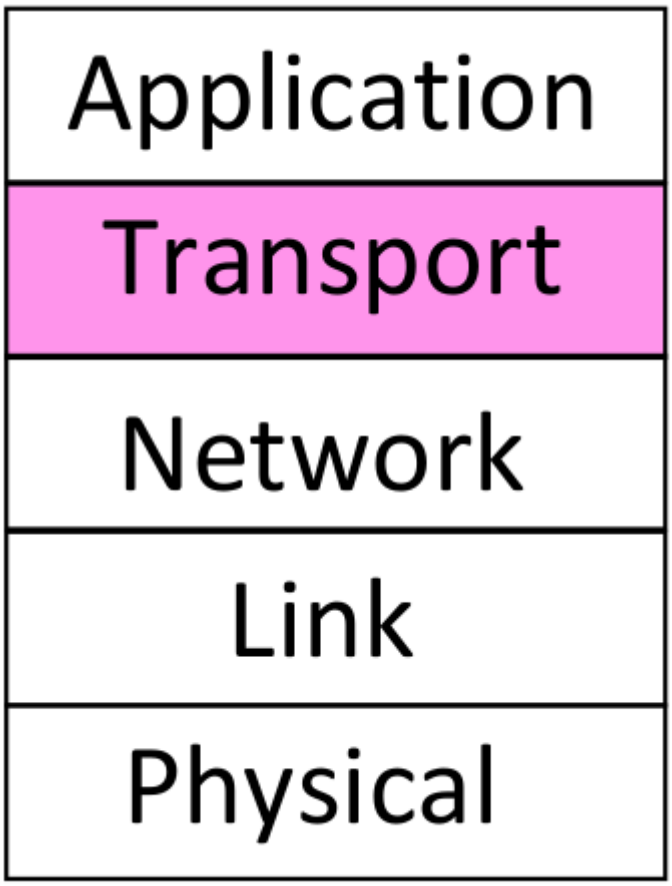
Capas:

- Aplicación
- Transporte
- Red
- Enlace
- Fisica



Las capas van con azul (Titulo)
Los diferentes temas van con rojo (Subtitulo)
Los diferentes aspectos de un tema va en verde (Titulo 1)
Las definiciones van en negrita y cursiva (Titulo 2)

Acronimos y siglas:

- **ISPs:** Proveedores de servicios de internet de acceso
- **WAN:** Red de area amplia
- **L:** Bits per packet
- **R**: Bits per second
- **MAN:** Red de area metropolitana
- **SOR:** Sistemas operativos de redes
- **API:** Interfaz para programas de aplicacion
- **CT:** Capa de transporte
- **ET:** Entidad de transporte = software / hardware de la CT
-

IP: Internet protocol

Tipos de comunicacion:

- **Interredes:** Existen muchas redes con hardware y software diferente
    - Problema: ¿Como comunicar personas pertenecientes a redes diferentes?
    - Solucion: Usar interredes
        - Interred (internet): Conjunto de redes interconectadas
        - Puertas de enlace: Conectan redes de distintas tecnologias

Estructura de la internet: Problema: Dados miles de ISP (Internet Service Provider) de acceso, como conectarlos entre si?
Conectarlos a todos entre si es un desproposito super ineficiente.

- Idea 1: Conectar cada ISP de acceso a un ISP global de transito.
- Idea 2: Tener ISPs globales de transito que conectan los IPS de acceso.

Los ISP globales de transito no tienen presencia en cada ciudad o region del mundo, esto implica que hay ISPs de acceso que no se pueden conectar a ISP globales .
En este caso existen ISPs regionales al cual se conectan los ISP de acceso en la region.

Tipos de redes:

- **Redes de area amplia (WANs):**
  Cubre un area geográfica grande, tipicamente un pais o hasta un continente:
  Esta organizada como:
    - Subred: Varios enrutadores conectados entre si que forman un grafo
    - A una subred pueden estar conectadas computadoras o LAN enteras
    - Para ir de una maquina a otra hay distintas **rutas alternativas**

  Como se hace para enviar mensajes en una WAN?:
  Algoritmo de almacenamiento y reenvio:
    - Un paquete sigue una ruta de enrutadores
    - El paquete se almacena enteramente en cada enrutador de la ruta
    - El paquete almacenado en un enrutador espera alli hasta que la linea requerida de salida este libre y luego se reenvia al siguiente enrutador.

  (Como saber cuanto tarda en transmitir un paquete en una WAN: L/R, L = bits per packet, R = speed in bps.)

  Encolado y perdida de paquetes: Si la tasa de llegada al enlace (en bits) excede la tasa de transmision del enlace por un periodo de tiempo.
  Si se da ese caso:
    - Los paquetes se van a encolar y esperaran a ser transmitidos en el enlace
    - Los paquetes pueden ser descartados (perdidos) si la memoria (bufer) se llena.

Demora del almacenamiento y reenvio: Dnodal = Dproc + Dqueue + Dtrans + Dprop

Dproc: procesamiento del nodo
- Chequeo de errores
- Determinar la linea de salida
- typically < msec

Dqueue: Demora por encolado
- Tiempo de espera en el enlace de salida para transmisión.
- Depende de cuan congestionado esta el enrutador.

Dtrans: Demora de transmisión (Cuánto demora el cable en enviar la información)
Dprop: Demora de propagación (Tiempo que el tren de bits tarda en llegar al otro lado)

DSL: Digital subscriber line

MAN: (Redes de area metropolitana): Cubren una ciudad y las hay de dos tipos
- Redes de cable: se basan en la red de TV por cable
- Redes mobiles: son redes inalambricas de alta velocidad

MAN basada en tv por cable:
- Cable coaxial sirve para unir varias casas
- Elementos de conmutacion son para comunicar viviendas en distintos cables coaxiales
- Elementos de conmutación se unen por cables de fibra optica

MAN Wimax: -----

Redes de área local (LAN): es una red operada privadamente dentro de un edificio o casa. (Tambien puede operar en un campus de varios edificios.
Puede usarse en un hogar o en una organización, las LAN usadas por companias se llaman redes empresariales.

Dos tipos de LAN:
- LAN inalámbricas: en su forma mas simple las maquinas se comunican entre si por medio de una estacion base (access point).
- LAN Ethernet: en su forma mas simple, las maquinas se conectan por medio de cables a un switch

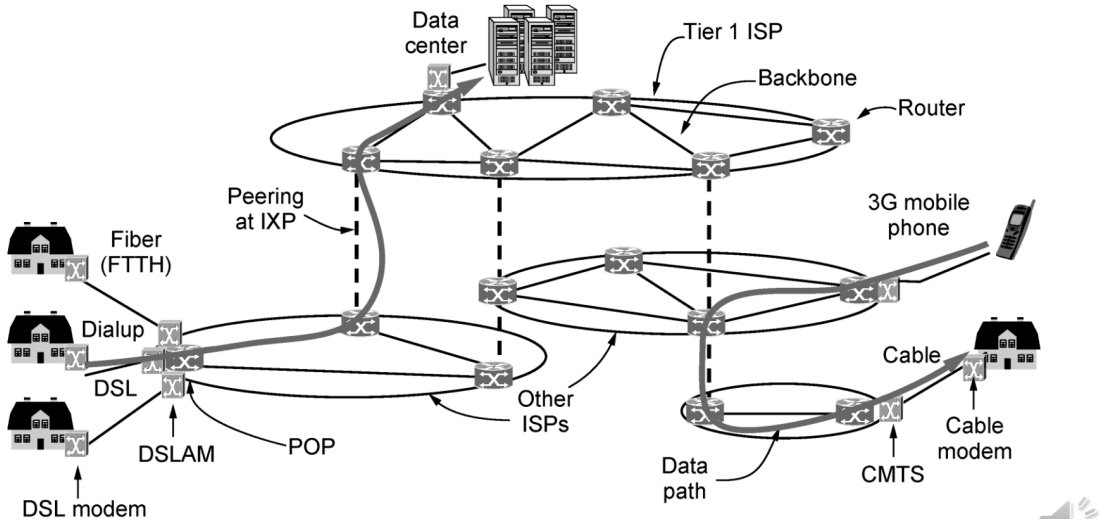
Difusion: Si una maquina envia un mensaje, todas las demas lo reciben

Puede pasar que se envian mensajes desde mas de una maquina simultáneamente, estos colisionan y se dañan. Hay que evitar o minimizar colisiones, detectarlas y tratarlas

Redes de accesio empresarial: Lo mismo que las hogareñas pero usualmente tienen servidores institucionales de web, mail, etc. Y ademas tienen tasas de transmision mas altas que la hogareña.

Internet

- Hay redes dorsales
- Red dorsales (backbone) están conectadas a varias WAN
- Redes metropolitanas pueden conectarse a WANs
- LANs están conectadas a WANs o a redes metropolitanas



Protocolo: conjunto de reglas que denominan una conversación.

SOR = Sistemas operativos de redes.

Para que las maquinas en un tipo de red se puedan comunicar hacen falta protocolos de comunicacion; los SOR contienen esos protocolos

Los SOR estan organizados como una pila de capas o niveles, que se usan para reducir la complejidad del diseño de los SOR.

Arquitectura Multicapa:

- Ofrece ciertos servicios a las capas superiores
- Oculta la implementación a las capas superiores

Para especificar como es la conversacion entre capas se definen **protocolos**.

Protocolo de capa n = Reglas y convenciones usadas en la conversacion entre la capa n de una maquina y la capa n de otra maquina.

Las capas se comunican entre si cuando se envia el mensaje (yendo desde la capa mas alta hacia la mas baja) y cuando se recibe el mensaje (yendo desde la capa mas baja hasta la mas alta)

Problemas en el diseño de las capas y sus soluciones :

Principal problema: Falta un mecanismo para identificar a las maquinas de una red
Solucion: se usan direcciones para las maquinas.

- **Control de flujo (capa 4, capa de transporte):**

Puede pasar que un emisor rapido satura de datos al receptor hasta que este ya no puede almacenar mas datos que le llegan y comienza a perder datos.
Esto se solcuiona con el uso de **retroalimentación al emisor,** osea avisarle al emisor cuando puede mandar paquetes.

- **Fragmentación de mensajes (capa 3, capa de red):**

Las capas imponen un tamaño máximo, puede pasar que mensajes que llegan no pueden ser aceptados en una capa, ya que los procesos son incapaces de aceptar mensajes que superan una cierta longitud. Para eso fragmentamos el mensaje original en mensajes mas pequeños, transmitir fragmentos y reensamblar mensajes

2 tipos de fragmentación:

- **Transparente:** Cuando el mensaje es transmitido, el receptor lo re-ensambla
- **No transparente:** El mensaje se transmite fragmentado y el receptor lo deja fragmentado hasta llegar al destino final (un host).

- **Congestión:**

Debido a las limitaciones de los enrutadores y las lineas de salida una red tiene una determinada capacidad de conduccion de mensajes, lo cual puede originar perdidas o demoras en la entrega de mensajes.

2 causas posibles de congestión:

- **Congestion por trafico de un unico nodo:** El nodo es mas rapido que el enlace, el buffer se llena y empieza a perder paquetes.
- **Congestion por trafico de multiples nodos:** Multiples nodos intentan usar un mismo enlace, y la suma de velocidad de tasas de estos nodos supera a la del enlace.

Idea de solución: Que las maquinas emisoras se enteren de la congestión y reduzcan el tráfico de salida. No es fácil darse cuenta de que hay un problema de congestión!.

Modelo de referencia TCP/IP:

Capa de aplicación

En la capa de aplicación tenemos las aplicaciones de red, cada aplicacion nos ofrece un servicio especifico.
Como esta por encima de la capa de transporte, usa sus servicios.

Para desarrollar aplicaciones de red el programador puede usar:
1. Para especificar la comunicacion, una API (Application Programming Interface)
2. Usar una middleware que provee servicios al software de la aplicacion

Arquitectura de aplicaciones:
- Cliente-servidor
- peer to peer (P2P)

Cliente servidor:

Hay 2 procesos que se comunican; uno en la maquina cliente y otro en la maquina servidor.

Forma de comunicación:
1. El proceso cliente le manda solicitud al proceso servidor
2. El proceso cliente espera un mensaje de respuesta
3. Luego el proceso servidor recibe y procesa la solicitud
4. El proceso servidor manda mensaje de respuesta al proceso cliente

Características de los servidores:
- Siempre estan en un host
- Con direccion IP permanente
- Se pueden usar centros de datos para escalabilidad

Características de los clientes:
- Pueden estar conectados intermitenmente
- Usando direcciones IP dinamicas
- Los clientes no se pueden comunicar entre si

Arquitectura P2P:

- Minimo o ningun apoyo en los servidores
- Hostst arbirtrarios (llamados compañeros) que se comunican directamente entre si.
- Compañeros piden servicio de otros compañeros y proveen servicio en retorno a otros compañeros.
- Nuevos compañeros traen nueva capacidad de servicio, asi como nuevas demandas de servicios.

- Los compañeros se conectan intermitentemente y cambian las direcciones IP.

Cosas a definir en un protocolo de capa de aplicacion:

- **Tipos de mensajes:** Intercambiados (pej de pedido, de respuesta)
- **Sintaxis del mensaje:** Que campos hay en un mensaje y como los campos son delineados
- **Semantica del mensaje:** Significado de la informacion en los campos
- **Reglas:** de cuando y como los procesos envian y responden a mensajes
- **Estado:** De la aplicacion. En que consiste y como se lo mantiene

Tipos de protocolos:

- **Abiertos:**
    - Son definidos en RFCs
    - Permiten interoperabilidad
- Propietarios

La web

Un enlace incrustado en una pagina web necesita una manera de nombrar una pagina en la web, para eso existe las URLs.

Como funciona un browser:

- El navegador permite pedir (usando protocolo HTTP) una pagina/objetos a un servidor web
- El servidor web retorna (usando HTTP) la pagina/objetos en respuesta al pedido
- El navegador interpreta el texto y los comandos de formateo que contiene y despliega la pagina adecuadamente formateada en la pantalla.

Navegadores:

Problema: Cuando el cliente recibe una pagina, no necesariamente es HTML, Como sabe el navegador de que tipo de pagina se trata?

Solucion: Cuando un servidor regresa una pagina tambien regresa alguna informacion adicional acerca de ella:

- Tipo MIME de la pagina
- Las paginas de tipo text/html de despliegan de una
- Si el tipo MIME no es de los integrados
    - El navegador consulta una tabla de tipos de MIME que asocia un tipo MIME con un visor

Servidores Web: A un servidor web se le proporcionara el nombre de un archivo correspondiente a una pagina a buscar y regresar

Problema: En el diseño anterior cada solicitud requiere un acceso al disco para obtener el archivo, Esto es ineficiente por que la misma pagina puede ser pedida innumerables veces.

Solucion: cache en la memoria

Problema: Hasta ahora un servidor web es un proceso con un solo hilo de ejecucion .¿Como se podria hacer al servidor mas rapido?

Solucion: Arquitectura con un modulo front ed y k modulos de procesamiento - MP - (hilos)

Cuales son los pasos de un servidor Web con multiples hilos?:

1. Cuando llega una solicitud el front end la acepta y construye un

registro corto que la describe.
2. Después entrega el registro a uno de los MP.
3. El MP primero verifica el caché para ver si el archivo está allí.
4. Si el archivo está en caché actualiza el registro para incluir un apuntador al archivo
5. Si el archivo no está en caché el MP inicia una operación de disco.
    a. Cuando el archivo llega del disco se coloca en la caché y se regresa al cliente.
6. Mientras uno o más MP están bloqueados esperando a que termine una operación del disco, otros MP pueden estar trabajando en otras solicitudes.
7. Conviene tener además múltiples discos, para que más de un disco pueda estar ocupado al mismo tiempo.

¿Como funciona un MP?

1. Resuelve el nombre de la página web solicitada.
    a. Manejo de solicitud entrante sin el nombre real del archivo.
2. Realiza control de acceso en la página web.
    a. Para páginas no disponibles para el público en general.
    b. Si la solicitud se puede satisfacer a partir de la identidad y ubicación del cliente.
    c. Se puede prohibir que dominios particulares accedan a la página web.
3. Verifica el caché.
4. Obtiene del disco la página solicitada o ejecuta un programa para construirla.
5. Determina el tipo MIME que se incluirá en la respuesta.
6. Regresa la respuesta al cliente.
7. Realiza una entrada en el registro del servidor (log).

Cookies:

Proposito: Comprender como se organiza y comunica informacion de estado de sesion de una aplicacion web.

En los pedidos y respuestas HTTP se envia informacion de estado de sesion

Que es una cookie y que tamaño tiene?:

- Una cookie es un pequeño archivo o string de a lo subo (4 KB)
- El contenido de una cookie toma la forma de nombre = valor

Estructura de una cookie:

- **Dominio:** Cada dominio puede almacenar hasta 20 cookies por cliente
- **Ruta:** En la estructura del directorio del servidor, identifica que partes del arbol de archivos del servidor podrian usar el cookie
- El **campo contenido** toma la forma de nombre = valor
- El campo **expira:**
    - Si este campo esta ausente:

        - El navegador descarta el cookie cuando sale
    - Si se proporciona una hora y una fecha:
        - Se mantiene la cookie hasta que expira ese horario
- El campo **Seguro:**
    - Se usa para indicar que el navegador solo puede retornar la cookie a un servidor usando un transpore seguro
    - Esto se usa para alicaciones seguras (p.ej. comercio electrónico, actividades bancarias)

Del lado del cliente se guardan en un directorio de cookies, del lado del servidor se guardan en una base de datos.

Protocolo de comunicacion HTTP (Hyper Text Transfer Protocol):

Cosas que se necesita que soporte un protocolo para la web:
- Pedido de páginas, de objetos, o de ejecución de programas que generan páginas.
- Manejo del estado de sesión
- Poder mantener el sistema de archivos del servidor web
- Recepción de páginas por un browser
- Seguridad (encriptación de mensajes)
- Feedback adecuado cuando no se puede responder los pedidos.
- Comunicación confiable

Http transfiere paginas de servidores web a navegadores y manda pedidos de navegadores a servidores web.

2 tipos de HTTP:

- No persistente: Por cada conexion TCP se manda un objeto (un solo objeto por conexion) (HTTP 1.0, ineficiente)
- Persistente: Multiples objetos pueden ser enviados a traves de una unica conexion TCP entre el cliente y el servidor. (HTTP 1.1)

Informaciones que deberia tener un mensaje de pedido:
- En caso que se quiera recibir una pagina:
    - El **URL** de un documento
    - La **especificación de programa que genera pagina web**
- El **tipo de accion** que se quiere hacer en el sistema de archivos del servidor web (meter paginas, borrar paginas, etc)
- Mandar **informacion sobre la maquina/software del cliente** para que el servidor web pueda retornar paginas adecuadas al cliente
- **Informacion del estado de sesion** para que el servidor se entere
- **Restricciones sobre el tipo de paginas** que el cliente puede aceptar.

Informaciones que debería tener un mensaje de respuesta:
• Feedback adecuado sobre el pedido realizado

P. ej: cuando no se puede cumplir con el pedido.
• Página o documento solicitado.
• En ese caso información sobre el tipo de documento enviado.
– P.ej. el tipo MIME del documento, cuando fue modificado por última vez el documento, etc.
• Información de estado de sesión para mantener actualizado al cliente.

