Capa de enlace de datos:

Las Funciones de la capa de enlace de datos son:

1) Proporciona una buena interfaz con la *capa de red*, para brindarle servicios.
2) Recibe de la *capa de red* paquetes y los encapsula en *tramas* para mejor procesamiento (*Entramado*).
3) Maneja errores de transmisión.
4) Regula el flujo de datos.

Para cumplir con estas metas la capa de enlace de datos toma de la capa de red los paquetes y los encapsula en tramas para transmitirlos. Cada trama contiene un encabezado, un campo de carga útil para almacenar el paquete y un terminador. El manejo de las tramas es la tarea primordial de la capa de enlace de datos.

Servicios proporcionados a la capa de red

El servicio principal es transferir datos de la *capa de red* en la maquina origen a la *capa de red* en la maquina destino. La transmisión real se realiza en la capa física. Pero es más fácil pensar en términos de 2 (dos) procesos de la *capa de enlace* de datos, que se comunican usando un protocolo de enlace de datos.

Cada *trama* contiene un *encabezado*, un *campo de carga útil* para almacenar el paquete y un *terminador* o *final*.

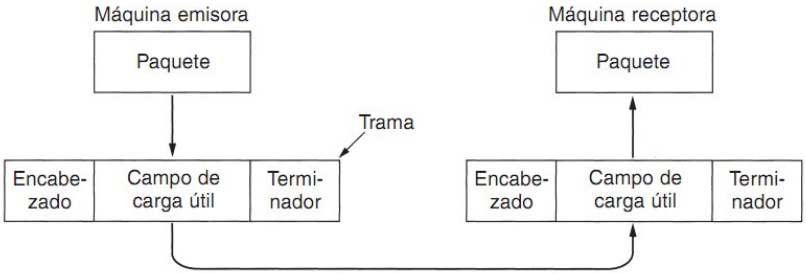


Figura 3-1. Relación entre los paquetes y las tramas.

Servicios proporcionados a la capa de red:
La función de la capa de enlace de datos es suministrar servicios a la capa de red. El servicio principal es transferir datos de la capa de red en la máquina de origen a la capa de red en la máquina de destino. En la capa de red de la máquina de origen hay una entidad llamada proceso que entrega algunos bits a la capa de enlace de datos para transmitirlos a la maquina destino. El trabajo de la capa de enlace de dato es transmitir los bits a la maquina de destino para que puedan ser entregados a su capa de red.

La capa de enlace de datos puede ofrecer distintos servicios:
a) *Servicio no orientado a la conexión no confiable*: La maquina de origen envía tramas independientes a la maquina destino sin pedir que esta confirme la recepción. No se establece ninguna conexión de antemano, ni se libera después. Este tipo de servicios es apropiado cuando la tasa de errores es baja.
b) *Servicio no orientado a la conexión confiable*: No se establece la conexión de antemano, pero se confirma de manera individual cada trama enviada.
c) *Servicio orientado a la conexión confiable*: Se establece una conexión entre las maquinas de origen y destino antes de transmitir los datos. Cada trama enviada esta numerada (números de secuencia) y la capa de enlace garantiza que cada trama llegará a destino solo una vez, y en orden adecuado.

Entramado

Técnica que consiste en dividir el flujo de bits que llegan de la capa de red en tramas separadas. Resuelve el problema de marcar el inicio y el final de cada trama.
Veremos 4 métodos:
a) *Conteo de caracteres*: Se trata de un campo en el encabezado para especificar el número de caracteres en la trama. Cuando la capa de enlace de la maquina destino ve la cuenta de caracteres sabe cuantos le siguen y por ende donde finaliza dicha trama.
OJO: Esta cuenta puede alterarse por un error de transmisión y así el receptor perdería la sincronía y sería incapaz de localizar el inicio de la siguiente trama.
b) *Bandera, con relleno de bytes*: Cada trama inicia y termina con bytes especiales (*banderas = FLAGS*). Dos banderas continuas indican el fin de una e inicio de la siguiente. Puede ocurrir que estas banderas (*patrón de 8 bits*) aparezcan en los datos, e interferir en el entramado. Como solución se agrega un byte de escape (*ESC*) justo antes de cada bandera accidental en los datos. Entonces una bandera de entramado puede distinguirse de una en los datos por la ausencia o presencia del byte *ESC* que la antecede. Si *ESC* es también un dato, nuevamente se rellena con un byte *ESC*. Esta técnica se llama *relleno de caracteres*.
c) *Bandera, con relleno de bits*: Presenta el mismo problema que el orientado al byte con la diferencia de que en este las tramas pueden dimensionarse mas chicas y tomar tamaños diferentes a potencias de 2. Cada trama comienza y termina con “01111110”. Cada vez que la *capa de enlace* emisora encuentra un patrón de 5 unos consecutivos en los datos inserta un bit 0 en el flujo de bit saliente.
Estas técnicas de relleno de bytes o bits son transparentes a la *capa de red*.
d) *Consiste en una violación de la capa física*: violar los 1 y 0 para saber cuando empieza y termina una trama.

Control de errores

¿Cómo asegurar la entrega confiable de datos al receptor y en forma ordenada?
La manera normal de asegurar la entrega confiable de datos es proporcionar realimentación al emisor sobre lo que ocurre del otro lado de la línea. Normalmente se trata de una confirmación de recepción positiva o negativa de una trama.
Casos:
- Puede que una trama desaparezca por completo por lo que el receptor (o emisor en caso de ser de confirmación) se quedará esperando eternamente. Como solución a esto se introducen ***temporizadores*** en la capa de enlace de datos**.** Cada trama, cuando se envía, el emisor inicia un *temporizador*, que se ajusta de modo que expire cuando haya transcurrido un intervalo suficiente para que la trama llegue a destino, se procesa y regresa la confirmación de recepción al emisor.
- Si la trama enviada llega correctamente al receptor, la acepta, la pasa *a la capa de red* y envía la confirmación de recepción, y se pierde.
-Si la trama o la confirmación de recepción se pierden, el temporizador expirara alertando al emisor sobre un problema potencial. SOLUCION: transmitir de nuevo la trama. Sin embargo, aunque las tramas pueden transmitirse muchas veces existe el peligro de que el receptor acepta la misma trama dos o más veces y que la pase a la capa de red más de una vez. SOLUCION: Es necesario asignar números de secuencia a las tramas que salen a fin de que el receptor pueda distinguir las retransmisiones de los originales.

Control de flujo

Sirve para controlar que un emisor rápido no sature de datos a un receptor lento. Aunque las tramas lleguen sin error, el receptor no será capaz de manejar tramas conforme lleguen

y comenzará a perder algunas.

Métodos de control de flujo:

a) *Control de flujo basado en retroalimentación*: El receptor regresa información al emisor autorizándolo para enviar más datos o indicándole su estado. El puede decirle cuantas tramas puede enviarle, antes de enviarle una confirmación de recepción.
b) *Control de flujo basado en tasa*: El protocolo tiene un mecanismo integrado que limita la tasa a la que el emisor puede transmitir los datos.
Detección y corrección de errores

Los errores en una transmisión pueden aparecer en ráfaga o de manera independiente. Los diseñadores de redes han diseñado 2 (dos) estrategias principales para manejar errores:

a) Incluir suficiente información redundante en cada bloque de dato transmitido para que el receptor pueda deducir lo que debió ser el carácter transmitido. Esta técnica usa código de corrección de errores. Entre ellos podemos nombrar el *código de Hamming*.
b) Incluir solo suficiente redundancia para permitir que el receptor sepa que ha ocurrido un error y solicite una retransmisión. Esta técnica usa código de detección de errores. Entre ellos: *Suma de verificación – CRC*; *Paridad*.

Suponemos entonces que una trama tiene ***m*** bits de datos y ***r*** bits de redundancia o verificación. La unidad de ***n = m + r*** bits se llama *palabra codificada*.
Distancia de Hamming: cantidad de posiciones en que difieren 2 (dos) palabras codificadas.

Si ***d => p + 1*** entonces se puede detectar un error de peso "p"

Si ***d => 2p + 1*** entonces se puede corregir p dígitos.

Ejemplo:

La Distancia Hamming entre 1011101 y 1001001 es 2.

La Distancia Hamming entre 2143896 y 2233796 es 3.

La Distancia Hamming entre "tener" y "reses" es 3.

Las propiedades de detección y corrección de errores dependen de su *Distancia de Hamming*:

- *Para detectar **d** errores*: Se necesita un código con distancia ***p + 1***. O sea, agregarle 1 bit de redundancia.

Ejemplo: Bit de paridad -> se envía 1011010, agregando 1 bit de paridad:
+ *Par*: 10110100
+ *Impar*: 10110101

Un código con 1 bit de paridad tiene una distancia de 2, por lo que cualquier error de 1 bit produce una palabra con paridad equivocada.

- *Para corregir **d** errores*: Se necesita un código con distancia ***2p + 1***.

Ejemplo: Sea el código de 4 palabras validas: 0000000000; 0000011111; 1111100000; 1111111111. → *Distancia de Hamming = 5*. Entonces, puede corregir errores de dos dígitos, ***5 = 2p + 1 → p = 2***.

Código de corrección de errores Hamming

Este código puede corregir errores de 1 bit. Los bits de la palabra codificada se numeran en forma consecutiva comenzando por el 1 a la izquierda. Los bits que son potencia de 2 son de verificación y el resto se rellena con los ***m*** bits de datos. Cada bit de verificación obliga a que la paridad de un grupo de bits, incluyéndose, sea par (o impar). Un bit puede estar incluido en varios cálculos de paridad.
Para ver a que bit de verificación contribuye el bit de datos en la posición k, escriba k como

suma de potencias de 1.
Ejemplo: Sea el dato *1001000* → 3=1+2; 5=1+4; 6=2+4; 7=1+2+4; 9=1+8; 10=2+8; 11=1+2+8.
Entonces, p1=3,5,7,9,11; p2=3,6,7,10,11; p3=5,6,7; p4=9,10,11.

Cuando llega una palabra codificada al receptor, este inicializa a 0 un contador y luego examina cada bit de verificación para ver si tiene paridad correcta, sino suma k al contador. Al finalizar, si el contador es *0=palabra valida*; sino, error en el bit k.

Para corregir errores de ráfaga: Se dispone como matriz una secuencia de k palabras codificadas consecutivas con 1 por fila, y transmitiendo los datos por columnas comenzando por el extremo izquierdo. Cuando todos los k bits han sido enviados, se envía la 2da. columna y así sucesivamente. Si ocurre un error en ráfaga de longitud k como mucho, se habrá afectado 1 bit de las k palabras, así que puede restaurarse la totalidad del bloque.

