

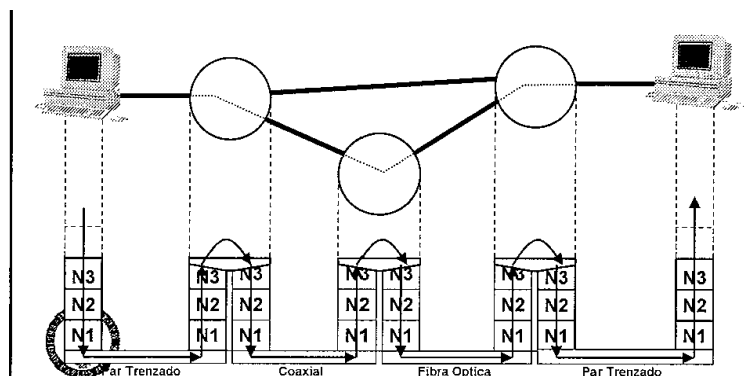
Tema 3: El Nivel de Enlace

3.1. Funciones del Nivel de Enlace

La principal función del nivel de enlace es la de resolver los problemas derivados de la falta de fiabilidad de los circuitos físicos. El objetivo es por lo tanto proporcionar una **transferencia fiable** de bloques de información (tramas) entre equipos directamente conectados.

La descripción anterior introduce el concepto de **trama**. Vamos a definir un **mensaje** como una *secuencia de caracteres o bits que representa la información a enviar de un origen a un destino*. Un **bloque** es un *conjunto de caracteres o bits que se agrupan por razones técnicas para ser transmitidas como una unidad*. Por último, una **trama** es la *estructura de datos que maneja el nivel de enlace para enviar un bloque*.

Para implementar la funcionalidad de transferencia fiable, el nivel de enlace utiliza una técnica diferente en cada uno de los enlaces atravesados por el camino de la comunicación. Por lo tanto, en una comunicación entre un origen y un destino vamos a tener tantas instancias del nivel de enlace como enlaces haya en el camino (e.g. el N2 del par trenzado es distinto al del coaxial).



Las principales funciones que implementa el nivel de enlace son las siguientes:

- **Delimitación de tramas.** Como hemos dicho anteriormente, la información se manda en estructuras o bloques denominados tramas. La función de delimitación de tramas permite detectar el inicio y el fin de una trama.
- **Control de flujo.** El control de flujo controla que la estación emisora no manda tramas a una tasa superior a la que puede absorberlas la estación receptora.
- **Control de errores.** En caso que el circuito físico introduzca errores de bit, el nivel de enlace tiene que corregir estos errores.
- **Coordinación y compartición del canal.** En caso de que un circuito sea compartido por dos o más estaciones, es responsabilidad del nivel de enlace

determinar qué estación está autorizada a transmitir en un instante determinado, ya que si transmite más de una estación simultáneamente no va a recibirse correctamente ninguna de las señales transmitidas.

- **Direccionamiento.** En el caso de un circuito compartido por más de dos estaciones, al mandar una trama tenemos que identificar a qué estación va dirigida. El direccionamiento nos permite cumplir esta funcionalidad.

A continuación se proporciona una descripción genérica de las técnicas existentes para implementar las diversas funcionalidades del nivel de enlace. En las secciones 4.6 y 4.7 se presentan dos protocolos de nivel de enlace ampliamente utilizados, HDLC y Ethernet, que representan una implementación específica de estas funcionalidades.

3.2. Delimitación de tramas

La delimitación de tramas permite saber donde empieza y donde acaba una secuencia de datos o trama. Existen las siguientes soluciones para esta funcionalidad:

- Utilización de tramas de tamaño fijo
- Delimitación por carácter de principio y fin
- Delimitación por guiones

A continuación vamos a proceder a evaluar las diferentes soluciones. Uno de los aspectos de esta evaluación se va a referir a la **transparencia**. Se dice que un protocolo es transparente si es capaz de enviar cualquier dato o secuencia de bits. Un protocolo no transparente, por lo tanto, tiene ciertas secuencias de bits prohibidas (normalmente aquellas que representan un carácter de control).

Tramas de tamaño fijo

La solución de **utilizar tramas de tamaño fijo** es intrínsecamente transparente. El problema de esta solución radica en que es *poco flexible*. Por ejemplo, si se desea poder mandar tramas de diferentes tamaños, el tamaño de trama tendrá que ser igual al máximo que se desee mandar, lo cual comporta un desperdicio de canal cuando se mandan tramas cortas.

Delimitación por carácter de principio y fin

La **delimitación por carácter de principio y fin** conlleva problemas de transparencia, tal y como se puede observar en los siguientes ejemplos:

Ejemplo 1:

Carácter de principio/fin: \$

Datos a enviar: abcde

Trama: \$abcde\$

Ejemplo 2:

Carácter de principio/fin: \$

Datos a enviar: abc\$de

Trama: \$abc\$de\$ ERROR: se va a interpretar como dos tramas diversas

El protocolo anterior, por lo tanto, no puede mandar \$ en los datos, por lo que no es transparente.

Una solución al problema anterior es utilizar un carácter especial, denominado de escape, que precede a los caracteres de control. Este carácter de escape se va a duplicar cuando aparece en los datos. De esta forma conseguimos transparencia a costa de mandar información extra. El ejemplo siguiente ilustra el mecanismo.

Carácter de principio/fin: \$

Carácter de escape: %

Datos a enviar: abc%\$d\$e

Trama: %\$abc%%\$d\$e%\$

La solución anterior proporciona una eficiencia que en el peor de los casos puede llegar a ser del 50% (cuando todos los caracteres enviados son el carácter de escape).

Delimitación por guiones

Por último, la **delimitación por guiones** consiste en utilizar una secuencia de bits, el **guión**, como separación de tramas (e.g. 01111110) que garantizamos que no aparece en los datos (siempre que veamos 5 1's en los datos añadimos un 0 y en el receptor eliminamos este 0 antes de entregar los datos al nivel superior).

Ejemplo:

Datos a enviar: 001111111001111110

Trama: 01111110 001111110110011111100 01111110

La eficiencia en el peor caso del protocolo anterior es de 5/6 (en caso de que en el mensaje todo sean 1's).

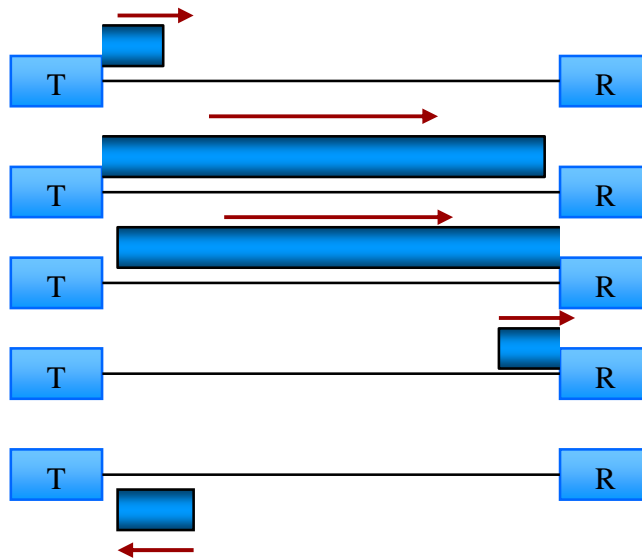
3.3 Técnicas de control de flujo

El objetivo del control de flujo es el de limitar la cantidad de información que el transmisor puede enviar al receptor para que no sature los recursos (memoria, ...) disponibles en el receptor. En el caso de que los recursos, como, por ejemplo, la memoria del receptor, se saturaran, éste no tendría otra solución que proceder a descartar las tramas recibidas, lo cual no es deseable. Dos técnicas para el control de flujo son la **parada y espera** y la **ventana deslizante**.

