión de banda base y la transmisión de banda ancha.

En la transmisión de banda base, una señal digital se transmite a través de un canal sin convertir la señal digital en una señal analógica. La transmisión de banda base requiere un canal con un ancho de banda que comienza desde cero llamado canal de paso bajo.

En la transmisión de banda ancha, una señal digital se transmite a través de un canal después de convertirla en una señal analógica. La transmisión de banda ancha permite utilizar un canal con un ancho de banda que no parte de cero llamado canal pasabanda.

Ahora, las siguientes técnicas se utilizan para convertir datos digitales en señales digitales.

3.4.1 CODIFICACIÓN DE LÍNEA

La codificación de línea es el proceso de convertir datos digitales en señales digitales. La codificación de línea convierte una secuencia de bits almacenada en la memoria de la computadora en una señal digital. En el extremo del remitente, los datos digitales se codifican en una señal digital y en el extremo del receptor, la señal digital se decodifica en sus datos digitales.

Un elemento de datos es la unidad más pequeña para representar una pieza de información. En las comunicaciones de datos digitales, un elemento de señal transporta elementos de datos. Un elemento de señal es la unidad más corta con respecto al tiempo de una señal digital. Entonces podemos decir que los elementos de datos están siendo transportados y los elementos de señal son los portadores.

El número de elementos de señal enviados en 1 segundo se denomina tasa de señal. La unidad se llama baudios. La tasa de señal también se puede denominar tasa de pulso, tasa de modulación o tasa de baudios.
Entonces, un objetivo principal en la comunicación de datos es aumentar la velocidad de datos, lo que aumenta la velocidad de transmisión, y disminuir las velocidades de señal, lo que disminuye el requisito de ancho de banda.
Ahora bien, hay algunos problemas en la decodificación de una señal digital discutidos a continuación:

1.Al decodificar una señal digital, la potencia de la señal se evalúa frente al promedio móvil de la potencia de la señal recibida denominada línea base para determinar el valor del elemento de datos. Ahora, en el caso de una cadena larga de 0 y 1, puede causar una desviación en la línea de base llamada desviación de la línea de base. Debido a esta desviación de la línea de base, al receptor le resulta difícil decodificar correctamente la señal digital. Se requiere un buen esquema de codificación de línea para evitar la desviación de la línea de base.

2.Cuando el nivel de voltaje en una señal digital es constante por un tiempo, el espectro crea frecuencias muy bajas. Estas frecuencias alrededor de cero se denominan componentes de corriente continua (CC). Ahora bien, estos componentes de CC crean problemas para un sistema que no puede pasar bajas frecuencias o un sistema que utiliza acoplamiento eléctrico. Para estos sistemas, se requiere un esquema sin componentes de CC.

3.Para decodificar correctamente una señal digital, los intervalos de bits del receptor deben ser los mismos que los intervalos de bits del emisor. Si el reloj del receptor es más rápido o más lento que el del emisor, los intervalos de bits no serán los mismos para ellos. Como resultado de esto, el receptor podría malinterpretar las señales. Para resolver este problema, una señal digital incluye información de temporización en los datos transmitidos a través de un medio de comunicación.

4.Los errores ocurridos durante la transmisión de la señal deben detectarse al momento de decodificar las señales digitales. Por lo tanto, se requiere una capacidad integrada de detección de errores en el código generado para detectar algunos o todos los errores.

5.También se deben minimizar el ruido y otras interferencias de las señales digitales.

Esquemas de codificación de línea:
En general, los esquemas de codificación de líneas se dividen en algunas categorías que se analizan como sigue.

Esquema unipolar: En un esquema unipolar, todos los niveles de señal están en un lado del eje del tiempo. En general, un esquema unipolar se diseñó como un esquema sin retorno a cero (NRZ). Aquí, el voltaje positivo define el bit 1 y el voltaje cero define el bit 0. Se llama NRZ porque la señal no vuelve a cero en la mitad del bit. Este esquema no se utiliza en comunicaciones de datos ya que es muy costoso.

esquema polar:
En el caso de esquemas polares, los niveles de señal están a ambos lados del eje del tiempo.
En la codificación polar sin retorno a cero (NRZ), se utilizan dos niveles de amplitud de voltaje. El NRZ polar se divide en dos versiones que son NRZ-Level y NRZ-Invert. En NRZ-Level, el nivel del voltaje determina el valor del bit. Por otro lado, en NRZ-Invert, si hay un cambio en el nivel de voltaje, el bit es 1 y si no hay cambio, el bit es 0. Tanto NRZ-Level como NRZ-Invert tienen una tasa de señal promedio. de N/2 Bd y tiene un problema de componente de CC.

En el caso de la codificación NRZ, cuando los relojes del emisor y del receptor no están sincronizados, el receptor no sabe cuándo finaliza un bit y comienza el siguiente. Entonces, debido a este problema, se puede usar el esquema de retorno a cero (RZ). Este esquema utiliza tres valores que son positivo, negativo y cero. En el esquema RZ, la señal cambia durante el bit.
Las desventajas de la codificación RZ son las siguientes:

1. Ocupa mayor ancho de banda ya que requiere dos cambios de señal para codificar un bit.
2. Utiliza tres niveles de voltaje que son más complejos de crear y discernir.

Debido a estas desventajas, la codificación RZ no se usa en los últimos tiempos.

Esquema Manchester y Manchester Diferencial: T El esquema de Manchester es la combinación de RZ y NRZ-Level. En la codificación Manchester, la duración del bit se divide en dos mitades. Durante la primera mitad, el voltaje permanece en un nivel y se mueve al otro nivel en la segunda mitad. Aquí la sincronización se mantiene mediante la transición en el medio del bit.
En el caso del esquema Manchester diferencial, la idea de RZ y NRZ-Invert se combinan. Aquí los valores de bit se determinan al principio del bit. Si el siguiente bit es 0, hay una transición; de lo contrario, si el siguiente bit es 1, no se produce ninguna transición.
El esquema Manchester resuelve varios problemas que se encuentran en NRZ-Level y, por otro lado, el esquema Manchester diferencial resuelve los problemas asociados con NRZ-Invert. En estos esquemas, no se encuentra deambulación de línea de base. Aquí cada bit tiene una contribución de voltaje positiva y negativa, por lo que tampoco se encuentra ningún componente de CC.
Ahora, la desventaja de Manchester y Manchester diferencial es que aquí la tasa de señal es el doble que para NRZ porque siempre hay una transición en el medio del bit y tal vez una transición al final de cada bit. Manchester y Manchester diferencial también se denominan esquemas bifásicos.

Esquemas bipolares: En el esquema bipolar, hay tres niveles de voltaje que son positivo, negativo y cero. Aquí el nivel de voltaje para un elemento de datos está en cero y el nivel de voltaje para el otro elemento cambia entre positivo y negativo. La ventaja del esquema bipolar es que tiene la misma tasa de señal que NRZ sin componente de CC. La concentración de la energía en la codificación bipolar está alrededor de la frecuencia N/2.

Esquemas Multinivel: El Los esquemas multinivel están diseñados para aumentar el número de bits por baudio mediante la codificación de un patrón de m elementos de datos en un patrón de n elementos de señal. Un grupo de m elementos de datos puede producir una combinación de 2_{metro}los patrones de datos como datos pueden ser 1 o 0. Para L niveles diferentes y n elementos de señal, L_{norte}se producen combinaciones de patrones de señal. ahora si $2_{metro}= L_{norte}$entonces cada patrón de datos se codifica en un patrón de señal y si $2_{metro}<L_{norte}$, los patrones de datos ocupan solo un subconjunto de patrones de señal. En esquemas multinivel, este subconjunto de patrones de señal se puede diseñar de modo que pueda evitar la desviación de la línea de base y detectar errores de transmisión de datos. Otra ventaja de este diseño es que proporciona sincronización. Por otro lado si $2_{metro}> L_{norte}$entonces la codificación de datos no es posible ya que algunos de los patrones de datos no se pueden codificar.
Dos binarios, uno cuaternario (2B1Q) es un esquema multinivel con patrones de datos de tamaño 2 y codifica los patrones de 2 bits como un elemento de señal que pertenece a una señal de cuatro niveles. La tasa de señal promedio de 2B1Q es N/4.
Ocho binario, seis ternario (8B6T) es un esquema multinivel con tres niveles de señal y codifica un patrón de 8 bits como un patrón de 6 elementos de señal. La tasa de señal promedio de este esquema es $\frac{1}{2}\text{XNX}\ \frac{6}{8}$.

La modulación de amplitud de pulso de cinco niveles en cuatro dimensiones (4D-PAM5) es un esquema de codificación multinivel con cinco niveles de voltaje que son -2, -1,0,1 y 2. Aquí los datos se envían a través de cuatro cables al mismo tiempo. Entonces, la tasa de señal en este esquema se puede reducir a N/8.

Transmisión multilínea: El esquema de tres niveles de transmisión multilínea denominado MLT-3 es un esquema de codificación diferencial con tres niveles +V, 0 y –V y tres reglas de transición para moverse entre los niveles. Las tres reglas de transición son las siguientes:

1. Si el siguiente bit es 0, no hay transición.
2. Si el siguiente bit es 1 y el nivel actual no es 0, el siguiente nivel es 0.

3. Si el siguiente bit es 1 y el nivel actual es 0, entonces el siguiente nivel es el opuesto del último nivel distinto de cero.

3.4.2. CODIFICACIÓN DE BLOQUES

La codificación de bloques cambia un bloque de m bits en un bloque de n bits. Aquí n es mayor que m. La codificación de bloques también se denomina técnica de codificación mB/nB. La codificación de bloques se divide en tres pasos de la siguiente manera:

1. En el primer paso, una secuencia de bits se divide en grupos de m bits.
2. En el segundo paso, un grupo de bits m se sustituye por un grupo de bits n.
3. En el último paso, los grupos de n bits se combinan para formar un flujo. La nueva secuencia tiene más bits que los bits originales.

El esquema de codificación de cuatro binarios/cinco binarios (4B/5B) es un esquema de codificación de bloques que se puede utilizar en combinación con NRZ-Invert. En el caso del esquema 4B/5B, la salida de 5 bits reemplaza la entrada de 4 bits con no más de un cero inicial y no más de dos ceros finales. Como resultado de esto, cuando se combinan diferentes grupos para hacer una nueva secuencia, no ocurrirán más de tres 0 consecutivos.

La codificación de ocho binarios/diez binarios (8B/10B) también es un esquema de codificación de bloques en el que un grupo de 8 bits de datos se sustituye por un código de 10 bits. La ventaja de este esquema es que aquí la capacidad de detección de errores es más de 4B/5B. La codificación de bloques 8B/10B es en realidad una combinación de codificación 5B/6B y 3B/4B.

3.4.3 CONVERSIÓN DE ANALÓGICO A DIGITAL

Para la conversión de señales analógicas a datos digitales, hay dos técnicas disponibles que son la modulación de código de pulso y la modulación delta.

Modulación de código de pulso (PCM):

Un codificador PCM tiene tres procesos de la siguiente manera

1. En este primer proceso de PCM, la señal analógica se muestrea cada N segundo, donde N es el intervalo o período muestral. El inverso del intervalo de muestreo se denomina tasa de muestreo o frecuencia de muestreo. El proceso de muestreo también se conoce como modulación de amplitud de pulso. Hay tres métodos de muestreo que son ideales, naturales y planos.

   En caso de muestreo ideal, se muestrean pulsos de la señal analógica. Pero este método es difícil de implementar.

En caso de muestreo natural, se enciende un interruptor de alta velocidad solo durante el breve período de tiempo en que se produce el muestreo.

En el caso del muestreo de superficie plana, se utiliza un circuito para crear muestras de superficie plana.

2. En el segundo proceso, la señal muestreada se cuantifica porque el resultado del muestreo es una serie de pulsos con valores de amplitud entre la amplitud máxima y mínima de la señal que puede ser infinita con valores no integrales entre los dos límites y, por lo tanto, no se puede usar en el proceso de codificación. Ahora los pasos en la cuantificación son los siguientes:

   A. En el primer paso, se estima la amplitud máxima y mínima de la señal analógica original. Ahora dejemos que estas amplitudes sean V_{min} y $V_{máximo}$.

   B. En el segundo paso, el rango de amplitudes se divide en L zonas con altura Δ (delta) para cada zona.

$$\Delta = \frac{V_{máximo} - V_{min}}{L}$$

   C. Ahora los valores cuantificados se asignan de 0 a L-1 al punto medio de cada zona.

   D. En el último paso, el valor de la amplitud de la muestra se aproxima a los valores cuantificados.

   Ahora, en este proceso, la elección de L y el número de niveles de cuantificación dependen del rango de amplitudes de la señal analógica y la cantidad de precisión requerida para recuperar la señal. Si se eligen valores más bajos de L, aumentará el error de cuantificación si hay mucha fluctuación en la señal. Los valores de entrada al proceso de cuantificación son los valores reales y los valores de salida son los valores aproximados.

3. En el último paso, los valores cuantificados se codifican como flujos de bits, lo que significa que cada muestra se puede cambiar a una palabra de código de n bits.

Modulación delta (DM)

Como la técnica PCM es una técnica muy compleja, se puede usar otra técnica llamada modulación delta (DM) para convertir la señal analógica en datos digitales. En DM, la diferencia entre muestras sucesivas se codifica en flujos de datos de n bits. Aquí, la señal analógica se aproxima con una serie de segmentos y cada segmento se compara con la onda analógica original para determinar el aumento o la disminución de la amplitud relativa. Si no hay cambio en la amplitud de la señal de la muestra anterior, la señal modulada permanecerá en el mismo estado 0 o 1 de la muestra anterior.

3.4.4 MODO DE TRANSMISIÓN

El modo de transmisión de datos binarios a través de un medio de comunicación puede ser en serie o en paralelo.

Transmisión en serie:

En el caso de la transmisión en serie, se transmite un bit de datos binarios a la vez, por lo que solo requiere un canal de comunicación.

La ventaja de la transmisión en serie sobre la paralela es que es menos costosa porque requiere solo un canal de comunicación. La desventaja de este modo es que la velocidad de transferencia de datos es menor que la velocidad del modo de transmisión en paralelo.

Ahora se analizan tres formas de transmisión en serie de la siguiente manera:

1. **Transmisión asíncrona:**En la transmisión asíncrona, los datos binarios son recibidos y traducidos por algunos patrones dados. Ahora bien, estos patrones se basan en agrupar el flujo de bits en bytes. En la transmisión asíncrona, los datos transmitidos se codifican con bits de inicio y parada. Aquí se envía un bit de inicio que es 0 al principio y uno o más bits de parada que son 1 se envían al final de cada byte. Puede haber un espacio entre cada byte.

2. **Transmisión síncrona:**En la transmisión síncrona, el flujo de datos binarios se agrupa en algunas tramas que pueden contener varios bytes. Aquí, los datos binarios de cada cuadro se transmiten como una cadena continua de 1 y 0. Ahora es responsabilidad del receptor separar la cadena en bytes o caracteres y reconstruir la información. No hay espacio entre los bits en la transmisión en serie síncrona, pero puede haber espacios desiguales entre diferentes tramas.

3. **Isócrono:**En caso de transmisión isócrona, la llegada de datos binarios se mantiene a una tasa fija. Este tipo de transmisión se usa en el caso de transmisión de audio y video en tiempo real donde no deberían estar disponibles retrasos desiguales entre fotogramas.

Transmisión en paralelo:

En caso de transmisión paralela de datos binarios, se envían n bits de datos a la vez. Entonces, en este modo de transmisión, los datos binarios se organizan en grupos de n bits cada uno. El mecanismo de transmisión en paralelo utiliza n cables para enviar n bits a la vez, de modo que cada bit tenga su propio cable y todos los n bits de un grupo se puedan transmitir con cada tictac del reloj desde el dispositivo de origen hasta el de destino.

La ventaja de la transmisión en paralelo sobre la transmisión en serie es que su velocidad de transferencia de datos es mayor. Pero la desventaja de este modo de transmisión es su costo. Aquí el costo aumenta porque requiere n líneas de comunicación para transmitir el flujo de datos binarios.

3.5 TRANSMISIÓN ANALÓGICA

En la transmisión analógica, la información se convierte en señales analógicas. Ahora, la conversión de digital a analógico es un proceso para convertir datos digitales en una señal analógica de paso de banda y la conversión de analógico a analógico es un proceso para convertir una señal analógica de paso bajo en una señal analógica de paso de banda.

3.5.1 DATOS DIGITALES DE MODULACIÓN

La conversión de digital a analógico se implementa cambiando cualquiera de las tres características de una señal analógica en función de la información de los datos digitales. Estas tres características son amplitud, frecuencia y fase. Hay cuatro mecanismos para modular datos digitales en una señal analógica que son modulación por desplazamiento de amplitud (ASK), modulación por desplazamiento de frecuencia (FSK), modulación por desplazamiento de fase (PSK) y modulación de amplitud en cuadratura (QAM).

Señal portadora: La señal portadora es una señal de alta frecuencia producida por el dispositivo emisor**I**n transmisión analógica. En la transmisión analógica, el dispositivo receptor se sintoniza a la frecuencia de la señal portadora para que la información digital cambie la señal portadora modificando una o más de sus características, que son amplitud, frecuencia o fase.

Modulación por desplazamiento de amplitud (ASK):

En el caso de modulación por desplazamiento de amplitud, la amplitud de la señal portadora varía para crear elementos de señal. Aquí tanto la frecuencia como la fase permanecen constantes. Los dos tipos de ASK se analizan a continuación:

Modulación por desplazamiento de amplitud binaria (BASK): En general, ASK se implementa utilizando solo dos niveles de elementos de señal y se denomina modulación por desplazamiento de amplitud binaria o modulación de encendido y apagado. Aquí, la amplitud máxima de un nivel de señal es 0 y el otro es igual a la amplitud de la señal portadora.

PREGUNTA multinivel: Cuando hay más de dos niveles de elementos de señal disponibles, se puede implementar ASK multinivel.

Frecuencia de modulación por desplazamiento:

En el caso de la modulación por desplazamiento de frecuencia, la frecuencia de la señal portadora varía para representar los elementos de la señal. Aquí, la frecuencia de la señal modulada es constante durante la duración de un elemento de señal y cambia para el siguiente elemento de señal si cambia el elemento de datos. Por otro lado, tanto la amplitud máxima como la fase permanecen constantes para todos los elementos de la señal.

En la modulación por desplazamiento de frecuencia binaria (BFSK), se consideran dos frecuencias portadoras y se utilizan más de dos frecuencias portadoras en la modulación por desplazamiento de frecuencia multinivel (MFSK).

Modulación por cambio de fase:

En el caso de la modulación por desplazamiento de fase, la fase de la señal portadora varía para representar dos o más elementos de señal diferentes. Aquí tanto la amplitud máxima como la frecuencia permanecen constantes.

En la modulación por desplazamiento de fase binaria (BPSK), se utilizan dos elementos de señal. La fase de un elemento de señal es 0_0y la fase del otro elemento de señal es 180_0. La ventaja del PSK binario sobre el ASK binario es que es menos susceptible al ruido. Otra ventaja de PSK sobre FSK es que no requiere dos señales portadoras.

En la modulación por desplazamiento de fase en cuadratura (QPSK), se utilizan dos modulaciones BPSK separadas para aumentar la velocidad en baudios.

Modulación de amplitud en cuadratura (QAM):

En el caso de la modulación de amplitud en cuadratura, se utilizan dos portadoras con diferentes niveles de amplitud para cada portadora. El esquema 4-QAM es un QAM. Un tipo de esquema 4-QAM utiliza una señal NRZ unipolar para modular cada portadora. Aquí el mecanismo utilizado

es lo mismo con ASK (OOK). Otro tipo de esquema 4-QAM usa NRZ polar y es lo mismo con QPSK. Existe otro QAM-4 que utiliza una señal con dos niveles positivos para modular cada una de las dos portadoras.

3.5.2 SEÑALES ANALÓGICAS DE MODULACIÓN

La conversión de analógico a analógico se puede implementar de tres formas: modulación de amplitud (AM), modulación de frecuencia (FM) y modulación de fase (PM).

Amplitud modulada:

En la modulación de amplitud, la amplitud de la señal portadora varía con las amplitudes cambiantes de la señal moduladora. La frecuencia y la fase de la portadora permanecen constantes.

Modulación de frecuencia:

En la modulación de frecuencia, la frecuencia de la señal portadora se modula para que la frecuencia de la señal portadora cambie con el cambio de amplitud de la señal de información. La amplitud máxima y la fase de la señal portadora permanecen constantes.

Modulación de fase:

En la modulación de fase, la amplitud máxima y la frecuencia de la señal portadora permanecen constantes, pero la fase de la señal portadora cambia cuando cambia la amplitud de la señal de información. En FM, el cambio instantáneo en la frecuencia portadora es proporcional a la amplitud de la señal moduladora. Por otro lado, en PM el cambio instantáneo en la frecuencia portadora es proporcional a la derivada de la amplitud de la señal moduladora.

3.5.3 MÓDEM DE TELÉFONO

A**módem**es un dispositivo también llamado modulador-demodulador utilizado para modular una señal portadora analógica para codificar información digital y para demodular una señal portadora para decodificar la información transmitida. Por lo tanto, los módems se pueden usar para transmitir señales analógicas. Un ejemplo de módem es un módem de banda de voz que convierte los datos digitales de una computadora personal en señales eléctricas moduladas en el rango de frecuencia de voz de un canal telefónico. Ahora estas señales pueden transmitirse a través de líneas telefónicas y demodularse por otro módem en el lado del receptor para recuperar los datos digitales.

3.6 MULTIPLEXACIÓN

La multiplexación es el conjunto de técnicas utilizadas para la transmisión simultánea de múltiples señales a través de un solo enlace de datos. Entonces, en un sistema multiplexado, n líneas comparten el ancho de banda de un enlace. Los tres tipos de técnicas de multiplexación se describen a continuación:

3.6.1 MULTIPLEXACIÓN POR DIVISIÓN DE FRECUENCIA

Cuando el ancho de banda de un enlace de datos es mayor que los anchos de banda combinados de las señales que se van a transmitir, se puede utilizar la multiplexación por división de frecuencia para desarrollar un sistema multiplexado. En FDM, las señales generadas por cada dispositivo emisor modulan diferentes señales portadoras y estas señales moduladas luego se combinan en una única señal compuesta. Ahora las señales portadoras están separadas por un ancho de banda suficiente para acomodar la señal modulada. Los canales se pueden separar por tiras de ancho de banda no utilizado llamadas bandas de protección que evitan que las señales se superpongan. En general, FDM se considera una técnica de multiplexación analógica, pero también se puede utilizar para combinar fuentes que envían señales digitales.

El demultiplexor utiliza una serie de filtros para descomponer la señal multiplexada en sus señales componentes y cada señal se pasa a un demodulador que las separa de sus portadoras y las pasa a las líneas de salida.

Las compañías telefónicas tienen señales multiplexadas desde líneas de menor ancho de banda a líneas de mayor ancho de banda para maximizar su eficiencia. En este caso, muchas líneas conmutadas o alquiladas se pueden combinar en menos canales pero más grandes y FDM se usa para líneas analógicas. Aquí se multiplexan 12 canales de voz en una línea de mayor ancho de banda para crear un grupo con un ancho de banda de 48 kHz. Ahora, como máximo se pueden multiplexar cinco grupos para crear una señal compuesta llamada supergrupo con un ancho de banda de 240 kHz y admite hasta 60 canales de voz. De nuevo, se multiplexan 10 supergrupos para crear un grupo maestro con un ancho de banda de 2.40 MHz de ancho de banda y admite hasta 600 canales de voz. Aquí se requieren las bandas de guarda entre los supergrupos, por lo que aumenta el ancho de banda necesario a 2.52 MHz. Ahora, seis grupos maestros se pueden combinar en un grupo jumbo con un ancho de banda de 15.12 MHz. Aquí también las bandas de guardia

entre los grupos maestros son requeridos por lo que aumenta el ancho de banda necesario a 16**.** 984 MHz.

El FDM se utiliza en transmisiones de radio AM y FM y transmisiones de televisión.

3.6.2 MULTIPLEXACIÓN POR DIVISIÓN DE LONGITUD DE ONDA

La multiplexación por división de longitud de onda (WDM) es lo mismo que FDM, pero aquí la multiplexación y la demultiplexación involucran señales ópticas transmitidas a través de canales de fibra óptica. La velocidad de datos de la fibra óptica es más alta que la velocidad de datos del cable de transmisión metálico. Entonces, si se usa un cable de fibra óptica para una sola línea, se desperdiciará el ancho de banda disponible. Por lo tanto, la multiplexación se puede utilizar para combinar varias líneas en una sola. En WDM, múltiples fuentes de luz se combinan en una sola luz en el multiplexor y hacen lo contrario en el demultiplexor. Un prisma maneja fácilmente la combinación y división de fuentes de luz.

Una aplicación de WDM es la red SONET en la que se multiplexan y demultiplexan múltiples líneas de fibra óptica.

Dense WDM (DWDM) es una técnica WDM en la que se multiplexa un gran número de canales espaciando los canales muy cerca unos de otros. Proporciona una mayor eficiencia.

3.6.3 MULTIPLEXACIÓN POR DIVISIÓN DE TIEMPO

La multiplexación por división de tiempo (TDM) es una técnica de multiplexación digital en la que se permite que varias conexiones compartan el alto ancho de banda de un enlace y cada conexión ocupa una parte del tiempo en el enlace. TDM se divide en dos esquemas diferentes que son sincrónicos y estadísticos.

En el caso de TDM síncrono, el flujo de datos de cada conexión de entrada se divide en algunas unidades donde cada unidad de entrada tiene un intervalo de tiempo de entrada. Aquí una unidad puede ser 1 bit, un carácter o un bloque de datos. Cada unidad de entrada se convierte en una unidad de salida y cada unidad de salida tiene un intervalo de tiempo de salida. Ahora, la duración de un intervalo de tiempo de salida es n veces más corta que la duración de un intervalo de tiempo de entrada, donde n es el número de conexiones. Aquí, en este sistema de multiplexación, se forma un marco con una colección de unidades de datos recopiladas de cada conexión de entrada en una ronda de entrada. Ahora, una trama se divide en n intervalos de tiempo, donde n es el número de conexiones y se asigna un intervalo para cada unidad. Entonces, un marco consiste en un ciclo completo de intervalos de tiempo donde se dedica un intervalo a cada dispositivo de envío.

En TDM síncrono, la velocidad de datos del enlace es n veces más rápida y la duración de la unidad es n veces más corta, donde n es el número de conexiones.

En la multiplexación por división de tiempo estadística, los intervalos de tiempo se asignan dinámicamente para mejorar la eficiencia del ancho de banda. Entonces, en este sistema, la cantidad de ranuras en cada cuadro es menor que la cantidad de líneas de entrada. Aquí, el multiplexor verifica cada línea de entrada y asigna una ranura para una línea de entrada si la línea tiene datos para enviar, pero si la línea de entrada no tiene datos para enviar, salta la línea y verifica la siguiente línea.

En TDM estadístico, una ranura debe transportar datos y la dirección del destino, por lo que la relación entre el tamaño de los datos y el tamaño de la dirección debe tener algún límite para que la transmisión sea eficiente. No es necesario sincronizar las tramas en TDM estadístico, por lo que no requiere bits de sincronización.

En el caso de TDM, si las tasas de datos de entrada no son las mismas, se produce un problema y, en esta situación, se pueden utilizar tres estrategias o una combinación de ellas, que son la multiplexación multinivel, la asignación de múltiples ranuras y el relleno de pulsos.

Multiplexación multiniveles una técnica utilizada cuando la tasa de datos de una línea de entrada es un múltiplo de otras.

La asignación de ranuras múltiples es una técnica en la quese asigna más de una ranura en un marco a una sola línea de entrada.

Relleno de pulso: Cuando las tasas de bits de las fuentes no son múltiples enteros entre sí, no se puede utilizar ninguna de las dos técnicas anteriores. Entonces, en esta situación, se usa relleno de pulso donde la velocidad de datos de entrada más alta es la velocidad de datos dominante y se agregan bits ficticios a las líneas de entrada con velocidades más bajas.

En el caso de TDM, se debe mantener la sincronización entre el multiplexor y el demultiplexor. Si el multiplexor y el demultiplexor no están sincronizados, es posible que el canal equivocado reciba un bit. Entonces, para sincronizar el multiplexor y el demultiplexor, generalmente se agregan uno o más bits de sincronización al comienzo de cada trama, que se denominan bits de trama. Estos bits de trama siguen un patrón de trama a trama que permite que el demultiplexor se sincronice con el flujo de datos entrante.

3.7 MEDIOS DE TRANSMISIÓN

Un medio de transmisión se puede definir como cualquier tipo de sustancia material como sólido, líquido, gas o plasma que puede transportar cualquier señal o información desde una fuente hasta un destino. En el caso de las ondas electromagnéticas como la luz y las ondas de radio, se utiliza el vacío como medio de transmisión. Los diferentes tipos de medios de transmisión se describen a continuación:

3.7.1 MEDIOS GUIADOS

En el caso de los medios guiados, la transmisión de datos se mantiene a lo largo de una ruta física de un dispositivo a otro. Ejemplos de medios guiados son el cable de par trenzado, el cable coaxial y el cable de fibra óptica. La capacidad de un medio de transmisión guiado en términos de velocidad de datos o ancho de banda depende de la distancia desde la fuente hasta el receptor.

Medios magnéticos:

Se pueden utilizar cintas magnéticas o cualquier medio extraíble, como CD y DVD grabables, para transportar datos de una computadora a otra. Aquí, los datos se escriben en los medios magnéticos y se transportan físicamente a la máquina de destino donde los datos se vuelven a leer. La ventaja de los medios magnéticos es que se puede lograr un gran ancho de banda.

Cable de par trenzado:

Un cable de par trenzado consta de dos hilos de cobre aislados trenzados entre sí. Uno de los cables se usa para llevar señales al receptor y el otro se usa solo como referencia de tierra. En general, varios de estos pares se agrupan en un cable envolviéndolos en una funda protectora. El cable de par trenzado acepta y transporta señales en forma de corriente eléctrica.

En el cable de par trenzado, la interferencia y la diafonía pueden afectar a ambos cables y crear señales no deseadas. Si los dos cables son paralelos, el efecto de estas señales no deseadas no es el mismo en ambos cables porque están en ubicaciones diferentes en relación con las fuentes de ruido o diafonía. Entonces, al torcer los pares, el efecto de las señales no deseadas en ambos cables puede ser el mismo y, como resultado, se eliminan la mayoría de las señales no deseadas.

Hay dos tipos de cable de par trenzado que se utilizan en la comunicación de datos, denominados par trenzado sin blindaje (UTP) y par trenzado blindado (STP).

El par trenzado sin blindaje (UTP) es un cable telefónico ordinario. Este es el menos costoso de todos los medios de transmisión utilizados en las redes de área local. La desventaja de UTP es que el par de cables puede verse más afectado por la interferencia electromagnética externa, la interferencia del par trenzado cercano y el ruido ambiental. El conector UTP más común es el RJ45 (RJ significa conector registrado).

En el cable STP, el par trenzado está blindado con una trenza metálica que reduce las interferencias. Pero es más voluminoso y más caro.

Los cables de par trenzado sin blindaje se dividen en siete categorías que se detallan a continuación:

- Categoría 1: Se utiliza en red telefónica. Su velocidad de datos es inferior a 0,1 Mbps.
- Categoría 2: Se utiliza en líneas T. Su tasa de datos es de 2 Mbps.
- Categoría 3: Se utiliza en LANs. Su velocidad de datos es de 10 Mbps.
- Categoría 4: Se utiliza en redes Token Ring. Su tasa de datos es de 20 Mbps.
- Categoría 5: Se utiliza en LANs. Su velocidad de datos es de 100 Mbps.
- Categoría 5E: Es una extensión de la categoría 5. También se usa en LAN. Su velocidad de datos es de 125 Mbps.
- Categoría 6: Se utiliza en LANs. Su tasa de datos es de 200 Mbps.
- Categoría 7: También se denomina par trenzado con pantalla blindada. También se utiliza en LAN. Su tasa de datos es de 600 Mbps.

Cable coaxial:

El cable coaxial consta de un conductor cilíndrico exterior hueco que rodea un conductor de alambre interior único. El conductor interno es el conductor del núcleo central de un cable sólido o trenzado que suele ser un cobre sólido rodeado por una capa aislante y todo encerrado por un escudo. El conductor exterior está cubierto con una chaqueta o blindaje. Este blindaje metálico exterior protege al conductor interior del ruido. Todo el cable coaxial está protegido por una cubierta de plástico. Un solo cable coaxial tiene un diámetro de 1 a 2,5 cm. Los cables coaxiales transportan señales de rangos de frecuencia más altos que los cables de par trenzado. La atenuación es mucho mayor en los cables coaxiales que en los cables de par trenzado. Entonces, en los cables coaxiales, la señal se debilita rápidamente y requiere el uso frecuente de repetidores.

Los cables coaxiales se clasifican según sus clasificaciones de radiogobierno (RG). Entonces, las diferentes categorías son RG-11, RG-58 y RG-59.

El cable coaxial se usó en redes telefónicas analógicas donde una sola red coaxial puede soportar 10,000 señales de voz y también se usó en redes telefónicas digitales donde un solo cable coaxial podía transportar datos digitales hasta 600 Mbps. Pero hoy, el cable de fibra óptica reemplaza al cable coaxial en la mayor parte de las redes telefónicas.

Los cables coaxiales se utilizaron en las redes tradicionales de televisión por cable. Pero ahora, los proveedores de televisión por cable reemplazaron la mayoría de los medios con cables de fibra óptica. Los cables coaxiales se utilizan solo en los límites de la red en redes híbridas. El cable coaxial RG-59 se utiliza en la televisión por cable.

Los cables coaxiales también se utilizan en las LAN Ethernet tradicionales. El cable coaxial RG-58 con conector Bayone-Neill-Concelman(BNC) se utiliza en 10Base-2 o Thin Ethernet para transmitir datos a 10 Mbps con un alcance de 185 m. El cable coaxial RG-11 se utiliza en el 10Base5 o Ethernet Grueso para transmitir 10Mbps con un alcance de 5000 m.

Cable de fibra óptica:

Un cable de fibra óptica está construido con vidrio y plástico para transmitir señales en forma de luz. El centro del cable es el núcleo de vidrio a través del cual se propaga la luz. En las fibras multimodo, el núcleo suele tener un diámetro de 50 micras y en las fibras monomodo, el núcleo tiene entre 8 y 10 micras. Un revestimiento de vidrio con un índice de refracción más bajo que el

El núcleo rodea al núcleo y se usa una fina cubierta de plástico para proteger el revestimiento. En general, las fibras se agrupan en haces que están protegidos por una cubierta exterior. El ancho de banda práctico de este medio es de unos 10 Gbps, pero su ancho de banda alcanzable es de más de 50 000 Gbps. En el laboratorio, se han logrado 100 Gbps en un solo cable de fibra óptica.

Ahora la fuente de luz, el medio de transmisión y el detector son los componentes más importantes en el sistema de transmisión de fibra óptica. Aquí un pulso de luz significa un bit 1 y la ausencia de luz significa un bit 0. Las fuentes de luz en los cables de fibra óptica son diodos emisores de luz (LED) y amplificación de luz por radiación de emisión estimulada (láseres). El medio de transmisión es una fibra de vidrio ultrafina. El detector genera un pulso eléctrico cuando la luz incide sobre él. Entonces, se desarrolla un sistema de transmisión de datos unidireccional conectando una fuente de luz a un extremo de una fibra óptica y un detector al otro extremo. Ahora bien, este sistema de transmisión acepta una señal eléctrica y la convierte en pulsos de luz para transmitirla a través del medio. En el extremo receptor, los pulsos de luz se reconvierten en una señal eléctrica.

Ahora, en este sistema de transmisión óptica, la luz puede propagarse a través del medio durante muchos kilómetros prácticamente sin pérdida alguna debido a un principio de la luz. Este principio es cuando un rayo de luz pasa de un medio a otro medio, entonces el rayo de luz se refracta en el límite de los dos medios. Ahora la cantidad de refracción depende de las propiedades de los dos medios. De acuerdo con el principio de la luz, si los ángulos de incidencia están por encima de cierto valor crítico, la luz se refracta de regreso al primer medio y nada escapa al segundo medio. Entonces, en el caso de un cable de fibra óptica, un rayo de luz incide en el ángulo crítico o por encima de él y queda atrapado dentro de la fibra.

Modos de propagación: Hay dos tipos de modos de propagación de la luz a lo largo de los canales ópticos disponibles en la tecnología actual. Estos son la propagación multimodo y monomodo.

En el caso de multimodo, múltiples haces de luz de una fuente de luz se propagan a través del núcleo en diferentes caminos. El multimodo se puede implementar de dos formas, que son el índice escalonado y el índice graduado.

En la fibra multimodo de índice escalonado, la densidad del núcleo permanece constante desde el centro hasta los bordes. Aquí la luz se mueve a través de esta densidad constante en línea recta hacia el

interfaz del núcleo y el revestimiento. En la interfaz, el ángulo del movimiento del haz de luz cambia debido a la menor densidad y, como resultado, se produce la distorsión de la señal a medida que pasa a través de la fibra.

En el caso de fibra de índice graduado multimodo, la densidad en el centro del núcleo es más alta y disminuye gradualmente hacia el borde. Entonces la densidad en el borde es la más baja. Debido a estas densidades variables, se reduce la distorsión de la señal a través del cable.

En el caso de monomodo, se utiliza fibra de índice escalonado y una fuente de luz que limita los haces a un pequeño rango de ángulos. El diámetro de la fibra monomodo es más pequeño que el de la fibra multimodo y, por lo tanto, tiene una densidad más baja. Como resultado de esta menor densidad, el ángulo crítico se acerca lo suficiente a 90^0para hacer que la propagación de los haces sea casi horizontal, por lo que aquí la propagación de diferentes haces es casi idéntica. En este caso, todos los haces llegan al destino al mismo tiempo, por lo que la distorsión de la señal es muy inferior.

Ventajas de los cables de fibra óptica:

1. El ancho de banda del cable de fibra óptica es mucho mayor que el del cable de par trenzado o coaxial.
2. La distancia de transmisión de fibra óptica es mucho mayor que la de otros medios guiados. Una señal puede funcionar durante 50 km sin necesidad de regeneración. Por otro lado, se requieren repetidores cada 5 km para cable coaxial o par trenzado.
3. Los cables de fibra óptica no se ven afectados por el ruido electromagnético ni por fallas de energía.
4. El vidrio es más resistente a los materiales corrosivos que el cobre. Por lo tanto, los cables de fibra óptica no se ven afectados por los productos químicos corrosivos del aire.
5. Los cables de fibra óptica son mucho más livianos que los cables de cobre. Mil cables de cobre de pares trenzados de 1 km de largo pesan 8000 kg pero dos fibras tienen más capacidad y pesan solo 100 kg.
6. Los cables de fibra óptica no pierden luz y son tan difíciles de tocar. Por otro lado, los cables de cobre crean efectos de antena que se pueden interceptar fácilmente.

Desventajas de los cables de fibra óptica:

1. El cable de fibra óptica es una tecnología relativamente nueva. Por lo tanto, todos los ingenieros no tienen las habilidades requeridas para su instalación y mantenimiento.
2. Dado que la propagación de la luz es unidireccional, para la comunicación bidireccional se requieren dos fibras.
3. El cable de fibra óptica y las interfaces son más caras que las de otros medios guiados.

Las fibras ópticas se clasifican por la relación entre el diámetro de su núcleo y el diámetro de su revestimiento y ambos diámetros se miden en micrómetros. Los diferentes tipos de cables de fibra óptica se dan a continuación:

1. 50/125: Es un cable de fibra óptica con un diámetro de núcleo de 50,0 µm y un diámetro de revestimiento de 125 µm. Aquí el modo de propagación es multimodo.
2. 62.5/125: Es un cable de fibra óptica con un diámetro de núcleo de 62.5 µm y el diámetro del revestimiento es de 125 µm. Aquí el modo de propagación es multimodo.
3. 100/125: Es un cable de fibra óptica con un diámetro de núcleo de 100.0 µm y el diámetro del revestimiento es de 125 µm. Aquí el modo de propagación es multimodo.
4. 7/125: Es un cable de fibra óptica con un diámetro de núcleo de 7,0 µm y un diámetro de revestimiento de 125 µm. Aquí el modo de propagación es monomodo.

Hay tres tipos de conectores disponibles para cables de fibra óptica.

1. El conector del canal de abonado (SC): se utiliza para la televisión por cable. Utiliza un sistema de bloqueo push/pull.
2. El conector de punta recta (ST): se utiliza para conectar el cable a los dispositivos de red. Utiliza un sistema de cierre de bayoneta. Es más fiable que SC.
3. MT-RJ es un conector utilizado en conexiones multimodo dúplex.

Usos de los cables de fibra óptica:

1. Algunas empresas de televisión por cable crean una red híbrida al combinar fibra óptica y cable coaxial.
2. Las redes de área local como la red 100Base-FX (Fast Ethernet) y 1000Base-X utilizan cable de fibra óptica.

3.7.2 MEDIOS NO GUIADOS

La comunicación inalámbrica se implementa utilizando medios no guiados. Los medios no guiados no requieren ningún conductor físico para transportar ondas electromagnéticas. En este tipo de comunicación, las señales normalmente se emiten a través del espacio libre, por lo que cualquier persona puede recibir estas señales con un dispositivo capaz de hacerlo. Las señales no guiadas pueden moverse desde la fuente hasta el destino de tres maneras: propagación terrestre, propagación aérea y propagación de línea de visión.

En caso de propagación terrestre, las señales no guiadas se mueven a través de la parte más baja de la atmósfera. Por ejemplo, las ondas de radio de baja frecuencia se mueven en todas las direcciones desde la antena transmisora. La cantidad de distancia cubierta por estas señales depende de la cantidad de potencia en la señal.

En el caso de la propagación por el cielo, las ondas de radio de mayor frecuencia se irradian hacia la ionosfera y luego se reflejan de regreso a la tierra. En este caso, las señales no guiadas cubren mayores distancias con menor potencia de salida.

En caso de propagación con visibilidad directa, las señales de muy alta frecuencia viajan en línea recta directamente de antena a antena. Una antena se puede definir como un conductor eléctrico o un grupo de conductores utilizados para emitir ondas electromagnéticas o para recibir ondas electromagnéticas. En la propagación con visibilidad directa, las antenas deben ser direccionales y lo suficientemente altas o lo suficientemente juntas para que no se vean afectadas por la curvatura de la tierra.

Ahora la transmisión inalámbrica se divide en tres grupos que son las ondas de radio, las microondas y las ondas infrarrojas.

Ondas de radio:

Las ondas electromagnéticas que tienen frecuencias entre 3 kHz y 1 GHz se denominan ondas de radio. Las ondas de radio viajan en todas las direcciones desde la fuente, por lo que no es necesario que el transmisor y el receptor tengan una alineación física específica. Las ondas de radio son fáciles de generar, pueden viajar largas distancias y pueden penetrar fácilmente en los edificios. Las ondas de radio de baja frecuencia pueden pasar fácilmente a través de obstáculos, pero la potencia cae bruscamente a medida que aumenta la distancia desde la fuente. Las ondas de radio de alta frecuencia viajan en línea recta y rebotan en los obstáculos. También son absorbidos por la lluvia. Todas las ondas de radio se ven afectadas por la interferencia de motores y otros equipos eléctricos. Ahora se analizan diferentes bandas de ondas de radio de la siguiente manera:

- Frecuencia muy baja (VLF): El rango de frecuencia de esta banda es de 3 kHz a 30 kHz. Sigue la propagación terrestre. Se utiliza en la radionavegación de largo alcance.
- Baja frecuencia (LF): El rango de frecuencia de esta banda es de 30 kHz a 300 kHz. Sigue la propagación terrestre. Se utiliza en radiobalizas y localizadores de navegación.
- Frecuencia media (MF): El rango de frecuencia de esta banda es de 300 kHz a 3 MHz. Sigue la propagación del cielo. La radiodifusión AM utiliza la banda MF.
- Alta frecuencia (HF): El rango de frecuencia de esta banda es de 3 MHz a 30 MHz. Sigue la propagación del cielo. Se utiliza en la comunicación de barcos y aviones.
- Muy Alta Frecuencia (VHF): El rango de frecuencia de esta banda es de 30 MHz a 300 MHz. Sigue la propagación del cielo y la línea de visión. Se utiliza en TV VHF y radio FM.

Microondas:

Las ondas electromagnéticas con frecuencias entre 1 y 300 GHz se denominan microondas. Las microondas son unidireccionales. Para transmitir microondas, las antenas de envío y recepción deben estar alineadas. Aquí se puede alinear un par de antenas sin interferir con otro par de antenas alineadas. Las microondas siguen la línea de propagación visual. Dado que las microondas viajan en línea recta, si las torres con las antenas montadas están demasiado separadas, estas torres deben ser muy altas y se necesitan repetidores periódicamente. Para torres de 100 metros de altura, los repetidores se pueden espaciar a 80 km de distancia. Las microondas no atraviesan las paredes.

La comunicación por microondas es muy utilizada para comunicaciones telefónicas de larga distancia, teléfonos móviles, distribución de televisión, etc.

Las diferentes bandas de microondas se dan de la siguiente manera:

- Frecuencia ultra alta (UHF): El rango de frecuencia de esta banda es de 300 MHz a 3 GHz. Se utiliza en TV UHF, teléfonos móviles, satélite, etc.
- Superhigh Frequency (SHF): El rango de frecuencia de esta banda es de 3 GHz a 30 GHz. Se utiliza en la comunicación por satélite.
- Frecuencia extremadamente alta (EHF): el rango de frecuencia de esta banda es de 30 GHz a 300 GHz. Se utiliza en radar y satélite.

Infrarrojo:

Las ondas infrarrojas y milimétricas no guiadas se utilizan para la comunicación de corto alcance. Por ejemplo, estas ondas se utilizan en los controles remotos de televisión. El rango de frecuencia de estas ondas es de 300 GHz a 400 THz. La ventaja de estas olas es que son relativamente direccionales, baratas y fáciles de construir. Pero la desventaja de estas ondas es que no atraviesan objetos sólidos. Por lo tanto, no se necesita una licencia del gobierno para operar un sistema de infrarrojos. La comunicación por infrarrojos se puede utilizar para conectar computadoras portátiles e impresoras.

Transmisión de ondas de luz

Las señales ópticas no guiadas también se utilizan en la transmisión de datos. Los láseres se pueden utilizar para conectar las LAN en dos edificios. En esta aplicación, los láseres se montan en los techos de los dos edificios. Ahora la señalización óptica mediante láseres es unidireccional, por lo que cada edificio necesita su propio láser y su propio fotodetector. La ventaja de este tipo de transmisión es que se puede lograr un ancho de banda muy alto a un costo muy bajo. También es relativamente fácil de instalar y no requiere ninguna licencia para usarlo.

La desventaja de este tipo de transmisión de datos es que los rayos láser no pueden penetrar la lluvia o la niebla espesa y se ven afectados por el calor del sol.

3.8 CAMBIO DE CIRCUITO

Una red de conmutación de circuitos consta de un conjunto de conmutadores conectados por enlaces físicos en los que cada enlace se divide en n canales. Una conexión entre dos estaciones es una ruta dedicada hecha de uno o más enlaces. Cada enlace normalmente se divide en n canales utilizando FDM o TDM.

