



Protocolo HDLC: Marco teórico y escenarios

Objetivo y alcance

El objetivo de este marco teórico es enfatizar los elementos del protocolo que nos permitirán abordar la práctica; no pretende ser exhaustivo. Nos concentraremos sólo en el modo que consideramos más *eficaz* para simular el tráfico, que es el Modo Normal de Respuesta, al que usaremos con Módulo 8. Con él, veremos intercambios sencillos de tramas sobre escenarios sencillos, con el propósito de poder simular el tráfico. A estas simulaciones las mostraremos con un flujograma. Así llamamos a un diagrama que muestra el tráfico con escala de tiempos.

Sobre la nomenclatura

En este documento hemos seguido la siguiente nomenclatura

- *Link* Las palabras en itálica son de otro idioma o están en lenguaje propio de la disciplina.
- HDLC En mayúsculas se expresan las siglas.
- DIRECCIÓN En versales se identifican los nombres de campo.
- **DISC** En mayúsculas+negrita se indican los nombres que reciben las tramas.
- **Eficiencia** En negrita se expresan los conceptos nuevos o que se desean destacar

Marco Teórico

HDLC significa *High-level Data Link Control* o sea Control de Enlace de Datos de Alto nivel y como su nombre indica, es un protocolo de enlace que ofrece un control minucioso de la capa de enlace de datos. Este protocolo se encuentra totalmente descrito en la norma IS13239 de la ISO, y es de hecho el de referencia cuando se relaciona la Capa 2 en el modelo de Capas OSI de la ISO; dio origen a una gran cantidad de protocolos que conservan sus principios de trabajo y, en general, sus algoritmos, es decir su máquina de estados.

HDLC contempla dos modos de trabajo claramente distintos:

- Modo entre pares, llamado también modo **balanceado**
- Modo subordinado o llamado también modo **normal**.

El primero se denomina Modo Asíncrono Balanceado o ABM (por *Asynchronous Balanced Mode*) y usa una comunicación dúplex en la que ambos extremos hablan y escuchan simultáneamente, y cualquiera de ellos puede administrar la conexión; es el más usado actualmente.

El segundo se denomina Modo Normal de Respuesta o NRM (por *Normal Response Mode*) y usa una comunicación semidúplex en la que siempre hay un lado maestro o primario y otro esclavo o secundario, y un extremo habla o escucha pero no ambas cosas a la vez.

Existe un tercer modo llamado “de respuesta balanceada” o ARM (por *Asynchronous Response Mode*) en que la comunicación es dúplex y un extremo puede hablar y escuchar simultáneamente pero el nodo primario o maestro administra la conexión. Este modo se encuentra en la práctica casi no usado.

Usaremos el modo normal de respuesta o NRM y llamaremos nodo (A) al lado maestro o primario y al otro (B), que es el esclavo o secundario, quien no transmite si no es sondeado. (A) direcciona a (B) obligándolo a escuchar cuando necesita hablarle; cuando (A) terminó de enviar lo que tenía en *buffer* y en ventana de envío, sondea a (B) (sondeo = *poll*) para escucharlo, lo que equivale a pasarle el turno para hablar. Cuando (B) termina de enviar lo que tenía en *buffer* y en ventana de envío, sondea a (A), y así se desarrolla el control del tráfico, como si se pasaran entre ellos un *token* (un testigo, un objeto que sirve de señal) que habilita al otro a hablar.

En este modo, el protocolo trabaja orientado a la conexión y confiable, utiliza un módulo M=8 (modo básico) o M=128 (módulo extendido o /E) y más recientemente M=32.768 aunque solo en algunas implementaciones. Usa ventana corrediza, una para transmitir y otra para recibir, que pueden ser o no iguales. No admite que la “línea” quede sin gestionar, es decir que mientras no hay tráfico de datos los nodos se intercambian tramas de control



que mantienen activo el estado de la comunicación, lo que se logra haciendo que un nodo le informe al otro si está o no listo para recibir. Si tienen datos para transmitirse y además hay temporizadores definidos y activos entonces se transmiten al compás de los temporizadores, caso contrario se sondean de inmediato.

La estructura de tramas

Este protocolo está orientado a bits, por lo cual carece de sentido alinear a octetos la cadena de información entramada. Todas las tramas sin excepción están limitadas por banderas (*flags*) de inicio y fin (en la jerga, FLAG.I por bandera inicial y FLAG.F por bandera final) cuyo patrón binario es $(7E)_X$. Además de las banderas, las tramas todas tienen un campo DIRECCIÓN y uno CONTROL y en el caso de las tramas I además tienen un campo INFO (algunas tramas U también tienen campo INFO) donde va la carga útil. A ésta se le aplica *Zero Insertion* (un cero incondicional después de 5 bits 1, retirado en destino) para romper los patrones que se asemejen a un *flag*.

Se detalla la estructura de la trama I para $M=8$. Si fuera distinto de 8, el campo CONTROL sería de otra longitud:

0111 1110	aaaa aaaa	cccc cccc	Campo de carga útil de longitud variable	sssss ssss ssss sss	0111 1110
FLAG.I	DIRECCIÓN	CONTROL	INFO	FCS	FLAG.F

Las demás tramas no tienen campo INFO, a excepción de algunas tramas U.

Los 8 bits que siguen al campo FLAG.I constituyen el campo DIRECCIÓN y tienen el sentido de campo de dirección de capa 2, útil para topología PMP. Si el enlace es P2P, DIRECCIÓN siempre contiene 0 binarios. Si el enlace en cambio es PMP entonces contiene la dirección binaria del destinatario (trama desde A hacia los múltiples nodos B_n) o del remitente (trama desde algún nodo B_n hacia A) aunque si la trama es del tipo broadcasting se fuerza a valor de ocho 0 binarios.

Los 16 bits previos al FLAG.F contienen el CRC en el campo llamado FCS (por *Field Check Sequence*) calculados según el generador CCITT-16 ($M=8$ y $M=128$) o CCITT-32

Las tramas de tipo I

Son las utilizadas para el intercambio de información entre los nodos. Con Módulo 8, los 8 bits del campo CONTROL contienen:

- El primer bit con 0 binario que indica que esta trama es tipo I;
- Los tres bits siguientes con el número de secuencia enviado (el de la propia trama) llamado campo N_s ;
- Luego un bit llamado campo P/F o de sondeo/final; se usa para cambiar el sentido del flujo cuando está puesto a uno binario (este lado terminó el envío y ahora escucha) y es ignorado cuando está puesto a 0 (esta trama es inicial o intermedia);
- Los últimos tres bits con el número de secuencia que se espera recibir del otro nodo, llamado secuencia N_r ; este campo es un “Acuse de recibo implícito” de la trama cuya secuencia es N_r-1 , acción que es llamada generalmente *piggybacking*

La longitud de tres bits de los campos N_r y N_s determinan el $M=8$, y éste resulta en $Max_Seq = 7$ siendo este parámetro el más alto número de trama posible ($N_r = 111$), lo que a su vez determina $Max_Window = 7$ (es decir $M-1$) siendo este límite lo que evita la ambigüedad entre el direccionamiento y los acuses.

Los bits restantes hasta el FCS son la carga útil de capas superiores y constituyen un campo de longitud variable. Recuérdese que HDLC puede contener en la carga útil: una fracción, uno entero o varios paquetes de red, con una cantidad arbitraria de bits.

Las tramas de tipo S

Son las utilizadas con propósito de control. Le siguen a DIRECCIÓN 8 bits que, en orden, son:

- Dos bits con el valor binario 10 indicando que es una trama S;
- Dos bits que dan lugar al tipo de comando de control de flujo: Los cuatro comandos pueden ser:
 - **RR** (indica listo para recibir); es de hecho un “Acuse de recibo explícito” y también se lo llama ack;



- **RNR** (No listo para recibir); también es un acuse de recibo explícito; funciona como ack
- **SRJ** (Rechazo selectivo de Nr); funciona como un “Acuse Negativo” y también se lo llama nack;
- **RJ** (Rechazo y regresar a Nr); funciona como un “Acuse Negativo” desde Nr en adelante.