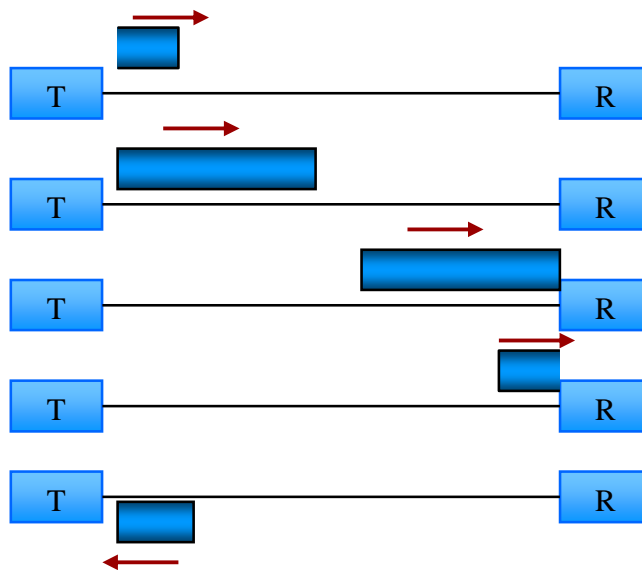
Parada y espera

La técnica de la parada y espera se basa en el siguiente algoritmo:

- La fuente envía una trama y espera confirmación.
- El receptor envía la confirmación al recibir la trama.



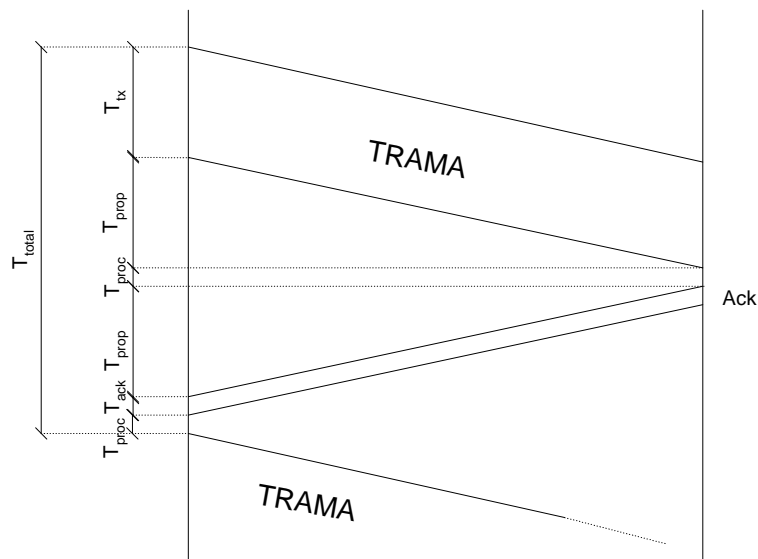
El problema de la parada y espera es que sólo permite una trama en tránsito, lo cual provoca un uso ineficiente del caudal físico del canal. Este problema se agrava cuanto mayor es el retardo de propagación.



A continuación se analizan las prestaciones del protocolo parada y espera. El parámetro en el que se basa la evaluación es la eficiencia U , que es el porcentaje máximo de tiempo en el que el enlace puede ocuparse para transmisiones de datos (tiempo empleado para transmisiones útiles). En un canal ideal la eficiencia es del 100%.

$$U = \frac{T_{util}}{T_{total}}$$

Para analizar la eficiencia del protocolo parada y espera, seleccionamos un intervalo que se repita periódicamente y analizamos cuál es la eficiencia en este intervalo. Concretamente, el intervalo que escogemos es el comprendido entre la transmisión de una trama de datos y la siguiente.



El tiempo dedicado a transmisiones útiles en el intervalo anterior es el tiempo de transmisión de una trama de datos:

$$T_{util} = T_{tx}$$

y el tiempo total es igual a:

$$T_{total} = T_{tx} + 2T_{proc} + 2T_{prop} + T_{ack}$$

con lo que:

$$U = \frac{T_{tx}}{T_{tx} + 2T_{proc} + 2T_{prop} + T_{ack}}$$

Si suponemos que los retardos de procesamiento y el de transmisión del acknowledgement (ack) o confirmación son despreciables, y definimos $a = T_{prop}/T_{tx}$, nos queda la siguiente expresión para la eficiencia:

$$U = \frac{1}{1 + 2a}$$

Podemos observar que la eficiencia aumenta con el retardo de transmisión y disminuye con el de propagación.

Otro parámetro que se utiliza para evaluar las prestaciones de un protocolo es el caudal eficaz C_{ef} , definido como el caudal de información útil (i.e. información proporcionada por los niveles superiores) experimentado. Si C es el caudal físico del enlace, y U es la eficiencia del protocolo, el caudal eficaz puede calcularse como

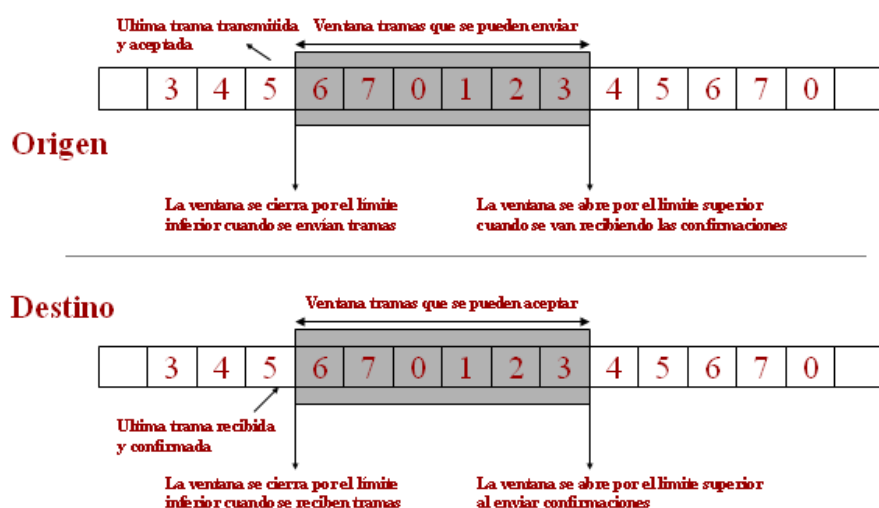
$$C_{ef} = U \cdot C$$

Ventana deslizante

La técnica de la ventana deslizante se basa en el siguiente algoritmo:

- Se permite el envío simultáneo de varias tramas en tránsito.
- El destino reserva n buffers para la recepción de tramas.
- El origen puede enviar n tramas sin esperar confirmación.
- Las tramas deben numerarse mediante el uso de un campo de longitud finita