La comunicación en una red de conmutación de circuitos se logra en tres fases que son el establecimiento de la conexión, la transferencia de datos y el desmantelamiento de la conexión.

En la fase de establecimiento de la conexión, se establece un circuito dedicado entre las dos partes que intentan comunicarse antes de comenzar la comunicación. Entonces, la configuración de la conexión significa crear canales dedicados entre los conmutadores.

En la fase de transferencia de datos, los datos se transfieren desde el origen hasta el destino después del establecimiento del circuito dedicado.

En la fase de desmontaje, se envía una señal a cada conmutador para liberar los recursos cuando una de las partes requiere desconectarse.

Las redes de conmutación de circuitos no son muy eficientes porque aquí los recursos pueden no estar disponibles para otras conexiones durante mucho tiempo. Pero la ventaja de una red de conmutación de circuitos es que el retraso en este tipo de red es mínimo. Durante la transferencia de datos, los datos no se retrasan en cada conmutador, ya que los recursos se asignan durante la duración de la conexión. Aquí el retraso total se debe al tiempo requerido para establecer la conexión, transferir datos y desconectar el circuito. Ahora, el retraso causado por la configuración es la suma de cuatro partes, que son el tiempo de propagación de la solicitud de la computadora de origen, el tiempo de transferencia de la señal de solicitud, el tiempo de propagación del reconocimiento desde la computadora de destino y el tiempo de transferencia de la señal del reconocimiento.

3.9 RED TELEFÓNICA

La red telefónica se diseñó a fines de 1800. El sistema telefónico simple y antiguo (POTS, por sus siglas en inglés) era un sistema analógico que utilizaba señales analógicas para transmitir voz. Las redes telefónicas utilizan conmutación de circuitos. En los últimos tiempos, la red telefónica ha cambiado técnicamente en muchas áreas. Ahora es tanto digital como analógico.

La red telefónica consta de tres componentes principales que son bucles locales, troncales y oficinas de conmutación.

Bucles locales:

El bucle local es un cable de par trenzado que conecta el teléfono del suscriptor con la oficina final más cercana o la oficina central local. El ancho de banda del bucle local utilizado para voz es de 4000 Hz. Los primeros tres dígitos de un número de teléfono local definen la oficina y los siguientes cuatro dígitos proporcionan el número de bucle local.

Bañador:

Los troncales son los medios de transmisión como fibras ópticas o satélite que manejan la comunicación entre oficinas. Un troncal maneja cientos o miles de conexiones mediante multiplexación.

Oficinas de conmutación:

La red telefónica tiene varios niveles de oficinas de conmutación como oficinas finales, oficinas en tándem y oficinas regionales. La compañía telefónica dispone de conmutadores ubicados en una centralita en la que un conmutador conecta varios bucles locales o troncales que permiten la conexión entre diferentes abonados.

Las compañías telefónicas brindan dos tipos de servicios que son analógicos y digitales.

Servicios Analógicos:

Hay dos tipos de servicios analógicos discutidos a continuación:

1.El servicio analógico conmutado es la primera categoría de los servicios analógicos. Es un servicio de acceso telefónico. La señal en un bucle local es analógica y el ancho de banda suele estar entre 0 y 4000 Hz. En general, se proporciona un servicio de llamadas locales por una tarifa plana mensual.

El servicio 800 es un servicio en el que un suscriptor puede proporcionar conexiones gratuitas para otros suscriptores. En este caso, la llamada es gratuita para el llamante, pero la paga el destinatario. Aquí la tarifa es más económica que la de una llamada normal de larga distancia.

El servicio telefónico de área amplia (WATS) es un servicio en el que los suscriptores pagan las llamadas salientes. Este servicio es menos costoso que las llamadas interurbanas regulares. Aquí los cargos se basan en el tipo de número de llamadas de las llamadas salientes. Aquí hay tres tipos de llamadas salientes disponibles que son llamadas salientes al mismo estado, llamadas salientes a varios estados y llamadas salientes a todo el país.

Los servicios 900 son un servicio en el que la llamada la paga la persona que llama y normalmente es mucho más cara que una llamada normal de larga distancia porque el operador cobra dos tarifas. El primero es el peaje de larga distancia y el segundo es la tarifa que se paga al destinatario por cada llamada.

2. Servicio analógico arrendado la segunda categoría de los servicios analógicos. Debido a este servicio, los clientes pueden tener una línea dedicada, también llamada línea arrendada, que está permanentemente conectada a otro cliente.

Servicios digitales: En el caso de la red telefónica, los servicios digitales se ven menos afectados por el ruido y otras formas de interferencia que los servicios analógicos. Los dos tipos más comunes de servicios digitales son el servicio conmutado/56 y el servicio de datos digitales (DDS).

El Servicio Switched/56 es un servicio digital conmutado que permite velocidades de datos de hasta 56 kbps y para comunicarse a través de este servicio, ambas partes deben suscribirse. La línea en un servicio conmutado/56 es digital, por lo que los suscriptores no necesitan módems para transmitir datos digitales. Pero en este caso, los suscriptores requieren otro dispositivo llamado unidad de servicio digital (DSU).

El servicio de datos digitales es la versión digital de una línea arrendada analógica con una velocidad de datos máxima de 64 kbps.

REVISA TU PROGRESO

1. Preguntas de opción múltiple:

   (I) Según Nyquist, la tasa de bits máxima teórica se puede calcular mediante las siguientes fórmulas:

   A. Tasa de bits = 2 registros XBX2norte

   B. Capacidad = registro BX2(1+SNR)

   C. Tasa de bits = registro BX2norte

   D. Ninguna de las anteriores

   (II) ¿Cuál de las siguientes es la causa del deterioro de la señal?

   A. Atenuación

   B. Distorsión

   C ruido

   Todo lo anterior

   (III) ¿Cuál no es una técnica de multiplexación?

   A. Multiplexación por división de frecuencia

   B. Multiplexación por división de longitud de onda

   C. Multiplexación por división de amplitud

   D. Multiplexación por división de tiempo

(IV) ¿Cuál es un medio de transmisión no guiado?

A. Cable de fibra óptica

B. Onda de radio

C Microondas

D. Tanto B como C

(V) ¿Cuál no es una forma de transmisión serial?

A. Transmisión síncrona

B. Transmisión asíncrona

C. Transmisión isócrona

D. Transmisión simétrica

(VI) ¿Cuál no es un componente de la red telefónica?

Un tronco

B. Módem

C. Bucles locales

D. Cambio de oficinas

(VII) ¿Qué tipo de ondas se utilizan en los mandos a distancia de la televisión?

A. Infrarrojos

B. Onda de luz

C Microondas

D. Tanto A como C

(VIII) El rango de frecuencia de EHF es_____.

A. 30 GHz a 300 GHz

B. 3 GHz a 300 GHz

C. 30 MHz a 300 MHz

D. 3 MHz a 300 MHz

(IX) Las ondas electromagnéticas que oscilan en frecuencias entre 3 kHz y 1 GHz se denominan_____.

A. Microondas

B. Ondas de radio

C infrarrojos

D. Onda de luz

(X) ¿Cuál no es una categoría de cable coaxial?

A. RG-10

BRG-11

C. RG-58

D. RG-59

2. Complete los espacios en blanco:

I. La señal _____ es una combinación de ondas sinusoidales simples con diferentes frecuencias, amplitudes y fases.

II. El tiempo _____ es la cantidad de tiempo que requiere un bit para viajar desde el origen hasta el destino.

tercero La codificación de línea es el proceso de convertir _____ en señales digitales

IV. La NRZ polar se divide en dos versiones: _____ y _____.

V. La modulación delta se puede utilizar para convertir la señal analógica en _____.

VI. Un _____ es un dispositivo también llamado modulador-demodulador.

VIII. _____ es el conjunto de técnicas utilizadas para la transmisión simultánea de múltiples señales a través de un solo enlace de datos.

VIII. _____ medios no requiere ningún conductor físico para transportar ondas electromagnéticas.

IX. Las ondas de radio tienen frecuencias entre _____ y _____.

X. La comunicación en una red de conmutación de circuitos se logra en tres fases: _____, _____ y _____.

3. Indique si las siguientes afirmaciones son verdaderas o falsas

I. El cable coaxial consta de un conductor cilíndrico exterior hueco.

II. Los cables de fibra óptica se ven afectados por el ruido electromagnético.

tercero El número de elementos de señal enviados en 1 segundo se denomina tasa de señal.

IV. Cuando una señal cambia su forma o forma original en el momento del movimiento desde la fuente hasta el destino final, se denomina atenuación de la señal.

V. Una señal analógica es una onda electromagnética que varía continuamente.

VI. La señal digital es una señal física que tiene una secuencia de pulsos de voltaje que pueden transmitirse a través de un medio de transmisión.

VIII. La cantidad de tiempo requerida en segundos por una señal para completar 1 ciclo se llama fase.

VIII. En la transmisión en serie, solo se requiere un canal de comunicación.

IX. En caso de transmisión isócrona, la llegada de datos binarios se mantiene a una tasa fija.

X. La velocidad de datos del cable de par trenzado con pantalla blindada es de 200 Mbps.

3.10 RESUMAMOS

El resumen de esta unidad es el siguiente:

- La capa física proporciona la transmisión de bits sin procesar a través de cualquier medio de transmisión de hardware.
- Tanto los datos como la señal electromagnética pueden ser de forma analógica o digital.
- Una señal analógica es una onda electromagnética que varía continuamente para la cual la característica variable en el tiempo de la señal es una representación de alguna otra cantidad variable en el tiempo que puede propagarse a través de una variedad de medios, dependiendo del espectro.
- Una señal digital es una señal física que tiene una secuencia de pulsos de voltaje que pueden transmitirse a través de un medio de transmisión.
- Tanto las señales analógicas como las digitales pueden tener forma periódica o no periódica.
- Una señal periódica completa un patrón dentro de un marco de tiempo medible, llamado período, y repite ese patrón en períodos idénticos posteriores.
- Una señal no periódica cambia sin exhibir un patrón o ciclo que se repita con el tiempo.
- Las señales analógicas periódicas se pueden clasificar en simples o compuestas.
- Una señal analógica periódica simple, una onda sinusoidal, no se puede descomponer en señales más simples.
- Una señal analógica periódica compuesta se compone de múltiples ondas sinusoidales.
- Tres factores de los que depende la tasa de datos: el ancho de banda, el nivel de las señales, la calidad del canal.
- En el caso de un canal sin ruido, según Nyquist, la tasa de bits máxima teórica se puede calcular mediante las siguientes fórmulas: Tasa de bits = 2 XBX $\log_2$N Aquí, B es el ancho de banda del canal, N es el número de niveles de señal y la tasa de bits se calcula en bits por segundo.
- En el caso de un canal ruidoso, Claude Shannon desarrolló una fórmula en 1944 para calcular la tasa de datos teórica más alta para un canal ruidoso. Esta fórmula se llama la capacidad de Shannon: Capacidad = BX $\log_2$(1+SNR) Aquí, B es el ancho de banda del canal, SNR es la relación señal/ruido y la capacidad es la tasa de bits del canal en bits por segundo.
- Tres factores para el deterioro de la señal: atenuación, distorsión y ruido.

- La codificación de línea es el proceso de convertir datos digitales en señales digitales.
- Los diferentes esquemas de codificación de línea son: Esquema Unipolar, Esquema Polar, Esquema Manchester y Manchester Diferencial, Esquemas Bipolares, Esquemas Multinivel, Transmisión Multilínea:

- La codificación de bloques cambia un bloque de m bits en un bloque de n bits.
- Dos técnicas para la conversión de señal analógica a datos digitales son: modulación de código de pulso y modulación delta.
- El modo de transmisión de datos binarios a través de un medio de comunicación puede ser en serie o en paralelo.
- En caso de transmisión en serie, se transmite un bit de datos binarios a la vez.

-Las tres formas de transmisión en serie son: transmisión asíncrona, transmisión síncrona, transmisión isócrona.

- La multiplexación es el conjunto de técnicas utilizadas para la transmisión simultánea de múltiples señales a través de un solo enlace de datos.
- Los tres tipos de técnicas de multiplexación son: multiplexación por división de frecuencia, multiplexación por división de longitud de onda, multiplexación por división de tiempo.
- Un medio de transmisión se puede definir como cualquier tipo de sustancia material como sólido, líquido, gas o plasma que puede transportar cualquier señal o información desde una fuente hasta un destino.
- En el caso de las ondas electromagnéticas como la luz y las ondas de radio, se utiliza el vacío como medio de transmisión.
- Dos clases de medios de transmisión son: medios guiados y medios no guiados.
- Ejemplos de diferentes medios guiados son: medios magnéticos, cable de par trenzado, cable coaxial, cable de fibra óptica.
- Un cable de fibra óptica está construido con vidrio y plástico para transmitir señales en forma de luz.
- Los medios no guiados no requieren ningún conductor físico para transportar ondas electromagnéticas.
- Las señales no guiadas pueden moverse desde la fuente hasta el destino de tres maneras: propagación terrestre, propagación aérea y propagación de línea de visión.
- La transmisión inalámbrica se divide en tres grupos que son las ondas de radio, las microondas y las ondas infrarrojas.

- Las ondas de radio viajan en todas las direcciones desde la fuente, por lo que no es necesario que el transmisor y el receptor tengan una alineación física específica.
- Las ondas infrarrojas y milimétricas no guiadas se utilizan para la comunicación de corto alcance.
- Las ondas electromagnéticas con frecuencias entre 1 y 300 GHz se denominan microondas.
- Los láseres montados en los techos de los dos edificios se pueden usar para conectar las LAN en dos edificios.
- Una red de conmutación de circuitos consta de un conjunto de conmutadores conectados por enlaces físicos en los que cada enlace se divide en n canales.
- La comunicación en una red de conmutación de circuitos se logra en tres fases: establecimiento de la conexión, transferencia de datos y desconexión de la conexión.
- La red telefónica consta de tres componentes principales: bucles locales, troncales y centrales de conmutación.

3.11 RESPUESTAS PARA COMPROBAR TU PROGRESO

1. (I) A , (II) D , (III) C , (IV) D , (V) D ,(VI) B ,(VII) A ,(VIII) A ,(IX) B ,(X) A

2. I. compuesto, II. propagación, III. datos digitales, IV. NRZ-Nivel, NRZ-Invertir, V. datos digitales, VI. módem, VII. multiplexación, VIII. sin guía, IX. 3 kHz y 1 GHz,
   X. configuración de conexión, transferencia de datos, interrupción de conexión.

3. I. Verdadero, II. Falso, III. Cierto, IV. Falso, V. Verdadero, VI. Cierto, VII. Falso, VIII. Cierto , IX. Verdadero, X. Falso

3.12 LECTURAS ADICIONALES

- Behrouz A Forouzan: Comunicaciones de datos y redes, TATA McGraw Hill
- William Stallings: Datos y comunicaciones informáticas, Pearson Education
- Andrew S. Tanenbaum: Redes informáticas, Prentice-Hall India

3.13 PREGUNTAS MODELO

1. Explique los factores del deterioro de la señal.
2. Explicar los diferentes modos de transmisión.
3. Explicar los diferentes tipos de técnicas de multiplexación.
4. ¿Qué es el módem telefónico?
5. Explicar los cables de fibra óptica. ¿Cuáles son las ventajas de los cables de fibra óptica sobre los cables coaxiales?

6. ¿Qué es la conmutación de circuitos?
7. Explicar los servicios de red telefónica.

UNIDAD - 4: CAPA DE ENLACE DE DATOS

ESTRUCTURA DE LA UNIDAD

4.1 OBJETIVOS DE APRENDIZAJE

Después de pasar por esta unidad, podrá:

- aprender sobre control de errores y control de flujo
- describir la detección y corrección de errores
- describir los códigos de corrección y detección de errores
- aprender sobre el funcionamiento del protocolo Stop-and-Wait
- ilustrar el protocolo Go-Back-N
- ilustrar el protocolo de repetición selectiva
- aprender sobre el HDLC y el protocolo punto a punto
- aprender sobre ethernet

4.2 INTRODUCCIÓN

En el modelo OSI de siete capas de redes informáticas, la capa de enlace de datos es la capa 2. En el modelo de referencia TCP/IP, corresponde o es parte de la capa de enlace. La capa de enlace de datos es la capa de protocolo que transfiere datos entre nodos de red adyacentes en una red de área amplia o entre nodos en el mismo segmento de red de área local. La capa de enlace de datos proporciona los medios funcionales y de procedimiento para transferir datos entre entidades de red y podría proporcionar los medios para detectar y posiblemente corregir errores que puedan ocurrir en la capa física. La capa de enlace de datos se ocupa de la entrega local de tramas entre dispositivos en la misma LAN. Las tramas de enlace de datos no cruzan los límites de una red local. El enrutamiento entre redes y el direccionamiento global son funciones de capa superior, lo que permite que los protocolos de enlace de datos se centren en la entrega local, el direccionamiento, y arbitraje de medios. El enlace de datos proporciona así la transferencia de datos a través del enlace físico. Esa transferencia puede ser confiable o no confiable; muchos protocolos de enlace de datos no tienen reconocimientos de recepción y aceptación de tramas exitosas, y algunos protocolos de enlace de datos podrían ni siquiera tener ninguna forma de suma de verificación para verificar errores de transmisión. En esos casos, los protocolos de nivel superior deben proporcionar control de flujo, verificación de errores y reconocimientos y retransmisión.

En esta unidad estudiaremos los principios de diseño de la capa de enlace de datos, que se ocupan de los algoritmos para lograr una comunicación confiable y eficiente de unidades completas de información llamadas tramas entre dos máquinas adyacentes conectadas por un canal de comunicación que actúa conceptualmente como un cable. La propiedad esencial de un canal que lo hace "similar a un cable" es que los bits se entregan exactamente en el mismo orden en que se envían. Los canales de comunicación cometen errores ocasionalmente. Además, solo tienen una velocidad de datos finita y hay un retraso de propagación distinto de cero entre el momento en que se envía un bit y el momento en que se recibe. Estas limitaciones tienen implicaciones importantes para la eficiencia de la transferencia de datos. Los protocolos utilizados para las comunicaciones deben tener en cuenta todos estos factores. Estos protocolos son el tema de esta unidad.

4.3 ENCUADRE

Para brindar servicio a la capa de red, la capa de enlace de datos debe usar el servicio que le proporciona la capa física. Lo que hace la capa física es aceptar un flujo de bits sin procesar e intentar entregarlo al destino. Si el canal es ruidoso, la capa física agregará algo de redundancia a sus señales para reducir la tasa de error de bit a un nivel tolerable. Sin embargo, no se garantiza que el flujo de bits recibido por la capa de enlace de datos esté libre de errores. Algunos bits pueden tener valores diferentes y el número de bits recibidos puede ser menor, igual o mayor que el número de bits transmitidos. Corresponde a la capa de enlace de datos detectar y, si es necesario, corregir errores. El enfoque habitual es que la capa de enlace de datos divida el flujo de bits en tramas discretas, calcule un token corto llamado suma de verificación para

cada trama e incluir la suma de comprobación en la trama cuando se transmite. Cuando una trama llega al destino, se vuelve a calcular la suma de comprobación. Si la suma de verificación recién calculada es diferente de la contenida en la trama, la capa de enlace de datos sabe que se ha producido un error y toma medidas para solucionarlo (p. ej., descartando la trama incorrecta y posiblemente también enviando un informe de error). Dividir el flujo de bits en fotogramas es más difícil de lo que parece a primera vista. Un buen diseño debe hacer que sea fácil para un receptor encontrar el comienzo de nuevos cuadros mientras usa poco del ancho de banda del canal. Veremos cuatro métodos:

1. Recuento de bytes.
2. Marcar bytes con relleno de bytes.
3. Marcar bits con relleno de bits.
4. Violaciones de codificación de capa física.

El primer método de trama utiliza un campo en el encabezado para especificar el número de bytes en la trama. Cuando la capa de enlace de datos en el destino ve el conteo de bytes, sabe cuántos bytes siguen y, por lo tanto, dónde está el final de la trama. Esta técnica se muestra en **Figura 4.1(a)**para cuatro marcos de ejemplo pequeños de tamaños 5, 5, 8 y 8 bytes, respectivamente. El problema con este algoritmo es que el conteo puede distorsionarse por un error de transmisión. Por ejemplo, si el recuento de bytes de 5 en el segundo marco de**Figura 4.1(b)**se convierte en un 7 debido a un solo cambio de bit, el destino perderá la sincronización. Entonces no podrá ubicar el comienzo correcto del siguiente cuadro. Incluso si la suma de verificación es incorrecta, por lo que el destino sabe que el marco es incorrecto, aún no tiene forma de saber dónde comienza el siguiente marco. Enviar una trama de regreso a la fuente solicitando una retransmisión tampoco ayuda, ya que el destino no sabe cuántos bytes omitir para llegar al inicio de la retransmisión. Por esta razón, el método de conteo de bytes rara vez se usa solo.

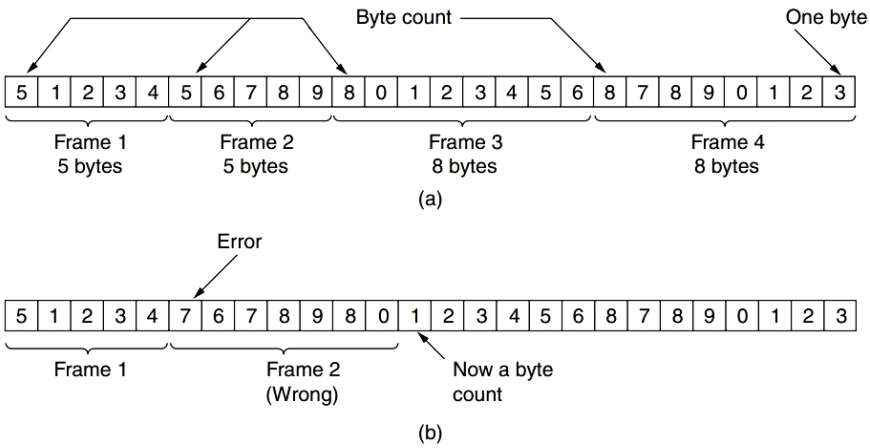


Fig. 4.1: Un flujo de bytes. (a) Sin errores. (b) Con un error.

El segundo método de trama soluciona el problema de la resincronización después de un error al hacer que cada trama comience y termine con bytes especiales. A menudo, el mismo byte, denominado byte indicador, se utiliza como delimitador inicial y final. Este byte se muestra en**Figura 4.2(a)**como **BANDERA**. dos consecutivos*bytes de bandera*índica el

final de un cuadro y el comienzo del siguiente. Por lo tanto, si el receptor alguna vez pierde la sincronización, solo puede buscar dos*bytes de bandera* para encontrar el final del cuadro actual y el comienzo del siguiente cuadro. Sin embargo, todavía hay un problema que tenemos que resolver. Puede suceder que el*byte indicador*ocurre en los datos, especialmente cuando se transmiten datos binarios como fotografías o canciones. Esta situación interferiría con el encuadre. Una forma de resolver este problema es hacer que la capa de enlace de datos del remitente inserte un byte de escape especial (**ESC**) justo antes de cada "**accidental**"*byte indicador*en los datos Así, un encuadre*byte indicador*puede distinguirse de uno en los datos por la ausencia o presencia de un*byte de escape*antes de eso. La capa de enlace de datos en el extremo receptor elimina los bytes de escape antes de entregar los datos a la capa de red. Esta técnica se llama**relleno de bytes**. Por supuesto, la siguiente pregunta es: ¿qué sucede si un *byte de escape*ocurre en medio de los datos? La respuesta es que también está relleno de un*byte de escape*. En el receptor, el primero*byte de escape*se elimina, dejando el byte de datos que le sigue. Algunos ejemplos se muestran en**Figura 4.2(b)**. En todos los casos, la secuencia de bytes entregada después del desrelleno es exactamente la misma que la secuencia de bytes original. Todavía podemos buscar un límite de marco buscando dos*bytes de bandera*seguidos, sin molestarse en deshacer escapes. El esquema de relleno de bytes representado en**Figura 4.2**es una ligera simplificación de la utilizada en**PPA**(Protocolo punto a punto), que se utiliza para transportar paquetes a través de enlaces de comunicaciones.

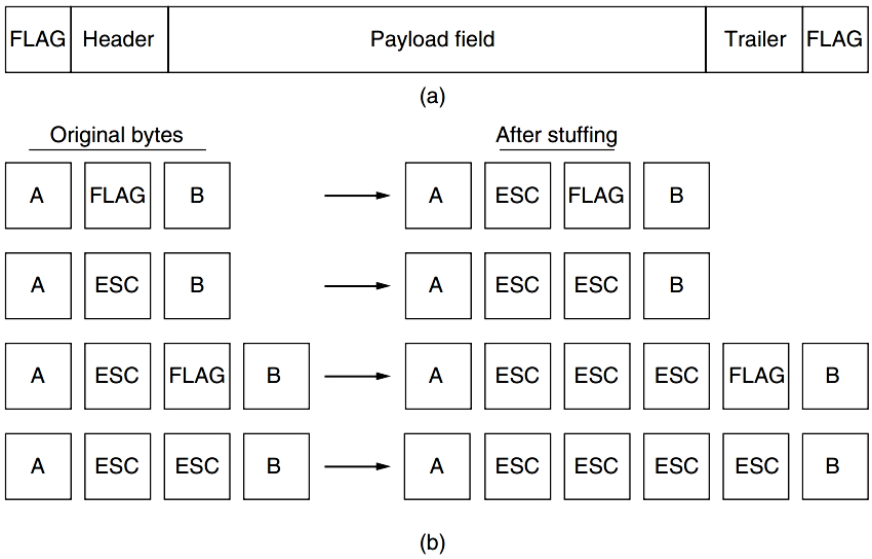


Fig. 4.2: (a) Un marco delimitado por bytes de bandera. (b) Cuatro ejemplos de secuencias de bytes antes y después del relleno de bytes.

El tercer método para delimitar el flujo de bits evita una desventaja del relleno de bytes, que es que está ligado al uso de bytes de 8 bits. El enmarcado también se puede realizar a nivel de bits, por lo que los marcos pueden contener un número arbitrario de bits compuestos por unidades de cualquier tamaño. Fue desarrollado para el otrora muy popular**HDLC**(Protocolo de control de enlace de datos de alto nivel). Cada cuadro comienza y termina con un patrón de bits especial, 01111110 o 0x7E en hexadecimal. Este patrón es un*byte indicador*. Cada vez que la capa de enlace de datos del remitente encuentra cinco 1 consecutivos en los datos, automáticamente introduce un bit 0 en el

flujo de bits saliente. Este relleno de bits es análogo al relleno de bytes, en el que un*byte de escape*se inserta en el flujo de caracteres saliente antes de un *byte indicador*en los datos También asegura una densidad mínima de transiciones que ayudan a la capa física a mantener la sincronización. USB (Universal Serial Bus) utiliza relleno de bits por este motivo. Cuando el receptor ve cinco bits 1 entrantes consecutivos, seguidos de un bit 0, automáticamente elimina (borra) el bit 0. Así como el relleno de bytes es completamente transparente para la capa de red en ambas computadoras, también lo es el relleno de bits. Si los datos del usuario contienen el patrón de bandera, 01111110, esta bandera se transmite como 011111010 pero se almacena en la memoria del receptor como 01111110.**Figura 4.3**da un ejemplo de relleno de bits. Con el relleno de bits, el límite entre dos fotogramas se puede reconocer sin ambigüedades por el patrón de bandera. Por lo tanto, si el receptor pierde la pista de dónde está, todo lo que tiene que hacer es escanear la entrada en busca de secuencias de banderas, ya que solo pueden ocurrir en los límites del marco y nunca dentro de los datos.

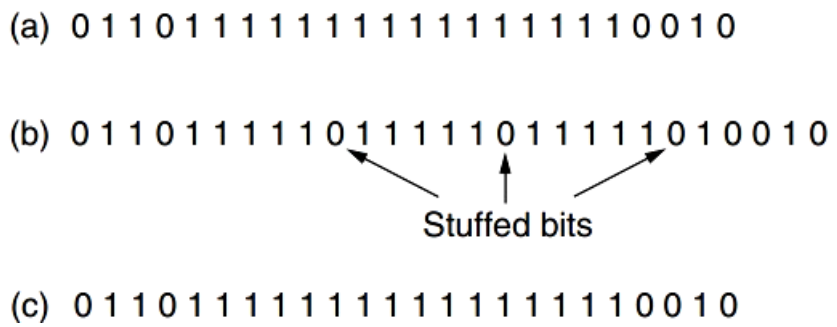


Fig 4.3: Relleno de bits. (a) Los datos originales. (b) Los datos tal como aparecer en la línea. (c) Los datos tal como están almacenados en el la memoria del receptor después de desllenar

Con el relleno de bits y bytes, un efecto secundario es que la longitud de un marco ahora depende del contenido de los datos que transporta. Por ejemplo, si no hay *bytes de bandera*en los datos, se pueden transportar 100 bytes en una trama de aproximadamente 100 bytes. Sin embargo, si los datos consisten únicamente en *bytes de bandera*, cada byte de marca se escapará y el marco tendrá una longitud aproximada de 200 bytes. Con el relleno de bits, el aumento sería de aproximadamente un 12,5 %, ya que se agrega 1 bit a cada byte. El último método de encuadre es usar un atajo desde la capa física. La codificación de bits como señales a menudo incluye redundancia para ayudar al receptor, lo que significa que algunas señales no aparecerán en los datos regulares. Por ejemplo, en el código de línea 4B/5B, 4 bits de datos se asignan a 5 bits de señal para garantizar suficientes transiciones de bits. Esto significa que 16 de las 32 posibilidades de señal no se utilizan. Podemos usar algunas señales reservadas para indicar el inicio y el final de los cuadros. En efecto, estamos usando "**violaciones de codificación**" para delimitar fotogramas. La belleza de este esquema es que, debido a que son señales reservadas, es fácil encontrar el inicio y el final de los cuadros y no hay necesidad de rellenar los datos. Muchos protocolos de enlace de datos utilizan una combinación de estos métodos por motivos de seguridad. Un patrón común utilizado para Ethernet y 802.11 es que una trama comience con un patrón bien definido llamado**preámbulo**. Este patrón puede ser bastante largo (72 bits es típico para 802.11) para permitir que el receptor se prepare para un paquete entrante. Luego, el preámbulo es seguido por un campo de longitud (es decir, conteo) en el encabezado que se usa para ubicar el final del marco.

4.4 CONTROL DE ERRORES

Habiendo resuelto el problema de marcar el inicio y el final de cada trama, llegamos al siguiente problema: cómo asegurarnos de que todas las tramas finalmente se entreguen a la capa de red en el destino y en el orden correcto. Supongamos por el momento que el receptor puede saber si una trama que recibe contiene información correcta o defectuosa. Para el servicio sin conexión no reconocido, podría estar bien si el remitente siguiera enviando marcos sin importar si llegaban correctamente. Pero para un servicio confiable y orientado a la conexión, no estaría bien en absoluto.

La forma habitual de garantizar una entrega confiable es proporcionar al remitente algunos comentarios sobre lo que sucede al otro lado de la línea. Por lo general, el protocolo requiere que el receptor envíe tramas de control especiales con reconocimientos positivos o negativos sobre las tramas entrantes. Si el remitente recibe un acuse de recibo positivo sobre un marco, sabe que el marco ha llegado de manera segura. Por otro lado, un reconocimiento negativo significa que algo salió mal y la trama debe transmitirse nuevamente.

Una complicación adicional surge de la posibilidad de que los problemas de hardware hagan que un marco desaparezca por completo. En este caso, el receptor no reaccionará en absoluto, ya que no tiene motivos para hacerlo. De manera similar, si se pierde la trama de acuse de recibo, el remitente no sabrá cómo proceder. Debe quedar claro que un protocolo en el que el remitente transmite una trama y luego espera un reconocimiento, positivo o negativo, se bloqueará para siempre si alguna vez se pierde una trama debido, por ejemplo, a un mal funcionamiento del hardware o a un canal de comunicación defectuoso. Esta posibilidad se aborda introduciendo temporizadores en la capa de enlace de datos. Cuando el remitente transmite una trama, generalmente también inicia un temporizador. El temporizador está configurado para expirar después de un intervalo lo suficientemente largo para que la trama llegue al destino, se procese allí, y hacer que el acuse de recibo se propague al remitente. Normalmente, la trama se recibirá correctamente y la confirmación se recuperará antes de que se agote el tiempo, en cuyo caso se cancelará el tiempo. Sin embargo, si se pierde el marco o el reconocimiento, el temporizador se apagará, alertando al remitente de un posible problema. La solución obvia es simplemente transmitir el marco nuevamente. Sin embargo, cuando las tramas pueden transmitirse varias veces, existe el peligro de que el receptor acepte la misma trama dos o más veces y la pase a la capa de red más de una vez. Para evitar que esto suceda, generalmente es necesario asignar números de secuencia a las tramas salientes, para que el receptor pueda distinguir las retransmisiones de los originales. la trama se recibirá correctamente y se recuperará el reconocimiento antes de que se agote el tiempo, en cuyo caso se cancelará el tiempo. Sin embargo, si se pierde el marco o el reconocimiento, el temporizador se apagará, alertando al remitente de un posible problema. La solución obvia es simplemente transmitir el marco nuevamente. Sin embargo, cuando las tramas pueden transmitirse varias veces, existe el peligro de que el receptor acepte la misma trama dos o más veces y la pase a la capa de red más de una vez. Para evitar que esto suceda, generalmente es necesario asignar números de secuencia a las tramas salientes, para que el receptor pueda distinguir las retransmisiones de los originales. la trama se recibirá correctamente y se recuperará el reconocimiento antes de que se agote el tiempo, en cuyo caso se cancelará el tiempo. Sin embargo, si se pierde el marco o el reconocimiento, el temporizador se apagará, alertando al remitente de un posible problema. La solución obvia es simplemente transmitir el marco nuevamente. Sin embargo, cuando las tramas pueden transmitirse varias veces, existe el peligro de que el receptor acepte la misma trama dos o más veces y la pase a la capa de red más de una vez. Para evitar que esto suceda, generalmente es necesario asignar números de secuencia a las tramas salientes, para que el receptor pueda distinguir las retransmisiones de los originales. alertando al remitente de un problema potencial. La solución obvia es simplemente transmitir el marco nuevamente. Sin embargo, cuando las tramas pueden transmitirse varias veces, existe el peligro de que el receptor acepte la misma trama dos o más veces y la pase a la capa de red más de una vez. Para evitar que esto suceda, generalmente es necesario asignar números de secuencia a las tramas salientes, para que el receptor pueda distinguir las retransmisiones de los originales. alertando al remitente de un problema potencial. La solución obvia es simplemente transmitir el marco nuevamente. Sin embargo, cuando las tramas pueden transmitirse varias veces, existe el peligro de que el receptor acepte la m

Todo el tema de administrar los temporizadores y los números de secuencia para garantizar que cada trama se pase finalmente a la capa de red en el destino exactamente una vez, ni más ni menos, es una parte importante de las funciones de la capa de enlace de datos (y capas superiores).

4.5 CONTROL DE FLUJO

Otro problema de diseño importante que ocurre en la capa de enlace de datos es qué hacer con un remitente que sistemáticamente quiere transmitir tramas más rápido de lo que el receptor puede aceptarlas. Esta situación puede ocurrir cuando el remitente se ejecuta en una computadora potente y rápida y el receptor se ejecuta en una máquina lenta y de gama baja. Incluso si la transmisión está libre de errores, es posible que el receptor no pueda manejar las tramas tan rápido como llegan y perderá algunas. Está claro que hay que hacer algo para evitar esta situación. Se utilizan comúnmente dos enfoques. En el primero, **control de flujo basado en retroalimentación**, el receptor devuelve información al remitente dándole permiso para enviar más datos, o al menos diciéndole al remitente cómo está el receptor. En el segundo, **control de flujo basado en tasa**, el protocolo tiene un mecanismo incorporado que limita la velocidad a la que los remitentes pueden transmitir datos, sin utilizar la retroalimentación del receptor. Los esquemas basados en retroalimentación se ven tanto en la capa de enlace como en las capas superiores. Este último es más común en estos días, en cuyo caso el hardware de la capa de enlace está diseñado para ejecutarse lo suficientemente rápido como para no causar pérdidas. Por ejemplo, a veces se dice que las implementaciones de hardware de la capa de enlace como NIC (tarjetas de interfaz de red) se ejecutan a "velocidad de cable", lo que significa que pueden manejar tramas tan rápido como pueden llegar al enlace. Cualquier desbordamiento no es un problema de enlace, por lo que las capas superiores los manejan. Se conocen varios esquemas de control de flujo basados en retroalimentación, pero la mayoría de ellos utilizan el mismo principio básico. El protocolo contiene reglas bien definidas sobre cuándo un remitente puede transmitir la siguiente trama.

4.6 DETECCIÓN Y CORRECCIÓN DE ERRORES

Los canales de comunicación tienen una gama de características. Algunos canales, como la fibra óptica en las redes de telecomunicaciones, tienen tasas de error muy pequeñas, por lo que los errores de transmisión son raros. Pero otros canales, especialmente los enlaces inalámbricos y los bucles locales obsoletos, tienen tasas de error que son órdenes de magnitud mayores. Para estos enlaces, los errores de transmisión son la norma. No pueden evitarse a un costo o gasto razonable en términos de desempeño. La conclusión es que los errores de transmisión llegaron para quedarse. Tenemos que aprender a lidiar con ellos.

Los diseñadores de redes han desarrollado dos estrategias básicas para tratar los errores. Ambos agregan información redundante a los datos que se envían. Una estrategia es incluir suficiente información redundante para permitir que el receptor deduzca cuáles deben haber sido los datos transmitidos. La otra es incluir solo la redundancia suficiente para permitir que el receptor deduzca que se ha producido un error y solicite una retransmisión. La estrategia anterior utiliza**códigos de corrección de errores** y este último utiliza**códigos de detección de errores**. El uso de códigos de corrección de errores a menudo se denomina**FEC**(Corrección de errores de reenvío).

Cada una de estas técnicas ocupa un nicho ecológico diferente. En canales que son altamente confiables, como la fibra, es más económico usar un código de detección de errores y simplemente retransmitir el bloque ocasional que se encuentre defectuoso. Sin embargo, en canales como enlaces inalámbricos que cometen muchos errores, es mejor agregar redundancia a cada bloque para que el receptor pueda averiguar cuál era el bloque transmitido originalmente. FEC se usa en canales ruidosos porque las retransmisiones tienen la misma probabilidad de ser erróneas que la primera transmisión. Una consideración clave para estos códigos es el tipo de errores que es probable que ocurran. Ni los códigos de corrección de errores ni los códigos de detección de errores pueden manejar todos los errores posibles, ya que es tan probable que los bits redundantes que ofrecen protección se reciban con error como los bits de datos. Sería bueno que el canal tratara los bits redundantes de manera diferente a los bits de datos, pero no es así. Todos son solo bits para el canal. Esto significa que para evitar errores no detectados, el código debe ser lo suficientemente fuerte para manejar los errores esperados.

Un modelo es que los errores son causados por valores extremos de ruido térmico que abruma la señal de forma breve y ocasional, dando lugar a errores aislados de un solo bit. Otro modelo es que los errores tienden a aparecer en ráfagas en lugar de uno solo. Este modelo se deriva de los procesos físicos que los generan, como un desvanecimiento profundo en un canal inalámbrico o una interferencia eléctrica transitoria en un canal cableado. Ambos modelos son importantes en la práctica y tienen diferentes ventajas y desventajas. Tener los errores en ráfagas tiene ventajas y desventajas sobre los errores aislados de un solo bit. Por el lado de las ventajas, los datos de la computadora siempre se envían en bloques de bits. Suponga que el tamaño del bloque era de 1000 bits y la tasa de error era de 0,001 por bit. Si los errores fueran independientes, la mayoría de los bloques contendrían un error. Sin embargo, si los errores se produjeron en ráfagas de 100, solo un bloque de cada 100 se vería afectado, en promedio. La desventaja de los errores de ráfaga es que, cuando ocurren, son mucho más difíciles de corregir que los errores aislados. También existen otros tipos de errores. A veces, se sabrá la ubicación de un error, quizás porque la capa física recibió una señal analógica que estaba lejos del valor esperado para un 0 o un 1 y declaró que el bit se había perdido. Esta situación se denomina **canal de borrado**. Es más fácil corregir errores en canales de borrado que en canales que cambian bits porque aunque se haya perdido el valor del bit, al menos sabemos cuál es el bit erróneo. Sin embargo, a menudo no tenemos el beneficio de los borrados.

A continuación, examinaremos tanto los códigos de corrección de errores como los códigos de detección de errores. Sin embargo, es importante tener en cuenta dos puntos. Primero, cubrimos estos códigos en la capa de enlace porque este es el primer lugar en el que nos encontramos con el problema de transmitir grupos de bits de manera confiable. Sin embargo, los códigos se usan ampliamente porque la confiabilidad es una preocupación general. Los códigos de corrección de errores también se ven en la capa física, particularmente para canales ruidosos, y en capas superiores, particularmente para medios en tiempo real y distribución de contenido. Los códigos de detección de errores se utilizan comúnmente en las capas de enlace, red y transporte. El segundo punto a tener en cuenta es que los códigos de error son matemáticas aplicadas. A menos que sea particularmente experto en los campos de Galois o en las propiedades de las matrices dispersas, debe obtener códigos con buenas propiedades de una fuente confiable en lugar de crear los suyos propios. De hecho, esto es lo que muchos

los estándares de protocolo sí lo hacen, con los mismos códigos apareciendo una y otra vez.

4.6.1 CÓDIGOS DE CORRECCIÓN DE ERRORES

Examinaremos cuatro códigos de corrección de errores diferentes:

1. Códigos de Hamming.
2. Códigos convolucionales binarios.
3. Códigos Reed-Solomon.
4. Códigos de verificación de paridad de baja densidad.

Todos estos códigos agregan redundancia a la información que se envía. Un marco consta de ***metro*** bits de datos (es decir, mensajes) y ***r*** bits redundantes (es decir, de verificación). En un código de bloque, el ***r*** Los bits de verificación se calculan únicamente como una función de la ***metro*** bits de datos con los que están asociados, como si el ***metro*** los bits se buscaron en una tabla grande para encontrar su correspondiente ***r*** comprobar bits. En un código sistemático, el ***metro*** los bits de datos se envían directamente, junto con los bits de verificación, en lugar de codificarse antes de enviarlos. En un código lineal, el ***r*** Los bits de control se calculan como una función lineal de la ***metro*** bits de datos. La adición exclusiva OR (XOR) o módulo 2 es una opción popular. Esto significa que la codificación se puede realizar con operaciones como multiplicaciones de matrices o circuitos lógicos simples. Los códigos que veremos en esta sección son códigos de bloque sistemáticos y lineales, a menos que se indique lo contrario.

Sea la longitud total de un bloque ***norte*** (es decir, ***n=m+r***). Describiremos esto como un (***norte***, ***metro***) código. Un ***norte*** La unidad de bits que contiene datos y bits de verificación se conoce como ***norte*** -palabra de código de bits. La tasa de código, o simplemente tasa, es la fracción de la palabra de código que transporta información que no es redundante, o ***Minnesota***. Las tasas utilizadas en la práctica varían ampliamente. Pueden ser 1/2 para un canal ruidoso, en cuyo caso la mitad de la información recibida es redundante, o cerca de 1 para un canal de alta calidad, con solo una pequeña cantidad de bits de verificación agregados a un mensaje grande. Para entender cómo se pueden manejar los errores, primero es necesario mirar de cerca qué es realmente un error. Dadas dos palabras de código que pueden transmitirse o recibirse, por ejemplo, 10001001 y 10110001, es posible determinar cuántos bits correspondientes difieren. En este caso, 3 bits difieren. Para determinar cuántos bits difieren, simplemente use XOR en las dos palabras de código y cuente el número de 1 bits en el resultado.

Por ejemplo:

$$\begin{array}{r} 10001001 \\ \underline{10110001} \\ 00111000 \end{array}$$

El número de posiciones de bits en las que difieren dos palabras de código se denomina **distancia de hamming** (Hamming, 1950). Su importancia es que si dos palabras clave son una distancia de Hamming ***D*** aparte, requerirá ***D*** errores de un solo bit para convertir uno en el otro. Dado el algoritmo para calcular los bits de control, es posible construir una lista completa de las palabras de código legales y, a partir de esta lista, encontrar las dos palabras de código con la distancia de Hamming más pequeña. Esta distancia es la distancia de Hamming del código completo. En la mayoría de las aplicaciones de transmisión de datos, todos $\mathbf{2}^{\mathbf{metro}}$ los posibles mensajes de datos son legales,

pero debido a la forma en que se calculan los bits de control, no todos los 2^{norte} se utilizan posibles palabras de código. De hecho, cuando hay r bits de verificación, sólo la pequeña fracción de $2^{metro}/2^{norte}$ o $1/2^{r}$ de los posibles mensajes serán palabras clave legales. Es la escasez con la que el mensaje está incrustado en el espacio de las palabras clave lo que permite al receptor detectar y corregir errores. Las propiedades de detección y corrección de errores de un código de bloque dependen de su **distancia de hamming**. Para detectar de forma fiable D errores, necesitas una distancia $D+1$ código porque con tal código no hay forma de que D los errores de un solo bit pueden cambiar una palabra de código válida en otra palabra de código válida. Cuando el receptor ve una palabra clave ilegal, puede decir que se ha producido un error de transmisión. Del mismo modo, para corregir D errores, necesitas una distancia $2D+1$ código porque de esa manera las palabras clave legales están tan separadas que incluso con D cambia la palabra clave original está aún más cerca que cualquier otra palabra clave. Esto significa que la palabra clave original se puede determinar de forma única en función de la suposición de que es menos probable que se produzca un mayor número de errores.