Nótese que ambos nack, **RJ** y **SRJ**, son dos modos de trabajo mutuamente exclusivos.

- Un bit para el campo P/F; éste es forzado a 1 para todos los casos en MNR.
- Tres bits para Nr. Para los ack contiene el número de secuencia de la próxima trama esperada. Si es un rechazo selectivo indica qué trama se rechazó; si es rechazo no selectivo indica a qué número de secuencia se debe regresar y reacomodar la ventana de envío.

Para simular el tráfico, salvo que se especifique lo contrario en algún ejercicio, usaremos sólo acks. Esto implica que nuestro modelo de protocolo no usará rechazos selectivos (**SRJ**).

Respecto al otro nack, la trama **RJ** tiene un uso equivalente a un **RR** o **RNR**. Esto sumado a que no se admitirán **SRJ** permite trabajar sólo con los acks.

En resumen: se puede controlar perfectamente el flujo usando sólo **RR** y **RNR** en las que Nr representa un acuse de recibo explícito de las tramas pendientes de confirmación con números de secuencia hasta Nr-1 incluidos. De hecho, nuestras simulaciones contendrán sólo estas tramas.

Las tramas de tipo U

Éstas contienen las secuencias de comandos y respuestas (C/R). Con el protocolo trabajando en Modo Normal de Respuesta los comandos son emitidos sólo por el lado maestro (A) y reciben el nombre de Indicadores (*indicator*). El nodo que no puede gobernar el enlace (B) emite solicitudes (*request*), también llamados requerimientos. Luego de DIRECCIÓN, en orden se encuentran:

- Dos bits con el valor binario 11 indicando que es una trama U;
- Dos bits con el código que identifica al C/R;
- Un bit para el campo P/F éste es forzado a 1 binario para todos los casos en MNR;
- Tres bits que se llaman campo MODIFICADOR y, entre otras funciones, indica si sigue un campo de información que es usado por la capa de enlace únicamente.

El campo MODIFICADOR es útil para indicar los parámetros de negociación para el establecimiento de sesión que pueden ir contenidos en un **XID** y para indicar que hay información de una trama inesperada (fuera de secuencia, ininteligible, etc.) que se rechazó mediante el comando **FRRJ** (*frame reject*). También es usado en una trama **UI** (*unnumbered information* o información no numerada) para intercambiar información entre los agentes de capa 2 que puede o no ser propietaria, es decir propia de la implementación. Una trama UI no es acusada ni retransmitida.

Las secuencias de C/R se pueden agrupar de la siguiente manera:

- Inicialización y fijación del modo:
 - **SIM/RIM** *Set Initialization Mode/Request Initialization Mode*
 - **SNRM/E, SABM/E** *Set Response Mode (Normal o Asynchronous Balanced)* normal o extendido
 - **UA** *Unnumbered Acknowledgement*
- Identificación y negociación:
 - **XID** *Exchange Identification* [con datos para negociar] (recibe XID)
- Desconexión:
 - **DISC** *Disconnect*
 - **RDISC** *Request Disconnect*
 - **DM** *Disconnect Mode*
 - **UA** *Unnumbered Acknowledgement*
- Control de flujo urgente:
 - **FMRJ** *Frame Reject* [con datos de la trama rechazada]



- **UP** *Unnumbered Poll*
- **RSET** *Reset*
- Intercambio de información de capa 2 urgente:
UI *Unnumbered Information* [con datos] (no se confirma)

Modo de funcionamiento

Los nodos atraviesan tres fases de trabajo: conexión, intercambio de tramas y desconexión.

En la fase de conexión, se hace contacto entre ambas partes (**SIM/UA**), se fija la variante del protocolo (en nuestro caso **SNRM/SNRME /UA**) y opcionalmente, se negocian los parámetros según las capacidades de cada nodo (**XID/XID**). Cuando termina esta fase los nodos quedan conectados y listos para intercambiar tramas.

Durante la fase de intercambio se intercambian, si hay, datos de manera confiable controlando el flujo y manteniendo activo el enlace según las reglas del modo fijado (**RR, RNR**). Los nodos pasan alternativamente del ciclo *listen* (que en rigor incluye escuchar la línea y recibir por ella) al ciclo *send* en el que envían lo que tienen en *buffer* de envío y dentro de la ventana de Tx. Un nodo deja de estar en modo *listen* cuando encontró el fin de la ventana de recepción, o cuando un temporizador (T1) le indica que deje de esperar que llegue una trama, o recibió un sondeo explícito. En cambio, deja de estar en modo *send* cuando agota lo que tenía en *buffer* de envío dentro de la ventana de Tx o envió un sondeo explícito.

Cuando no hay más intención de mantenerse conectados, pasan a fase desconexión en la cual los nodos se desconectan mediante solicitud (**RD**) o indicación (**DISC**) y se confirman el éxito de la operación (**UA**). Eventualmente pueden informarse que están desconectados cuando se envían el informe de estado (**DM**)

Sobre los temporizadores y las latencias

Los temporizadores no están incluidos en el estándar y son totalmente dependientes de la implementación. La mayoría de las diversas variantes de HDLC tienen en común un temporizador de actividad de línea (T) que acompaña el ritmo de sondeo mientras no hay datos; de todos modos, convencionalmente el valor $T=0$ determina que el temporizador no se use y eso define que mientras no hay tráfico, (A) y (B) intercambian **RR** (o eventualmente **RNR**) sin dilaciones, al ritmo de la latencia. Esto debe ser especialmente tenido en cuenta para las simulaciones ya que es el modo de trabajo que adoptamos.

También debe ser especialmente tenido en cuenta el modelo de cálculo de latencias, que se encuentra en el repositorio de la Cátedra bajo el título de “Latencia en los enlaces”. No se podrá calcular adecuadamente los temporizadores, incluso costará entenderlos, sino se ha entendido primero conceptualmente como incide la latencia de un enlace y cómo la calculamos.

Para nuestras simulaciones, usaremos tres temporizadores simétricos, es decir que se encontrarán en ambos nodos. Para entenderlos, debe tenerse presente que un temporizador **arranca** planificadamente ante un evento y **para** (se detiene) cuando se cumple un evento esperado para el cual fue programado, o **expira** si llegó al total del tiempo previsto y el evento no se cumplió. Los temporizadores que usaremos son:

T1, que define cuánto tiempo el lado receptor va a esperar que llegue por la línea una trama que se está esperando, y ayuda a cerrar la ventana de recepción (evita el deadlock si la trama esperada no llega).

Nombre:	T1 o Temporizador de Cierre de ventana de recepción
Propósito:	Limitar la espera del arribo de una trama I en modo <i>listen</i>
Corre en:	El lado receptor
Arranque:	Terminada de demodular la FLAG.F de la trama I que acaba de llegar, excepto que $P/F = 1$.
Parada:	Con la siguiente trama I recibida completa y en secuencia
Reintentos:	No genera
Excepción:	Si la trama recibida está fuera de secuencia la ignora y continúa corriendo
No corre:	Cuando se reciben tramas S o U
Al expirar:	Se da por terminado el ciclo <i>listen</i> (se cierra la ventana de Rx) y se pasa al ciclo <i>send</i> haciendo descarte silencioso de lo que siga llegando. Inmediatamente después del cierre, si hay un T2 se ejecuta; si no, se envía lo que hay en <i>buffer</i> o en caso de <i>buffer</i> vacío se envía un ack. Si expiró con una trama fuera de secuencia o inesperada el nodo contesta con FRRJ
Cálculo:	Puede ser arbitrariamente mayor que la modulación de la trama más larga y depende de la tasa de arribo de tramas



T2, que define cuánto tiempo el lado receptor esperará que un paquete de red arribe a la capa de enlace y se convierta en una trama dispuesta para envío en su *buffer*, una vez que la ventana de recepción se cerró:

Nombre:	T2 o Temporizador de <i>piggyback</i>
Propósito:	Limitar el tiempo de espera de un paquete para entrarlo
Corre en:	El lado receptor
Arranque:	Con el cierre de ventana de Rx (P/F = 1 en I, o T1 expirado o W alcanzado)
Parada:	Con el primer paquete entrado y listo en <i>buffer</i> de envío, dentro de ventana de Tx.
Reintentos:	No genera
Excepción:	No tiene
No corre:	Cuando hay tramas listas (aunque sea 1) esperando para Tx y cuando el sondeo es explícito (trama S con P/F = 1)
Al expirar:	Se da por terminado el ciclo <i>send</i> , se envía una trama S y se pasa al modo <i>listen</i>
Cálculo:	Debería ser arbitrariamente mayor que el período de la tasa de arribo de paquetes

T3, que define cuánto tiempo el lado emisor esperará la confirmación de las tramas enviadas.