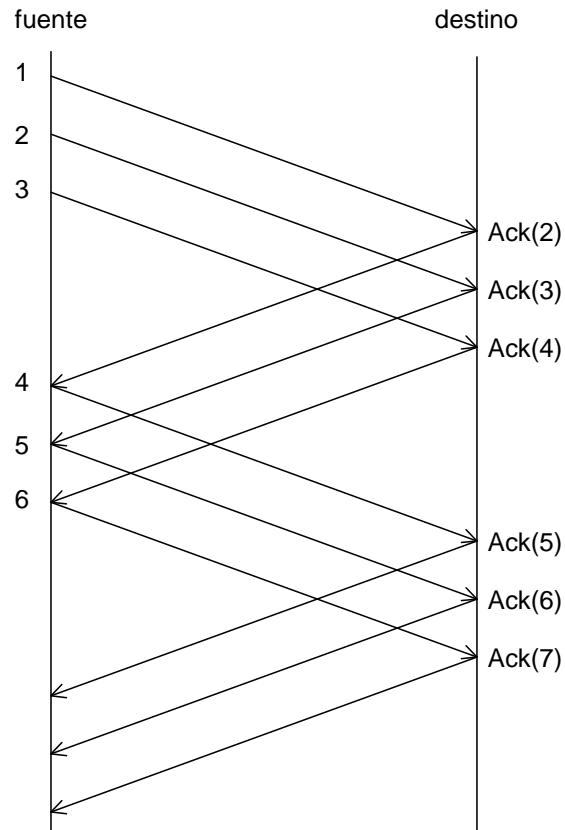
Este comportamiento se ilustra en el siguiente gráfico



El siguiente gráfico ilustra un ejemplo en el que se utiliza una ventana $W = 3$. La notación que se utiliza en el ejemplo es la siguiente:

$F(n)$: trama de datos n

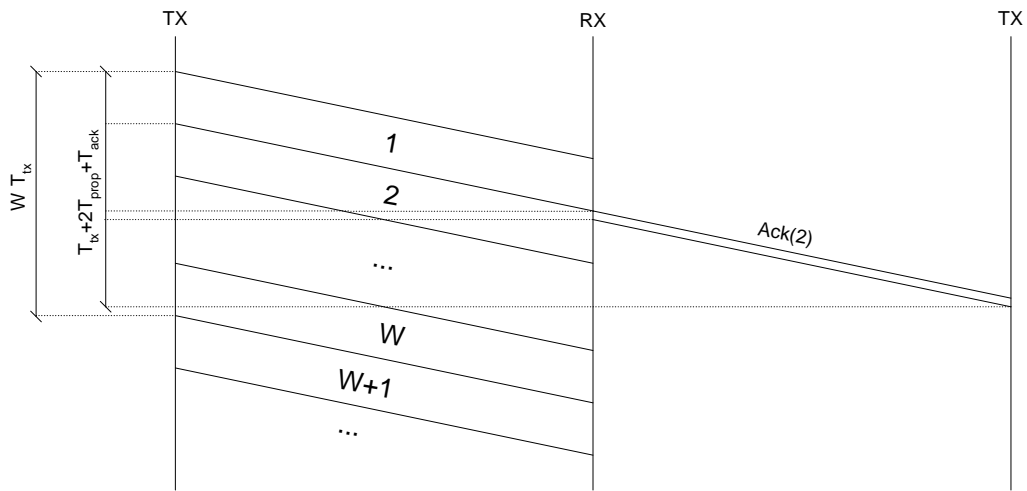
$Ack(n)$: he recibido hasta la trama $n-1$ y espero recibir la n



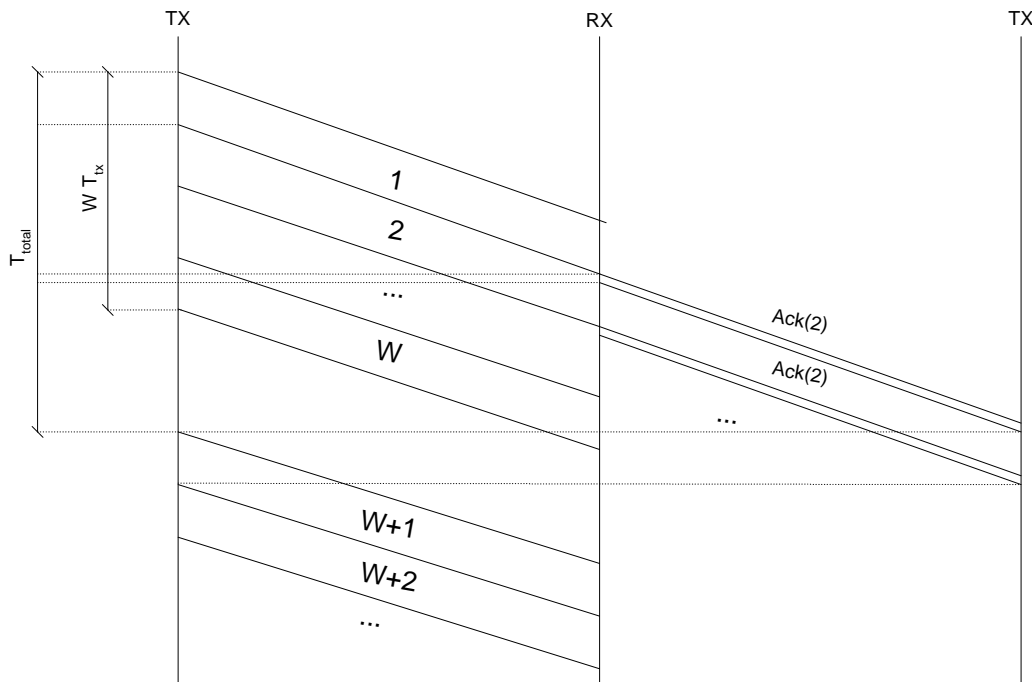
Para analizar las prestaciones del algoritmo de ventana deslizante, distinguimos dos casos. El primer caso es cuando, al terminar la transmisión de la W -ésima trama, se ha recibido ya la confirmación (Ack) de la primera (ver gráfico adjunto). En este caso tenemos un envío continuo de tramas y por lo tanto una eficiencia igual a la unidad ($U = 1$). Para que se de este caso, se tiene que cumplir la siguiente condición¹:

$$W \cdot T_{tx} \geq T_{tx} + 2T_{proc} \Rightarrow W \geq 1 + 2a$$

¹ Estamos suponiendo que los retardos de procesamiento y de transmisión del Ack son despreciables.



El caso en el que la anterior condición no se cumpla ($W < 1 + 2a$), procedemos de la siguiente forma para calcular la eficiencia. Puede verse que el intervalo indicado en el gráfico adjunto va a repetirse periódicamente, puesto que vamos a tener una ráfaga de W transmisiones, que provoca la recepción de una ráfaga de W confirmaciones, con un tiempo de transmisión entre dos confirmaciones consecutivas, lo que nos lleva de nuevo a mandar una nueva ráfaga de W tramas.