Código de detección de errores

- *Suma de verificación*: Consiste en agrupar el mensaje a transmitir en tramas de longitud determinada y asociar cada cadena con un numero entero (decimal). Después se suma el valor de todas esas tramas y se agrega el resultado al mensaje a transmitir pero cambiado de signo. Finalmente el receptor suma todas las cadenas/tramas, y si el resultado es 0, no hay error.
- *Código polinomial o CRC*: Código que usa la aritmética modular para detectar mayor cantidad de errores. Esta basado en el tratamiento de cadenas de bits como polinomios con coeficientes 0 y 1. Una trama de k
- bits se considera como la lista de coeficientes del polinomio con k términos que va desde x^{k-1} a x^0; tal polinomio es de grado $k - 1$. El emisor y receptor deben acordar por adelantado un polinomio generador que tiene como propiedad minimizar la redundancia.
  Cálculos que realiza el equipo transmisor para calcular su CRC:
  + Agrega tantos ceros a la derecha del mensaje original como el grado del polinomio generador, generando un nuevo polinomio.
  + Divide el mensaje "nuevo" entre el polinomio generador.
  + El resto que se obtiene se suma al mensaje "nuevo".
  + Se envía el resultado obtenido.
  El equipo receptor debe comprobar el código CRC para ver si no hubo errores:
  + Divide el código recibido entre el polinomio generador y comprueba el resto. Si es 0, es correcto, sino reenvía el mensaje.

Protocolos elementales de enlace de datos

Supuestos implícitos del modelo de comunicaciones

- En las capas física, de enlace y de red hay procesos independientes que se comunican pasando mensajes de un lado a otro.
- La maquina A desea mandar un flujo de datos a B usando un servicio confiable orientado a la conexión. B también quiere mandar a A simultáneamente.
- La capa de red esta siempre disponible.
- Nunca se entrega a la capa de red el encabezado de una trama para mantener completamente separados el protocolo de red y el de enlace de datos.
- Formato de una trama (*Frame Relay*):
  + *Kind*: Indica si hay datos en la trama.
  + *Seq, Ack*: Numero de secuencia y confirmación de recepción respectivamente.

+ *Info* (*de una trama de datos*): Contiene solo un paquete.

- *Relación paquete-trama*: La capa de red construye un paquete tomando un mensaje de la trama de transporte, y agregándole el encabezado de la capa de red. Este paquete se pasa a la capa de enlace de datos para incluirlo en el campo info de una trama saliente. Cuando este llega a destino la capa de enlace extrae de ella el paquete y luego lo pasa a la cada de red.
- *La capa de red puede habilitarse y deshabilitarse*: Cuando la capa de enlace habilita la capa de red, esta tiene permitido interrumpir cuando tenga que enviar un paquete, de lo contrario no puede hacerlo (si no hay espacio en el buffer emisor, evita que se sature).
- Los números de secuencia van de 0 a *MAX_SEQ* y avanzan circularmente.

Protocolo 1

Un protocolo Simplex sin restricciones – UTOPIA

Supone transmisión Simplex(en una sola dirección), absoluta disponibilidad de emisores y receptores, espacio infinito de búferes (receptores/emisores), ignoramos el tiempo de procesamiento y canal de comunicación libre de errores (no se usan números de secuencias ni confirmaciones de recepción).
Consiste en 2 (dos) procesos emisor y receptor que se ejecutan en la capa de enlace de la maquina de origen y destino respectivamente.
Emisor: Este esta en un ciclo while infinito que solo envía datos a la línea tan rápidamente como puede. *Cuerpo del while*
- 1º Obtiene un paquete de la capa de red.
- 2º Construye una trama de salida usando las variables y envía la trama al destino. Solo usa el campo info del frame ya que no hay restricciones de control ni flujo.
Receptor:
- 1º Espera que llegue una trama “sin daños”.
- 2º En algún momento la trama llega. Esta se elimina del buffer de hardware y se coloca en la variable ***r***, donde el código receptor pueda obtenerla.
- 3º Parte de los datos se pasa a la capa de red y la capa de enlace se retira para esperar la siguiente trama.

Protocolo 2
Protocolo Simplex de parada y espera:

Los mismos supuestos que el protocolo anterior pero con espacio finito de buffer del receptor y capacidad finita de procesamiento de datos.
Con este protocolo se trata de evitar que el emisor sature de datos al receptor. La solución es hacer que el receptor proporcione retroalimentación al emisor. Tras haber pasado a su capa de red el receptor regresa al emisor una pequeña trama ficticia, que de hecho autoriza al emisor para transmitir la trama siguiente.
NOTA: Aunque el tráfico de datos es Simplex, las tramas viajan en ambas direcciones. Este protocolo implica una alternancia estricta de flujo: 1º emisor, 2º receptor; 1º emisor, 2º receptor; 1º emisor, 2º receptor, etc. por lo que seria suficiente con un canal Semiduplex.
Funciona igual que el protocolo 1, solo que ahora el emisor deberá esperar a recibir la trama de confirmación para poder obtener el siguiente paquete a enviar.

Protocolo 3
Protocolo Simplex para un canal con ruido

Canal de comunicación normal que comete errores. Consideremos 2 (dos) casos:

- Si la trama esta dañada, el hardware detecta cuando calcula la suma de verificación (CRC).
- Si la trama esta dañada pero pese a ello la suma de verificación es correcta, el protocolo puede fallar.

Usamos el protocolo 2 con un temporizador. Cada trama, cuando se envía, el emisor inicia un temporizador, que se ajusta de modo que expire cuando haya transcurrido un intervalo suficiente para que la trama llegue a destino, se procese y regrese la confirmación de recepción al emisor. Si la confirmación de recepción no llega antes de que este expire el emisor la retransmite.
Puede ocurrir que la trama de confirmación se pierda, y el dato, ya fue aceptado y entregado a la capa de red, entonces expirará el temporizador y se reenviará otra vez la trama llegando duplicada a la capa de red.
Lo que hace este protocolo es distinguir entre una trama que está viendo por primera vez y una retransmisión. Solucionamos esto poniendo un numero de secuencia en el encabezado de cada trama que envía la maquina emisora, y en cada instante de tiempo, el receptor esperará un numero de secuencia determinado. Basta con un numero de secuencia de 1 bit (0 o 1) ya que la única ambigüedad es entre una trama y su antecesor, o sucesor inmediato, ya que se supone que si ***m + 2*** llego confirmada, también llego ***m + 1*** y ***m***.

Funcionamiento: Cuando el emisor transmite una trama arranca el temporizador. Existen 2 posibilidades:

- *Llega la trama de confirmación sin daño*: Entonces el emisor obtiene el siguiente paquete a la capa de red y lo coloca en el buffer del emisor sobrescribiendo el anterior y avanzando 1 el número de secuencia.
- *Llega la trama de confirmación dañada*: Entonces no cambia nada, se envía el duplicado.
- *Expira el temporizador*: Igual a anterior.
- *Cuando llega una trama valida al receptor*: Su numero de secuencia se verifica para ver si es duplicado, si no lo es se acepta, se pasa a la capa de red y se genera la confirmación de recepción.

Diferencia con los protocolos 1 y 2: Tanto el emisor como el receptor tienen una variable cuyo valor se recuerda mientras la capa de enlace esta en estado de espera. El emisor recuerda el número de secuencia de la siguiente trama a enviar, y el receptor el de la siguiente trama esperada.

Protocolos de ventana corrediza

Se usa un solo circuito para enviar datos en ambas direcciones.
Soluciones/mejoras a los problemas que este acarrea:

- Las tramas de datos de A a B se mezclan con las confirmaciones de recepción de A a B. El receptor puede saber si la trama es de datos o de confirmación analizando el campo *Kind* en el encabezado de una trama.
- Superposición (retardo de las confirmaciones). Consiste en anexar la confirmación de recepción a la trama de datos de salida, usando el campo *Ack* (ocupa pocos bits) de cada trama. De esta manera la confirmación viaja gratuitamente en la siguiente trama de datos de salida.

Ventajas:
+ Mejor aprovechamiento del ancho de banda.
+ Enviar menos tramas implica menos interrupciones y menos segmentos de buffer en el receptor.

Se agrega un temporizador para manejar el tiempo que debe esperar un paquete a la capa de enlace para superponer la confirmación. Si llega rápidamente un nuevo paquete, la

confirmación de recepción se superpone a el; si este expira antes la capa de enlace manda una trama de confirmación de recepción independiente.

Funcionamiento general: En cualquier instante el emisor mantiene un grupo de números de secuencias que corresponde a las tramas que tiene pendiente enviar. Estas tramas caen dentro de la ventana emisora y dentro de la ventana receptora las tramas que tiene permitido aceptar.

NOTA: En todos los protocolos de ventana corrediza, cada trama contiene un número de secuencia que va de 0 hasta $2^n - 1$, donde n es la cantidad de bits del nº de secuencia. Un tamaño de ventana receptora de 1, significa que la capa de enlace solo acepta tramas en orden, pero con ventanas más grandes no es así. Pero OJO, la capa de red siempre recibe los datos en orden correcto.