Nombre:	T3 o Temporizador de <i>Time-out</i>
Propósito:	Limitar el tiempo de espera de una respuesta válida que se ha requerido
Corre en:	El lado emisor
Arranque:	Terminada de modular la FLAG.F de cualquier trama I con P/F = 1 o de trama S
Parada:	Con la primera trama completa I o S con P/F = 1 y Nr correcto
Reintentos:	Según definición en la implementación y XID
Excepción:	En lugar del reintento puede generar un RSET
No corre:	Cuando se envían tramas I con P/F = 0, o tramas U
Al expirar:	A) genera reintentos, o b) genera SIM/RIM , o c) genera RSET
Cálculo:	Debe ser arbitrariamente mayor que la suma de dos latencias, más T2 más la modulación de la trama más larga

Dos aspectos para tener en cuenta en las simulaciones:

- Cuando un nodo arrancó T2, ese tiempo de espera vale sólo para la primera trama que se presenta. Sucesivas tramas dentro de la ventana de envío deberán estar ya en *buffer* de Tx para ser consideradas.
- Cuando a un nodo le expira T3, la acción que consideraremos es **RIM** o **SIM** según corresponda, salvo que en el escenario se indique algo específicamente distinto.

Sobre la eficiencia y la ocupación de la línea

La **eficiencia** y la **ocupación** que se usan para tomar decisiones y que se pueden calcular, no están contenidas en el estándar. Esos parámetros, usuales en la disciplina, se usan con ese nombre dado que son comunes en la jerga. No obstante, los analizadores y simuladores de tráfico pueden (y de hecho lo hacen) usar otros nombres para esos parámetros.

Una manera de medir la **eficiencia** es viendo cuántos bits útiles respecto a los bits totales se intercambian para un período dado. El período puede ser cualquiera que se preestablezca: desde que se contactan hasta que se desconectan, desde que comienza el tráfico de datos hasta que termina, un tramo arbitrario del intercambio de datos, etc. Por esa razón cuando se habla de la eficiencia (si no se dan precisiones) se habla de una eficiencia media. De todos modos, cuando hagamos los cálculos estableceremos claramente el período y lo calcularemos con el cociente indicado, teniendo en cuenta que el concepto de **bits útiles** corresponde a bits de la carga útil de enlace o sea los contenidos en el campo INFO de las tramas I no retransmitidas y el de **bits totales** corresponde exactamente a eso, es decir a la suma de bits que se intercambiaron durante el período en estudio, en todas las tramas intercambiadas en ambas direcciones. Se puede expresar como una tasa (menor que 1) o un porcentaje.

Una manera de medir la **ocupación** es viendo cuánto ha sido usado el enlace respecto a lo que se podría haber usado. Durante un período cualquiera (y el concepto de “período cualquiera” es igual que el anterior) un enlace podría estar ocupado tanto tiempo como dura el período mismo, más aún en métodos semidúplex. Eso se verá reflejado por la actividad del transmisor: transmisor activo, enlace ocupado. Sin embargo, como producto de que existe tráfico en ambos sentidos en enlaces con latencias y de que existen tiempos de espera merced a temporizadores y otros eventos, los transmisores no transmiten todos los bits posibles para el período sino sólo los que pueden. La tasa entre los bits transmitidos y los que podrían haberse transmitido para ese período nos da la ocupación, que también puede expresarse como un porcentaje.



Sobre el uso de la memoria

La memoria disponible para toda la sesión es la cantidad de memoria que el componente HDLC va a alocar de modo estático o dinámico para todos los *buffers* de Rx y de Tx. En condiciones de configuración real, si fuera necesario saber el tamaño de esta área, sino es un dato se deberá calcular.

El tamaño mínimo del área de *buffers*, sea para recepción o para transmisión, es el que permite asignar (alocar) la cantidad de *buffers* necesarios para atender la operación unitaria del ciclo de recepción o de transmisión

Tomemos el caso de la recepción. Un *buffer* contiene la trama recibida hasta que ésta se confirma y recién después de ese evento se limpia; pero la limpieza y puesta en disponibilidad ocurre en ciclos de varias tramas (como límite, W tramas), no por cada trama. Lo mismo ocurre con el de transmisión: contiene la trama que se envía hasta que ésta se confirma, y luego se limpia. Recuérdese que se confirma por lotes.

El tamaño del área surge de los valores de longitud de la carga útil y del tamaño de la ventana que se use. El área mínima de almacenamiento tiene que tener al menos W *buffers* de L bytes cada uno, donde W es el valor de ventana y L es la longitud total sin banderas (longitud de carga útil, más cabeceras, más cola y sin *flags*) de una trama I o una trama U del tipo **UI**, ya que los tramas S y U (excepto la **UI**) no se almacenan en *buffers* en memoria. Debe tenerse presente que si el ritmo de entramado es crítico, entonces deberá disponerse al menos de dos lotes de *buffers* de Tx y/o dos de Rx para poder trabajar sobre uno mientras otro está siempre disponible. Y finalmente, si la tasa de arribo de tramas o la tasa de arribo de paquetes son mayores que las que permiten desagotar los *buffers*, entonces se necesitará ir alocando dinámicamente más memoria.

Convenciones

Para este trabajo asumiremos una notación que permita estandarizar los flujogramas, de la siguiente manera:

Notación de Tramas I	Significado	Ejemplo
NS:x NR:z P/F	Se envía una trama I con número de secuencia x; acuse de recibo para las tramas recibidas con número de secuencia hasta z-1; 1 para el bit de final y nada si es trama inicial o intermedia	NS:2 NR:4 NS:3 NR:4 NS:4 NR:4 1
NS:[i,j,k] NR:z P/F	Uso opcional alternativo al anterior. Se envían varias tramas I cuyos números de secuencia son (en este caso) i, j, k; acuse de recibo las tramas recibidas con número de secuencia hasta z-1; 0 o 1 para el bit de sondeo o final	NS:[2,3,4] NR:41

Notación de Tramas SV	Significado	Ejemplo
RR:z	<i>Receive Ready</i> y acuse de recibo para la trama con secuencia z-1. Se asume P/F=1	RR:4
RNR:z	<i>Receive Not Ready</i> y acuse de recibo para la trama con secuencia z-1. Se asume P/F=1	RNR:3

Notación de Tramas U	Significado	Ejemplo
C	Significa un comando no numerado de (A)	DISC
R	Significa un requerimiento no numerado de (B)	RD
UA	Significa aceptación de comando no numerado	UA

Notación de Eventos	Significado	Ejemplo
(nnn,nn)	Instante en que la acción* se completó en unidades de tiempo preestablecidas con hasta dos decimales.	(72,1) (105,12)
[x]	Indica el número de secuencia de trama para enviar que está lista en buffer de envío.	[3,4,5] o [3] [4] [5] [4]