Dividiendo el tiempo empleado en transmisiones útiles por el total del intervalo:

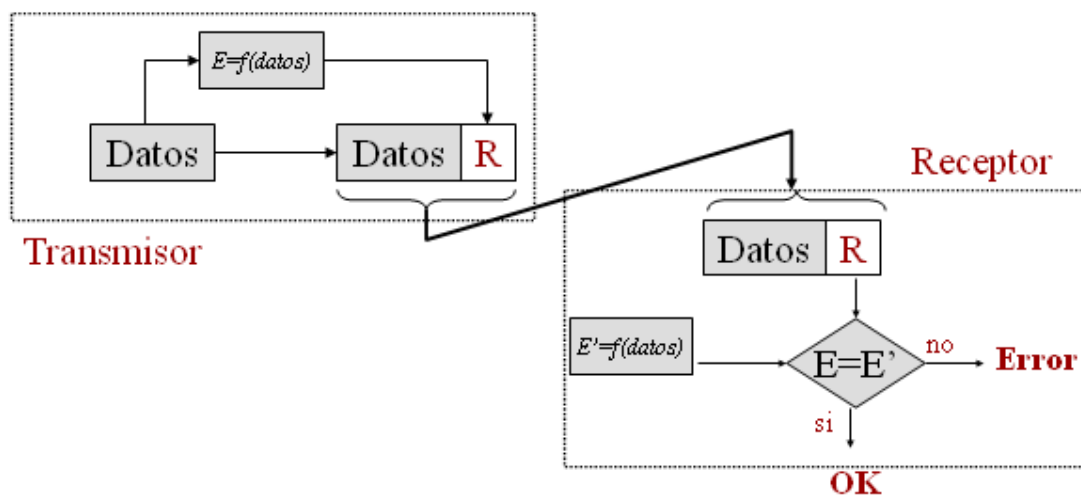
$$U = \frac{T_{util}}{T_{total}} = \frac{W \cdot T_{tx}}{T_{tx} + 2T_{prop}} = \frac{W}{1 + 2a}$$

3.4. Protocolos de Control de Errores

Se denomina control de errores al conjunto de técnicas que permiten resolver los problemas introducidos por canales ruidosos con probabilidades de error inaceptables para las aplicaciones finales. Existen tres familias distintas de técnicas de control de errores: las de detección, las de corrección y las de retransmisión.

Detección de Errores

El método que se utiliza para la detección de errores es añadir información redundante a los datos (concretamente a cada trama) para detectar errores, tal y como se ilustra en el siguiente gráfico.



La clave del método anterior reside en cómo calcular la redundancia (R). Un posible mecanismo es el denominado **paridad**, que añade un bit a la secuencia de datos indicando si el número de 0's es par o no:

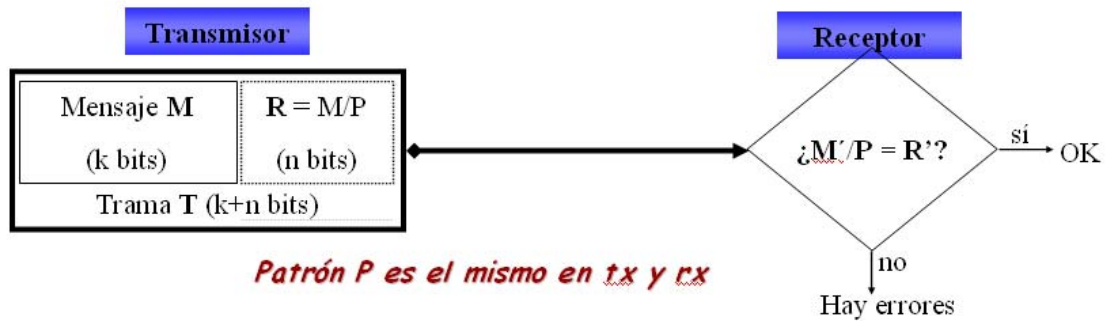
- R = 1: número par de "0s"
- R = 0: número impar de "0s"

El problema del mecanismo anterior es que sólo detecta error cuando el número de bits erróneos es impar.

Otro mecanismo más elaborado para la detección de errores es el denominado **CRC** (Cyclic Redundancy Check). Éste es el método más comúnmente utilizado en los protocolos de comunicación.

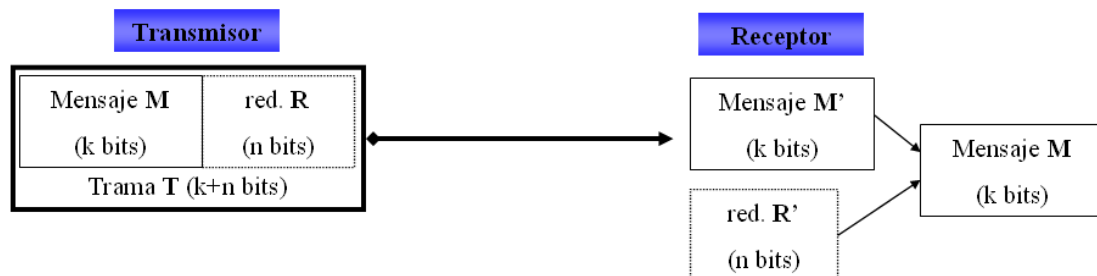
Con el CRC, el transmisor, dado un mensaje M de k bits, genera un código R de n bits, que es el resultado de coger el resto de la división M/P , donde P es una secuencia de $n+1$ bits conocida tanto por el transmisor como por el receptor.

El receptor, con los k primeros bits recibidos, M' , genera R' utilizando el mismo algoritmo y patrón P . Si el resultado R es igual a los últimos n bits recibidos, se considera que no hay errores.



Corrección de Errores

Las técnicas **FEC** (Forward Error Correction) añaden suficiente información redundante no sólo para detectar que se ha producido un error en el mensaje sino para corregir los errores ocurridos directamente en el destino. Con esta técnica, no es necesario retransmitir los bits erróneos para recuperar el mensaje original. Como contraprestación, la redundancia necesaria es mayor.



Técnicas de retransmisión ARQ

Las técnicas de retransmisión (ARQ- Automatic Repeat Request) se basan en utilizar técnicas de detección de errores en el receptor y retransmitir aquellas tramas que se detectan como erróneas. Distinguimos tres técnicas ARQ: **ARQ de parada y espera**, **ARQ de rechazo simple** y **ARQ de rechazo selectivo**.

ARQ de parada y espera

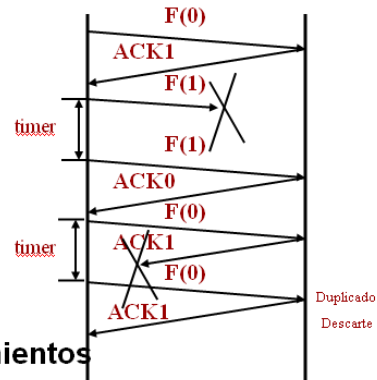
El protocolo ARQ de parada y espera se basa en el protocolo de parada y espera visto en la sección 2.3 al que se le añade la funcionalidad de retransmisión de tramas erróneas.

Existen dos mecanismos, que pueden usarse de modo alternativo o complementario, para programar las retransmisiones: 1) el uso de un timer o timeout, que, al expirar sin recibirse el ACK correspondiente, dispara la retransmisión, y 2) el uso de tramas de confirmación negativa (NACK o REJ).

◆ **Dos tipos de error:**

- ❖ Trama dañada o perdida
 - ✓ Detección y descarte de trama
 - ✓ Temporizador en fuente para retransmisión

- ❖ ACK dañado o perdida
 - ✓ Numeración de tramas y asentimientos (0,1), para evitar duplicados en el receptor.



Antes de analizar las prestaciones del ARQ de parada y espera, realizamos un **inciso de teoría de probabilidad**.