Paginas estaticas: Escritas en HTML, si se actualiza la informacion hay que cambiarla a mano
Paginas Dinamicas: Paginas HTML generadas por medio de programas del lado del esrvidor, estos programas toman parametros de entrada que suelen ser ingersados como valores de campos de formulario HTML

Pasos para generar paginas dinamicante del lado del servidor:

1. Un usuario llena el formulario y hace click en el boton de envio
2. Se envia un mensaje al servidor web con el contenido del formulario
    a. Se propociona el mensaje a un programa o secuencia de comandos
    b. El programa procesa el mensaje
3. El programa solicita informacion a un servidor de bases de datos
4. El servidor de bases de datos responde con la informacion requerida
5. El programa genera una pagina HTML personalizada y la envia al cliente
6. El browser muestra la pagina recibida al usuario

En las paginas dinamicas con una URL no basta para especificar la pagina dinamica deseada, es por eso que en la URL tambien se envian los parametros

Capa de transporte (CT)

Provee comunicacion logica entre procesos de aplicacion que ejecutan en diferentes sistemas finales.

Comunicación Logica: como si los hosts ejecutando los procesos estuvieran directamente conectados.

Entidad de transporte (ET) = software/hardware de la CT

Problemas que soluciona la capa de transporte:

- – Uso de temporizadores y las retransmisiones de paquetes.
- – El direccionamiento explícito de los destinos.
  - • ¿Cómo hacer para que un proceso adecuado atienda a las necesidades de una máquina cliente?
  - • El proceso podría no estar activo, el cliente podría no saber cuál proceso usar, etc.
- – Uso de búferes y control de flujo.
- – Evitar congestionar la red poniendo demasiados paquetes en ella.
  - • Cuando la CR pierde paquetes, la CT puede solucionarlo.

Segmento: unidad de datos del protocolo de transporte

Confirmaciones de recepcion de paquetes enviados
Tipos de paquetes que deben ser confirmados:

- paquete de datos
- Paquetes con información de control

La capa de transporte debe permitir

- La entrega de segmentos al host de destino

- Que la entrega de segmentos sea ordenada (respetando el orden del flujo de datos a enviar recibido de la capa de aplicación)

¿Como lo hace?

Para la entrega ordenada de segmentos al host de destino se puede:

- Numerar los segmentos a enviar (usando *Numeros de secuencia)* repestando el orden del flujo de datos recibido de la capa de aplicacion
- Usar para cada numero de segmento enviado un ***temporizador de retransmisiones***
- Mandar ***confirmaciones de recepcion (ACK)***
- Si expira el temporizador de un segmento sin recibir el ACK, retransmitir el segmento correspondiente
- Los segmentos recibidos son re-ensamblados en orden y entregados a la capa de aplicacion del receptor

TCP (protocolo de control de transmision):

- Meta: proporiconar un flujo de bytes confiable de extremo a extremo a traves de una interred no confiable

TCP se adapta dinamicamente a las propiedades de la interred y se sobrepone a muchos tipos de fallas

ET TCP (ETCP)

Usaremos la palabra TCP para referirnos a veces a la ETCP y a veces al protocolo TCP

Problemas que resuelve TCP:

– Retransmisión de paquetes:
  • uso de números de secuencia, confirmaciones de recepción y temporizadores.
– Fijar la duración de temporizadores de retransmisiones (algoritmo complejo)
– Manejo de conexiones entre pares de procesos
– Direccionamiento
– Control de congestión
– Control de flujo

Una ETCP acepta flujos de datos a transmitir de proceso locales, Cada flujo de datos se divide en fragmentos que no excedan los 64 KB llamados segmentos, y se envia cada segmento dentro de un datagrama IP

El servicio TCP se obtiene al hacer que tanto el servidor como el cliente creen sockets, Direccion de un socket = IP + Puerto
Para obtener el servicio TCP se debe establecer una conexion explicitamente entre el socket en la maquina emisora y uno en la maquina receptora

Importante: Cada byte de un flujo de datos a enviar en una conexión TCP tiene su propio número desecuencia de 32 bits que se usa para confirmaciones recepcion y otros asuntos.

Límites que restringen el tamaño de un segmento
– Cada segmento, debe caber en la carga útil de 65.515 bytes del IP.
– Cada red tiene una ***unidad máxima de transferencia (MTU)*** y cada segmento debe caber en la MTU.
• En la práctica la MTU es usualmente de 1500 bytes (el tamaño de la carga útil de Ethernet).

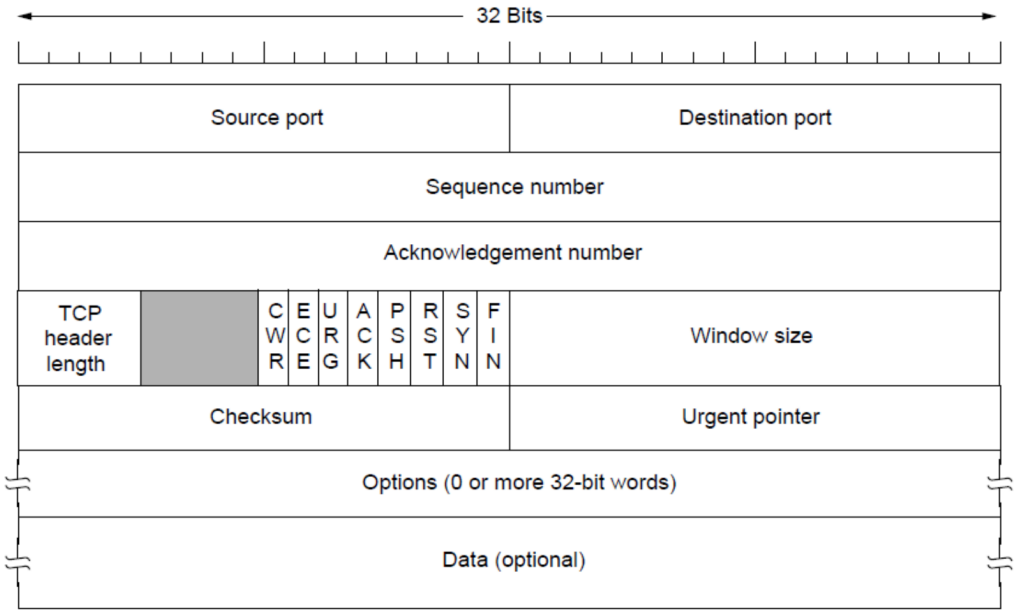
Problema: La capa de red (que incluye IP) no proporciona ninguna garantia de que los datagramas se entregaran de manera apropiada (ni que se vayan a entregar)

Solucion:

- Si un datagrama se recibe correctamente se confirma su recepcion
- Si no se confirma la recepcion de un datagrama luego de un intervalo de tiempo enconces se debe retransmitir

Segmentos TCP

1. **Encabezado fijo** de 20 bytes
2. **Opciones de encabezado** en palabras de 32 bits
3. **Datos** opcionales



Los segmentos sin datos se usan para acks y mensajes de control.

Puerto de origen y puerto de destino:

- Son de 16 b cada uno
- La direccion de un puerto mas la direccion IP del host forman un punto terminal unico de 48 b

- Los puntos terminales de origen y de destino en conjunto identifican la conexion

El campo **numero de secuencia** de un segmento es un numero de byte en el flujo de bytes transmitido y corresponde al primer byte en el segmento. Tiene 32 b de longitud

El campo **numero de confirmacion de recepcion** indica el siguiente byte esperado del flujo de bytes a transmitir, Tiene 32 b de longitud

Longitud del encabezado TCP: Nº de palabras de 32 bits en el encabezado TCP
Longitud del campo de opciones: variable

Direccionamiento

Problema: El direccionamiento explicito de los destinos
¿Como hacer para que un proceso servidor adecuado atienda a las necesidades de una maquina cliente?
Pues el cliente podria no saber cual proceso servidor es el adecuado para un servicio particular, o podria saberlo pero el proceso servidor no atiende por que esta inactivo

Solucion: Existe un proceso especial llamado **servidor de directorio** que para cada tipo de servicio sabe cuales son los puertos de los servidores que prestan ese tipo de servicio.
Para esto:

1. El usuario establece conexion con el servidor de directorio (que escuha en un puerto conocido)
2. El usuario envia un mensaje especificando el nombre del servicio
3. El servidor de directorio le devuelve la direccion puerto
4. El usuario libera la conexion con el servidor de directorio y establece una nueva con el servicio deseado

Ahora, para el problema de que el servidor este inactivo, hay que usar el servidor que ejecuta los servidores inactivos (**protocolo incial de conexion)**

Servidor de procesos = intermediario de los servidores de menor uso

Demonios = procesos servidores que atienden en un puerto

inetd = demonio de internet

Control de flujo

"Hay que evitar que un host emisor rápido
Desborde a un host receptor lento"

La ET emisora debe manejar buferes para los mensajes de salida, ya que puede hacer falta retransmitirlos. El emisor almacena en bufer todos los segmentos hasta que se confirma su recepcion.

El receptor tambien maneja buferes para los mensajes que llegan, ya que si la llegada de segmentos del emisor es mucho mas rapido que el receptor para procesar los segmentos recibidos, entonces el receptor necesitara poder almacenar segmentos antes de procesarlos.
Tambien lo necesitan dado que si llegan un grupo de segmentos y faltan segmentos previos a ellos habra que almacenar los segmentos de ese grupo en buffer.

Cuando entra un segmento al receptor, intenta adquirir un bufer nuevo, si hay uno disponible se acepta, sino se lo descarta.

La capa de enlace de datos se ocupa del control de flujo entre dos maquinas directamente conectadas entre si (puede ser enrutador o host)

De todas formas en la capa de transporte se necesita control de flujo ya que el receptor puede demorarse en procesar mensajes debido a los problemas de la red[1]

Protocolo de parada y espera:

Comportamiento del emisor:

1. El emisor envia paquete P y para de enviar
2. Espera: el emisor espera una cantidad razonable de tiempo para la confirmación de recepcion (ACK)
3. Si llega el ACK a tiempo, se envia el siguiente paquete, GOTO 2
4. Sino se retransmite paquete P, GOTO 2

Si hay paquete o ACK demorado: La retransmision va a ser un duplicado con igual Nº de secuencia; luego se descarta.

Parada y espera tiene un desempeño pobre

Denvio : demora en enviar un paquete (L/R)
Usender: *utilizacion - fraccion del tiempo en que el emisor esta ocupado enviando ((L/R)/(RTT+(L/R)))*
RTT: *es el tiempo de ida y vuelta de un bit*

Tuberia:

Puede pasar que se perdio un ACK y se envio el paquete de nuevo, lo cual genera que el mismo paquete llegue 2 o mas veces al receptor y la capa de transporte la pasa a la capa de aplicacion mas de una vez.
Para esto se asignan *numeros de secuencia* a los paquetes que salen.

Es por eso que en tuberia el emisor puede mandar multiples paquetes al vuelo a ser confirmados, para esto el rango de numeros de sequencia debe ser incrementado y se deben usar buferes en el emisor y/o en el receptor.

Hay 2 formas genericas de protocolos de tuberia Retroceso N y Repetición selectiva

Retroceso-N:

Receptor envia ack acumulativo, el cual no confirma paquetes si hay un agujero.
El emisor tiene un timer para el paquete mas viejo no confirmado, cuando expira el timer retransmite todos los paquetes no confirmados

Si un paquete T a la mitad de una serie larga se daña o pierde, no se pueden entregar a la capa de aplicacion los paquetes que llegaron bien despues de T, ya que la CT receptora debe entregar paquetes a la capa de aplicacion en sequencia.

Cuando se daña un paquete, Retroceso-N descarta todos los paquetes subsecuentes al paquete perdido, sin enviar ack para los paquetes descartados

***Ventana corrediza (Intervalos de numeros de secuencia)*:** se usan dentro del espacio de numeros de secuencia. Se usan para representar el limite de cantidad de paquetes enviados + paquetes por enviar que puede almacenar el emisor en buferes.

Ventana emisora: Tramas enviadas sin ack positivo o tramas listas para ser enviadas.

Espacio de secuencia: MAX_SEQ + 1 numeros de secuencia (comenzando en 0)

Ventana emisora: MAX_SEQ

Para evitar que haya mas de MAX_SEQ paquetes sin ack pendientes hay que prohibir a la CR que moleste con mas trabajo

El principal problema de este metodo es el uso ineficiente del canal frente a segmentos perdidos o demorados.

Repeticion selectiva:

El receptor envia confirmaciones idividuales para cada paquete.
El emisor mantiene un timer para cada paquete no confirmado, Cuando el timer expira, retransmite solo ese paquete no confirmado.

Si un paquete T a la mitad de una serie larga se pierde, tenemos el mismo problema que en retroceso N.
¿Como lo solucionamos?
Los paquetes en buen estado recibidos despues de un paquete dañado E se almacenan en bufer, Cuando el paquete E llega correctamente, el receptor entrega a la capa de aplicacion, en secuencia todos los paquetes posibles que ha almacenado en el bufer.

En vez de retransmitir un paquete cuando la transmision falla, Repeticion selectiva propone el uso de una ACK negativa (***NAK***) por el receptor, asi se estimula la retransmision de paquetes antes que los temporizadores terminen y asi se mejora el rendimiento.

Si la NAK se pierde, el receptor confirma individualmente todos los paquetes recibidos correctamente, El emisor solo reenvia paquetes para los cuales el ACK no fue recibido o se recibio un NAK.

Ventana del emisor: Contiene N Nº de secuencias consecutivos, Limita Nº de secuencias a enviar a paquetes no confirmados

En la ventana del emisor puede haber:

- Paquetes enviados y confirmados por que antes hay paquetes no confirmados
- Paquetes enviados y no confirmados
- Paquetes listos para enviarse en búfer

Detallitos de repeticion selectiva:

- Tamaño de ventana emisora Comienza en 0 y crece hasta MAX_SEQ
- El receptor tiene un bufer para cada Nº de secuencia en su ventana

Regla para el tamaño de la ventana receptora: Tamaño= (MAX_SEQ + 1)/2

Control de flujo cuando la cantidad de datos que quiere recibir y procesar el receptor varia:

La cantidad de datos que quiere recibir y procesar el receptor varia, esto se puede deber al cambio del patron de trafico en la red, o que se abran o cierren varias conexiones en el receptor. Como consecuencia el receptor debe ajustar dinamicamente su alojamiento de buferes y ahora el emisor no sabe cuantos datos puede mandar en un momento dado, pero si sabe cuantos datos le gustaria mandar.

Es por eso que el emisor **solicita espacio en bufer en el otro extremo.**
Cuando el receptor recibe este pedido, sabe cual es su situacion y cuanto espacio puede otorgar.

Una **situación** que se puede dar es que la informacion de reserva de buferes viaja en un segmento que no contiene datos y ese segmento se pierde, lo cual termina ocasionando un deadlock.
Para evitar eso cada host puede enviar periodicamente un segmento de control con el ack y estado de buferes de cada conexion.

Control de flujo en TCP:

No se requiere:

- Que los emisores envien datos tan pronto como llegan de la aplicacion
- Que los receptores envien confirmaciones de recepcion tan pronto como sea posible
- Que los receptores entreguen datos a la aplicacion apenas los reciben

Campo tamaño de ventana en el encabezado TCP:

- Nº de bytes que pueden enviarse comenzando por el byte cuya recepcion se ha confirmado, 0 indica que se han recibido los bytes hasta nº de confirmacion de recepcion -1 inclusive, pero el receptor quisiera no recibir mas datos por el momento.
- El permiso para enviar puede otorgarse enviando un segmento con el mismo nº de confirmacion de recepcion y un campo tamaño de ventana distinto de 0

Si la ventana anunciada por el receptor es de 0, el emisor debe detenerse hasta que el proceso de aplicacion del host receptor retire algunos datos del bufer, en ese momento el TCP puede anunciar una ventana mas grande.

Cuando la ventana es de 0 el emisor no puede enviar segmentos, salvo en 2 situaciones:

- Datos urgentes (p.ej para uqe el usuario elimine el proceso en ejecución en la maquina remota)
- Re anunciar el siguiente byte esperado y el tamaño de la ventana

• Situación: El receptor recibe un segmento malo y luego una gran cantidad de segmentos buenos.
• ¿Cómo se puede manejar esta situación?
• Solución : el receptor solicita segmento/s específico/s mediante segmento especial llamado NAK.

– Tras recibir segmento/s faltante/s, el receptor puede enviar una confirmación de recepción de todos los datos que tiene en búfer,
• Para habilitar esta solución se usa una opción (del campo de opciones) llamada repetición selectiva (muy usada).

En las lineas con alto ancho de banda, alto retardo o ambas cosas, la ventana de 64 KB con frecuencia es un problema, Un tamaño de ventana mas grande permitiria al emisor continnuar enviando datos, pero como el campo de tamaño de ventana es de 16 bits, es imposible expresar tal tamaño.

Solucion: Permitir al emisor y al receptor negociar un factor de escala de ventana

Aun si el receptor esta de acuerdo en usar buferes, todavia queda la cuestion del tamaño de estos.

Soluciones:

- Usar todos los buferes del mismo tamaño : Si hay una variacion grande en el tamaño de los segmentos, el grupo de buferes de tamaño fijo representa problemas de perdida de memoria.
- Usar buferes de tamaño variable: Mejor uso de memoria : Administracion de buferes mas complicada
- Dedicar un solo bufer circular grande por conexion : Buen uso de la memoria cuando todas las conexiones tienen una carga alta : Deficiente si algunas conexiones cuentan con poca carga.

Control de congestión

Si un emisor manda a un receptor mas informacion que la capacidad de carga de la subred, la subred se congestionara, pues sera incapaz de entregar los segmentos a la velocidad con que llegan.

Control de congestión en TCP:

Algunos hosts disminuiran la tasa de datos.

Ventana de congestion: Para llevar la cuenta de cuantos datos un host puede enviar por la red, TCP. Su tamaño es el numero de bytes que el emisor puede tener en la red en todo momento.

Cuando en TCP un host detecta congestion, El host ajusta el tamaño de la VC.

¿Como detecta TCP la congestion?:
La expiracion de un temporizador causada por un paquete perdido se puede deber a:

- Ruido en la linea de transmisión
- El descarte de paquetes en el enrutador congestionado

TCP supone que las expiraciones de tiempo son causadas por congestion

Para calcular el tamaño de la **ventana de congestion** (**VC)**
Esta el algoritmo de **Arranque Lento** que se basa en la idea de probar con un mínimo de datos e ir duplicando gradualmente hasta que no se pueda más

1. El emisor asigna a la VC el segmento de tamaño máximo (STM) usado por la conexión; entonces envía 1 STM.
2. Emisor y receptor se ponen de acuerdo en el tamaño del STM.
3. Si se recibe el ack de este segmento antes que expire el temporizador, el emisor agrega el equivalente en bytes de un segmento a la VC para hacerla de 2 STM y envía dos segmentos.
4. Cuando la VC es de n segmentos, si de todos los n se reciben acks a tiempo, se aumenta la VC en la cuenta de bytes correspondiente a n segmentos.
5. La VC sigue creciendo exponencialmente hasta expiración temporizador (timeout) o alcanzar el tamaño de la ventana receptora.

6. Si ocurre timeout se recorta la VC a tamaño *VC/*2, o sea no se enviarán ráfagas de segmentos mayores a VC/2.

Asumimos que cada paquete que llega al receptor dispara un paquete ACK

ACK`s duplicados: Cuando se pierde un segmento y despues llegan otros segmentos el receptor genera acks que confirman lo mismo.