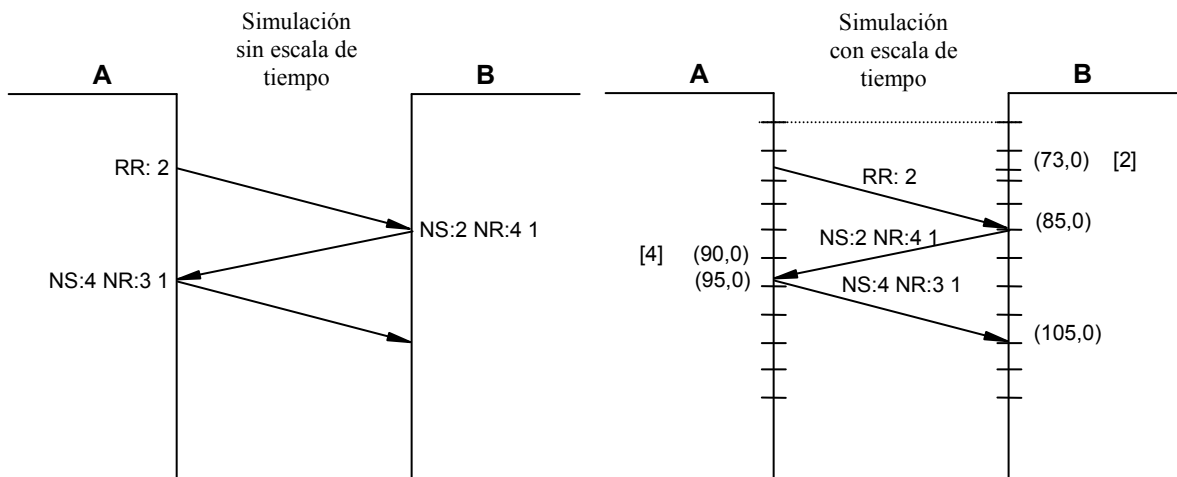
* Las acciones más comunes que se identifican, necesarias para la simulación, son: una trama está totalmente disponible para enviar; una trama terminó de ser recibida desde la línea; un temporizador arranca o expira, etc.

Ejemplo de Tráfico

Se debe tener presente que son necesarias estas estandarizaciones para identificar las tramas y los eventos y armar así la simulación.

Si no es necesario marcar tiempos porque, por ejemplo, los eventos se suceden unos a otros refiriéndose como eventos (algo pasa antes o después que alguna otra cosa), entonces simplemente se marcan los eventos sin escala de tiempo. No es necesario respetar ninguna escala, aunque si usar siempre la misma.

Pero si hay que hacer referencia al instante en que algo ocurre, los eventos se simbolizan sobre una línea del tiempo asociada a cada nodo, ambas corren paralelas y sincrónicas y los envíos de un lado al otro respetan la escala para que esos tiempos se vean reflejados:



El resto del documento

Se presentan aquí simulaciones de tráfico mediante flujogramas, para la comprensión del funcionamiento del protocolo.

Las negociaciones de parámetros que se hacen intercambiando **XID** generalmente no se usan, salvo que específicamente se indique. Si se usaran se le asignará al campo INFO para portar parámetros una longitud igual al de las tramas I. La misma longitud fija de asigna a las tramas UI.

Las tramas I las tramas U indicadas se usarán todas de longitud fija. Esto, si bien se aparta claramente de la especificación de HDLC, facilita el trabajo.

Asumiremos que la respuesta ack para un C/R contenido en una trama U es UA, mientras que la respuesta nack para una trama U es **FRMR**.

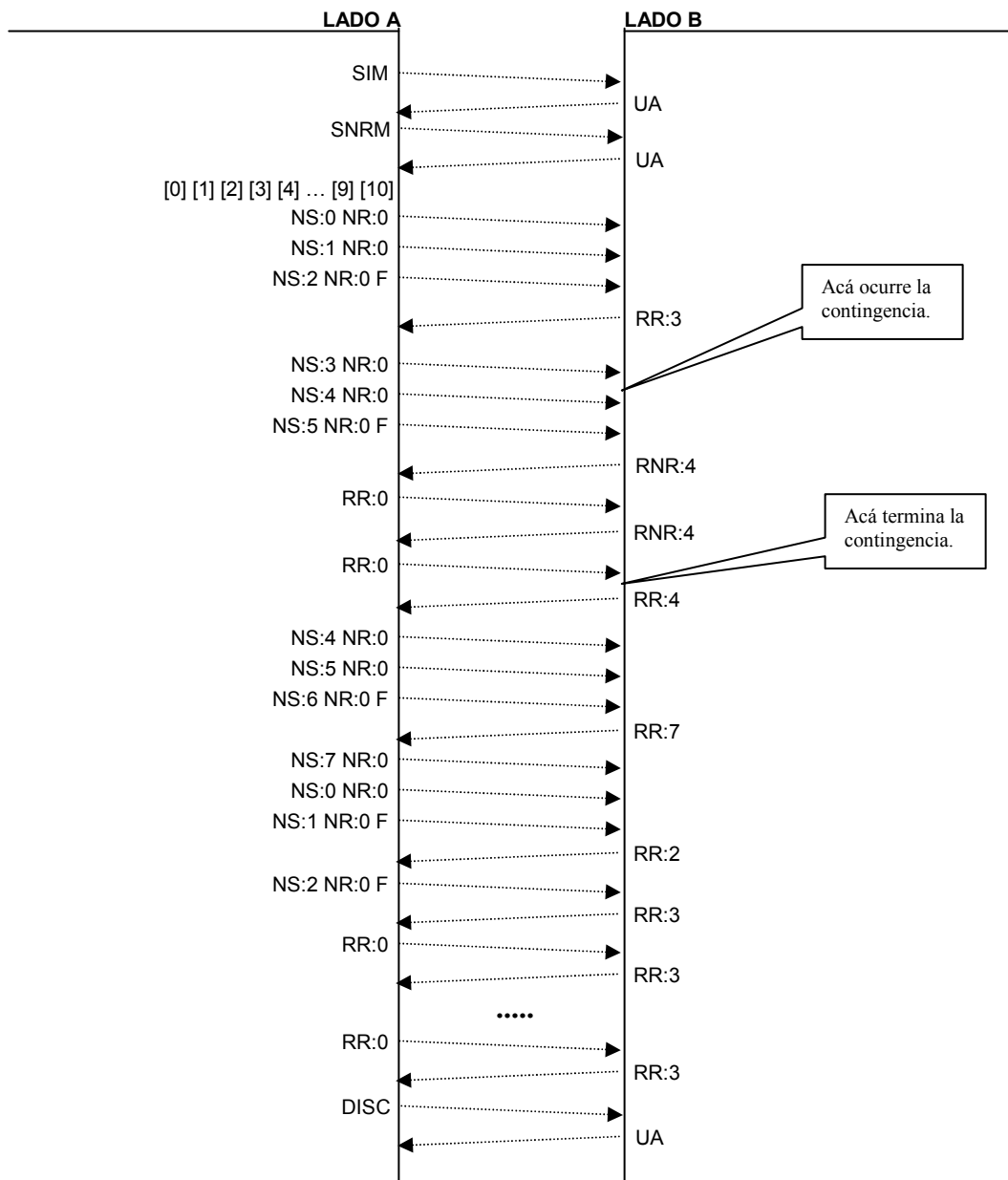
La utilización del bit del campo P/F se ha simplificado respecto al estándar, no usándose (es decir el bit no puesto) en las tramas que no invierten el flujo y el bit puesto en 1 significando que la estación del otro lado está siendo sondeada.

Finalmente, se observarán distintas redacciones para un mismo tipo de escenario. Se ha hecho con el propósito de acostumbrar en la lectura de un problema aún con distintas redacciones.



Ejercicio 1

Supóngase un intercambio con HDLC en modo Normal de Respuesta en la que el lado A está definido como ETD maestro. Se utiliza Módulo 8 y ventana 3 y no se usan temporizadores. La conexión está cerrada al principio y se cierra al final, y no se negocian parámetros de la conexión ni es necesario para los nodos identificarse. Al comenzar, el lado A ya tiene en *buffers* 11 tramas para enviar y el lado B ninguna, ni va a tener. El único inconveniente se produce cuando al lado B se le agotan los *buffers* después de recibir completa la cuarta trama; la contingencia dura todo el tiempo de recepción de las tramas quinta y sexta más dos sondeos, después de los cuales el lado B se recupera y continúa aceptando tramas. Agotadas las tramas para envío, continúa el sondeo hasta que A decide la desconexión.