Como un ejemplo simple de un código de corrección de errores, considere un código con solo cuatro palabras de código válidas:

0000000000, 0000011111, 1111100000 y 1111111111

Este código tiene una distancia de 5, lo que significa que puede corregir errores dobles o detectar errores cuádruples. Si llega la palabra clave 0000000111 y esperamos solo errores de uno o dos bits, el receptor sabrá que el original debe haber sido 0000011111. Sin embargo, si un error triple cambia 0000000000 en 0000000111, el error no se corregirá correctamente. . Alternativamente, si esperamos todos estos errores, podemos detectarlos. Ninguna de las palabras de código recibidas son palabras de código legales, por lo que debe haber ocurrido un error. Debería ser evidente que en este ejemplo no podemos corregir errores dobles y detectar errores cuádruples porque esto requeriría que interpretáramos una palabra clave recibida de dos maneras diferentes. En nuestro ejemplo, la tarea de decodificación mediante la búsqueda de la palabra de código legal más cercana a la palabra de código recibida se puede realizar mediante inspección. Desafortunadamente, en el caso más general en el que todas las palabras de código deben evaluarse como candidatas, esta tarea puede ser una búsqueda que requiere mucho tiempo. En cambio, los códigos prácticos están diseñados para admitir atajos para encontrar lo que probablemente era la palabra clave original. Imagina que queremos diseñar un código con $metro$ bits de mensaje y r bits de verificación que permitirán corregir todos los errores individuales. Cada una de las 2^{metro} mensajes legales tiene $norte$ palabras de código ilegales a una distancia de 1 de él. Estos se forman invirtiendo sistemáticamente cada uno de los $norte$ pedacitos en el $norte$-bit código-palabra formado a partir de él. Así, cada uno de los 2^{metro} mensajes legales requiere $norte+1$ patrones de bits dedicados a él. Dado que el número total de patrones de bits es 2^{norte}, Debemos tener $(n+1)2^{metro} \leq 2^{norte}$. Utilizando $norte = metro + r$, este requisito se convierte en

$$(m+r+1) \leq 2^{r} \quad \text{———} \quad 4.1$$

Dado $metro$, esto pone un límite inferior en el número de bits de verificación necesarios para corregir errores individuales. Este límite inferior teórico puede, de hecho, lograrse utilizando un método debido a Hamming (1950). En los códigos de Hamming, los bits de la palabra clave se numeran consecutivamente, comenzando con el bit 1 en el extremo izquierdo, el bit 2 inmediatamente a la derecha, y así sucesivamente. Los bits que son potencias de 2 (1, 2, 4,

8, 16, etc.) son bits de control. El resto (3, 5, 6, 7, 9, etc.) se rellenan con el ***metro***bits de datos. Este patrón se muestra para un código Hamming (11,7) con 7 bits de datos y 4 bits de verificación en**Figura 4.4**. Cada bit de control obliga a que la suma del módulo 2, o paridad, de algún conjunto de bits, incluido él mismo, sea par (o impar). Un bit puede incluirse en varios cálculos de bits de verificación. Para ver qué bits de verificación el bit de datos en posición***k***contribuye a, reescribir***k***como una suma de potencias de 2. Por ejemplo, 11 = 1 + 2 + 8 y 29 = 1 + 4 + 8 + 16. Un bit se verifica solo con los bits de verificación que ocurren en su expansión (por ejemplo, el bit 11 se verifica mediante bits 1, 2 y 8). En el ejemplo, los bits de verificación se calculan para sumas de paridad par para un mensaje que es la letra ASCII "**A**".

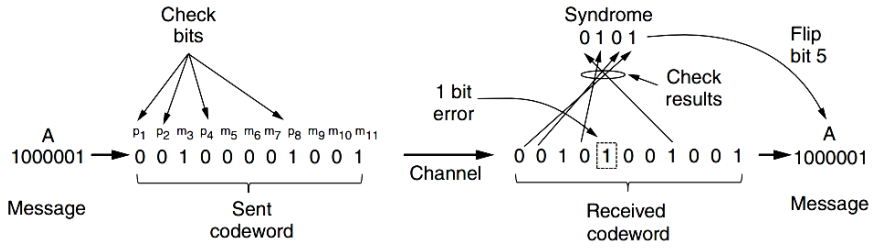


Fig 4.4: Ejemplo de un código de Hamming (11, 7) que corrige un error de un solo bit.

Esta construcción da un código con una distancia de Hamming de 3, lo que significa que puede corregir errores simples (o detectar errores dobles). La razón de la cuidadosa numeración de mensajes y bits de verificación se hace evidente en el proceso de decodificación. Cuando llega una palabra de código, el receptor vuelve a realizar los cálculos de bits de control, incluidos los valores de los bits de control recibidos. Los llamamos los resultados de la verificación. Si los bits de verificación son correctos, para sumas de paridad par, cada resultado de verificación debe ser cero. En este caso, la palabra clave se acepta como válida. Sin embargo, si los resultados de la verificación no son todos cero, se ha detectado un error. El conjunto de resultados de verificación forma el**síndrome de error**que se utiliza para identificar y corregir el error. En**Figura 4.4**, se produjo un error de un solo bit en el canal, por lo que los resultados de la comprobación son 0, 1, 0 y 1 bifurcación = 8, 4, 2 y 1, respectivamente. Esto da un síndrome de 0101 o 4+1=5. Por el diseño del esquema, esto significa que el quinto bit es erróneo. Voltear el bit incorrecto (que podría ser un bit de control o un bit de datos) y descartar los bits de control da el mensaje correcto de un ASCII "**A**". Las distancias de Hamming son valiosas para comprender los códigos de bloque, y los códigos de Hamming se utilizan en la memoria de corrección de errores. Sin embargo, la mayoría de las redes usan códigos más fuertes. El segundo código que veremos es un**convolucional**código. Este código es el único que cubriremos que no es un código de bloque. En un código convolucional, un codificador procesa una secuencia de bits de entrada y genera una secuencia de bits de salida. No existe un tamaño de mensaje natural ni un límite de codificación como en un código de bloque. La salida depende de los bits de entrada actuales y anteriores. Es decir, el codificador tiene memoria. El número de bits anteriores de los que depende la salida se denomina longitud de restricción del código. Los códigos convolucionales se especifican en términos de su tasa y longitud de restricción. Los códigos convolucionales se utilizan ampliamente en redes desplegadas, por ejemplo, como parte del sistema de telefonía móvil GSM, en comunicaciones por satélite y

en 802.11. Como ejemplo, un código convolucional popular se muestra en**Figura 4.5**. Este código se conoce como el código convolucional de la NASA de **r = 1/2**y **k=7**, ya que se utilizó por primera vez para las misiones espaciales Voyager a partir de 1977. Desde entonces, se ha reutilizado generosamente, por ejemplo, como parte de 802.11.

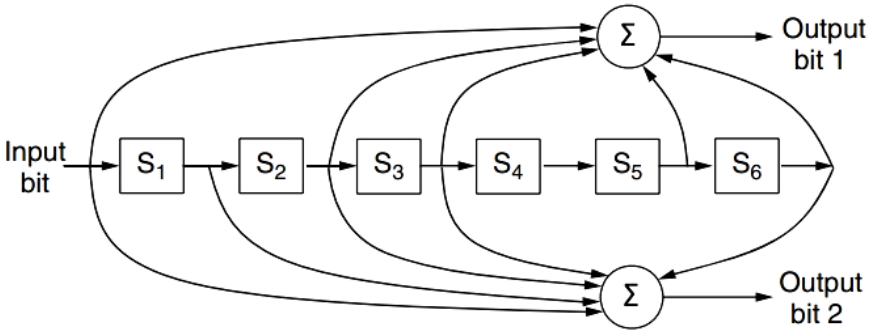


Fig 4.5: El código convolucional binario de la NASA utilizado en 802.11

En**Figura 4.5**, cada bit de entrada en el lado izquierdo produce dos bits de salida en el lado derecho que son sumas XOR de la entrada y el estado interno. Dado que trata con bits y realiza operaciones lineales, se trata de un código convolucional lineal binario. Dado que 1 bit de entrada produce 2 bits de salida, la tasa de código es 1/2. No es sistemático ya que ninguno de los bits de salida es simplemente el bit de entrada. El estado interno se mantiene en seis registros de memoria. Cada vez que se ingresa otro bit, los valores en los registros se desplazan hacia la derecha. Por ejemplo, si se ingresa 111 y el estado inicial es todo ceros, el estado interno, escrito de izquierda a derecha, se convertirá en 100000, 110000 y 111000 después de que se hayan ingresado los bits primero, segundo y tercero. Los bits de salida serán 11, seguidos de 10 y luego 01. Se necesitan siete turnos para vaciar una entrada por completo para que no afecte la salida. La longitud de restricción de este código es por lo tanto**k=7**. Un código convolucional se decodifica encontrando la secuencia de bits de entrada que es más probable que haya producido la secuencia observada de bits de salida (que incluye cualquier error). Para pequeños valores de***k***, esto se hace con un algoritmo ampliamente utilizado desarrollado por Viterbi (Forney, 1973). El algoritmo recorre la secuencia observada, manteniendo para cada paso y para cada posible estado interno la secuencia de entrada que habría producido la secuencia observada con la menor cantidad de errores. La secuencia de entrada que requiere la menor cantidad de errores al final es el mensaje más probable. Los códigos convolucionales han sido populares en la práctica porque es fácil factorizar la incertidumbre de que un bit sea un 0 o un 1 en la decodificación. Por ejemplo, supongamos−**1V**es el nivel 0 lógico y**+1V**es el nivel lógico 1, podríamos recibir 0,9 V y −0,1 V para 2 bits. En lugar de asignar estas señales a 1 y 0 de inmediato, nos gustaría tratar 0,9 V como "muy probablemente un 1" y −0,1 V como "quizás un 0" y corregir la secuencia como un todo. Las extensiones del algoritmo de Viterbi pueden funcionar con estas incertidumbres para proporcionar una corrección de errores más fuerte. Este enfoque de trabajar con la incertidumbre de un bit se llama **decodificación de decisión suave**. Por el contrario, decidir si cada bit es un 0 o un 1 antes de la subsiguiente corrección de errores se llama**decodificación de decisión difícil**.

El tercer tipo de código de corrección de errores que describiremos es el **Código Reed-Solomon**. Como los códigos de Hamming, Reed-Solomon

Los códigos son códigos de bloques lineales y, a menudo, también son sistemáticos. A diferencia de los códigos Hamming, que operan en bits individuales, los códigos Reed-Solomon operan en***metro***símbolos de bits. Naturalmente, las matemáticas son más complicadas, por lo que describiremos su funcionamiento por analogía. Los códigos Reed-Solomon se basan en el hecho de que cada ***norte***El polinomio de grado está determinado únicamente por**n+1**puntos. Por ejemplo, una línea que tiene la forma **hacha+b**está determinada por dos puntos. Los puntos adicionales en la misma línea son redundantes, lo que es útil para la corrección de errores. Imagine que tenemos dos puntos de datos que representan una línea y enviamos esos dos puntos de datos más dos puntos de control elegidos para estar en la misma línea. Si uno de los puntos se recibe por error, aún podemos recuperar los puntos de datos ajustando una línea a los puntos recibidos. Tres de los puntos estarán en la recta y un punto, el del error, no. Al encontrar la línea hemos corregido el error. Los códigos Reed-Solomon en realidad se definen como polinomios que operan sobre campos finitos, pero funcionan de manera similar. Para***metro***símbolos de bits, las palabras de código son$\mathbf{2_{metro}-1}$símbolos de largo. Una opción popular es hacer**metro=8** para que los símbolos sean bytes. Una palabra clave tiene entonces 255 bytes de longitud. El código (255, 233) es muy utilizado; agrega 32 símbolos redundantes a 233 símbolos de datos. La decodificación con corrección de errores se realiza con un algoritmo desarrollado por Berlekamp y Massey que puede realizar eficientemente la tarea de ajuste de códigos de longitud moderada (Massey, 1969). Los códigos Reed-Solomon se usan ampliamente en la práctica debido a sus fuertes propiedades de corrección de errores, particularmente para errores de ráfaga. Se utilizan para DSL, datos por cable, comunicaciones por satélite y, quizás, de forma más generalizada en CD, DVD y discos Blu-ray. porque se basan en***metro***símbolos de bits, un error de un solo bit y un***metro*****poco**el error de ráfaga se trata simplemente como un error de símbolo. Cuando***2t*** se agregan símbolos redundantes, un código Reed-Solomon puede corregir hasta***t***errores en cualquiera de los símbolos transmitidos. Esto significa, por ejemplo, que el código (255, 233), que tiene 32 símbolos redundantes, puede corregir hasta 16 errores de símbolo. Dado que los símbolos pueden ser consecutivos y cada uno tiene 8 bits, se puede corregir una ráfaga de errores de hasta 128 bits. La situación es incluso mejor si el modelo de error es uno de borrados (por ejemplo, un rasguño en un CD que borra algunos símbolos). En este caso, hasta***2t***los errores se pueden corregir. Los códigos Reed-Solomon a menudo se usan en combinación con otros códigos, como un código convolucional. El pensamiento es el siguiente. Los códigos convolucionales son efectivos para manejar errores de bits aislados, pero fallarán, probablemente con una ráfaga de errores, si hay demasiados errores en el flujo de bits recibido. Al agregar un código Reed-Solomon dentro del código convolucional, la decodificación Reed-Solomon puede eliminar las ráfagas de error, una tarea en la que es muy bueno. El código general proporciona una buena protección contra errores únicos y de ráfaga.

El último código de corrección de errores que cubriremos es el**LDPC** (**Comprobación de paridad de baja densidad**) código. Los códigos LDPC son códigos de bloques lineales que fueron inventados por Robert Gallagher en su tesis doctoral (Gallagher, 1962). Como la mayoría de las tesis, fueron rápidamente olvidadas, solo para ser reinventadas en 1995 cuando los avances en el poder de la computación las hicieron prácticas. En un código LDPC, cada bit de salida se forma a partir de solo una fracción de los bits de entrada. Esto conduce a una representación matricial del código que tiene una densidad baja de 1, de ahí el nombre del código. Las palabras de código recibidas se decodifican con un

algoritmo de aproximación que mejora iterativamente el mejor ajuste de los datos recibidos a una palabra clave legal. Esto corrige errores. Los códigos LDPC son prácticos para tamaños de bloque grandes y tienen excelentes capacidades de corrección de errores que superan a muchos otros códigos en la práctica. Por esta razón, se están incluyendo rápidamente en los nuevos protocolos. Forman parte del estándar para transmisión de video digital, Ethernet de 10 Gbps, redes de línea eléctrica y la última versión de 802.11.

4.6.2 CÓDIGOS DE DETECCIÓN DE ERRORES

Los códigos de corrección de errores se utilizan ampliamente en enlaces inalámbricos, que son notoriamente ruidosos y propensos a errores en comparación con las fibras ópticas. Sin códigos de corrección de errores, sería difícil pasar algo. Sin embargo, sobre fibra o cobre de alta calidad, la tasa de error es mucho menor, por lo que la detección y retransmisión de errores suele ser más eficiente para tratar errores ocasionales.

Examinaremos tres códigos diferentes de detección de errores. Todos son códigos de bloque sistemáticos y lineales:

1. Paridad.
2. Sumas de verificación.
3. Comprobaciones de redundancia cíclica (CRC).

Para ver cómo pueden ser más eficientes que los códigos de corrección de errores, considere el primer código de detección de errores, en el que un solo **bit de paridad**se adjunta a los datos. El bit de paridad se elige de modo que el número de bits 1 en la palabra de código sea par (o impar). Hacer esto es equivalente a calcular el bit de paridad (par) como la suma de módulo 2 o XOR de los bits de datos. Por ejemplo, cuando se envía 1011010 en paridad par, se agrega un bit al final para convertirlo en 10110100. Con paridad impar, 1011010 se convierte en 10110101. Un código con un solo bit de paridad tiene una distancia de 2, ya que cualquier error de un solo bit produce una palabra clave con la paridad incorrecta. Esto significa que puede detectar errores de un solo bit. Considere un canal en el que los errores están aislados y la tasa de error es$\mathbf{10}^{-6}$ por bit Esto puede parecer una tasa de error pequeña, pero es, en el mejor de los casos, una tasa justa para un cable largo que es un desafío para la detección de errores. Los enlaces LAN típicos proporcionan tasas de error de bit de $\mathbf{10}^{-10}$. Deje que el tamaño del bloque sea de 1000 bits. Para proporcionar corrección de errores para bloques de 1000 bits, sabemos por**Ecuación 4.1**que se necesitan 10 bits de control. Por lo tanto, un megabit de datos requeriría 10.000 bits de verificación. Para detectar simplemente un bloque con un solo error de 1 bit, será suficiente un bit de paridad por bloque. Una vez cada 1000 bloques, se encontrará un bloque erróneo y se deberá transmitir un bloque extra (1001 bits) para reparar el error. La sobrecarga total del método de detección de errores y retransmisión es de solo 2001 bits por megabit de datos, frente a los 10 000 bits de un código Hamming. Una dificultad con este esquema es que un solo bit de paridad solo puede detectar de manera confiable un error de un solo bit en el bloque. Si el bloque está muy distorsionado por un error de ráfaga larga, la probabilidad de que se detecte el error es de solo 0,5, lo que es difícilmente aceptable. Las probabilidades se pueden mejorar considerablemente si cada bloque a enviar se considera como una matriz rectangular***norte***bits de ancho y***k***bits alto. Ahora, si calculamos y enviamos un bit de paridad para cada fila, hasta***k***los errores de bit se detectarán de forma fiable siempre que haya como máximo un error por fila. Sin embargo,

hay algo más que podemos hacer que brinda una mejor protección contra los errores de ráfaga: podemos calcular los bits de paridad sobre los datos en un orden diferente al orden en que se transmiten los bits de datos. Hacerlo se llama**intercalado**. En este caso, calcularemos un bit de paridad para cada uno de los***norte***columnas y enviar todos los bits de datos como***k***filas, enviando las filas de arriba hacia abajo y los bits de cada fila de izquierda a derecha de la manera habitual. En la última fila, enviamos el***norte***bits de paridad. Este orden de transmisión se muestra en **Figura 4.6**para n=7 y k=7.

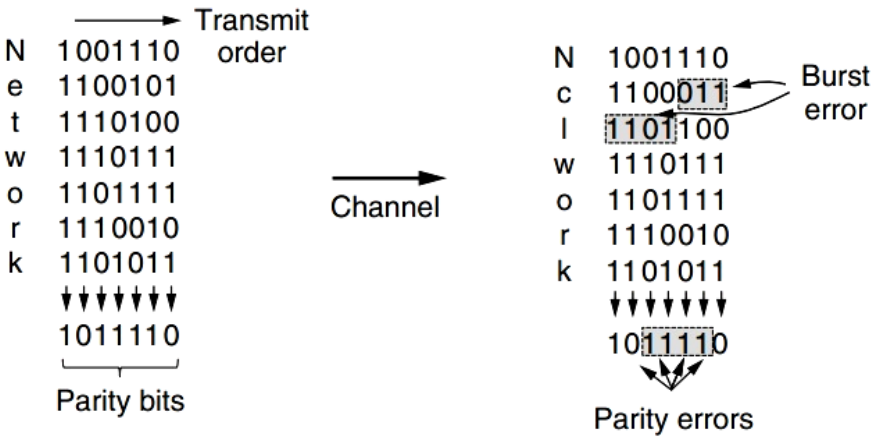


Fig 4.6: Intercalado de bits de paridad para detectar un error de ráfaga

El intercalado es una técnica general para convertir un código que detecta (o corrige) errores aislados en un código que detecta (o corrige) errores de ráfaga. En**Figura 4.6**, cuando un error de ráfaga de longitud**n=7**ocurre, los bits erróneos se distribuyen en diferentes columnas. (Un error de ráfaga no implica que todos los bits sean incorrectos; solo implica que al menos el primero y el último son incorrectos. En**Figura 4.6**, 4 bits se invirtieron en un rango de 7 bits.) Como máximo 1 bit en cada uno de los***norte*** las columnas se verán afectadas, por lo que los bits de paridad en esas columnas detectarán el error. Este método utiliza***norte*** bits de paridad en bloques de***kn*** bits de datos para detectar un solo error de ráfaga de longitud***norte***o menos. Una explosión de longitud**n+1**pasará desapercibido, sin embargo, si el primer bit se invierte, el último bit se invierte y todos los demás bits son correctos. Si el bloque está muy distorsionado por una ráfaga larga o por múltiples ráfagas más cortas, la probabilidad de que cualquiera de los***norte***columnas tendrán la paridad correcta por accidente es 0.5, por lo que la probabilidad de que se acepte un bloque defectuoso cuando no debería ser es**2** $^{-n}$.

El segundo tipo de código de detección de errores, el**suma de control**, está estrechamente relacionado con grupos de bits de paridad. La palabra "suma de control" se usa a menudo para referirse a un grupo de bits de control asociados con un mensaje, independientemente de cómo se calculen. Un grupo de bits de paridad es un ejemplo de suma de comprobación. Sin embargo, existen otras sumas de verificación más sólidas basadas en una suma continua de los bits de datos del mensaje. La suma de comprobación suele colocarse al final del mensaje, como complemento de la función de suma. De esta forma, los errores pueden detectarse sumando la palabra de código completa recibida, tanto los bits de datos como la suma de comprobación. Si el resultado es cero, no se ha detectado ningún error. Un ejemplo de una suma de verificación es la suma de verificación de Internet de 16 bits utilizada en todos los paquetes de Internet como parte del protocolo IP (Braden et al., 1988). Esta suma de comprobación es una suma de los bits del mensaje divididos en palabras de 16 bits. Porque este método

opera con palabras en lugar de bits, como en la paridad, los errores que dejan la paridad sin cambios aún pueden alterar la suma y ser detectados. Por ejemplo, si el bit de orden más bajo en dos palabras diferentes cambia de 0 a 1, una verificación de paridad en estos bits no detectará un error. Sin embargo, se agregarán dos 1 a la suma de verificación de 16 bits para producir un resultado diferente. Entonces se puede detectar el error. La suma de comprobación de Internet se calcula en la aritmética del complemento a uno en lugar de como el módulo$\mathbf{2_{dieciséis}}$suma. En la aritmética del complemento a uno, un número negativo es el complemento bit a bit de su contraparte positiva. Las computadoras modernas ejecutan la aritmética del complemento a dos, en la que un número negativo es el complemento a uno más uno. En una computadora con complemento a dos, la suma del complemento a uno es equivalente a tomar la suma módulo$\mathbf{2_{dieciséis}}$y añadir cualquier desbordamiento de los bits de orden superior de vuelta a los bits de orden inferior. Este algoritmo proporciona una cobertura más uniforme de los datos por parte de los bits de suma de comprobación. De lo contrario, se pueden agregar, desbordar y perder dos bits de orden superior sin cambiar la suma. También hay otro beneficio. El complemento a uno tiene dos representaciones de cero, todos 0 y todos 1. Esto permite que un valor (p. ej., todos 0) indique que no hay suma de verificación, sin necesidad de otro campo. Durante décadas, siempre se ha asumido que los cuadros que se van a sumar contienen bits aleatorios. Todos los análisis de los algoritmos de suma de comprobación se han realizado bajo esta suposición. Inspección de datos reales por Partridge et al. (1995) ha demostrado que esta suposición es bastante errónea. Como consecuencia, los errores no detectados son en algunos casos mucho más comunes de lo que se pensaba anteriormente. La suma de comprobación de Internet, en particular, es eficaz y sencilla, pero proporciona una protección débil en algunos casos precisamente porque es una suma simple. No detecta la eliminación o adición de datos cero, ni el intercambio de partes del mensaje, y proporciona una protección débil contra los empalmes de mensajes en los que se juntan partes de dos paquetes. Puede parecer muy poco probable que estos errores ocurran mediante procesos aleatorios, pero son solo el tipo de errores que pueden ocurrir con hardware defectuoso. Una mejor opción es Puede parecer muy poco probable que estos errores ocurran mediante procesos aleatorios, pero son solo el tipo de errores que pueden ocurrir con hardware defectuoso. Una mejor opción es Puede parecer muy poco probable que estos errores ocurran mediante procesos aleatorios, pero son solo el tipo de errores que pueden ocurrir con hardware defectuoso. Una mejor opción es **Suma de comprobación de Fletcher**(Fletcher, 1982). Incluye un componente posicional, sumando el producto de los datos y su posición a la suma acumulada. Esto proporciona una detección más sólida de los cambios en la posición de los datos.

Aunque los dos esquemas anteriores a veces pueden ser adecuados en capas superiores, en la práctica, un tercer y más fuerte tipo de código de detección de errores se usa ampliamente en la capa de enlace: el **CDN**(**Verificación de redundancia cíclica**), también conocida como**código polinomial**. Los códigos de polinomios se basan en el tratamiento de cadenas de bits como representaciones de polinomios con coeficientes de 0 y 1 solamente. A **k bits**marco se considera como la lista de coeficientes para un polinomio con ***k***términos, que van desde$\mathbf{X_{k-1}}$para$\mathbf{X_0}$. Se dice que tal polinomio es de grado**k**−1. El bit de orden superior (más a la izquierda) es el coeficiente de$\mathbf{X_{k-1}}$, el siguiente bit es el coeficiente de$\mathbf{X_{k-2}}$, y así. Por ejemplo, 110001 tiene 6 bits y, por lo tanto, representa un polinomio de seis términos con coeficientes 1, 1, 0, 0, 0 y 1:$\mathbf{1}X_5+\mathbf{1}X_4+\mathbf{0}X_3+\mathbf{0}X_2+\mathbf{0}X_1+\mathbf{1}X_0$. La aritmética de polinomios se realiza módulo 2, según las reglas de la teoría algebraica de campos. No tiene acarreos para sumas ni préstamos para restas. Tanto la suma como la resta son idénticas al OR exclusivo. Por ejemplo:

10011011	00110011	11110000	01010101
+ 11001010	+ 11001101	− 10100110	− 10101111
01010001	11111110	01010110	11111010

La división larga se realiza exactamente de la misma manera que en binario, excepto que la resta se realiza de nuevo en módulo 2. Se dice que un divisor "entra" en un dividendo si el dividendo tiene tantos bits como el divisor. Cuando se emplea el método de código polinomial, el emisor y el receptor deben acordar un**polinomio generador**, G(*X*), por adelantado. Los bits de orden alto y bajo del generador deben ser 1. Para calcular el CRC para algún marco con ***metro***bits correspondientes al polinomio***METRO*(*X*)**, el marco debe ser más largo que el polinomio generador. La idea es agregar un CRC al final del marco de tal manera que el polinomio representado por el marco de suma de verificación sea divisible por***GRAMO*(*X*)**. Cuando el receptor obtiene el marco de suma de verificación, intenta dividirlo por***GRAMO*(*X*)**. Si hay un resto, ha habido un error de transmisión.

El algoritmo para calcular el CRC es el siguiente:

1. Deja***r***ser el grado de***GRAMO*(*X*)**. Adjuntar***r***cero bits al extremo de orden inferior del marco, por lo que ahora contiene***metro*+*r***bits y corresponde al polinomio***X*ʳ*METRO*(*X*)**.
2. Divida la cadena de bits correspondiente a**g(x)**en la cadena de bits correspondiente a***X*ʳ*METRO*(*X*)**, utilizando la división módulo 2.
3. Resta el resto (que siempre es***r***o menos bits) de la cadena de bits correspondiente a***X*ʳ*METRO*(X)**utilizando la resta de módulo 2. El resultado es la trama de suma de verificación que se va a transmitir. llama a su polinomio**T(x)**.

Figura 4.7ilustra el cálculo para un marco 1101011111 usando el generador**G(x)=x₄+x+1**.

debe quedar claro que**T(x)**es divisible (módulo 2) por**g(x)**. En cualquier problema de división, si disminuyes el dividendo por el resto, lo que sobra es divisible por el divisor. Por ejemplo, en base 10, si divides 210.278 entre 10.941, el resto es 2399. Si luego restas 2399 a 210.278, lo que sobra (207.879) es divisible por 10.941. Ahora analicemos el poder de este método. ¿Qué tipo de errores se detectarán? Imagine que se produce un error de transmisión, de modo que en lugar de la cadena de bits para**T(x)**llegando, **T(x)+E(x)** llega Cada 1 bit en**Ex)**corresponde a un bit que se ha invertido. Si hay k 1 bits en**Ex)**,***k***se han producido errores de un solo bit. Un error de ráfaga simple se caracteriza por un 1 inicial, una combinación de 0 y 1, y un 1 final, siendo todos los demás bits 0. Al recibir la trama con suma de verificación, el receptor la divide por**g(x)**; es decir, calcula**[T(x)+E(x)]/G(x)**.**T(x)/G(x)**es 0, por lo que el resultado del cálculo es simplemente**E(x)/G(x)**. Esos errores que corresponden a polinomios que contienen**g(x)**como un factor se deslizará; todos los demás errores serán detectados. Si ha habido un error de un solo bit, **E(x)=xᵢ**, donde**I**determina qué bit está en error. Si**g(x)**contiene dos o más términos, nunca se dividirá en**Ex)**, por lo que se detectarán todos los errores de un solo bit.

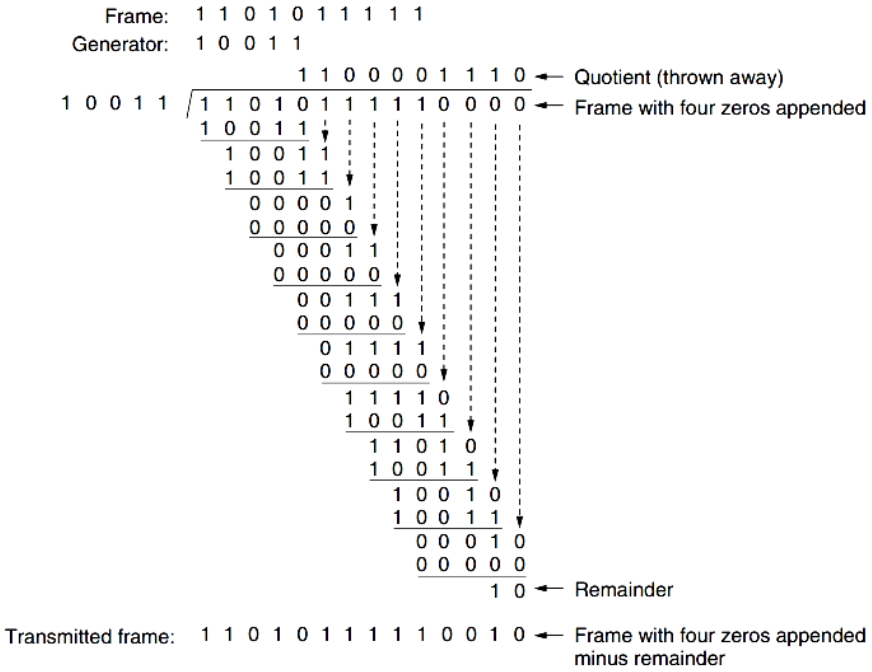


Fig 4.7: Ejemplo de cálculo del CRC

Si ha habido dos errores aislados de un solo bit,**E(x)=x$_I$+x$_j$**, donde **yo>j**. Alternativamente, esto se puede escribir como**E(x)=x$_j$(X$_{yo-j}$+1)**. Si asumimos que**g(x)**no es divisible por**X**, una condición suficiente para que se detecten todos los errores dobles es que**g(x)**no divide**X$_k$+1**para cualquier**k**hasta el valor máximo de**yo-j**(es decir, hasta la longitud máxima del marco). Se conocen polinomios simples de bajo grado que brindan protección a marcos largos. Por ejemplo,**X$_{15}$+x$_{14}$+1**no se dividirá **X$_k$+1**por cualquier valor de**k**por debajo de 32.768. Si hay un número impar de bits erróneos,**EX)**contiene un número impar de términos (p. ej., x$_5$+x$_2$+1, pero no x$_2$+1). Curiosamente, ningún polinomio con un número impar de términos tiene**x+1**como factor en el sistema módulo 2. Haciendo**x+1**un factor de**g(x)**, podemos detectar todos los errores con un número impar de bits invertidos.

Finalmente, y lo que es más importante, un código polinomial con***r*** los bits de verificación detectarán todos los errores de ráfaga de longitud≤r. Un error de ráfaga de longitud***k***puede ser representado por**X$_I$(X$_{k-1}$+...+1)**, donde**I**determina a qué distancia del extremo derecho de la trama recibida se encuentra la ráfaga. Si **g(x)**contiene un**X$_0$**término, no tendrá**X$_I$**como un factor, por lo que si el grado de la expresión entre paréntesis es menor que el grado de **g(x)**, el resto nunca puede ser cero. Si la longitud de la ráfaga es**r+1**, el resto de la división por**g(x)**será cero si y solo si la ráfaga es idéntica a**g(x)**. Por definición de una ráfaga, el primer y el último bit deben ser 1, por lo que si coincide depende de la**r**−1 pedacitos intermedios. Si todas las combinaciones se consideran igualmente probables, la probabilidad de que se acepte como válido un marco incorrecto de este tipo es**½$^{r-1}$**. También se puede demostrar que cuando una ráfaga de error dura más de **r+1**bits o cuando ocurren varias ráfagas más cortas, la probabilidad de que una trama defectuosa pase desapercibida es**½^r**, asumiendo que todos los patrones de bits son igualmente probables.

Ciertos polinomios se han convertido en estándares internacionales. El utilizado en IEEE 802 siguió el ejemplo de Ethernet y es

X$_{32}$+X$_{26}$+X$_{23}$+X$_{22}$+X$_{dieciséis}$+X$_{12}$+X$_{11}$+X$_{10}$+X$_8$+X$_7$+X$_5$+X$_4$+X$_2$+X$_1$+1

Entre otras propiedades deseables, tiene la propiedad de que detecta todas las ráfagas de longitud 32 o menos y todas las ráfagas que afectan a un número impar de bits. Se ha utilizado ampliamente desde la década de 1980. Sin embargo, esto no significa que sea la mejor opción. Usando una búsqueda computacional exhaustiva, Castagnoli et al. (1993) y Koopman (2002) encontraron los mejores CRC. Estos CRC tienen una distancia de Hamming de 6 para tamaños de mensaje típicos, mientras que el estándar IEEE CRC-32 tiene una distancia de Hamming de solo 4. Aunque el cálculo requerido para calcular el CRC puede parecer complicado, es fácil calcular y verificar los CRC en hardware. con circuitos simples de registro de desplazamiento (Peterson y Brown, 1961). En la práctica, casi siempre se utiliza este hardware. Docenas de estándares de red incluyen varios CRC, incluidas prácticamente todas las LAN (p. ej., Ethernet, 802.11) y enlaces punto a punto (p. ej.,

REVISA TU PROGRESO

1. Complete los espacios en blanco:

(a) La capa _____ es la capa de protocolo que transfiere datos entre nodos de red adyacentes en una red de área amplia.

(b) El protocolo de control de flujo _____ tiene un mecanismo incorporado que limita la velocidad de transmisión de datos.

(c) Los códigos _____ incluyen suficiente información redundante para permitir que el receptor deduzca cuáles deben haber sido los datos transmitidos.

(d) Los códigos _____ incluyen solo la redundancia suficiente para permitir que el receptor deduzca que ha ocurrido un error y solicite una retransmisión.

(e) Las propiedades de detección y corrección de errores de un código de bloque dependen de su _____.

(f) En un _____, un codificador procesa una secuencia de bits de entrada y genera una secuencia de bits de salida.

(g) _____ se basan en el hecho de que cada polinomio de n grados está determinado por n+1 puntos.

(h) _____ son prácticos para tamaños de bloques grandes y tienen excelentes capacidades de corrección de errores.

(i) El _____ generalmente se coloca al final del mensaje.

(j) _____ se basan en el tratamiento de cadenas de bits como representaciones de polinomios con coeficientes de 0 y 1 únicamente.

4.7 PROTOCOLOS DE ENLACE DE DATOS

Para introducir el tema de los protocolos, comenzaremos analizando tres protocolos de complejidad creciente. Antes de que analicemos los protocolos, es útil hacer explícitos algunos de los supuestos que subyacen al modelo de comunicación. Para empezar, asumimos que la capa física, la capa de enlace de datos y la capa de red son procesos independientes que se comunican pasando mensajes de un lado a otro. Una implementación común se muestra en**Figura 4.8**. El proceso de la capa física y algunos de los procesos de la capa de enlace de datos se ejecutan en un hardware dedicado llamado NIC (**Tarjeta de interfaz de red**). El resto del proceso de la capa de enlace y el proceso de la capa de red se ejecutan en la CPU principal como parte del sistema operativo, y el software para el proceso de la capa de enlace suele adoptar la forma de un controlador de dispositivo. Sin embargo, también son posibles otras implementaciones (p. ej., tres procesos descargados en un hardware dedicado denominado**acelerador de red**, o tres procesos que se ejecutan en la CPU principal en una proporción definida por software). En realidad, la implementación preferida cambia de una década a otra con compensaciones tecnológicas. En cualquier caso, tratar las tres capas como procesos separados hace que la discusión sea conceptualmente más clara y también sirve para enfatizar la independencia de las capas.

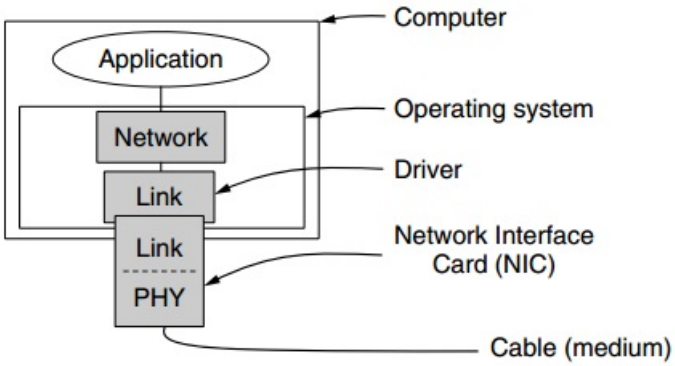


Fig. 4.8 Implementación del enlace físico, de datos y de red capas

Otra suposición clave es que la máquina A desea enviar una gran cantidad de datos a la máquina B mediante un servicio confiable orientado a la conexión. Se supone que A tiene un suministro infinito de datos listos para enviar y nunca tiene que esperar a que se produzcan los datos. En cambio, cuando la capa de enlace de datos de A solicita datos, la capa de red siempre puede cumplir de inmediato. También asumimos que las máquinas no fallan. Es decir, estos protocolos se ocupan de los errores de comunicación, pero no de los problemas causados por las fallas y reinicios de las computadoras. En lo que respecta a la capa de enlace de datos, el paquete que pasa a través de la interfaz desde la capa de red son datos puros, cuyos bits se entregarán a la capa de red de destino. El hecho de que la capa de la red de destino pueda interpretar parte del paquete como un encabezado no le preocupa a la capa de enlace de datos. Cuando la capa de enlace de datos acepta un paquete, lo encapsula en una trama al agregarle un encabezado y un tráiler de enlace de datos. Por lo tanto, un marco consta de

un paquete incrustado, alguna información de control (en el encabezado) y una suma de verificación (en el tráiler). Luego, la trama se transmite a la capa de enlace de datos en la otra máquina. Asumiremos que existen procedimientos bibliotecarios adecuados*a la capa física*para enviar un marco y*de la capa física*para recibir un marco. Estos procedimientos calculan y agregan o verifican la suma de verificación (que generalmente se realiza en hardware) para que no tengamos que preocuparnos por ello como parte de los protocolos que desarrollamos en esta sección, por ejemplo. Inicialmente, el receptor no tiene nada que hacer. Simplemente se sienta a esperar que suceda algo. Indicamos que la capa de enlace de datos está esperando que algo suceda por la llamada al procedimiento***esperar_por_evento* (&*evento*)**. Este procedimiento solo regresa cuando algo ha sucedido (por ejemplo, ha llegado un marco). Al regresar, la variable *evento*cuenta lo sucedido. El conjunto de posibles eventos difiere para los diversos protocolos a describir y se definirá por separado para cada protocolo. Tenga en cuenta que en una situación más realista, la capa de enlace de datos no se quedará en un bucle cerrado esperando un evento, como hemos sugerido, sino que recibirá una interrupción, lo que hará que detenga lo que sea que esté haciendo y se encargue de la entrada. marco. Sin embargo, para simplificar, ignoraremos todos los detalles de la actividad paralela dentro de la capa de enlace de datos y supondremos que se dedica a tiempo completo a manejar solo nuestro canal.

```
#define MAX_PKT 1024                                    /* determines packet size in bytes */

typedef enum {false, true} boolean;                     /* boolean type */
typedef unsigned int seq_nr;                            /* sequence or ack numbers */
typedef struct {unsigned char data[MAX_PKT];} packet;   /* packet definition */
typedef enum {data, ack, nak} frame_kind;               /* frame_kind definition */

typedef struct {                                        /* frames are transported in this layer */
  frame_kind kind;                                      /* what kind of frame is it? */
  seq_nr seq;                                           /* sequence number */
  seq_nr ack;                                           /* acknowledgement number */
  packet info;                                          /* the network layer packet */
} frame;


/* Wait for an event to happen; return its type in event. */
void wait_for_event(event_type *event);

/* Fetch a packet from the network layer for transmission on the channel. */
void from_network_layer(packet *p);

/* Deliver information from an inbound frame to the network layer. */
void to_network_layer(packet *p);

/* Go get an inbound frame from the physical layer and copy it to r. */
void from_physical_layer(frame *r);

/* Pass the frame to the physical layer for transmission. */
void to_physical_layer(frame *s);

/* Start the clock running and enable the timeout event. */
void start_timer(seq_nr k);

/* Stop the clock and disable the timeout event. */
void stop_timer(seq_nr k);

/* Start an auxiliary timer and enable the ack_timeout event. */
void start_ack_timer(void);

/* Stop the auxiliary timer and disable the ack_timeout event. */
void stop_ack_timer(void);

/* Allow the network layer to cause a network_layer_ready event. */
void enable_network_layer(void);

/* Forbid the network layer from causing a network_layer_ready event. */
void disable_network_layer(void);

/* Macro inc is expanded in-line: increment k circularly. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

Fig 4.9: Algunas definiciones necesarias en los protocolos a seguir. Estas definiciones se encuentran en el archivo protocol.h

Cuando una trama llega al receptor, se vuelve a calcular la suma de comprobación. Si la suma de verificación en la trama es incorrecta (es decir, hubo un error de transmisión), se informa a la capa de enlace de datos (***evento=cksum err***). Si la trama entrante llegó sin daños, también se informa a la capa de enlace de datos (***evento= frame_arrival***) para que pueda adquirir el marco para la inspección usando***de_la_capa_física***. Tan pronto como la capa de enlace de datos de recepción ha adquirido una trama en buen estado, verifica la información de control en el encabezado y, si todo está bien, pasa la porción del paquete a la capa de red. Bajo ninguna circunstancia se asigna un encabezado de trama a una capa de red. Hay una buena razón por la cual la capa de red nunca debe recibir ninguna parte del encabezado de la trama: para mantener los protocolos de enlace de red y de datos completamente separados. Siempre que la capa de red no sepa nada sobre el protocolo de enlace de datos o el formato de la trama, estas cosas se pueden cambiar sin necesidad de cambios en el software de la capa de red. Esto sucede cada vez que se instala una nueva NIC en una computadora.**Figura 4.9**muestra algunas declaraciones (en C) comunes a muchos de los protocolos que se discutirán más adelante. Allí se definen cinco estructuras de datos: ***booleano***, ***seq_nr***,***paquete***,***marco_tipo***, y***marco***. A*booleano*es un tipo enumerado y puede tomar los valores*cierto*y*falso*. A *seq_nr*es un pequeño número entero que se usa para numerar los cuadros para que podamos diferenciarlos. Estos números de secuencia van desde 0 hasta e incluyendo*SEC_MAX*, que se define en cada protocolo que lo necesita. A*paquete*es la unidad de información intercambiada entre la capa de red y la capa de enlace de datos en la misma máquina, o entre pares de la capa de red. En nuestro modelo siempre contiene *MAX_PKT* bytes, pero de manera más realista sería de longitud variable. Un marco se compone de cuatro campos:***tipo***,***secuencia***, ***reconocer***, y***información***, los tres primeros de los cuales contienen información de control y el último de los cuales puede contener datos reales para ser transferidos. Estos campos de control se denominan colectivamente encabezado de trama. los*tipo*El campo indica si hay datos en el marco, porque algunos de los protocolos distinguen los marcos que contienen solo información de control de aquellos que también contienen datos. los*secuencia*y*reconocer*los campos se utilizan para números de secuencia y acuses de recibo, respectivamente. los*información* el campo de una trama de datos contiene un solo paquete; los*información*no se utiliza el campo de un marco de control. Una implementación más realista usaría una longitud variable*información*campo, omitiéndolo por completo para los marcos de control.

Nuevamente, es importante comprender la relación entre un paquete y una trama. La capa de red crea un paquete tomando un mensaje de la capa de transporte y agregándole el encabezado de la capa de red. Este paquete se pasa a la capa de enlace de datos para su inclusión en el campo de información de una trama saliente. Cuando la trama llega al destino, la capa de enlace de datos extrae el paquete de la trama y pasa el paquete a la capa de red. De esta manera, la capa de red puede actuar como si las máquinas pudieran intercambiar paquetes directamente. También se enumeran varios procedimientos en**Figura 4.9** . Estas son rutinas de biblioteca cuyos detalles dependen de la implementación. El procedimiento*esperar_por_evento*se sienta en un círculo cerrado esperando que suceda algo. Los procedimientos

*a_capa_de_red*y *from_network_layer*son utilizados por la capa de enlace de datos para pasar paquetes a la capa de red y aceptar paquetes de la capa de red, respectivamente. Tenga en cuenta que de la capa física a la capa física pasan tramas entre la capa de enlace de datos y la capa física. En otras palabras,*a_capa_de_red*y *from_network_layer*se ocupan de la interfaz entre las capas 2 y 3, mientras que de la capa física y la capa física se ocupan de la interfaz entre las capas 1 y 2.

En la mayoría de los protocolos, asumimos que el canal no es confiable y pierde tramas completas de vez en cuando. Para poder recuperarse de tales calamidades, la capa de enlace de datos de envío debe iniciar un temporizador o reloj interno cada vez que envía una trama. Si no se ha recibido respuesta dentro de un determinado intervalo de tiempo predeterminado, el reloj expira y la capa de enlace de datos recibe una señal de interrupción. En nuestros protocolos esto se maneja permitiendo el procedimiento *esperar_por_evento*regresar*evento=se acabó el tiempo*. Los procedimientos*inicio_temporizador* y*stop_timer*encender y apagar el temporizador, respectivamente. Los eventos de tiempo de espera solo son posibles cuando el temporizador está funcionando y antes *stop_timer*se llama. Está expresamente permitido llamar*inicio_temporizador*mientras el temporizador está funcionando; dicha llamada simplemente restablece el reloj para provocar el siguiente tiempo de espera después de que haya transcurrido un intervalo completo del temporizador (a menos que se restablezca o se apague). Los procedimientos*start_ack_timer*y *stop_ack_timer* controlar un temporizador auxiliar utilizado para generar reconocimientos bajo ciertas condiciones. Los procedimientos *enable_network_layer*y *desactivar_capa_de_red*se utilizan en los protocolos más sofisticados, donde ya no asumimos que la capa de red siempre tiene paquetes para enviar. Cuando la capa de enlace de datos habilita la capa de red, se permite que la capa de red interrumpa cuando tiene que enviar un paquete. Lo indicamos con *evento=capa_de_red_lista*. Cuando la capa de red está deshabilitada, es posible que no cause tales eventos. Al tener cuidado acerca de cuándo habilita y deshabilita su capa de red, la capa de enlace de datos puede evitar que la capa de red la inunde con paquetes para los que no tiene espacio de búfer. Los números de secuencia de cuadros siempre están en el rango de 0 a*SEC_MAX*, donde*SEC_MAX*es diferente para los diferentes protocolos. Con frecuencia es necesario avanzar un número de secuencia en 1 circularmente (es decir,*SEC_MAX*va seguido de 0). la macro*C* realiza este incremento. Se ha definido como una macro porque se usa en línea dentro de la ruta crítica. El factor que limita el rendimiento de la red suele ser el procesamiento de protocolos, por lo que definir operaciones simples como esta como macros no afecta la legibilidad del código, pero mejora el rendimiento.

4.7.1 PARAR Y ESPERAR ARQ

Ahora abordaremos el problema de evitar que el remitente inunde al receptor con tramas más rápido de lo que este último puede procesarlas. Esta situación puede ocurrir fácilmente en la práctica, por lo que poder prevenirla es de gran importancia. Sin embargo, todavía se supone que el canal de comunicación está libre de errores y el tráfico de datos sigue siendo símplex. Una solución es construir el receptor para que sea lo suficientemente potente como para procesar un flujo continuo de tramas consecutivas. Debe tener suficiente almacenamiento en búfer y capacidades de procesamiento para ejecutarse a la velocidad de línea y debe poder pasar los marcos que son

recibida en la capa de red con la suficiente rapidez. Sin embargo, esta es una solución en el peor de los casos. Requiere hardware dedicado y puede ser un desperdicio de recursos si la utilización del enlace es mayormente baja. Además, simplemente traslada el problema de tratar con un remitente que es demasiado rápido a otra parte; en este caso a la capa de red. Una solución más general a este problema es hacer que el receptor proporcione información al remitente. Después de haber pasado un paquete a su capa de red, el receptor envía una pequeña trama ficticia al remitente que, en efecto, le da permiso al remitente para transmitir la siguiente trama. Después de haber enviado una trama, el protocolo requiere que el remitente espere hasta que llegue la pequeña trama ficticia (acuse de recibo). Este retraso es un ejemplo simple de un protocolo de control de flujo.**detener y esperar**.**Figura 4.10**da un ejemplo de un protocolo de parada y espera símplex. Aunque el tráfico de datos en este ejemplo es símplex, y solo va del remitente al receptor, las tramas viajan en ambas direcciones. En consecuencia, el canal de comunicación entre las dos capas de enlace de datos debe ser capaz de transferir información bidireccional. Sin embargo, este protocolo implica una estricta alternancia de flujo: primero el remitente envía una trama, luego el receptor envía una trama, luego el remitente envía otra trama, luego el receptor envía otra, y así sucesivamente. Un canal físico semidúplex sería suficiente aquí. Como en el protocolo 1, el remitente comienza por obtener un paquete de la capa de red, lo usa para construir una trama y lo envía a su destino. Pero ahora, a diferencia del protocolo 1, el remitente debe esperar hasta que llegue una trama de reconocimiento antes de regresar y obtener el siguiente paquete de la capa de red. La capa de enlace de datos de envío ni siquiera necesita inspeccionar la trama entrante, ya que solo hay una posibilidad. La trama entrante es siempre un acuse de recibo. La única diferencia entre el receptor 1 y el receptor 2 es que después de entregar un paquete a la capa de red, el receptor 2 envía una trama de reconocimiento al remitente antes de ingresar nuevamente al ciclo de espera. Debido a que solo es importante la llegada de la trama al remitente, no su contenido, el receptor no necesita poner ninguna información en particular en ella. La única diferencia entre el receptor 1 y el receptor 2 es que después de entregar un paquete a la capa de red, el receptor 2 envía una trama de reconocimiento al remitente antes de ingresar nuevamente al ciclo de espera. Debido a que solo es importante la llegada de la trama al remitente, no su contenido, el receptor no necesita poner ninguna información en particular en ella. La única diferencia entre el receptor 1 y el receptor 2 es que después de entregar un paquete a la capa de red, el receptor 2 envía una trama de reconocimiento al remitente antes de ingresar nuevamente al ciclo de espera. Debido a que solo es importante la llegada de la trama al remitente, no su contenido, el receptor no necesita poner ninguna información en particular en ella.