Llega un ACK duplicado por que llego otro segmento al receptor y el segmento perdido no dio señales de vida

Muchas causas pueden llevar a ACK`s duplicados, por lo que TCP asume que 3 ACK`s duplicados implican que el paquete se perdio. **Luego ese paquete puede retransmitirse inmediatamente y antes de que expire el temporizador. <- *Retransmision rapida.***

Algoritmo de control de congestion de Internet (TCP Talhoe):

Usa un umbral ademas de las ventanas de recepcion y congestion.

Al ocurrir una expiracion del temporizador o detectarse 3 acks duplicados, se fija el umbral en la mitad de la ventana de congestion actual, y la ventana de congestion se restablece a un segmento maximo.

- A partir del punto en el que se alcanza el umbral las transmisiones exitosas aumentan linealmente la ventana de congestión (en un segmento máximo por ráfaga).
- Recomenzar con una ventana de congestión de un paquete toma un RTT (para todos los datos previamente transmitidos que dejen la red y para ser confirmados, incluyendo el paquete retransmitido).
- Si no ocurren mas expiraciones de temporizador/3 acks duplicados, la ventana de congestion cotinuara creciendo hasta el tamaño de la ventana del receptor.

Comenzar con arranque lento cada vez que haya un timeout puede ser un poco ineficiente. Es por eso que tenemos como solucion a!

Algoritmo de TCP Reno:

1. Luego de iniciada la conexion se comienza con arranque lento
2. La VC crece linealmente hasta que se detecta una perdida de paquete.
    a. Se cuentan acks duplicados
3. El paquete perdido es retransmitido usando retransmision rapida
4. **Recuperación rápida:**
    a. Se manda un paquete por cada ack duplicado recibido
    b. Un RTT luego de la retransmision rapida el paquete perdido es confirmado
    c. La recuperacion rapida termina con esa confirmacion de recepcion
5. Luego de recibir el nuevo ack:
    a. La ventana de congestion de una conexion se achica a la mitad de lo que era cuando se encontraron 3 duplicados (decrecimiento multiplicativo)

   b. El conteo de ack duplicados se pone en 0
6. Luego la ventana de congestion va incrementando de a un segmento por cada RTT (crecimiento aditivo)
7. Este comportamiento se repite indefinidamente

Problemas de tener segmentos duplicados retrasados y su resolucion.

Es necesario saber si un segmento que llega a un host es duplicado o no
Queremos saber cuando 2 segmentos son distintos, ¿que hacemos?:

- Enumerarlos con numero de secuencia : No por que estos no pueden tener un tamaño arbitrario y queremos que los segmentos tengan longitud máxima.

Si un paquete queda vivo mucho tiempo dentro de la red, este puede generar problemas.
Es por eso que hay que asegurar que ningún paquete viva más allá de T sec

Tenemos que lograr que al regresar al principio de los nº de secuencia, los segmentos viejos con el mismo nº de secuencia hayan desaparecido hace mucho tiempo, Para eso el espacio de secuencia debe ser lo suficientemente grande para garantizar eso.
Para saber cuan grande (en bits) debe ser el numero de secuencia necesitamos saber cuanto es la mayor cantidad de segmentos de STM Bytes se pueden mandar continuamente durante T a Ancho de Banda Mbps.

Hay que escoger como numero inicial de secuencia de la conexion nueva un nº de secuencia que haga imposible o improbable que el duplicado retrasado de nº de secuencia genere problemas.

Soluciones posibles:

1. Al crear una nueva conexion, cada extremo genera un nº de secuencia de 32 bits aleatorio que pasa a ser el numero inicial de secuencia para los datos enviados
2. Vincular nº de secuencia de algun modo al tiempo y para medir el tiempo usar un reloj.
    - Cuando se establece una conexion los k bits de orden mayor del reloj = numero inicial de secuencia

Establecimiento y liberación de conexiones

Como al establecer una conexion se usan segmentos, una conexion deberia tener un Nº inicial de secuencia con el que se comienza a operar, Usamos la idea de vincular el Nº inicial de secuencia al tiempo y para medir el tiempo usar el reloj.

Implementacion de la idea (Tomlinson):

- Cada host tiene un reloj de hora del dia
    - Los relojes de los hosts no necesitan ser sincronizados
    - Se supone que cada reloj es un contador binario que se incrementa a si mismo en intervalos uniformes
    - El reloj continua operando aun ante la caida del host.
- Cuando se establece una conexion los k bits de orden mayor del reloj = numero inicial de secuencia

El espacio de secuencia debe ser lo suficientemente grande; para lograr que al regresar al principio de los nº de secuencia los segmentos viejos con el mismo nº de secuencia hayan desaparecido hace mucho tiempo

Caida de hosts:

Problema: Cuando un host se cae, al reactivarse sus ET no saben donde estaban en el espacio de secuencia. Por lo que no se sabe que numero de secuencia generar para el siguiente segmento a ser enviado.

Solucion: Requerir que las ET esten inactivas durante T segundos tras una recuperacion para permitir que todos los segmentos viejos expiren.

Establecimiento de Conexión:

Para establecer conexion el host de origen envia un segmento CONNECTION REQUEST al destino y espera una respuesta CONNECTION ACCEPTED.
Supongamos que se establecen conexiones haciendo que un host 1 envia segmento S = CR N, P a host 2, Donde N es nº de secuencia y P es nº de puerto.

- Host 2 confirma ese pedido con segmento CA N

Caso: S se demora demasiado en llegar a host 2, vence timer en host 1 y host 1 manda un duplicado S' = CR,N,P al host 2.

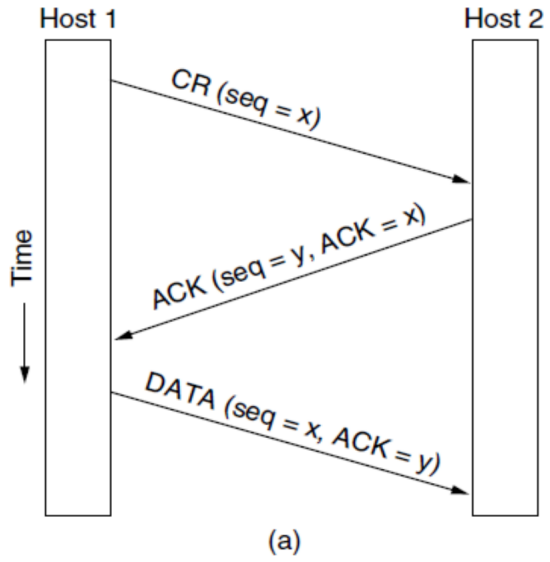
- Luego puede pasar que host 2 reciba S' y un buen tiempo despues S.

Situacion: No se recuerda en el destino nº de secuencias para conexiones.

Problema: No tenemos forma de saber si un segmento CR conteniendo un nº de secuencia inicial es un duplicado de una conexion reciente o una conexion nueva.

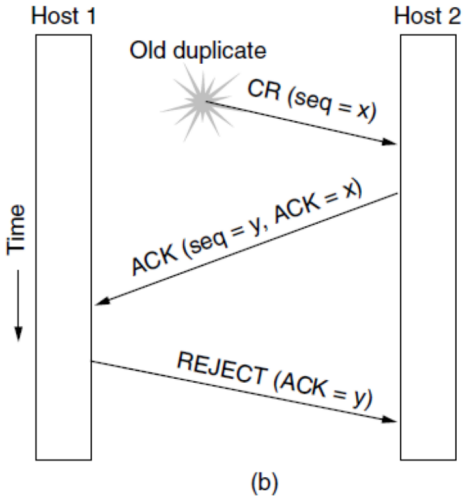
Solucion: Acuerdo de tres vias de Tomlinson.

- **Happy path:**



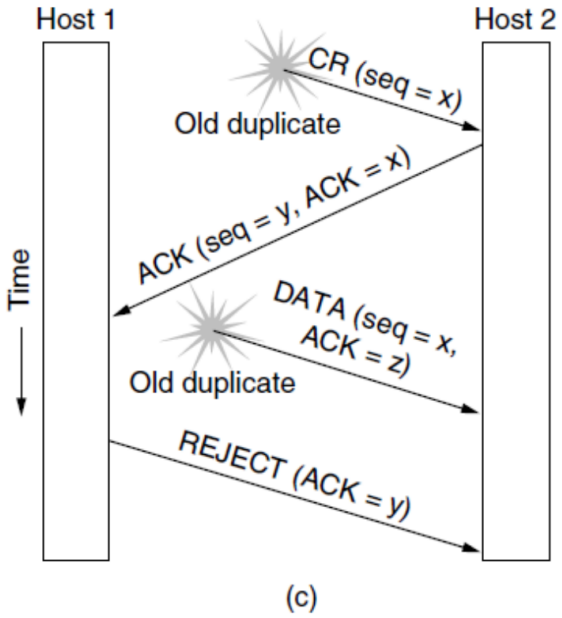
(a)

- **Caso en el que llega un CR duplicado al host 2**
  Al rechazar el host 1 el intento de establecimiento de conexion del host 2, el host 2 se da cuenta de que fue engañado por un duplicado con retardo y abandona la conexion; asi un duplicado con retardo no causa daño



(b)

- **Caso de tanto segmento CR como de datos con retraso:**
  Cuando llega el segundo segmento retrasado al host 2 , el hecho de que se confirmo la recepcion de z en lugar de y indica al host 2 que este tambien es un duplicado viejo.



(c)

Establecimiento de una conexión TCP:

El nº de secuencia inicial de una conexión no es 0.

- Se usa un esquema basado en reloj con un pulso de reloj cada 4 ***usec***
- Al caerse un host no podra reiniciarse durante el tiempo maximo de paquete (120 seg)

Campos del encabezado TCP para el establecimiento de conexiones:

SYN: Se usa para establecer conexiones

- Solicitud de conexion: SYN = 1, ACK = 0
- La respuesta de conexion si lleva una confirmacion de recepcion, por lo que tiene SYN = 1, ACK = 1

En TCP las conexiones usan el acuerdo de 3 vias

1. Para establecer una conexion, el servidor espera pasivamente una conexion entrante ejecutando LISTEN y ACCEPT
2. En el lado del cliente ejecuta CONNECT la cual envia un segmento TCP con el bit SYN encendido y el bit ACK apagado y espera una respuesta
3. Al llegar el segmento al destino, la ETCP alli revisa si hay un proceso que haya ejecutado un LISTEN en el puerto indicado en el campo puerto destino
4. Si no lo hay envia una respuesta con el bit RST encendido para rechazar la conexion
5. Si algun proceso esta escuchando en el puerto ,ese proceso recibe el segmento TCP entrante y puede entonces aceptar o rechazar la conexion; si la acepta se envia un segmento de ACK.
6. La secuencia de segmentos TCP enviados en el caso normal se muestra en la figura siguiente.

Liberacion de conexiones:

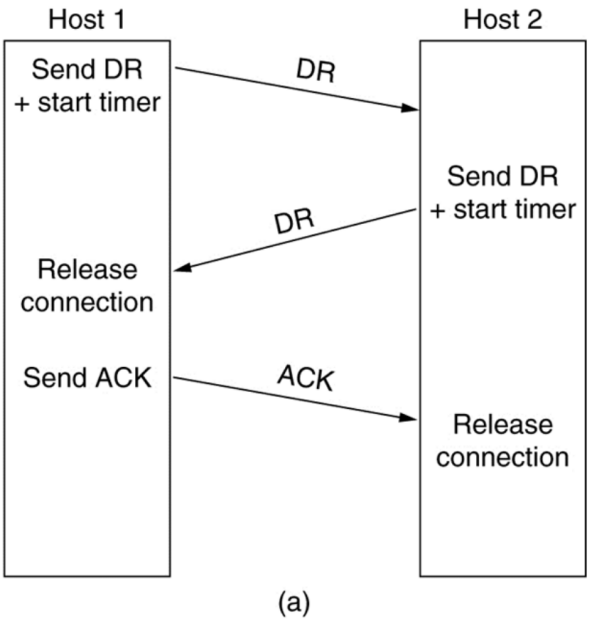
Idea 1: Podriamos pensar en un protocolo en el que:
- El host 1 dice “ya termine ¿Terminaste tambien?”
- Si el host 2 responde “Ya termine tambien. Adios”, la conexion puede librarse con seguridad

Un protocolo asi no siempre funciona por el *problema de los 2 ejercitos*
En el cual se plantea que basicamente un host nunca sabe si el otro recibio el mensaje o no.

Idea: Permitir que cada parte decida cuando la conexion esta terminada.
Cuatro escenarios de liberacion de conexion usando un acuerdo de 3 vias:

La liberacion de conexion en un host significa que la ET remueve la informacion sobre la conexion de su tabla de conexiones abiertas y avvisa de alguna manera al dueño de la conexion.

- **Happy Path:**
    a. Host 1 envia un segmento DR e inicia un temporizador para el caso que lo llegue DR de host 2.
    b. Al llegar DR al host 2, este emite un segmento DR e inicia un temporizador para el caso que no llegue respuesta de host 1;
    c. Al llegar esta DR el host 1 envia de regreso un segmento ACK y libera la conexion
    d. Cuando el segmento ACK llega el host 2 tambien libera la conexion



(a)

- Caso 2: se pierde el ultimo segmento ACK:
    - Al expirar el temporizador la conexion se libera de todos modos.
- Caso 3: Se pierde el segundo DR:
    - El host 1 no recibira la respuesta esperada, su temporizador expirara y todo comenzara de nuevo
- Caso 4: Respuesta perdida y DRs subsiguientes perdidos:

    - Tras N reintentos el emisor se da por vencido y libera la conexion.
    - Mientras tanto tambien termina el temporizador del receptor y tambien se sale

Este protocolo falla si se pierde la DR inicial y N retransmisiones, ya que el emisor se dara por vencido y liberara la conexion, pero el otro lado no sabra nada sobre los intentos de desconexion y seguira plenamente activo. Esta situacion origina una conexion abierta a medias.

Para matar conexiones abiertas a medias:

- Si no llego ningun segmento durante una cierta cantidad de segundos al host 2, se librea automaticamente la conexion en el host 2.
- Luego el host 1 detectara la falta de actividad y tambien se desconectara
- Esta solucion tambien resuelve el caso que la red se rompio y los hosts ya no pueden conectarse.

Para la implementación, es necesario que cada ET tenga un temporizador que se detenga y se reinicie con cada envio de segmento.

Liberacion simetrica:

- Una de las partes emite un DISCONNECT porque ya no tiene mas datos por enviar y aun esta dispuesta a recibir datos de la otra parte.
- Una conexion se libera cuando ambas partes han emitido una primitiva DISCONNECT

La liberacion simetrica es ideal cuando cada proceso tiene una cantidad fija de datos por enviar y sabe con certidumbre cuando los ha enviado. En otras situaciones la determinacion de si se ha efectuado o no todo el trabajo o si debe terminarse o no la conexion no es tan obvia.

Liberacion de una conexion TCP:

FIN: Especifica que el emisor no tiene mas datos que transmitir.
Tras cerrar una conexion, un proceso puede continuar recibiendo datos indefinidamente.
Ambos segmentos SYN y FIN tienen nº de secuencia y por tanto tenen la garantia de procesarse en el orden correcto.

Capa de red

Proposito: llevar paquetes de un host de origen a uno de destino siguiendo una ruta conveniente

Asuntos de los que se encarga la capa de red:

- Almacenamiento y reenvio
- Enrutamiento
- Control de congestion
- Conectar redes de dstintas tecnologias
- Fragmentacion

Subred: Formada por enrutadores interconectados

Como es el hardware subyacente a la Capa de red?

Varias subredes de distinta tecnologia unidas entre si usando puertas de enlace

Los formatos y tamaños máximos de paquetes difieren de una red a otra, es por eso que un paquete no puede pasar tal cual de una red a otra

Para mandar un conjunto de paquetes desde un host de origen a un host de destino se puede:

- **Servicio orientado a la conexion:** Usar una ruta fija para mandar todos los paquetes
- **Servicio no orientado a la conexion:** La ruta puede cambiar, por lo que distintos paquetes pueden seguir distintos caminos.

Servicio no orientado a la conexion:

La ruta de los hosts va a cambiar cada cierto tiempo

Paquetes = diagramas
Subredes = subredes de datagramas

¿Cómo diseñar la tabla de un enrutador?
La tabla de enrutamiento solo necesita entradas para los enrutadores de la subred.
Entrada de tabla de enrutador formada por filas: <enrutador de destino, línea de salida>.
La línea de salida es la dirección de un enrutador

Cuando llega un paquete al enrutador:

1. Se lo almacena y se comprueba que llego bien
2. Se determina el enrutador de destino asociado al host de destino
3. Se usa fila de ese enrutador de destino para reenviar el paquete por linea de salida de esa fila.

Podemos pensar que direccion de un host es un numero con dos partes:

<direccion de red, numero de maquina>

Direccion de red: Sirve para identificar una red
Numero de maquina: Sirve para identificar una maquina dentro de la red

Dada direccion de host de destino, para encontrar enrutador de destino apropiado:

- Buscar en la tabla de enrutamiento enrutador de destino cuya direccion concuerde con la mayor cantidad de bits desde la izquierda con la direccion de host de destino.

Servicio orientado a la conexion:

- Todos los paquetes se mandan por la misma ruta
- **Trabajo a realizar antes de mandar paquetes**
    - Configurar una ruta del host de origen al destino
        - Esto se llama crear una conexion
        - **Circuito virtual (CV) = conexión**

Cada paquete lleva un identificador que indica a cual CV pertenece,
Cuando no se necesita enviar mas paquetes se libera la conexion. Al hacer esto tambien se termina el CV.

Enrutamiento jerárquico:

Cuando crece mucho el tamaño de las subredes tambien lo hacen las tablas de enrutamiento

Consecuencias de tener tablas de enrutameinto grandes:
Estas tablas consumen memoria del enrutador, necesitan mas tiempo de CPU para examinarlas

Para que no crezcan tanto, existe el enrutamiento jerarquico

- los enrutadores se dividen en regiones
- Un enrutador sabe como enrutar paquetes a destinos en su region
- Tambien sabe como enrutar a otras regiones
- Pero no sabe nada de la estructura interna de las regiones en las que no esta
- Enrutamiento jerarquico trae consigo una longitud de ruta mayor

Lo cual nos lleva a:

Tablas de enrutamiento jerarquico:

- Entradas para todos los routers locales
- Entradas para las demas regiones en las que no esta en enrutador

Problema: En las redes enormes, una jerarquia de dos niveles es insuficiente.
Solucion: Agrupar las regiones en clusteres, los clusteres en zonas, las zonas en grupos, etc.

Arquitectura de un Router:

Funciones claves de un router:

- Ejecutar algoritmos de enrutamiento/protocolos (RIP, OSPF, BGP)
- Enviar paquetes de enlaces de ingreso a enlaces de salida

Queremos evitar los siguientes efectos indeseados:

- Algunos routers pueden quedar inactivos
- Los caminos pueden ser innecesariamente largos
- Se pueden sobrecargar algunas de las lineas de comunicacion y los routers asociados a ellas.

Causa: La CR elige mal las rutas para enviar paquetes.