Imaginemos el evento consistente en que al que en tirar un dado salga el número 3. La probabilidad de este evento es $p = 1/6$.

La probabilidad de que el resultado sea diferente de 3 será $1 - p = 1 - 1/6$.

La probabilidad de que tirando el dado 2 veces, ambas salga 3 es de $p^2 = (1/6)^2$.

La probabilidad de que tirando el dado 2 veces, ninguna salga 3 es de $(1-p)^2 = (1-1/6)^2$.

La probabilidad de que tirando dos veces, alguna vez (una o dos) salga 3 es $1-(1-p)^2$.

En general, la probabilidad de que tirando el dado n veces, ninguna salga 3 será $(1-p)^n$, y que en una o más veces salga 3 es $1-(1-p)^n$.

La probabilidad de que tirando el dado n veces, la primera y sólo en la primera salga 3 será $p(1-p)^{n-1}$.

La probabilidad de que tirando el dado n veces, una sola vez salga 3, será $np(1-p)^{n-1}$.

Si tiramos el dado las veces que haga falta hasta que salga 3, el número de veces que tendremos que tirar el dado, en promedio, podemos calcularlo siguiendo el siguiente

razonamiento. Con probabilidad p tiraremos el dado una sola vez (la primera será tres). Con probabilidad $(1-p)p$ lo tiraremos dos veces (la primera no sale 3 pero la segunda sí). Así podemos calcular la media de acuerdo con la siguiente fórmula:

$$m = \sum_{i=1}^{\infty} ip(1-p)^{i-1} = \frac{1}{p}$$

Con lo que hemos dicho hasta ahora, podemos ya analizar las prestaciones del ARQ de parada y espera. La probabilidad de que se produzca un error en algún bit de la trama, si la probabilidad de error de bit es p y el número de bits de la trama es n , se puede calcular como

$$P_{err} = 1 - (1-p)^n$$

Si cogemos el mínimo timer posible ($T_{tx} + 2T_{prop}$), y despreciamos las pérdidas que no sean causadas por errores y la probabilidad de error en la confirmación:

$$U = \frac{T_{tx}}{T_{total}} = \frac{T_{tx}}{N_t * (T_{tx} + 2T_{prop})} = \frac{1}{N_t(1 + 2a)}$$

N_t	Prob
1	$1 - P_{err}$
2	$P_{err}(1 - P_{err})$
.....
n	$P_{err}^{n-1}(1 - P_{err})$

$$N_t = \sum_{i=1}^{\infty} i(1 - P_{err})P_{err}^{i-1} = \frac{1}{1 - P_{err}}$$

$$U = \frac{1 - P_{err}}{1 + 2a}$$

ARQ de rechazo simple

El ARQ de rechazo simple utiliza una ventana de W tramas. El receptor, al recibir una trama correcta envía una confirmación positiva (ACK) y al recibir una trama desordenada envía una confirmación negativa (REJ) indicando la siguiente trama que espera recibir. El origen al recibir el REJ retransmite la trama rechazada y todas las posteriores. En caso de no recibir ninguna confirmación (ni positiva ni negativa), en expirar el temporizador correspondiente la fuente retransmite la trama. En la siguiente figura se ilustra este comportamiento para una ventana $W = 3$.

Reserva: TDM

La técnica TDM (Time Division Multiplexing) divide el tiempo en bloques de N ranuras, de tal forma que la ranura *i-ésima* se asigna a la estación *i*. Si una estación no tiene nada que transmitir en su ranura, ésta se queda vacía.

estación 1	estación 2	estación 3		estación N-1	estación N
ranura 1	ranura 2	ranura 3	ranura N-1	ranura N

Las desventajas de TDM son su poca flexibilidad (no puede satisfacer el caso de que una estación ofrezca un caudal elevado y otra un caudal pequeño) y el elevado retardo (una estación tiene que esperar su turno antes de poder transmitir su trama).

Sondeo o Polling

Con esta técnica, existe una entidad central que ejerce de **controlador**. El controlador se encarga de enviar un mensaje (*poll*) a cada uno de las estaciones, por turno, preguntando si la estación tiene alguna información que transmitir. Estos mensajes tienen una dirección destino, identificando la estación que se está direccionando. Cada estación conoce su propia dirección, y sólo responde cuando se le interroga con dirección, aún cuando recibe todas las interrogaciones generadas en el sondeo. Si la estación que se interroga tiene alguna información para enviar, envía los datos; pero si no devuelve un mensaje especial indicando que "rechaza el sondeo". El controlador generalmente controla a todas las estaciones de forma secuencial y cíclica.

Las desventajas que tiene esta técnica es la falta de robustez que implica una dependencia total con respecto al controlador, y el que se dedica una parte importante del caudal a enviar mensajes de control (sondeo y respuestas al sondeo).

Contienda

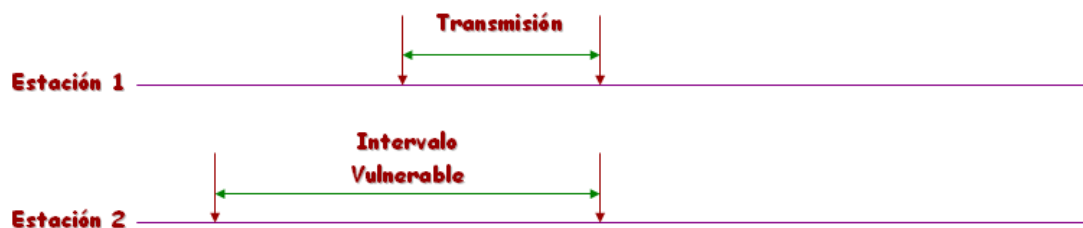
Las técnicas de contienda permiten que más de una estación transmita simultáneamente, con lo cual se puede producir una colisión. Aunque existen diferentes variantes, todas ellas se basan en un mismo principio: en caso de que se produzca una colisión, las estaciones involucradas dejan pasar un tiempo aleatorio antes de retransmitir la trama colisionada.

Aloha

El primer protocolo que se diseñó basado en la técnica de contienda fue el protocolo Aloha, diseñado por Abramson en la Universidad de Hawaii en 1970. El funcionamiento del protocolo Aloha es muy simple:

- La estación que desea transmitir lo hace y se pone a escuchar
- Si dos o más transmiten a la vez se produce una **colisión**
- En caso de colisión, el terminal retransmite la trama pasado un plazo de tiempo
- Para evitar una nueva colisión los plazos no son constantes sino aleatorios de media fija

Se define el periodo vulnerable como el periodo de tiempo con respecto a una trama en el que, si otra estación transmite, se produce una colisión. En Aloha, la duración del periodo vulnerable es $2T$, siendo T el tiempo de transmisión de una trama.