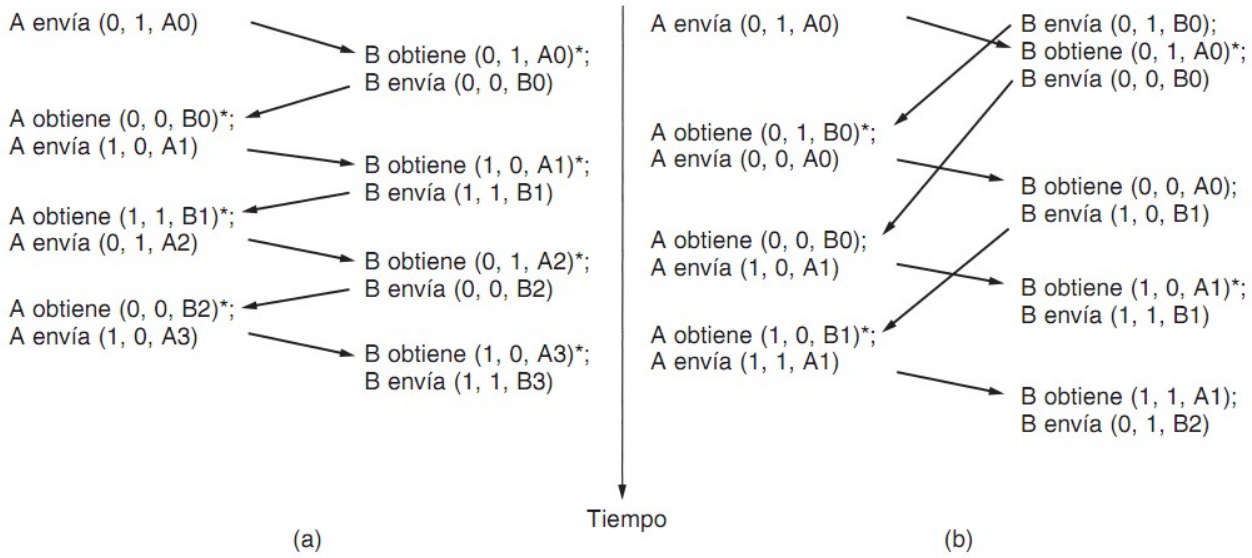
Protocolo 4

Un protocolo de ventana corrediza de 1 bit

Tamaño de ventana de 1 bit. Este protocolo usa parada y espera ya que envía una trama, y espera su confirmación antes de transmitir la siguiente. Presenta 2 (dos) casos:

- Una de las 2 (dos) capas de enlace comienza a transmitir: La maquina que arranca obtiene el primer paquete de la capa de red, construye la trama y la envía. Llega a la capa de enlace de la maquina receptora, revisa el numero de secuencia. Si es la esperada se pasa a la capa de red y la ventana del receptor se recorre hacia arriba. El campo de confirmación contiene el número de la última trama recibida sin error. Si es igual al número de secuencia de la trama que esta tratando de enviar el emisor sabe que esta ya se ha enviado y que puede obtener el siguiente paquete de su capa de red.
- Si ambas maquinas (A y B) comienzan a transmitir de manera simultanea, la mitad de las tramas contienen duplicados, aun cuando no hay errores de transmisión.

Como las tramas que están en la ventana del emisor pueden perderse o dañarse en transito, el emisor debe mantener todas estas tramas en memoria para su posible retransmisión. Entonces, si el tamaño máximo de la ventana es ***n***, el emisor necesita ***n*** buffer para contener las tramas sin confirmación. Si la ventana llega a crecer hasta su máximo, la capa de enlace de datos emisora deberá hacer que la capa de red se detenga hasta que se libere otro buffer.
El siguiente esquema muestra cada uno.
Notación (secuencia, confirmación de recepción, nº de paquete), el asterisco (*) indica que la capa de red acepta el paquete.



Protocolo que usa retroceso N

La suposición de que el tiempo de transmisión mas el necesario para su confirmación es insignificante, es falsa.
Este protocolo consiste en permitir que el emisor envíe hasta ***w*** tramas antes de bloquearse. Eligiendo correctamente el ***w***, el emisor podría transmitir tramas continuamente durante un tiempo constante sin llenar la ventana.
El producto del ancho de banda y el retardo indica cual es la capacidad del canal, el emisor necesita la capacidad de poder llenarlo sin detenerse para poder funcionar con una eficiencia máxima.
Si la capacidad del canal es de ***b*** bps, el tamaño de la trama de ***l*** bits y el tiempo de propagación de ***R*** segundos, entonces el tiempo requerido para transmitir una sola trama es de ***l/b*** segundos.
En parada y espera, la línea esta ocupada durante ***l/b*** segundos e inactiva durante ***R*** segundos.
Esta técnica se llama *canalización*.

¿Como controlar errores?

Protocolo 5

Funcionamiento: El receptor descarta todas las tramas subsecuentes sin enviar confirmación de recepción para las tramas descartadas. Es decir, la capa de enlace se niega a aceptar cualquier trama excepto la siguiente que debe entregar a la capa de red (Ventana de recetor de tamaño 1). Si la ventana del emisor se llena antes que el tiempo expire (se perdió una trama), el canal comenzará a vaciarse. Cuando el emisor termina de “esperar” retransmite todas las tramas cuya recepción no se haya confirmado, comenzando por la dañada.

Protocolo 6

Protocolo que utiliza repetición selectiva

Funcionamiento: Se descarta la trama dañada pero las tramas subsecuentes a esta se almacenan en el buffer del receptor (ventana del receptor mayor a 1). Cuando el emisor termina, solo la última trama sin confirmación se retransmite. Si llega correctamente, el

receptor puede entregar a la capa de red en secuencia todas las tramas que ha almacenado en el buffer.
Esta técnica se combina con el hecho de que el receptor envíe una confirmación de recepción negativa (*NACK*) cuando detecta un error, estas estimulan la retransmisión antes de que el temporizador expire y mejoran el rendimiento. *NACK* acelera la retransmisión de una trama especifica.
Este protocolo acepta tramas en desorden y pasa los paquetes en orden a la capa de red. Cada trama pendiente tiene un temporizador asociado, cuando este expira, a diferencia del protocolo 5, solo retransmite esa trama, no todas las pendientes.
Nota: El tamaño máximo de la ventana debe ser como mínimo la mitad del intervalo de los números de secuencia, ***(MAX_SEQ + 1)/2***.

Ejemplo de protocolos de enlace de datos

HDLC – *Control de enlace de datos de alto nivel*

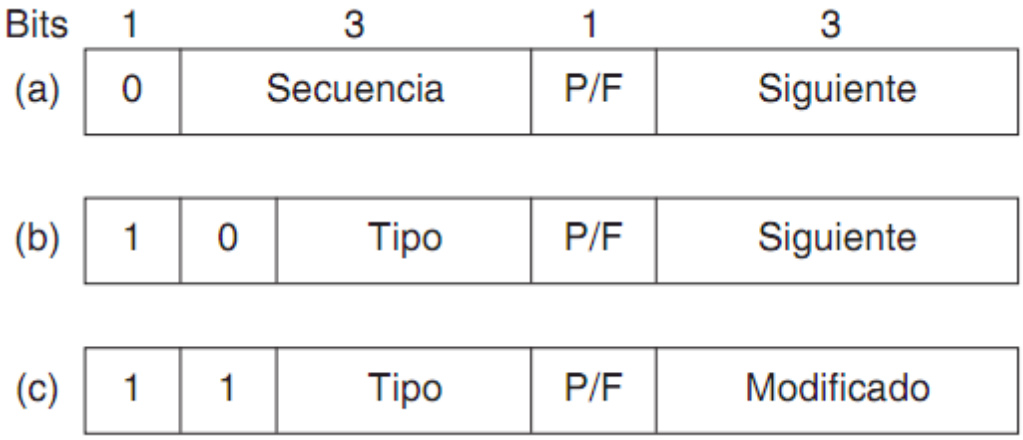
Este protocolo, todos sus antecesores y sucesores (*SDLC → ADCCP → HDLC → LAP → LAPB*) se basan en un mismo principio: todos son orientados al bit y usan el relleno de bit para lograr la transferencia de los datos. Este emplea una ventana corrediza con un número de secuencia de 3 bits, lo que significa que en cualquier momento pueden estar pendientes hasta 7 tramas sin confirmación de recepción.
Todos los protocolos orientados al bit usan esta estructura de la trama:

Bits	8	8	8	≥0	16	8
	0 1 1 1 1 1 1 0	Dirección	Control	Datos	Suma de verificación	0 1 1 1 1 1 1 0

Figura 3-24. Formato de trama para protocolos orientados a bits.

a) La trama esta delimitada por ***Banderas*** “01111110”. La trama minima contiene 3 campos y un total de 32 bits y excluye banderas a ambos lados.
b) El campo ***Dirección*** es de importancia primordial en líneas con múltiples terminales porque sirve para indicar una de ellas. En las líneas punto a punto a veces se usa para distinguir comandos de las respuestas.
c) El campo de ***Control*** se usa para números de secuencia, confirmación de recepción y otros.
d) El campo ***Datos*** puede contener cualquier información y una longitud arbitraria. A mayor tamaño, menor eficiencia en la suma de verificación.
e) El campo de ***Suma de Verificación*** es un código de redundancia cíclica.

Hay 3 tipos de tramas: De ***información***, de ***supervisión*** y ***no numeradas***.



Campo de control de trama de información

El campo *secuencia* es el numero de secuencia de la trama. El campo siguiente es una confirmación de recepción superpuesta. El bit *P/F* significa *sondeo/final*; en algunos protocolos sirve para obligar a la otra maquina a enviar de inmediato una trama de supervisión en lugar de esperar el trafico de regreso, el cual superpone la información de la ventana.

Campo de control de una trama de supervisión

Se distinguen diferentes tipos por el campo "Tipo"

- Tipo 0: "Receive ready". Trama de confirmación de recepción que sirve para indicar la trama esperada. Se usa cuando no hay tráfico de regreso.
- Tipo 1: "Reject". Trama de confirmación de recepción negativa para indicar que se detectó un error y el campo siguiente indica la primera trama en la secuencia que no se ha recibido de forma correcta.
- Tipo 2: "Receive not ready" (receptor no listo). Reconoce todas las tramas hasta determinado momento pero sin incluir siguiente, le dice al emisor que detenga el envío.
- Tipo 3: "Selective reject". Solicita la retransmisión de solo la trama especificada.
-

Campo de control de una trama no numerada

Usada a veces para propósitos de control o para llevar datos cuando se solicita un servicio no confiable sin conexión.
Las tramas de control pueden perderse o dañarse igual que las de datos, por lo que también debe confirmarse su recepción. Para esto se usa una trama de control especial llamada *UA* (confirmación *de recepción no numerada*).
Solo puede estar pendiente una trama de control.

La capa de enlace de datos en Internet

Internet consiste en un conjunto de maquinas individuales (*host y enrutadores*) y la infraestructura de comunicación que las conecta.

***PPP –** Protocolo punto a punto*

Internet necesita un protocolo punto a punto para diversos propósitos, entre ellos para el trafico enrutador a enrutador y el trafico usuario domestico a *ISP*, este es el *PPP*.
PPP es un mecanismo de entramado que soporta detección de errores, negociación de

diferentes direcciones IP en el momento de la conexión, permite la autenticación y opcionalmente transmisión confiable con formato de tramas similar a *HDLC*.

PPP proporciona 3 características:

1) Un método de entramado.
2) Protocolo de control de enlace *LCP* para actuar, probar, negociar opciones y desactivar líneas.
3) Protocolo de control de red *NCP* para negociar opciones de capa de red con independencia del protocolo de red usado.

A diferencia de *HDLC*, *PPP* esta orientado a caracteres y usa relleno de byte en las líneas de acceso telefónico con modems.