```
/* Protocol 2 (Stop-and-wait) also provides for a one-directional flow of data from
   sender to receiver. The communication channel is once again assumed to be error
   free, as in protocol 1. However, this time the receiver has only a finite buffer
   capacity and a finite processing speed, so the protocol must explicitly prevent
   the sender from flooding the receiver with data faster than it can be handled. */

typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender2(void)
{
  frame s;                              /* buffer for an outbound frame */
  packet buffer;                        /* buffer for an outbound packet */
  event_type event;                     /* frame_arrival is the only possibility */

  while (true) {
      from_network_layer(&buffer);      /* go get something to send */
      s.info = buffer;                  /* copy it into s for transmission */
      to_physical_layer(&s);            /* bye-bye little frame */
      wait_for_event(&event);           /* do not proceed until given the go ahead */
  }
}

void receiver2(void)
{
  frame r, s;                           /* buffers for frames */
  event_type event;                     /* frame_arrival is the only possibility */
  while (true) {
      wait_for_event(&event);           /* only possibility is frame_arrival */
      from_physical_layer(&r);          /* go get the inbound frame */
      to_network_layer(&r.info);        /* pass the data to the network layer */
      to_physical_layer(&s);            /* send a dummy frame to awaken sender */
  }
}
```

Fig. 4.10: Un protocolo simplex de parada y espera

4.7.2 VOLVER N ARQ

Hasta ahora hemos asumido tácitamente que el tiempo de transmisión requerido para que una trama llegue al receptor más el tiempo de transmisión para que regrese el acuse de recibo es insignificante. A veces esta suposición es claramente falsa. En estas situaciones, el largo tiempo de ida y vuelta puede tener implicaciones importantes para la eficiencia de la utilización del ancho de banda. Como ejemplo, considere un canal satelital de 50 kbps con un retraso de propagación de ida y vuelta de 500 ms. Imaginemos intentar usar el protocolo 4 para enviar tramas de 1000 bits a través del satélite. En t=0, el remitente comienza a enviar el primer cuadro. En t = 20 ms, la trama se ha enviado por completo. No hasta que t = 270 ms la trama llega completamente al receptor, y no hasta que t = 520 ms el acuse de recibo llega al remitente, en el mejor de los casos. Esto significa que el remitente fue bloqueado 500/520 o el 96% del tiempo. En otras palabras, solo se utilizó el 4% del ancho de banda disponible. Claramente, la combinación de un tiempo de tránsito largo, un ancho de banda alto y una longitud de trama corta es desastrosa en términos de eficiencia.

El problema descrito aquí puede verse como una consecuencia de la regla que requiere que un remitente espere un reconocimiento antes de enviar otra trama. Si relajamos esa restricción, se puede lograr una eficiencia mucho mayor. Básicamente, la solución radica en permitir que el remitente transmita hasta *w* marcos

antes de bloquear, en lugar de solo 1. Con una opción lo suficientemente grande de ***w*** el remitente podrá transmitir tramas continuamente ya que los reconocimientos llegarán para las tramas anteriores antes de que la ventana se llene, evitando que el remitente se bloquee. Para encontrar un valor apropiado para ***w*** necesitamos saber cuántos marcos caben dentro del canal a medida que se propagan del emisor al receptor. Esta capacidad está determinada por el ancho de banda en bits/seg multiplicado por el tiempo de tránsito unidireccional, o el producto ancho de banda-retardo del enlace. Podemos dividir esta cantidad por el número de bits en un cuadro para expresarlo como un número de cuadros. Llame a esta cantidad BD. Luego ***w*** debe establecerse en **2*BD*+1**. El doble de la demora del ancho de banda es el número de tramas que pueden estar pendientes si el remitente envía tramas continuamente cuando se considera el tiempo de ida y vuelta para recibir un reconocimiento. El ''+1'' se debe a que no se enviará una trama de reconocimiento hasta que se reciba una trama completa.

Para el enlace de ejemplo con un ancho de banda de 50 kbps y un tiempo de tránsito unidireccional de 250 mseg, el producto ancho de banda-retardo es 12,5 kbit o 12,5 tramas de 1000 bits cada una. ***2HAB+1*** es entonces 26 fotogramas. Suponga que el remitente comienza a enviar el marco 0 como antes y envía un nuevo marco cada 20 mseg. Cuando haya terminado de enviar 26 tramas, en t=520 mseg, habrá llegado el acuse de recibo de la trama 0. A partir de entonces, los acuses de recibo llegarán cada 20 milisegundos, por lo que el remitente siempre obtendrá permiso para continuar justo cuando lo necesite. A partir de ahí, siempre quedarán pendientes 25 o 26 fotogramas no reconocidos. Dicho de otro modo, el tamaño máximo de ventana del remitente es 26. Para tamaños de ventana más pequeños, la utilización del enlace será inferior al 100 %, ya que el remitente se bloqueará en ocasiones. Podemos escribir la utilización como la fracción de tiempo que el remitente no está bloqueado:

utilización del enlace ≤ con (1+2BD)

Este valor es un límite superior porque no permite ningún tiempo de procesamiento de tramas y trata la trama de reconocimiento como si tuviera una longitud cero, ya que suele ser corta. La ecuación muestra la necesidad de tener una ventana grande ***w*** siempre que el producto ancho de banda sea grande. Si el retraso es alto, el emisor agotará rápidamente su ventana incluso para un ancho de banda moderado, como en el ejemplo del satélite. Si el ancho de banda es alto, incluso con un retraso moderado, el remitente agotará su ventana rápidamente a menos que tenga una ventana grande (p. ej., un enlace de 1 Gbps con un retraso de 1 ms tiene 1 megabit). Con parar y esperar para que **w=1**, si hay incluso un retardo de propagación equivalente a una trama, la eficiencia será inferior al 50 %. Esta técnica de mantener varios fotogramas en vuelo es un ejemplo de canalización. La canalización de fotogramas a través de un canal de comunicación poco fiable plantea algunos problemas graves. Primero, ¿qué sucede si se daña o se pierde un cuadro en medio de una transmisión larga? Una gran cantidad de tramas sucesivas llegarán al receptor antes de que el remitente descubra que algo anda mal. Cuando una trama dañada llega al receptor, obviamente debe descartarse, pero ¿qué debe hacer el receptor con todas las tramas correctas que le siguen? Recuerde que la capa de enlace de datos de recepción está obligada a entregar los paquetes a la capa de red en secuencia.

Hay dos enfoques básicos disponibles para tratar los errores en presencia de canalización, los cuales se muestran en **Figura 4.11**.

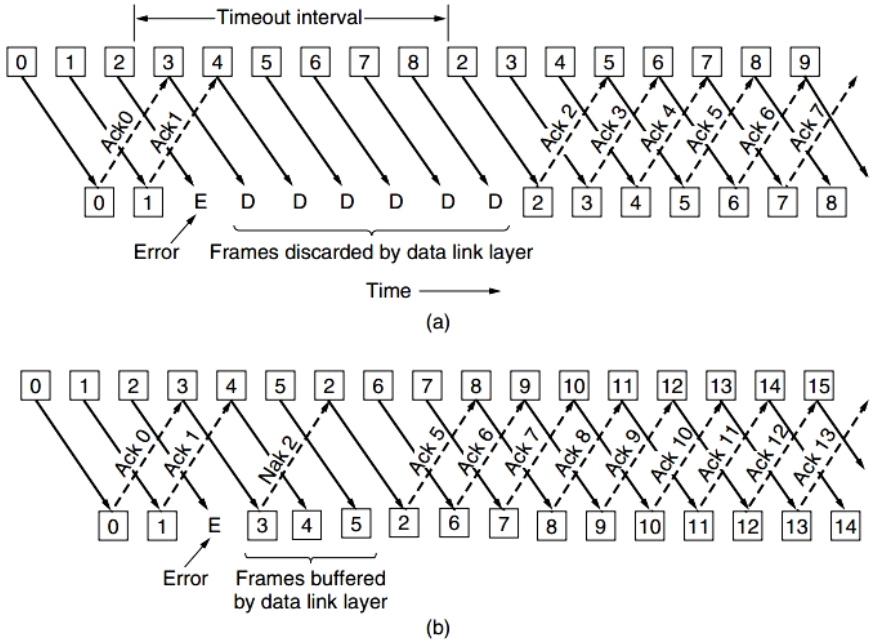


FIG 4.11: Canalización y recuperación de errores. Efecto de un error cuando (a) el tamaño de la ventana del receptor es 1 y (b) la ventana del receptor el tamaño es grande

Una opción, llamada**Regresar-N**, es que el receptor simplemente descarte todas las tramas subsiguientes, sin enviar acuses de recibo por las tramas descartadas. Esta estrategia corresponde a una*recibir_ventana* de tamaño 1. En otras palabras, la capa de enlace de datos se niega a aceptar cualquier trama excepto la siguiente que debe entregar a la capa de red. Si la ventana del remitente se llena antes de que se agote el tiempo, la canalización comenzará a vaciarse. Eventualmente, el remitente expirará y retransmitirá todas las tramas no reconocidas en orden, comenzando con la dañada o perdida. Este enfoque puede desperdiciar mucho ancho de banda si la tasa de error es alta. En**Figura 4.11 (b)**vemos go-back-n para el caso en que la ventana del receptor es grande. Las tramas 0 y 1 se reciben y confirman correctamente. El marco 2, sin embargo, está dañado o perdido. El remitente, sin darse cuenta de este problema, continúa enviando tramas hasta que expira el temporizador para la trama 2. Luego retrocede hasta el cuadro 2 y comienza de nuevo con él, enviando 2, 3, 4, etc., todo de nuevo. La siguiente estrategia utilizada es el protocolo de repetición selectiva y se discutirá en la siguiente sección.

4.7.3 ARQ DE REPETICIÓN SELECTIVA

El protocolo go-back-n funciona bien si los errores son raros, pero si la línea es deficiente, desperdicia mucho ancho de banda en las tramas retransmitidas. Una estrategia alternativa, la**repetición selectiva**protocolo, es permitir que el receptor acepte y almacene en búfer las tramas que siguen a una dañada o perdida. En este protocolo, tanto el emisor como el receptor mantienen una ventana de números de secuencia pendientes y aceptables, respectivamente. El tamaño de la ventana del remitente comienza en 0 y crece hasta un máximo predefinido. La ventana del receptor, por el contrario, siempre tiene un tamaño fijo e igual al máximo predeterminado. los

El receptor tiene un búfer reservado para cada número de secuencia dentro de su ventana fija. Asociado con cada búfer hay un bit que indica si el búfer está lleno o vacío. Cada vez que llega un cuadro, la función between verifica su número de secuencia para ver si se encuentra dentro de la ventana. En caso afirmativo y si aún no se ha recibido, se acepta y almacena. Esta acción se lleva a cabo independientemente de si la trama contiene o no el siguiente paquete esperado por la capa de red. Por supuesto, debe mantenerse dentro de la capa de enlace de datos y no pasarse a la capa de red hasta que todas las tramas con números más bajos ya se hayan entregado a la capa de red en el orden correcto. La recepción no secuencial introduce restricciones adicionales en los números de secuencia de tramas en comparación con los protocolos en los que las tramas solo se aceptan en orden. Podemos ilustrar el problema más fácilmente con un ejemplo. Supongamos que tenemos un número de secuencia de 3 bits, de modo que el remitente puede transmitir hasta siete tramas antes de tener que esperar un acuse de recibo. El remitente ahora transmite las tramas 0 a 6. La ventana del receptor le permite aceptar cualquier trama con un número de secuencia entre 0 y 6 inclusive. Los siete marcos llegan correctamente, por lo que el receptor los reconoce y avanza su ventana para permitir la recepción de 7, 0, 1, 2, 3, 4 o 5. Los siete búferes se marcan como vacíos. Es en este punto cuando ocurre el desastre en forma de un rayo que golpea el poste de teléfono y borra todos los reconocimientos. El protocolo debería funcionar correctamente a pesar de este desastre. El remitente eventualmente expira y retransmite la trama 0. Cuando esta trama llega al receptor, se hace una verificación para ver si cae dentro de la ventana del receptor. Desafortunadamente, el marco 0 está dentro de la nueva ventana, por lo que se acepta como un nuevo marco. El receptor también envía un acuse de recibo (superpuesto) para el cuadro 6, ya que se han recibido del 0 al 6. El remitente se alegra de saber que todas sus tramas transmitidas realmente llegaron correctamente, por lo que avanza su ventana e inmediatamente envía las tramas 7, 0, 1, 2, 3, 4 y 5. La trama 7 será aceptada por el receptor y su El paquete se pasará directamente a la capa de red. Inmediatamente después, la capa de enlace de datos receptora verifica si ya tiene una trama 0 válida, descubre que la tiene y pasa el paquete almacenado en búfer antiguo a la capa de red como si fuera un paquete nuevo. En consecuencia, la capa de red recibe un paquete incorrecto y el protocolo falla.

La esencia del problema es que después de que el receptor avanzó su ventana, el nuevo rango de números de secuencia válidos se superpuso al anterior. En consecuencia, el siguiente lote de tramas puede ser duplicados (si se perdieron todos los reconocimientos) o nuevos (si se recibieron todos los reconocimientos). El receptor no tiene manera de distinguir estos dos casos. La salida a este dilema radica en asegurarse de que después de que el receptor haya avanzado su ventana, no haya superposición con la ventana original. Para garantizar que no haya superposición, el tamaño máximo de la ventana debe ser como máximo la mitad del rango de los números de secuencia.

4.7.4 PROTOCOLO HDLC

Control de enlace de datos de alto nivel(HDLC) es un protocolo de capa de enlace de datos síncrono transparente de código orientado a bits desarrollado por

Organización Internacional de Normalización (ISO). HDLC proporciona servicios tanto orientados a la conexión como sin conexión. HDLC se puede usar para conexiones punto a multipunto, pero ahora se usa casi exclusivamente para conectar un dispositivo a otro, usando lo que se conoce como**Modo equilibrado asíncrono**(AMB). Las tramas HDLC se pueden transmitir a través de enlaces síncronos o asíncronos. Esos enlaces no tienen ningún mecanismo para marcar el comienzo o el final de un cuadro, por lo que se debe identificar el principio y el final de cada cuadro. Esto se hace mediante el uso de un delimitador de cuadro, o indicador, que es una secuencia única de bits que se garantiza que no se verá dentro de un cuadro. Esta secuencia es '01111110' o, en notación hexadecimal, 0x7E. Cada cuadro comienza y termina con un delimitador de cuadro. Un delimitador de cuadro al final de un cuadro también puede marcar el comienzo del cuadro siguiente. Una secuencia de 7 o más bits 1 consecutivos dentro de una trama hará que la trama se aborte.

El contenido de una trama HDLC se muestra en la siguiente tabla:

Flag	Address	Control	Information	FCS	Flag
8 bits	8 or more bits	8 or 16 bits	Variable length, 0 or more bits	16 or 32 bits	8 bits

Tenga en cuenta que el indicador final de un cuadro puede ser (pero no tiene que ser) el indicador inicial (inicio) del cuadro siguiente. Los datos generalmente se envían en múltiplos de 8 bits, pero solo algunas variantes lo requieren; otros teóricamente permiten alineaciones de datos en límites que no sean de 8 bits. los**secuencia de verificación de fotogramas**(FCS) es un CRC-CCITT de 16 bits o un CRC-32 de 32 bits calculado sobre los campos Dirección, Control e Información. Proporciona un medio por el cual el receptor puede detectar errores que pueden haberse inducido durante la transmisión de la trama, como bits perdidos, bits invertidos y bits extraños. Sin embargo, dado que los algoritmos utilizados para calcular la FCS son tales que la probabilidad de que ciertos tipos de errores de transmisión pasen desapercibidos aumenta con la longitud de los datos que se verifican en busca de errores, la FCS puede limitar implícitamente el tamaño práctico de la trama. Si el cálculo del FCS del receptor no coincide con el del remitente, lo que indica que la trama contiene errores, el receptor puede enviar un paquete de reconocimiento negativo al remitente o no enviar nada. Después de recibir un paquete de reconocimiento negativo o de esperar un paquete de reconocimiento positivo, el remitente puede retransmitir la trama fallida. El FCS se implementó porque muchos de los primeros enlaces de comunicación tenían una tasa de error de bits relativamente alta, y el FCS podía calcularse fácilmente mediante un circuito o software simple y rápido. Los esquemas de corrección de errores de reenvío más efectivos ahora son ampliamente utilizados por otros protocolos

Hay tres tipos fundamentales de tramas HDLC.

- **Marcos de información**, o I-frames, transportan datos de usuario desde la capa de red. Además, también pueden incluir información de control de errores y flujo respaldada en datos.
- **Marcos de supervisión**, o S-frames, se utilizan para controlar el flujo y los errores siempre que la superposición sea imposible o inapropiada, como cuando una estación no tiene datos para enviar. Los marcos S no tienen campos de información.

- **Fotogramas sin numerar**, o U-frames, se utilizan para diversos fines, incluida la gestión de enlaces. Algunos marcos en U contienen un campo de información, según el tipo.

4.7.5 PROTOCOLO PUNTO A PUNTO

En la creación de redes, el**Protocolo punto a punto**(PPP) es un protocolo de enlace de datos comúnmente utilizado para establecer una conexión directa entre dos nodos de red. Puede proporcionar autenticación de conexión, cifrado de transmisión (usando ECP, RFC 1968) y compresión. Se utiliza en muchos tipos de redes físicas, incluido el cable serial, la línea telefónica, la línea troncal, el teléfono celular, los enlaces de radio especializados y los enlaces de fibra óptica como SONET. PPP también se utiliza en conexiones de acceso a Internet (ahora comercializadas como "banda ancha"). Los proveedores de servicios de Internet (ISP) han utilizado PPP para el acceso telefónico de los clientes a Internet, ya que los paquetes IP no se pueden transmitir a través de una línea de módem por sí mismos, sin algún protocolo de enlace de datos. Dos derivados de PPP, Protocolo punto a punto sobre Ethernet (PPPoE) y Protocolo punto a punto sobre ATM (PPPoA), son utilizados más comúnmente por los proveedores de servicios de Internet (ISP) para establecer una conexión de servicio de Internet de línea de suscriptor digital (DSL) con los clientes. PPP se usa comúnmente como un protocolo de capa de enlace de datos para la conexión a través de circuitos síncronos y asíncronos, donde ha reemplazado en gran medida al antiguo Protocolo de Internet de línea en serie (SLIP) y los estándares obligatorios de la compañía telefónica (como el Protocolo de acceso de enlace, balanceado (LAPB) en el conjunto de protocolos X.25). PPP fue diseñado para funcionar con numerosos protocolos de capa de red, incluido el Protocolo de Internet (IP), TRILL, Internetwork Packet Exchange (IPX) de Novell, NBF y AppleTalk. PPP permite que múltiples protocolos de capa de red operen en el mismo enlace de comunicación. Para cada protocolo de capa de red utilizado, se proporciona un Protocolo de control de red (NCP) separado para encapsular y negociar opciones para los múltiples protocolos de capa de red. Negocia la información de la capa de red, por ejemplo, la dirección de red o las opciones de compresión, una vez que se ha establecido la conexión.

Estructura de un marco PPP

Name	Number of bytes	Description
Protocol	1 or 2	setting of protocol in data field
Information	variable (0 or more)	datagram
Padding	variable (0 or more)	optional padding

los**Protocolo**El campo indica el tipo de paquete de carga útil (p. ej., LCP, NCP, IP, IPX, AppleTalk, etc.).

los**Información**el campo contiene la carga útil de PPP; tiene una longitud variable con un máximo negociado denominado Unidad Máxima de Transmisión. Por defecto, el máximo es 1500 octetos. Puede estar acolchado en la transmisión; si la información para un particular

el protocolo se puede rellenar, ese protocolo debe permitir que la información se distinga del relleno.

Las tramas PPP se encapsulan en un protocolo de capa inferior que proporciona tramas y puede proporcionar otras funciones, como una suma de verificación para detectar errores de transmisión. PPP en enlaces seriales generalmente se encapsula en un marco similar a HDLC.

Name	Number of bytes	Description
Flag	1	indicates frame's begin or end
Address	1	broadcast address
Control	1	control byte
Protocol	1 or 2	I in information field
Information	variable (0 or more)	datagram
Padding	variable (0 or more)	optional padding
FCS	2 (or 4)	error check

los**Bandera**El campo está presente cuando se utiliza PPP con tramas de tipo HDLC.

los**Habla a**y**Control**Los campos siempre tienen el valor FF hexadecimal (para "todas las estaciones") y 03 hexadecimal (para "información no numerada"), y se pueden omitir siempre que se negocie la compresión de campo de dirección y control (ACFC) PPP LCP.

los**Secuencia de comprobación de fotogramas**(FCS) se utiliza para determinar si una trama individual tiene un error. Contiene una suma de comprobación calculada sobre la trama para proporcionar protección básica contra errores en la transmisión. Este es un código CRC similar al que se usa para otros esquemas de protección contra errores de protocolo de capa dos, como el que se usa en Ethernet. De acuerdo con RFC 1662, puede tener un tamaño de 16 bits (2 bytes) o 32 bits (4 bytes) (el valor predeterminado es 16 bits: polinomio x16 + x12 + x5 + 1). El FCS se calcula sobre los campos Dirección, Control, Protocolo, Información y Relleno después de encapsular el mensaje.

4.8 ETERNET

Ethernet fue diseñado en la década de 1970 en el Centro de Investigación de Palo Alto. El primer prototipo utilizó un cable coaxial como medio compartido y 3 Mbps de ancho de banda. Ethernet se mejoró a fines de la década de 1970 y en la década de 1980, Digital Equipment, Intel y Xerox publicaron la primera especificación oficial de Ethernet. Esta especificación define varios parámetros importantes para las redes Ethernet. La primera decisión fue estandarizar la Ethernet comercial a 10 Mbps. La segunda decisión fue la duración de la franja horaria. En Ethernet, un intervalo de tiempo largo permite que las redes abarquen una gran distancia, pero obliga al host a utilizar un tamaño de trama mínimo mayor. El compromiso fue un intervalo de tiempo de 51,2 microsegundos, que corresponde a un tamaño de trama mínimo de 64 bytes.

Ethernet rápida:

Al mismo tiempo que los conmutadores se estaban volviendo populares, la velocidad de Ethernet de 10 Mbps estaba bajo presión. Al principio, 10 Mbps parecían el paraíso, al igual que los módems de cable parecían el paraíso para los usuarios de módems telefónicos. Pero la novedad se desvaneció rápidamente. Parecía que los datos se expandían para llenar el ancho de banda disponible para su transmisión. Muchas instalaciones necesitaban más ancho de banda y, por lo tanto, tenían numerosas LAN de 10 Mbps conectadas por un laberinto de repetidores, concentradores y conmutadores, aunque a los administradores de la red a veces les parecía que los mantenían unidos con goma de mascar y alambre de gallinero. Pero incluso con conmutadores Ethernet, el ancho de banda máximo de una sola computadora estaba limitado por el cable que lo conectaba al puerto del conmutador. Fue en este entorno que IEEE volvió a convocar al comité 802.3 en 1992 con instrucciones para crear una LAN más rápida. Una propuesta fue mantener el 802.3 exactamente como estaba, pero hacerlo más rápido. Otra propuesta fue rehacerlo por completo y darle muchas características nuevas, como tráfico en tiempo real y voz digitalizada, pero manteniendo el nombre anterior. Después de algunas discusiones, el comité decidió mantener el 802.3 como estaba y simplemente hacerlo funcionar más rápido. Esta estrategia haría el trabajo antes de que cambiara la tecnología y evitaría problemas imprevistos con un diseño completamente nuevo. El nuevo diseño también sería compatible con versiones anteriores de las LAN Ethernet existentes. Las personas detrás de la propuesta perdedora hicieron lo que cualquier persona de la industria informática que se precie habría hecho en estas circunstancias: pisotearon y formaron su propio comité y estandarizaron su LAN de todos modos (eventualmente como 802.12). Fracasó miserablemente. El trabajo se realizó rápidamente y el resultado, 802.3u, fue aprobado por IEEE en junio de 1995. Técnicamente, 802.3u no es un estándar nuevo, sino una adición al estándar 802.3 existente. Esta estrategia se usa mucho. Dado que prácticamente todo el mundo lo llama Fast Ethernet, en lugar de 802.3u, también lo haremos. La idea básica detrás de Fast Ethernet era simple: mantener todos los formatos de marco, interfaces y reglas de procedimiento antiguos, pero reducir el tiempo de bit de 100 nseg a 10 nseg. Técnicamente, habría sido posible copiar Ethernet clásico de 10 Mbps y aun así detectar colisiones a tiempo simplemente reduciendo la longitud máxima del cable por un factor de 10. Sin embargo, las ventajas del cableado de par trenzado eran tan abrumadoras que se basa en Ethernet rápido. totalmente en este diseño. Por lo tanto, todos los sistemas Fast Ethernet utilizan concentradores y conmutadores; No se permiten cables multipunto con derivaciones vampiro o conectores BNC. Sin embargo, aún quedaban por hacer algunas elecciones, siendo la más importante qué tipos de cables soportar. Un contendiente fue el par trenzado de Categoría 3. El argumento para ello era que prácticamente todas las oficinas del mundo occidental tenían al menos cuatro pares trenzados de categoría 3 que iban desde allí hasta un armario de cableado telefónico en un radio de 100 metros. A veces existían dos de esos cables. Por lo tanto, el uso de par trenzado de categoría 3 permitiría conectar computadoras de escritorio mediante Ethernet rápido sin tener que volver a cablear el edificio, una enorme ventaja para muchas organizaciones. La principal desventaja de un par trenzado de Categoría 3 es su incapacidad para transportar 100 Mbps a lo largo de 100 metros, la distancia máxima entre la computadora y el concentrador especificada para los concentradores de 10 Mbps. Por el contrario, el cableado de par trenzado de categoría 5 puede manejar 100 m fácilmente y la fibra puede llegar mucho más lejos.**Figura 4.12**, sino para animar la solución de Categoría 3 para darle la capacidad de carga adicional necesaria. La categoría 3

El esquema UTP, denominado 100 Base-T4, utilizaba una velocidad de señalización de 25 MHz, solo un 25 % más rápida que los 20 MHz estándar de Ethernet. Sin embargo, para lograr la tasa de bits necesaria, 100 Base-T4 requiere cuatro pares trenzados. De los cuatro pares, uno siempre es hacia el concentrador, uno siempre es desde el concentrador y los otros dos son conmutables a la dirección de transmisión actual. Para obtener 100 Mbps de los tres pares trenzados en la dirección de transmisión, se usa un esquema bastante complicado en cada par trenzado. Implica el envío de dígitos ternarios con tres niveles de voltaje diferentes. No es probable que este esquema gane ningún premio a la elegancia, y nos saltaremos los detalles. Sin embargo, dado que el cableado telefónico estándar durante décadas ha tenido cuatro pares trenzados por cable, la mayoría de las oficinas pueden utilizar la planta de cableado existente. 100Base-T4 se quedó en el camino ya que muchos edificios de oficinas fueron reconfigurados con UTP de Categoría 5 para 100 Base-TX Ethernet, que llegó a dominar el mercado. Este diseño es más simple porque los cables pueden manejar velocidades de reloj de 125 MHz. Solo se utilizan dos pares trenzados por estación, uno hacia el hub y otro desde el mismo.

Name	Cable	Max. segment	Advantages
100Base-T4	Twisted pair	100 m	Uses category 3 UTP
100Base-TX	Twisted pair	100 m	Full duplex at 100 Mbps (Cat 5 UTP)
100Base-FX	Fiber optics	2000 m	Full duplex at 100 Mbps; long runs

Fig 4.12: El cableado Fast Ethernet original

No se utiliza codificación binaria directa (es decir, NRZ) ni codificación Manchester. En su lugar, se utiliza la codificación 4B/5B que describimos en la Sección 2.5. 4 bits de datos se codifican como 5 bits de señal y se envían a 125 MHz para proporcionar 100 Mbps. Este esquema es simple pero tiene suficientes transiciones para la sincronización y utiliza relativamente bien el ancho de banda del cable. El sistema 100 Base-TX es dúplex completo; las estaciones pueden transmitir a 100 Mbps en un par trenzado y recibir a 100 Mbps en otro par trenzado al mismo tiempo. La última opción, 100 Base-FX, utiliza dos hilos de fibra multimodo, uno para cada dirección, por lo que también puede funcionar en dúplex completo con 100 Mbps en cada dirección. En esta configuración, la distancia entre una estación y el interruptor puede ser de hasta 2 km. Fast Ethernet permite la interconexión mediante concentradores o conmutadores. Para asegurarse de que el algoritmo CSMA/CD sigue funcionando, la relación entre el tamaño mínimo de la trama y la longitud máxima del cable debe mantenerse a medida que la velocidad de la red aumenta de 10 Mbps a 100 Mbps. Por lo tanto, el tamaño mínimo de trama de 64 bytes debe aumentar o la longitud máxima del cable de 2500 m debe disminuir, proporcionalmente. La elección fácil fue que la distancia máxima entre dos estaciones cualesquiera se redujera por un factor de 10, ya que un concentrador con cables de 100 m ya se encuentra dentro de este nuevo máximo. Sin embargo, los cables 100 Base-FX de 2 km son demasiado largos para permitir un concentrador de 100 Mbps con el algoritmo de colisión normal de Ethernet. En cambio, estos cables deben estar conectados a un interruptor y operar en un modo full-duplex para que no haya colisiones. Los usuarios rápidamente comenzaron a implementar Ethernet rápido, pero no estaban dispuestos a desechar las tarjetas Ethernet de 10 Mbps en las computadoras más antiguas. Como consecuencia, prácticamente todos los conmutadores Fast Ethernet pueden manejar una combinación de estaciones de 10 Mbps y 100 Mbps. Para facilitar la actualización, el estándar mismo proporciona un mecanismo llamado negociación automática que permite que dos estaciones negocien automáticamente el

velocidad óptima (10 o 100 Mbps) y dúplex (media o completa). Funciona bien la mayor parte del tiempo, pero se sabe que genera problemas de discrepancia de dúplex cuando un extremo del enlace negocia automáticamente pero el otro extremo no lo hace y está configurado en modo dúplex completo. La mayoría de los productos Ethernet utilizan esta función para configurarse.

Gigabit Ethernet -

Apenas se había secado la tinta en el estándar Fast Ethernet cuando el comité 802 comenzó a trabajar en un Ethernet aún más rápido, rápidamente denominado Gigabit Ethernet. IEEE ratificó la forma más popular como 802.3ab en 1999. Los objetivos del comité para Gigabit Ethernet eran esencialmente los mismos que los objetivos del comité para Fast Ethernet: multiplicar por diez el rendimiento manteniendo la compatibilidad con todos los estándares Ethernet existentes. En particular, Gigabit Ethernet tenía que ofrecer servicios de datagramas no reconocidos con unidifusión y transmisión, usar el mismo esquema de direccionamiento de 48 bits que ya estaba en uso y mantener el mismo formato de trama, incluidos los tamaños de trama mínimo y máximo. La norma final cumplió con todos estos objetivos. Al igual que Fast Ethernet, todas las configuraciones de Gigabit Ethernet utilizan enlaces punto a punto. En la configuración más simple, ilustrada en**Figura 4.13(a)**, dos computadoras están conectadas directamente entre sí. Sin embargo, el caso más común utiliza un conmutador o un concentrador conectado a varias computadoras y posiblemente conmutadores o concentradores adicionales, como se muestra en**Figura 4.13(b)**. En ambas configuraciones, cada cable Ethernet individual tiene exactamente dos dispositivos, ni más ni menos. Al igual que Fast Ethernet, Gigabit Ethernet admite dos modos de funcionamiento diferentes: modo dúplex completo y modo dúplex medio. El modo “normal” es el modo full-duplex, que permite el tráfico en ambas direcciones al mismo tiempo. Este modo se usa cuando hay un conmutador central conectado a computadoras (u otros conmutadores) en la periferia. En esta configuración, todas las líneas se almacenan en búfer, por lo que cada computadora y conmutador pueden enviar tramas cuando lo deseen. El remitente no tiene que detectar el canal para ver si alguien más lo está utilizando porque la contención es imposible. En la línea entre una computadora y un conmutador, la computadora es el único remitente posible al conmutador,

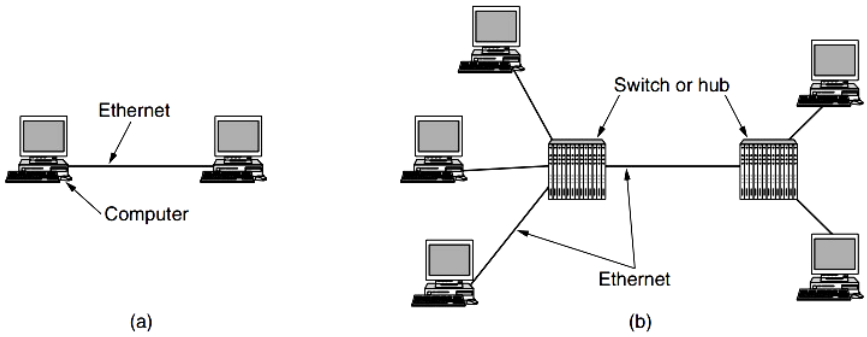


Fig 4.13: (a) Ethernet de dos estaciones. (b) Una Ethernet multiestación

Dado que no es posible la contención, no se utiliza el protocolo CSMA/CD, por lo que la longitud máxima del cable está determinada por problemas de intensidad de la señal en lugar de por el tiempo que tarda una ráfaga de ruido en propagarse de vuelta al remitente en el peor de los casos. Los cambios son gratis

para mezclar y combinar velocidades. La negociación automática se admite al igual que en Fast Ethernet, solo que ahora la elección es entre 10, 100 y 1000 Mbps. El otro modo de operación, semidúplex, se usa cuando las computadoras están conectadas a un concentrador en lugar de a un conmutador. Un concentrador no almacena en búfer los marcos entrantes. En su lugar, conecta eléctricamente todas las líneas internamente, simulando el cable multipunto utilizado en Ethernet clásico. En este modo, las colisiones son posibles, por lo que se requiere el protocolo CSMA/CD estándar. Debido a que una trama de 64 bytes (la más corta permitida) ahora se puede transmitir 100 veces más rápido que en la Ethernet clásica, la longitud máxima del cable debe ser 100 veces menor, o 25 metros, para mantener la propiedad esencial de que el remitente todavía está transmitiendo cuando el La ráfaga de ruido vuelve a él, incluso en el peor de los casos. Con un cable de 2500 metros de largo, el remitente de una trama de 64 bytes a 1 Gbps terminaría mucho antes de que la trama llegara ni una décima parte del camino hasta el otro extremo, y mucho menos hasta el final y de regreso. Esta restricción de longitud fue tan dolorosa que se agregaron dos características al estándar para aumentar la longitud máxima del cable a 200 metros, lo que probablemente sea suficiente para la mayoría de las oficinas. La primera característica, llamada extensión de portadora, esencialmente le dice al hardware que agregue su propio relleno después del marco normal para extender el marco a 512 bytes. Dado que este relleno lo agrega el hardware de envío y lo elimina el hardware de recepción, el software no lo reconoce, lo que significa que no se necesitan cambios en el software existente. La desventaja es que usar 512 bytes de ancho de banda para transmitir 46 bytes de datos de usuario (la carga útil de una trama de 64 bytes) tiene una eficiencia de línea de solo el 9 %. La segunda característica, llamado ráfaga de tramas, permite que un remitente transmita una secuencia concatenada de múltiples tramas en una sola transmisión. Si la ráfaga total es inferior a 512 bytes, el hardware la rellena de nuevo. Si hay suficientes tramas esperando para la transmisión, este esquema es muy eficiente y se prefiere a la extensión de la portadora. Para ser justos, es difícil imaginar que una organización compre computadoras modernas con tarjetas Gigabit Ethernet y luego las conecte con un concentrador anticuado para simular la Ethernet clásica con todas sus colisiones. Las interfaces y conmutadores Gigabit Ethernet solían ser costosos, pero sus precios cayeron rápidamente a medida que aumentaban los volúmenes de ventas. Aún así, la compatibilidad con versiones anteriores es sagrada en la industria informática, por lo que se solicitó al comité que la implementara. Hoy en día, la mayoría de las computadoras se envían con una interfaz Ethernet que es capaz de y funcionamiento a 1000 Mbps y compatible con todos ellos. Gigabit Ethernet admite cableado de fibra y cobre, como se indica en**Figura 4.14**. La señalización a 1 Gbps o cerca requiere la codificación y el envío de un bit cada nanosegundo. Este truco se logró inicialmente con cables de cobre blindados cortos (la versión 1000 Base-CX) y fibras ópticas. Para las fibras ópticas se permiten dos longitudes de onda y resultan dos versiones diferentes: 0,85 micras (corta, para 1000Base-SX) y 1,3 micras (larga, para 1000Base-LX).

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

Figura 4.14: Cableado Gigabit Ethernet

La señalización en la longitud de onda corta se puede lograr con LED más baratos. Se utiliza con fibra multimodo y es útil para conexiones dentro de un edificio, ya que puede recorrer hasta 500 m para fibra de 50 micras. La señalización en la longitud de onda larga requiere láseres más caros. Por otro lado, cuando se combina con fibra monomodo (10 micras), la longitud del cable puede ser de hasta 5 km. Este límite permite conexiones de larga distancia entre edificios, como para una red troncal de campus, como un enlace punto a punto dedicado. Las variaciones posteriores del estándar permitieron enlaces aún más largos sobre fibra monomodo. Para enviar bits a través de estas versiones de Gigabit Ethernet, se tomó prestada la codificación 8B/10B de otra tecnología de red llamada Fibre Channel. Ese esquema codifica 8 bits de datos en palabras clave de 10 bits que se envían por cable o fibra, de ahí el nombre 8B/10B. Las palabras de código se eligieron para que pudieran equilibrarse (es decir, tener el mismo número de 0 y 1) con suficientes transiciones para la recuperación del reloj. Enviar los bits codificados con NRZ requiere un ancho de banda de señalización de un 25 % más que el requerido para los bits no codificados, una gran mejora con respecto a la expansión del 100 % de la codificación Manchester. Sin embargo, todas estas opciones requerían nuevos cables de cobre o fibra para admitir una señalización más rápida. Ninguno de ellos hizo uso de la gran cantidad de UTP de categoría 5 que se había instalado junto con Fast Ethernet. En un año, apareció 1000 Base-T para llenar este vacío, y desde entonces ha sido la forma más popular de Gigabit Ethernet. A la gente aparentemente no le gusta volver a cablear sus edificios. Se necesita una señalización más complicada para que Ethernet funcione a 1000 Mbps a través de cables de categoría 5. Para comenzar, se utilizan los cuatro pares trenzados del cable, y cada par se utiliza en ambas direcciones al mismo tiempo mediante el procesamiento de señales digitales para separar las señales. Sobre cada cable, se utilizan cinco niveles de voltaje que transportan 2 bits para señalizar a 125 millones de símbolos/seg. El mapeo para producir los símbolos a partir de los bits no es sencillo. Implica codificación, para transiciones, seguida de un código de corrección de errores en el que se incrustan cuatro valores en cinco niveles de señal. Una velocidad de 1 Gbps es bastante rápida. Por ejemplo, si un receptor está ocupado con alguna otra tarea incluso durante 1 ms y no vacía el búfer de entrada en alguna línea, es posible que se hayan acumulado hasta 1953 tramas en ese espacio. Además, cuando una computadora en un gigabit Ethernet envía datos a una computadora en un Ethernet clásico, es muy probable que se desborde el búfer. Como consecuencia de estas dos observaciones, Gigabit Ethernet admite el control de flujo. El mecanismo consiste en que un extremo envía un marco de control especial al otro extremo diciéndole que haga una pausa durante un período de tiempo. Estos marcos de control de PAUSA son marcos Ethernet normales que contienen un tipo de 0x8808. Las pausas se dan en unidades del tiempo de marco mínimo. Para Gigabit Ethernet, la unidad de tiempo es 512 nseg, lo que permite pausas de hasta 33,6 mseg. Hay una extensión más que se introdujo junto con Gigabit Ethernet. Las tramas gigantes permiten que las tramas tengan más de 1500 bytes, generalmente hasta 9 KB. Esta extensión es propietaria. El estándar no lo reconoce porque, si se usa, Ethernet ya no es compatible con versiones anteriores, pero la mayoría de los proveedores lo admiten de todos modos. La razón es que 1500 bytes es una unidad corta a velocidades de gigabit.

llegado, o dividir y recombinar mensajes que eran demasiado largos para caber en una trama de Ethernet.

REVISA TU PROGRESO

2. Complete los espacios en blanco:

(a) El proceso de la capa física y algunos de los procesos de la capa de enlace de datos se ejecutan en un hardware dedicado llamado _____.

(b) Los protocolos en los que el remitente envía una trama y luego espera un acuse de recibo antes de continuar se denominan _____.

(c) En el protocolo _____, el receptor simplemente descarta todas las tramas subsiguientes, sin enviar acuses de recibo por las tramas descartadas.

(d) En el protocolo de repetición selectiva, tanto _____ como _____ mantienen una ventana de números de secuencia pendientes y aceptables, respectivamente.

(e) _____ tramas pueden transmitirse a través de enlaces síncronos o asíncronos.

4.9 RESUMAMOS

- En el modelo OSI de siete capas de redes informáticas, el ***Enlace de datos*** la capa es la capa 2.

- los ***encuadre de conteo de bytes*** El método utiliza un campo en el encabezado para especificar el número de bytes en el marco.

- ***control de errores*** se asegura de que todas las tramas finalmente se entreguen a la capa de red en el destino y en el orden correcto.

- En ***control de flujo basado en retroalimentación***, el receptor devuelve información al remitente dándole permiso para enviar más datos.

- El número de posiciones de bits en las que difieren dos palabras de código se denomina ***distancia de hamming***.

- ***Códigos convolucionales*** se especifican en términos de su tasa y longitud de restricción.

- ***Reed-Salomón*** Los códigos se utilizan ampliamente en la práctica debido a sus fuertes propiedades de corrección de errores, particularmente para errores de ráfaga.

- ***Repetición selectiva*** protocolo, es permitir que el receptor acepte y almacene en búfer las tramas que siguen a una dañada o perdida.

- ***HDLC*** proporciona orientación a la conexión y servicio sin conexión.

- ***Protocolo punto a punto*** (PPP) es un protocolo de enlace de datos comúnmente utilizado para establecer una conexión directa entre dos nodos de red.

4.10 RESPUESTAS PARA COMPROBAR TU PROGRESO

1.
(a) enlace de datos
(b) Basado en tasas
(c) Corrección de errores
(d) Detección de errores
(e) Distancia de Hamming
(f) código convolucional
(g) Códigos Reed-Solomon
h) Códigos LDPC
(i) suma de control
(j) Códigos de polinomios.

2.
(a) Tarjeta de interfaz de red
(b) parar y esperar
(c) Regresar-N
(d) emisor, receptor
(e) HDLC.

4.11 LECTURAS ADICIONALES

Red de computadoras

- Andrew S. Tanenbaum, David J. Wetherall
  PRENTICE HALL

Redes informáticas: principios, protocolos y práctica

-Olivier Buenaventura

4.12 PREGUNTAS MODELO

1. ¿Qué es la capa de enlace de datos del modelo OSI? Describa sus funciones.

2. ¿Qué es Framing en la capa de enlace de datos? Describir los diferentes métodos de encuadre utilizados.

3. Describir el concepto de control de errores en la capa de enlace de datos.

4. ¿Qué es el control de flujo en la capa de enlace de datos? ¿Cuáles son las técnicas de control de flujo disponibles?

5. ¿Qué entiende por Detección de errores y Corrección de errores en la capa de enlace de datos?

6. Ilustre con ejemplos tres códigos de corrección de errores presentes en la capa de enlace de datos.

7. ¿Qué son los códigos de detección de errores en la capa de enlace de datos? Describir.

8. Ilustre el protocolo Stop-and-Wait de la capa de enlace de datos.

9. Describa el protocolo Go-Back-N de la capa de enlace de datos.

10. Describa el funcionamiento del protocolo de repetición selectiva de la capa de enlace de datos.

11. Escriba una breve nota sobre el protocolo HDLC de la capa de enlace de datos.

12. Explicar el protocolo Punto a Punto de la capa de Enlace de Datos.

13. Escriba una nota sobre Ethernet y sus otros tipos.

UNIDAD - 5 CAPA DE RED

ESTRUCTURA DE LA UNIDAD

5.1 OBJETIVOS DE APRENDIZAJE

Después de pasar por esta unidad, podrá:

- aprender sobre el concepto básico de la capa de red
- aprender sobre interredes y direccionamiento de red
- describir el concepto de enrutamiento y diferentes protocolos de enrutamiento
- aprender acerca de los diferentes protocolos de capa de red.

5.2 INTRODUCCIÓN A LA CAPA DE RED

En esta unidad discutiremos sobre la capa de red del modelo OSI. La capa de red proporciona el mecanismo de transferencia de secuencias de datos de longitud variable desde un host de origen en una red a un host de destino en una red diferente. En este proceso, también se mantiene la calidad de servicio solicitada por la capa de transporte. La capa de red agrega un encabezado que incluye las direcciones lógicas del remitente y el receptor al paquete que proviene de la capa superior.

La capa de red realiza funciones de enrutamiento de red. Los enrutadores se utilizan en esta capa para enviar datos a través de las diferentes redes.

La capa de red debe conocer la topología de la subred de comunicación y elegir las rutas adecuadas a través de ella para realizar su trabajo correctamente. Debe elegir cuidadosamente las rutas para evitar sobrecargar algunas de las líneas de comunicación y enrutadores. Cuando el origen y el destino están en redes diferentes, se producen diferentes problemas. Corresponde a la capa de red hacer frente a estos problemas.

5.3 INTERREDES

Una colección de redes interconectadas, que permite que los datos se muevan libremente entre esta gran cantidad de redes y poblaciones diferentes, se denomina interred o internet. Entonces, las personas conectadas a una red pueden comunicarse con personas conectadas a una red diferente con la ayuda de internetwork. Las puertas de enlace se utilizan para realizar la conexión entre diferentes redes y proporcionar la traducción necesaria para el hardware y el software. En otras palabras, Internet es una colección de LAN conectadas por una WAN. Diferentes organizaciones construyen diferentes partes de la interred y cada organización mantiene su propia parte.

La interred es una red de conmutación de paquetes en la capa de red. La interred utiliza direcciones universales definidas en la capa de red para enrutar paquetes desde el origen hasta el destino. La entrega de un paquete se puede lograr utilizando un servicio de red orientado a la conexión o sin conexión. En el servicio sin conexión, el protocolo de la capa de red trata cada paquete de forma independiente. Los paquetes en un mensaje pueden o no viajar por la misma ruta hasta su destino. Este tipo de servicio se utiliza en el enfoque de datagramas para la conmutación de paquetes. Internet ha elegido este tipo de servicio en la capa de red porque Internet está compuesto por diferentes tipos de redes, por lo que es imposible crear una conexión desde el origen hasta el destino sin conocer de antemano la naturaleza de las redes.

5.4 DIRECCIONAMIENTO

En la capa de red, el paquete transmitido por la computadora de envío puede viajar a través de diferentes LAN o WAN en el camino hacia la computadora de destino. Ahora, en esta parte de la comunicación, se utiliza un esquema de direccionamiento global que se denomina direccionamiento lógico. El término dirección IP se utiliza para referirse a una dirección lógica en la capa de red del conjunto de protocolos TCP/IP.

Las direcciones de Internet utilizadas en la actualidad tienen una longitud de 32 bits y se denominan direcciones IPv4 (IP versión 4) o simplemente direcciones IP. Entonces, en el caso de la dirección IPv4, el máximo de 2_{32}Las direcciones son posibles.
Ahora se diseña una nueva versión de dirección de internet debido al requerimiento de más de 2_{32} direcciones. Esta nueva versión se denomina IPv6 (IP versión 6). En esta versión, Internet utiliza direcciones de 128 bits que se denominan direcciones IPv6. Entonces, en el caso de la dirección IPv6, el máximo de 2_{128}Las direcciones son posibles.

Direcciones IPv4:

Una dirección IPv4 es una dirección de 32 bits que define de forma única y global la conexión de un dispositivo a Internet. Por lo tanto, dos dispositivos en Internet nunca pueden tener la misma dirección IPv4 al mismo tiempo y este esquema de direccionamiento debe ser aceptado por cualquier host que desee conectarse a Internet.

La longitud de una dirección IPv4 es de 32 bits. Cada bit de una dirección IPv4 puede tener dos valores diferentes, que son 0 o 1. Entonces, el espacio de direcciones de IPv4 es 2_{32}o 4.294.967.296.

Para mostrar una dirección IPv4, se pueden usar dos tipos de notaciones, que son la notación binaria y la notación decimal con puntos.

En notación binaria, la dirección IPv4 se muestra como 32 bits. Por ejemplo:

00111000 10110011 00111011 00001100

En notación decimal con puntos, las direcciones IPv4 se escriben en forma decimal con un punto decimal que separa los bytes. Por ejemplo, la notación decimal con puntos de la dirección anterior se da a continuación:

56.179.59.12

En el direccionamiento con clase, hay cinco clases de direcciones IPv4 que son A, B, C, D y E. Cada clase ocupa una parte del espacio de direcciones. Si la dirección se da en notación binaria, la clase de la dirección se puede encontrar con los primeros bits más a la izquierda de la siguiente manera:

- Si el bit más a la izquierda de la dirección es 0, entonces está en la clase A. Por ejemplo: 00000100 10001001 00010010 11101101 es una dirección de clase A.
- Si los dos bits más a la izquierda de la dirección son 10, entonces está en la clase B. Por ejemplo: 10100001 00010010 11100010 10111001 es una dirección de clase B.
- Si los tres bits más a la izquierda de la dirección son 110, entonces está en Clase C. Por ejemplo: 11000001 10001001 00001010 11111010 es una dirección de clase C.

- Si los cuatro bits más a la izquierda de la dirección son 1110, entonces está en la clase D. Por ejemplo: 11101010 00010101 10111010 11111111 es una dirección de clase D.