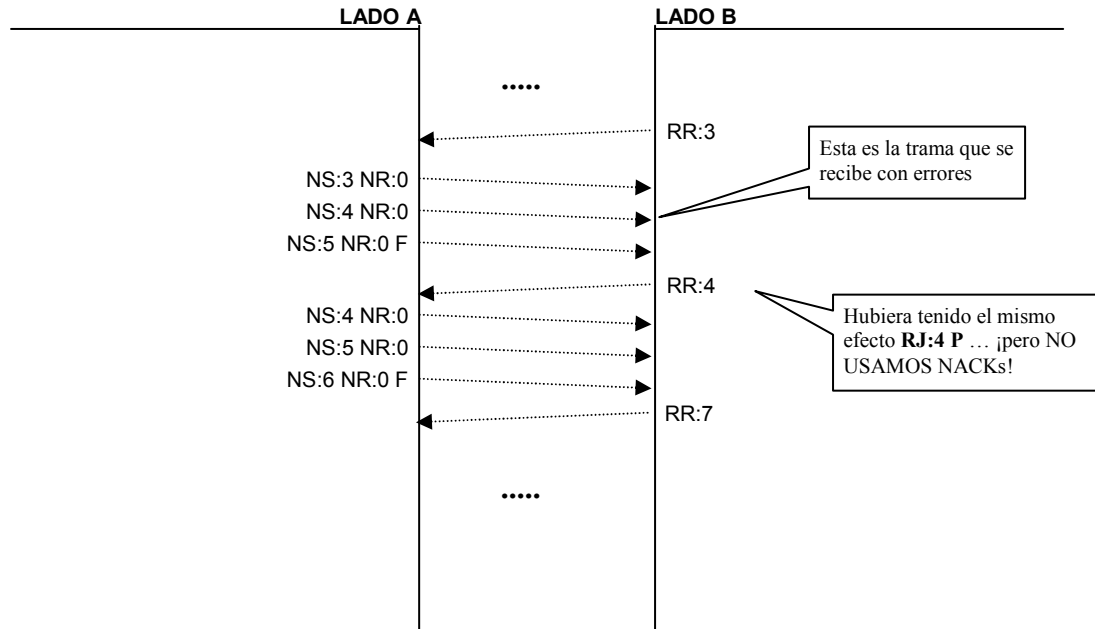


El flujograma está fuera de escala de tiempos. No hizo falta respetar dicha escala ya que no hay elementos precisos que determinar. Los puntos suspensivos significan que el flujo continúa igual hasta el siguiente evento mostrado.



Ejercicio 2

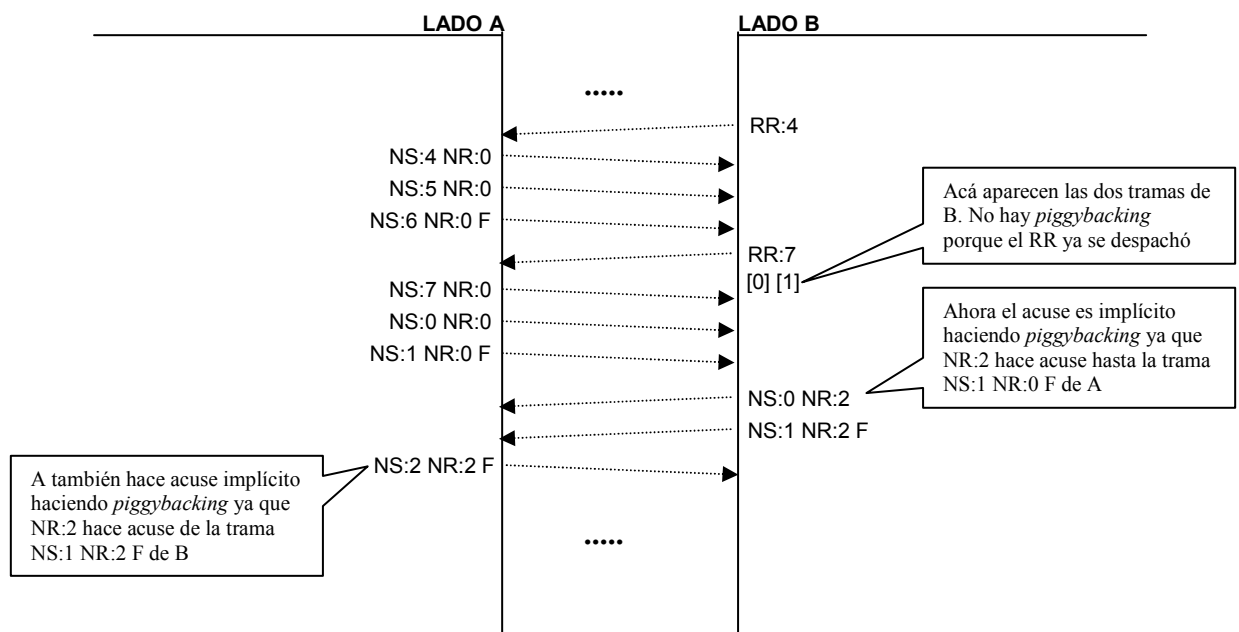
Mostrar cuál hubiera sido el efecto si en vez de agotarse el *buffer* de B después de la recepción de la cuarta trama, se hubiera producido un error durante la recepción de la quinta trama (por un mal CRC, por ejemplo).



Después de RR:7 de B hacia a A, el flujo terminará como está descrito en el Ejercicio 1.

Ejercicio 3

Para el escenario anterior, supóngase que aparecen en B, 2 tramas para enviar inmediatamente después de despachar el RR:7. Recuérdese que el flujo completo está descrito en el Ejercicio 1; nótese que B no está sondeado para hablar en el momento que aparecen las tramas.

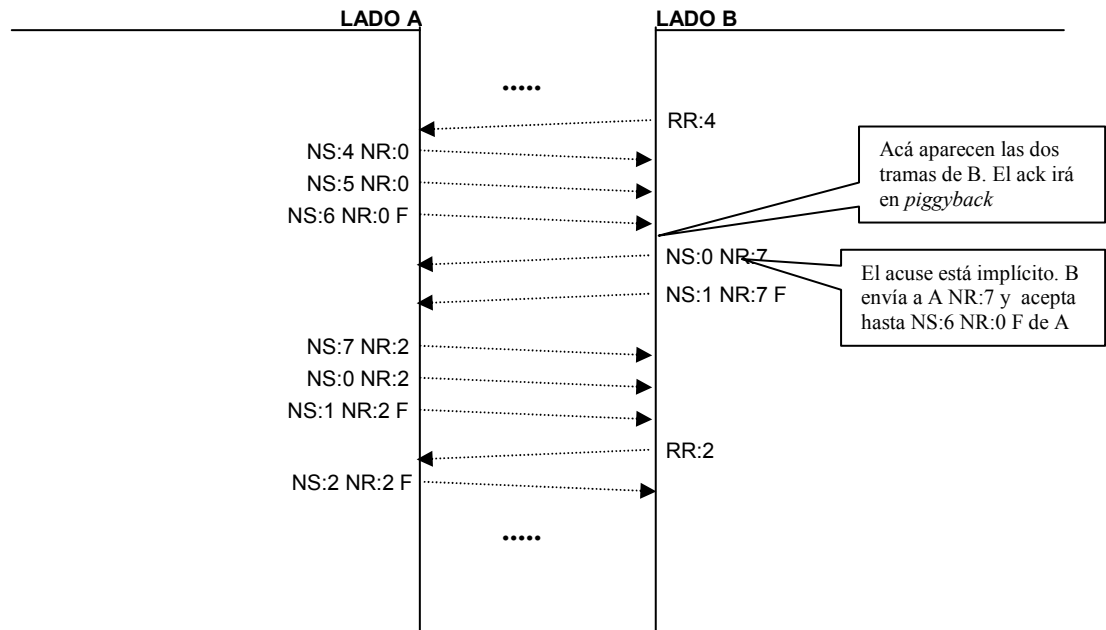


Después de NS:2 NR:2 F de A, el flujo terminará como está descrito en el Ejercicio 1.