Formato de la trama PPP

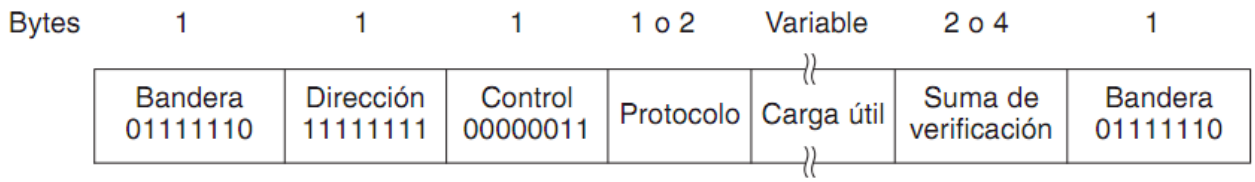
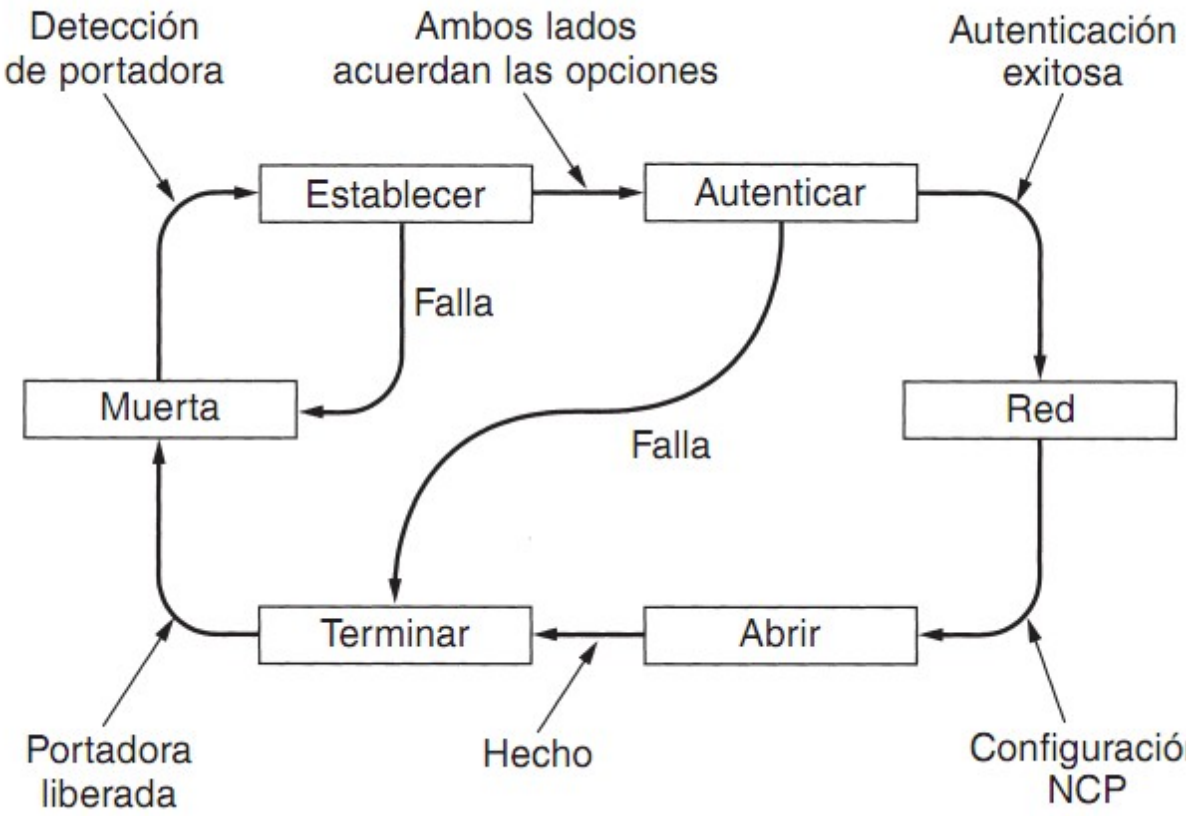


Figura 3-27. Formato de trama completa PPP para el modo de operación no numerado.

a) Comienzan y terminan con una ***Bandera***; si esta ocurre en la carga útil se rellena con un byte de escape.
b) Indica que todas las estaciones deben aceptar la trama (***Dirección***).
c) El valor indica que es una trama no numerada (***Control***). *PPP* es orientado a la conexión NO confiable, por lo tanto no usa números de secuencia ni confirmaciones de recepción.
b y ***c*** son constantes en la configuración predeterminada.
d) Indica la clase de paquete que hay en la carga útil (*LCP*, *NCP*, *IP*). Todos los que comienzan con un bit 0 son de la capa de red, los que comienzan con un bit 1 se usan para negociar otros protocolos (***Protocolo***).
e) Contiene cualquier información de longitud variable (***Carga útil***).
d) El campo de ***Suma de Verificación*** es un código de redundancia cíclica.

Diagrama de fases simplificado para activar y desactivar una línea



El protocolo inicia con la línea que tiene el estado ***MUERTA***, significa que no hay portadora de capa física y que no existe conexión. Una vez establecida la conexión física, la línea pasa a ***ESTABLECER***. En este punto comienza la negociación de opciones ***LCP*** para acordar los parámetros ***PPP*** por usar, que de tener éxito pasa a ***AUNTENTICAR***. Al entrar en la fase ***RED***, se invoca al protocolo ***NCP*** apropiado para configurar la red. Si esta tiene éxito se llega a ***ABRIR*** y puede comenzar el transporte de datos. Al terminar, la línea pasa a la fase ***TERMINAR*** donde regresa a ***MUERTA*** al liberar la portadora.

NOTA: Generalmente la *PC* requiere ejecutar una pila de protocolos *TCP/IP*, por lo que necesita una dirección *IP*. No hay suficientes *IP* para todos, por lo que normalmente el proveedor de Internet asigna dinámicamente a cada *PC* que acaba de conectarse una *IP* para que la use durante su sesión. Se usa el *NCP* de *IP* para asignar la dirección *IP*. Cuando termina la conexión se usa de nuevo *NCP* para finalizar la conexión de la capa de red y liberar *IP*, después se usa LCP para cancelar la conexión de la capa de enlace.

La subcapa MAC de control de acceso al medio

Redes de difusión y sus protocolos

El problema de asignación del canal: Forma de asignar un solo canal de difusión entre usuarios competidores.

Asignación estática del canal en LANs y MANs

Formas tradicionales:

1) *Multiplexión por división de frecuencias FDM*: Esta técnica implica dividir el canal en tantas partes como usuarios tenga y determinar que parte le pertenece a cada uno. Si hay ***N*** usuarios, el ancho de banda se divide en ***N*** partes iguales asignando a cada usuario una parte. Así todos los usuarios pueden transmitir información por el canal

simultáneamente sin sufrir interferencia con algún otro usuario.
Desventajas:
- Si el numero de usuarios es grande, cada usuario recibirá una porción del ancho de banda muy limitada trayendo como consecuencia disminución de velocidad de transmisión.
- Tráfico en forma de ráfaga: Cuando algunos usuarios estén inactivos, su ancho de banda simplemente se desperdicia porque no puede ser usado por ningún otro.

2) *Multiplexión por división de tiempo TDM*: Consiste en asignar todo el ancho de banda durante un intervalo muy corto de tiempo (*ranura*) a un usuario especifico.
Desventajas:
- Si un usuario no usa la *ranura* asignada, se desperdicia.

Asignación dinámica del canal en LANs y MANs

Para poder asignar dinámicamente un medio de transmisión se deben tener en cuenta las siguientes hipótesis:

1) *Modelo de estación*: ***N*** estaciones independientes cada una con un programa o usuario que generan tramas para la transmisión. Una vez que se genera una trama, la estación se bloquea y no hace nada.
2) *Supuesto de canal único*: Se tiene un solo medio de transmisión por el que podrán transmitir y recibir todas las estaciones.
3) *Supuesto de colisión*: Cuando dos tramas se transmiten simultáneamente se traslapan en el tiempo y la señal resultante se altera.
4) *Tiempo continuo o ranurado*: En un sistema continuo cada estación puede transmitir cuando quiera. En un sistema ranurado las estaciones comienzan a transmitir al inicio de una ranura (intervalo de tiempo predefinido)
5) *Detección o no de la portadora*: Los mecanismos de asignación del canal podrían tener o no la posibilidad de saber si el canal esta en uso o no.

Protocolos de acceso múltiple

Aloha: Tiene por objetivo resolver el problema de la asignación del canal de comunicación a varios usuarios sin ningún tipo de coordinación entre ellos.

Aloha puro – *Protocolo de acceso múltiple*: Ninguna estación se preocupa por saber si otra estación esta usando el canal y simplemente transmiten cuando necesitan, en cualquier instante de tiempo. Obviamente esto trae como consecuencia muchas colisiones.
Debido al peligro de que ocurran colisiones existen 2 (dos) alternativas:

- Escuchar el canal durante la transmisión para saber que no hubo colisión.
- Esperar una respuesta de confirmación de recepción.

Si la transmisión no fue exitosa, esta se reenvía después de un tiempo aleatorio para reducir la posibilidad de una nueva colisión.
Ventajas: Se adapta a un número variable de estaciones.
Desventajas: Tiene un rendimiento máximo de 18,4% y requiere almacenar la trama transmitida debido a posible retransmisiones.

Una trama no sufrirá colisión si no se envían otras tramas durante el tiempo de transmisión desde su envío. Sea t el tiempo de transmisión de su envío. Si cualquier otro usuario genero una trama entre el tiempo t_0, y el tiempo $t_0 + t$ es el final de la trama, esta chocará con el comienzo de la trama sombreada.