Problema: Como escoger bien las rutas para enviar paquetes.
Solucion: usar algoritmos de enrutamiento eficientes

Subred representada como grafo:
G = (N,E)
N = Conjunto de routers = {u,v,w,x,y,z}
E = Conjunto de enlaces = {(u,v),(u,x),...}
Los arcos tienen etiquetas para el costo de atravesarlos

Algoritmos de calculo de la ruta mas corta entre dos nodos
Uno de ellos el Dijkstra (1959):

- Dado grafo conexo con costos en los enlaces, y nodo n en el grafo, obtiene árbol de caminos más cortos desde n hacia todos los demás nodos.
- El árbol de caminos más cortos se representa con un mapeo donde para cada nodo del grafo de la subred asigna su padre (en el árbol de caminos más cortos).

Procedimiento para calcular tablas de reenvío en redes de datagramas usando algoritmo de Dijtstra.
1. Construir grafo de la subred con costos.
2. Ingresar grafo de la subred con costos en los enrutadores.
3. En cada enrutador construir tabla de enrutamiento; para eso:
    a. Ejecutar algoritmo de Dijkstra en el enrutador
    b. A partir de árbol de caminos más cortos con raíz en el enrutador obtenido generar la tabla de reenvío del enrutador.

Enrutamiento de vector de distancia:

Calculo de tabla de reenvio

- Cada enrutador mantiene una tabla de enrutamiento (o de reenvio) indizada por cada enrutador en la subred.
    - Cada entrada comprende la linea preferida de salida hacia ese destino y una estimacion del tiempo o distancia a ese destino.
- A partir de su tabla de enrutamiento un enrutador E puede obtener un veRe,x = Retardo de e a x si E ~ Xctor de distancia que contiene una lista de pares <destino, retardo estimado>
- El retardo de un enrutador a un vecino suyo, puede medirlo con
    - Paquetes de ECO que el receptor simplemente marca con la hora y los regresa tan rapido como puede
- Cada t msec, cada enrutador envia a todos sus vecinos un vector de distancia y tambien recibe un vector de distancia de cada vecino.

VDx = Vector de distancia del enrutador X
VDx(i) = Distancia estimada para llegar al enrutador i desde X

-
- Se usa paquete ECO para obtenerlo

Distancia estimada desde E enrutador a **i** a traves de **X** es: Re,x + VDx(i)

Enrutamiento de Vector de Distancia

- El enrutador E estima la distancia desde E al enrutador de destino i de la siguiente manera:
    - – d(E, i) = min{Re,x +VDx(i) | X vecino de E}
- El mejor vecino para ir de E a i se define como:
    - MV(E, i) = elegir {V : Re,v +VDv(i) = d(E, i)} .
        - elegir elige un elemento de un conjunto.
- ¿Cómo se actualiza tabla de enrutamiento de E?
    - E recibió de todo vecino X suyo: VDx y Re,x
    - La tabla de enrutamiento de E en la fila del enrutador de destino i va a tener los valores: d(E, i) y MV(E, i).
    - Observar que la vieja tabla de enrutamiento no se usa en este cálculo.

Enrutamiento de estado de enlace

Problema: Se quiere un algoritmo de enrutamiento que se adapte a cambios en la topología y el trafico de la red.

Solucion: *Algoritmo de enrutamiento de estado de enlace (LSR)*

- En cada enrutador usar algoritmo de Dijkstra para encontrar la ruta mas corta de un enrutador a los demas enrutadores.
- La topologia y retardos en las lineas se distribuyen a cada enrutador
- Este algoritmo es valioso porque:
    - Repsonde rapido frente a cambios en la topologia de la red
    - Es ampliamente usado en internet (como parte del protocolo OSPF)

Que tareas debe hacer un enrutador LSR?

1. Descubrir sus vecinos

2. Medir el costo a cada uno de sus vecinos
3. Cnnstruir un paquete diciendo lo que ha aprendido
4. Enviar este paquete a todos los demas enrutadores
5. Computar el camino mas corto a cada uno de los otros enrutadores

Paquete Hello: Se envia a cada linea punto a punto para averiguar quienes son los vecinos de un enrutador. Se espera que el enrutador del otro extermo regrese una respuesta indicando quien es.

Paquete ECHO: Se envia a traves de la linea para que el enrutador conozca el retardo a sus vecinos. Una vez que llegue al otro extremo este debe regresarlo inmediatamente.
Metodo: se mide el tiempo de ida y vuelta y se divide por 2.
Problema del metodo: Asume implicitamente que los retardos son simetricos.

Cada router construye un **paquete de estado de enlace (LSP)**

Datos del LSP:
- Identidad del emisor
- Numero de secuencia
- Edad
- Lista de <vecino, retardo al vecino>

Cuando se pueden construir los LSP?:
- Construirlos a intervalos regulares
- Construirlos cuando ocurra un evento significativo, como la caida o la reactivacion de la linea o de un vecino, o el cambio apreciable de sus propiedades.

Distribución confiable de los LSP:
- Usar inundacion para distribuir los LSP.
- Se lleva registro de los paquetes difundidos.
    - Cada paquete contiene un numero de secuencia que se incrementa con cada paquete nuevo enviado (desde su router de origen)
    - Los routers llevan el registro de todos los pares <router de origen, secuencia> que ven

Cuando llega un LSP a un router, ¿Que se hace con el?:
- Ayuda: comparar el valor de su número de secuencia con el que figura en la tabla (de paquetes difundidos) para el enrutador que lo mandó.
- Si es nuevo (nuevo número de secuencia mayor que los anteriores)
    - Se reenvía a través de todas las líneas, excepto aquella por la que llegó.
- Si es un duplicado (número de secuencia mayor visto, pero repetido),
    - Se descarta.
- ▪ Si llega un paquete con número de secuencia menor que el mayor visto hasta el momento
    - Se rechaza como obsoleto debido a que el enrutador tiene datos más recientes.

¿Cuándo se puede construir la tabla de enrutamiento de un enrutador?

Una vez que el enrutador ha acumulado un grupo completo de paquetes de estado del enlace

- Construir el grafo de la subred completa.
    - Cada enlace se representa dos veces, una para cada dirección.
    - Los dos valores pueden promediarse o usarse por separado.
- Se ejecuta el algoritmo de Dijkstra para construir la ruta más corta a todos los destinos posibles.
- Con los resultados del mismo se actualiza la tabla de enrutamiento.

Inundacion:

Inundacion con contador de saltos: integrar un cotador de saltos en el encabezado de cada paquete, que disminuya con cada salto y el paquete se descarte cuando el contador llega a 0.
El contador de saltos se inicaliza segun la longitud de la ruya entre el origen y el destino, si el emisor desconoce el tamaño de la ruta, puede inicializar el cotnador al peor caso, es decir, al diametro total de la subred.

Inundacion selectiva:
Los routers no envian cada paqete de entrada por todas las lineas, sino solo pro aquellas que van aporximadamente en la direccion correcta.
Para poder aplicar inundacion selectiva, el router necesita almacenar la informacion de la direccion en la que va cada linea y en que direccion esta el destino.

Enrutamiento de vector de distancia:
Las buenas notiicias se toman rapido, las malas lento.
La razón de porqué las malas noticias viajan con lentitud es: ningún enrutador jamás tiene un valor mayor en más de una unidad que el mínimo de todos sus vecinos.
Gradualmente todos los enrutadores elevan cuentas hacia el infinito, pero el número de intercambios requeridos depende del valor numérico usado para el infinito.

- Si la métrica usada es el número de saltos, es prudente hacer que el infinito sea igual a la ruta más larga más 1.

El algoritmo de inundacion de paquetes de estado de enlace tiene algunos problemas

1. Si los numeros de secuencia vuelven a comenrazar, reinara la confusion.
    a. Usar un numero de secuencia de longitud suficiente para que el problema anterior no suceda.
2. Si llega a corromperse un numero de secuencia y se escribe 65540 en lugar de 4 (un error de un bit), los paquetes 5 a 65540 seran rechazados como obsoletos, dado que se piensa que el numero de secuencia actual es 65540.
    a. Como proteccion contra los errores en las lineas router-router, se confirma la recepcion de todos los pquetes de estado del enlace.
    b. Haria falta que antes de actualizarse el numero de secuencia mas grande, el router mande una confirmacion de recepcion al transmisor y luego espera una respuesta afirmativa o negativa del transmisor.

        i. En el primer caso se actualiza el numero de secuencia mas grande
        ii. En el segundo caso se descarta el paquete que se recibio por estar errado
3. Si llega a caerse un router (de origen) perdera el registro de su numero de secuencia. Si comienza nuevamente en 0, se rechazara el siguiente paquete.
    a. La informacion de los routers solo expira (a lo largo de la red) cuando el router esta caido.

¿Cuando se puede detectar que un enrutador esta caido?

- Cuando se actualizen las tablas de enrutamiento y se manden los paquetes Hello, se puede detectar que el router esta caido.

¿Una vez identificado que el router esta caido como proceder?:

- Se propaga eficientemente la informacion de este hecho por toda la red.
- Se hace que la informacion asociada al router caido expire (paquetes pendientes a enviar, numero de secuencia mas grande recibido, etc).

¿Como hacer que el algoritmo de inundacion de paquetes de estado de enlace sea mas eficiente?

- Una vez que un paquete de estado del enlace llega a un enrutador para ser inundado, no se encola para transmision inmediata. En vez de ello, entra en un buffer de almacenamiento donde se espera un tiempo breve.
- Si antes de transmitirlo, llega otro paquete de estado del enlace proveniente del mismo origen, se comparan sus numeros de secuencia.
    - Si son iguales, se descarta el duplicado
    - Si son diferentes, se desecha el mas viejo

El buffer de paquetes para un enrutador contiene una celda por cada paquete de estado de enlace recien llegado, pero aun no procesado por completo.

Una fila de la tabla del bufer de paquetes de un enrutador contiene:

- Origen del paquete, numero de secuencia, edad, datos de los estados de enlaces.
- Banderas que pueden ser:
    - Banderas de confirmacion de recepcion: indica a donde tiene que enviarse la confirmacion de recepcion del paquete.
    - Banderas de envio: significan que el paquete debe enviarse a traves de las lineas indicadas.
    - Si llega un duplicado mientras el original aun esta en el bufer, los bits de las banderas tienen que cambiar.

Control de congestion:

La cola en un bufer que precede a un enlace tiene capacidad finita
¿Que pasa con un paquete cuando llega a una linea de salida con buffer lleno?
El paquete se pierde.

Los paquetes perdidos deben ser retransmitidos por el enrutador previo o el host emisor.

La meta del control de congestion es asegurar que la subred sea capaz de transportar el trafico ofrecido.

Problemas de los algoritmos de control de congestion de TCP estudiados.
El host de destino demora demastiado en enterarse de congestion (solo por expiracion de temporizador de retransmisiones o 3 acks duplicados).
Los hosts solo se enteran de perdidas de paquetes, no pueden controlar que paquetes perder y cuales no.

Formas de disminuir la carga en la subred:
- Regulacion de trafico:
    - Hacer que hosts responsables de la congestion se enteren mas rapido (que con protocolos de TCP) de la congestion y reduzcan su tasa de transferencia.
- Desprendimiento de carga:
    - Enrutadores descartan paquetes inteligentemente antes que se saturen buferes.

Identificacion de la congestion:
Cada entrutador monitorea la demora de la cola de linea de salida
- Asociar a cada linea: d = demora reciente de cola de esta linea
- Tomar periodicamente una muestra de la lognitud de cola instantanea de la linea, s
- Actualizar d peridociamente usando: **d**nvo = a **d**ant + (1-a) * s
    - Donde a determina la repidez con que el enrutador olvida la historia reciente

Siempre que d rebasa un umbral, la linea de salida entra un estado de advertencia.
- Cada paquete nuevo que llega se revisa para ver si su linea de salida esta en estado de advertencia.
- Si es asi, se realiza alguna accion

Regulacion de trafico:
Es cuando los emisores ajustan sus transmisiones para enviar un trafico que la red pueda soportar.
Una vez que un enrutador tiene una linea de salida en estado de advertencia puede avisar a los hosts responsables de los paquetes que llegan a esa linea congestionada.

Metodo de paquetes reguladores:
1. Usar paquetes reguladores si la linea de salida esta en estado de advertencia
    a. El enrutador regresa un paquete regulador al host de origen, proporcionandole el destino encontrado en el paquete.
2. Para que el paquete original no genere mas PR mas adelante en la ruta, en el paquete original se activa un bit del encabezado y despues se reenvia
3. El PR le pide al host de origen que reduzca en un porcentaje X el trafico enviado al destino especificado.
4. El host ignora los PR que se refieran a ese destino por un intervalo fijo.
5. Una vez que haya expirado ese tiempo, el host escucha mas PR durante un intervalo I.

   a. Si llega alguno el host reduce el flujo aun mas y comineza a ignorar nuevamente los PR.
   b. SI no llega ningun PR durante I el host incrementa el flujo

Como sabemos que se pueden llegar a perder paquetes indiscriminadamente , elijamos bien cuales se pierden, ya que algunos son mas importantes que otros.

Criterios para escoger que paquetes descartar:

- Si nos basamos en el tipo de aplicacion que se esta usando
  - Estrategia vino: Descartar primero los paquetes mas nuevos
  - Estrategia leche: Descartar primero los paquetes mas viejos.
- si nos basamos en la importancia de los paquetes:
  - Marcar los paquetes con clases de prioridades
  - Los enrutadores primero se desprender de paquetes de la clase mas baja, luego de la siguiente clase, etc.

Solucion 2: Usar desprendimiento de carga junto con reduccion de trafico.

- La respuesta a paquetes perdidos por desprendimiento de carga es que el origen disminuya su tasa de transferencia.
- Si expira el temporizador de retransmisiones, el emisor lo toma como perdida de paquete.
- Vemos ahora una implementacion de esta solucion.

Algoritmo de deteccion temprana aleatoria (RED):

- Para detectar cuando comienza a descartar paquetes, los enrutadores mantienen un promedio movil de sus longitudes de cola.
- Cuando este promedio de una cola C sobrepasa el umbral
  a. Una pequeña fraccion de los paquetes son descartados al azar.
- Con cada uno de esos paquetes:
  a. El enrutador elige un paquete al azar de C.
  b. Se descarta el paquete seleccionado
  c. El origen notara falta de ACK y la capa de transporte disminuira la velocidad de transmision.
- Consecuencias de elegir paquetes al azar:
  a. Hace mas probable que los hosts emisores mas rapidos pierdan un paquete, lo noten y reduzcan su tasa de transferencia.

Aprovisionamiento de redes:

Como reducir las congestiones?

Idea 1: Usar una tecnica de control de admision para evitar que empeoren las congestiones que ya han comenzado y que consiste en que una vez que se ha detectado la congestion no se estavlecen CVs nuevos hasta que ha desaparecido el problema.

Idea 2: Permitir el establecimiento de nuevos CV, pero enrutando cuidadosamente los cicuitos nuevos por otras rutas que no tengan problemas.

Idea 3: Negociar un acuerdo entre el host y la subred cuando se establece un CV.

- Este arreglo normalmente especifica el volumen y la forma de trafico, la calidad de servicio requerido y otros parametros.
- Para cumplir con su parte del acuerdo, la subred por lo general reservara recursos a lo largo de la ruta cuando se establezca el circuito.
- Estos recursos pueden incluir espacio en tablas y en bufer en los enrutadores y ancho de banda en las lineas.
    - De este modo es poco probable que ocurran congestiones en los CV nuevos.

Metodo de bit de advertencia: Señalar el estado de advertencia activando un bit especial en el encabezado del paquete.

- Cuando el paquete llega a su destino, la entidad transportadora copia el bit en la siguiente confirmacion de recepcion que regresa al origen.
- A continuación el origen reduce el trafico.
- Mientras el enrutador esta en estado de advertencia, continua activando el bit de advertencia, lo que significa que el origen continua obteniendo confirmaciones de recepcion dicho bit activado.
- El origen monitorea la fraccion de confirmaciones de recepcion con el bit activado y ajusta su tasa de transmision de manera acorde.
    - En tanto los bits de advertencia continuan fluyendo, el origen conitnua disminuyendo su tasa de transmision.
- Cuando la tasa de transmision disminuye lo suficiente, el origen incrementa su tasa de transmision.
    - Debido a que cada enrutador a lo largo de la ruta puede activar el bit de advertencia, el trafico se incrementa solo cuando no habia enrutadores con problemas.

Una implementacion de bit de advertencia usada por TCP es **ECN (Explicit Congestion Notification):**

- Se usa en TCP/IP
- Se marcan 2 bits en el encabezado IP con distintos fines:
    - 00: transporte no capaz de ECN
    - 10: transporte capaz de ECN, ECT(0)
    - 01: transporte capaz de ECN, ECT(1)
    - 11: congestion encontrada, CE
- Si ambos extremos soportan ECN mandan sus paquetes con ECT(0) y ECT(1) respectivamente.
- Si paquete atraviesa cola congestionada y el enrutador soporta ECN, se cambia codigo en el paquete a CE para avisar al receptor de la congestion.

Se usan dos banderas en encabezado TCP para soportar ECN:

- ECE (ECN echo): se usa para mandar indicacion de congestion al emisor
- CWR (ventana de congestion reducida): es usada para confirmar que la indicacion ECE fue recibida.

Secuencia de ejecución de ECN típica:

1. Se negocia ECN en conexión TCP

2. Emisor manda paquete IP P con ECT(0)
3. P llega a enrutador congestionado que soporta ECN y enrutador marca P con CE.
4. Receptor recibe P con CE y manda segmento Q (con ACK de P) de vuelta usando bandera ECE prendida.
5. Emisor recibe Q con ECE prendido, entonces emisor reduce ventana de congestión.
6. Emisor manda siguiente segmento al otro extremo usando bandera CWR prendida para confirmar recepción de aviso de congestión.
7. Nota: Se contnua transmitiendo segmentos con ECE prendido hasta recibirse segmento con CWR prendido.

Probelma del metodo paquetes reguladores:
A altas velocidades o distancias grandes, el envio de un paquete regulador a los hosts de origen no funciona bien porque la reaccion es muy lenta

Solucion: Metodo de paquetes reguladores salto por salto:
Hacer que el paquete regulador ejerza su efecto en cada salto que da.

- Cuando el paquete regulador llega a un enrutador F, se le obliga a F a reducir el flujo al siguiente enrutador D (F debera destinar mas buferes a flujo)
- Luego el paquete regulador llega al enrutador E anterior a F e indica a E que reduzca el flujo a F. Esto impone una mayor carga a los buferes de E, pero da un alivio inmediato a F. Y se sigue asi sucesivamente.