Ejercicio 4

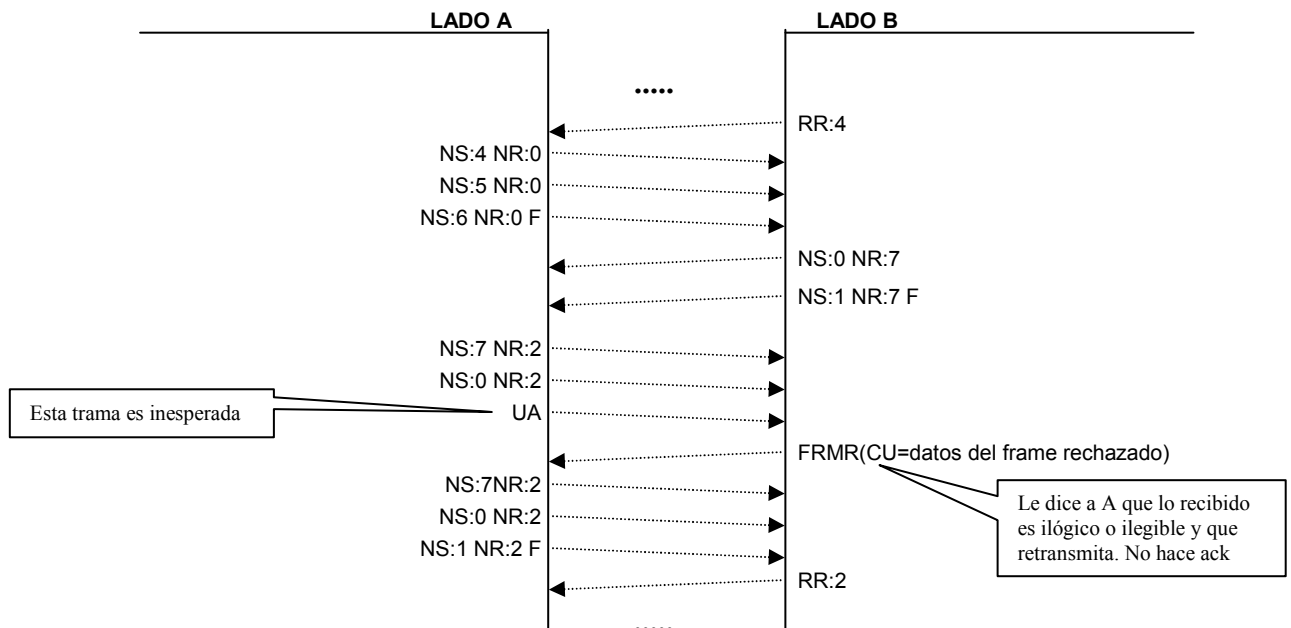
En el escenario anterior, supóngase que ambas tramas hubieran aparecido en B antes de que hubiera mandado el acuse explícito RR:7. Es decir inmediatamente después de la trama NS:6 NR:0 que A le envía a B. En tal caso, B contestará con su primera trama I con un acuse implícito en *piggyback*. Se muestra el flujograma sólo de esta porción del tráfico y se señala especialmente dónde habría un *piggyback* y dónde queda implícito el acuse.



Al igual que en el Ejercicio 3, luego de la trama NS:2 NR:2 F de A, el flujo termina como en el Ejercicio 1.

Ejercicio 5

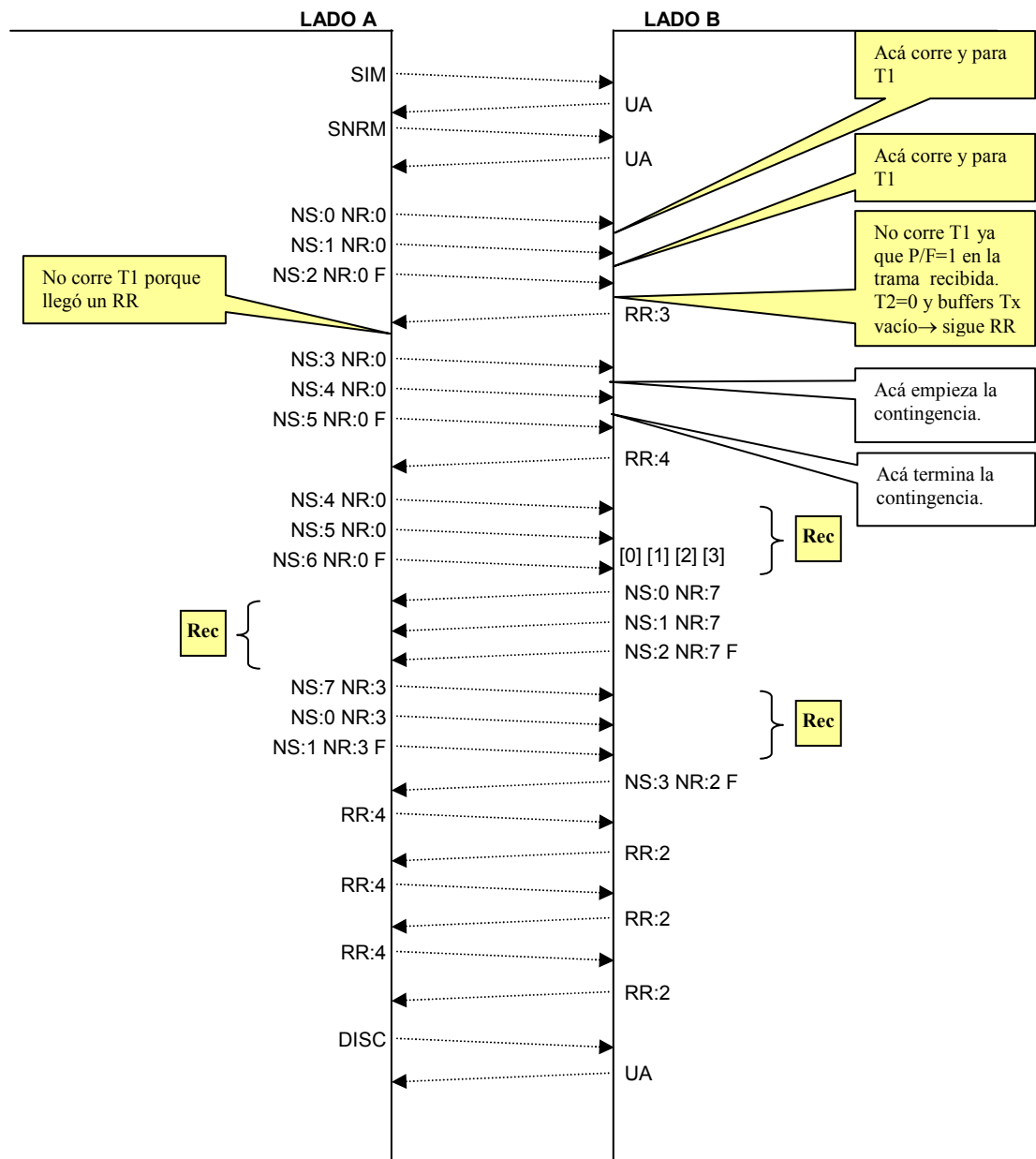
Ahora, y sobre ese mismo escenario anterior, imaginemos que por error el nodo A envía un UA después de la trama NS:0 NR:2. Obviamente, no es lo que B espera, ya que no emitió ningún C/R ni UI para esperar un UA. Se muestra cómo se resolvería la situación.





Ejercicio 6

Dos nodos se conectan con $W = 3$ y $M = 8$, y temporizadores $T1 = 50$ ms, $T2 = 0$ y $T3 = 0$. Inicialmente la conexión está cerrada y se abre y acepta sin necesidad de negociación ni identificación. El lado A comienza con 10 tramas en *buffers* listas para ser despachadas, a las que llamaremos tramas primera a décima, y B ninguna. Comienza la transmisión y cuando B terminó de recibir la trama cuarta de A, la aloja y a partir de allí se le agotan los *buffers* temporariamente. Esa contingencia dura un tiempo de recepción de trama, luego de lo cual la contingencia desaparece. Luego, en B aparecen cuatro tramas para transmitir después de haberse recibido bien la trama sexta de A. El intercambio continúa normalmente y termina el flujo después de que A sondea tres veces a B sin encontrar tramas I de respuesta, y cierra la transmisión. Se muestra el flujograma y se señalan los puntos más significativos.