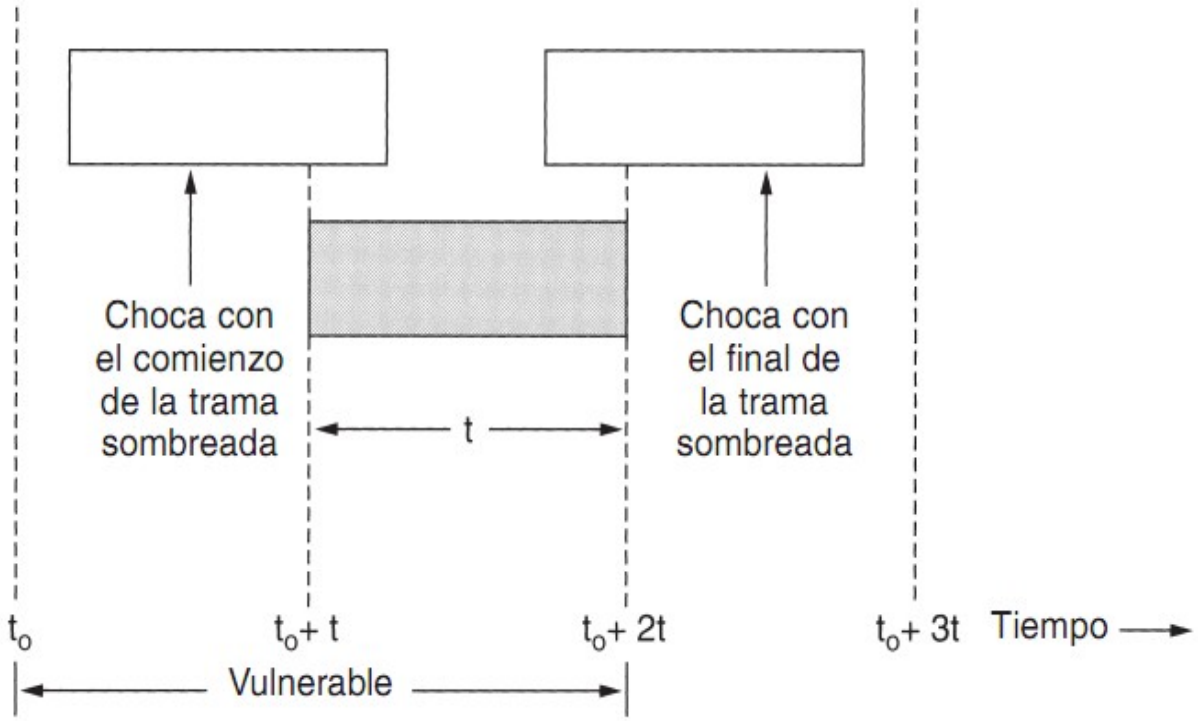
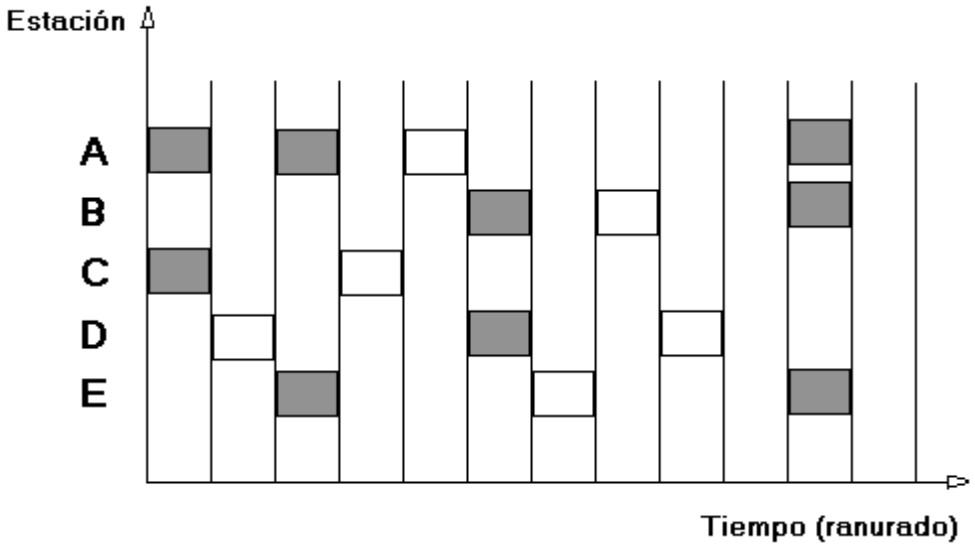


Figura 4-2. Periodo vulnerable para la trama sombreada.

Aloha ranurado – *Protocolo de acceso múltiple*

Se divide el tiempo de transmisión en ranuras equivalentes al tiempo de transmisión de una sola trama. Cada estación tiene permitido enviar un protocolo al inicio de una ranura de tiempo. Cuando se produzca una colisión, las tramas se superpondrán totalmente en vez de parcialmente. Esto hace que el rendimiento aumente un 50% respecto de *Aloha puro*.
Ventajas: La eficiencia de este protocolo es el doble de *Aloha puro* (*36,8%*) y se adapta a un número variable de estaciones.
Desventajas: Se requiere de sincronización entre las estaciones para determinar ranuras comunes de tiempo para todas ellas, y almacenar la trama transmitida debido a posibles retransmisiones.



Protocolos de acceso múltiple con detección de portadora CSMA

CSMA Persistente y No persistente: Son protocolos en los que las estaciones escuchan una transmisión y actúan en consecuencia.

***CSMA –** Persistente*

Cuando una estación tiene datos por transmitir, primero escucha el canal para saber si alguien esta transmitiendo; si esta ocupado, la estación espera hasta que quede libre. Cuando detecta un canal libre, empieza a transmitir la trama. Si ocurre una colisión, la estación permanece esperando que el canal este libre para empezar a transmitir de nuevo. Se llama así porque la estación transmite con probabilidad 1 cada vez que encuentre el canal ocupado.
El retardo de propagación tiene un efecto importante en el desempeño del protocolo. Puede pasar que justo después de que una estación comienza a transmitir, otra estación esta lista para enviar y detectar el canal. Si la señal de la primera estación no ha llegado todavía a la segunda, esta última detecta un canal inactivo y enviara produciéndose una colisión. Cuanto mayor sea el retardo de propagación, mas importante el efecto y peor rendimiento del protocolo.
Que el retardo de propagación sea cero no significa que no habrá colisiones. Si dos estaciones quedan listas a la mitad de transmisión de una tercera, ambas esperaran hasta el fin de la trasmisión y comenzaran a transmitir simultáneamente produciendo una colisión.

***CSMA –** No persistente*

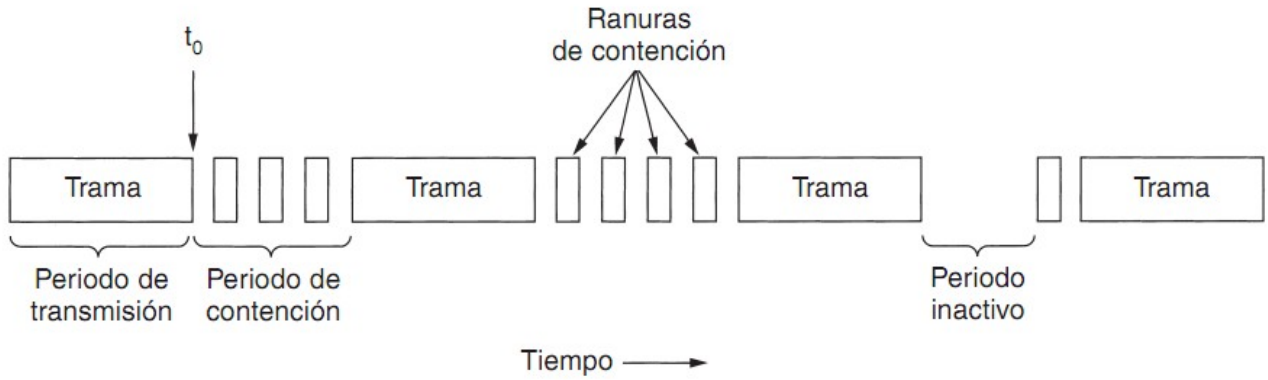
Antes de empezar a transmitir, la estación escucha el canal; si nadie esta transmitiendo la estación empieza a hacerlo sola. Sin embargo, si el canal ya esta en uso, la estación no estará escuchando continuamente a fin de tomarlo de inmediato al detectar el final de la transmisión previa, sino espera un tiempo aleatorio para después repetir el algoritmo.

***CSMA** – P-persistente*

Es aplicado a canales ranurados y trabaja de la siguiente manera. Una estación que esta lista para transmitir escucha el canal y si este esta libre, la estación transmite con una probabilidad ***p***, y retarda esta transmisión hasta la siguiente ranura, con una probabilidad ***q = 1 - p***. Si la siguiente ranura esta desocupada, el canal transmite o retarda de nuevo la transmisión con una probabilidad ***p*** y ***q***, respectivamente. Este proceso se repite hasta que la trama se haya transmitido u otra estación haya comenzado a transmitir.
En el último caso actúa como si hubiera existido una colisión. Si la estación en un inicio detecta el canal ocupado, espera hasta que llegue la siguiente ranura y repite el algoritmo.

***CSMA** – Con detección de colisiones*

Consiste en abortar inmediatamente la transmisión en el preciso momento en que las estaciones detectan una colisión.
Si dos estaciones detectan que el canal esta inactivo y comienzan a enviar en forma simultanea, ambas detectaran la colisión casi de inmediato, entonces detienen la transmisión de manera abrupta ahorrando tiempo y ancho de banda. La estación entonces, espera un tiempo aleatorio e intenta de nuevo.
Este protocolo emplea este modelo:



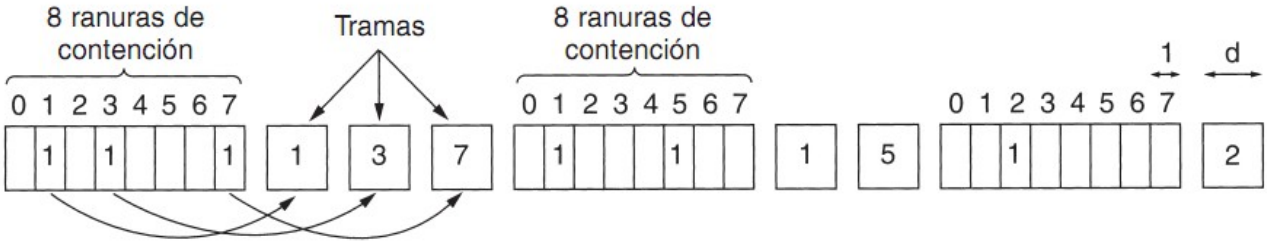
Consiste en 3 estados: ***Contención***, ***Transmisión*** e ***Inactividad***.

Protocolos sin colisiones

Supondremos que hay ***N*** estaciones, cada una con una dirección única de 0 a ***N-1*** incorporada en hardware. También supondremos que el retardo de propagación no importa.