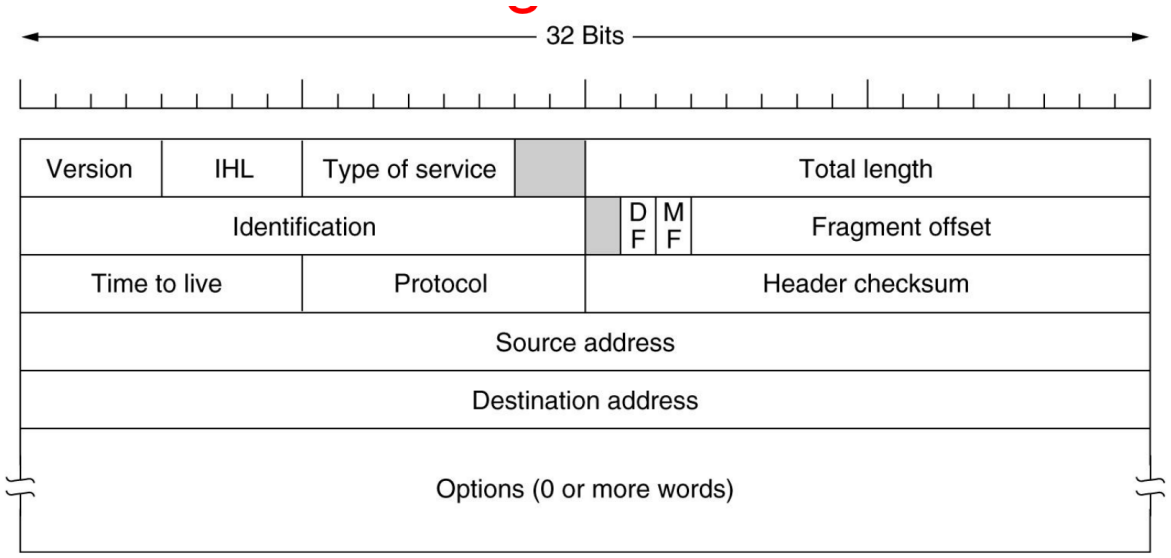
IP y NAT:

IP tiene 2 versiones:

- IPv4 con direccion IP de 32 bits.
- IPv6 trabaja con direcciones IP de 128 bits.

Datagrama IP = encabezado + texto
Encabezado = parte fija de 20 bytes + parte opcional

- Un encabezado tiene varios campos
- Cada tipo de informacion que necesito va en uno o mas campos
- La parte opcional tiene longitud variable



- **IHL (4b)**
  - Longitud del encabezado en palabras de 32b (5 <= valor <= 15)
  - 5 Cuando no hay opciones
- **Total length (2B)** de encabezado + datos <= 2^16 B
- **Type of service:**
  - Los 2 ultimos bits se usan para informacion de notificacion de congestion (para ECN)
  - Los 6 primeros bits se usan para indicar clase de serivico (p.ej entrega rapida, transmision libre de errores, etc)
- **Protocol (8 b)** dice a cual proceso de transporte (TCP,UDP) entregar el paquete
- **Identification** se usa para que el host de destino determine a que paquete un fragmento pertenece.
- **Time to live** se usa para limitar el tiempo de vida de un paquete.
  - Debe decrementarse en cada salto
  - Cuando llega a cero el paquete es descartado y se manda un paquete de advertencia al host de origen
  - Esto evita que los paquetes anden dando vueltas demasiado tiempo
- **Header Checksum:** Se usa para detectar errores cuando el paquete viaja a lo largo de la red.
  - Debe recalcularse en cada salto, porque el campo tiempo de vida siempre cambia

Direcciones IPv4:
En un datagrama IP los campos direcciones de origen y de destino
- Cada una tiene 32 b
- Indican el numero de red y el numero de maquina
- Consecuencias
  - Uso numeros IP diferentes para distinguir las maquinas de una red
  - Las direcciones IP son jerarquicas

Una maquina puede tener mas de un IP:
- Una maquina tiene un IP por cada red a la que esta conectada

- Pero el asunto es mas complejo como vemos a continuacion.

Interfaz: Conexion entre host/enrutador y enlace fisico.
- Un enrutador tiene muchas interfaces, una por cada linea de salida
- Un host tiene una o ods interfaces:
    - Con Ethernet cableada
    - con inalambrica 802.11

Cada interfaz tiene asociada una direccion IP.

128.208.0.0/24 =
- La porcion de la red es de 24 bits
- Tengo 2^8 maquinas en la red
- La direccion IP mas baja en el bloque es 128.208.0.0

Una **mascara** esta formada de 1s para identificar la red seguido de 0s para identificar las maquinas

Subredes:
Conjunto de interfaces de dispositivos con la misma parte de red de la direccion IP
Otra definicion: Maquinas que se pueden alcanzar fisicamente entre si sin la necesidad de un enrutador interviniente.

Asignacion de redes a organizaciones:

Efecto sobre el reenvio de paquetes de tener una tabla grande:
- Los enrutadores deben buscar en esa tabla para enviar cada paquete y enrutadores en un ISP grande pueden tener que enviar millones de paquetes por segundo
    - PAra esto hace falta hardware especial y una computadora de proposito general no alcanza

Efecto sobre el algoritmo de enrutamiento de tener una tabla grande:
- El costo de actualizar las tablas de enrutamiento es grande

Conclusion: Evitar las tablas de reenvio demasiado grandes.

Idea de la solucion para la primera parte de la pregunta: Alojar las direcciones IP de una red en un bloque contiguo que permite 2^k maquinas.

Solucion: CIDR (Classless Inter Domain Routing):
- En todas las maquinas de la red, la parte de la direccion IP para identificar la red es la misma.
- Se representa la red asignada con un unico prefijo

Para saber como queda la ultima direccion segun cuantas direcciones pida hacemos esto

University	First address	Last address	How many	Written as
Cambridge	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Available)	194.24.12.0	194.24.15.255	1024	194.24.12/22
Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20

Obviamente supongamos que no sabemos las last adress.
A la first address la pasamos a binario, recordando que son de 32 bits.
192.24.0.0 (no nos importan los dos primeros campos, solo el 0.0) = 00000000 00000000
a eso le sumamos lo que queramos sumar en binario, en este caso
2048 = 00000111 = 7 11111111 = 255
Entonces la ultima direccion nos queda 192.24.7.255

Como podria definirse la tabla de enrutamiento?
El enrutamiento es jerarquico y solo se representan redes de organismos

- Cada entrada de la tabla de enrutamiento se extiende para darle una mascara de 32 bits.
- **Tabla de enrutamiento** para todas las redes tiene entradas:
    - (direccion IP inicio subred, mascara, linea de salida)

¿Como se usa la tabla de enrutamiento cuando llega un paquete?

1. Extraer direccion de destino IP.
2. Luego analizar la tabla entrada por entrada
    a. Hacer AND de la mascara de la entrada con la direccion de destino y comparar el resultado con la direccion IP de inicio de la subred de la entrada.
3. Si coinciden entradas multiples se usa la mascara mas larga (la red mas pequeña)

Continuacion de CIDR:
Para evitar que las tablas de enrutamiento crezcan demasiado se combinan varios prefijos en un prefijo unico mas grande (conocido como superred)

- A esto se le llama agregacion de prefijos.

A distintas regiones geograficas se asignan distintos espacios de direcciones, esto se aprovecha combinando prefijos de varias redes que estan en una misma region geografica en un prefijo para un enrrutador que esta en otra region alejada.

Situacion: Un ISP tiene una red de /c esto quiere decir que se le dan 2^(32-c) numeros IP para estas maquinas

- Con el esquema actual los clientes no pueden tener mas de 2^(32-c) maquinas usando el servicio del ISP en un momento dado.

Problema: Como aumentar la cantidad de maquinas que usan el serviicio del ISP bien por arriba de las 2^(32-c) a pesar de tener una red de /c?

Resolverlo aumentaria drasticamente la cantidad de maquinas que pueden acceder a internet.

Solucion:

NAT (Network Adress Translate):

Asignar un solo Nº de IP a cada organizacion para el trafico de internet.

1. Dentro de la organizacion cada computadora tiene una direccion IP unica que se usa para el trafico interno.
2. Cuando un paquete sale de la organizacion y va al ISP, se presenta una traduccion de direccion (de la direccion de la computadora en la organizacion a la direccion IP usada por la organizacion en internet).

La unica regla esque ningun paquete que contiene estas direcciones pueda aparecer en la intenet:

- 10.0.0.0 – 10.255.255.255/8 (16,777,216 hosts)
- 172.16.0.0 – 172.31.255.255/12 (1,048,576 hosts)
- 192.168.0.0 – 192.168.255.255/16 (65,536 hosts)

Cada mensaje TCP saliente contiene puertos de origen y de destino que sirven para identificar los procesos que usan la conexion en ambos extremos.

¿Qué pasa con el uso de puertos cuando un proceso quiere establecer una conexión TCP con un proceso remoto?

- Se asocia a un puerto TCP sin usar en su máquina conocido como puerto de origen (indica donde enviar mensajes entrantes de esta conexión)
- El proceso proporciona tambien un puerto de destino para decir a quien dar los mensajes en el lado remoto.

Problema: Cuando la respuesta vuelve, por ejemplo de un servidor web, se dirige naturalmente a direccion IP de la compañia, **¿Como sabe ahora la caja NAT con que direccion se reemplaza?**

Solucion 1: Guardar asociacion en la caja NAT de numero IP al puerto de origen que viene en el mensaje TCP/UDP dentro del paquete.

Pero esto logra que si dos maquinas usan el mismo puerto de origen entonces ya no se pueda distinguir cual es cual

Solucion 2: Distinguir entre en Nº de puerto usado para identificar la maquina (o sea IPs en la red interna) y el Nº de puerto usado por TCP/UDP para identificar la conexion.

- Cuando llega un paquete con puerto de origen se busca en la tabla el IP del nodo y el Nº del puerto que se usa para la conexion.

Tabla de traduccion de la caja NAT:

Los indices en la tabla son numeros de puerto para identificar la maquina

- Una entrada de la tabla contiene:
    - (numero de puerto para identificar la conexion, direccion IP)

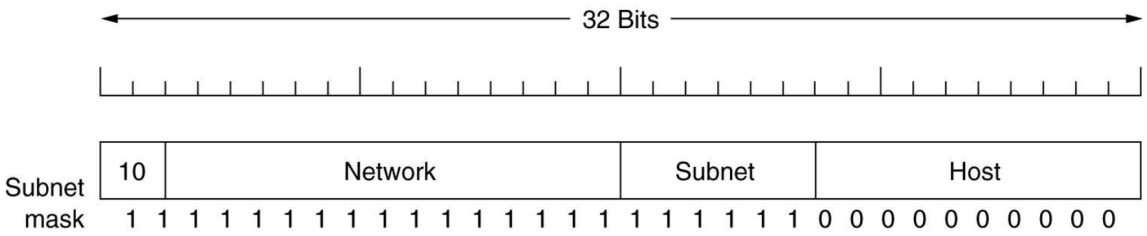
El protocolo IP

El campo tiempo de vida es un contador usado para limitar el tiempo de vida de los paquetes

- Es decrementado en cada salto
- Cuando llega a 0 el paquete es descartado
    - Cuando esto sucede se envia un paquete de advertencia al host de origen

Problema: Cuando un paquete entra en el enrutador principal ¿Como sabe a cual subred pasarlo?

Solucion: algunos bits se eliminan del Nº de host para crear un numero de subred.



Problema: ¿Como expresar subredes?

Solucion: El enrutador principal usa una mascara de subred que indique la division entre el numero de red + numero de subred y el host.

¿Como serian las tablas de enrutamiento para el enrutador principal?

- Se tienen entradas con forma de
    - (direccion IP inicio subred, mascara)
    - Cuando un paquete llega al enrutador principal, el enrutador hace un AND booleano de la direccion de destino con la mascara de subred para deshacerse del numero de host y buscar la direccion resultante en sus tablas (hay que ver si coincide con la direccion de inicio de subred o prefijo).

Problema: Queremos hacer la red mas grande.

La cantidad máxima de hosts se da por la cantidad de 0 a la derecha del ultimo 1 en la dirección de origen:

IPv6

Problema: El espacio de direcciones de 32 bit y ha sido agotado en varias regiones del mundo
Solucion: Considerar un espacio de direcciones mucho mas grande,
Problema: Con IPv4 algunos campos del encabezado hacen que el procesamiento de datagramas en los enrutadores lleve tiempo:

Requisitos:

- Que el formato de encabezado ayude a aumentar la velocidad de procesamiento y reenvio
- Cambios en el encabezado para facilitar la calidad de servicio

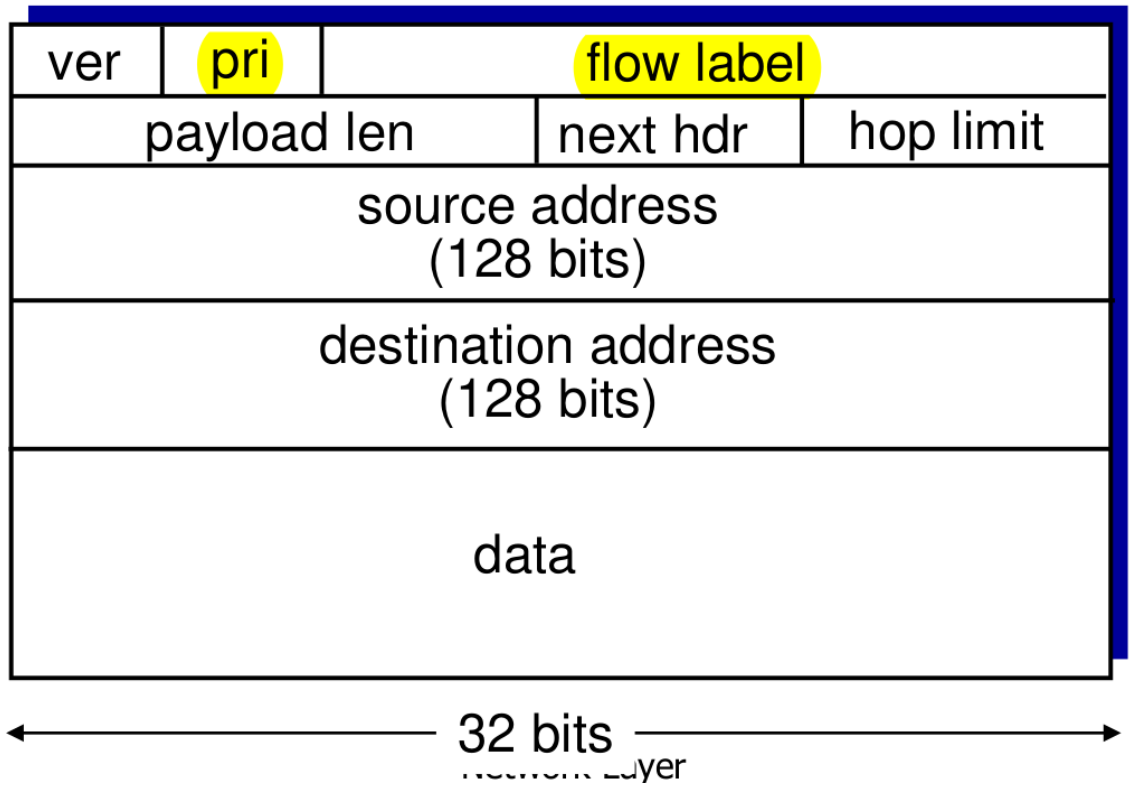
¿Por que hace falta que el procesamiento de encabezados sea mas rapido?:

- Por que las redes cada vez son mas rapidas, en cambio la velocidad de los procesadores se esta estabilizando
- Entonces para compensar hay que agilizar el procesamiento de los datagramas

Formato de datagrama IPv6:

- Encabezado de longitud fija de 40 bytes para procesamiento mas rapido de datagramas
- Capacidad de redireccionamiento expandida: direcciones de 128 bits
- Etiquetado de flujos: se etiquetan paquetes que pertenecen a un mismo flujo para los cuales el emisor requiere manejo especial.

Si un paquete tiene una etiqueta de flujo, los enrutadores pueden ver en las tablas internas para ver que tipo de tratamiento especial requiere.



Etiqueta de flujo: (20 b) para identificar datagramas en el mismo “flujo”.
Prioridad tiene dos usos:

- Para dar prioridad a ciertos datagramas dentro de un flujo
- Para dar prioridad a datagramas de ciertas aplicaciones sobre datagramas de otras aplicaciones.

Longitud de carga util: (16 b) numero de bytes en el datagrama IPv6 luego del encabezado de (40 b)

Limite de saltos: (8 b) el contenido de este campo se decrementa en 1 por cada enrutador que entrega el datagrama. Si el contador alcanza 0 el datagrama se descarta.

Proximo encabezado (8 b) significa:

- Cual de los 6 encabezados extension de opciones actuales le sigue al encabezado
- Si este encabezado es el ultimo encabezado IP, el campo dice a cual protocolo de transporte entregar el datagrama.
- Los encabezados de opciones tambien tienen este campo.

Direcciones IPv6:

- Son escritas como 8 grupos de 4 digitos hexadecimales
- Para separar los grupos se usa :
- Grupos con 16 bits iguales a 0 pueden remplazarse con dos “:”
- Ceros a la izquierda de grupos pueden ser omitidos

Otros cambios en relacion a IPv4:

- No se permite fragmentacion ni re-ensamblado en enrutadores intermedios.
    - Esto solo puede hacerse por el origen y el destino
- Suma de vertificacion: Removido para reducir el tiempo de procesamiento en cada salto (ya la capa de transporte y de enlace de datos usan suma de verificacion).
- Opciones: estan permitidas pero fuera del encabezado, indicado por el campo de proximo encabezado.

Problema: ¿Que se puede hacer si un datagrama es demasiado grande para pasar por una linea de salida de un enrutador?

- Un enrutador descarta paquetes que son demasiado grandes para la linea de salida
- Manda al emisor un mensaje de paquete demasiado grande
- Luego el emisor puede reenviar los datos usando datagramas IP mas chicos.

NAT:

NAT 444:
Los proveedores de servicio de internet (PSI) tambien pueden tener NAT
El espacio de direcciones IP reservado para NAT 444 es 100.64.0.0/10

DHCP (Dinamic Host Configuration Protocol):

Meta: Permitir a los hosts cuando se unen a la red obtener dinamicamente su direccion IP a partir de servidor de red.

- Un host podra renovar la direccion IP que usa.
- Permitira reutilizar las direcciones (solo se sostendra direcciones mientras este conectado/prendido.
- Soporte a usuarios moviles que quieren unirse a la red.

Resumen de DHCP:

1. Host transmite “DHCP discover” msg[opcional]
2. Servidor DHCP responde con “DHCP offer” msg[opcional]
3. Host pide direccion IP “DHCP request” msg
4. Servidor DHCP envia direccion: “DHCP ack” msg

DHCP puede retornar mas que la direccion IP alojada en una subred:

- Direccion del enrutador del primer salto para el cliente
- Nombre y direccion IP del servidor DNS
- Mascara de red

Problema: ¿Como puede un host detras de NAT permitir pedidos de conexiones entrantes?
Solucion:

UPnP (Universal Plug and Play)

- Una aplicacion ejecutada en un host puede pedir un mapeo NAT entre su (IP privado, Port privado) y su (IP publico, Port publico)
- ¿Si se acepta el pedido y se crea el mapeo que consecuencias tiene?
    - Nodos de afuera pueden iniciar conexiones TCP con el (IP publico, Port publico) asignado
- ¿Como se pueden enterar maquinas de afuera de un servicio disponible por detras de NAT?
    - UPnP permite a la aplicacion conocer el valor de (IP publico, Port publico) de modo que la aplicacion lo puede avisar al mundo externo

• Ejercicio: tu host detrás de UPnP y NAT tiene dirección privada 10.0.0.1 y ejecuta BitTorrent en el puerto 3345. La dirección pública del NAT es 138.76.29.7. BitTorrent pide crear mapeo en NAT y se obtiene mapeo de (10.0.0.1, 3345) a (138.76.29.7, 5001). ¿Cómo se entera otro host ejecutando BitTorrent de la aplicación en (138.76.29.7, 5001)?
– La aplicación avisa al tracker que está disponible en (138.76.29.7, 5001).
– El host externo que ejecuta BitTorrent contacta el tracker y aprende que tu aplicación BitTorrent ejecuta en (138.76.29.7, 5001).
– El host externo puede enviar paquete SYN a (138.76.29.7, 5001).
– La caja NAT traduce esa dirección a (10.0.0.1, 3345) y luego se envía el paquete.
– Luego tu host contesta con otro SYN al host externo y se establece la conexión.