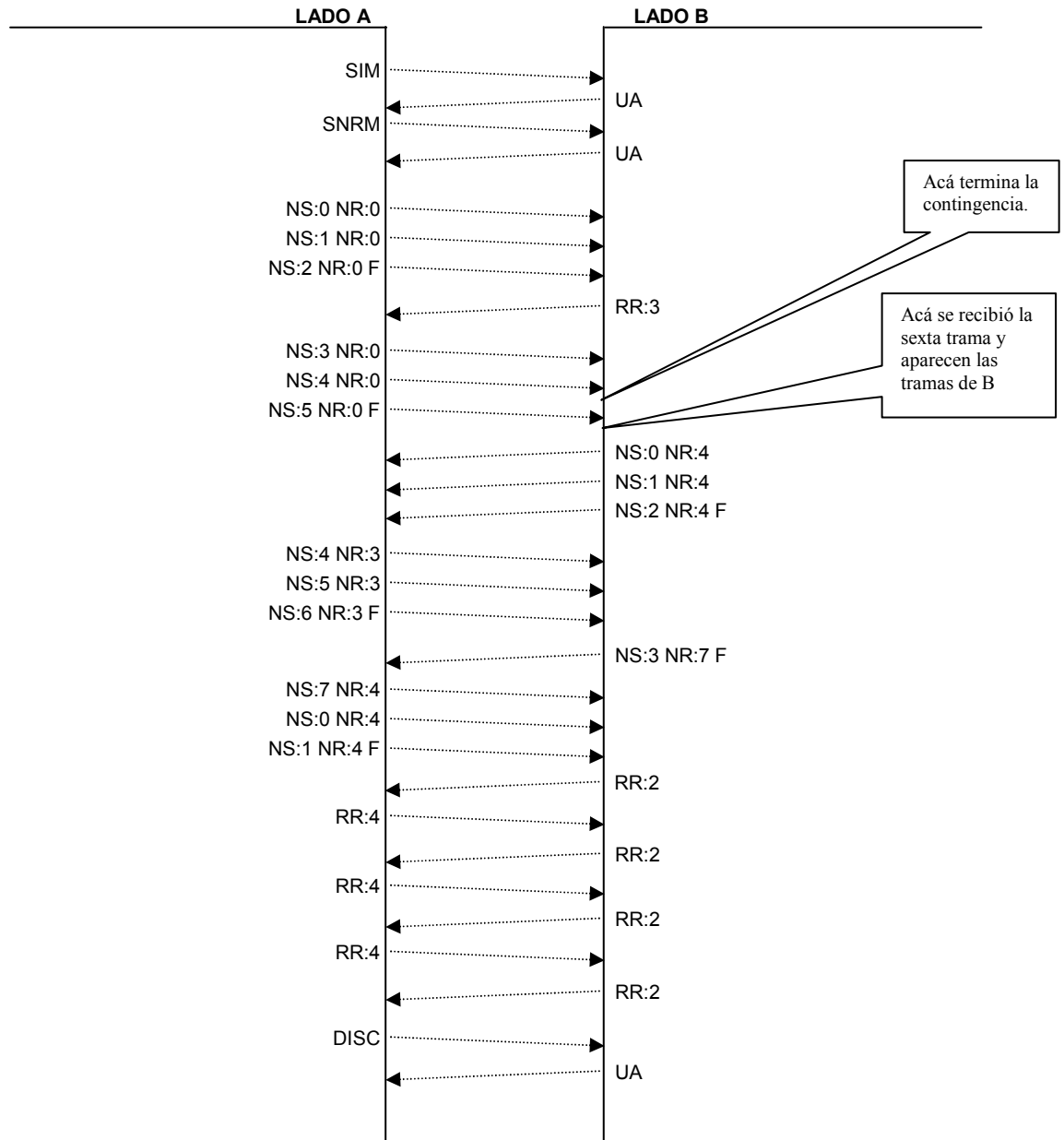


En cada una de las situaciones marcadas como Rec, se verifica lo mismo que para lo identificado en el flujograma con T1: arranca y para una primera vez, arranca y para una segunda vez y la tercera no arranca porque detecta un $P/F = 1$. De igual modo, cuando se recibe un RR o un RNR no arranca T1. Nunca se logra contar el T1



Ejercicio 7

Como variante, supóngase que en el Ejercicio 6 la aparición en B de tramas para transmitir, ocurre inmediatamente después de la recepción de la sexta trama de A aunque ésta se descarte por el problema de la contingencia. ¿Cómo sería el intercambio en este escenario? ¿cambiará?



En este caso, el flujo es distinto: En lugar de enviar B un acuse explícito informando que recibió sólo hasta la cuarta trama, envía ahora una trama de datos con el ack en *piggyback*.

Contemos las tramas y las latencias:

Cantidad de Latencias: 20

De A a B:

Tramas U cortas: 3 (SIM, SNRM, DISC)

Tramas U largas: 0

Tramas S: 3 (RR)

Tramas I: 12

De B a A

Tramas U cortas: 3 (UA)

Tramas U largas: 0

Tramas S: 5 (RR)

Tramas I: 4



Ejercicio 8

Con el escenario del Ejercicio 6 y si la carga útil fuera de 768 bytes, supóngase que ambos nodos se encuentran vinculados por un tramo satelital con un par de módems 64-QAM funcionando a 512 Kbps y que cada traspondedor trabaja como un retransmisor de capa 1 con retransmisión secuencial que inserta 100 tiempos de bits en la transmisión de cada haz. El NVP promedio del medio es 0,87 y la distancia promedio al satélite desde un nodo es de 38.750 Km. Calcular a) cuánto duró esta sesión, b) cuál fue la eficiencia del protocolo y c) cuál fue la ocupación del enlace, indicando los tiempos en milisegundos con dos decimales.

a) Tiempo de la transacción:

Tiempo total = Tiempo de Latencia + tiempo de modulación + tiempo de temporizadores

Tiempo de Latencia:

$$T_{aire} = \frac{38750 \times 2}{300000 \times 0,87} = 0,29693s = 296,93ms \text{ (Tiempo de un bit en el aire)}$$

$$T_{retardo} = \frac{100}{512000} = 0,000195s = 0,19ms \text{ (Tiempo de inserción de 100 bits)}$$

$$Latencia = T_{aire} + T_{retardo} = 296,93 + 0,19 = 297,12ms$$

$$T_{Lat} = 20 \times 297,12ms = 5942,40ms$$

Tiempo de modulación:

$$T_I = \frac{(768 + 6) \times 8}{512000} = 0,01209s = 12,09ms \text{ (Tiempo de Trama I)}$$

$$T_S = \frac{6 \times 8}{512000} = 0,0000930s = 0,09ms \text{ (Tiempo de Trama S = Tiempo de Trama Ucorta)}$$

$$T_m = 16 \times 12,09 + 14 \times 0,09 = 194,7ms \text{ (Tiempo total de modulación)}$$

Tiempo de temporizadores:

$$T_{T1} = 50 \times 0 = 0ms$$

Tiempo Total:

$$T_{Total} = T_{Lat} + T_m + T_{T1} = 5942,40 + 194,7 + 0 = 6137,10ms$$

$$T_{Total} = 6.137,10 \text{ ms}$$

b) Eficiencia

Bits útiles: 14 tramas; 768 Bytes

Bits totales: 16 tramas I de 768 Bytes + 6 Bytes overhead; 14 tramas S de 6 Bytes

$$Eficiencia = \frac{14 \times 768 \times 8}{16 \times (768 + 6) \times 8 + 14 \times 6 \times 8} = 0,8623$$