- Si los cuatro bits más a la izquierda de la dirección son 1111, entonces está en la clase E. Por ejemplo: 11111001 00010110 11101101 00011101 es una dirección de clase E.

Ahora, si la dirección se da en notación con puntos decimales, el valor decimal del primer byte define la clase de la siguiente manera:

- Si el valor decimal del primer byte de la dirección está en el rango de 0 a 127, entonces está en la clase A. Por ejemplo: 4.137.18.237 es una dirección de clase A.
- Si el valor decimal del primer byte de la dirección está en el rango de 128 a 191, entonces es de clase B. Por ejemplo: 161.18.226.185 es una dirección de clase B.
- Si el valor decimal del primer byte de la dirección está en el rango de 192 a 223, entonces es de clase C. Por ejemplo: 193.137.20.250 es una dirección de clase C.
- Si el valor decimal del primer byte de la dirección está en el rango de 224 a 239, entonces está en la clase D. Por ejemplo: 234.21.186.255 es una dirección de clase D.
- Si el valor decimal del primer byte de la dirección está en el rango de 240 a 255, entonces está en la clase E. Por ejemplo: 249.22.237.29 es una dirección de clase E.

Las direcciones de clase A se diseñaron para organizaciones grandes con una gran cantidad de hosts o enrutadores conectados. Las direcciones de clase B se diseñaron para organizaciones medianas con decenas de miles de hosts o enrutadores conectados. Las direcciones de clase C se diseñaron para organizaciones pequeñas con una pequeña cantidad de hosts o enrutadores conectados. Las direcciones de clase D se diseñaron para multidifusión. Las direcciones de clase E se reservaron para uso futuro.

En el direccionamiento con clase, una dirección IP de clase A, B o C dividida en netid y hostid. En la clase A, un byte define el netid y tres bytes definen el hostid. En la clase B, dos bytes definen el netid y dos bytes definen el hostid. En la clase C, tres bytes definen el netid y un byte define el hostid.

En el direccionamiento con clase, la máscara o la máscara predeterminada se utilizan para las clases A, B y C. Una máscara es un número de 32 bits en el que los n bits más a la izquierda son 1 y los 32 n bits más a la derecha son 0. La clase D y la clase E no tienen ninguna máscara predeterminada. Las máscaras para las clases A, B y C se dan en la siguiente tabla.

Clase	Binario	Punto decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/dieciséis
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

En la tabla dada, la última columna muestra la máscara en la forma /n donde n puede ser 8, 16 o 24 en direccionamiento con clase. Esta notación también se denomina notación de barra inclinada o notación de enrutamiento entre dominios sin clase (CIDR). Esta notación se utiliza en el direccionamiento sin clases.

Debido al rápido crecimiento de Internet, es posible que se agoten las direcciones disponibles en el caso del esquema de direccionamiento con clase. En los últimos tiempos, el número de dispositivos en Internet es inferior a los 2^{32}espacio de direcciones, pero la disponibilidad de direcciones de clase A y clase B está disminuyendo y un bloque de clase C es demasiado pequeño para la mayoría de las organizaciones medianas.

Para resolver el inconveniente del esquema de direccionamiento con clase, se diseñó e implementó el direccionamiento sin clase. Este esquema de direccionamiento supera el agotamiento de direcciones y brinda a más organizaciones acceso a Internet. En este esquema no hay clases.

En el direccionamiento sin clase, cuando un dispositivo necesita conectarse a Internet, se le otorga un bloque de direcciones. El tamaño del bloque varía según la naturaleza y el tamaño del dispositivo.
Las autoridades de Internet imponen tres restricciones a los bloques de direcciones sin clase para simplificar el manejo de las direcciones, que son:

1. Las direcciones de un bloque deben ser contiguas.
2. El número de direcciones en un bloque debe ser una potencia de 2 como 2^1, 2^2, 2^4etc
3. La primera dirección debe ser divisible por el número de direcciones.

En el direccionamiento sin clases, la máscara de un bloque puede tomar cualquier valor de 0 a 32.

En el direccionamiento IPv4, un bloque de direcciones se puede definir como xyzt/n, donde xyzt define una de las direcciones y /n define la máscara.

Ahora, la primera dirección en el bloque se puede encontrar configurando los 32-n bits más a la derecha en la notación binaria de la dirección a 0 y la última dirección en el bloque se puede encontrar configurando los 32-n bits más a la derecha en la notación binaria de la dirección a 1s. El número de direcciones en el bloque es la diferencia entre la última y la primera dirección. Se puede encontrar usando la formula 2^{32-n}.

Traducción de direcciones de red (NAT)

En un futuro próximo, el espacio de direcciones de IPv4 no será suficiente para dar cabida a todos los dispositivos en Internet debido al rápido crecimiento de Internet. Ahora, una solución a este problema es la traducción de direcciones de red (NAT). Se describe en RFC 3022.

El concepto básico de NAT es asignar a cada empresa un gran conjunto de direcciones internamente y una dirección o un pequeño conjunto de direcciones externamente. Cada computadora dentro de la empresa obtiene una dirección IP única del gran conjunto de direcciones. Cuando un paquete de datos sale de la empresa y va al ISP, se lleva a cabo una traducción de direcciones. En este esquema, se han declarado como privados tres rangos de direcciones IP. Estas direcciones pueden ser utilizadas internamente por diferentes empresas, pero los paquetes que contienen estas direcciones no pueden aparecer en Internet. Estos tres rangos reservados de direcciones se dan de la siguiente manera:

10.0.0.0	para 10.255.255.255	(16.777.216 anfitriones)
172.16.0.0	para 172.31.255.255	(1.048.576 anfitriones)
192.168.0.0	para 192.168.255.255	(65.536 anfitriones)

El primer rango prevé 16.777.216 direcciones. En general, la mayoría de las empresas eligen este rango de direcciones. Entonces, al usar el primer rango de direcciones, cada computadora dentro de la empresa tiene una dirección única de la forma 10.xyz. Cuando un paquete sale de la red de la empresa, pasa a través de un cuadro NAT que convierte la dirección IP de origen interna en la verdadera IP de la empresa. habla a.

Direcciones IPv6:

Protocolo de Internet versión 6(**IPv6**) es la última versión de las direcciones de Internet. IPv6 fue desarrollado por el Grupo de Trabajo de Ingeniería de Internet (IETF).

IPv6 tiene un espacio de direcciones de 2_{128}direcciones porque utiliza una dirección de 128 bits. Entonces, el espacio de direcciones de IPv6 es mucho más grande que el espacio de direcciones de IPv4.

Notación hexadecimal de dos puntos:IPv6 especifica la notación de dos puntos hexadecimales para sus direcciones, donde 128 bits se dividen en ocho secciones y cada una tiene 2 bytes de longitud. Dos bytes en notación hexadecimal requieren cuatro dígitos hexadecimales. Entonces, una dirección IPv6 consta de 32 dígitos hexadecimales con cada cuatro dígitos separados por dos puntos. Por ejemplo: EFED: 07E4: 0070: 0040: 00E0: D0FF: 0000: EEEE es una dirección IPv6 en notación hexadecimal de dos puntos.

En el caso de IPv6, las direcciones IP se dividen en varias categorías. Unos pocos bits más a la izquierda en cada dirección IP que se denominan prefijo de tipo definen la categoría de cada dirección. El prefijo de tipo está diseñado de manera que ningún código sea idéntico a la primera parte de ningún otro código. Entonces, cuando se proporciona una dirección, el prefijo de tipo se puede determinar fácilmente. La siguiente tabla muestra el tipo de prefijo y su tipo de dirección

Prefijo de tipo	Escribe
0000 0000	Reservado
0000 0001	Sin asignar
0000 001	Direcciones de red ISO
0000 010	Red IPX (Novell) direcciones
0000 011	Sin asignar
0000 1	Sin asignar
0001	Reservado
001	Reservado
010	Unidifusión basada en proveedor direcciones
011	Sin asignar
100	Direcciones de unidifusión basadas en la geografía
101	Sin asignar
110	Sin asignar
1110	Sin asignar
1111 0	Sin asignar
1111 10	Sin asignar
1111 110	Sin asignar

1111 1110 0	Sin asignar
1111 1110 10	Vincular direcciones locales
1111 1110 11	Direcciones locales del sitio
1111 1111	Direcciones de multidifusión

Las diferentes categorías de direcciones IPv6 se describen a continuación:

Direcciones de unidifusión:

Una dirección de unidifusión se utiliza para definir una sola computadora. El paquete enviado a una dirección de unidifusión debe entregarse a una computadora específica. IPv6 define dos tipos de direcciones de unidifusión que se basan en la ubicación geográfica y en el proveedor.

Direcciones de multidifusión:

Las direcciones de multidifusión se utilizan para definir un grupo de hosts. Un paquete enviado a una dirección de multidifusión debe entregarse a cada miembro del grupo.

Direcciones Anycast:

IPv6 define direcciones anycast donde cada dirección anycast define un grupo de nodos. Ahora, un paquete enviado a una dirección anycast se entrega solo a uno de los miembros del grupo anycast, que es el más cercano con la ruta más corta. Las direcciones Anycast se pueden asignar a todos los enrutadores de un ISP que cubre un área lógica grande en Internet. No se asigna ningún bloque para las direcciones anycast.

Direcciones reservadas:

Las direcciones reservadas comienzan con ocho ceros. Hay algunas subcategorías en esta categoría que son dirección no especificada, dirección de bucle invertido, dirección compatible y dirección asignada.

En la dirección no especificada, los 128 bits son 0. Esta dirección se utiliza cuando un host no conoce su propia dirección y envía una consulta para encontrar su dirección.

En la dirección de bucle invertido, solo el bit más a la derecha es 1 y todos los demás 127 bits son 0. Esta dirección es utilizada por un host para probarse a sí mismo sin entrar en la red.

En una dirección compatible, los 32 bits más a la derecha son una dirección IPv4 y todos los demás 96 bits son 0. Estas direcciones se usan cuando un host que usa IPv6 quiere enviar un mensaje a otro host que usa IPv6, pero el mensaje debe viajar a través de una parte de la red que todavía usa IPv4.

En una dirección mapeada, todos los 80 bits más a la izquierda son 0 y los siguientes 16 bits a estos 80 bits son 1. Aquí los 32 bits más a la derecha son una dirección IPv4. Estas direcciones se utilizan cuando un host que migró a IPv6 desea enviar un mensaje a un host que todavía usa IPv4.

Direcciones locales:

Las direcciones locales se usan cuando una organización quiere usar el protocolo IPv6 sin estar conectada a Internet global. Entonces, en este caso, no se pueden enviar mensajes desde fuera de la organización a los nodos que usan estas direcciones. Hay dos tipos de direcciones locales en IPv6 que son la dirección local del enlace y la dirección local del sitio.

Se usa una dirección local de enlace en una subred aislada y una dirección local de sitio se usa en un sitio aislado con varias subredes.

5.5 ENRUTAMIENTO

La función principal de la capa de red es transferir una secuencia de datos de longitud variable desde la máquina de origen a la máquina de destino a través de la mejor ruta posible, que también se denomina enrutamiento de paquetes IP desde la máquina de origen a la máquina de destino. Ahora los algoritmos llamados algoritmos de enrutamiento que son la parte del software de la capa de red responsable de decidir las rutas y las estructuras de datos para transmitir los paquetes entrantes. Si la subred usa datagramas internamente, esta decisión debe ser nueva para cada vez que llega un paquete de datos porque la mejor ruta puede haber cambiado desde la última vez. Si la subred usa circuitos virtuales internamente, las decisiones de enrutamiento se toman solo cuando se configura un nuevo circuito virtual. Esto también se denomina enrutamiento de sesión porque una ruta sigue siendo la misma para una sesión de usuario completa.

Los diferentes objetivos de un algoritmo de enrutamiento se describen a continuación:

1. **Exactitud:** El enrutamiento debe realizarse de manera adecuada para que se pueda mantener la corrección para enviar los paquetes a su destino adecuado.
2. **Sencillez:** Se debe mantener la simplicidad en el desarrollo de un algoritmo de enrutamiento para que la sobrecarga sea lo más baja posible.
3. **Robustez:** Una vez que una red importante entra en funcionamiento, se puede esperar que funcione continuamente durante años sin fallas. Por lo tanto, los algoritmos de enrutamiento deben ser lo suficientemente robustos para manejar las fallas de hardware y software y deben poder hacer frente a los cambios en la topología y el tráfico sin requerir que se cancelen todos los trabajos en todos los hosts y que la red se reinicie cada vez que algún enrutador falla.
4. **Estabilidad:** Los algoritmos de enrutamiento deben ser estables en todas las situaciones posibles.
5. **Justicia:** Cada nodo conectado a la red debería tener una oportunidad justa de transmitir sus paquetes.
6. **Optimalidad:** Los algoritmos de enrutamiento deben ser óptimos en caso de rendimiento y minimizar los retrasos medios de los paquetes.

Los algoritmos de enrutamiento se pueden dividir en dos clases principales, que son algoritmos adaptativos y no adaptativos.

Los algoritmos no adaptativos son algoritmos de enrutamiento estático. Aquí, la elección de la ruta para transmitir paquetes IP de un nodo a otro se calcula de antemano y se descarga a los enrutadores cuando se inicia la red. No depende de las mediciones del tráfico y la topología actuales.

Por otro lado, en los algoritmos adaptativos, las decisiones de enrutamiento se cambian cada vez que hay un cambio en la topología y el tráfico de la red. Ahora los diferentes algoritmos adaptativos tienen diferencias entre ellos en el caso de los siguientes puntos:

(a) De dónde obtienen su información.

(b) Cuando cambian las rutas

(c) Cuando cambia la carga o cuando cambia la topología.

(d) Tipo de métrica utilizada para la optimización.

El Principio de Optimalidad

El principio de optimización se utiliza en los algoritmos de enrutamiento. Establece que si el enrutador B está en la ruta óptima del enrutador A al enrutador C, entonces la ruta óptima de B a C también se encuentra en la misma ruta.

Entonces, de acuerdo con el principio de optimización, el conjunto de rutas óptimas desde todas las fuentes hacia un destino particular forma un árbol con raíz en ese destino llamado árbol sumidero. Un árbol sumidero no es necesariamente único porque pueden existir otros árboles con las mismas longitudes de camino. El objetivo de todos los algoritmos de enrutamiento es descubrir y utilizar los árboles sumideros para todos los enrutadores.

Protocolos de enrutamiento

Los protocolos de enrutamiento se utilizan para actualizar continuamente las tablas de enrutamiento que se consultan para reenviar y enrutar paquetes IP. Un protocolo de enrutamiento es una combinación de reglas y procedimientos que permite a los enrutadores compartir todo lo que saben sobre Internet o su vecindario. Los protocolos de enrutamiento se dividen en categorías que son protocolos de unidifusión y multidifusión.

5.5.1 ENRUTAMIENTO UNIDIFUSIÓN

Un host o un enrutador tiene una tabla llamada tabla de enrutamiento que almacena una entrada para cada destino o un grupo de destinos host para enrutar paquetes IP. Una tabla de enrutamiento puede ser estática o dinámica.

Una tabla estática es una con entradas manuales que ingresa el administrador de la red. No se puede actualizar automáticamente cuando hay un cambio en Internet. Es responsabilidad del administrador actualizar la tabla manualmente.

Por otro lado, una tabla de enrutamiento dinámico se actualiza automáticamente cuando hay un cambio en Internet. En los últimos tiempos, se requieren tablas de enrutamiento dinámico en Internet para un mejor rendimiento.

En el enrutamiento de unidifusión, cuando un enrutador recibe un paquete para enrutar, necesita encontrar la ruta más corta hacia el destino del paquete. El enrutador usa su tabla de enrutamiento para encontrar la ruta más corta para ese destino en particular. Ahora, la entrada del siguiente salto correspondiente al destino en la tabla de enrutamiento es el comienzo de la ruta más corta. El enrutador tiene un árbol de ruta más corto para llegar de manera óptima a todos los destinos. En el enrutamiento de unidifusión, cada enrutador necesita solo un árbol de ruta más corto para reenviar un paquete.

Enrutamiento por vector de distancia

El enrutamiento por vector de distancia es un algoritmo de enrutamiento dinámico. Este algoritmo también se denomina algoritmo de enrutamiento Bellman-Ford distribuido o algoritmo Ford-Fulkerson. Fue desarrollado por Bellman en 1957 y Ford y Fulkerson en 1962.

En este algoritmo, cada enrutador mantiene una tabla de enrutamiento que proporciona la mejor distancia conocida a cada destino y qué ruta usar para llegar allí. Cada entrada en una tabla de enrutamiento de un enrutador contiene dos partes que son (a) la línea saliente preferida para usar para el destino y (b) una estimación del tiempo o la distancia hasta el destino. Así que aquí, con la ayuda de las tablas de enrutamiento, se calcula la ruta del host de origen al host de destino con una distancia mínima. Ahora las tablas de enrutamiento de los enrutadores se actualizan automáticamente al intercambiar información con los vecinos. Inicialmente, cada enrutador solo puede conocer la distancia entre él y sus vecinos inmediatos. Posteriormente, cada enrutador comparte su tabla de enrutamiento con sus vecinos inmediatos periódicamente y cuando hay un cambio. Al hacer esto,

El problema de la cuenta hasta el infinito

La cuenta hasta el infinito es un problema que puede ocurrir en el enrutamiento por vector de distancia. El problema de la cuenta hasta el infinito ocurre cuando un enrutador le dice a otro enrutador que tiene una ruta en alguna parte, pero no hay forma de que el segundo enrutador sepa que es parte de la ruta. El problema de la cuenta hasta el infinito es causado por fallas en los enlaces que dividen la red en dos o más segmentos.

Enrutamiento de estado de enlace

El enrutamiento de estado de enlace también es un algoritmo de enrutamiento dinámico. En este algoritmo, cada enrutador debe realizar los siguientes cinco pasos:

1. Cada enrutador debe descubrir a sus vecinos y conocer sus direcciones de red. Se realiza enviando un paquete HELLO especial desde un enrutador a todos sus vecinos conectados a él. Ahora, el enrutador del otro extremo debe enviar una respuesta indicando quién es. Estos nombres deben ser globalmente únicos.
2. Cada enrutador debe tener una estimación razonable del retraso o costo para cada uno de sus vecinos. Se puede hacer enviando un paquete ECHO especial a los vecinos de un enrutador. Ahora, al recibir un paquete ECHO por un enrutador, se requiere enviarlo inmediatamente al enrutador emisor. Al medir el tiempo de ida y vuelta y dividirlo por dos, el enrutador de envío puede obtener una estimación razonable del retraso. Para obtener mejores resultados, esta prueba se puede realizar varias veces y se utiliza el promedio. En este método, se supone que los retrasos son simétricos. Ahora aquí la carga del tráfico de red juega un papel importante. Si la carga se considera como un factor en este método, entonces el temporizador de ida y vuelta debe iniciarse cuando el paquete ECHO se carga.

en cola y si se ignora la carga, entonces el temporizador debe iniciarse cuando el paquete ECHO llega al frente de la cola.

3. Después de recopilar la información necesaria requerida para el intercambio, cada enrutador debe construir un paquete llamado paquete de estado de enlace, que contiene toda esta información. Este paquete contiene la identidad del remitente, un número de secuencia, edad, una lista de vecinos y el retraso de cada vecino. Ahora, el punto más importante es cuándo deben construirse estos paquetes. En algunos casos, los paquetes de estado de enlace se construyen a intervalos regulares. Otra posibilidad es construir estos paquetes cuando ocurre algún evento significativo, como que un vecino se caiga o vuelva a subir o cambie sus propiedades.

4. Cada enrutador debe distribuir su paquete de estado de enlace a todos los demás enrutadores. Los enrutadores que reciben el primer paquete cambiarán sus rutas. Ahora, los diferentes enrutadores pueden estar usando diferentes versiones de la topología, por lo que pueden ocurrir algunos problemas como inconsistencias, bucles, máquinas inalcanzables. Entonces, en este proceso de distribución, se usa un algoritmo de distribución para distribuir los paquetes de estado de enlace de manera confiable. La idea básica es utilizar la inundación para distribuir los paquetes. Aquí cada paquete contiene un número de secuencia que se incrementa para cada nuevo paquete enviado. Cuando llega un nuevo paquete, se compara con la lista de paquetes que ya llegaron. Si es uno nuevo, se reenvía en todas las líneas excepto en la que viene. Ahora, si es un paquete duplicado, se descarta. Si un paquete entrante tiene un número de secuencia inferior a la secuencia más alta de cualquier paquete que haya llegado antes, se rechaza porque el enrutador ahora tiene datos más recientes. Ahora bien, este algoritmo tiene algunos problemas. El primer problema es que puede ser posible que en algún momento el valor del número de secuencia alcance su valor máximo. La solución para este problema es usar un número de secuencia de 32 bits que sea lo suficientemente grande. El segundo problema es que si un enrutador falla, perderá la pista de su número de secuencia. Si comienza de nuevo en 0, el próximo paquete será rechazado como duplicado. El tercer problema es si un número de secuencia está dañado. En este caso, es posible que se rechacen algunos paquetes nuevos.

número de secuencia y disminuirlo una vez por segundo. Cuando la edad llega a cero, la información de ese enrutador se descarta. Cada enrutador también reduce el campo Edad durante el proceso de inundación inicial para asegurarse de que ningún paquete se pierda y viva por un período de tiempo indefinido. Se pueden hacer algunas modificaciones a este algoritmo. La primera modificación es cuando un paquete ingresa a un enrutador para inundación, no se pone en cola para la transmisión de inmediato. En este caso, primero se coloca en un área de espera para esperar un rato. Si llega otro paquete de la misma fuente antes de que se transmita el primer paquete, se comparan sus números de secuencia. Si son iguales, el duplicado se descarta y si son diferentes, el anterior se inunda. La segunda modificación es que todos los paquetes de estado de enlace son reconocidos. Ayudará a manejar los errores ocurridos en el enrutador a las líneas del enrutador. En este caso, cuando una línea queda inactiva, el área de espera se escanea en orden rotatorio para seleccionar un paquete para enviar el acuse de recibo.

5. Cada enrutador debe calcular la ruta más corta a todos los demás enrutadores. En este paso, el algoritmo de Dijkstra se puede ejecutar localmente para construir la ruta más corta a todos los destinos posibles. Los resultados de este algoritmo se pueden instalar en las tablas de enrutamiento. Ahora, en el caso de este algoritmo, para subredes grandes, el requisito de memoria para almacenar datos de entrada puede ser un problema porque si una subred tiene n enrutadores y cada enrutador tiene k vecinos, entonces la memoria requerida para almacenar los datos de entrada es proporcional a k* norte. El tiempo de cálculo también puede ser un problema en este algoritmo.

5.5.2 PROTOCOLOS DE ENRUTAMIENTO UNIDIAST

Dos protocolos de enrutamiento de unidifusión se analizan a continuación:

Protocolo de información de enrutamiento

El Protocolo de información de enrutamiento (RIP) es un protocolo de enrutamiento intradominio que se utiliza dentro de un sistema autónomo basado en el enrutamiento por vector de distancia. RIP implementa el enrutamiento por vector de distancia con algunas consideraciones:

1. Un sistema autónomo tiene enrutadores y redes. Aquí los enrutadores tienen tablas de enrutamiento pero las redes no tienen ninguna tabla.
2. La primera columna de las tablas de enrutamiento define una dirección de red.
3. La métrica utilizada por RIP es la distancia, que es la cantidad de redes para llegar al destino y se llama conteo de saltos.
4. Cualquier ruta en un sistema autónomo que utilice RIP no puede tener más de 15 saltos.

5. La columna del siguiente nodo en un enrutamiento define la dirección del enrutador al que se enviará el paquete para llegar a su destino.

Abrir el primer protocolo de la ruta más corta (OSPF):

El protocolo de ruta más corta abierta es un protocolo de enrutamiento de unidifusión intradominio basado en el enrutamiento de estado de enlace.

El OSPF divide un sistema autónomo en algunas áreas para manejar el enrutamiento de manera eficiente y oportuna. Un área es una colección de redes, hosts y enrutadores de un sistema autónomo. Todas las redes dentro de un área deben estar conectadas. Enrutadores dentro de un área requerida para distribuir la información de enrutamiento en el área. Hay algunos enrutadores especiales en el borde de un área llamados enrutadores de borde de área que resumen la información sobre el área y la pasan a otras áreas. Hay un área especial dentro de un sistema autónomo que se llama la red troncal y los enrutadores dentro de la red troncal se denominan enrutadores de la red troncal. Todas las demás áreas dentro de un sistema autónomo deben estar conectadas a la red troncal.

Cuando por algún problema se interrumpe la conectividad entre un backbone y un área, el administrador debe crear un enlace virtual entre los enrutadores para permitir la continuidad de las funciones del backbone como área principal.

El protocolo OSPF permite al administrador asignar una métrica a cada ruta. La métrica es en realidad un costo que se basa en un tipo de servicio, como retraso mínimo, rendimiento máximo, etc. Por lo tanto, un enrutador puede tener varias tablas de enrutamiento, cada una basada en un tipo diferente de servicio en el protocolo OSPF.

En el protocolo OSPF se han especificado cuatro tipos de enlaces que son punto a punto, transitorio, stub y virtual.

Un enlace punto a punto conecta dos enrutadores sin ningún otro host o enrutador en el medio.

Un enlace transitorio es una red con varios enrutadores conectados. Los datos pueden entrar por cualquiera de los routers y salir de la red por cualquier router.

Un enlace stub es una red que está conectada a un solo enrutador. Los paquetes de datos ingresan a la red a través de este único enrutador y salen de la red a través de este mismo enrutador.

Cuando se rompe el enlace entre dos enrutadores, el administrador puede crear un enlace virtual entre ellos utilizando una ruta más larga que posiblemente pase por varios enrutadores.

5.5.3 ENRUTAMIENTO MULTIDIFUSIÓN

En el enrutamiento de multidifusión, el enrutador recibe paquetes de multidifusión para enrutar a los destinos en más de una red. Ahora, en esta situación, el enrutamiento de un solo paquete de multidifusión a cada miembro de un grupo requiere un árbol de ruta más corto. Entonces, si hay n grupos, entonces se requieren n árboles de ruta más corta. En este caso, aumenta la complejidad del enrutamiento de multidifusión. Ahora, para resolver el problema, se utilizan enfoques de árboles basados en fuentes y árboles compartidos por grupos.

Árbol basado en fuentes: En el enfoque de árbol basado en fuente, cada enrutador necesita tener un árbol de ruta más corto para cada grupo.

Árbol compartido del grupo: En el enfoque de árbol compartido de grupo, solo un enrutador designado, denominado núcleo central, tiene m árboles de rutas más cortas en su tabla de enrutamiento. Aquí, si un enrutador recibe un paquete de multidifusión, encapsula el paquete en un paquete de unidifusión y

lo envía al enrutador central. El enrutador central elimina el paquete de multidifusión del paquete de unidifusión y enruta el paquete de multidifusión con la ayuda de su tabla de enrutamiento.

Enrutamiento de estado de enlace de multidifusión:

El enrutamiento de estado de enlace de multidifusión es una extensión del enrutamiento de estado de enlace de unidifusión. Este algoritmo de enrutamiento de multidifusión utiliza un enfoque de árbol basado en fuentes. En el enrutamiento de estado de enlace de unidifusión, cada nodo necesita anunciar el estado de sus enlaces. Ahora, en el enrutamiento de multidifusión, el estado de un enlace especifica los grupos que están activos en ese enlace. Aquí un nodo anuncia cada grupo que tiene algún miembro en el enlace. Ahora la información sobre el grupo se puede recibir ejecutando el Protocolo de administración de grupos de Internet (IGMP).

Cuando un enrutador recibe todos los paquetes de estado del enlace, construye n topologías a partir de las cuales*norte*Los árboles de ruta más corta se crean utilizando el algoritmo de Dijkstra. Aquí n es el número de grupos Entonces, cada enrutador tiene una tabla de enrutamiento que especifica n árboles de rutas más cortas.

Ahora, el problema con este algoritmo de enrutamiento es que requiere más tiempo y espacio para crear y guardar muchos árboles de caminos más cortos que el enrutamiento de unidifusión. La solución a este problema es que la creación de los árboles debe hacerse solo cuando sea necesario. Cuando un enrutador recibe un paquete con una dirección de destino de multidifusión, utiliza el algoritmo de Dijkstra para calcular el árbol de ruta más corto para ese grupo. El resultado se puede almacenar en caché en caso de que haya paquetes adicionales para ese destino.

Enrutamiento de vector de distancia de multidifusión:

El enrutamiento por vector de distancia de multidifusión es la extensión del enrutamiento por vector de distancia de unidifusión. En el caso del enrutamiento de multidifusión, los enrutadores no pueden enviar su tabla de enrutamiento a sus vecinos. Aquí la idea es crear una tabla utilizando la información de las tablas de vectores de distancia de unidifusión.

El enrutamiento por vector de distancia de multidifusión utiliza árboles basados en fuentes, pero aquí el enrutador nunca construye una tabla de enrutamiento. En este algoritmo, cuando un enrutador recibe un paquete de multidifusión, lo reenvía como si estuviera consultando una tabla de enrutamiento. Después de reenviar un paquete, la tabla se destruye.

El algoritmo de vector de distancia de multidifusión utiliza un proceso para reenviar paquetes basado en cuatro estrategias de toma de decisiones que se analizan a continuación:

1. **Inundación:** La inundación es un algoritmo estático. En esta estrategia, cada paquete IP entrante se pasa por cada línea saliente, excepto por la que llegó. La inundación genera un gran número de paquetes duplicados. Una solución para resolver este problema es tener un contador de saltos contenido en el encabezado de cada paquete que se decrementa en cada salto. Cuando el contador llega a cero, el paquete se descarta. En general, el contador de saltos debe inicializarse con la longitud de la ruta desde el origen hasta el destino, pero si el remitente no conoce la longitud, puede inicializar el contador con la longitud máxima de cualquier origen a cualquier destino.

Se puede utilizar otra técnica para realizar un seguimiento de los paquetes que se han desbordado y evitar enviarlos por segunda vez. En esta técnica, el enrutador de origen coloca un número de secuencia en cada paquete que recibe de sus hosts. Aquí, cada enrutador requiere una lista de números de secuencia por enrutador de origen que se originan en esa fuente que ya se ven. Si el número de secuencia de un paquete entrante está en la lista, se descarta.

2. **Reenvío de ruta inversa (RPF):** En RPF, un enrutador reenvía solo la copia de un paquete que ha viajado por la ruta más corta desde la fuente hasta el enrutador para evitar bucles. Ahora RPF usa la tabla de enrutamiento de unidifusión para encontrar esta copia. Las otras copias del paquete se descartan.
3. **Transmisión de ruta inversa (RPB):** En el esquema RPF, una red puede recibir dos o más copias de un paquete porque aquí el reenvío se basa en la dirección de origen. Ahora, para eliminar esta duplicación, solo se debe especificar un enrutador principal para cada red. Entonces, para esto, se puede hacer una restricción por la cual una red puede recibir un paquete de multidifusión de una fuente en particular solo a través de un enrutador principal designado. Este esquema se denomina radiodifusión de ruta inversa (RPB). Ahora, el enrutador principal designado puede ser el enrutador con la ruta más corta a la fuente.

4. **Multidifusión de ruta inversa (RPM):** En RPB, se realizan transmisiones de paquetes, lo que no es eficiente. Entonces, para aumentar la eficiencia, en el esquema RPM, el paquete de multidifusión se reenvía solo a aquellas redes que tienen miembros activos para el grupo en particular.

5.5.4 PROTOCOLOS DE ENRUTAMIENTO MULTIDIFUSIÓN

A continuación se analizan dos protocolos de enrutamiento de multidifusión:

Protocolo Multicast Open Shortest Path First (MOSPF):

El protocolo MOSPF es una extensión del protocolo OSPF. Utiliza enrutamiento de estado de enlace de multidifusión para crear árboles basados en fuentes. En este protocolo, se construye un árbol que contiene todos los hosts que pertenecen a un grupo en particular. En esta construcción, se utiliza la dirección de unidifusión del host. Para mayor eficiencia, el enrutador calcula los árboles de la ruta más corta y el árbol se puede guardar en la memoria caché para uso futuro por parte del mismo par de fuente y grupo. MOSPF es un protocolo basado en datos. Entonces, un enrutador MOSPF ve un datagrama con una fuente y una dirección de grupo dadas por primera vez, construye el árbol de ruta más corta de Dijkstra.

Protocolo de enrutamiento de multidifusión de vector de distancia (DVMRP):

DVMRP es un protocolo de enrutamiento de multidifusión que utiliza enrutamiento de vector de distancia de multidifusión. Es un protocolo de enrutamiento basado en fuente basado en RIP.

5.6 PROTOCOLOS DE LA CAPA DE RED

Cuatro protocolos de capa de red se describen a continuación:

5.6.1 PROTOCOLO DE INTERNET

El principal protocolo de red en el modelo de Internet es el Protocolo de Internet (IP). El Protocolo de Internet versión 4 (IPv4) es utilizado por el protocolo TCP/IP.

IPv4 es un protocolo poco confiable y sin conexión para una red de conmutación de paquetes que utiliza el enfoque de datagramas. Aquí, cada datagrama se maneja de forma independiente y cada datagrama puede seguir una ruta diferente desde el origen hasta el destino. Entonces, en este caso, los datagramas enviados por la misma fuente al mismo destino pueden no llegar en el mismo orden que el pedido de envío. Algunos datagramas también pueden perderse o corromperse durante la transmisión. El IPv4 no proporciona ningún mecanismo de control de flujo y control de errores. Proporciona detección de errores en el encabezado. IPv4 debe combinarse con un protocolo confiable como TCP para brindar confiabilidad.

Un datagrama en IPv4 es un paquete de longitud variable que consta de dos partes que son encabezado y datos. El encabezado tiene una longitud de 20 a 60 bytes y contiene la información necesaria para el enrutamiento y la entrega. Los diferentes campos del encabezado se dan de la siguiente manera:

Versión (VER): Es un campo de 4 bits que define la versión del protocolo IPv4. Actualmente la versión es la 4 pero puede ser reemplazada por la versión 6 en el futuro. Este campo le dice al software IPv4 que el formato del datagrama es la versión 4. Si la máquina está usando alguna otra versión de IPv4, entonces el datagrama se descarta.

Longitud del encabezado (HLEN): Es un campo de 4 bits que define la longitud total del encabezado del datagrama en palabras de 4 bytes. Este campo es obligatorio porque la longitud del encabezado es variable. Tiene una longitud de 20 a 60 bytes. Ahora, cuando no hay opciones, la longitud del encabezado es de 20 bytes, por lo que el valor de este campo es 5, lo que significa 5 palabras con cada palabra de 4 bytes de longitud. Cuando el campo de opción está en su tamaño máximo, el valor de este campo es 15, lo que significa 15 palabras con cada palabra de 4 bytes de longitud.

Servicios: Es un campo de 8 bits. Anteriormente, este campo se denominaba tipo de servicio y ahora se denomina servicios diferenciados.

Tipo de servicio: En esta interpretación, los primeros 3 bits se denominan bits de precedencia. Los siguientes 4 bits se denominan bits de tipo de servicio (TOS) y el último bit no se utiliza.

Los bits de precedencia van de 0 (000 en binario) a 7 (111 en binario). Los bits de precedencia proporcionan la prioridad del datagrama en caso de problemas como la congestión. Si un enrutador está congestionado y necesita descartar algunos datagramas, los datagramas con la precedencia más baja se descartarán primero.

TOS bits es un subcampo de 4 bits con uno y solo uno de los bits puede tener el valor de 1 en cada datagrama. Cada bit de TOS tiene un significado especial dado en la siguiente tabla:

Bits de TOS	Descripción
0000	Normal
0001	Minimice el costo
0010	Maximice la confiabilidad
0100	Maximice el rendimiento
1000	Minimizar el retraso

Servicios Diferenciados: En esta interpretación, los primeros 6 bits constituyen el subcampo de punto de código y los últimos 2 bits no se utilizan. Ahora, el subcampo de punto de código se puede usar de dos maneras diferentes, como se indica a continuación:

un. Si los 3 bits más a la derecha son 0, los 3 bits más a la izquierda se interpretan igual que los bits de precedencia en la interpretación del tipo de servicio.

B. Si los 3 bits más a la derecha no son todos 0, entonces los 6 bits definen 64 servicios en función de la asignación de prioridad por parte de Internet o las autoridades locales. Hay tres categorías de servicios. La primera categoría contiene 32 tipos de servicios. El segundo y el tercero contienen cada uno 16 tipos de servicios. Los números de la primera categoría son 0, 2, 4,...., 62
Es asignado por las autoridades de Internet (IETF). Los números para la segunda categoría son 3, 7, 11, 15 y 63. Puede ser utilizado por las autoridades locales. Los números de la tercera categoría son 1, 5, 9, 13, 17,..., 61. Es temporal y se puede utilizar con fines experimentales.

Longitud total: Sees un campo de 16 bits que define la longitud total del datagrama IPv4 en bytes.

Identificación: Es un campo de 16 bits que identifica un datagrama que se origina en el host de origen.

Banderas: Es un campo de 3 bits. Aquí el primer bit está reservado. El segundo bit se llama "bit de no fragmentar". Si su valor es 1, la máquina no debe fragmentar el datagrama. Si no puede pasar el datagrama a través de ninguna red física disponible, descarta el datagrama y envía un mensaje de error ICMP al host de origen. Si su valor es 0, el datagrama se puede fragmentar si es necesario. El tercer bit se llama "bit más fragmento". Si su valor es 1, significa que el datagrama no es el último fragmento. Si su valor es 0, significa que este es el último o único fragmento.

Compensación de fragmentación: Es un campo de 13 bits que muestra la posición relativa del fragmento con respecto al datagrama completo. Es el desplazamiento de los datos en el datagrama original medido en unidades de 8 bytes.

Tiempo para vivir(TTL): Es un campo de 8 bits. Almacena un valor que especifica el número de saltos de enrutador que el paquete aún puede viajar antes de que deba descartarse o devolverse. Cuando un host de origen envía un datagrama, almacena un valor en este campo que es aproximadamente 2 veces el número máximo de rutas entre dos hosts. Cada enrutador que procesa el datagrama decrementa el valor de este campo en 1 y ahora si este valor es cero entonces el enrutador descarta el datagrama.

Este campo es obligatorio porque a veces las tablas de enrutamiento en Internet pueden estar dañadas. Como resultado de esto, un datagrama puede viajar entre dos o más enrutadores infinitamente sin llegar al host de destino. Entonces, este campo se usa para limitar el viaje del paquete.

Protocolo: Es un campo de 8 bits que define el protocolo de nivel superior que utiliza los servicios de la capa IPv4. Un datagrama IPv4 puede encapsular datos de varios protocolos de nivel superior, como TCP, UDP, ICMP e IGMP. Este campo especifica el protocolo de destino final al que se entrega el datagrama IPv4. La siguiente tabla muestra los valores posibles en este campo y los protocolos de nivel superior correspondientes:

Valor	Protocolo
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Dirección de origen: es un campo de 32 bits que proporciona la dirección IPv4 de la fuente. Este campo debe permanecer igual durante el tiempo que el datagrama IPv4 viaja desde el host de origen hasta el host de destino.

Dirección de destino: Es un campo de 32 bits que proporciona la dirección IPv4 del destino. Este campo debe permanecer igual durante el tiempo que el datagrama IPv4 viaja desde el host de origen hasta el destino.

IPv4 tiene algunos inconvenientes y, como resultado de estos, no será adecuado para el rápido crecimiento de Internet.

-Debido al rápido crecimiento de Internet, en un futuro cercano el espacio de direcciones de IPv4 no será suficiente para acomodar todos los dispositivos en Internet.

-Internet debe adaptarse a la transmisión de audio y video en tiempo real, lo que requiere estrategias de demora mínima y reserva de recursos. Pero IPv4 no los proporciona.

-IPv4 no proporciona ningún mecanismo de encriptación y autenticación.

5.6.2 IPV6

IPv6 (Protocolo de interconexión de redes, versión 6) es la nueva versión del protocolo de capa de Internet para interconexión de redes conmutadas por paquetes y proporciona transmisión de datagramas de extremo a extremo a través de múltiples redes IP. IPv6 se describió formalmente por primera vez en el documento estándar de Internet RFC 2460. IPv6 también se conoce como IPng (Protocolo de Internet, próxima generación). En IPv6, se cambiaron el formato del paquete y la longitud de la dirección IP. Aquí también se modificaron protocolos relacionados como ICMP y otros protocolos en la red.

capas como ARP, RARP e IGMP se eliminaron o se incluyeron en el protocolo ICMPv6. También se modificaron los protocolos de enrutamiento como RIP y OSPF.

Las ventajas de IPv6 se describen a continuación:

1. La longitud de la dirección IPv6 es de 128 bits y, por lo tanto, el espacio de direcciones, 2_{128}es mucho más grande que el espacio de direcciones, 2_{32}de IPv4.
2. IPv6 usa un nuevo formato de encabezado. En este formato, las opciones están separadas del encabezado base. En IPv6, las opciones se insertan cuando es necesario entre el encabezado base y los datos de la capa superior. Por lo tanto, simplifica y aumenta la velocidad del proceso de enrutamiento porque la mayoría de las opciones no necesitan ser verificadas por los enrutadores.
3. IPv6 tiene nuevas opciones para permitir funcionalidades adicionales.
4. IPv6 permite la extensión del protocolo si lo requieren nuevas tecnologías o aplicaciones.
5. En IPv6, se ha agregado un nuevo mecanismo para admitir tráfico como audio y video en tiempo real.
6. Las opciones de encriptación y autenticación están disponibles en IPv6 que brindan confidencialidad e integridad del paquete.

Formato de paquete:

El paquete IPv6 consta de un encabezado base seguido de la carga útil. La carga útil consta de dos partes que son encabezados de extensión opcionales y datos de una capa superior. La longitud del encabezado base es de 40 bytes y la carga útil contiene hasta 65535 bytes de información.

El encabezado base consta de ocho campos que se describen a continuación: 1.

Versión: Es un campo de 4 bits que define el número de versión de la IP.

2.**Prioridad:**Es un campo de 4 bits que define la prioridad del paquete con respecto a la congestión del tráfico. La siguiente tabla muestra diferentes valores de prioridad posibles y su significado.

Prioridad	Sentido
0	Sin tráfico específico
1	Datos de fondo
2	Tráfico de datos desatendido
3	Reservado
4	Tráfico de datos masivos atendidos
5	Reservado
6	Tráfico interactivo
7	Controlar el tráfico

3. **Etiqueta de flujo**: Es un campo de 24 bits que está diseñado para proporcionar un manejo especial para un flujo de datos particular.
4. **Longitud de la carga útil:** Es un campo de 16 bits que define la longitud del datagrama IP excluyendo el encabezado base.
5. **Siguiente encabezado:** Es un campo de 8 bits que define el encabezado que sigue al encabezado base en el datagrama.
6. **Límite de salto:** Es un campo de 8 bits que almacena el número de segmentos de red en los que el paquete puede viajar antes de que un enrutador lo descarte. El host de envío establece el límite de saltos y se usa para evitar que los paquetes circulen infinitamente en una red IPv6. En el momento de reenviar un paquete IPv6, los enrutadores IPv6 deben reducir el límite de saltos en 1 y el paquete IPv6 se descarta cuando el límite de saltos es 0.
7. **Dirección de la fuente:** Es un campo de 128 bits que almacena una dirección de Internet de 128 bits para identificar la fuente original del datagrama.
8. **Dirección de destino:** Es un campo de 128 bits que almacena una dirección de Internet de 128 bits utilizada generalmente para identificar el destino final del datagrama. Pero si se utiliza el enrutamiento de origen, este campo contiene la dirección del próximo enrutador.

Encabezados de extensión:

Hay seis encabezados de extensión en formato de paquete IPv6 que se describen a continuación:

Opción Hop-by-Hop: La opción salto por salto se utiliza cuando la fuente necesita enviar información a todos los enrutadores que son visitados por el datagrama.

Enrutamiento de origen: El encabezado de extensión de enrutamiento de origen se usa para especificar una lista de nodos intermedios para que un paquete viaje en su ruta hacia su destino final.

Fragmentación: El encabezado de extensión de fragmentación se utiliza con el fin de fragmentar el datagrama en IPv6. En IPv6, solo la fuente original puede fragmentar un datagrama.

Autenticación: El encabezado de la extensión de autenticación se utiliza para validar el remitente del mensaje y garantiza la integridad de los datos.

Carga útil de seguridad cifrada: La carga útil de seguridad cifrada proporciona confidencialidad.

Opción de destino: La opción de destino se utiliza cuando el origen necesita enviar un mensaje solo al destino y los demás enrutadores no pueden acceder a este mensaje.

5.6.3 PROTOCOLO DE RESOLUCIÓN DE DIRECCIONES (ARP)

El Protocolo de resolución de direcciones (ARP) está diseñado para crear un mapeo entre direcciones físicas y lógicas. Los paquetes IP se encapsulan en una trama. Los paquetes IP usan direcciones lógicas y el marco requiere direcciones físicas. En Internet, un paquete que parte de un host de origen puede pasar por diferentes redes físicas de camino al host de destino. Ahora los hosts y enrutadores se reconocen a nivel de red por sus direcciones lógicas (IP) y a nivel físico, los hosts y enrutadores se reconocen por sus direcciones físicas.

Una dirección física es una dirección local y, por lo tanto, debe ser única en una red local, pero puede que no sea única universalmente. Se llama dirección física porque generalmente se implementa en hardware.

Ahora, la entrega de un paquete a un host o un enrutador requiere un direccionamiento tanto lógico como físico. La asignación de una dirección lógica a su dirección física correspondiente y viceversa se puede realizar mediante asignación estática o dinámica.

En el mapeo estático, se crea una tabla que asocia una dirección lógica con una dirección física. Esta tabla se almacena en cada máquina de la red. La tabla de mapeo estático debe actualizarse periódicamente porque las direcciones físicas pueden cambiar de las siguientes maneras:

1. Cuando una máquina cambia su NIC, se crea una nueva dirección física.
2. La dirección física cambia cada vez que se enciende la computadora en el caso de algunas LAN. Por ejemplo: Local Talk
3. En el caso de una computadora móvil, la dirección física cambia cada vez que se mueve de una red física a otra.

Ahora, actualizar la tabla de asignación estática podría degradar el rendimiento de la red.