Protocolo de mapa de bits

Cada periodo de contención consiste en exactamente ***N*** ranuras. Si la estación ***0*** tiene una trama por enviar, transmite un bit 1 durante la ranura ***0*** (ninguna otra estación puede transmitir durante esta ranura). En general, la estación ***j*** puede anunciar que tiene una trama por enviar introduciendo un bit 1 en la ranura ***j***. Una vez que han pasado las ***N*** ranuras, cada estación sabe cuales son todas las estaciones que quieren transmitir. En este punto las estaciones comienzan a transmitir en orden numérico.



Nunca habrá colisiones ya que todos están de acuerdo en quien continua. Una vez transmitida todas las tramas, comienza otro período de contención de ***N*** bits.

NOTA: *Protocolos de reservación* -> Son aquellos en los que el deseo de transmitir se difunde antes de la transmisión.

La eficiencia del canal cuando la carga es baja es fácil de calcular. La sobrecarga por trama es de ***N*** bits, y la cantidad de datos es de ***d*** bits, dando una eficiencia de ***d/(N + d)***.
Si la carga es alta y todas las estaciones tienen algo que enviar todo el tiempo, el periodo de contención de ***N*** bits se prorratea en ***N*** tramas, arrojando una sobrecarga de solo 1 bit por trama, o una eficiencia de ***d/(d + 1)***.

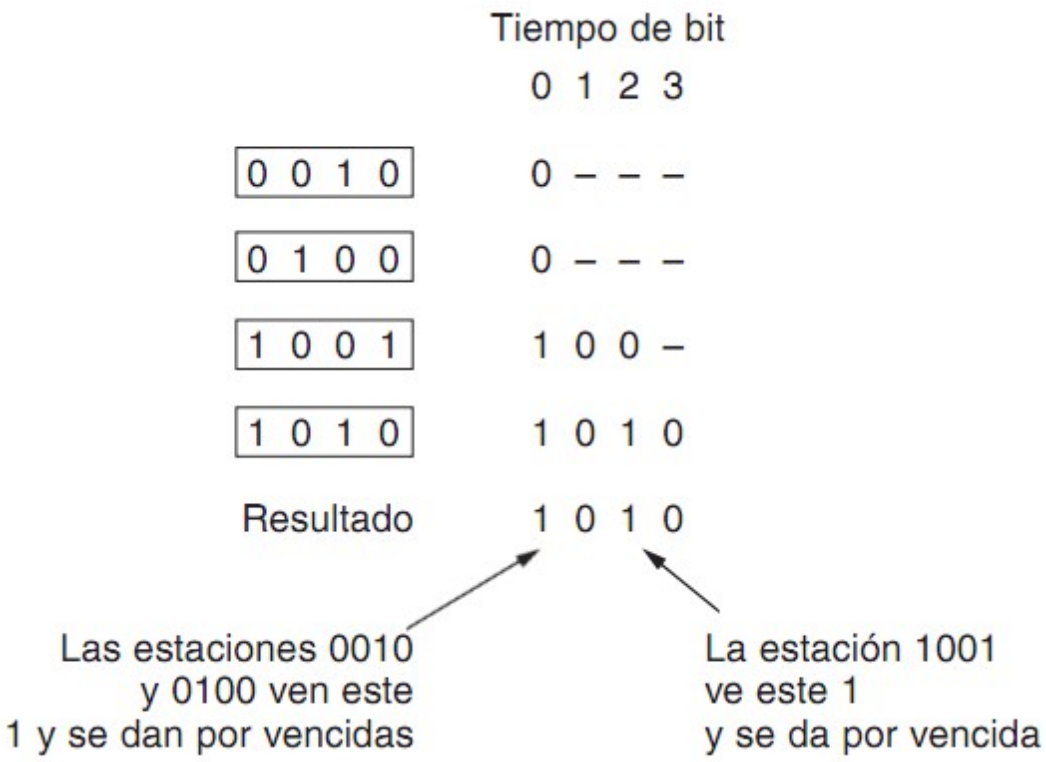
Conteo descendente binario

Una estación que quiere utilizar el canal ahora difunde su dirección como una cadena

binaria de bits (todas de la misma longitud), comenzando por el bit de orden mayor. A los bits de cada posición de las diferentes estaciones se les aplica un ***OR BOOLEANO***.

Regla de arbitraje: Una vez que una estación ve que una posición de bits de orden alto en su dirección es 0 y ha sido sobrescrita con un 1, se da por vencida. La dirección que gana la contienda, ahora puede transmitir la trama. Después de lo cual comienza otro ciclo de contienda.
Este protocolo tiene la propiedad de que estaciones con números grandes tienen una prioridad mayor que las que tienen números pequeños.

Veamos un ejemplo (los guiones indican silencio):



Protocolos de acceso múltiple por división de longitud de onda

Consiste en dividir el canal en subcanales usando *FDM*, *TDM* o ambas y asignarlos de manera dinámica según se necesite.

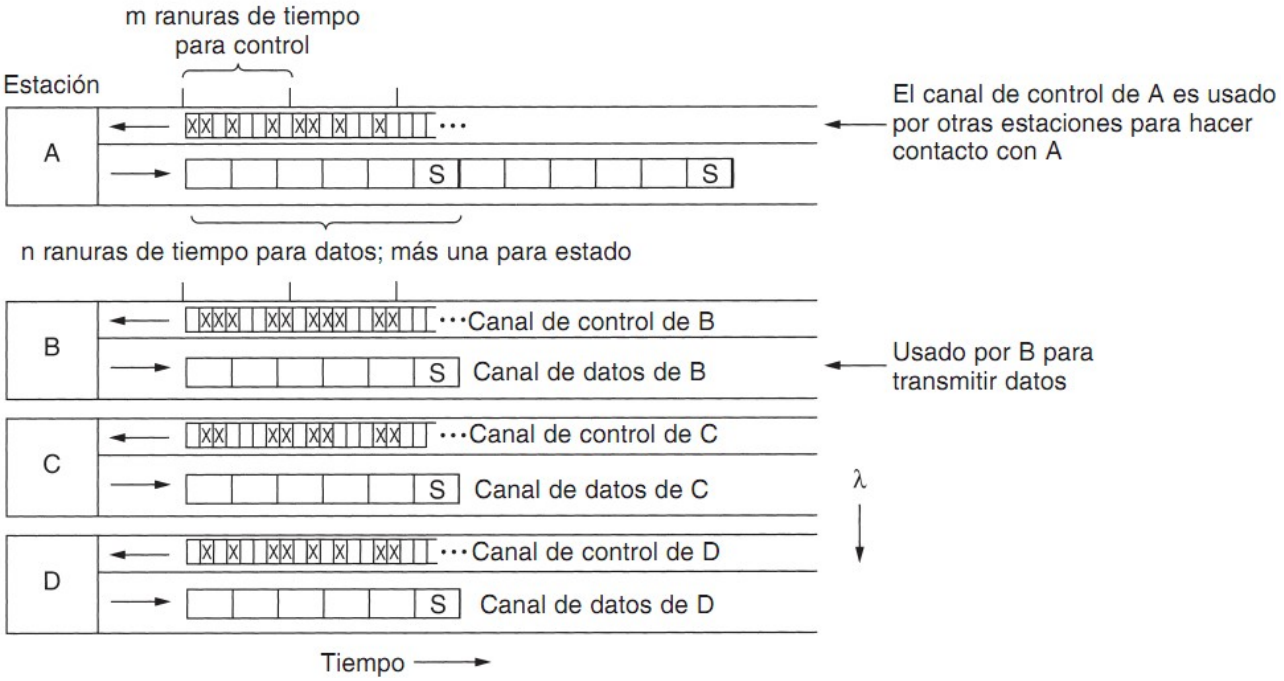
WDMA – *Acceso múltiple por división de longitud de onda*

Se asignan dos canales a cada estación, por lo que cada una tiene 2 emisores y 2 receptores. Un receptor de longitud de onda fija para escuchar su propio canal de control y uno sintonizable para seleccionar al emisor de datos a escuchar. Un emisor de longitud de onda fija para la salida de tramas de datos, y uno sintonizable para enviar por el canal de control de otra estación. Se proporciona un canal estrecho como canal de control para señalizar la estación, y uno ancho para que la estación pueda enviar tramas de datos. Todos los canales se sincronizan con un solo reloj global.

El protocolo reconoce 3 clases de tráfico:

1) *Orientado a la conexión con tasa de datos constante*: Se usa una variación de este protocolo. Cuando A solicita una conexión, simultáneamente dice algo como ¿Esta bien si te envío una trama cada vez que ocurra la ranura 3? Si B puede aceptar, se establece una conexión de ancho de banda garantizado. Si no, A puede intentarlo después con una propuesta distinta.
2) *Orientado a la conexión con tasa de datos variable* (Como transferencia de archivos - A comunicación con B): Primero A sintoniza su receptor de datos con el canal de datos de B y espera la ranura de estado (indica cuales ranuras están ocupadas/libres). A elige una de las ranuras de control libre, e introduce su mensaje de solicitud de conexión. Ya que B revisa de manera constante su canal de control, ve la solicitud y la acepta asignando la ranura solicitada a A. Esta asignación se anuncia en la ranura de estado del canal de datos de B. Cuando A ve el anuncio, sabe que tiene una conexión unidireccional. Si A solicita una conexión bidireccional, B repite ahora el mismo algoritmo con A.
   A ahora envía a B un mensaje de control, avisando que hay un mensaje en ***x*** ranura. Cuando B recibe el mensaje de control, sintoniza su receptor al canal de salida de A para leer la trama de datos.

3) *Trafico de datagramas* (como paquetes *UDP*): Usa otra variación del protocolo en lugar de escribir un mensaje de solicitud de conexión en la ranura de control que acaba de encontrar, diciendo que hay datos para el en la ranura ***x***. Si B esta libre durante la siguiente ranura de datos ***x***, la transmisión tendrá éxito. Sino, se perderá la trama de datos.



NOTA: Es posible también arreglárselas con un solo emisor y un solo receptor sintonizables por estación haciendo que el canal de cada estación se divida en ***m*** ranuras de control seguidas de ***n + 1*** ranuras de datos. La desventaja acá es que los emisores tienen que esperar más tiempo para capturar una ranura de control, y las tramas de datos consecutivas están más distantes porque se interpone cierta información de control.