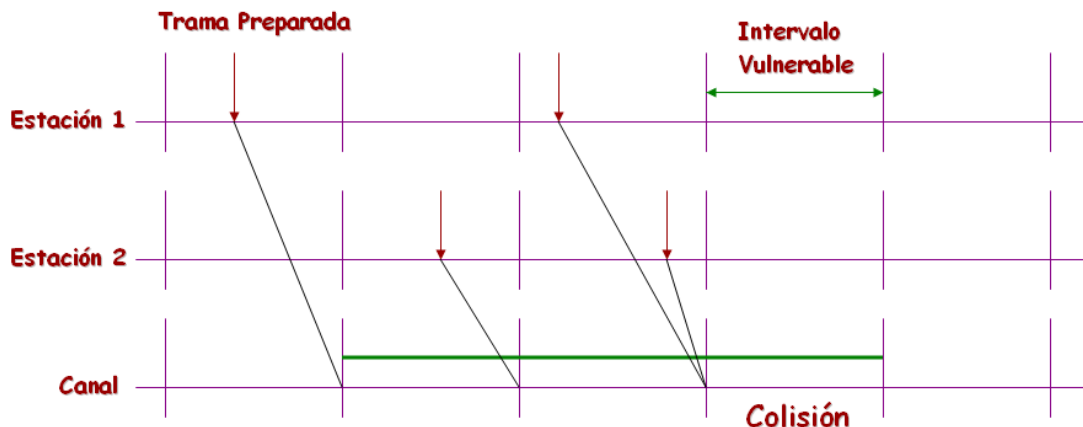


Aloha Ranurado

El protocolo aloha ranurado fue diseñado en 1972 por Roberts para mejorar las prestaciones del protocolo Aloha simple. La única modificación que se introduce con respecto al protocolo Aloha es que no se permite a las estaciones transmitir sus tramas en cualquier instante:

- El tiempo se divide en cuantos, denominados **ranuras**
- Sólo puede transmitirse al principio de cada ranura.
- La duración de una ranura es igual al tiempo necesario para el envío de una trama (i.e. T)

Con esta modificación, el periodo de vulnerabilidad queda reducido a la mitad, tal y como se muestra en el siguiente gráfico. Puede verse que esto permite aumentar la eficiencia hasta el doble.



Análisis de las prestaciones del protocolo Aloha Ranurado

Por simplicidad, asumimos $T = 1$, es decir, que la duración de una transmisión es de una unidad de tiempo. Si G es el número de intentos de transmisión que realiza una estación por unidad de tiempo², nos interesa conocer la eficiencia (el porcentaje de tiempo que el canal puede utilizarse para transmisiones exitosas).

La utilización S es igual a la probabilidad de éxito

$$S = \frac{N_{\text{éxitos}}}{N_{\text{total}}} = P_{\text{éxito}}$$

La probabilidad de éxito de una estación es igual a la probabilidad de que esta estación transmita y ninguna otra lo haga. Como cada estación transmite en una ranura con una probabilidad G , esto da

$$P_{\text{éxito, estación}} = G(1 - G)^{N-1}$$

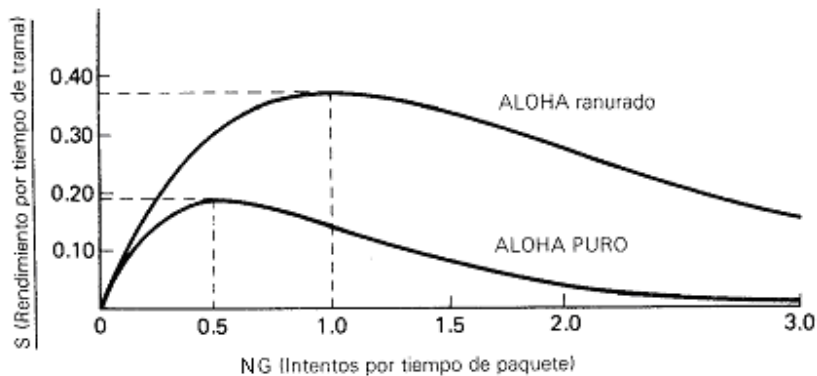
siendo N el número de estaciones en el canal.

La probabilidad de éxito de cualquier estación será la suma de las probabilidades individuales

$$P_{\text{éxito}} = NG(1 - G)^{N-1}$$

Si representamos S como función de NG para N muy elevado vemos que la eficiencia (S máxima) es del 36%. Con métodos similares puede verse que con Aloha la eficiencia es del 18%. NG , en las abscisas, representa el número total de intentos de transmisión por unidad de tiempo.

² El número de intentos de transmisión por unidad de tiempo, G , incluye tanto las transmisiones de tramas nuevas como las retransmisiones.



El gráfico anterior muestra que es importante que G (el número de intentos por unidad de tiempo) sea adecuado: si es demasiado pequeño o demasiado grande la utilización se aleja del máximo.

Para determinar el valor de G óptimo, tenemos que encontrar el valor que maximiza S . Para ello igualamos la derivada de S respecto G a 0:

$$\frac{dS}{dG} = \frac{d}{dG} (NG(1-G)^{N-1}) = N(1-G)^{N-1} - NG(N-1)(1-G)^{N-2} = 0$$

con lo que

$$(1-G) - G(N-1) = 0$$

y finalmente

$$G = 1/N$$

Es decir, cuanto más estaciones, menor tendrá que ser su tasa de intentos de transmisión para que la utilización sea óptima, lo cual resulta intuitivo.

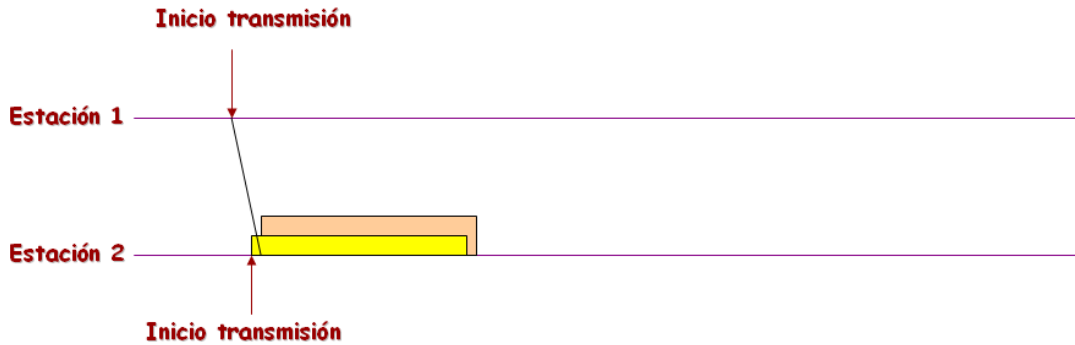
CSMA

CSMA significa Collision Sense Multiple Access (contienda con escucha) y básicamente la novedad que introduce con respecto al protocolo Aloha es que las estaciones tienen la capacidad de detectar si el canal se encuentra ocupado en un instante dado, en cuyo caso no transmiten.

Funcionamiento:

- Las estaciones tienen capacidad para conocer el estado del canal (libre u ocupado).
- Si "creen" que está libre, transmiten.
- Si detectan actividad, esperan.
- Si dos estaciones intentan transmitir al mismo tiempo se produce una colisión.

Lo que conseguimos con lo anterior es reducir el periodo de vulnerabilidad. Concretamente, una colisión sólo se producirá cuando, al iniciar una transmisión, otra estación haya iniciado una transmisión que aún no hemos detectado a causa del retardo de propagación. Es decir, el periodo de vulnerabilidad se reduce al retardo de propagación en la red. En caso de redes locales, con retardos de propagación pequeños, el periodo de vulnerabilidad será muy reducido y por lo tanto el protocolo resultante muy eficiente.