En el caso del mapeo dinámico, una máquina utiliza ARP para encontrar una dirección lógica si conoce la dirección física correspondiente y viceversa. Cuando un host o un enrutador tiene un datagrama IP para enviar a otro host o enrutador, tiene la dirección lógica (IP) del receptor. Si el remitente es el host, la dirección lógica (IP) se obtiene del DNS y si el remitente es un enrutador, la dirección lógica se obtiene en una tabla de enrutamiento. Ahora el remitente requiere la dirección física del receptor. Entonces, el host o el enrutador envía un paquete de consulta ARP que incluye las direcciones IP y física del remitente y la dirección IP del receptor. Esta consulta se transmite a través de la red, por lo que cada host o enrutador de la red recibe el paquete de consulta ARP. Ahora el destinatario real reconoce su dirección IP en el paquete de consulta ARP y envía un paquete de respuesta ARP. Este paquete de respuesta ARP contiene la dirección lógica (IP) y la dirección física del destinatario. El paquete de respuesta ARP se envía directamente solo al remitente del paquete de consulta ARP utilizando la dirección física recibida en el paquete de consulta.

El rendimiento de ARP se puede mejorar utilizando la memoria caché para almacenar los paquetes de respuesta de ARP porque un sistema normalmente envía varios paquetes al mismo destino. Un sistema que almacena paquetes de respuesta ARP en la memoria caché siempre verifica la memoria caché para encontrar la asignación requerida antes de enviar una solicitud ARP.

Los diferentes campos de un paquete ARP son los siguientes:

1. El tipo de hardware es un campo de 16 bits que define el tipo de red en la que se ejecuta ARP. Cada LAN ha sido especificada por un número entero basado en su tipo.
2. El tipo de protocolo es un campo de 16 bits que define el protocolo.
3. La longitud del hardware es un campo de 8 bits que define la longitud de la dirección física en bytes.
4. La longitud del protocolo es un campo de 8 bits que define la longitud de la dirección lógica en bytes.
5. La operación es un campo de 16 bits que define el tipo de paquete que puede ser una solicitud ARP (1) o una respuesta ARP (2).
6. La dirección de hardware del remitente es un campo de longitud variable que define la dirección física del remitente.

7. La dirección del protocolo del remitente es un campo de longitud variable que define la dirección lógica del remitente. En caso de protocolo IP, la longitud de este campo es de 4 bytes.
8. La dirección de hardware de destino es un campo de longitud variable que define la dirección física del receptor.
9. La dirección del protocolo de destino es un campo de longitud variable que define la dirección lógica del receptor. En caso de protocolo IPv4, la longitud de este campo es de 4 bytes.

Proxy ARP:

Un proxy ARP es un ARP que actúa en nombre de un conjunto de hosts. Cuando un enrutador que ejecuta un proxy ARP recibe una solicitud ARP que busca la dirección IP de uno de estos hosts, el enrutador envía una respuesta ARP con su propia dirección física. Después de que el enrutador recibe el paquete IP real, envía el paquete al host o enrutador apropiado.

Entonces, al usar proxy ARP, una red puede extenderse sin el conocimiento del enrutador ascendente.

5.6.4 PROTOCOLO DE MENSAJES DE CONTROL DE INTERNET (ICMP)

El protocolo IP no tiene ningún mecanismo de notificación o corrección de errores y tampoco tiene ningún mecanismo para consultas de host y administración. Ahora, el Protocolo de mensajes de control de Internet (ICMP) está diseñado para proporcionar estos mecanismos.

Los mensajes ICMP se dividen en dos categorías, que son mensajes de informe de errores y mensajes de consulta.

Los mensajes de informe de errores informan problemas que ocurren cuando un enrutador o un host de destino procesa un paquete IP.

Un host o administrador de red obtiene información específica de un enrutador u otro host a partir de mensajes de consulta que se producen en pares.

Un mensaje ICMP tiene dos partes que son un encabezado de 8 bytes y una sección de datos de tamaño variable. Ahora los primeros 4 bytes del encabezado son comunes para todos los mensajes. El primer campo del encabezado es tipo ICMP que define el tipo del mensaje. El segundo campo del encabezado es el campo de código que especifica el motivo del tipo de mensaje en particular y el último campo común es el campo de suma de verificación. La parte restante del encabezado es específica para cada tipo de mensaje.

La sección de datos en el mensaje de error contiene la información para encontrar el paquete original donde ocurrió el error. La sección de datos en los mensajes de consulta contiene información basada en el tipo de consulta.

ICMP utiliza la dirección IP de origen para enviar el mensaje de error al origen del datagrama. En general, ICMP informa de cinco tipos de errores: destino inalcanzable, atenuación de fuente, problemas de parámetro de tiempo excedido y redirección.

Cuando un enrutador no puede enrutar un datagrama o un host no puede entregar un datagrama, el datagrama se descarta y el enrutador o el host envía un mensaje de destino inalcanzable creado por un enrutador o el host de destino, al host de origen que ha producido el datagrama. .

En el caso del protocolo IP, no hay comunicación entre el host de origen, los enrutadores y el host de destino. Entonces, en esta situación, la congestión puede ocurrir porque IP no

tienen algún mecanismo de control de flujo. Si los datagramas se reciben mucho más rápido de lo que pueden reenviarse o procesarse, el búfer de un enrutador o un host puede desbordarse ya que su tamaño es limitado. En este caso, el enrutador o el host deben descartar algunos de los datagramas. Ahora, el mensaje source-quench en ICMP proporciona una especie de control de flujo a la IP. Cuando un enrutador o host descarta un datagrama debido a la congestión, envía un mensaje de apagado de fuente al remitente del datagrama. Este mensaje informa a la fuente que el datagrama ha sido descartado y advierte a la fuente que debe ralentizar el proceso de envío ya que hay congestión en algún lugar de la ruta desde la fuente hasta el host de destino.

Los enrutadores usan tablas de enrutamiento para encontrar el siguiente enrutador para enviar paquetes. Ahora, si hay errores en una o más tablas de enrutamiento, a veces un paquete puede viajar en un bucle de un enrutador al siguiente o una serie de enrutadores infinitamente. Entonces, en esta situación, cada datagrama contiene un campo llamado tiempo de vida para controlarlo. Cuando un datagrama visita un enrutador, el valor de este campo se reduce en 1. Cuando el valor del tiempo de vida llega a 0, el enrutador descarta el datagrama y debe enviar un mensaje de tiempo excedido a la fuente original. También se genera otro mensaje de tiempo excedido cuando todas las partes de un mensaje no llegan al host de destino dentro de un límite de tiempo determinado.

Si un enrutador o el host de destino descubre un valor ambiguo o faltante en cualquier parte del datagrama, descarta el datagrama y envía un mensaje de problema de parámetro a la fuente.

En IP, tanto los enrutadores como los hosts requieren una tabla de enrutamiento para encontrar la dirección del enrutador o del próximo enrutador. Los enrutadores actualizan las tablas de enrutamiento constantemente, pero los hosts no participan en el proceso de actualización de enrutamiento, ya que la cantidad de hosts en Internet es mucho mayor que la de los enrutadores. Entonces, en general, el host usa enrutamiento estático con una tabla de enrutamiento que tiene un número limitado de entradas. Como resultado de esto, el host puede enviar un datagrama al enrutador equivocado. En este caso, el enrutador que recibe el datagrama reenviará el datagrama al enrutador correcto y enviará un mensaje de redirección al host para actualizar la tabla de enrutamiento del host.

Un mensaje de consulta se encapsula en un paquete IP y el paquete IP se encapsula en una trama de capa de enlace de datos. Los cuatro pares de mensajes de consulta se describen a continuación:

Los administradores de red y los usuarios utilizan los mensajes de solicitud de eco y de respuesta de eco para identificar problemas en la red. Este par de mensajes se puede utilizar para determinar la presencia de comunicación a nivel de IP. Los mensajes ICMP se encapsulan en datagramas IP, por lo que si una máquina que envió una solicitud de eco recibe un mensaje de respuesta de eco, significa que los protocolos IP en el remitente y el receptor se comunican entre sí utilizando el datagrama IP y los enrutadores intermedios están también recibir, procesar y reenviar datagramas IP.

Los mensajes de solicitud de marca de tiempo y de respuesta de marca de tiempo son utilizados por dos máquinas cualesquiera para determinar el tiempo de ida y vuelta necesario para que un datagrama IP viaje entre ellas. Este par de mensajes de consulta también se utiliza para sincronizar los relojes en dos máquinas.

Los mensajes de solicitud y respuesta de máscara de dirección son utilizados por un host para obtener su máscara. Un host envía un mensaje de solicitud de máscara de dirección a un enrutador en la LAN para obtener su máscara. Ahora, si el host conoce la dirección del enrutador, envía la solicitud directamente al enrutador; de lo contrario, transmite el mensaje. Después de recibir el mensaje de solicitud de máscara de dirección, el enrutador envía un mensaje de respuesta de máscara de dirección, proporcionando la máscara necesaria para el host.

Un host que desea enviar datos a un host en otra red puede obtener la dirección de los enrutadores conectados a su red mediante la solicitud de enrutador y los mensajes de consulta de publicidad. Este par de mensajes de consulta también se utiliza para saber si los enrutadores están activos y en funcionamiento. Un host puede transmitir un mensaje de solicitud de enrutador. Los enrutadores reciben el mensaje de solicitud y difunden su información de enrutamiento utilizando el mensaje de anuncio de enrutador.

Hay dos herramientas que son ping y traceroute que se pueden usar en Internet para la depuración. Estas herramientas utilizan ICMP para la depuración.

El programa ping se usa para determinar si un host está vivo y responde.

El programa traceroute en UNIX o tracert en Windows se puede utilizar para rastrear la ruta de un paquete desde el origen hasta el destino.

REVISA TU PROGRESO

1. Preguntas de opción múltiple:

(I) La longitud de la dirección IPv4 es

A 32 bits

B 128 bits

C 32 bytes

D. Ninguna de las anteriores

(II) El tamaño del espacio de direcciones de IPv6 es

A $2_{dieciséis}$

B 2_{32}

C 2_{128}

D. Tanto B como C

(III) ¿Cuál no es una subcategoría de direcciones reservadas en IPv6?

A. Dirección de bucle invertido

B. Dirección asignada

C. Dirección local

D. Dirección compatible

(IV) ¿Cuál es la estrategia de toma de decisiones del algoritmo de enrutamiento de vector de distancia de multidifusión?

A. Inundaciones

B. Reenvío de ruta inversa

C. Transmisión de ruta inversa

Todo lo anterior

(V) ¿Cuál es un protocolo de enrutamiento de unidifusión?

A. Abrir el protocolo Primero la ruta más corta

B. Protocolo de resolución de direcciones.

C. Protocolo DVMRP

Re. Ninguna de las anteriores

(VI) Qué protocolo está diseñado para crear un mapeo entre direcciones físicas y lógicas.

A. Protocolo de mensajes de control de Internet

B Protocolo de Internet

C. Protocolo de resolución de direcciones.

D. Tanto B como C

(VII) La longitud de la parte del encabezado en un mensaje del Protocolo de mensajes de control de Internet

A. 2 bytes

B 4 bytes

C 8 bytes

D 16 bytes

(VIII) El protocolo de información de enrutamiento es un

A. Protocolo de enrutamiento de multidifusión

B. Protocolo de enrutamiento de difusión

C. Protocolo de enrutamiento de unidifusión

Re. Ninguna de las anteriores

(IX) ¿Cuál no es un tipo de enlace especificado en el protocolo OSPF?

A. Punto a punto

B. Transitorio

C. virtuales

D. Columna vertebral

(X) ¿Cuál no es un algoritmo de enrutamiento dinámico?

A. Inundaciones

B. Algoritmo de enrutamiento por vector de distancia

C. Algoritmo de enrutamiento de estado de enlace

D. Tanto B como C

2. Complete los espacios en blanco:

I. La interred es una red _____.

II. El servicio _____ se utiliza en el enfoque de datagramas para la conmutación de paquetes.

tercero El término dirección IP se usa para referirse a una dirección _____ en la capa de red del conjunto de protocolos TCP/IP.

IV. La longitud de una dirección IPv6 es _____ bytes.

V. En IPv6, las direcciones reservadas comienzan con _____s.

VI. En el caso del protocolo OSPF, hay un área especial dentro de un sistema autónomo llamado _____.

VIII. El Protocolo de enrutamiento de multidifusión de vector de distancia (DVMRP) es un protocolo de enrutamiento basado en fuente basado en _____.

VIII. El rendimiento de ARP se puede mejorar utilizando _____memoria.

IX. Un mensaje del Protocolo de mensajes de control de Internet tiene dos partes: _____ y _____.

X. Si el valor decimal del primer byte de una dirección IPv4 está en el rango _____, entonces es una dirección de clase B.

3. Indique si las siguientes afirmaciones son verdaderas o falsas

I. Si el valor decimal del primer byte de una dirección IPv4 está en el rango de 192 a 223, entonces es una dirección de clase A.

II. El concepto básico de NAT es asignar a cada empresa un gran conjunto de direcciones internamente y una dirección o un pequeño conjunto de direcciones externamente.

tercero El enrutamiento de estado de enlace es un algoritmo de enrutamiento estático.

IV. El tipo de hardware es un campo de 16 bits que define el tipo de red en la que se ejecuta ARP.

V. El protocolo MOSPF utiliza el enrutamiento de estado de enlace de multidifusión para crear árboles basados en fuentes.

VI. IPv4 es un protocolo poco fiable y orientado a la conexión.

VIII. El paquete IPv6 consta de un encabezado base seguido de la carga útil.

VIII. Usando proxy ARP, una red puede extenderse sin el conocimiento del enrutador ascendente.

IX. El protocolo IP tiene un mecanismo de notificación o corrección de errores.

X. El programa Ping se usa para determinar si un host está vivo y responde.

5.7 RESUMAMOS

El resumen de esta unidad es el siguiente:

- La capa de red proporciona el mecanismo de transferencia de secuencias de datos de longitud variable desde un host de origen en una red a un host de destino en una red diferente.
- Una colección de redes interconectadas, que permite que los datos se muevan libremente entre diferentes redes, se denomina interconexión de redes o internet.
- Una dirección IPv4 es una dirección de 32 bits que define de forma única y global la conexión de un dispositivo a Internet.
- Se utilizan dos tipos de notaciones para mostrar una dirección IPv4: notación binaria y notación decimal con puntos.
- Hay dos esquemas de direccionamiento en IPv4: direccionamiento con clase y direccionamiento sin clase.
- La traducción de direcciones de red (NAT) es una solución para el agotamiento de direcciones en el direccionamiento IPv4.
- IPv6 tiene un espacio de direcciones de 2^{128}direcciones porque utiliza una dirección de 128 bits. IPv6
- especifica la notación de dos puntos hexadecimales para sus direcciones, donde 128 bits se dividen en ocho secciones y cada una tiene 2 bytes de longitud.

- En el caso de IPv6, las direcciones IP se dividen en varias categorías: Direcciones Unicast, Direcciones Multicast, Direcciones Anycast, Direcciones Reservadas y Direcciones Locales.
- Los algoritmos de enrutamiento son la parte del software de capa de red responsable de decidir las rutas y las estructuras de datos para transmitir los paquetes entrantes de manera eficiente.
- Los diferentes objetivos de un algoritmo de enrutamiento son: Corrección, Simplicidad, Robustez, Estabilidad, Imparcialidad y Optimalidad.
- Los algoritmos de enrutamiento se pueden dividir en dos clases principales: algoritmos adaptativos y no adaptativos.
- Los algoritmos no adaptativos son algoritmos de enrutamiento estático en los que la elección de la ruta para transmitir paquetes IP de un nodo a otro se calcula de antemano y se descarga a los enrutadores cuando se inicia la red.
- Los algoritmos adaptativos son algoritmos dinámicos en los que las decisiones de enrutamiento cambian cada vez que hay un cambio en la topología y el tráfico de la red. Los protocolos de
- enrutamiento se dividen en categorías: protocolos de unidifusión y multidifusión.
- En el enrutamiento de unidifusión, cuando un enrutador recibe un paquete para enrutar, necesita encontrar la ruta más corta hacia el destino del paquete.
- En el enrutamiento de multidifusión, el enrutador recibe paquetes de multidifusión para enrutar a los destinos en más de una red.
- El enrutamiento por vector de distancia y el enrutamiento por estado de enlace son dos algoritmos de enrutamiento dinámico.
- El Protocolo de información de enrutamiento (RIP) es un protocolo de enrutamiento intradominio que se utiliza dentro de un sistema autónomo basado en el enrutamiento por vector de distancia.
- El protocolo de ruta más corta abierta es un protocolo de enrutamiento de unidifusión intradominio basado en el enrutamiento de estado de enlace.
- El enrutamiento de estado de enlace de multidifusión es una extensión del enrutamiento de estado de enlace de unidifusión.
- El enrutamiento por vector de distancia de multidifusión es la extensión del enrutamiento por vector de distancia de unidifusión.
- El protocolo Multicast Open Shortest Path First (MOSPF) es una extensión del protocolo OSPF.
- DVMRP es un protocolo de enrutamiento de multidifusión que utiliza enrutamiento de vector de distancia de multidifusión.
- El Protocolo de Internet versión 4 (IPv4) es utilizado por el protocolo TCP/IP.
- IPv4 es un protocolo poco confiable y sin conexión para una red de conmutación de paquetes que utiliza el enfoque de datagramas.
- IPv6 (Protocolo de interconexión de redes, versión 6) es la nueva versión del protocolo de capa de Internet para interconexión de redes conmutadas por paquetes y proporciona transmisión de datagramas de extremo a extremo a través de múltiples redes IP.
- El paquete IPv6 consta de un encabezado base seguido de la carga útil.
- El Protocolo de resolución de direcciones (ARP) está diseñado para crear un mapeo entre direcciones físicas y lógicas.

- El Protocolo de mensajes de control de Internet (ICMP) está diseñado para proporcionar informes de errores o un mecanismo de corrección de errores. También proporciona un mecanismo para consultas de host y administración.
- Los mensajes ICMP se dividen en dos categorías, que son mensajes de informe de errores y mensajes de consulta.

5.8 RESPUESTAS PARA COMPROBAR TU PROGRESO

1. (I) A , (II) C , (III) C , (IV) D , (V) A ,(VI) C ,(VII) C ,(VIII) C ,(IX) D ,(X) A

2. I. conmutación de paquetes, II. sin conexión, III. lógico, IV. 16 , V. ocho 0 , VI. columna vertebral, VII. DEP, VIII. caché, IX. Cabecera, sección de datos, X. 128 a 191.

3. I. Falso, II. Cierto, III. Falso, IV. Verdadero, V. Verdadero, VI. Falso, VII. Cierto, VIII. Cierto , IX. Falso , X. Verdadero

5.9 LECTURAS ADICIONALES

- Behrouz A Forouzan: Comunicaciones de datos y redes, TATA McGraw Hill
- William Stallings: Datos y comunicaciones informáticas, Pearson Education
- Andrew S. Tanenbaum: Redes informáticas, Prentice-Hall India

5.10 PREGUNTAS MODELO

- Explique el esquema de direccionamiento diferente en Pv4.
- Definir enrutamiento. ¿Cuáles son los diferentes objetivos del algoritmo de enrutamiento?
- Explicar el enrutamiento por vector de distancia y el enrutamiento por estado de enlace.
- Explicar los diferentes campos de un paquete IPv4.
- Explicar el Protocolo de resolución de direcciones (ARP).

UNIDAD 6: LA CAPA DE TRANSPORTE

ESTRUCTURA DE LA UNIDAD

6.1 INTRODUCCIÓN

La capa de transporte es 4elcapa del modelo OSI de la red. La capa de transporte del modelo OSI ofrece comunicación de extremo a extremo entre dispositivos finales a través de una red. También ofrece comunicación de aplicación a aplicación entre las dos aplicaciones en ambos hosts. Dependiendo de la aplicación, la capa de transporte ofrece comunicaciones confiables, orientadas a conexión o sin conexión, de mejor esfuerzo.

Los dos protocolos de capa de transporte más comunes son el Protocolo de control de transmisión (TCP) TCP orientado a conexión y el Protocolo de datagrama de usuario (UDP) UDP sin conexión.

La capa de transporte garantiza que los mensajes se entreguen sin errores, en secuencia y sin pérdidas ni duplicaciones. Libera a los protocolos de capa superior de cualquier problema con la transferencia de datos entre ellos y sus pares. El tamaño y la complejidad de un protocolo de transporte depende del tipo de servicio que pueda obtener de la capa de red. Para una capa de red confiable con capacidad de circuito virtual, se requiere una capa de transporte mínima. Si la capa de red no es confiable y/o solo admite datagramas, el protocolo de transporte debe incluir una amplia detección y recuperación de errores.

La capa de transporte proporciona la transferencia de datos entre los usuarios finales, proporcionando servicios de transferencia de datos fiables a las capas superiores. Controla la confiabilidad de un enlace dado a través del control de flujo, segmentación/dessegmentación y control de errores. La capa de transporte puede realizar un seguimiento de los segmentos y retransmitir los que fallan. La capa de transporte también proporciona el reconocimiento de la transmisión de datos exitosa y envía los siguientes datos si no ocurrieron errores.

La capa de transporte proporciona:

- **Segmentación de mensajes:**Acepta un mensaje de la capa (de sesión) superior, divide el mensaje en unidades más pequeñas, si es necesario, y pasa las unidades más pequeñas a la capa de red. La capa de transporte en el extremo de destino vuelve a ensamblar el mensaje para recuperar el mensaje original.
- **Confirmación de mensaje:**Proporciona entrega confiable de mensajes de extremo a extremo con acuses de recibo.
- **Control de tráfico de mensajes:**Le dice al host transmisor que "espere un momento" cuando no hay búferes de mensajes disponibles en el host receptor.
- **Multiplexación de sesiones:**Multiplexa varios flujos de mensajes o sesiones en un enlace lógico y realiza un seguimiento de qué mensajes pertenecen a qué sesiones.

Por lo general, la capa de transporte puede aceptar mensajes relativamente grandes, pero existen límites estrictos de tamaño de mensaje impuestos por la capa de red (o inferior). En consecuencia, la capa de transporte debe dividir los mensajes en unidades más pequeñas, o marcos, colocar un encabezado de capa de transporte en cada marco. El encabezado de la capa de transporte incluye información de control, como indicadores de inicio y finalización del mensaje, para permitir que la capa de transporte en el otro extremo reconozca

límites del mensaje. Además, si las capas inferiores no mantienen la secuencia, el encabezado de transporte debe contener información de secuencia para permitir que la capa de transporte en el extremo receptor reúna las piezas en el orden correcto antes de entregar el mensaje recibido a la capa superior.

Al residir entre las capas de aplicación y red, la capa de transporte tiene la función fundamental de proporcionar servicios de comunicación directamente a los procesos de aplicación que se ejecutan en diferentes hosts. En este capítulo, examinaremos los posibles servicios proporcionados por un protocolo de capa de transporte y los principios subyacentes a varios enfoques para proporcionar estos servicios.

6.1.1 Relación entre las capas de transporte y red

En teoría, la capa de transporte y la capa de red son distintas, pero en la práctica suelen estar muy relacionadas entre sí. Podemos ver esto fácilmente con solo mirar los nombres de las pilas de protocolos comunes; a menudo reciben el nombre de los protocolos de capa tres y cuatro en la suite, lo que implica su estrecha relación. Por ejemplo, el nombre “TCP/IP” proviene del protocolo de capa de transporte (TCP) y del protocolo de capa de red (IP) más utilizados de la suite. De manera similar, la suite Novell NetWare a menudo se denomina "IPX/SPX" por sus protocolos de capa tres (IPX) y capa cuatro (SPX). Por lo general, los protocolos de capa de transporte específicos consideran las capas de red en la misma familia.

En el extremo de envío, la capa de transporte del software de red recibe información de la capa anterior. En el caso del protocolo TCP/IP, es la capa de sesión. Luego, la capa de transporte los procesa de acuerdo con su propio mecanismo. Después del proceso, los marcos se transfieren a la siguiente capa inferior del traje de protocolo. Para TCP/IP, es la capa de red. La capa de red los prepara para transferir a través del medio físico utilizado en la red.

En el extremo receptor, se ejecuta el proceso inverso. Las tramas entrantes son aceptadas por la capa de red. La capa de red elimina el encabezado de la capa de red y procesa esos marcos en consecuencia. Una vez que finaliza el procesamiento de esta capa, los marcos se entregan a la capa de transporte.

Los protocolos de capa de transporte más utilizados son el Protocolo de control de transmisión (TCP) y el Protocolo de datagramas de usuario (UDP) en la suite TCP/IP; el protocolo Sequenced Packet Exchange (SPX) en el conjunto de protocolos NetWare; y el protocolo NetBEUI en la suite NetBIOS/NetBEUI/NBF.

6.1.2 Descripción general de la capa de transporte en Internet

La columna vertebral de Internet es el protocolo TCP/IP. Este traje de protocolo hace posible la transferencia de datos entre redes heterogéneas. Utiliza dos protocolos de capa de transporte distintos y los pone a disposición de la capa de aplicación. Uno es**UDP**(Protocolo de datagramas de usuario), que proporciona un servicio no confiable y sin conexión a la aplicación que lo invoca. el segundo es **TCP**(Protocolo de control de transmisión), que proporciona un servicio confiable y orientado a la conexión a la aplicación que lo invoca. Al diseñar una aplicación de red utilizando el protocolo TCP/IP, el desarrollador de la aplicación debe especificar cualquiera de estos dos protocolos de capa de transporte.

6.1.3 Puertos y enchufes

Esta sección introduce los conceptos de**Puerto**y**enchufe**, que son necesarios para determinar qué aplicación local en el extremo emisor realmente desea comunicarse con qué aplicación remota en el extremo receptor.

6.1.3.1 Puertos

Un puerto es un número de 16 bits, utilizado por el protocolo de host a host para identificar a qué protocolo de nivel superior o proceso de aplicación debe entregar los mensajes entrantes.

Hay dos tipos de puerto:

• **Conocido:**Los puertos conocidos pertenecen a aplicaciones de servidor estándar. Por ejemplo, Telnet usa el número de puerto conocido 23, HTTP usa el número de puerto conocido 80, etc. Los números de puerto conocidos varían entre 1 y 1023. La mayoría de los servidores requieren solo un puerto. Las excepciones son el servidor FTP, que usa dos: 20 y 21. Los puertos conocidos están controlados y asignados por la Autoridad de números asignados de Internet (IANA) y en la mayoría de los sistemas solo pueden ser utilizados por procesos del sistema o por programas ejecutados por usuarios privilegiados. . El motivo de los puertos conocidos es permitir que los clientes puedan encontrar servidores sin información de configuración. Los números de puerto conocidos se definen en STD 2 - Números de Internet asignados

• **Efímero:**Los clientes no necesitan números de puerto conocidos porque inician la comunicación con los servidores y el número de puerto que utilizan está contenido en el datagrama TCP/UDP enviado al servidor. A cada aplicación cliente se le asigna un número de puerto durante el tiempo que lo necesite por parte del host en el que se ejecuta. Los números de puerto efímeros tienen valores superiores a 1023, normalmente en el rango de 1024 a 65535. Un cliente puede utilizar cualquier número que se le haya asignado, siempre que la combinación*de <protocolo de transporte, dirección IP, número de puerto>* es único. Los puertos efímeros no están controlados por IANA y pueden ser utilizados por programas ordinarios desarrollados por usuarios en la mayoría de los sistemas.

Debido a que dos aplicaciones diferentes intentan usar los mismos números de puerto en un host, se evita la confusión al escribir esas aplicaciones para solicitar un puerto disponible de TCP/IP. Debido a que este número de puerto se asigna dinámicamente, puede diferir de una invocación de una aplicación a la siguiente. UDP y TCP utilizan el mismo principio de puerto. En la mayor medida posible, se utilizan los mismos números de puerto para los mismos servicios además de UDP y TCP.

6.1.3.2 Enchufes

Dado que la capa de transporte en el host receptor entrega datos al socket, debe haber un identificador único para cada socket.

El identificador de socket se llama dirección de socket.

Dirección de socket = Dirección IP: Número de puerto (Figura 1)

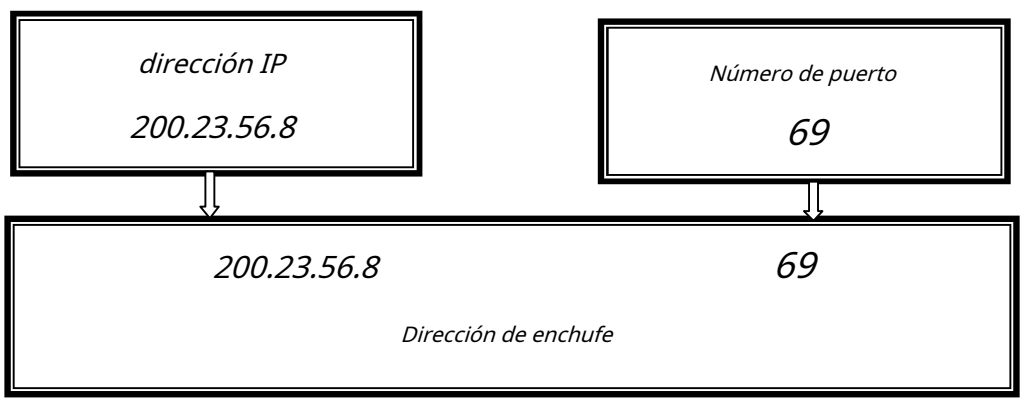


Figura 1: Dirección del zócalo

• Un socket es un tipo especial de identificador de archivo que utiliza una aplicación para solicitar servicios de red del sistema operativo.

• Una dirección de socket es: <dirección IP y aplicación local>

Por ejemplo, en la suite TCP/IP: <193.44.234.3, 12345>

Un socket es un punto final para la comunicación que se puede nombrar y direccionar en una red. Dos aplicaciones en dos PC remotas se comunican a través de sockets TCP. El modelo de socket proporciona a una aplicación una conexión con otra aplicación. Estas instalaciones son proporcionadas por TCP. TCP utiliza el mismo principio de puerto que UDP para proporcionar la comunicación. Al igual que UDP, TCP utiliza puertos conocidos y efímeros. Si dos aplicaciones se comunican a través de TCP, tienen una conexión lógica que es identificable únicamente por los dos sockets involucrados, es decir, por la combinación <dirección IP local, puerto local, dirección IP remota, puerto remoto>.

6.2 Entrega de proceso a proceso

La capa de transporte es responsable de la entrega de proceso a proceso: la entrega de un paquete, parte de un mensaje, de una aplicación a otra a través de la red. Dos procesos se comunican en una arquitectura cliente/servidor. UDP y TCP son protocolos de capa de transporte que crean una comunicación de proceso a proceso. UDP es un protocolo poco confiable y sin conexión que requiere poca sobrecarga y ofrece una entrega rápida.

En el paradigma cliente-servidor, un programa de aplicación en el host local, llamado cliente, necesita servicios de un programa de aplicación en el host remoto, llamado servidor. Cada programa de aplicación tiene un número de puerto único que lo distingue de otros programas que se ejecutan al mismo tiempo en la misma máquina. Al programa cliente se le asigna un número de puerto aleatorio denominado número de puerto efímero. Al programa servidor se le asigna un número de puerto universal denominado número de puerto conocido. La combinación de la dirección IP y el número de puerto, denominada dirección de socket, define de forma única un proceso y un host. TCP utiliza un mecanismo de ventana deslizante para el control de flujo.

6.3 Protocolo de datagrama de usuario (UDP)

UDP es un protocolo de transporte sin conexión y poco fiable. No agrega nada a los servicios de IP excepto proporcionar comunicación de proceso a proceso en lugar de comunicación de host a host. UDP es un protocolo muy simple que utiliza un mínimo de sobrecarga. Si un proceso quiere enviar un pequeño mensaje y no le importa mucho la confiabilidad, puede usar UDP.

UDP utiliza un modelo de transmisión simple con un mínimo de mecanismo de protocolo. No tiene diálogos de negociación y, por lo tanto, expone cualquier falta de confiabilidad del protocolo de red subyacente al programa del usuario. Como normalmente se trata de IP sobre medios poco fiables, no hay garantía de entrega, pedido o protección duplicada. UDP proporciona la suma de verificación para la integridad de los datos y el número de puerto para abordar diferentes funciones en el origen y el destino del datagrama. UDP es adecuado para fines en los que la verificación y corrección de errores no es necesaria o no se realiza en la aplicación, lo que evita la sobrecarga de dicho procesamiento en el nivel de la interfaz de red. Las aplicaciones sensibles al tiempo a menudo usan UDP porque descartar paquetes es preferible a esperar paquetes retrasados, lo que puede no ser una opción en un sistema en tiempo real.

El datagrama UDP tiene un encabezado de 8 bytes que se divide en cuatro campos, cada uno con 2 bytes. (Figura 2)

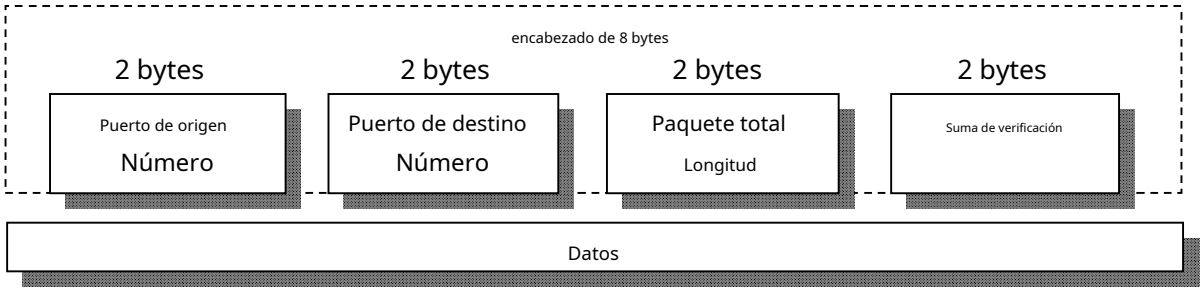


Figura 2: datagrama UDP

1. Número de puerto de origen: Este es el número de puerto utilizado por el proceso que se ejecuta en el host de origen.

Este campo identifica el puerto del remitente cuando es significativo y se debe suponer que es el puerto al que responder si es necesario. Si no se usa, entonces debería ser cero. Si el host de origen es el cliente, es probable que el número de puerto sea un número de puerto efímero. Si el host de origen es el servidor, es probable que el número de puerto sea un número de puerto conocido.

2. Número de puerto de destino: Este es el número de puerto utilizado por el proceso que se ejecuta en el host de destino. Este campo identifica el puerto del receptor y es obligatorio. Similar al número de puerto de origen, si el cliente es el host de destino, entonces el número de puerto probablemente será un número de puerto efímero y si el host de destino es el servidor, entonces el número de puerto probablemente será un número de puerto conocido.

3. Longitud total del paquete: Define la longitud total del datagrama de usuario. Un campo que especifica la longitud en bytes de todo el datagrama: encabezado y datos. La longitud mínima es de 8 bytes, ya que esa es la longitud del encabezado. El tamaño del campo establece un límite teórico de 65 535 bytes (8 bytes de encabezado + 65 527 bytes de datos) para un datagrama UDP. El límite práctico para la longitud de datos que impone el protocolo IPV4 subyacente es de 65 507 bytes (65 535 − encabezado UDP de 8 bytes − encabezado IP de 20 bytes).

4. Suma de comprobación: El campo de suma de comprobación se utiliza para la comprobación de errores del encabezado.*y* datos. Si el transmisor no genera ninguna suma de comprobación, el campo utiliza el valor todos ceros. Este campo no es opcional para IPv6.

6.3.1 Aplicaciones de UDP

Numerosas aplicaciones clave de Internet utilizan UDP, entre ellas: el Sistema de nombres de dominio (DNS), donde las consultas deben ser rápidas y consistir en una única solicitud seguida de un único paquete de respuesta, el Protocolo simple de administración de red (SNMP), el Protocolo de información de rutina (RIP) y el Protocolo de configuración dinámica de host (DHCP). El tráfico de voz y video generalmente se transmite mediante UDP. Los protocolos de transmisión de audio y video en tiempo real están diseñados para manejar paquetes perdidos ocasionalmente, por lo que solo se produce una ligera degradación en la calidad, en lugar de grandes retrasos si los paquetes perdidos se retransmiten.

6.3.2 Idoneidad de UDP para ciertas aplicaciones

- Está**orientado a transacciones**, adecuado para protocolos simples de consulta y respuesta, como los sistemas de nombres de dominio o el protocolo de tiempo de red.
- Proporciona datagramas adecuados para modelar otros protocolos, como túneles IP o llamadas a procedimientos remotos y el sistema de archivos de red.
- Está**sencillo**, adecuado para arranque u otros fines sin una pila de protocolos completa, como DHCP y Trivial File Transfer Protocol.
- Está**apátrida**, adecuado para un gran número de clientes, como en aplicaciones de transmisión de medios, por ejemplo, IPTV.
- los**falta de retrasos en la retransmisión**lo hace adecuado para aplicaciones en tiempo real como Voz sobre IP (VoIP), juegos en línea y muchos protocolos creados sobre el Protocolo de transmisión en tiempo real.
- funciona bien en**unidireccional**comunicación, adecuada para la información de transmisión, como en muchos tipos de descubrimiento de servicios e información compartida, como el tiempo de transmisión o el Protocolo de información de enrutamiento.
- Enviar un mensaje pequeño mediante UDP requiere mucha menos interacción entre el remitente y el receptor que usar TCP o SCTP. Los paquetes UDP, llamados datagramas de usuario, tienen un encabezado de tamaño fijo de 8 bytes.

6.3.3 Algunos inconvenientes de UDP

- No hay control de flujo: el receptor puede desbordarse con los mensajes entrantes.
- No existe un mecanismo de control de errores en UDP excepto por la suma de verificación.
- El remitente no sabe si un mensaje se ha perdido o duplicado.
- Cuando el receptor detecta un error a través de la suma de verificación, el datagrama de usuario se descarta.
- Cada datagrama de usuario puede viajar por un camino diferente. No hay relación entre los diferentes datagramas de usuario, incluso si provienen del mismo proceso de origen y van al mismo programa de destino. Además, no hay establecimiento de conexión ni terminación de conexión.

- UDP no ofrece garantías al protocolo de capa superior para la entrega de mensajes y la capa de protocolo UDP no retiene el estado de los mensajes UDP una vez enviados. Por esta razón, UDP a veces se denomina Protocolo de datagramas no confiable.

6.4 Protocolo de control de transmisión (TCP)

los**Protocolo de Control de Transmisión**(**TCP**) es uno de los dos protocolos centrales originales del conjunto de protocolos de Internet (IP) y es tan común que el conjunto completo suele denominarse TCP/IP. TCP proporciona una entrega confiable, ordenada y con verificación de errores de un flujo de octetos entre programas que se ejecutan en computadoras conectadas a una intranet o a la Internet pública. Los navegadores lo usan cuando se conectan a servidores en los sitios de la World Wide Web, y se usa para enviar correos electrónicos y transferir archivos de una ubicación a otra con precisión. Las aplicaciones que no requieren la confiabilidad de una conexión TCP pueden usar el Protocolo de datagramas de usuario (UDP) sin conexión.

El protocolo corresponde a la capa de transporte de la suite TCP/IP. TCP proporciona un servicio de comunicación a un nivel intermedio entre un programa de aplicación y el Protocolo de Internet (IP). Es decir, cuando un programa de aplicación desea enviar una gran cantidad de datos a través de Internet utilizando IP, en lugar de dividir los datos en partes del tamaño de IP y emitir una serie de solicitudes de IP, el software puede emitir una sola solicitud a TCP y dejar que TCP maneja los detalles de IP.

Debido a la congestión de la red, el equilibrio de la carga del tráfico u otro comportamiento impredecible de la red, los paquetes IP pueden perderse, duplicarse o entregarse desordenados. TCP detecta estos problemas, solicita la retransmisión de datos perdidos, reorganiza los datos desordenados e incluso ayuda a minimizar la congestión de la red para reducir la ocurrencia de otros problemas. Una vez que el receptor TCP ha reensamblado la secuencia de octetos transmitidos originalmente, los pasa al programa de aplicación. Por lo tanto, TCP abstrae la comunicación de la aplicación de los detalles de red subyacentes.

TCP es ampliamente utilizado por muchas de las aplicaciones más populares de Internet, incluida la World Wide Web (WWW), el correo electrónico, el protocolo de transferencia de archivos, Secure Shell, el uso compartido de archivos entre pares y algunas aplicaciones de transmisión de medios.

TCP está optimizado para una entrega precisa en lugar de una entrega oportuna y, por lo tanto, TCP a veces incurre en retrasos relativamente largos (del orden de segundos) mientras espera mensajes fuera de servicio o retransmisiones de mensajes perdidos. No es particularmente adecuado para aplicaciones en tiempo real como VoIP. Para este tipo de aplicaciones, se suelen recomendar protocolos como el Protocolo de transporte en tiempo real (RTP) que se ejecuta sobre el Protocolo de datagramas de usuario (UDP).

TCP es un servicio de entrega de transmisión confiable que garantiza que todos los bytes recibidos serán idénticos a los bytes enviados y en el orden correcto. Dado que la transferencia de paquetes no es confiable, se utiliza una técnica conocida como acuse de recibo positivo con retransmisión para garantizar la confiabilidad de las transferencias de paquetes. Esta técnica fundamental requiere que el receptor responda con un mensaje de reconocimiento a medida que recibe los datos. El remitente mantiene un registro de cada paquete que envía. El remitente también mantiene un cronómetro desde el momento en que se envió el paquete, y

retransmite un paquete si el temporizador expira antes de que se haya reconocido el mensaje. El temporizador es necesario en caso de que un paquete se pierda o se corrompa.

TCP consta de un conjunto de reglas: para el protocolo, que se utilizan con el Protocolo de Internet, y para el IP, para enviar datos "en forma de unidades de mensaje" entre computadoras a través de Internet. Mientras que IP maneja la entrega real de los datos, TCP realiza un seguimiento de las unidades individuales de transmisión de datos, llamadas segmentos en los que se divide un mensaje para un enrutamiento eficiente a través de la red. Por ejemplo, cuando se envía un archivo HTML desde un servidor web, la capa de software TCP de ese servidor divide la secuencia de octetos del archivo en segmentos y los reenvía individualmente a la capa de software IP (capa de Internet). La capa de Internet encapsula cada segmento TCP en un paquete IP agregando un encabezado que incluye (entre otros datos) la dirección IP de destino. Aunque todos los paquetes tienen la misma dirección de destino, se pueden enrutar en diferentes caminos a través de la red. Cuando el programa cliente en la computadora de destino los recibe, la capa TCP (capa de transporte) vuelve a ensamblar los segmentos individuales y garantiza que estén ordenados correctamente y sin errores mientras los transmite a una aplicación.

6.4.1. TCP se puede caracterizar por las siguientes facilidades que proporciona a las aplicaciones que lo utilizan:

- **Transmisión de transferencia de datos:**

Desde el punto de vista de la aplicación, TCP transfiere un flujo continuo de bytes a través de la red. TCP hace esto agrupando los bytes en segmentos TCP, que se pasan a IP para su transmisión al destino. Además, el propio TCP decide cómo segmentar los datos y puede reenviarlos a su conveniencia. A veces, una aplicación necesita asegurarse de que todos los datos pasados a TCP se hayan transmitido realmente al destino. Por esa razón, se define una función de empuje. Empujará todo el segmento TCP restante aún en almacenamiento al host de destino. La función de conexión cercana normal también envía los datos al destino.

• Fiabilidad:

Usos de TCP*un número de secuencia*para identificar cada byte de datos. El número de secuencia identifica el orden de los bytes enviados desde cada computadora para que los datos puedan reconstruirse en orden, independientemente de cualquier fragmentación, desorden o pérdida de paquetes que pueda ocurrir durante la transmisión. Por cada byte de carga útil transmitido, el número de secuencia debe incrementarse. En los primeros dos pasos del protocolo de enlace de 3 vías, ambas computadoras intercambian un número de secuencia inicial (ISN). Este número puede ser arbitrario y, de hecho, debería ser impredecible para defenderse de los ataques de predicción de secuencia TCP.

TCP utiliza principalmente un*acuse de recibo acumulativo*esquema, donde el receptor envía un reconocimiento que significa que el receptor ha recibido todos los datos que preceden al número de secuencia reconocido. El remitente establece el campo de número de secuencia en el número de secuencia del primer byte de carga útil en el campo de datos del segmento, y el receptor envía un acuse de recibo especificando el número de secuencia del siguiente byte que esperan recibir. Por ejemplo, si una computadora emisora envía un paquete que contiene cuatro bytes de carga útil con un campo de número de secuencia de 100, entonces los números de secuencia de los cuatro bytes de carga útil son 100,

101, 102 y 103. Cuando este paquete llega a la computadora receptora, enviaría un número de reconocimiento de 104, ya que ese es el número de secuencia del siguiente byte que espera recibir en el siguiente paquete.

Además de los acuses de recibo acumulativos, los receptores TCP también pueden enviar *reconocimientos selectivos* para proporcionar más información. Si el remitente infiere que se han perdido datos en la red, retransmite los datos.

- **Control de flujo**

El TCP receptor, cuando envía un acuse de recibo al remitente, también indica al remitente la cantidad de bytes que puede recibir más allá del último segmento TCP recibido, sin causar saturación ni desbordamiento en sus búferes internos. Esto se envía en el acuse de recibo en forma del número de secuencia más alto que puede recibir sin problemas. Este mecanismo también se conoce como *mecanismo de ventana*.

TCP utiliza un protocolo de control de flujo de extremo a extremo para evitar que el remitente envíe datos demasiado rápido para que el receptor TCP los reciba y procese de manera confiable. Tener un mecanismo para el control de flujo es fundamental en un entorno donde se comunican máquinas de diversas velocidades de red. Por ejemplo, si una PC envía datos a un teléfono inteligente que procesa lentamente los datos recibidos, el teléfono inteligente debe regular el flujo de datos para no sobrecargarse.

TCP utiliza un *protocolo de control de flujo de ventana deslizante*. En cada segmento TCP, el receptor especifica en el *recibir ventana* campo la cantidad de datos recibidos adicionales (en bytes) que está dispuesto a almacenar en el búfer para la conexión. El host emisor puede enviar solo hasta esa cantidad de datos antes de que deba esperar un reconocimiento y una actualización de la ventana del host receptor.

Los números de secuencia TCP y las ventanas de recepción se comportan como un reloj. La ventana de recepción cambia cada vez que el receptor recibe y reconoce un nuevo segmento de datos. Una vez que se queda sin números de secuencia, el número de secuencia vuelve a 0.

Cuando un receptor anuncia un tamaño de ventana de 0, el remitente deja de enviar datos y comienza el *temporizador persistente*. El temporizador de persistencia se usa para proteger TCP de una situación de interbloqueo que podría surgir si se pierde una actualización posterior del tamaño de la ventana del receptor y el remitente no puede enviar más datos hasta que reciba una nueva actualización del tamaño de la ventana del receptor. Cuando el temporizador de persistencia expira, el remitente TCP intenta la recuperación enviando un paquete pequeño para que el receptor responda enviando otro reconocimiento que contiene el nuevo tamaño de ventana.

- **Control de congestión**

Otro aspecto principal de TCP es el control de congestión. TCP utiliza una serie de mecanismos para lograr un alto rendimiento y evitar el colapso de la congestión, donde el rendimiento de la red puede caer en varios órdenes de magnitud. Estos mecanismos controlan la tasa de datos que ingresan a la red, manteniendo el flujo de datos por debajo de una tasa que desencadenaría el colapso. También producen una asignación justa aproximadamente máximo-mínimo entre flujos.

Los remitentes utilizan los acuses de recibo de los datos enviados, o la falta de acuses de recibo, para inferir las condiciones de la red entre el remitente TCP y el receptor. Junto con temporizadores, remitentes TCP

y los receptores pueden alterar el comportamiento del flujo de datos. Esto se conoce más generalmente como control de congestión y/o evitación de congestión de red.

Mejorar TCP para manejar pérdidas de manera confiable, minimizar errores, administrar la congestión e ir rápido en entornos de muy alta velocidad son áreas de investigación y desarrollo de estándares en curso. Como resultado, hay una serie de variaciones del algoritmo para evitar la congestión de TCP.

- **Tamaño máximo de segmento**

El tamaño máximo de segmento (MSS) es la mayor cantidad de datos, especificados en bytes, que TCP está dispuesto a recibir en un solo segmento. Para obtener el mejor rendimiento, el MSS debe configurarse lo suficientemente pequeño para evitar la fragmentación de IP, lo que puede provocar la pérdida de paquetes y retransmisiones excesivas. Para intentar lograr esto, normalmente cada lado anuncia el MSS mediante la opción MSS cuando se establece la conexión TCP, en cuyo caso se deriva del tamaño de la unidad de transmisión máxima (MTU) de la capa de enlace de datos de las redes a las que el emisor y el receptor están directamente conectados. Además, los remitentes de TCP pueden usar el descubrimiento de MTU de ruta para inferir la MTU mínima a lo largo de la ruta de red entre el remitente y el receptor, y usar esto para ajustar dinámicamente el MSS para evitar la fragmentación de IP dentro de la red.