Protocolos de LANs inalámbricas

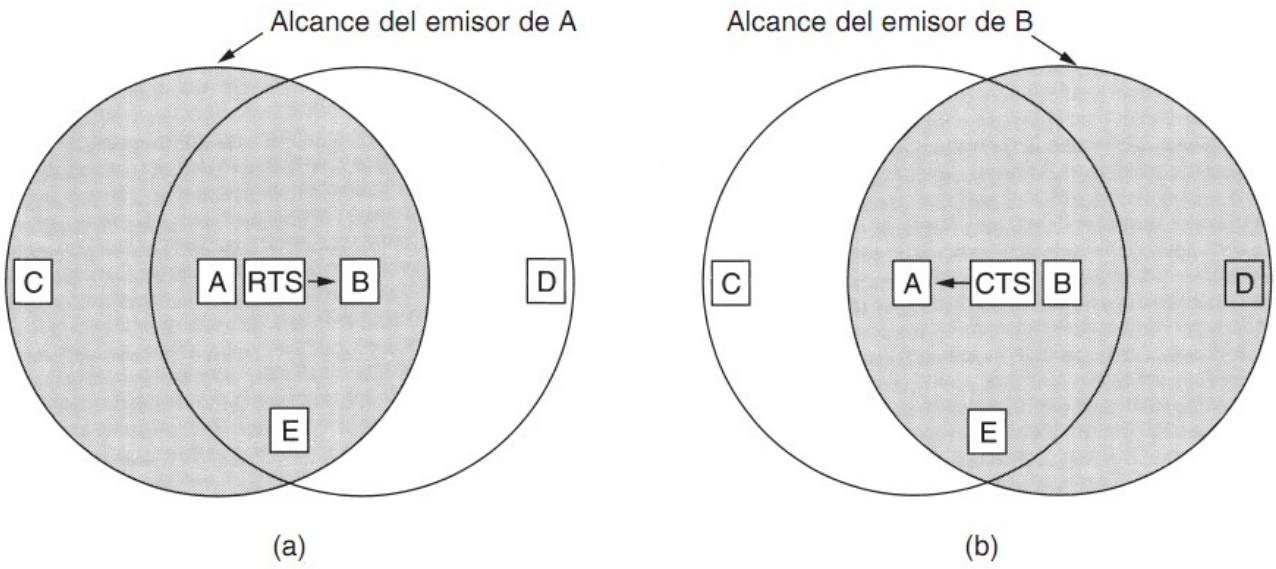
Suponemos que todos los emisores de radio tienen algún alcance fijo. Cuando el receptor esta dentro del alcance de dos emisores activos, la señal resultante generalmente se altera y resulta inútil. Ya no consideraremos los sistemas de tipo *CDMA* porque el problema es que antes de comenzar una transmisión, una estación realmente necesita saber si hay actividad o no alrededor del receptor.

***MACA** – Acceso múltiple con prevención de colisiones*

Se basa en el concepto de que el emisor estimule al receptor a enviar una trama corta, de manera que las estaciones cercanas puedan detectar esta transmisión y eviten ellas mismas hacerlo durante la siguiente trama de datos.

Funcionamiento de comunicación de A con B: A comienza por enviar una trama *RTS* (*solicitud de envío*) a B. Esta trama corta (30 bytes) tiene la longitud de la trama que seguirá posteriormente. Después B contesta con una trama *CTS* (*libre para envío*). La trama *CTS* contiene la longitud de los datos (copiada de la *RTS*), una vez que se recibe *CTS*, A comienza a transmitir.

¿Como reaccionan las estaciones que escuchan cualquiera de estas tramas?
Cualquier estación que escuche *RTA* esta bastante cerca de A y debe permanecer en silencio durante el tiempo suficiente para que *CTS* se transmita de regreso a A sin conflicto. Cualquier estación que escuche el *CTS* esta bastante cerca de B y debe permanecer en silencio durante la siguiente transmisión de datos.



Aun así pueden ocurrir colisiones. Supongamos que dos estaciones A y B envían tramas *RTS* a C al mismo tiempo. Estas chocarán y se perderán. Entonces el emisor espera un tiempo aleatorio y reintenta.

MACAW – *MACA inalámbrico*

Cambios en el *MACA* para mejorar su desempeño:

- Se descubrió que sin confirmación de recepción de la capa de enlace de datos, las tramas no eran retransmitidas sino hasta que la capa de transporte notaba su ausencia mucho después. Como solución introdujeron una trama *ACK* tras cada trama de datos exitosa.
- Decidieron también que *CSMA* puede servir para evitar que una estación transmita un *RTS* al mismo tiempo y destino que otra estación cercana, por lo que se agregó la detección de portadora.
- También decidieron ejecutar el algoritmo de retroceso por separado para cada flujo de datos en lugar de para cada estación.
- Agregaron un mecanismo para que las estaciones intercambiaran información sobre congestionamientos.

Ethernet

Ethernet e *IEEE 802.3* son idénticos, excepto por dos diferencias mínimas que analizaremos pronto.
Cableado Ethernet: Se usan 4 tipos de cableado. En orden de aparición, ellos son:

Nombre	Cable	Seg. máx.	Nodos/seg	Ventajas
10Base5	Coaxial grueso	500 m	100	Cable original; ahora obsoleto
10Base2	Coaxial delgado	185 m	30	No se necesita concentrador
10Base-T	Par trenzado	100 m	1024	Sistema más económico
10Base-F	Fibra óptica	2000 m	1024	Mejor entre edificios

Primero llego el cable ***10Base5***, llamado ***Ethernet grueso***. Este es totalmente rígido, no se dobla y tiene marcas cada 2.5 metros para indicar los puntos de derivación. Las

conexiones al cable *10Base5* se hacen usando derivaciones vampiro, en las que se introduce cuidadosamente una punta hasta la mitad del núcleo del cable coaxial (se pincha). Esta notación significa: el primer número es la velocidad en Mbps, después viene la palabra Base que indica transmisión en banda base y por ultimo, si el medio es coaxial, su longitud se da redondeada a unidades de 100 metros. Por eso podemos decir que opera a 10Mbps y maneja segmentos de hasta 500 metros.
Para *10Base5*, se sujeta firmemente un transceptor alrededor del cable, de modo que su derivación haga contacto con el núcleo interno. Este contiene la electrónica que maneja detección de la portadora y detección de colisiones.
Un cable de derivación conecta el transceptor a una tarjeta de interfaz en la computadora. Esta tarjeta tiene un chip controlador que transmite tramas al transceptor y recibe tramas de el.

Después, apareció el cable ***10Base2***, llamado ***Ethernet delgado***. A diferencia del grueso, este era más flexible, se dobla con facilidad. Las conexiones se hacen usando solo un conector *BNC* pasivo de unión *T*. La electrónica del transceptor esta en la tarjeta controladora, y cada estación siempre tiene su propio transceptor.

Estos 2 (dos) sufren muchas rupturas, malas derivaciones o conectores flojos. Para rastrear el problema se inyecta un pulso de forma conocida en el cable. Si el pulso incide en un obstáculo o en el final del cable, se generara un eco que viajará de regreso (reflectomía en el dominio del tiempo).

Como solución a estos problemas, surge el cable ***10Base-T*** llamado ***par-trenzado***, en el que todas las estaciones tienen cables que conducen a un concentrador central (*hub*), en el que se conectan de manera eléctrica.
Los concentradores no almacenan en el buffer el trafico de entrada. Agregar o eliminar estaciones es más sencillo con esta configuración y las rupturas de cable pueden detectarse con facilidad.
Desventaja: La longitud máxima de cable es de solo 100 metros.

Otra opción de cableado más rápida que 10Base-T es ***10Base-F***, que usa ***fibra óptica***. Es cara debido al costo de los conectores y los terminadores pero tiene excelente inmunidad al ruido y permite separaciones entre concentradores de kilómetros (longitud máxima de 2000 metros).

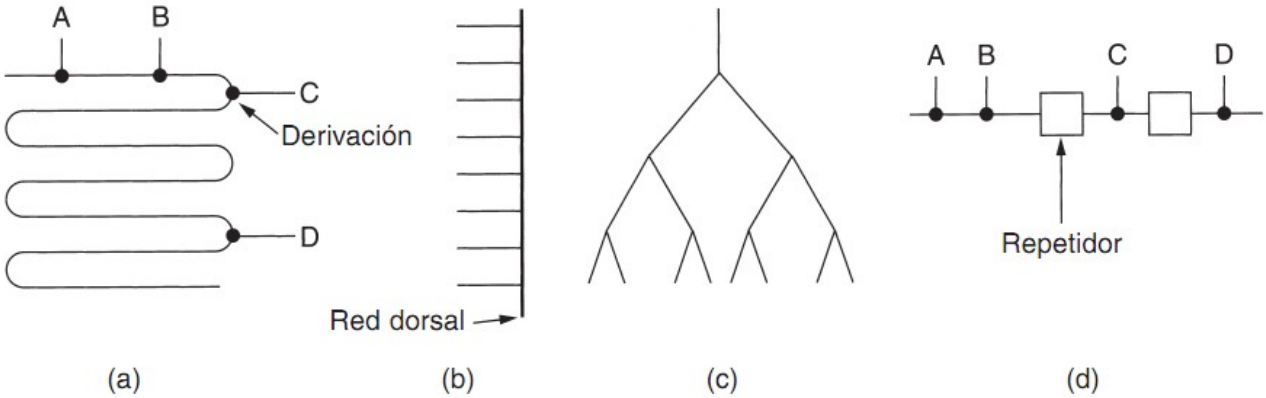


Figura 4-15. Topologías de cableado. (a) Lineal. (b) Columna vertebral. (c) Árbol. (d) Segmentada.

Para permitir redes mayores, se pueden conectar múltiples cables mediante *repetidores*, que amplifican y retransmiten las señales en ambas direcciones. Un sistema puede tener múltiples segmentos de cable y muchos repetidores pero ningún par de transceptores puede estar separado por más de 2.5 Km. (longitud máxima) y ninguna ruta de transceptores puede atravesar mas de 4 repetidores.

Codificación binaria que usa Ethernet

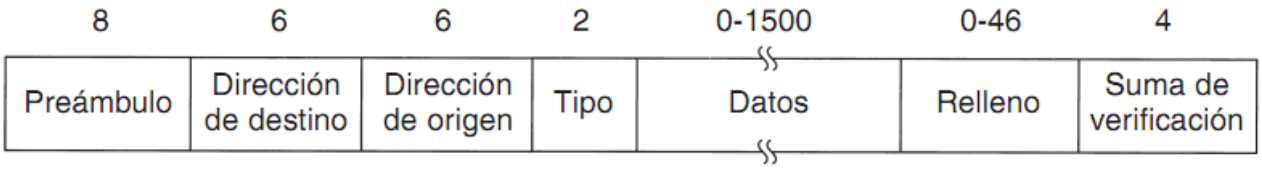
- *Codificación binaria directa*: No sirve porque no puede distinguirse entre un emisor inactivo (0 voltios) y un bit 0 (0 voltios).
- *Codificación bipolar*: Tampoco sirve porque puede pasar que un receptor muestre la señal a una frecuencia ligeramente distinta a la que haya utilizado el emisor para generarla y esto causa perdida de sincronismo.