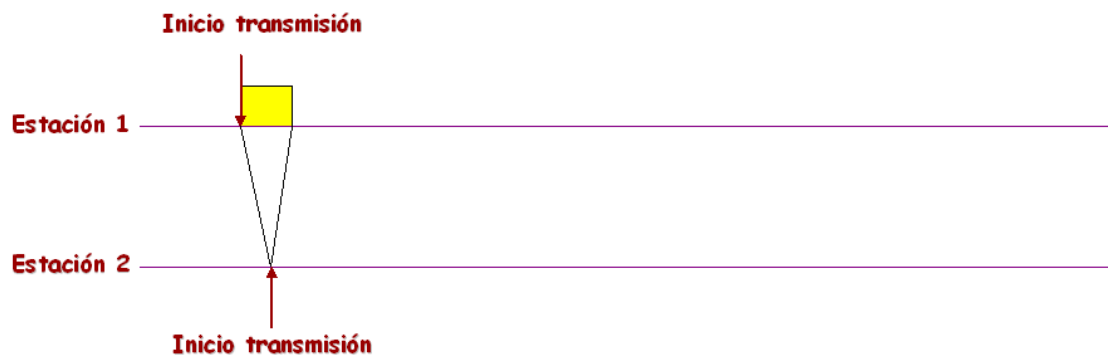
En caso de que una estación con una trama para transmitir encuentre el canal ocupado, ésta puede tomar diferentes decisiones, lo que define diferentes variantes del protocolo CSMA:

- CSMA 1 persistente. La estación escucha constantemente el canal hasta que queda libre y transmite en este momento. Si hay colisión, entonces esperará un periodo aleatorio.
- CSMA no persistente. No transmite en cuanto el canal queda libre sino que espera un periodo aleatorio.
- CSMA p persistente. En cuanto el canal se libera, transmite con probabilidad p , y espera un periodo aleatorio con probabilidad $1-p$.

CSMA/CD

CSMA/CD introduce la capacidad de detectar colisiones (CD = collision detection). Además de escuchar antes de transmitir, una estación en CSMA/CD permanece a la escucha durante la transmisión. Si se detecta una colisión, interrumpe la transmisión y espera un plazo.

Con esto lo que se consigue es que el tiempo perdido a causa de una colisión sea muy inferior, lo que supone una ganancia en eficiencia con respecto a CSMA. Concretamente, la duración máxima que va a tener una colisión será $2a$, siendo a el retardo de propagación en la red.



Comparación cualitativa

Comparación de Aloha, CSMA y CSMA/CD en base a:

- Intervalo vulnerable. Supuesta la existencia de una trama en el canal, intervalo posible de ocurrencia de colisiones:

$$\text{Aloha} > \text{CSMA} = \text{CSMA/CD}$$
- Tiempo perdido. Tiempo ocupación del canal desaprovechado por una colisión:

$$\text{CSMA/CD} < \text{Aloha} = \text{CSMA}$$

Como el intervalo de vulnerabilidad en CSMA y CSMA/CD es muy reducido, con estos protocolos se consiguen valores de U muy elevados, cercanos al 100%.

3.6. El protocolo HDLC

El protocolo HDLC es un protocolo de nivel de enlace extensamente utilizado en sistemas de comunicación que utiliza varias de las técnicas que hemos visto anteriormente.

Funcionamiento básico

Para satisfacer necesidades diversas, HDLC define tres tipos de estaciones y dos tipos de configuraciones básicas. Los tres tipos de estaciones son:

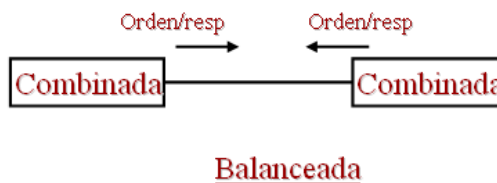
- **estaciones primarias.** Son las estaciones responsables de controlar la operación del canal, y son las únicas que pueden mandar *comandos*.
- **estaciones secundarias.** Operan bajo el control de una estación primaria, y mandan únicamente tramas *respuesta*.
- **estaciones combinadas.** Combinan las funciones de las primarias y las secundarias, y pueden mandar tanto *comandos* como *respuestas*.

Típicamente se consideran dos modos de transferencia para la operación del protocolo HDLC:

- **Modo de Respuesta Normal (NRM).** Se utiliza en caso de una configuración no balanceada, en la que tenemos una estación primaria que controla el canal y varias estaciones secundarias. En este caso se utiliza una técnica de **sondeo** para compartir el canal, en la que la estación primaria interroga a las secundarias, y las secundarias únicamente pueden transmitir datos como respuesta a los comandos enviados por la primaria.



- **Modo Balanceado Asíncrono (ABM).** Este modo se utiliza para configuraciones balanceadas con dos estaciones conectadas entre sí, en que ambas pueden mandar tramas sin permiso previo de la otra.



Formato de trama

La trama que se utiliza en HDLC está formada por los campos que se ilustran en la siguiente figura:



Flag

El **flag o guión** cumple la función de delimitar el inicio y fin de la trama. Consiste en la secuencia de bits *01111110*, que no puede reproducirse en el campo de datos puesto que en caso de que aparezcan 5 1's seguidos se añade un 0 en transmisión que después se quita en recepción (ver sección 4.2). Entre dos tramas podemos encontrar uno o más octetos de flag.

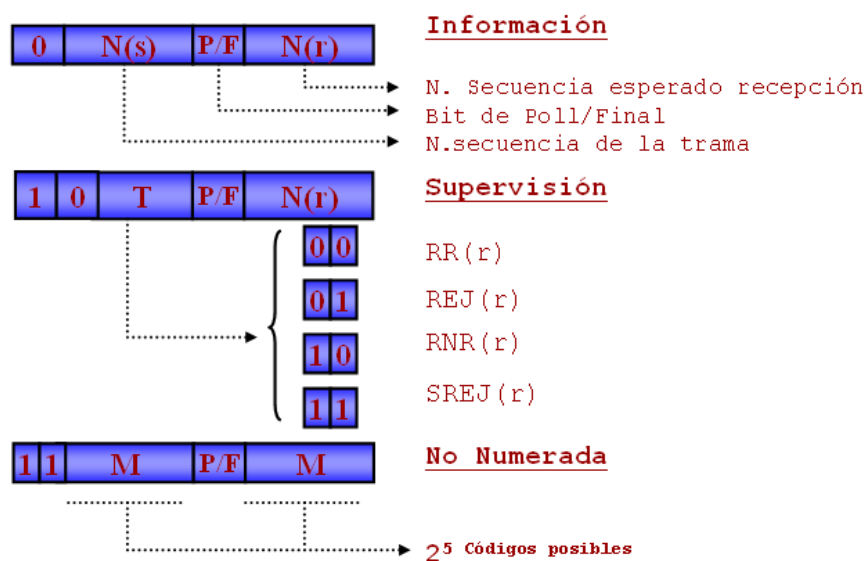
Dirección

En caso de que haya más de dos estaciones en el canal (modo de respuesta normal) es necesario indicar a qué estación va dirigida una trama. Para ello se utiliza el campo *dirección*, que normalmente tiene una longitud de un octeto (8 bits) aunque puede en ocasiones tener una longitud mayor. Este campo no es necesario en el caso de enlaces punto a punto (modo balanceado asíncrono), aunque se mantiene por uniformidad.

Campo de control

El campo de control puede ocupar 1 ó 2 octetos. Nosotros vamos a estudiar el primer caso.