El anuncio de MSS también se suele llamar "*negociación MSS*". Estrictamente hablando, el MSS no es "*negociado*" entre el originador y el receptor, porque eso implicaría que tanto el originador como el receptor negociarán y acordarán un MSS único y unificado que se aplica a todas las comunicaciones en ambas direcciones de la conexión. De hecho, se permiten dos valores de MSS completamente independientes para las dos direcciones de flujo de datos en una conexión TCP. Esta situación puede surgir, por ejemplo, si uno de los dispositivos que participan en una conexión tiene una cantidad extremadamente limitada de memoria reservada (quizás incluso más pequeña que la MTU de ruta descubierta general) para el procesamiento segmentos TCP entrantes.

- **Agradecimientos selectivos**

Confiar únicamente en el esquema de reconocimiento acumulativo empleado por el protocolo TCP original puede generar ineficiencias cuando se pierden paquetes. Por ejemplo, suponga que se envían 10 000 bytes en 10 paquetes TCP diferentes y el primer paquete se pierde durante la transmisión. En un protocolo de acuse de recibo acumulativo puro, el receptor no puede decir que recibió correctamente los bytes 1000 a 9999, pero no pudo recibir el primer paquete, que contiene los bytes 0 a 999. Por lo tanto, es posible que el remitente tenga que volver a enviar los 10 000 bytes.

Para resolver este problema TCP emplea el*reconocimiento selectivo (SACK)*opción, que permite al receptor reconocer bloques discontinuos de paquetes que fueron recibidos correctamente, además del número de secuencia del último byte contiguo recibido sucesivamente, como en el reconocimiento TCP básico. El acuse de recibo puede especificar una cantidad de bloques SACK, donde cada bloque SACK es transmitido por los números de secuencia inicial y final de un rango contiguo que el receptor recibió correctamente. En el ejemplo anterior, el receptor enviaría SACK con los números de secuencia 1000 y 9999. Por lo tanto, el remitente retransmite solo el primer paquete, los bytes 0 a 999.

Una extensión de la opción SACK es la opción duplicate-SACK. Una entrega de paquetes desordenada a menudo puede indicar falsamente el remitente TCP del paquete perdido y, a su vez, el remitente TCP retransmite el paquete que se sospecha que se ha perdido y ralentiza la entrega de datos para evitar la congestión de la red. El emisor TCP deshace la acción de desaceleración que es una recuperación del ritmo original de transmisión de datos, al recibir un D-SACK que indica que el paquete retransmitido está duplicado.

La opción SACK es opcional y solo se usa si ambas partes la admiten. Esto se negocia cuando se establece la conexión. SACK usa la parte opcional del encabezado TCP. El uso de SACK está muy extendido: todas las pilas TCP populares lo admiten. El reconocimiento selectivo también se utiliza en *Protocolo de transmisión de control de flujo (SCTP)*.

- **Multiplexación:**Se logra mediante el uso de puertos, al igual que con UDP.

- **Conexiones lógicas:**

Los mecanismos de confiabilidad y control de flujo descritos anteriormente requieren que TCP inicialice y mantenga cierta información de estado para cada flujo de datos. La combinación de este estado, incluidos los sockets, los números de secuencia y los tamaños de ventana, se denomina *conexión lógica*. Cada conexión se identifica de forma única por el par de sockets utilizados por los procesos de envío y recepción.

• **Duplex completo:**TCP proporciona flujos de datos concurrentes en ambas direcciones.

6.4.2. ESTRUCTURA DEL SEGMENTO TCP

El Protocolo de control de transmisión acepta datos de un flujo de datos, los divide en fragmentos y agrega un encabezado TCP creando un segmento TCP. Luego, el segmento TCP se encapsula en un datagrama de Protocolo de Internet (IP). Un segmento TCP es el paquete de información que utiliza TCP para intercambiar datos con sus pares.

El término paquete TCP no está en consonancia con la terminología actual, donde el segmento se refiere a la Unidad de datos del protocolo TCP (PDU), el datagrama a la PDU IP y la trama a la PDU de la capa de enlace de datos.

Los procesos transmiten datos llamando al TCP y pasando búferes de datos como argumentos. El TCP empaqueta los datos de estos búferes en segmentos y llama al módulo de Internet, por ejemplo, IP, para transmitir cada segmento al TCP de destino.

Un segmento TCP consta de *un encabezado de segmento* y *una sección de datos*. El encabezado TCP contiene 10 campos obligatorios y un campo de extensión opcional. La sección de datos sigue al encabezado. Su contenido son los datos de carga útil transportados por la aplicación. La longitud de la sección de datos no se especifica en el encabezado del segmento TCP. Se puede calcular restando la longitud combinada del encabezado TCP y el encabezado IP encapsulado de la longitud total del datagrama IP..

La siguiente figura 3 muestra el diseño de un encabezado de segmento TCP. El encabezado tiene un tamaño de 20 a 60 bytes. Los 20 bytes están reservados para los campos fijos y los otros 40 bytes están reservados para las opciones, que es la información adicional que lleva el encabezado al destino. Estos 40 bytes no son obligatorios.

Analicemos brevemente estos campos de encabezado:

1. **Número de puerto de origen:** Este es el número de puerto de 2 bytes de la aplicación en la computadora de origen que desea enviar el segmento TCP.

2. **Número de puerto de destino:** Este es el número de puerto de 2 bytes de la aplicación en la computadora de destino que se espera que reciba el segmento TCP.

3. **Secuencia de números:** TCP envía múltiples segmentos de origen a destino en una conexión TCP. Por lo tanto, se vuelve importante numerar esos segmentos en una secuencia creciente para mantener la conectividad. Al número de secuencia de 4 bytes se le asigna un número al primer byte de la porción de datos en el segmento TCP. Le aclara al host de destino el primer byte del segmento TCP.

   Durante la fase de conexión TCP, el host de origen y el de destino generan aleatoriamente diferentes números únicos. Supongamos que para el host de origen, este número único aleatorio es 2120 y el primer segmento TCP transporta 2000 bytes de datos. Los números 2120 y 2121 se utilizarán para establecer la conexión TCP con el host de destino. Luego, el siguiente número 2122 se usará en el campo de número de secuencia del primer segmento TCP que transportará 2000 bytes de datos. El segundo segmento llevará el número de secuencia 4122 (= 2122+2000).

4. **Número de acuse de recibo:** Si el host de destino recibe un segmento con el número de secuencia X correctamente, envía el número de acuse de recibo de 4 bytes X+1 de vuelta al host de origen que reconoce la recepción correcta del segmento anterior del origen.

5. **Longitud del encabezado:** Este campo de 4 bits especifica el número de palabras de 4 bytes en el segmento TCP. Dado que el tamaño del encabezado TCP puede ser de 20 a 60 bytes, el valor de este campo puede estar entre 5 y 15.

   Si es 5 entonces la longitud del encabezado TCP es 5 X 4 = 20 bytes, que es el tamaño mínimo que puede tener. De lo contrario, si es 15, el tamaño del encabezado TCP es 15 X 4 = 60, que es el tamaño máximo.

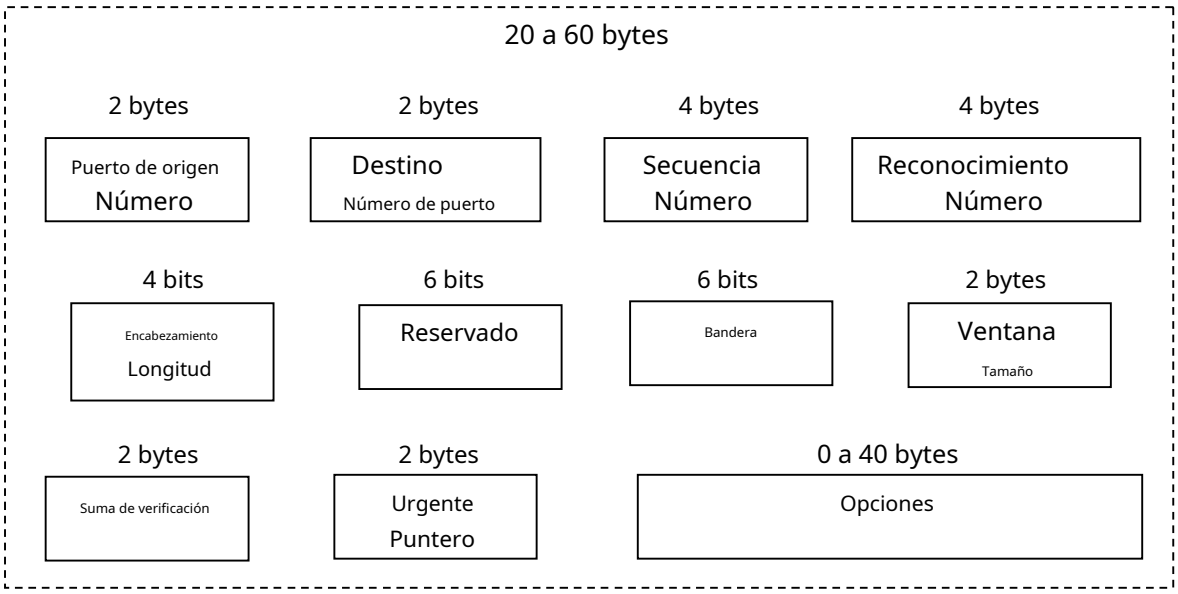


Figura 3: Diseño del encabezado del segmento TCP

6.**Reservado:** Este campo de 6 bits está reservado para uso futuro.

7.**Bandera:** Este campo de 6 bits define 6 banderas de control diferentes donde cada una de ellas ocupa 1 bit. Ellos son

un.**URG:** Establecer en 1 si*puntero urgente*Está en uso. Se utiliza para indicar un desplazamiento de bytes del número de secuencia actual en el que se encuentran los datos urgentes.

B.**RECONOCIMIENTO:** Establézcalo en 1 para indicar que el*Número de reconocimiento*es válida. Si se establece en 0, *Número de acuse de recibo*el campo es ignorado.

C.**PSH:** Se usa para indicar*Datos empujados*.si es s*et*a 1, dio instrucciones al receptor para que entregara los datos a la aplicación correspondiente una vez recibidos. De lo contrario, el receptor espera hasta que el búfer esté lleno.

D.**PRIMERA:** Se utiliza para restablecer una conexión que se ha vuelto confusa por algún motivo. También se utiliza para rechazar un segmento o negarse a abrir una conexión.

mi.**SYN:** Se utiliza para establecer una conexión. Una solicitud de conexión contiene SYN = 1 y ACK = 0. La respuesta a esta solicitud lleva SYN = 1 y ACK = 1.

F.**ALETA:** Se utiliza para cerrar una conexión.

8.**Tamaño de ventana:** Este campo de 2 bytes determina el tamaño de la ventana deslizante que debe mantener el otro extremo.

9.**Suma de comprobación:** Este campo de 2 bytes contiene la suma de comprobación que se utiliza para la detección y corrección de errores.

10**Puntero Urgente:** Este campo de 2 bytes se utiliza para indicar que algunos datos en un segmento TCP son más urgentes que otros en la misma conexión.

6.4.3 FUNCIONAMIENTO DEL PROTOCOLO TCP

Las operaciones del protocolo TCP se pueden dividir en tres fases.

Fase 1:Las conexiones deben establecerse correctamente en un proceso de protocolo de enlace de varios pasos **(establecimiento de conexión)**antes de entrar en la fase 2.

Fase 2:los**transferencia de datos**fase. Una vez completada la transmisión de datos, se ejecuta la fase 3.

Fase 3:los**terminación de conexión**cierra los circuitos virtuales establecidos y libera todos los recursos asignados.

Una conexión TCP es administrada por un sistema operativo a través de una interfaz de programación que representa el punto final local para las comunicaciones, el*toma de internet*. Durante el tiempo de vida de una conexión TCP, el punto final local sufre una serie de cambios de estado:

un. ESCUCHA: (Servidor) representa la espera de una solicitud de conexión desde cualquier puerto y TCP remoto.

B. SYN-ENVIADO: (Cliente) representa la espera de una solicitud de conexión coincidente después de haber enviado una solicitud de conexión.

C. SYN-RECIBIDO: (Servidor) representa la espera de un reconocimiento de solicitud de conexión de confirmación después de haber recibido y enviado una solicitud de conexión.

D. ESTABLECIDO:(Tanto el servidor como el cliente) representan una conexión abierta, los datos recibidos se pueden entregar al usuario. El estado normal para la fase de transferencia de datos de la conexión.

mi. FIN-ESPERA-1:(Tanto servidor como cliente) representa la espera de una solicitud de terminación de conexión del TCP remoto, o un reconocimiento de la solicitud de terminación de conexión enviada previamente.

F. FIN-ESPERA-2:(Tanto el servidor como el cliente) representa la espera de una solicitud de terminación de conexión del TCP remoto.

gramo. CERRAR-ESPERAR:(Tanto el servidor como el cliente) representan la espera de una solicitud de terminación de conexión del usuario local.

H. CLAUSURA:(Tanto el servidor como el cliente) representan la espera de una confirmación de solicitud de terminación de conexión del TCP remoto.

I. ÚLTIMO ACK:(Tanto el servidor como el cliente) representan la espera de un reconocimiento de la solicitud de finalización de la conexión enviada previamente al TCP remoto (que incluye un reconocimiento de su solicitud de finalización de la conexión).

j. TIEMPO DE ESPERA:(Ya sea servidor o cliente) representa esperar a que pase suficiente tiempo para asegurarse de que el TCP remoto recibió el reconocimiento de su solicitud de terminación de conexión.

k. CERRADO:(Tanto el servidor como el cliente) no representan ningún estado de conexión.

6.4.4 Establecimiento de la conexión

Para establecer una conexión, TCP utiliza un*apretón de manos de tres vías*. Antes de que un cliente intente conectarse con un servidor, el servidor primero debe vincularse y escuchar en un puerto para abrirlo a las conexiones. Esto se llama un*pasivo abierto*. Una vez que se establece la apertura pasiva, un cliente puede iniciar una*activo abierto*. Para establecer una conexión, se produce el protocolo de enlace de tres vías (o 3 pasos):

1. **SYN**: La apertura activa la realiza el cliente enviando un SYN al servidor. El cliente establece el número de secuencia del segmento en un valor aleatorio A.
2. **SYN-ACK**: En respuesta, el servidor responde con un SYN-ACK. El número de reconocimiento se establece en uno más que el número de secuencia recibido (A + 1), y el número de secuencia que elige el servidor para el paquete es otro número aleatorio, B.
3. **ACK**: Finalmente, el cliente envía un ACK de vuelta al servidor. El número de secuencia se establece en el valor de acuse de recibo recibido, es decir, A + 1, y el número de acuse de recibo se establece en uno más que el número de secuencia recibido, es decir, B + 1.

En este punto, tanto el cliente como el servidor han recibido un reconocimiento de la conexión. Los pasos 1, 2 establecen el parámetro de conexión (número de secuencia) para una dirección y se reconoce. Los pasos 2, 3 establecen el parámetro de conexión (número de secuencia) para la otra dirección y se reconoce. Con estos se establece una comunicación full-duplex.

6.4.5 TERMINACIÓN DE LA CONEXIÓN

La fase de terminación de la conexión utiliza*un apretón de manos de cuatro vías*, con cada lado de la conexión terminando de forma independiente. Cuando un extremo desea detener su mitad de la conexión, transmite un paquete FIN, que el otro extremo reconoce con un ACK. Por lo tanto, un desmontaje típico requiere un par de segmentos FIN y ACK de cada extremo TCP. Después de que concluyen ambos intercambios FIN/ACK, el lado que envió el primer FIN antes de recibir uno espera un tiempo de espera antes de cerrar finalmente la conexión, tiempo durante el cual el puerto local no está disponible para nuevas conexiones; esto evita la confusión debido a la entrega de paquetes retrasados durante las conexiones posteriores.

Una conexión puede ser "*medio abierto*", en cuyo caso un lado ha terminado su extremo, pero el otro no. El lado que ha terminado ya no puede enviar ningún dato a la conexión, pero el otro lado puede. El lado que termina debe continuar leyendo los datos hasta que el otro lado termina también.

También es posible terminar la conexión mediante un*apretón de manos de 3 vías*, cuando el host A envía un FIN y el host B responde con un FIN y ACK (simplemente combina 2 pasos en uno) y el host A responde con un ACK. Este es quizás el método más común.

Es posible que ambos hosts envíen FIN simultáneamente, luego ambos solo tienen que ACK. Esto posiblemente podría considerarse un apretón de manos de 2 vías ya que la secuencia FIN/ACK se realiza en paralelo para ambas direcciones.

Algunas pilas TCP de host pueden implementar una secuencia de cierre semidúplex, como lo hacen Linux o HP-UX. Si dicho host cierra activamente una conexión pero aún no ha leído todos los datos entrantes que la pila ya recibió del enlace, este host envía un RST en lugar de un FIN. Esto permite que una aplicación TCP se asegure de que la aplicación remota ha leído todos los datos que la primera envió en espera del FIN desde el lado remoto, cuando cierra la conexión de forma activa. Sin embargo, la pila TCP remota no puede distinguir entre un *Conexión abortando RST* y esto *RST de pérdida de datos*. Ambos hacen que la pila remota deseche todos los datos que recibió, pero que la aplicación aún no leyó.

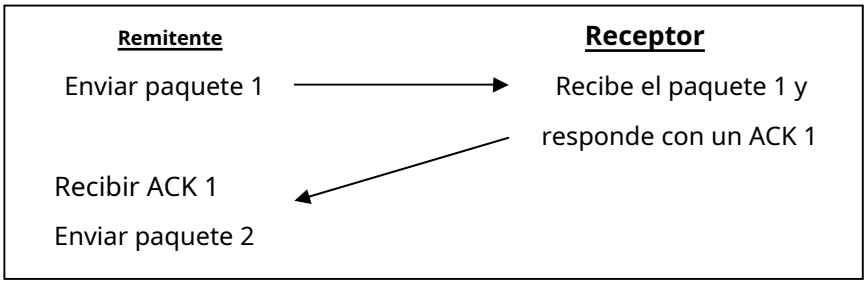
Algunos protocolos de aplicación pueden violar las capas del modelo OSI, utilizando el protocolo de enlace de apertura/cierre de TCP para el protocolo de aplicación de protocolo de enlace de apertura/cierre; estos pueden encontrar el problema RST en el cierre activo.

Para un flujo de programa habitual como el anterior, una pila TCP/IP como la descrita anteriormente no garantiza que todos los datos lleguen a la otra aplicación.

6.4.6 VENTANA CORREDIZA

Se utiliza una ventana deslizante para hacer que la transmisión sea más eficiente, así como para controlar el flujo de datos para que el destino no se vea abrumado por la cantidad de datos.
Las ventanas deslizantes de TCP están orientadas a bytes.

Un protocolo de transporte simple podría usar el siguiente principio: enviar un paquete y luego esperar un reconocimiento del receptor antes de enviar el siguiente paquete. Si no se recibe el ACK dentro de un cierto período de tiempo, retransmita el paquete. Consulte la figura para obtener más detalles.



Si bien este mecanismo garantiza la confiabilidad, solo utiliza una parte del ancho de banda disponible en la red. Ahora, considere un protocolo donde el remitente agrupa sus paquetes para ser transmitidos y usa las siguientes reglas:

• El remitente puede enviar todos los paquetes dentro de la ventana sin recibir un ACK, pero debe iniciar un temporizador de tiempo de espera para cada uno de ellos. • El receptor debe reconocer cada paquete recibido, indicando el número de secuencia del último paquete bien recibido. El remitente desliza la ventana en cada ACK recibido.

Este mecanismo de ventana asegura:

un. Transmisión confiable.

B. Mejor uso del ancho de banda de la red (mejor rendimiento).

C. Control de flujo, ya que el receptor puede retrasar la respuesta a un paquete con un acuse de recibo, sabiendo que sus búferes libres están disponibles y el tamaño de la ventana de la comunicación.

6.4.7 COMPARACIÓN DE UDP Y TCP

El Protocolo de control de transmisión es un protocolo orientado a la conexión, lo que significa que requiere un protocolo de enlace para establecer comunicaciones de extremo a extremo. Una vez que se establece una conexión, los datos del usuario pueden enviarse bidireccionalmente a través de la conexión.

- *De confianza*–TCP gestiona el reconocimiento de mensajes, la retransmisión y el tiempo de espera. Se realizan varios intentos de entregar el mensaje. Si se pierde en el camino, el servidor volverá a solicitar la parte perdida. En TCP, no faltan datos o, en caso de múltiples tiempos de espera, la conexión se interrumpe.

- *Ordenado*–si se envían dos mensajes a través de una conexión en secuencia, el primer mensaje llegará primero a la aplicación receptora. Cuando los segmentos de datos llegan en el orden incorrecto, los búferes TCP retrasan los datos desordenados hasta que todos los datos se pueden reordenar correctamente y entregar a la aplicación.

- *De peso pesado*–TCP requiere tres paquetes para configurar una conexión de socket, antes de que se puedan enviar datos de usuario. TCP maneja la confiabilidad ycontrol de congestión .

- *Transmisión*–Los datos se leen como unbyte corriente, no se transmiten indicaciones distintivas a los límites del mensaje de señal (segmento).

UDP es un protocolo sin conexión basado en mensajes más simple. Los protocolos sin conexión no establecen una conexión de extremo a extremo dedicada. La comunicación se logra mediante la transmisión de información en una dirección desde el origen hasta el destino sin verificar la preparación o el estado del receptor. Sin embargo, un beneficio principal de UDP sobre TCP es la aplicación de voz sobre protocolo de Internet (VoIP), donde la latencia y la inestabilidad son las principales preocupaciones. En VoIP UDP se supone que los usuarios finales proporcionan la confirmación necesaria en tiempo real de que se ha recibido el mensaje.

- *No fidedigno*–Cuando se envía un mensaje, no se puede saber si llegará a su destino; podría perderse en el camino. No existe el concepto de reconocimiento, retransmisión o tiempo de espera.

- *No ordenado*–Si se envían dos mensajes al mismo destinatario, no se puede predecir el orden en que llegarán.

- *Ligero*–No hay ordenación de mensajes, ni seguimiento de conexiones, etc. Es una pequeña capa de transporte diseñada sobre IP.

- *datagramas*–Los paquetes se envían individualmente y se verifica su integridad solo si llegan. Los paquetes tienen límites definidos que se respetan al recibirlos, lo que significa que una operación de lectura en el socket del receptor generará un mensaje completo tal como se envió originalmente.

- *Sin control de congestión*–UDP en sí mismo no evita la congestión, y es posible que las aplicaciones de gran ancho de banda desencadenen el colapso de la congestión, a menos que implementen medidas de control de congestión en el nivel de la aplicación.

6.5 RESUMAMOS

- los transporte capa proporciona Mensaje segmentación, Mensaje acuse de recibo, control de tráfico de mensajes y multiplexación de sesiones.

- Esta capa utiliza dos protocolos para proporcionar servicios. Son el Protocolo de control de transmisión (TCP) y el Protocolo de datagramas de usuario (UDP).

- Esta capa utiliza los conceptos de puertos y sockets para brindar servicios a las aplicaciones que se ejecutan en ambas computadoras host.

- Un puerto es un número de 16 bits. Lo utiliza el protocolo de host a host para identificar a qué protocolo de nivel superior o proceso de aplicación debe entregar los mensajes entrantes.

- TCP es un protocolo orientado a la conexión, ya que TCP siempre crea una conexión virtual entre los dos hosts que intentan comunicarse entre sí mediante el método de negociación manual de tres vías.

- TCP es un protocolo confiable ya que lleva a cabo diferentes mecanismos para detectar errores como duplicación de paquetes, pérdida de paquetes, secuenciación y resecuenciación, etc.

-UDP es un protocolo sin conexión y no confiable.

REVISA TU PROGRESO

1.¿Qué capa del modelo OSI proporciona corrección de errores y control de flujo?

a)Presentación

B)Transporte

C)La red

D)Enlace de datos

2.El direccionamiento especialmente utilizado por la Capa de Transporte es

a)dirección de la estación

B)Dirección de red

C)Dirección del puerto de la aplicación

D)dirección de diálogo

3.¿Tanto TCP como UDP pertenecen a qué capa del modelo OSI?

a) Capa de sesión

b) Capa de transporte

c) Capa de red

d) Capa de enlace de datos

4.¿Qué Unidad de datos de protocolo (PDU) se emplea en la capa de transporte?

un. bits

B. marcos

C. Paquetes

D. Segmentos

5.¿Cuáles son las responsabilidades de la capa de transporte?

6.¿Qué procesos usa TCP, pero no UDP?

un. ventanas

B. Agradecimientos

C. Puerto de origen

D. Puerto de destino

7.¿Qué capa es responsable de proporcionar mecanismos para multiplexar la aplicación de la capa superior, el establecimiento de la sesión y el desmantelamiento de los circuitos virtuales?

un. Sesión

B. La red

C. Físico

D. Transporte

mi. Solicitud

F. Presentación

8.¿Cuáles dos de los siguientes protocolos se utilizan en la capa de transporte?

a) ARP

b) UDP

c) ICMP

d)RARP

e)TCP

f) BotaP

6.6 Respuestas para verificar su progreso

1. segundo
2. do
3. segundo
4. do
5. Escriba sobre confiabilidad, control de flujo, control de congestión, multiplexación, control de duplicación, segmentación de mensajes, control de pérdida y reconocimiento de mensajes.
6. segundo
7. re
8. b & e

6.7 LECTURAS ADICIONALES

1. Tanenbaum AS, Red informática, PHI (EEE)
2. Estancamiento, comunicación de datos e informática, PHI (EEE)
3. Stevens, Programación en red UNIX, PHI (EEE)
4. Forouzan, Comunicación de datos y redes, TMGH

6.8 PREGUNTAS MODELO

1. ¿Cuál es la unidad de datos de la "capa de transporte"? ¿Cuáles son las funciones de una capa de transporte en el modelo OSI?
2. ¿Qué son los puertos y enchufes?
3. Explique la entrega de proceso a proceso.
4. ¿Qué es un datagrama de usuario? Explique la aplicación de UDP y algunos inconvenientes de UDP.
5. ¿Cuál es el número de puerto de origen?
6. ¿Cuál es el número de puerto de destino?
7. ¿Qué es el número de secuencia?
8. ¿Cuál es el número de acuse de recibo?
9. ¿Cuál es la longitud del encabezado?

10. ¿Cuál es el tipo de segmento?

9.¿Qué es el protocolo de control de transmisión?

10¿Qué es la operación TCPProtocol?

11Explicar sobre el establecimiento de la conexión y la terminación de la conexión en términos del protocolo de control de transmisión.

12¿Qué es la ventana corrediza?

13¿Cuál es el tamaño de la ventana?

14¿Qué es la suma de verificación?

15.¿Cuáles son las diferencias entre TCP y UDP?

dieciséis.¿Cuál es la forma de establecer una conexión TCP?

17Marca una de las diferencias más importantes entre TCP y UDP.

18¿Qué capa de OSI es responsable de la comunicación de extremo a extremo?

UNIDAD - 7: CAPA DE APLICACIÓN

ESTRUCTURA DE LA UNIDAD

7.1 OBJETIVOS DE APRENDIZAJE

Después de pasar por esta unidad, podrá:

- conocer los conceptos básicos del modelo de servidor cliente
- obtener una descripción general del sistema de nombres de dominio
- conocer el espacio de nombres DNS
- aprender sobre los servidores de nombres y su uso
- conocer el protocolo SMTP
- saber sobre ftp

7.2 INTRODUCCIÓN

Habiendo terminado todos los preliminares, ahora llegamos a la capa donde se encuentran todas las Aplicaciones. La capa de aplicación es la capa más importante y más visible en las redes informáticas. Las aplicaciones residen en esta capa y los usuarios humanos interactúan a través de esas aplicaciones a través de la red. Las capas por debajo de la capa de aplicación están ahí para proporcionar servicios de transporte, pero no hacen un trabajo real para los usuarios. En este capítulo, estudiaremos algunas aplicaciones de red reales. Sin embargo, incluso en la capa de aplicación existe la necesidad de protocolos de soporte para permitir que las aplicaciones funcionen.

En consecuencia, veremos uno importante de estos antes de comenzar con las aplicaciones en sí. El elemento en cuestión es DNS, que maneja los nombres dentro de Internet. Aprenderemos sobre dos protocolos muy simples, a saber, SMTP y FTP, que son los primeros en aparecer en la capa de aplicación.

Dentro de Internet, el correo electrónico se entrega haciendo que la computadora emisora establezca una conexión TCP con el puerto 25 de la computadora receptora. A la escucha de este puerto hay un servidor de correo que habla SMTP (Protocolo simple de transferencia de correo). Este servidor acepta conexiones entrantes, sujeto a algunos controles de seguridad, y acepta mensajes para su entrega. El protocolo FTP se utiliza para acceder a archivos mediante FTP, el protocolo de transferencia de archivos de Internet. FTP es anterior a la web y ha estado en uso durante más de tres décadas. La web facilita la obtención de archivos colocados en numerosos servidores FTP en todo el mundo al proporcionar una interfaz simple en la que se puede hacer clic en lugar de una interfaz de línea de comandos. Este acceso mejorado a la información es una de las razones del espectacular crecimiento de la web.

7.3 MODELO CLIENTE SERVIDOR

Un modelo Cliente-Servidor es el modelo más antiguo utilizado para organizar una aplicación en red. En este modelo, un servidor brinda servicios a los clientes que intercambian información con él. Este modelo es muy asimétrico: los clientes envían solicitudes y los servidores realizan acciones y devuelven respuestas. Se ilustra en la siguiente figura.

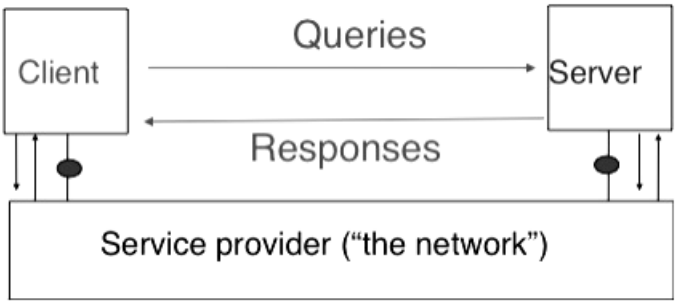


Fig 7.1: El modelo Cliente-Servidor

El modelo cliente-servidor fue el primer modelo que se utilizó para desarrollar aplicaciones en red. Este modelo proviene naturalmente de los mainframes y minicomputadoras que fueron las únicas computadoras en red utilizadas hasta la década de 1980. Una minicomputadora es un sistema multiusuario que es utilizado por decenas o más usuarios al mismo tiempo. Cada usuario interactúa con la minicomputadora usando una terminal. Esos terminales eran principalmente una pantalla, un teclado y un cable conectado directamente a la minicomputadora. Hay varios tipos de servidores, así como varios tipos de clientes. Un servidor web proporciona información en respuesta a la consulta enviada por sus clientes. Un servidor de impresión imprime documentos enviados como consultas por el cliente. Un servidor de correo electrónico reenviará a su destinatario los mensajes de correo electrónico enviados como consultas, mientras que un servidor de música entregará la música solicitada por el cliente.

figura), pero en la práctica estos mensajes se intercambian gracias a las capas subyacentes (las flechas verticales en la figura anterior). Las aplicaciones en red no intercambian mensajes aleatorios. Para garantizar que el servidor pueda comprender las consultas enviadas por un cliente, y también que el cliente pueda comprender las respuestas enviadas por el servidor, ambos deben acordar un conjunto de reglas sintácticas y semánticas. Estas reglas definen el formato de los mensajes intercambiados así como su ordenación. Este conjunto de reglas se denomina protocolo de nivel de aplicación. Un protocolo a nivel de aplicación es similar a una conversación estructurada entre humanos. Suponga que Alicia quiere saber la hora actual pero no tiene reloj. Si Bob pasa cerca, podría tener lugar la siguiente conversación:

- alicia: hola
- boba: hola
- Alicia: ¿Qué hora es?
- Bob: 11:55
- Alicia: gracias
- bob: de nada

Tal conversación tiene éxito si tanto Alice como Bob hablan el mismo idioma. Si Alice se encuentra con Tchang, que solo habla chino, no podrá preguntarle la hora actual. Una conversación entre humanos puede ser más compleja. Por ejemplo, suponga que Bob es un guardia de seguridad cuyo deber es permitir que solo agentes secretos de confianza ingresen a una sala de reuniones. Si todos los agentes conocen una contraseña secreta, la conversación entre Bob y Trudy podría ser la siguiente:

- Bob: ¿Cuál es la contraseña secreta?
- verdad: 1234
- Bob: Esta es la contraseña correcta, de nada.

Si Alicia quiere entrar a la sala de reuniones pero no sabe la contraseña, su conversación podría ser la siguiente:

- Bob: ¿Cuál es la contraseña secreta?
- Alicia: 3.1415
- Bob: Esta no es la contraseña correcta.

Las conversaciones humanas pueden ser muy formales, por ejemplo, cuando los soldados se comunican con su jerarquía, o informales, como cuando los amigos discuten. Las computadoras que se comunican son más parecidas a los soldados y requieren reglas bien definidas para garantizar un intercambio de información exitoso. Hay dos tipos de reglas que definen cómo se puede intercambiar información entre computadoras:

- Reglas sintácticas que definen con precisión el formato de los mensajes que se intercambian. Como las computadoras solo procesan bits, las reglas sintácticas especifican cómo se codifica la información como cadenas de bits.
- Organización del flujo de información. Para muchas aplicaciones, el flujo de información debe estar estructurado y existen relaciones de precedencia entre los diferentes tipos de información. En el ejemplo de la hora anterior, Alice debe saludar a Bob antes de preguntar la hora actual. Alice no preguntaría la hora actual primero y luego saludaría a Bob. Tales relaciones de precedencia existen en aplicaciones en red como

bien. Por ejemplo, un servidor debe recibir un nombre de usuario y una contraseña válida antes de aceptar comandos más complejos de sus clientes.

7.4 DNS (SISTEMA DE NOMBRES DE DOMINIO)

Aunque teóricamente los programas podrían hacer referencia a páginas web, buzones de correo y otros recursos mediante el uso de las direcciones de red (por ejemplo, IP) de las computadoras en las que están almacenados, estas direcciones son difíciles de recordar para las personas. Además, navegar por las páginas web de una empresa desde 128.111.24.41 significa que si la empresa mueve el servidor web a una máquina diferente con una dirección IP diferente, todos deben saber la nueva dirección IP. En consecuencia, se introdujeron nombres legibles de alto nivel para desacoplar los nombres de las máquinas de las direcciones de las máquinas. De esta forma, el servidor Web de la empresa podría ser conocido como*www.cs.washington.edu*independientemente de su dirección IP. Sin embargo, dado que la red en sí comprende solo direcciones numéricas, se requiere algún mecanismo para convertir los nombres en direcciones de red. En las siguientes secciones, estudiaremos cómo se logra este mapeo en Internet. En los días de ARPANET, simplemente había un archivo, hosts.txt, que enumeraba todos los nombres de las computadoras y sus direcciones IP. Todas las noches, todos los anfitriones lo traían del sitio en el que se mantenía. Para una red de unos pocos cientos de grandes máquinas de tiempo compartido, este enfoque funcionó razonablemente bien. Sin embargo, mucho antes de que muchos millones de PC estuvieran conectados a Internet, todos los involucrados se dieron cuenta de que este enfoque no podía seguir funcionando para siempre. Por un lado, el tamaño del archivo sería demasiado grande. Sin embargo, aún más importante, Los conflictos de nombres de host se producirían constantemente a menos que los nombres se gestionaran de forma centralizada, algo impensable en una gran red internacional debido a la carga y la latencia. Para solucionar estos problemas,**DNS**(**sistema de nombres de dominio**) se inventó en 1983. Ha sido una parte clave de Internet desde entonces. La esencia de DNS es la invención de un esquema de nombres jerárquico basado en dominios y un sistema de base de datos distribuida para implementar este esquema de nombres. Se utiliza principalmente para asignar nombres de host a direcciones IP, pero también se puede utilizar para otros fines. El DNS se define en los RFC 1034, 1035, 2181 y se elabora más en muchos otros. Muy brevemente, la forma en que se utiliza el DNS es la siguiente. Para asignar un nombre a una dirección IP, un programa de aplicación llama a un procedimiento de biblioteca llamado resolver y le pasa el nombre como parámetro. El resolutor envía una consulta que contiene el nombre a un servidor DNS local, que busca el nombre y devuelve una respuesta que contiene la dirección IP al resolutor, que luego se la devuelve a la persona que llama. Los mensajes de consulta y respuesta se envían como paquetes UDP.

7.4.1 EL ESPACIO DE NOMBRES DNS

El Sistema de Nombres de Dominio es una base de datos distribuida que permite mapear nombres en direcciones IP. Administrar un conjunto de nombres grande y en constante cambio no es un problema trivial. En el sistema postal, la gestión de nombres se realiza solicitando letras para especificar (implícita o explícitamente) el país, estado o provincia, ciudad, dirección y nombre del destinatario. El uso de este tipo de direccionamiento jerárquico garantiza que no haya confusión entre Marvin Anderson en Main St. en White Plains, NY y Marvin Anderson en Main St. en Austin, Texas. DNS funciona de la misma manera. Para Internet, la parte superior de la jerarquía de nombres es administrada por una organización llamada**ICANN**(**Corporación de Internet para Nombres y Números Asignados**). ICANN se creó con este propósito en 1998, como parte de la maduración de Internet para convertirse en una preocupación económica mundial. Conceptualmente, Internet se divide en más de 250 dominios de nivel superior, donde cada dominio cubre muchos hosts. Cada dominio se divide en subdominios, y estos se dividen aún más, y así sucesivamente. Todos estos dominios se pueden representar mediante un árbol, como se muestra en**Figura 7.2**. Las hojas del árbol representan dominios que no tienen subdominios. Un dominio hoja puede contener un solo host o puede representar una empresa y contener miles de hosts.

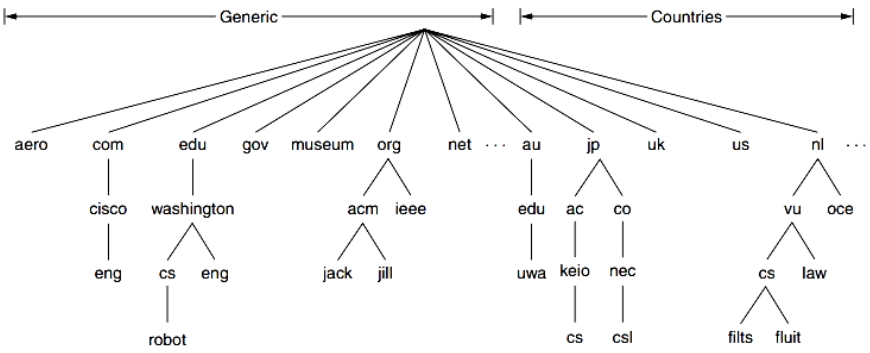


Fig. 7.2: Una parte del espacio de nombres de dominio de Internet

Los dominios de nivel superior vienen en dos sabores: genéricos y países. Los dominios genéricos, enumerados en**Figura 7.3**, incluyen dominios originales de la década de 1980 y dominios introducidos a través de solicitudes a la ICANN. En el futuro se agregarán otros dominios genéricos de nivel superior. Los dominios de países incluyen una entrada para cada país, tal como se define en ISO 3166. Los nombres de dominio de países internacionalizados que usan alfabetos no latinos se introdujeron en 2010. Estos dominios permiten a las personas nombrar hosts en árabe, cirílico, chino u otros idiomas. Obtener un dominio de segundo nivel, como nombre-de-empresa.com, es fácil. Los dominios de nivel superior están a cargo de registradores designados por ICANN. Obtener un nombre simplemente requiere acudir al registrador correspondiente (para com en este caso) para verificar si el nombre deseado está disponible y no es la marca registrada de otra persona. Si no hay problemas, el solicitante paga al registrador una pequeña tarifa anual y obtiene el nombre.

Sin embargo, a medida que Internet se ha vuelto más comercial e internacional, también se ha vuelto más polémico,

especialmente en cuestiones relacionadas con el naming. Esta controversia incluye a la propia ICANN. Por ejemplo, la creación de la*xxx*El dominio tomó varios años y casos judiciales para resolverse. ¿La colocación voluntaria de contenido para adultos en su propio dominio es algo bueno o malo? (Algunas personas no querían que el contenido para adultos estuviera disponible en Internet, mientras que otros querían ponerlo todo en un solo dominio para que los filtros de niñera pudieran encontrarlo fácilmente y bloquearlo para los niños). Algunos de los dominios se autoorganizan, mientras que otros tienen restricciones sobre quién puede obtener un nombre, como se indica en **Figura 7.3**. Pero, ¿qué restricciones son apropiadas? toma el*Pro*dominio,

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

Fig. 7.3: Dominios genéricos de nivel superior

por ejemplo. Es para profesionales cualificados. Pero, ¿quién es un profesional? Los médicos y los abogados son claramente profesionales. Pero, ¿qué pasa con los fotógrafos independientes, los profesores de piano, los magos, los plomeros, los peluqueros, los exterminadores, los tatuadores y los mercenarios? ¿Son elegibles estas ocupaciones? ¿Según quién?

También hay dinero en los nombres. Tuvalu (el país) vendió un contrato de arrendamiento de su dominio de televisión por $ 50 millones, todo porque el código de país se adapta bien a los sitios de publicidad de televisión. Prácticamente todas las palabras comunes (inglés) se han tomado en el*com*dominio, junto con las faltas de ortografía más comunes. Pruebe artículos para el hogar, animales, plantas, partes del cuerpo, etc. La práctica de registrar un dominio solo para dar la vuelta y venderlo a un interesado a un precio mucho más alto incluso tiene un nombre. Se llama ciberocupación. Muchas empresas que fueron lentas cuando llegó la era de Internet

comenzaron a encontrar sus nombres de dominio obvios ya tomados cuando intentaron adquirirlos. En general, siempre que no se violen las marcas registradas y no haya fraude, los nombres se asignan por orden de llegada. Sin embargo, las políticas para resolver disputas de nombres aún se están perfeccionando. Cada dominio se nombra por la ruta ascendente desde él hasta la raíz (sin nombre). Los componentes están separados por puntos. Por lo tanto, el departamento de ingeniería de Cisco podría estar *eng.cisco.com*, en lugar de un nombre de estilo UNIX como */com/cisco/eng*. Tenga en cuenta que esta denominación jerárquica significa que *eng.cisco.com* no entra en conflicto con un uso potencial de *ing* en *eng.washington.edu*, que podría ser utilizado por el departamento de inglés de la Universidad de Washington.

Los nombres de dominio pueden ser absolutos o relativos. Un nombre de dominio absoluto siempre termina con un punto (p. ej., *eng.cisco.com*), mientras que uno relativo no lo hace. Los nombres relativos deben interpretarse en algún contexto para determinar de manera única su verdadero significado. En ambos casos, un dominio con nombre se refiere a un nodo específico en el árbol y todos los nodos debajo de él. Los nombres de dominio no distinguen entre mayúsculas y minúsculas, por lo que *edu*, *educación*, y *educación* significar lo mismo. Los nombres de los componentes pueden tener hasta 63 caracteres y los nombres completos de las rutas no deben exceder los 255 caracteres.

En principio, los dominios se pueden insertar en el árbol en dominios genéricos o de países. Por ejemplo, *cs.washington.edu* igualmente podría figurar en la lista de *nosotros* dominio del país como *cs.washington.wa.us*. En la práctica, sin embargo, la mayoría de las organizaciones en los Estados Unidos están bajo dominios genéricos y la mayoría fuera de los Estados Unidos están bajo el dominio de su país. No existe una regla contra el registro bajo múltiples dominios de nivel superior. Las grandes empresas a menudo lo hacen (p. ej., *sony.com*, *sony.net*, y *sony.nl*). Cada dominio controla cómo asigna los dominios debajo de él. Por ejemplo, Japón tiene dominios *ac.jp* y *co.jp* ese espejo *edu* y *com*. Los Países Bajos no hacen esta distinción y ponen a todas las organizaciones directamente bajo *nl*. Por lo tanto, los tres siguientes son departamentos universitarios de informática:

1. *cs.washington.edu* (Universidad de Washington, en los Estados Unidos).
2. *cs.vu.nl* (Vrije Universiteit, en los Países Bajos).
3. *cs.keio.ac.jp* (Universidad de Keio, en Japón).

Para crear un nuevo dominio se requiere permiso del dominio en el que se incluirá. Por ejemplo, si se inicia un grupo VLSI en la Universidad de Washington y quiere ser conocido como *vlsi.cs.washington.edu*, tiene que obtener el permiso de quien administra *cs.washington.edu*. De manera similar, si se crea una nueva universidad, por ejemplo, la Universidad del Norte de Dakota del Sur, debe pedirle al gerente de la *edu* dominio para asignarlo *unsd.edu* (si todavía está disponible). De esta forma, se evitan conflictos de nombres y cada dominio puede realizar un seguimiento de todos sus subdominios. Una vez que se ha creado y registrado un nuevo dominio, puede crear subdominios, como *cs.unsd.edu*, sin obtener permiso de nadie más alto en el árbol. La denominación sigue los límites de la organización, no las redes físicas. Por ejemplo, si los departamentos de informática e ingeniería eléctrica están ubicados en el mismo edificio y comparten la misma LAN, pueden tener dominios distintos. Del mismo modo, incluso si la ciencia de la computación se divide entre Babbage Hall y

Turing Hall, los anfitriones de ambos edificios normalmente pertenecerán al mismo dominio.

REVISA TU PROGRESO

1. Complete los espacios en blanco:

(a) En el modelo _____, un servidor proporciona servicios a los clientes que intercambian información con él.

(b) El modelo cliente-servidor fue el primer modelo que se utilizó para desarrollar _____.

(c) El DNS se usa principalmente para asignar _____ a _____.

(d) Para Internet, la parte superior de la jerarquía de nombres es administrada por una organización llamada _____.

(e) Un dominio hoja puede contener un host _____, o puede representar una empresa y contener miles de _____.

(f) Los dominios de nivel superior vienen en dos tipos de _____ y _____.

(g) Los nombres de dominio pueden ser _____ o _____.

7.4.2 REGISTROS DE RECURSOS DE DOMINIO

Cada dominio, ya sea un host único o un dominio de nivel superior, puede tener un conjunto de registros de recursos asociados. Estos registros son la base de datos DNS. Para un solo host, el registro de recursos más común es solo su dirección IP, pero también existen muchos otros tipos de registros de recursos. Cuando un resolutor da un nombre de dominio a DNS, lo que obtiene son los registros de recursos asociados con ese nombre. Por lo tanto, la función principal de DNS es asignar nombres de dominio a registros de recursos.

Un registro de recursos es una tupla de cinco. Aunque están codificados en binario para mayor eficiencia, en la mayoría de las exposiciones los registros de recursos se presentan como texto ASCII, una línea por registro de recursos. El formato que usaremos es el siguiente:

Domain_name Time_to_live Clase Tipo Valor

los*Nombre de dominio*indica el dominio al que se aplica este registro. Normalmente, existen muchos registros para cada dominio y cada copia de la base de datos contiene información sobre múltiples dominios. Este campo es, por lo tanto, la clave de búsqueda principal que se utiliza para satisfacer las consultas. El orden de los registros en la base de datos no es significativo. los*Tiempo para vivir* El campo da una indicación de qué tan estable es el registro. A la información que es muy estable se le asigna un valor alto, como 86400 (el

número de segundos en 1 día). A la información que es muy volátil se le asigna un valor pequeño, como 60 (1 minuto).

El tercer campo de cada registro de recurso es el *Clase*. Para La información de Internet, siempre es *EN*. Para información que no sea de Internet, se pueden usar otros códigos, pero en la práctica rara vez se ven.

los *Escribe* El campo indica qué tipo de registro es este. Hay muchos tipos de registros DNS. Los tipos importantes se enumeran en **Figura 7.4**.

Un registro SOA proporciona el nombre de la fuente principal de información sobre la zona del servidor de nombres, la dirección de correo electrónico de su administrador, un número de serie único y varios indicadores y tiempos de espera. El tipo de registro más importante es el registro A (Dirección). Contiene una dirección IPv4 de 32 bits de una interfaz para algún host. El correspondiente *AAAA*, o ''quadA'', el registro contiene una dirección IPv6 de 128 bits. Cada host de Internet debe tener al menos una dirección IP para que otras máquinas puedan comunicarse con él. Algunos hosts tienen dos o más interfaces de red, en cuyo caso tendrán dos o más tipos *A* o *AAAA* registros de recursos. En consecuencia, el DNS puede devolver varias direcciones para un solo nombre.

Un tipo de registro común es el *MX* registro. Especifica el nombre del host preparado para aceptar correo electrónico para el dominio especificado. Se utiliza porque no todas las máquinas están preparadas para aceptar correo electrónico. Si alguien quiere enviar un correo electrónico a, por ejemplo, *cuenta*@*microsoft*.*com*, el servidor de envío necesita encontrar algún servidor de correo ubicado en *microsoft.com* que está dispuesto a aceptar correo electrónico. El registro MX puede proporcionar esta información.

Otro tipo de registro importante es el *NS* registro. Especifica un servidor de nombres para el dominio o subdominio. Este es un host que tiene una copia de la base de datos para un dominio. Se utiliza como parte del proceso de búsqueda de nombres.

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

Fig 7.4: Los principales tipos de registros de recursos DNS

CNOMBRE registros permiten crear alias. Por ejemplo, una persona familiarizada con los nombres de Internet en general y que desea enviar un mensaje al usuario *Pablo* en el departamento de ciencias de la computación en el MIT podría adivinar que *Pablo*@*cs.mit.edu* trabajará. En realidad, esta dirección no funcionará porque el dominio del departamento de informática del MIT es *navegar.mit.edu*. Sin embargo, como un servicio para las personas que no saben esto, el MIT podría crear un ***CNOMBRE*** entrada para señalar a las personas y los programas en la dirección correcta. Una entrada como esta podría hacer el trabajo:

cs.mit.edu 86400 EN CNOMBRE csail.mit.edu

Me gusta ***CNOMBRE***, ***PTR*** apunta a otro nombre. Sin embargo, a diferencia de ***CNOMBRE***, que en realidad es solo una definición de macro (es decir, un mecanismo para reemplazar una cadena por otra), ***PTR*** es un tipo de datos DNS normal cuya interpretación depende del contexto en el que se encuentre. En la práctica, casi siempre se usa para asociar un nombre con un **IP** address para permitir búsquedas de la dirección IP y devolver el nombre de la máquina correspondiente. Estos se llaman **búsquedas inversas**. ***SRV*** es un tipo de registro más nuevo que permite identificar un host para un servicio dado en un dominio. Por ejemplo, el servidor web para *cs.washington.edu* podría ser identificado como *cacatúa.cs.washington.edu*. Este registro generaliza la *MX* registro que realiza la misma tarea pero es solo para servidores de correo. *FPS* también es un tipo de registro más nuevo. Permite que un dominio codifique información sobre qué máquinas del dominio enviarán correo al resto de Internet. Esto ayuda a las máquinas receptoras a comprobar que el correo es válido. Si se recibe correo de una máquina que se llama a sí misma *astuto* pero los registros del dominio dicen que el correo solo será enviado fuera del dominio por una máquina llamada *SMTP*, lo más probable es que el correo sea correo no deseado falsificado. Último en la lista, *TXT* Los registros se proporcionaron originalmente para permitir que los dominios se identificaran de manera arbitraria. Hoy en día, suelen codificar información legible por máquina, típicamente la *FPS* información.

Finalmente, tenemos la *Valor* campo. Este campo puede ser un número, un nombre de dominio o una cadena ASCII. La semántica depende del tipo de registro. Una breve descripción de la *Valor* campos para cada uno de los principales tipos de registro se da en **Figura 7.4**.

Para ver un ejemplo del tipo de información que se puede encontrar en la base de datos DNS de un dominio de **Figura 7.5**. Esta figura representa parte de una base de datos (hipotética) para el *cs.vu.nl* dominio mostrado en **Figura 7.2**. La base de datos contiene siete tipos de registros de recursos.