Situacion: En internet una maquina tiene una o mas direcciones IP; estas no pueden usarse para enviar paquetes porque el hardware de la capa de enlace de datos no entiende las direcciones de internet.

Problema: ¿Como se convierten direcciones IP en direcciones de Ethernet?

Solucion: Protocolo de resolucion de direcciones (ARP): el host de origen da salida a un paquete de difusion hacia ethernet preguntando ¿Quien posee una direccion IP w,x,y,z?

1. La difusion llegara a cada maquina en ethernet y cada una verificara su direccion IP
2. Al host de destino le bastara con responder con su direccion de ethernet E
3. Asi el host de origen aprender que la direccion IP de w.x.y.z esta en el host con la direccion de Ethernet E

OSPF (Open Shortest Path First)

Sistema Autonomo (SA): Consiste de un grupo de enrutadores bajo el mismo control administrativo.

OSPF es un protocolo de puerta de enlace interior (IGP)
Considera una adaptación del metodo de enrutamiento de estado de enlace.

Los protocolos de enrutamiento estudiados no son compatibles con IP por la forma de las tablas de enrutamiento que se usaban

¿Por qué estudiar OSPF?

- Porque OSPF introduce mejoras interesantes al protocolo de enrutamiento de estado de enlace:
  – Es compatible con IP.
  – En OSPF el modelo de grafo asociado a un SA es bastante más flexible que el usado para los protocolos de enrutamiento anteriores al considerar redes de distintos tipos.
  – Para permitir SA grandes OSPF organiza un SA como una jerarquía de niveles.
  – Con OSPF para un destino se puede considerar más de una línea de salida (cuando hay más de un camino óptimo) para balancear la carga en la red.

• Estas mejoras introducen problemas nuevos para diseñar un algoritmo de enrutamiento.

OSPF soporta:

- Lineas punto a punto entre 2 enrutadores
- Redes de multiacceso con difusion (p e.j LAN)
- Redes de multiacceso con muchos enrutadores, cada uno de los cuales se puede comunicar directamente con los otros.

Organizacion de un SA en OSPF:

- Divide los SA en areas numeradas
- Un area puede contener varias redes adentro de ella
- Cada enrutador esta configurado para conocer que otros enrutadores estan en su area
- Las areas no se translapan

Tipos de areas en un SA:

- Hay area que es red dorsal que tiene numero 0
- Hay areas que se conectan a la red dorsal
    - Se puede entrar desde un area en el SA a cualquier otra area en el SA mediante la red dorsal
- La topologia de la red dorsal no es visible fuera de esta.

Clasificación de los enrutadores de un SA:

- Enrutadores internos: yacen ocomplentamente dentro de un area
- Enrutadores dorsales: enrutadores en un area dorsal
- Enrutador de borde de area (EBA): ES parte de una red dorsal y a la vez de una o mas areas.
- Enrutador de borde de SA (EBSA):
    - Inyecta en el area rutas a destinos externos en otros SA

Un tipo de aviso de estado de enlace (AEE) contiene el costo de un enrutador a todos sus vecinos.
Los EBA resumen informacion de enrutamiento aprenida de un area, para hacerla disponible en sus AEE que envian a las otras areas

Un **EBA** recibe avisos de estado de enlace de todos los enrutadores de una de sus areas A y con esa informacion determina el costo de alcanzar cada LAN de A.

¿Como definir la informacion resumida de un area dorsal?

Por medio de un grafo donde

- Todos los arcos unen pares de EBA
- El peso de cada uno de estos arcos es el costo de camino optimo (en el area dorsal) que une el par de EBAs.

Al ejecutarse OSPF los enrutadores dentro de un area ejecutan una adaptación del protocolo de estado de enlace.

Cuando un enrutador se inicia, envia mensajes Hello a:

- Todas las lineas punto a punto
- Al grupo de todos los otros enrutadores de una LAN (si esta conectado a una LAN)

De las respuestas cada enrutador aprende quienes son sus vecinos

- Los enrutadores en la misma LAN son todos vecinos

¿Como se fijan los pesos de los enlaces?

OSPF no fija una politica de como los pesos de los enlaces son fijados, ese es el trabajo del administrador de la red.

Cada enrutador tiene base de datos de estado de enlace ***(BDEE)***

- La BDEE contiene todos los AEE
- La BDEE debe ser creada y luego mantenerse
- Dentro de un area cada enrutador debe tener el mismo grafo (BDEE) para construir la tabla de re-envio.

Tipos de paquetes usados para intercambio de informacion entre enrutadores adyacentes:

- Pauqete de actualizacion de estado de enlace (PAEE): para mandar AEE asociado al enrutador emisor. Estos AEE tienen numero de secuencia.
    - Usando dicho numero de secuencia el recpetor puede ver si un AEE es mas nuevo o mas viejo que el que ya tiene.
- Paquete de confirmacion de estado de enlace (PCEE): para confirmar los PAEE
- Paquete de descripcion de base de datos (PDBD): llevan resumen de la descripcion de todos los AEE de la BDEE del enrutador emisor
    - Osea numeros de secuencia de los AEE del enrutador emisor
    - El receptor puede determinar cuales AEE de ese grupo necesita, comparando numero de secuencia de un AEE con numero de secuencia de AEE (del mismo enrutador) que ya tiene.
- Paquete de pedido de estado de enlace (PPEE): se usan para solicitar AEEs.

¿Como actualizan BDEE los enrutadores?
Dos enrutadores vecinos deben sincronizar sus BDEE:

- Un vecino es el maestro y el otro es el esclavo
- El maestro controla el intercambio de de PDBD
- Se intercambia PDBD, PPEE, PAEE, PCEE para asegurar que ambos vecinos tienen igual información en sus BDEE.

Es ineficiente tener cada enrutador en una LAN que intercambie informacion con todos los otros enrutadores de la LAN, es por eso que un enrutador de la LAN se elige como enrutador designado.
El enrutador designado es quien intercambia mensajes con todos los enrutadores de la LAN mediante sincronización.

Usando inundacion cada enrutador informa a todos los otros enrutadores de su area de sus enlaces a todos los otros enrutadores y redes y el costo de esos enlaces.
Este intercambio se hace periodicemente o cuando una linea se cae, o cuando regresa o su costo cambia.

Para un enrutador R dentro de un area se puede ejecutar el algoritmo de Dijkstra.
Para esto usar la BDEE de R.
Dijkstra calcula el camino mas corto desde R a cualquier otro enrutador de su area y red en el SA entero.
Pero queremos que si hay varios caminos mas corto se pueda balancear la carga entre ellos, es por eso que OSPF recuerda el conjunto de caminos mas cortos entre dos nodos y durante el envio de paquetes el trafico se divide entre ellos.

- Para esto se usa una adpatacione special del algoritmo de Dijkstra que usa una cola de prioridades.

Para enviar un paquete de un area a una red de otra area:
1. El paquete viaja de su red (area) local al area dorsal
2. Luego cruza el area dorsal
3. Luego viaja del area dorsal a la red de destino

Recordar que para las tablas de re-envio se usa CIDR.

Interredes:

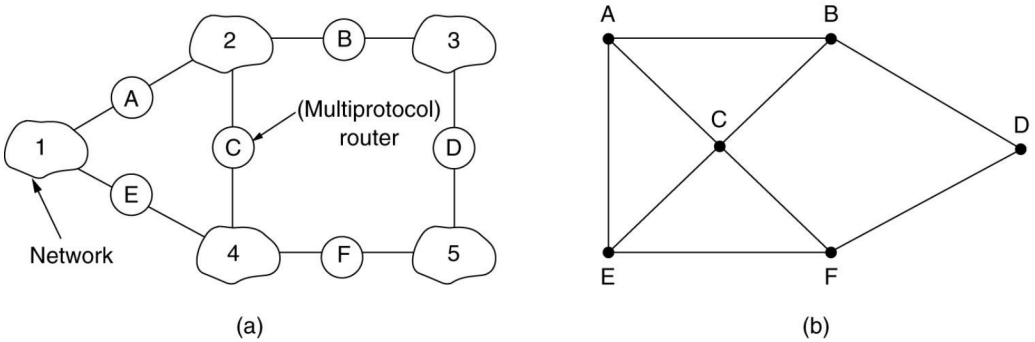
Tener diferentes redes implica tener diferentes protocolos
Enrutadores que pueden conectar dos redes de distinta tecnologia: enrutadores multiprotocolo (puertas de enlace).

¿Como se pueden enviar paquetes de una red a otra de distinta tecnologia?
Construir una capa arriba de las diferentes redes que oculet las diferencias entre las distintas redes (Cerf y Khan).

- IP provee un formato de paquete universal que
  - Todos los enrutadores multiprotocolo reconocen
  - Puede ser pasado a traves de casi toda red.

Problemas que surgen al pasar de una red a otra de tecnologia distinta

- Paquetes de una red de CVs deben transitar una red sin conexiones
- Con frecuencia se necesitaran conversiones de protocolo
- Se necesitaran conversiones de direcciones,
- Difrentes tamaños maximos de paquetes usados por diferentes redes.



Los nodos son enrutadores multiprotocolo
Un lado entre dos enrutadores multiprotocolo significa que esos enrutadores estan conectados via una subred.

Una vez construido el grafo de la interred, pueden aplicarse algoritmos de enrutamiento al grupo de enrutadores multiprotocolo

Organizacion de enrutamiento en 2 niveles:
- En cada red se utiliza un protocolo de puerta de enlace interior (IGP)
- Entre las redes se usa un protocolo de puerta de enlace exterior (EGP)

La red puede usar diferentes protocolos IGP, pero debe usarse el mismo protocolo EGP

Fragmentacion:

Cada red impone un tamaño maximo a sus paquetes.
Si se quiere enviar un paquete P a una red cuyo tamaño maximo de paquete es < P.
Las puertas de enlace dividen los paquetes en fragmentos, enviando cada fragmento como paquete de interred individual.
- Las redes tienen el problema de unir nuevamente los fragmentos.

Existen dos estrategias opuestas para re-combinar los fragmentos y recuperar el paquete original:
- Hacer transparente la fragmentación causada por una red de “paquete pequeño” (a las demas redes subsiguientes por las que debe pasar el paquete para llegar a su destino final).
    - Con este metodo la red de paquete pequeño tiene puertas de enlace (enrutadores especializados) que interactuan con otras redes.
    - Cuando un paquete de tamaño excesivo llega a una puera de enlace, esta lo divide en fragmentos
    - Todos los fragmentos se dirigen a la misma puerta de enlace de salida, donde se re-combinan las piezas.
    - Las redes ATM tienen hardware especial para esta estrategia.
    - Desventajas de este metodo
        - La puerta de enlace de salida debe saber cuando ha recibido todas las piezas, por lo que debe incluirse un campo de conteo o un bit de fin de paquete en cada paquete
        - Todos los paquetes deben salir por la misma puerta de enlace
            - Esto puede bajar un poco el desempeño
        - Hay una sobrecarga para re-ensamblar y volver a framentar repetidamente un paquete grande que pasa por varias redes de paquete pequeño.

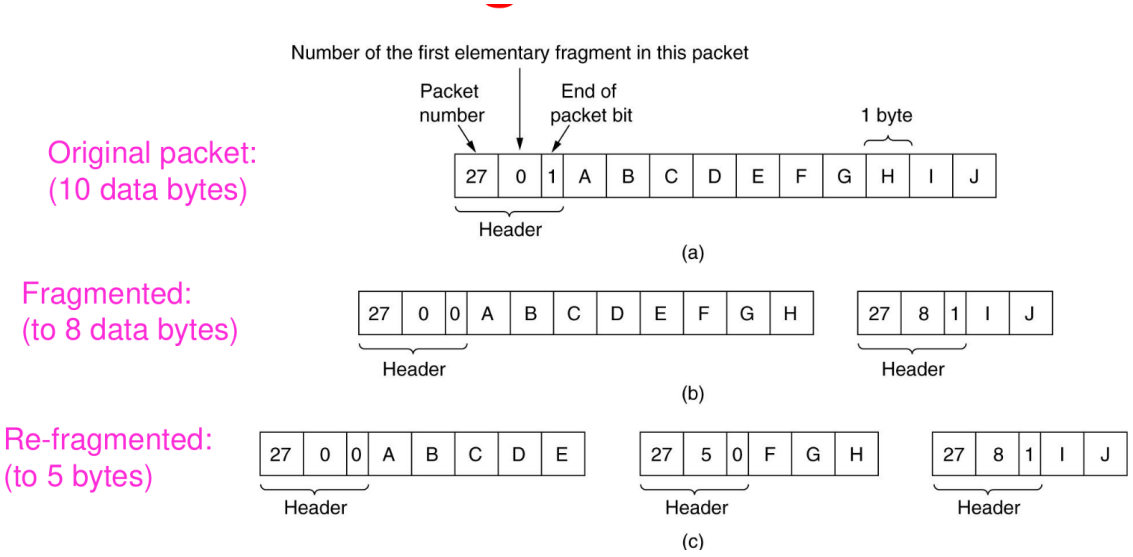
- Abstenerse de recombinar los fragmentos en las puertas de enlace intermedias
    - Una vez que se ha fragmentado un paquete, cada grafmento se trata como si fuera un pauqete original. Todos los pauqetes pasan por la puerta de enlace de salida
    - La recombinacion ocurre en el host de destino
    - IP funciona de este modo
    - Desventajas:
        - Requiere que todos los hosts sean capaces de hacer el re-ensamble
        - Al fragmentarse un paquete grande, aumenta la sobrecarga total, pues cada fragmento debe tener un encabezado.

Esquema de numeracion de fragmentos:

- El protocolo de interred define un tamaño de framento elemental.
  - Al fragmentarse un paquete todas las partes son iguales al tamaño de fragmento elemental, excepto la ultima que puede ser mas corta
- Para saber a que paquete pertenece un fragmento
  - Se numera el paquete original
- Para referirme a un fragmento puedo poner en el encabezado:
  - El desplazamiento del bit o byte inicial en el paquete original
- Para saber si vienen mas fragmentos
  - Poner un bit que indica si el fragmento es el ultimo del paquete original

osea

```
typedef struct fragment {
        void* content[ELEM_FRAGMENT_SIZE];
        int   original_packet_n;
        int   shift;
        byte  last;
}
```



Fragmentacion en IP:

- Enlaces de red tienen ***MTU (Tamaño maximo de transferencia)*** corresponde a trama de nivel de capa de enlace mas larga posible
  - Diferentes tipos de enlace tienen diferentes MTU
- Datagramas IP grandes son fragmentados dentro de la red
  - Un datagrama pasa a ser varios datagramas menores
  - Ellos son reensamblados solamente en el destino final
  - Bits de encabezados IP son usados para identificar y ordenar fragmentos relacionados.

- El campo de identificacion es necesario para que el host de destino determine a que datagrama pertenece un fragmento recien llegado.
    - Todos los fragmentos de un datagrama contienen el mismo valor en el campo de identificacion.

- **DF** de un bit significa cuando fijado en 1 una orden de no fragmentar (porque el destino es incapaz de juntar las piezas de nuevo)
- **MF** es un bit que significa mas fragmentos
    - Todos los fragmentos excepto el ultimo tienen establecido este bit, que es necesario para saber cuando han llegado todos los fragmentos de un datagrama.
- El desplazamiento del fragmento indica en que parte del datagrama actual va este fragmento.
    - Todos los fragmentos excepto el ultimo del datagrama deben tener un multiplo de 8 bytes que es la unidad de fragmentacion elemental.
    - Dado que se proporcionan 13 bits, puede haber un maximo de 8192 fragmentos por datagrama.

Entunelamiento

Se usa caundo un host de origen h1 y de destino h2 estan en la misma clase de red pero hay una diferente en medio.
En este metodo, los paquetes son encapsulados en la red del medio usando un encabezado de esta.

Protocolos de puerta de enlace exterior (EGP)

Es necesario estudiar protocolos de puerta de enlace exterior (PPEE) porque:

- Las tablas de reenvio deben permitir mandar mensajes entre maquinas conectadas a SA diferentes
    - El PPEE permite agregar informacion a ser usada con ese fin en las tablas de reenvio de los enrutadores
- El enrutamiento de PPEE se peocupa de establecer las rutas a usar (que pasan por diferentes SA) para permitir que se comuniquen maquinas pertenecientes a distintos SA.

Para enrutamiento inter SA encontrar un camino optimo es imposible en la practica ya que

- Cada SA corre su propio protocolo interno y usa cualquier esquema para asignar metricas a los caminos
- Por lo tanto es imposible calcular costos de caminos significativos para caminos que cruzan varios SA.

Como no se puede manejar informacion de caminos optimos ¿Que tipo de informacion sobre rutas manejar?

- El enrutameinto inter-SA permite:
    - Avisar alcanzabilidad de prefijos desde un SA
    - Considerar caminos formados por SAs para ir de un origen a un destino.

Requisitos de protocolos de puerta de enlace exterior:

- Para el enrutamiento es necesario encontrar algun camino de SAs para el destino deseado que es libre de ciclos.
- Ademas los caminos deben respetar las politicas de los SA a lo largo del camino

Una politica son reglas que se refieren a las preferencias de enrutamiento y a limitaciones de enrutamiento

Los PPEE suelen implementarse sobre enrutadores de borde de sistemas autonomos (EBSA)

Tareas que realiza un EBSA:

- Tienen que hacer una eleccion de varias rutas a un destino
- Va a elegir la mejor de acuerdo con sus propias politicas locales y este va a ser la ruta que avisa
- Le dice a sus vecinos el camino exacto que esta usando para cada destino.

Relaciones entre SA

Relacion proveedor-consumidor:

Supongamos que tenemos un PSI cliente (o PSI consumidor) y un PSI proveedor.

- El PSI cliente paga a PSI proveedor para entregar paquetes a otros destinos y recibir paquetes enviados de otros destinos
- ¿Que tipo de rutas publica el PSI proveedor? ¿Y el PSI consumidor?
    - El PSI proveedor debe dar publicidad de rutas a todos los destinos en internet al PSI cliente sobre el enlace que los conecta
        - Asi el PSI cliente va a tener rutas para enviar pauqetes para todos lados
    - El PSI cliente debe publicar rutas a los destinos en su red al PSI proveedor
        - Esto permite al PSI proveedor enviar trafico al PSI cliente solo para esas direcciones

Los PSI compañeros no se cobran por mandarse mensajes entre sus destinos
¿Que tipo de rutas publica un PSI a sus compañeros?

- Los SA compañeros mandan publicidad de enrutamiento de uno al otro para los destinos que residen en sus redes
- El compañerismo no es transitivo

Multihoming: Significa que un PSI esta conectado con varios PSI

BGP:

La naturaleza de los PPEE es muy distinta a la de los protocolos de enrutamiento de puerta de enlace interior, lo cual lleva a que surjan problemas no considerados antes de resolver.