Las próximas 2 (dos) si permiten a los receptores determinar el comienzo, final o mitad de cada bit sin referencia a un reloj externo.

- *Codificación Manchester*: Cada periodo de bit se divide en dos intervalos iguales. Un bit 1 binario se envía teniendo el voltaje alto durante el primer intervalo y bajo durante el segundo; y con un bit 0, justo lo inverso. Este esquema asegura que cada periodo de bit tenga una transición a la mitad, facilitando la sincronización del receptor con el emisor.
  Desventaja: Requiere el doble de ancho de banda que la codificación binaria común, ya que los pulsos son de la mitad de ancho.
- *Codificación Manchester diferencial***:** Igual que el anterior pero un bit 1 se indica mediante la ausencia de transición al principio del intervalo y un bit 0 por la presencia de transmisión al inicio del intervalo. Ofrece mejor inmunidad al ruido.

El protocolo de subcapa MAC de Ethernet

Cada trama inicia con un ***Preámbulo*** de 8 bytes, cada uno de los cuales contiene el patrón de bits *10101010* con el objetivo de sincronizar emisor y receptor.
Contiene 2 (dos) direcciones, una ***Dirección de origen*** y otra ***Dirección de destino***.
El bit de orden mayor de la dirección de destino es 0 para direcciones ordinarias y 1 para direcciones de grupo. Cuando una trama se envía a una dirección de grupo, todas las estaciones del grupo la reciben. Esto se llama *multidifusion* (*multicast*).
La dirección que consiste únicamente en bits 1 esta reservada para *difusión* (*broadcast* – Se acepta en todas las estaciones de la red).
Otra característica importante del direccionamiento es el empleo del bit 46 (adyacente al de orden mayor) para distinguir las direcciones locales de las globales. Las direcciones locales son asignadas por cada administrador de la red. Las direcciones globales son asignadas por el *IEEE* para asegurar que no haya 2 (dos) estaciones en ningún lugar del mundo con la misma dirección.
A continuación esta el campo ***Tipo*** que indica al receptor que hacer con la trama. Especifica a que proceso darle la trama.
Después están los ***Datos***, de hasta 1500 bytes.



Además de haber una longitud de trama máxima también hay una longitud de trama minima.
Razones:

- Cuando un *transceptor* detecta una colisión, trunca la trama actual, lo que significa que los bits perdidos y las piezas de las tramas aparecen todo el tiempo en el cable. Para que Ethernet pueda distinguir con facilidad las tramas validas de las tramas

basura, necesita que dichas tramas tengan una longitud de por lo menos 64 bytes, desde la ***Dirección de destino*** a la ***Suma de Verificación*** (incluyéndolas). Las tramas con menos de 64 byte se rellenan con 64 byte con el campo de ***Relleno***.
- Para evitar que una estación complete la transmisión de una trama corta antes del que el primer bit llegue al extremo mas alejado del cable, donde podría tener una colisión con otra trama. Cuando una estación detecta que esta recibiendo mas potencia de la que esta enviando, sabe que ha ocurrido una colisión entonces aborta su transmisión y genera una ráfaga de ruido de 48 bits para avisar a las demás estaciones.

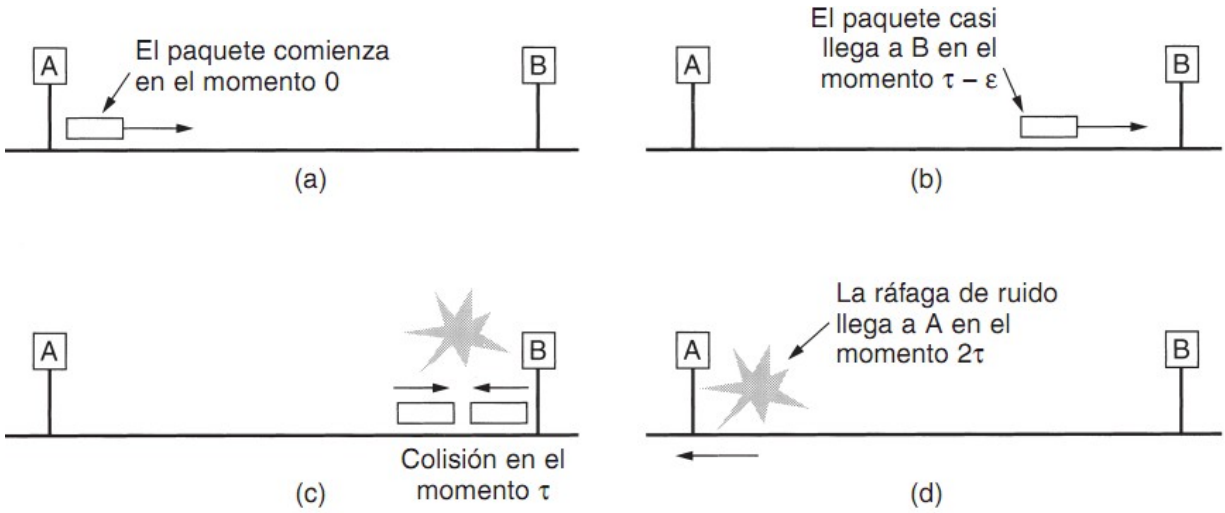


Figura 4-18. La detección de una colisión puede tardar hasta 2τ.

Para una *LAN* con 10Mbps con una longitud máxima de 2.5 Km. y 4 repetidores, el tiempo de ida y vuelta se ha determinado a aproximadamente 50 microsegundos en el peor caso, incluyendo el tiempo para pasar a través de los repetidores.

¿Porque? La distancia máxima permitida para Ethernet es de 2500 m., a 10Mbps y 4 repetidores.
Nota: Consideramos la *velocidad de la luz* $2x10^8$ m/s.
Entonces, si en 1 segundo son $2x10^8$ metros; en 12,5 microsegundos recorremos los 2500 metros. Pero nos interesa saber el tiempo de ida y vuelta, por lo que debemos hacer 12,5*2 obteniendo asi aproximadamente 50 microsegundos de transmisión en el peor caso.
Ahora, si en 1 segundo transmite $10x10^6$ bits, en 50 microsegundos transmite 500 bits. Pero para agregar un margen de seguridad, se redondeo a un número exacto de byte; y el más próximo es 512 bits = 64 bytes. Entonces, si transmite 500 bits en 50 microsegundos, 512 bit los transmite en 51,2 microsegundos. Las tramas con menos de 64 bytes se rellenan con 64 bytes con el campo de relleno.

El campo final de Ethernet es la ***Suma de Verificación***. Si alguno de los bits de datos se recibe erróneamente, es casi seguro que la suma de verificación esta mal y se detectará el error. El algoritmo es un *CRC*. Detecta pero no corrige.

Diferencias con Ethernet DIX e IEEE 802.3

IEEE 802.3 redujo el preámbulo a 7 bytes y usa el último como delimitador de inicio de trama por compatibilidad con *802.4* y *802.5*.
Cambió el campo de *Tipo* en un campo de *Longitud*. Es claro que ahora no había forma de

que el receptor supiera que hacer con la trama entrante. Esto se soluciono agregando un pequeño encabezado a la porción de datos que brinde esa información.

Algoritmo de retroceso exponencial binario

Proceso de aleatorización cuando hay una colisión: Tras una colisión, el tiempo se divide en ranuras discretas cuya longitud es igual al tiempo de propagación de ida y vuelta del peor caso (2T), es decir 51,2 microsegundos o 512 tiempos de bits.
Tras la primera colisión, cada estación espera 0 o 1 tiempos de ranura antes de intentarlo de nuevo. Si ambas eligen el mismo numero aleatorio, entraran otra vez en colisión. Después de la segunda cada una escoge 0,1,2 o 3 al azar y espera ese numero de tiempos de ranura. En general, tras ***i*** colisiones, se elige un numero aleatorio entre 0 y $\boldsymbol{2^i - 1}$, y se salta ese numero de ranura.
El intervalo de aleatorización se congela en un máximo de *1023* ranuras, para evitar que el límite crezca demasiado. Si se llega a las *16* colisiones, se rinde.

Ethernet conmutada

Surgió como solución al aumento de tráfico por incorporación de nuevos usuarios. Consiste en un *conmutador* (*switch*) que tiene una matriz de conmutación de alta velocidad (1Gbps) y espacio para 4-32 tarjetas de línea, cada una de las cuales contiene de 1-8 conectores. Comúnmente cada conector tiene una conexión de cable ***10Base-T*** a una sola computadora. Esto permite que cada estación disponga de un canal de *10 ó 100 Mbits/s*, en lugar de un único canal para todas.

Cuando una estación quiere transmitir una trama Ethernet, envía una trama estándar al conmutador. La tarjeta que la recibe la revisa para ver si esta destinada a una de las otras estaciones conectadas a esa tarjeta. Si es así se copia ahí, sino se envía a través de la *matriz de conmutación* de alta velocidad a la tarjeta de la estación destino.

Las tarjetas de conexión pueden ser de 2 (dos) tipos:

1) Todos los puertos de la tarjeta forman una *LAN* local dentro ella. Las colisiones en esta *LAN* se detectan y manejan igual que cualquier otra colisión en una red *CSMA/CD*, en las que las retransmisiones usan el algoritmo de retroceso exponencial binario. De esta manera solo es posible una transmisión por tarjeta en un momento dado, pero todas las tarjetas pueden estar transmitiendo en paralelo. Cada tarjeta forma su propio dominio de colisión, independiente de los demás.
2) Cada puerto de entrada se almacena en un buffer, por lo que las tramas de entrada se almacenan en la *RAM* de la tarjeta mientras van llegando. Permite que todos los puertos de entrada reciban y transmitan al mismo tiempo. Una vez que llega por completo la trama, la tarjeta puede determinar si la trama esta destinada a otro puerto de esa tarjeta o a otro distante. En el primer caso se transmite directamente, en el segundo a través de la *matriz de conmutación*. Con este diseño cada puerto es un dominio de colisión independiente.

Ya que el *conmutador* solo espera tramas Ethernet en cada puerto de entrada, es posible utilizar alguno de los puertos como *concentradores*, para conectar a mas estaciones.