```
; Authoritative data for cs.vu.nl
cs.vu.nl.       86400   IN   SOA     star boss (9527,
cs.vu.nl.       86400   IN   MX      1 zephyr
cs.vu.nl.       86400   IN   MX      2 top
cs.vu.nl.       86400   IN   NS      star

star            86400   IN   A       130.37.56.205
zephyr          86400   IN   A       130.37.20.10
top             86400   IN   A       130.37.20.11
www             86400   IN   CNAME   star.cs.vu.nl
ftp             86400   IN   CNAME   zephyr.cs.vu.nl

flits           86400   IN   A       130.37.16.112
flits           86400   IN   A       192.31.231.165
flits           86400   IN   MX      1 flits
flits           86400   IN   MX      2 zephyr
flits           86400   IN   MX      3 top

rowboat                 IN   A       130.37.56.201
                        IN   MX      1 rowboat
                        IN   MX      2 zephyr

little-sister           IN   A       130.37.62.23

laserjet                IN   A       192.31.231.216
```

Fig 7.5: Una parte de una posible base de datos DNS para *cs.vu.nl*

La primera línea sin comentarios de **Figura 7.5** da alguna información básica sobre el dominio, que no nos preocupará más. Luego vienen dos entradas que dan el primer y segundo lugar para intentar entregar el correo electrónico enviado a *persona*@*cs.vu.nl*. los *céfiro* (una máquina específica) debe probarse primero. Si eso falla, el *cima* debe probarse como la siguiente opción. La siguiente línea identifica el servidor de nombres para el dominio como *estrella*.

Después de la línea en blanco (agregada para mejorar la legibilidad) vienen líneas que dan las direcciones IP para el *estrella*, *céfiro*, y *cima*. Estos son seguidos por un alias, *www.cs.vu.nl*, para que esta dirección pueda usarse sin designar una máquina específica. Crear este alias permite *cs.vu.nl* para cambiar su servidor World Wide Web sin invalidar la dirección que la gente usa para llegar a él. Un argumento similar vale para *ftp.cs.vu.nl*.

La sección para la máquina *revolotea* enumera dos direcciones IP y se dan tres opciones para manejar el correo electrónico enviado a *flits.cs.vu.nl*. La primera opción es, naturalmente, la *revolotea* sí mismo, pero si está abajo, el *céfiro* y *cima* son la segunda y tercera opciones.

Las siguientes tres líneas contienen una entrada típica para una computadora, en este caso, *bote de remos.cs.vu.nl*. La información proporcionada contiene la dirección IP y las direcciones de correo primaria y secundaria. Luego viene una entrada para una computadora que no es capaz de recibir correo por sí misma, seguida de una entrada que probablemente sea para una impresora que está conectada a Internet.

7.4.3 SERVIDORES DE NOMBRES

Al menos en teoría, un solo servidor de nombres podría contener toda la base de datos DNS y responder a todas las consultas al respecto. En la práctica, este servidor estaría tan sobrecargado que sería inútil. Además, si alguna vez se cayera, todo Internet quedaría paralizado. Para evitar los problemas asociados con tener una única fuente de información, el espacio de nombres DNS se divide en partes que no se superponen. **zonas**. Una forma posible de dividir el espacio de nombres de**Figura 7.2**se muestra en **Figura 7.6**. Cada zona encerrada en un círculo contiene alguna parte del árbol.

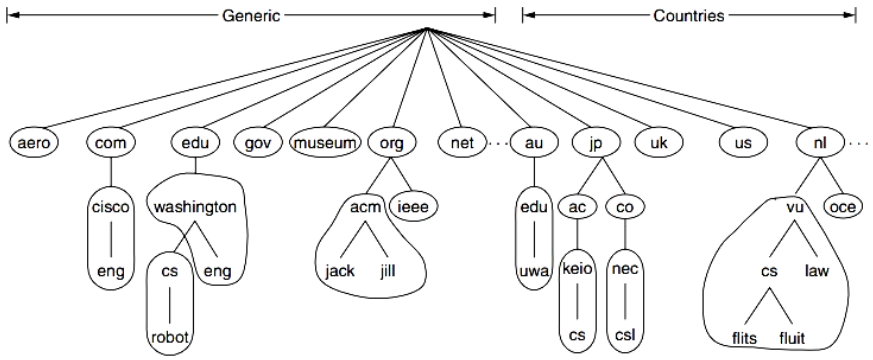


Fig 7.6: Parte del espacio de nombres DNS dividido en (encerrado en un círculo)

zonas

La ubicación de los límites de zona dentro de una zona depende del administrador de esa zona. Esta decisión se toma en gran parte en función de cuántos servidores de nombres se desean y dónde. por ejemplo, en **Figura 7.6**, la Universidad de Washington tiene una zona para *washington.edu*que maneja*eng.washington.edu*pero no maneja*cs.washington.edu*. Esa es una zona separada con sus propios servidores de nombres. Se puede tomar una decisión de este tipo cuando un departamento como Inglés no desea ejecutar su propio servidor de nombres, pero un departamento como Ciencias de la Computación sí lo desea.

Cada zona también está asociada con uno o más servidores de nombres. Estos son hosts que contienen la base de datos de la zona. Normalmente, una zona tendrá un servidor de nombres principal, que obtiene su información de un archivo en su disco, y uno o más servidores de nombres secundarios, que obtienen su información del servidor de nombres principal. Para mejorar la confiabilidad, algunos de los servidores de nombres se pueden ubicar fuera de la zona. El proceso de buscar un nombre y encontrar una dirección se llama**resolución de nombres**. Cuando un resolver tiene una consulta sobre un nombre de dominio, pasa la consulta a un servidor de nombres local. Si el dominio buscado cae bajo la jurisdicción del servidor de nombres, como*top.cs.vu.nl*cayendo bajo*cs.vu.nl*, devuelve los registros de recursos autorizados. Un**registro autorizado**es aquella que procede de la autoridad que gestiona el registro y por tanto es siempre correcta. Los registros autorizados contrastan con**registros en caché**, que puede estar desactualizado. ¿Qué sucede cuando el dominio es remoto, como cuando*flits.cs.vu.nl* quiere encontrar la dirección IP de *robot.cs.washington.edu*en la UW (Universidad de Washington)? En este caso, y si no hay información en caché sobre el dominio disponible localmente, el servidor de nombres inicia una consulta remota. Esta consulta sigue el proceso que se muestra en

Figura 7.7. El paso 1 muestra la consulta que se envía al servidor de nombres local. La consulta contiene el nombre de dominio buscado, el tipo (A) y la clase (IN).

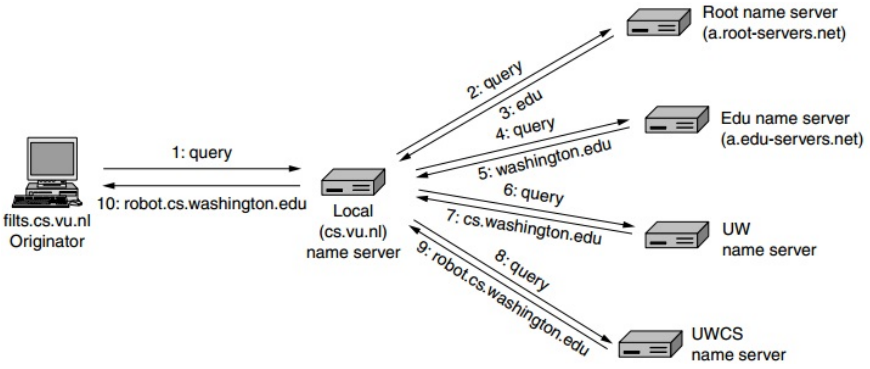


Fig 7.7: Ejemplo de un resolver que busca un nombre remoto en 10 pasos

El siguiente paso es comenzar en la parte superior de la jerarquía de nombres preguntando a uno de los**servidores de nombres raíz**. Estos servidores de nombres tienen información sobre cada dominio de nivel superior. Esto se muestra como el paso 2 en**Figura 7.7**. Para comunicarse con un servidor raíz, cada servidor de nombres debe tener información sobre uno o más servidores de nombres raíz. Esta información normalmente está presente en un archivo de configuración del sistema que se carga en la caché de DNS cuando se inicia el servidor DNS. Es simplemente una lista de registros NS para la raíz y los registros A correspondientes. Hay 13 servidores DNS raíz, llamados sin imaginación*a-root-servers.net* a través de*m.root-servers.net*. Cada servidor raíz podría ser lógicamente una sola computadora. Sin embargo, dado que todo Internet depende de los servidores raíz, estos son equipos potentes y muy replicados. La mayoría de los servidores están presentes en varias ubicaciones geográficas y se accede a ellos mediante cualquier enrutamiento de transmisión, en el que un paquete se entrega a la instancia más cercana de una dirección de destino.

Es poco probable que el servidor de nombres raíz sepa la dirección de una máquina en UW, y probablemente tampoco conozca el servidor de nombres para UW. Pero debe conocer el servidor de nombres para el*edu* dominio, en el que*cs.washington.edu*se encuentra. Devuelve el nombre y la dirección IP para esa parte de la respuesta en el paso 3. El servidor de nombres local luego continúa su búsqueda. Envía la consulta completa al*edu* nombre del servidor (*a.edu-servers.net*). Ese servidor de nombres devuelve el servidor de nombres para UW. Esto se muestra en los pasos 4 y 5. Más cerca ahora, el servidor de nombres local envía la consulta al servidor de nombres UW (paso 6). Si el nombre de dominio buscado estuviera en el departamento de inglés, se encontraría la respuesta, ya que la zona UW incluye el departamento de inglés. Pero el departamento de Ciencias de la Computación ha optado por ejecutar su propio servidor de nombres. La consulta devuelve el nombre y la dirección IP del servidor de nombres de UW Computer Science (paso 7).

Finalmente, el servidor de nombres local consulta el servidor de nombres de UW Computer Science (paso 8). Este servidor tiene autoridad para el dominio.*cs.washington.edu*, por lo que debe tener la respuesta. Devuelve la respuesta final (paso 9), que el servidor de nombres local reenvía como respuesta a flits.cs.vu.nl (paso 10). El nombre ha sido resuelto.

Puede explorar este proceso utilizando herramientas estándar como el*cavar* programa que está instalado en la mayoría de los sistemas UNIX. Por ejemplo, escribir

dig@a.edu-servers.net robot.cs.washington.edu

enviará una consulta para*robot.cs.washington.edu*al*a.eduservers.net* servidor de nombres e imprima el resultado. Esto le mostrará la información obtenida en el paso 4 en el ejemplo anterior, y aprenderá el nombre y la dirección IP de los servidores de nombres UW.

Hay tres puntos técnicos para discutir sobre este largo escenario. En primer lugar, dos mecanismos de consulta diferentes están en funcionamiento en **Figura 7.7**. cuando el anfitrión*flits.cs.vu.nl*envía su consulta al servidor de nombres local, ese servidor de nombres maneja la resolución en nombre de *revolotea* hasta que tenga la respuesta deseada para devolver. No devuelve respuestas parciales. Pueden ser útiles, pero no son lo que buscaba la consulta. Este mecanismo se denomina**consulta recursiva**.

Por otro lado, el servidor de nombres raíz (y cada servidor de nombres subsiguiente) no continúa recursivamente con la consulta del servidor de nombres local. Simplemente devuelve una respuesta parcial y pasa a la siguiente consulta. El servidor de nombres local es responsable de continuar con la resolución emitiendo más consultas. Este mecanismo se denomina**consulta iterativa**. Una resolución de nombre puede involucrar ambos mecanismos, como mostró este ejemplo. Una consulta recursiva siempre puede parecer preferible, pero muchos servidores de nombres (especialmente el raíz) no las manejarán. Están demasiado ocupados. Las consultas iterativas ponen la carga sobre el creador. La razón por la que el servidor de nombres local admite una consulta recursiva es que proporciona un servicio a los hosts de su dominio. Esos hosts no tienen que estar configurados para ejecutar un servidor de nombres completo, solo para llegar al local.

El segundo punto es el almacenamiento en caché. Todas las respuestas, incluidas todas las respuestas parciales devueltas, se almacenan en caché. De esta forma, si otro *cs.vu.nl*consultas de host para*robot.cs.washington.edu*la respuesta ya se sabrá. Aún mejor, si un host consulta por un host diferente en el mismo dominio, digamos*galah.cs.washington.edu*, la consulta se puede enviar directamente al servidor de nombres autorizado. Del mismo modo, las consultas de otros dominios en*washington.edu*puede empezar directamente desde el *washington.edu*nombre del servidor. El uso de respuestas en caché reduce en gran medida los pasos de una consulta y mejora el rendimiento. El escenario original que esbozamos es, de hecho, el peor caso que ocurre cuando no se almacena en caché información útil. Sin embargo, las respuestas almacenadas en caché no tienen autoridad, ya que los cambios realizados en*cs.washington.edu* no se propagará a todos los cachés del mundo que puedan saberlo. Por esta razón, las entradas de caché no deberían durar demasiado. Esta es la razón por la que el *Tiempo para vivir*El campo se incluye en cada registro de recurso. Le dice a los servidores de nombres remotos cuánto tiempo almacenar en caché los registros. Si una determinada máquina ha tenido la misma dirección IP durante años, puede ser seguro almacenar esa información en caché durante 1 día. Para obtener información más volátil, podría ser más seguro purgar los registros después de unos segundos o un minuto.

El tercer problema es el protocolo de transporte que se utiliza para las consultas y respuestas. es UDP. Los mensajes DNS se envían en paquetes UDP con un formato simple para consultas, respuestas y servidores de nombres que se pueden usar para continuar con la resolución. No entraremos en los detalles de este formato. Si no llega ninguna respuesta en un breve

tiempo, el cliente DNS repite la consulta, probando otro servidor para el dominio después de una pequeña cantidad de reintentos. Este proceso está diseñado para manejar el caso de que el servidor esté inactivo, así como la pérdida del paquete de consulta o respuesta. Se incluye un identificador de 16 bits en cada consulta y se copia en la respuesta para que un servidor de nombres pueda hacer coincidir las respuestas con la consulta correspondiente, incluso si hay varias consultas pendientes al mismo tiempo. Aunque su propósito es simple, debe quedar claro que el DNS es un sistema distribuido grande y complejo que se compone de millones de servidores de nombres que funcionan juntos. Forma un vínculo clave entre los nombres de dominio legibles por humanos y las direcciones IP de las máquinas. Incluye replicación y almacenamiento en caché para el rendimiento y la confiabilidad y está diseñado para ser muy robusto.

También hay demanda de aplicaciones para usar nombres de formas más flexibles, por ejemplo, nombrando contenido y resolviendo la dirección IP de un host cercano que tiene el contenido. Esto se ajusta al modelo de búsqueda y descarga de una película. Lo que importa es la película, no la computadora que tiene una copia de ella, por lo que todo lo que se necesita es la dirección IP de cualquier computadora cercana que tenga una copia de la película. Las redes de distribución de contenido son una forma de lograr este mapeo.

7.5 SMTP: PROTOCOLO DE TRANSFERENCIA DE CORREO SIMPLE

El Protocolo simple de transferencia de correo (SMTP) es un estándar de Internet para la transmisión de correo electrónico (e-mail) a través de redes de Protocolo de Internet (IP). SMTP se definió por primera vez en RFC 821 (1982, eventualmente declarado STD 10) y se actualizó por última vez en RFC 5321 (2008), que incluye las adiciones de SMTP extendido (ESMTP), y es el protocolo de uso generalizado en la actualidad. SMTP usa el puerto TCP 25. El protocolo para nuevos envíos (MSA) es efectivamente el mismo que SMTP, pero usa el puerto 587 en su lugar. Las conexiones SMTP protegidas por SSL se conocen con la abreviatura SMTPS, aunque SMTPS no es un protocolo por derecho propio. Mientras que los servidores de correo electrónico y otros agentes de transferencia de correo utilizan SMTP para enviar y recibir mensajes de correo, las aplicaciones de correo de cliente a nivel de usuario suelen utilizar SMTP solo para enviar mensajes a un servidor de correo para su retransmisión. Para recibir mensajes, las aplicaciones cliente suelen utilizar el Protocolo de oficina de correos (POP) o el Protocolo de acceso a mensajes de Internet (IMAP) o un sistema propietario (como Microsoft Exchange o Lotus Notes/Domino) para acceder a sus cuentas de casilla de correo en un servidor de correo. En la década de 1960 se utilizaron varias formas de mensajería electrónica uno a uno. Las personas se comunicaban entre sí utilizando sistemas desarrollados para computadoras centrales específicas. A medida que se interconectaron más computadoras, especialmente en ARPANET del gobierno de los EE. UU., se desarrollaron estándares para permitir que los usuarios de diferentes sistemas se enviaran correos electrónicos entre sí. SMTP surgió de estos estándares desarrollados durante la década de 1970. SMTP puede rastrear sus raíces en dos implementaciones descritas en 1971: el Protocolo de buzón de correo, cuya implementación ha sido cuestionada, pero se analiza en RFC 196 y otras RFC, y el programa SNDMSG, que,

ARPANET. Menos de 50 hosts estaban conectados a ARPANET en este momento. Otras implementaciones incluyen correo FTP y protocolo de correo, ambos de 1973. El trabajo de desarrollo continuó a lo largo de la década de 1970, hasta que ARPANET se convirtió en la Internet moderna alrededor de 1980. Jon Postel luego propuso un protocolo de transferencia de correo en 1980 que comenzó a eliminar la dependencia del correo en FTP. . SMTP se publicó como RFC 788 en noviembre de 1981, también por Postel. El estándar SMTP se desarrolló casi al mismo tiempo que Usenet, una red de comunicación de uno a muchos con algunas similitudes. SMTP se volvió ampliamente utilizado a principios de la década de 1980. En ese momento, era un complemento del correo Unix to Unix Copy Program (UUCP), que era más adecuado para manejar transferencias de correo electrónico entre máquinas que estaban conectadas de forma intermitente. SMTP, por otro lado, funciona mejor cuando las máquinas de envío y recepción están conectadas a la red todo el tiempo. Ambos utilizan un mecanismo de almacenamiento y reenvío y son ejemplos de tecnología push. Aunque los grupos de noticias de Usenet todavía se propagan con UUCP entre servidores, el correo UUCP prácticamente ha desaparecido junto con las "rutas explosivas" que usaba como encabezados de enrutamiento de mensajes.

Lanzado con 4.1cBSD, justo después de RFC 788, Sendmail fue uno de los primeros (si no el primero) agentes de transferencia de correo en implementar SMTP. Con el tiempo, a medida que BSD Unix se convirtió en el sistema operativo más popular en Internet, sendmail se convirtió en el MTA (agente de transferencia de correo) más común. Algunos otros programas de servidor SMTP populares incluyen Postfix, qmail, Novell GroupWise, Exim, Novell NetMail, Microsoft Exchange Server, Sun Java System Messaging Server. El envío de mensajes (RFC 2476) y SMTP-AUTH (RFC 2554) se introdujeron en 1998 y 1999 y ambos describen nuevas tendencias en la entrega de correo electrónico. Originalmente, los servidores SMTP solían ser internos de una organización, recibían correo para la organización desde el exterior y retransmitían mensajes de la organización al exterior. Pero con el paso del tiempo, los servidores SMTP (agentes de transferencia de correo), en la práctica, estaban ampliando sus funciones para convertirse en agentes de envío de mensajes para los agentes de usuarios de correo, algunos de los cuales ahora retransmitían correo desde el exterior de una organización. Este problema, consecuencia de la rápida expansión y popularidad de la World Wide Web, hizo que SMTP tuviera que incluir reglas y métodos específicos para retransmitir correo y autenticar a los usuarios para evitar abusos como la retransmisión de correo electrónico no solicitado (spam). El trabajo sobre el envío de mensajes (RFC 2476) se inició originalmente porque los servidores de correo populares a menudo reescribían el correo en un intento de solucionar los problemas, por ejemplo, agregando un nombre de dominio a una dirección no calificada. Este comportamiento es útil cuando el mensaje que se está reparando es un envío inicial, pero peligroso y dañino cuando el mensaje se originó en otro lugar y se está retransmitiendo. La separación limpia del correo en envío y retransmisión se consideró una forma de permitir y fomentar la reescritura de envíos y prohibir la reescritura de retransmisión. A medida que el spam se volvió más frecuente, también se lo vio como una forma de proporcionar autorización para el envío de correo desde una organización, así como la trazabilidad. Esta separación entre retransmisión y envío se convirtió rápidamente en la base de las prácticas modernas de seguridad del correo electrónico.

Como este protocolo comenzó puramente basado en texto ASCII, no funcionó bien con archivos binarios o caracteres en muchos idiomas distintos del inglés. Estándares tales como**Internet multipropósito**

Extensiones de correo(MIME) se desarrollaron para codificar archivos binarios para su transferencia a través de SMTP. Los agentes de transferencia de correo (MTA) desarrollados después de Sendmail también tendían a implementarse con limpieza de 8 bits, de modo que la estrategia alternativa "solo enviar ocho" podría usarse para transmitir datos de texto arbitrarios (en cualquier codificación de caracteres similar a ASCII de 8 bits) a través de SMTP. Mojibake seguía siendo un problema debido a las diferentes asignaciones de conjuntos de caracteres entre los proveedores, aunque las direcciones de correo electrónico solo permitían ASCII. Actualmente, los MTA limpios de 8 bits tienden a admitir la extensión 8 BITMIME, lo que permite que los archivos binarios se transmitan casi tan fácilmente como el texto sin formato. Recientemente, se creó la extensión SMTPUTF8 para admitir texto UTF-8, lo que permite contenido y direcciones internacionales en alfabetos no latinos, como cirílico o chino.

Recuperación de correo SMTP V/S

SMTP es solo un protocolo de entrega. En uso normal, el correo es "empujado" a un servidor de correo de destino (o servidor de correo de siguiente salto) a medida que llega. El correo se enruta en función del servidor de destino, no de los usuarios individuales a los que se dirige. Otros protocolos, como el Protocolo de oficina postal (POP) y el Protocolo de acceso a mensajes de Internet (IMAP), están diseñados específicamente para que los utilicen usuarios individuales que recuperan mensajes y administran buzones de correo. Para permitir que un servidor de correo conectado de forma intermitente extraiga mensajes de un servidor remoto a pedido, SMTP tiene una función para iniciar el procesamiento de la cola de correo en un servidor remoto. POP e IMAP son protocolos inadecuados para retransmitir correo a través de máquinas conectadas de forma intermitente; están diseñados para operar después de la entrega final, cuando la información crítica para el correcto funcionamiento de la retransmisión de correo (el "

Ejemplo de transporte SMTP

Un ejemplo típico de envío de un mensaje vía SMTP a dos buzones (alice y theboss) ubicados en el mismo dominio de correo (example.com o localhost.com) se reproduce en el siguiente intercambio de sesiones. (En este ejemplo, las partes de la conversación tienen el prefijo S: y C:, para el servidor y el cliente, respectivamente; estas etiquetas no forman parte del intercambio). Después de que el remitente del mensaje (cliente SMTP) establezca un canal de comunicación confiable para el mensaje receptor (servidor SMTP), el servidor abre la sesión con un saludo, que generalmente contiene su nombre de dominio completo (FQDN), en este caso*smtp.ejemplo.com*. El cliente inicia su diálogo respondiendo con un comando HELO identificándose en el parámetro del comando con su FQDN (o una dirección literal si no hay ninguna disponible).

S: 220 smtp.example.com ESMTP Sufijo HELO
C: relay.example.org
S: 250 Hola relay.example.org, me alegro de conocerte CORREO DE:<
C: bob@example.org >
S: 250 bien
C: RCPT A:< alice@example.com > 250
S: Ok
C: RCPT A:< theboss@example.com > 250
S: Ok
C: DATOS
S: 354 Finalizar datos con <CR><LF>.<CR><LF>

C: De: "Ejemplo de Bob" < bob@example.org > Para:
C: "Ejemplo de Alice" < alice@example.com > Cc:
C: theboss@example.com
C: Fecha: martes, 15 de enero de 2008 16:02:43 -0500
C: Asunto: Mensaje de prueba
C:
C: Hola Alicia.
C: Este es un mensaje de prueba con 5 campos de encabezado y 4 líneas en el cuerpo del mensaje.
C: Tu amigo,
C: Beto
C: .
S: 250 Ok: en cola como 12345
C: SALIR
S: 221 Adiós
{El servidor cierra la conexión}

El cliente notifica al receptor la dirección de correo electrónico de origen del mensaje en un**CORREO DE**mando. En este ejemplo, el mensaje de correo electrónico se envía a dos buzones en el mismo servidor SMTP: uno para cada destinatario enumerado en el**Para**y**CC**campos de encabezado. El comando SMTP correspondiente es**RCPT A**. Cada recepción y ejecución exitosa de un comando es reconocida por el servidor con un código de resultado y un mensaje de respuesta (por ejemplo, 250 Ok). La transmisión del cuerpo del mensaje de correo se inicia con un comando DATA, después de lo cual se transmite palabra por línea y finaliza con una secuencia de fin de datos. Esta secuencia consta de una nueva línea (<CR><LF>), un único punto final (punto), seguido de otra nueva línea. Dado que el cuerpo de un mensaje puede contener una línea con solo un punto como parte del texto, el cliente envía dos puntos cada vez que una línea comienza con un punto; correspondientemente, el servidor reemplaza cada secuencia de dos puntos al comienzo de una línea con uno solo. Tal método de escape se llama **relleno de puntos**. La respuesta positiva del servidor al final de los datos, como se ejemplifica, implica que el servidor ha asumido la responsabilidad de entregar el mensaje. Un mensaje se puede duplicar si hay una falla de comunicación en este momento, por ejemplo, debido a un corte de energía: hasta que el remitente haya recibido esa respuesta, debe asumir que el mensaje no se entregó. Por otro lado, después de que el receptor ha decidido aceptar el mensaje, debe asumir que el mensaje le ha sido entregado. Así, durante este lapso de tiempo, ambos agentes tienen copias activas del mensaje que intentarán entregar. La probabilidad de que se produzca un error de comunicación exactamente en este paso es directamente proporcional a la cantidad de filtrado que realiza el servidor en el cuerpo del mensaje, con mayor frecuencia con fines antispam. El límite de tiempo de espera se especifica en 10 minutos. El comando QUIT finaliza la sesión. Si el correo electrónico tiene otros destinatarios ubicados en otro lugar, el cliente SALDRÁ y se conectará a un servidor SMTP apropiado para los destinatarios posteriores después de que los destinos actuales se hayan puesto en cola. La información que el cliente envía en los comandos HELO y MAIL FROM se agrega como campos de encabezado adicionales al mensaje por parte del servidor receptor. Agrega un campo de encabezado Received y Return-Path, respectivamente. Algunos clientes están implementados para cerrar la conexión después de que se acepta el mensaje (250 Ok: en cola como 12345), por lo que las dos últimas líneas pueden omitirse. Esto provoca un error en el servidor al intentar enviar la respuesta 221. el cliente saldría y se conectaría a un servidor SMTP apropiado para los destinatarios posteriores después de que los destinos actuales se hayan puesto en cola. La información que el cliente envía en los comandos HELO y MAIL FROM se agrega como campos de encabezado adicionales al mensaje por parte del servidor receptor. Agrega un campo de encabezado Received y Return-Path, respectivamente. Algunos clientes están implementados para cerrar la conexión después de que se acepta el mensaje (250 Ok: en cola como 12345), por lo que las dos últimas líneas pueden omitirse. Esto provoca un error en el servidor al intentar enviar la respuesta 221. el cliente saldría y se conectaría a un servidor SMTP apropiado para los destinatarios posteriores después de que los destinos actuales se hayan puesto en cola. La información que el cliente envía en los comandos HELO y MAIL FROM se agrega como campos de encabezado adicionales al mensaje por parte del servidor receptor. Agrega un campo de encabezado Received y Return-Path, respectivamente. Algunos clientes están implementados para cerrar la conexión después de que se acepta el mensaje (250 Ok: en cola como 12345), por lo que las dos últimas líneas pueden omitirse. Esto provoca un error en el servidor al intentar enviar la respuesta 221. Algunos clientes están implementados para cerrar la conexión después de que se acepta el mensaje (250 Ok: en cola como 12345), por lo que las dos últimas líneas pueden omitirse. Esto provoca un error en el servidor al intentar enviar la respuesta 221. Algunos clientes están implementados para cerrar la conexión después de que se acepta el mensaje (250 Ok: en cola como 12345), por lo que las dos últimas líneas pueden omitirse. Esto provoca un error en el servidor al intentar enviar la respuesta 221.

7.6 PROTOCOLO DE TRANSFERENCIA DE ARCHIVOS FTP

El Protocolo de transferencia de archivos (FTP) es un protocolo de red estándar que se utiliza para transferir archivos de un host a otro a través de una red basada en TCP, como Internet. FTP se basa en una arquitectura cliente-servidor y utiliza conexiones de datos y control separadas entre el cliente y el servidor. Los usuarios de FTP pueden autenticarse mediante un protocolo de inicio de sesión de texto no cifrado, normalmente en forma de nombre de usuario y contraseña, pero pueden conectarse de forma anónima si el servidor está configurado para permitirlo. Para una transmisión segura que oculta (encripta) el nombre de usuario y la contraseña, y encripta el contenido, el FTP suele protegerse con SSL/TLS ("FTPS"). El protocolo de transferencia de archivos SSH ("SFTP") a veces también se usa en su lugar, pero es tecnológicamente diferente. Las primeras aplicaciones de cliente FTP fueron aplicaciones de línea de comandos desarrolladas antes de que los sistemas operativos tuvieran interfaces gráficas de usuario y aún se envían con la mayoría de los sistemas operativos Windows, Unix y Linux. Desde entonces, se han desarrollado decenas de clientes FTP y utilidades de automatización para equipos de escritorio, servidores, dispositivos móviles y hardware, y FTP se ha incorporado a cientos de aplicaciones de productividad, como editores de páginas web.

Comunicación y transferencia de datos

FTP puede ejecutarse en modo activo o pasivo, lo que determina cómo se establece la conexión de datos. En modo activo, el cliente crea una conexión de control TCP. En situaciones en las que el cliente está detrás de un cortafuegos y no puede aceptar conexiones TCP entrantes, se puede utilizar el modo pasivo. En este modo, el cliente usa la conexión de control para enviar un comando PASV al servidor y luego recibe una dirección IP del servidor y un número de puerto del servidor del servidor, que luego el cliente usa para abrir una conexión de datos desde un puerto de cliente arbitrario al servidor. Dirección IP del servidor y número de puerto del servidor recibidos. Ambos modos se actualizaron en septiembre de 1998 para admitir IPv6. En ese momento, se introdujeron más cambios en el modo pasivo, actualizándolo al modo pasivo extendido. El servidor responde a través de la conexión de control con códigos de estado de tres dígitos en ASCII con un mensaje de texto opcional. Por ejemplo, "200" (o "200 OK") significa que el último comando fue exitoso. Los números representan el código de la respuesta y el texto opcional representa una explicación o solicitud legible por humanos (p. ej., <necesita cuenta para almacenar archivo>). Una transferencia en curso de datos de archivo a través de la conexión de datos se puede cancelar mediante un mensaje de interrupción enviado a través de la conexión de control.

Al transferir datos a través de la red, se pueden usar cuatro representaciones de datos:

- **Modo ASCII**: utilizado para el texto. Los datos se convierten, si es necesario, de la representación de caracteres del host emisor a "ASCII de 8 bits" antes de la transmisión y (nuevamente, si es necesario) a la representación de caracteres del host receptor. Como consecuencia, este modo es inapropiado para archivos que contienen datos que no sean texto sin formato.
- **Modo de imagen**(comúnmente llamado modo binario): la máquina emisora envía cada archivo byte por byte, y el destinatario

almacena el flujo de bytes tal como lo recibe. (Se ha recomendado la compatibilidad con el modo de imagen para todas las implementaciones de
- FTP). **Modo EBCDIC**: se utiliza para texto sin formato entre hosts mediante el juego de caracteres EBCDIC. Este modo es por lo demás como el modo ASCII.

- **Modo local**: Permite que dos computadoras con configuraciones idénticas envíen datos en un formato propietario sin necesidad de convertirlo a ASCII

Para archivos de texto, se proporcionan diferentes opciones de control de formato y estructura de registro. Estas características fueron diseñadas para facilitar archivos que contienen Telnet o ASA.

La transferencia de datos se puede realizar en cualquiera de los tres modos:

- **Modo de transmisión**: Los datos se envían como un flujo continuo, lo que evita que FTP realice ningún procesamiento. Más bien, todo el procesamiento se deja en manos de TCP. No se necesita un indicador de fin de archivo, a menos que los datos se dividan en registros.
- **Modo bloque**: FTP divide los datos en varios bloques (encabezado de bloque, recuento de bytes y campo de datos) y luego los pasa a TCP.

- **modo comprimido**: Los datos se comprimen utilizando un único algoritmo.

REVISA TU PROGRESO

2. Complete los espacios en blanco:

(a) La función principal de DNS es asignar _____ a _____.

(b) El proceso de buscar un nombre y encontrar una dirección se llama _____.

(c) Los mensajes DNS se envían en paquetes _____ con un formato simple para consultas.

(d) Las conexiones SMTP aseguradas por SSL se conocen por la abreviatura _____.

(e) _____ tiene una función para iniciar el procesamiento de la cola de correo en un servidor remoto

(f) _____ se basa en una arquitectura cliente-servidor y usa _____ y _____ separados entre el cliente y el servidor.

(g) _____ usuarios pueden autenticarse mediante un inicio de sesión de texto claro _____, normalmente en forma de nombre de usuario y contraseña.

4.9 RESUMAMOS

- los ***Capa de aplicación*** Es la capa más importante y más visible en las redes informáticas.

- A ***Modelo cliente-servidor*** es el modelo más antiguo utilizado para organizar una aplicación en red.

- los ***sistema de nombres de dominio*** es una base de datos distribuida que permite mapear nombres en direcciones IP.

- La esencia de ***DNS*** es la invención de un esquema de nombres jerárquico basado en dominios y un sistema de base de datos distribuida para implementar este esquema de nombres.

- Cada dominio, ya sea un host único o un dominio de nivel superior, puede tener un conjunto de ***registros de recursos*** asociado a ello.

- los ***DNS*** el espacio de nombres se divide en zonas que no se superponen.

- ***Protocolo simple de transferencia de correo*** es un estándar de Internet para la transmisión de correo electrónico a través de redes de protocolo de Internet.

- ***Protocolo de transferencia de archivos*** es un protocolo de red estándar utilizado para transferir archivos de un host a otro host a través de una red basada en TCP.

- ***Extensiones de correo de Internet multipropósito*** (MIME) se desarrollaron para codificar archivos binarios para su transferencia a través de SMTP.

- ***FTP*** puede ejecutarse en modo activo o pasivo, lo que determina cómo se establece la conexión de datos.

4.10 RESPUESTAS PARA COMPROBAR TU PROGRESO

1.
(a) Cliente Servidor.
(b) aplicaciones en red.
(c) nombres de host, direcciones IP.
(d) Corporación de Internet para la Asignación de Nombres y Números.
(e) solteros, anfitriones.
f) genéricos, países.
(g) absoluto, relativo.

2.
(a) nombres de dominio, registros de recursos
(b) resolución de nombres.
(c) PDU.
(d) SMTPS.
(e) SMTP.
(f) FTP, control, conexiones de datos.
(g) FTP, protocolo.

4.11 LECTURAS ADICIONALES

Red de computadoras

- Andrew S. Tanenbaum, David J. Wetherall
  PRENTICE HALL

Redes informáticas: principios, protocolos y práctica

-Olivier Buenaventura

4.11 PREGUNTAS MODELO

1. Explicar el concepto de un modelo Cliente Servidor.

2. ¿Qué es el Sistema de Nombres de Dominio? Explique su significado.

3. Explique el espacio de nombres DNS con un diagrama.

4. ¿Qué son los registros de recursos de dominio? Explique su utilidad.

5. Describir el concepto de servidores de nombres.

6. Explain the concept of the Simple Mail Transfer Protocol giving suitable examples.

7. What is the File Transfer Protocol? Describe its application.