Problema: Construir un PPEE para internet
Solucion: BGP (Border Gateway Protocol) es el PPEE de facto que usa internet

Tareas que realiza BGP:
BGP provee a cada SA un medio para:

- Obtener informacion de alcanzabilidad de subredes desde SA vecinos.
- Propagar la informacion de alcanzabilidad a todos los enrutadores dentro del SA.
- Determinar buenas rutas a las subredes basandose en la informacion de alcanzabilidad y en las politicas del SA.
- BGP permite a cada subred publicar su existencia al resto de la internet
    - BGP se asegura que todos los SA de la internet conozcan acerca de la subred y como llegar alli.

BGP permite a cada SA aprender cuales destinos son alcanzables via sus SA vecinos
En BGP los destinos son prefijos, donde cada prefijo representa una subred o una coleccion de subredes (definida usando agregacion de prefijos CIDR)

Si un SA s avisa un prefijo a un SA s1, s esta prometiendo que va a enviar datagramas a ese prefijo.
En BGP un SA es identificado por un numero globalmente unico llamado numero de sistema autonomo (ASN)

Cuando un enrutador avisa de un prefijo a lo largo de una sesion BGP incluye con el prefijo una ruta que pasa por vaior SA para llegar al prefijo.

- Una ruta se compone de un prefijo mas atributos BGP.
- Algunos atributos importantes:
    - AS-PATH: contiene los Sa por los cuales el aviso del prefijo ha pasado. Cuando un prefijo pasa por un SA, el SA agrega su ASN al atributo AS-PATH.
    - NEXT-HOP: es el IP de la interfaz del enrutador que comienza el AS-PATH hacia el destino.

¿Como hacer para propagar informacion de rutas en BGP?:
En BGP pares de enrutadores intercambian informacion de rutas sobre conexiones TCP semipermanentes usando el puerto 179.

- Hay tipicamente una conexion BGP TCP para
    - Cada enlace que conecta directametne dos enrutadores EBSA (o enrutadores BGP) en dos diferentes
    - Para enlaces entre enrutadores dentro del SA
- Para cada conexion TCP, los 2 enrutadores al final de la conexion se llaman compañeros BGP.
    - Los compañeros BGP se avisan rutas.

Sesiones BGP:

- La conexion TCP con todos los mensajes BGP enviados por la conexion se llama sesion BGP
- Una sesion BGP entre enrutadores de dos SA se llama sesion externa BGP (eBGP)
- Una sesion BGP entre enrutadores en el mismo SA es llama sesion interna BGP (iBGP)
- Las lineas de sesiones BGP no siempre se corresponden ocn los enlaces fisicos

Aviso de rutas:

Cuando una puerta de enlace P recibe rutas:

- P usa las sesiones iBGP para distribuir las rutas a los otros enrutadores del SA de P
- Las sesiones iBGP se usan para disribuir rutas a los enrutadores dentro del SA.

Cuando un enrutador (puerta de enlace o no) sabe cual es la mejor ruta de que dispone hacia un nuevo prefijo, puede querer crear una entrada para el prefijo en su tabla de reenvio.

Atributos BGP:

Donde puede guardar un enrutador las rutas con que trabaja?:

- Usar una base de información de enrutamiento (BIE)
    - Es un repositorio donde se colocan esas rutas.

¿Que tipos de mensajes se pueden usar para el aviso de rutas?:
Se usan ***Mensajes de actualizacion,*** los cuales comunican dos tipos de informacion:

- Informacion acerca de una ruta a traves de la internet
    - La misma esta disponible para ser agregada en la BIE de todo enrutador BGP receptor
- Una lista de rutas previamente avisadas por el enrutador emisor que ya no son mas validas
- Un mensaje de actualizacion puede contener ambos o uno de estos tipos de informacion.

Uso por los enrutadores de los atributos AS-PATH:

- Para detectar y prevenir ciclos: si un enrutador ve que su SA esta contenido en la lista del camino, entonces va a rechazar el aviso.
- PAra elegir entre varios caminos al mismo prefijo

Política de importación

Un enrutador BGP no tiene obligacion de avisar una ruta a un destino
¿Si un enrutador BGP recibe una aviso de ruta, tiene que colocar el aviso de esa ruta en la tabla de reenvío?

- No siempre va a querer hacerlo.
- Porque puede que todas las rutas a un prefijo no cumplan con las políticas fijadas por el SA.

¿Que hace una puerta de enlace cuando recibe un aviso de ruta?

- Cuando una puerta de enlace recibe un aviso de ruta usa su politica de importacion para decidir si aceptar o filtrar la rta.
    - La politica de importacion puede filtrar una ruta porque el SA puede no querer enviar trafico por uno de los SA en la ruta del AS-PATH
    - La puerta de enlace puede filtrar la ruta porque ya conoce una ruta preferible al mismo prefijo.

Cuando los enlaces fallan y las politicas cambian, los enrutadores deben cambiar sus BIE para que respeten la nueva topoligia o las nuevas politicas

Para cumplir con ese requisito:
Los enrutadores BGP neceistan poder cancelar caminos previamente avisados

Implementacion:
Se usa un aviso conocido como ruta removida, para esto se usan los mensajes de actualizacion.

Situacion: Un enrutador puede recibir multiples rutas al mismo prefijo.
La mejor ruta debe guardarse en la BIE
Problema: Como escoge el enrutador una de esas rutas al mismo prefijo?

Solucion: SI hay mas de una ruta al mismo prefijo, BGP secuencialmente invoca las siguientes reglas de eliminacion hasta que queda una ruta:

1. LAs rutas con el mayor valor de preferencia local son elegidas
    a. A las rutas con el mayor valor de preferencia local que puede haber sido fijado por el enrutador o aprendido de otro enrutador en el mismo SA
2. De las rutas restantes, la ruta con el camino AS-PATH mas corto es eleiga (la metrica es la cantidad de saltos SA)
3. De las rutas restantes la ruta con el enrutador NEXT-HOP mas cercano es elegida
    a. osea se considera el enrutador NEXT-HOP con el camino mas corto determinado por el algoritmo de enrutamiento intra-SA (a esto se lo llama hot potato routing)
4. SI queda mas de una ruta, se usan criterios adicionales.

Capa de enlace

Los canales de comunicacion

- Cometen errores ocasionales
- Tienen una tasa de datos finita
- Hay retardo de propagacion

Por eso nuestra **meta necesaria** es lograr una comunicacion confiable y eficiente entre dos maquinas adyacentes, osea conectadas por un canal de comunicaciones.

¿Como cumplir con este requisito?:
Definimos una capa debajo de la capa de red que se encargue de esto.
Adivina como se llama esa capa…, **Capa de enlace de datos (CED)**

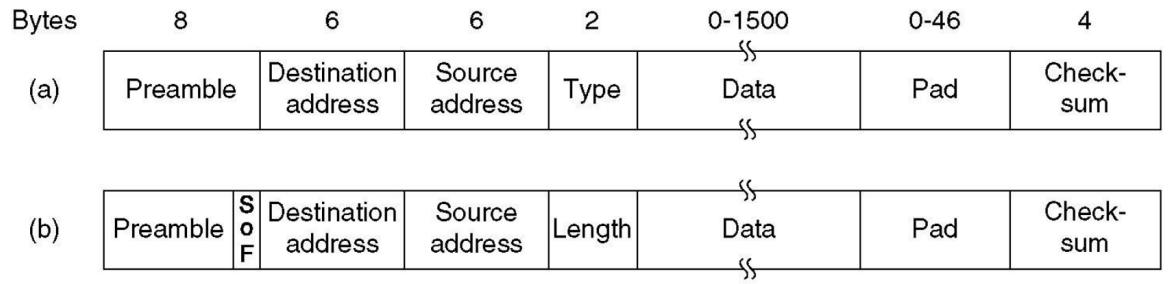
Funciones de la CED:

- Control de flujo: evitar que emisor rapido sature a receptor lento
    - Uso de protocolos de tuberia
- Entramado:
    - En el canal de difusion solo hay un stream de bits ¿Como detectar inicio y fin de cada trama?

        - Usualmente se usa patron especial de bits para ello (llamado bandera)
            - Igual no lo vamos a estudiar, no se para que lo nombran
- Deteccion y correccion de errores
- Manejo de colisiones:
    - Ocurren en canales de difusion usados por varias maquinas
    - Cuando dos maquinas intentan transmitir tramas al mismo tiempo ocurre colision

Informaciones que deberia contener una trama de capa de enlace de datos:

- Encabezado: suele contener
    - Direcciones del origen
    - Direcciones de destino
    - A veces la longitud de la trama
- Campo de carga util (El contenido que se quiere enviar)
- Terminador final (para control de errores)



Formato de trama Ethernet

Fundamentos en la comunicacion en la capa de enlace de datos:

Se trabaja con:

- Confiramciones de recepcion de tramas
- Temporizacion de reenvio
- Retransmisiones de tramas (perdidas o dañadas)
- Uso de numeros de secuencia en las tramas (para identificar tramas duplicadas)
- Llevar a caballito (piggybacking) para aprovechar mejor el canal de comunicaciones
- Control de flujo (para evitar que el emisor sature al receptor mas lento) bo back N, recepticion selectiva

Necesidad de canales de difusion

Situacion: Es costoso e incomodo hacer que todo par de maquinas de una organizacion estan conectadas directamente entre si por dos canales (dedicados exclusivamente para ellas).

- Si hah n maquinas daria un n*(n-1) conexiones

Hay que encontrar una manera mas economica para conectar varias maquinas entre si.

Solucion: Usar **Canales de difusion**

- En un canal de difusion estan conectadas varias maquinas que quieren transmitir tramas por el canal
- Si una maquina envia un mensaje, todas las demas lo reciben

Tipos de canales de difusion:
- Inalambricos
    - En su forma mas simple las maquinas se comunican entre si sin uso de cables
- Cableados
    - Las maquinas se comunican entre si por medio de cables

Control de colisiones:
Si dos tramas se tranmsiten en forma simultanea en un canal de difusion, se traslapan en el tiempo y la señal resultante se altera, este evento se llama colision.

Para evitarlas, definimos una subcapa de la capa de enlace de datos que se encargue del control de colisiones
- Esta subcapa de la CED se llama subcapa de control de acceso al medio (SCAM)
- La subcapa MAC es una subcapa inferior a la CED

En una red de difusion el asunto clave es como determinar quien puede usar el canal cuando hay competencia por el.

Protocolos de acceso multiple (PAM): Se usan para determinar quien sigue en un canal de difusion

SCAM

Para LANs cableadas:

- Modelo de estaciones
    - Hay N estaciones independientes que genera tramas para transmision
    - Una vez generada una trama, la estacion se bloquea hasta que la trama se ha generado con exito
- Suposicion de canal unico
    - Hay un solo canal disponible donde todas las estaciones pueden transmitir y recibir

Fenomenos que suceden en un canal que una estacion podria detectar:
- Detectar que el canal esta en uso (osea alguna estacion esta enviado una trama)
- Detectar que hay una colision en el canal

En las LAN actuales cada estacion puede deteectar si el canal esta en uso
- En realidad detecta si estan llegando bits de alguna trama a la maquina hace la deteccion
- Los protocolos que pueden hacer esto se llaman Protocolos de deteccion de portadora (CSMA)

- Ventaja de poder hacer deteccion de portadora:
    - Se evita generar colision poniendo tramas en el canal cuando estan llegando bits de alguna trama

En las LAN actuales cada estacion puede detectar si esta ocurriendo una colision cuando esta transmitiendo una trama.

- Para detectar colisiones:
    - El hardware de una estacion escucha el cable mientra transmite
    - Si lo que lee es distinto de lo que puso en el, sabe que esta ocurriendo una colision

¿Que se hace si una estacion que esta transmitiendo una trama detecta una colision?:

- No tiene sentido seguir enviando la trama
- Por lo tanto es mejor que las estaciones aborten sus transmisiones tan pronto como detecten una colision

PAM en Ethernet:

Estudiaremos el PAM CSMA/CD (Acceso Multiple con deteccion de portadora y deteccion de colisiones)

En CSMA/CD, el emisor:

1. Antes de transmitir una trama detecta la portadora
2. Si el canal esta libre transmite
3. Sino espera hasta que el canal se desocupe para transmitir
4. SI el eemisor detecta una colision, aborta la tranmsiion, espera un tiempo aleatorio y una vez que paso ese tiempo: goto 1.

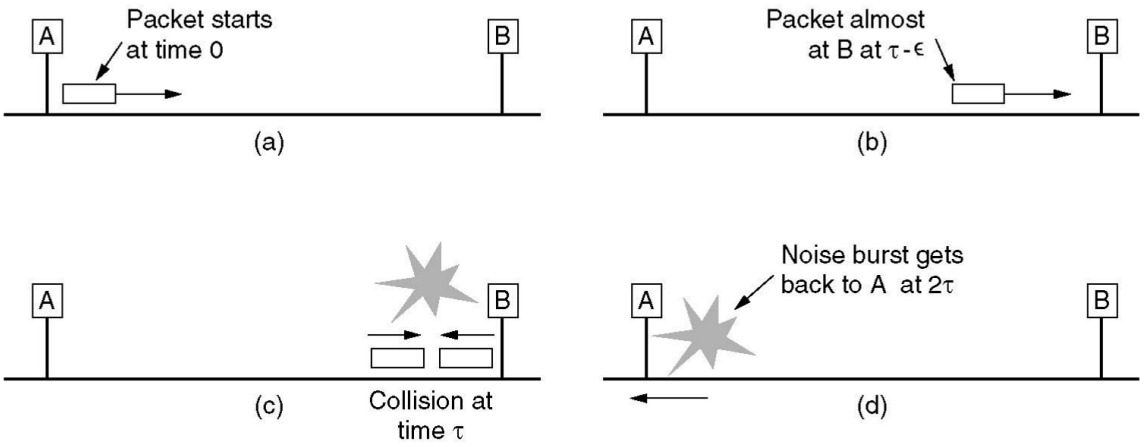
En CSMA/CD el receptor:

1. Recibe una trama buena si no hubo colision el medio no cometio errores
2. En caso contrario (hubo colision o el medio cometio errores) recibira una trama dañada la cual sera descartada
3. Al mandar una confirmacion de recepcion hace los pasos del emisor

Se dice que una estacion ha tomado el canal cuando todas las demas estaciones sabian que estaba transmitiendo y no interfirieron.
#¿Como sabe una estacion que ha tomado el canal?
¿Si dos estaciones comienzan a transmitir en momento t = 0, en cuanto tiempo se daran cuenta de que ha habido una colision?

- El minimo tiempo en detectar la colision es el tiempo que tarda la señal para propagarse de una estacion a otra.

¿Cual seria el peor caso de demora?



a. A comienza a transmitir en t = 0
b. En τ - ϵ un instante antes de que la señal llegue a B, B comienza a transmitir
c. B detecta la colision casi de inmediato y se detiene
    i. En Ethernet se genera rafaga de ruido de 48 bits
d. La rafaga de ruido causada por la colision no regresa a A hasta pasados 2τ - ϵ

¿Las tramas pueden ser tan chcias como uno quiera?
Si una estacion E intenta transmitir una trama demasiado corta y ocurre una colision

- La transmision de E se completa antes de que la rafaga de ruido llegue de regreso en el momento 2τ
- El emisor entonces supondra incorrectamente que la trama se envio cone xito.

Para evitar que la situacion anterior ocurra, las tramas deberan tardar mas que 2τ para enviarse, de manera que la transmision aun este llevandose a cabo cuando la rafaga de ruido regrese al emisor.

Ethernet

Ethernet e IEEE 802.3 son casi identicos, por eso usamos estos dos terminos indistintamente.

Algunos asutnos de la CED se pueden hacer por hardware (Entramado, control de errores, deteccion de portadora, deteccion de colisiones)

¿Que componentes de hardware conviene tener para estos asuntos?

- Transceptor: maneja deteccion de protadora y deteccion de colisioens
- Tarjeta controladora
    - Ensambla los datos en el formato de trama adecuado
    - Calcula terminador de las tramas de salida
    - Comprueba las tramas de entrada (e.g deteccion de errores)

Cada cableado de Ethernet tiene una longitud maxima de cable por segmento

A medida que se va proapagando una señal por un cable, se va debilitando, llega un punto en el cual ya no puede seguir con su camino ¿Como hacer para que la señal pueda viajar mucho mas alla de ese punto?

Usar **Repetidores:**

- Un repetidor es un dispositivo de capa fisica que recibe, amplifica (regenera) y retransmite señales en ambas direcciones.
- Los repetidores introducen un retardo.

Restriccion de Ethernet: Puede haber multiples segmentos de cable y multiples repetidores, pero ningun par de transceptores puede estar separado por mas de 2,5 km y ninguna reuta entre dos transcpetores puiede atravesar mas de 4 repetirdores

Para diseñar una red de mayor velocidad:
Si aumentamos la velocidad de la red y la longitud maxima del cable permanece igual, la longitud minima de trama tambien debe aumentar

Si aumenta la velocidad de red y la longitud de trama minima no cambia, la longitu maxima del cable debe disminuir, de manera proporcional.

Situacion: a medida que se agregan mas y mas estaciones a Ethernet, aumenta el trafico. En algun momento la LAN se saturara.

¿Como evitar esto?

Usamos **Ethernet Conmutada:**

- Un **conmutador (switch)** contiene una matriz de conmutacion de alta velocidad y de 4 a 32 tarjetas de linea
- Cada tarjeta de linea contiene de 1 a 8 conectores
- Hay matrices de conmutacion que funcionan a mas de 1 Gbps

  Tarea realizada por un conmutador:
  - Almacenamiento y reenvio de tramas de Ethernet.

  Transparencia: Los hosts no son conscientes de la presencia de conmutadores
  Los conmutadores no necesitan ser administrados, aprender por si solos.

Si dos máquinas conectadas a la misma tarjeta de conexión transmiten tramas al mismo tiempo:

- Si todos los puertos de la tarjeta forman una LAN local dentro de la tarjeta,
  - Las colisiones en esta LAN en tarjeta se detectan y manejan igual que en una red CSMA/CD.
  - Las tarjetas pueden estar transmitiendo en paralelo.
- Si cada puerto de entrada se almacena en un búfer,
  - Todos los puertos de entrada reciben y transmiten tramas al mismo tiempo, para una operación en paralelo duplex.
  - Cada puerto es un dominio de colisión independiente.

Cada conmutador tiene una tabla de conmutador <direccion MAC del host, interfaz para alcanzar el host, estampilla de tiempo>

Un conmutador aprende cuales hosts pueden ser alcanzados a traves de cuales interfaces

- Cuando el conmutador recibe una trama, registra el par emisor/localizacion en la tabla del conmutador

Filtrado y reenvio de tramas en conmutadores:

1. Registrar enlace de ingreso, dirección MAC del host emisor de la trama.
    a. Identificación de la interfaz del destino:
2. Se Busca en la tabla del conmutador la dirección MAC del destino.
3. if se encuentra la entrada para el destino
   then {
   if el destino está en el segmento por el cual vino la trama
   then descartar trama
   else enviar trama en la interfaz indicada por la entrada
   }
   else inundar /* enviar en todas las interfaces excepto aquella por la que llegó la trama */

- La CED toma de la CR paquetes y los encapsula en tramas
- Las tramas tienen una longitud maxima impuesta
- Cada paquete de la CR se divide en tramas
- En la CR de la maquina de origen hay un proceso que entrega bits a la CED para transmitirlos a la maquina de destino
- El trabajo de la CED es transmitir los bits a la maquina de destino para que puedan ser entregados a su CR.