Distinguimos tres tipos de trama: las tramas de **información**, las de **supervisión** y las **no numeradas**.



Las **tramas de información** se utilizan para transportar los datos de usuario. En el campo de control envían la información necesaria para el ARQ: el número de secuencia de la trama enviada, N(S), y el siguiente número de secuencia esperado en recepción, N(R).

Las **tramas de supervisión** sirven para proporcionar la funcionalidad ARQ en ausencia de datos para mandar. Con una trama de supervisión, mandamos el número de secuencia esperado en recepción y el siguiente código:

- RR: receive ready
- RNR: receive not ready
- REJ: Reject
- SREJ: Selective reject

lo que permite implementar tanto la técnica ARQ de rechazo simple (REJ) como la de rechazo selectivo (SREJ). El que se use una u otra técnica va a depender de la implementación concreta.

Por último, las **tramas no numeradas** son las que permiten el intercambio de información de control, como por ejemplo la información que se intercambia en al establecer una nueva conexión, eligiendo entre otras cosas el modo de operación, o para liberarla. Entre otras, encontramos los siguientes tipos de tramas no numeradas:

- SNRM: establecer una conexión en modo de respuesta normal
- SABM: establecer una conexión en modo balanceado asíncrono
- UA: unnumbered acknowledgement (asentimiento no numerado)
- DISC: liberar la conexión

Campo de información

Este campo únicamente se encuentra presente en las tramas de información y contiene los datos de usuario.

FCS (Frame Check Sequence)

Este campo contiene un CRC de 16 bits y sirve para detectar errores.

Funcionamiento

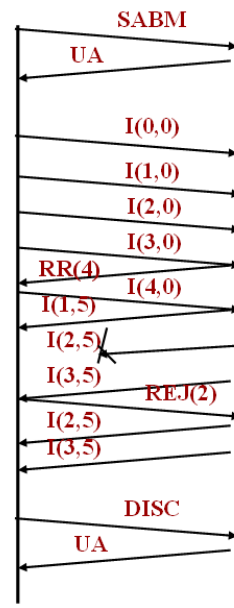
Modo Balanceado Asíncrono

Funciona como el ARQ

◆ Establecimiento del enlace

◆ Transferencia de datos

◆ Desconexión

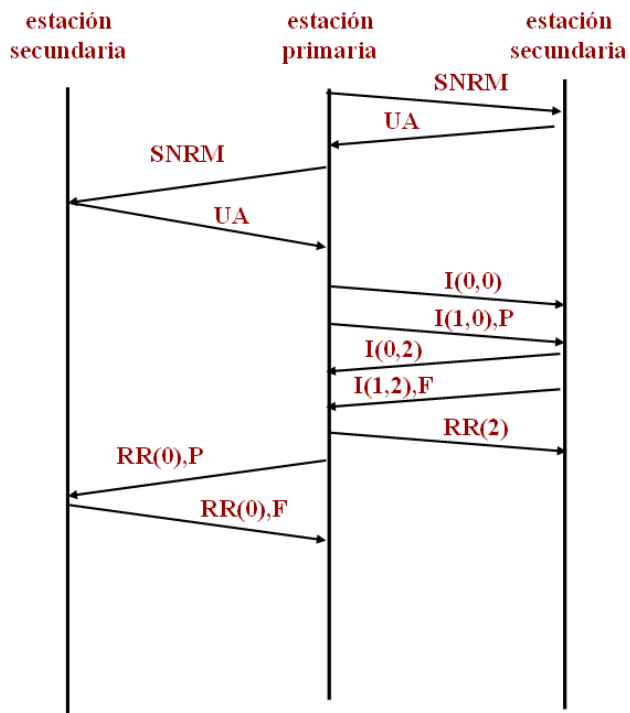


Modo de Respuesta Normal

En modo NRM, la estación primaria repite periódicamente el siguiente ciclo de sondeo para cada estación secundaria:

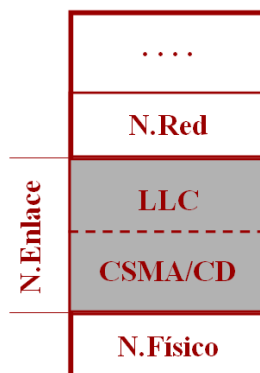
- manda todas las tramas que tenga para mandarle
- le da a la estación secundaria la oportunidad de transmitirle las tramas que tenga antes de pasar a la siguiente estación (sondeo).

El sondeo se realiza mandando una trama (de información o de supervisión tipo RR en caso de que la estación primaria no disponga de datos para mandar) a la estación secundaria con el bit Poll activado. La secundaria responde con una o más tramas de información, con el bit Final activado, la última de ellas.



3.7. El protocolo Ethernet

El protocolo Ethernet es el más extensamente utilizado para redes de área local. Para el control de flujo y de errores utiliza el protocolo LLC que es parecido al HDLC. Para la funcionalidad MAC utiliza el protocolo CSMA/CD.



Nivel LLC

El nivel LLC (Logical Link Control) en Ethernet cumple la funcionalidad de control de flujo y errores en redes Ethernet. Para ello utiliza técnicas ARQ parecidas a las que hemos visto con el protocolo HDLC. A diferencia de HDLC, LLC no implementa el control de acceso al medio (HDLC lo hacía definiendo estaciones primarias y secundarias y utilizando técnicas de contienda). LLC no distingue entre estaciones primarias y secundarias y descarga la funcionalidad MAC al nivel inferior, que implementa el protocolo CSMA/CD.

LLC permite básicamente dos modos de operación:

- orientado a conexión con confirmación: se mantiene estado para la conexión, y se retransmiten mediante técnicas ARQ las tramas erróneas.
- no orientado a conexión sin confirmación: no se mantiene estado ni se asienten/retransmiten tramas.

Nivel MAC (CSMA/CD)

El nivel MAC de Ethernet utiliza el protocolo CSMA/CD para arbitrar el acceso al medio.

El formato de trama es el siguiente:

Bytes: 7+1	2 o 6	2 o 6	2	0-4	0-1500	0-46	4
Preámbulo y 1 de SFD	D.Destino	D.Origen	Longitud	LLC	Datos	PAD	CRC

Preámbulo: es un patrón de 7 octetos que alterna 0s y 1s y que se utiliza para sincronizar el transmisor y el receptor a nivel de bit

SFD (Start Frame delimiter): consta de la secuencia 10101011, y permite al receptor conocer cuál es el primer bit de la trama.

Direcciones origen y destino: son identificadores únicos de las estaciones de origen y destino. Pueden constar de dos o de seis octetos, según la implementación.

Length: indica la longitud del campo LLC.

LLC: consta de 4 octetos en modo orientado a conexión y de 0 en modo no orientado a conexión.

Pad: añade los octetos necesarios para conseguir una longitud total de la trama mínima de 64 octetos. Esta longitud mínima es necesaria para garantizar el comportamiento correcto de la detección de colisiones.

FCS: CRC de 32 bits para detectar errores.

El caudal físico de una red Ethernet puede ser de 10 Mb/s (10^6 bits/segundo), en el caso de la Ethernet típica, o de 100 Mb/s, en el caso de Fast Ethernet.

Referencias

[1] William Stallings, "Data and Computer Communications", Prentice Hall.

[2] Andrew S. Tanenbaum, "Redes de Ordenadores", Prentice Hall.