Flujo entre enrutadores

1. Al llegar trama a enrutador: el hardware verifica si esta libre de errores
2. La CED comprueba si esta la trama esperada y de ser asi, entrega el paquete dentro de la trama al **software de enrutamiento**
3. El software de enrutamiento elije la linea de salida adecuada y entrega el paquete a la CED para enviarlo

Si una trama no se entrego, el emisor la reenvia, como se implementa esta idea?

Regresar tramas de control con confirmacion de recepcion positivas o negativas de las tramas que llagan.
La verdad que todo esto es igual que como se manejan los ACK en capa de transporte, usan timer y toda la pavada
Tambien tienen numeros de secuencia para controlar que no se envien tramas repetidas.

Tambien tiene control de flujo, en el que si un emisor esta mandando tramas a una velocidad mayor que la que el receptor pueda soportar, el receptor controla el flujo

Tambien usa piggybacking

Todo esto es lo mismo por que es basicamente TCP.

No entendi lo de ack independiente (filmina 9)

PAM: ALOHA puro

En ALOHA puro
El emisor:

- Transmite cuando tiene datos para enviar
- Escucha el canal por un tiempo igual a la demora de propagacion de ida y vuelta maxima en la red + un incremento fijo de tiempo
- Si se escucha un ack en ese tiempo, todo anduvo bien.
- Sino se espera un tiempo aleatorio y la trama se manda de nuevo
- SI se falla en recibir un ack luego de varias retransmisiones se tira la toalla

El receptor:

- Al recibir una trama chequea su validez y si lo es, inmediatamente manda un ack
- Si la trama es invalida el receptor la ignora
    - La trama puede ser invalida por ruido o por colision

Evaluación de ALOHA puro:

- El método ALOHA puro bajo carga baja es eficiente y tiene una demora baja.
- En ALOHA puro una estación no escucha el canal antes de transmitir; esto generará probablemente muchas colisiones.
- Como el número de colisiones crece rápidamente a medida que aumenta la carga, la máxima utilización del canal alrededor del 18%.

CSMA persistente 1

Protocolo CSMA persistente-1 para el emisor:

- Si una estacion tiene datos por enviar, primero escucha el canal para saber si otra esta transmitiendo en ese momento
- Si el canal esta ocupado, entonces la estacion espera hasta que se desocupe
- Cuando la estacion detecta un canal inactivo, transmite una trama
- Si ocure una colision, la estacion espera una cantidad aleatoria de tiempo y comienza de nuevo
- Comportamiento luego que el emisor envio una trama
- La estacion espera un tiempo razonable por un ACK
    - Teniendo en cuenta el tiempo de propagacion de ida y vuelta maximo en la red y el hecho que la estacion receptora tambien debe competir por el canal para responder
- Si no recibe ack en ese tiempo, la estacion espera una cantidad aleatoria de tiempo y comienza de nuevo

Para el receptor:

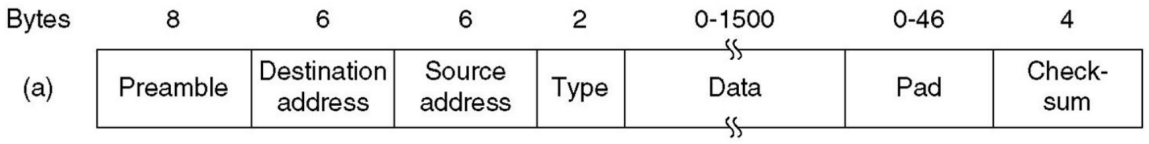
- Al recibir una trama chequea su validez y si lo es, inmediatamente manda un ack
- Si la trama es invalida el receptor la ignora
    - La trama puede ser invalida por ruido o por colision

El retardo de propagación tiene un efecto importante en el desempeño de CSMA persistente 1

- Caso de que justo despues de que una estacion comienza a transmitir, otra estacion esta lista para enviar
- Si la señal de la primera estacion no ha llegado aun a la segunda, esta ultima detectara un canal inactivo y comenzara a enviar tambien
    - Esto producira una colision
- Cuanto mayor sea el tiempo de propagacion, mas importante sea este efecot.

Aun si el retardo de propagacion es cero, habra colisiones
Supongamos que dos estaciones quiener enviar y detectan que una tercera esta transmitiendo.
Luego que la tercera termine de transmitir, las dos estaciones que quieren enviar detectaran un canal intactivo, por lo tanto enviaran y se producira una colision.

Volvemos a Ethernet (No se por que)



Trama DIX Ethernet (Dec, Intel, Xerox)

Preambulo de 8 bytes, cada uno es 10101010

Direcciones:

- Se usan direcciones de 6 bytes
- Se escriben como 6 pares de digitos hexadecimales separados por -
    - P ej: 1A-23-F9-CD-06-9B
- El bit de orden mayor de la direccion de destino es 0 para las direcciones ordinarias y de 1 para las direcciones de grupo
- Una trama que consiste unicamente de bits 1 en el campo de destino se acepta en todas las estaciones de la red (broadcasting)

Campo Tipo:

- Uso de multiples protocolos de CR a la vez en la misma maquina
- El kernel debe saber a cual entregarle la info de la trama que llego
- El campo de tipo indica al receptor a que proceso entregarle la trama

Longitud de trama minima:

- Las tramas deben tener al menos 64 bytes de largo, de la direccion de destino a la suma de verificacion
- Cuando los datos mas el encabezado ocupan menos de 64 bytes:
    - Cuando la porcion de datos de una trama es menor a 46 bytes: Uso del campo de relleno (para alcanzar los 64 B)
- Suma de verificación:
    - Tiene 32 bits de largo
    - Se usa metodo de deteccion de errores llamado codigo polinomial

Cuando IEE estandarizo la Ethernet hizo los siguientes cambios al formato DIX:

- Reducir el preambulo a 7 bytes y usar el ultimo byte para un delimitador de inicio de trama
- Cambiar el campo de Tipo por un campo de longitud
- Poner un pequeño encabezado a los datos para dar informacion de tipo.

Capa Física (CF)

Proposito: Transportar un stream de datos de una maquina a otra usando medios fisicos

La CF no consiste solo de medios fisicos, sino que los medios fisicos se conectan etre si usando dispositivos como codecs, modems, multiplexores, demultiplexores, conmutadores, puentes, enrutadores, puertas de enlace, etc. Formandose asi redes complejas de distitntos tipos.

Tipos de informacion de la comunicacion de datos en la CF:

- Señales analogicas
    - Caracterizadas por una ufncion matematica continua
- Señales digitales:
    - Con conjunto fijo de niveles validos.

Ondas sinusoidales:

Onda sinusoidal: $s(t) = A \sin(2\pi ft + \varphi)$, t numero real

Las ondas sinusoidales son producidas por fenomenos naturales, por ejemplo los tonos audibles

Propiedades:

- Frecuencia: Numero de oscilaciones por segundo
- Amplitud: diferencia entre las alturas maxima y minima
- Fase: cuanto es desplazado el comienzo de la onda sinusoidal a partir de un tiempo de referencia

Periodo(t): Tiempo requerido por un ciclo
Frecuencia: 1/T

Los sistemas de comunicacion usan altas frecuencias (expresadas en millones de ciclos por segundo - megahertz (MHz)

Hay distintas maneras de representar graficamente las señales:
- Representacion de dominio de tiempo
- Representacion de dominio de frecuencia
    - Grafo de dominio de frecuencia
        - Muestra conjunto ode ondas sinusoidales simples que consituyen la funcion compuesta
        - A sin (2pift) es representada por una linea simple de altura A que se posiciona en x = f

Espectro de una señal: Rango de frecuencias que contiene
Ancho de banda analogica: Ancho de espectro

Las señales digitales usan voltajes para representar valores digitales
- Mecanismos de transmision fisicos usan dos o mas niveles de voltaje para enviar señales digitales
    - Cada nivel representa un numero binario
- Usar 2^n niveles para representar un numero de n bits

¿Como se determina la cantidad de bits por segundo de una señal digital?
- La respuesta depdende de
    - El numero de niveles de señal
    - La cantidad de tiempo que el sistema permanece en un nivel dado antes de moverse al siguiente

El hardware coloca limites en cuan corto el tiempo en un nivel debe ser
- Si la señal no peramanece en un nivel por suficiente tiempo, el hardware receptor va a fallar en decetarlo
- La cantidad de veces que una señal puede cambiar por segundo se mide en **baudios**
- Si se requiere que la señal permanezca en un nivel por 0.001 segundos, decimos que el sistema opera a 1000 baud

baud y numero de niveles de señal controlan la tasa de bits

Si se tiene 2 niveles de señal y opera a 1000 baud, el sistema puede transferir exactamente 1000 bps
Si se tiene 4 niveles de señal y opera a 1000 baud, el sistema puede transferir 2000 bps

Relacion entre baudios, niveles de señal y tasa de bits
bits por segundo = Nº baudios * [log2(niveles)]

En algunos casos se introduce un filtro en el circuito para limitar la cantidad de ancho de banda disponible para cada cliente.

¿Como estimar la tasa de datos maximal de un canal?
Situacion: tenemos un canal de comunicaciones y queremos saber cual es la tasa maxima de datos que el canal permite

Solucion (Teorema de Nyquist): Si se pasa una señal a traves de un filtro pasa-bajas de ancho de banda H, la señal filtrada se puede reconstruir por completo tomando solo 2H muestras por sec.

Si la señal consiste de V niveles de voltaje, el teorema de Nyquist establece:
Tasa de datos maxima = 2H log2 V bps

Situacion: Va a existir un V maximo que permite enviar señales y para V mayores el ruido termico va a dañar las señales.

Consecuencia: No sabemos cuales son los valores de V permitidos

La cantidad de ruido termico se mide por la relacion entre la potencia de la señal y la potencia del ruido, llamada relacion señal a ruido

S = Potencia de señal
N = Potencia de ruido
Relacion señal a ruido = S / N

La relacion misma no se expresa; en su lugar se da la cantidad 10 Log*10* S/N

Estas unidades se conocen como decibeles (dB)

Problema: ¿Como calcular la tasa de datos maxima de un canal teniendo en cuenta el ruido termico y como calcular el V maximo permitido?

Solucion: Usar el metodo de **Shanon**

Resultado de Shannon: La tasa de datos maxima de un canal ruidoso cuyo ancho de banda es H Hz y cuya relacion señal ruido es S/N esta dada por:
Nº maximo de bps = H log2 (1+ S/N)
(La formula solo da un limite superior y los sistemas reales rara vez lo alcanzan)

Modems

Situacion: Las computadoras trabajan con señales digitales y la red telefonica usa comunicacion analogica.

Problema: ¿Como hacer para mandar mensajes de una computadora por la red telefonica?
Solucion: Usar un **Modem**

- Un modem permite convertir señales digitales a analogicas y reciprocamente
- Todos los modems modernos transmiten trafico en ambas direcciones al mismo tiempo (mediante el uso de frecuencias distintas para las diferentes direcciones)

Muchos sistemas de comunicacion de larga distancia usan una portadora (carrier) de onda sinusoidal

Modulacion:
El enviador debe cambiar una de las caracteristicas de la onda: amplitud, frecuencia, desplazamiento de fase.

Portadora de onda senoidal = tono continuo en el rango de 1000 a 2000 Hz
Modulacion de amplitud: Se usan dos niveles difrerentes de amplitud para representar 0 y 1
Modulacion de frecuencia:

- Se usan dos o mas tonos diferentes.
- Si la señal es mas fuerte, la frecuencia del carrier aumenta y si la señal es mas debil, la frecuencia del carrier disminuye
- Es mas facil de visualizar

Desplazamiento de fase (DF):

- Es posible usar cambios en la fase para representar una señal
- ¿Como se mide DF?
    - Por el angulo de cambio

Modulacion de fase:
– Ejemplo: la onda portadora se desplaza de modo sistemático de 0 a 180 grados a intervalos espaciados de manera uniforme (a esto se le llama BPSK).
– Ejemplo: Otro esquema es usar desplazamiento de 45, 135, 225, o 315 grados para transmitir 2b de información por intervalo.

- Al requerir el DF al final de cada intervalo, se facilita que el receptor reconozca los límites de los intervalos.

Un receptor puede medir la cantidad de portadora desplazada durante un DF:

- Sistema que reconoce un conjunto de DF y usa cada DF para representar valores de datos especificos.

Diagramas de constelacion:

Situacion: Los modems avaanzados usan una combinación de tecnicas de modulacion para trasnmitir muchos bits or baudio.

- Con frecuencia se combinan multiples amplitudes y varios DF
- Osea se tiene un conjunto de amplitudes CA y un conjunto de desplazamientos de fase CDF
- Cada combinacion es un par: (amplitud, desplazamiento de fase) Є CA x CDF

- Si hay 2^n combinaciones, entonces cada combinacion representa un numero binario de n bits

Problema: ¿Como se pueden representar graficamente las combinaciones de modulacion usadas por un modem?

Solucion: Usar diagramas de constelacion
- Distancia del origen refleja amplitud
- Angulo refleja DF
- Cada estandar de modem tiene su propio diagrama de constelacion y se puede comunicar solamente con otros modems que utilicen el mismo modelo

Multiplexado:

Situacion: Desde el punto de vista economico, es mucho mas conveniente usar un solo cable para transportar varias señales que isntalar un cable para cada señal
Requisito: Queremos que los canales de comunicacion puedan ser compartidos por multiples señales.
Problema: ¿Como hacer para poner muchas señales en un mismo canal?
Solucion: Usar multiplexores y demultiplexores

FDM (Multiplexado por division de frecuencia)
TDM (Multiplexado por division de tiempo)
CDM(Multiplexado por division de codigo)

Principio de superposición de ondas:

Propiedades fisicas de la interferencia: Si dos señales en un punto estan en fase se agregan para sumas sus amplitudes, pero si estan fuera de fase, se restan para dar una señal que es la diferencia de amplitudes.

Situacion: Tenemos varios circuitos analogicos, cada uno con su señal analogica.
- Queremos todas esas señales analogicas en un mismo canal

Problema: ¿Como hacer para multiplexar y demultiplexar un conjunto de señales analogicas?
Solucion: CDM (Multiplexado por division de codigo):
Permite varias señales de diferentes usuarios compartir la misma banda de frecuencias.
- Varios usuarios pueden coexisitr y transmitir simultaneamente con interferencia minima
- A CDM a menudo de lo llama CDMA (Code Division Multiple Access)

En CDMA las tramas que colisionan no son distorsionadas; en cambio, se agregan multiples señales en forma lineal.
Esto es debido al principio de superposición de ondas.
En CDMA cada tiempo de bit se subdivide en m intervalos cortos llamados **chips**
Hay 64 o 128 chips por bit

A cada estacion se le asigna un codigo unico de m bits llamado secuencia de chips

Notacion bipolar: El 0 binario es -1 y el 1 binario es +1.

Transmisión en un tiempo de bit
- Una estación puede transmitir un 1 enviando su secuencia de chips en bipolar,
- Puede transmitir un 0 enviando su negativo de su secuencia de chips (i.e. se cambia el signo de cada componente de su secuencia de chips en bipolar), o
- Puede quedarse en silencio y no transmitir nada.

S = vector de m chips para la estacion S
S` = negacion

Problema: ¿Como hacer par auqe un receptor pueda recuperar la señal enviada por una estacion de manera sencilla?
Solucion: Todas las secuencias de chips deben ser ortogonales dos a dos.

Para recuperar el flujo de bits de una estación, el receptor.
- Calcula el producto interno normalizado de la secuencia de chips recibida y la secuencia de chips de la estación cuyo flujo de bits se está tratando de recuperar.
- Si la secuencia de chips recibida es S y el receptor está tratando de escuchar una estación cuya secuencia de chips es C, simplemente calcula S•C.

Las secuencias ortogonales de chips para las estaciones se pueden generar utilizando un metodo conocido como codigo de Walsh

Propiedades:

- Si **S•T** = 0, entonces **S•$\underline{\mathbf{T}}$** = 0.
- El producto normalizado de cualquier secuencia de chips por si mismo es 1.

$$\mathbf{S}\bullet\mathbf{S} = \frac{1}{m}\sum_{i=1}^{m} S_i S_i = \frac{1}{m}\sum_{i=1}^{m} S_i^2 = \frac{1}{m}\sum_{i=1}^{m} (\pm 1)^2 = 1$$

- Además **S•$\underline{\mathbf{S}}$** = -1.

Señales

- **Periodicas:**
  - s(t+T) = s(t) para todo -∞ < t < ∞
- **Aperiodicas:**

Si una señal compuesta es periodica, entonces las partes constitutivas son tambien periodicas

Codificación de bits en Ethernet

Codificacion Manchester:

- Cada paeriodo de bit se divide en dos intervalos iguales
- Un bit 1 se envia teninedo un voltaje alto en el primer intervalo y bajo durante el segundo
- Un 0 binario es justo lo inverso: primero bajo y despues alto

Codificacion Manchester Diferencial:

- Un bit 1 se indica mediante la ausencia de una transicion al inicio del intervalo
- Un bit 0 se indica mediante la presencia de una transicion al inicio del intervalo
- En ambos casos, tambien hay una transicion a la mitad.

La señal alta es de 0.85 voltios y la baja de -0.85 voltios.

100BASE-FX (fast-ethernet):

- 2 lineas de fibra optica: una para recpecion (RX) y la otra para transmitir (TX)
- La distancia entre una estacion y el conmutador se de hasta 2 km.
- Los cables 100BaseFX deben conectarse a conmutadores
    - Los concentradores no estan permitidos con 100Base-fx
- La codificacion es mediante el esquema 4B/5B BRZI
- Cada 4 bits de datos son codificados en un simbolo con 5 bits de codigo, tal que cada bit de codigo contiene un simple elemento de señal. El bloque de codigo de 5 bits se llama grupo de codigo.
- Para asegurar sincronización cada bit de codigo del stream de 4B/5B es tratado como un valor binario y codificado asi: un bit 1 se representa con una transicion al comienzo del intervalo de bit y un 0 se representa con ninguna transicion al comienzo del intervalo de bit.
- Cada grupo de 5 periodos de reloj da 32 combinaciones, las 16 primeras se usan para transmitir numeros entre 0 y 15. Algunos de los 16 valores restantes se usan para control, como el marcado de limites de tramas.
- Una transicion esta presente al menos 2 veces para cada 5-code. No mas de 3 ceros son permitidos en un 5-code.

Señales digitales vs señales analogicas:

- Las señales digitales generalmente son mas baratas que las señales analogicas y son menos susceptibles a interferencia de ruido.
- Las señales digitales sufren mas de atenuacion (reduccion de fuerza de la señal) que las señales analogicas.

A frecuencias mayores los pulsos se tornan mas redondeados y pequeños

Multiplexado por division de frecuencia

En OFMD (Orthogonal Frequency Division Multiplexing)

- El ancho de banda del canal es dividido en varias portadoras que independientemente envian datos
- Estas portadoras son empaquetadas juntas en el dominio de frecuencias, de modo que las señales de cada portadora se extienden a las adyacentes
- Sin embargo, como se ve en la Fig. 2-26, la respuesta de cada portadora es diseñada de modo que es cero en el centro de las portadoras adyacentes.