$$Eficiencia = 86,23\%$$

c) Ocupación del enlace

Bits reales: 99.744

Bits potenciales: 512000 bps durante 6.125,01 ms

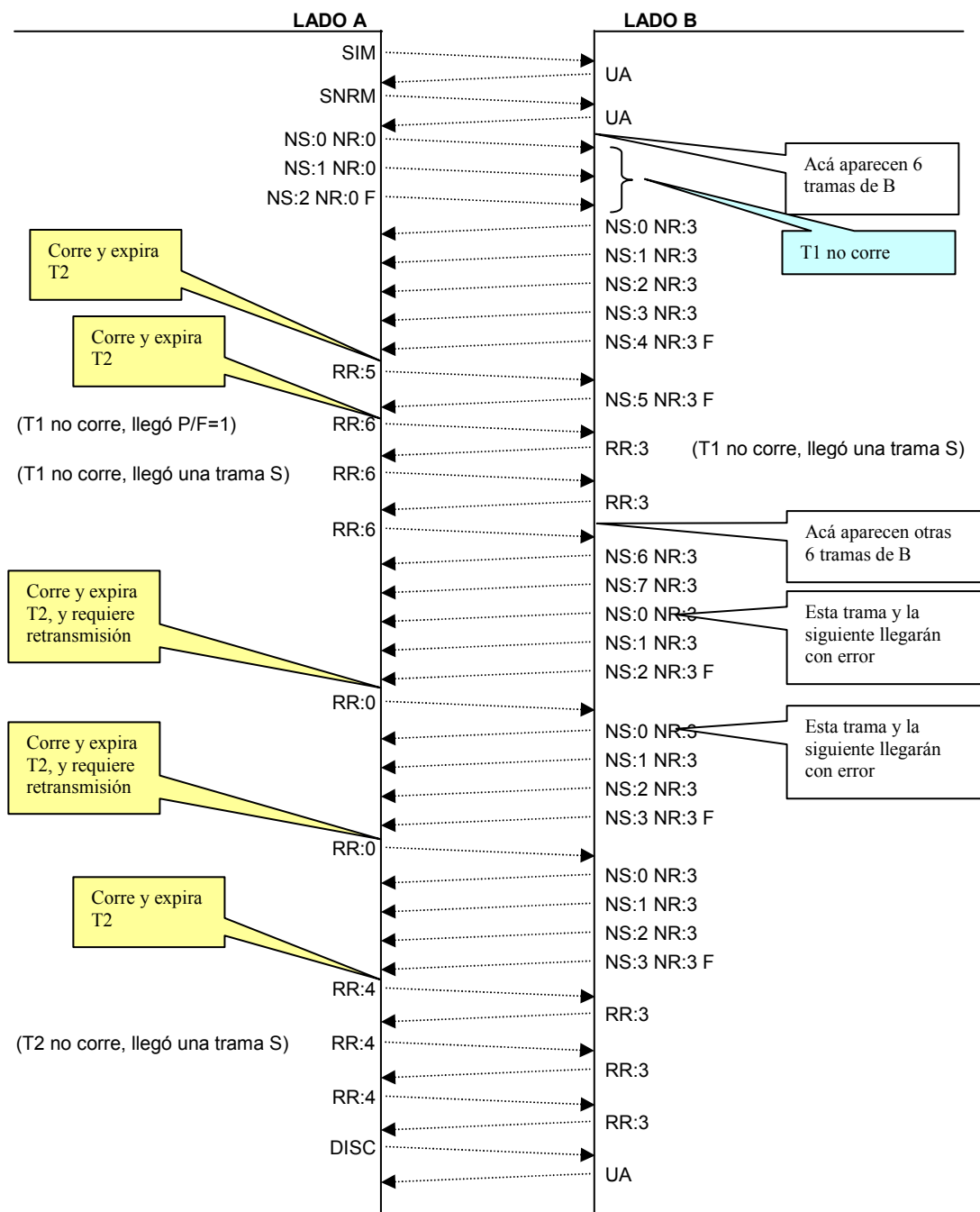
$$Ocupación = \frac{99744}{512000 \times 6,12501} = 0,0318$$

$$Ocupación = 3,18\%$$



Ejercicio 9

Dos nodos que usan $M=8$ se conectan usando $W=5$ y temporizadores $T1=20$ ms y $T2 = 50$ ms. Las tramas usan un campo INFO de 64 bytes. Al principio la conexión está cerrada y se abre sin necesidad de negociar. A tiene inicialmente 3 tramas en buffers listas para ser despachadas y en B aparecen 6 tramas para transmitir ni bien acepta el Set Mode. Comienza la transmisión y transcurre sin problemas, e inmediatamente después de que B contestó por segunda vez que está sin datos, aparecen en ese lado otras 6 tramas, de las cuales la tercera y cuarta llegan con error a A y llegan nuevamente con error a A cuando B las retransmite; la transmisión es exitosa la tercera vez. Termina el flujo después de que A sondea tres veces a B sin encontrar tramas I de respuesta, y cierra la transmisión. Veamos el flujograma:



Obsérvese que en B nunca corre T1 porque las tramas llegan en ráfaga y la última tiene $P/F=1$, y la única vez que podría haber corrido T2 había tramas en *buffer* de envío y entonces no corrió. En A pasa lo mismo con T1, y están marcados los instantes en los que corre T2.



Ejercicio 10

Para el escenario dado en el Ejercicio 9, suponer que se usó un módem de 256 Kbps y un medio que tiene una latencia de 23 ms, y calcular a) la eficiencia del protocolo y b) la ocupación del enlace. Antes de empezar, revisando el flujograma:

Cantidad de Latencias: 26

De A a B:

Tramas U cortas: 3 (SIM, SNRM, DISC)

Tramas U largas: 0

Tramas S: 9 (RR)

Tramas I: 3

En A:

T1: 0

T2: 5

De B a A:

Tramas U cortas: 3 (UA)

Tramas U largas: 0

Tramas S: 5 (RR)

Tramas I: 19 (12 útiles)

En B:

T1: 0

T2: 0

a) Cálculo de la Eficiencia

Bits útiles: 15 tramas I de 64 Bytes

Bits totales: 22 tramas I de (64 + 6) Bytes; 14 tramas S de 6 Bytes + 6 tramas U de 6 Bytes

$$Eficiencia = \frac{15 \times 64 \times 8}{22 \times (64 + 6) \times 8 + 14 \times 6 \times 8 + 6 \times 6 \times 8} = \frac{7680}{13280} = 0,5783$$

$$Eficiencia = 57,83\%$$

b) Cálculo de la ocupación

Ttotal = Tiempo de Latencia + Tiempo de modulación + Tiempo de temporizadores

$$T_{Lat} = 26 \times 23ms = 598ms \text{ (Tiempo de Latencia)}$$

$$T_I = \frac{(64 + 6) \times 8}{256000} = 0,00218s = 2,18ms \text{ (Tiempo de una Trama I)}$$

$$T_S = \frac{6 \times 8}{256000} = 0,00018s = 0,18ms \text{ (Tiempo de una Trama S = Tiempo de Trama U corta)}$$

$$T_m = 22 \times 2,18 + 20 \times 0,18 = 51,56ms \text{ (Tiempo total de modulación)}$$

$$T_{T1} = 20 \times 0 = 0ms ; T_{T2} = 50 \times 5 = 250ms \text{ (Tiempos de T1 y de T2)}$$

$$T_{Temp} = 0 + 250 = 250ms \text{ (Tiempo total de temporizadores)}$$

$$T_{TOTAL} = 598ms + 51,56ms + 250ms = 899,56ms \text{ (Tiempo total de toda la transacción)}$$

$$Bit_{REALES} = 13280 \text{ (Los que se enviaron)}$$

$$Bit_{POTENCIALES} = 256000 \frac{b}{s} \times 0,89956s = 230287b \text{ (Los que se enviaron)}$$

$$Ocupación = \frac{13280}{230287} = 0,0576$$

$$Ocupación = 5,76\%$$



Ejercicio 11

Se analiza el intercambio de tramas en un escenario en el que dos estaciones se enlazan conectándose con $M=8$ y $W=3$ y temporizan con $T1=20$ ms, $T2 = 20$ ms. No viene al caso la longitud de las tramas, pero el dato importante es que una trama I se modula en 5 ms y una trama S en mucho menos de 1 ms, por lo cual ese tiempo se desprecia. El enlace tiene una latencia de 5 ms. El arribo de tramas es irregular. Llegan a (A) tres tramas en el instante (25) y otras tres en (70), mientras que a (B) llegan primero dos tramas en (40) y luego una ráfaga de 5 tramas en (65), de las cuales la tercera, cuando se envía, se pierde y no llega a destino. Representar con un flujograma con escala de tiempos, y marcar especialmente cómo corren los temporizadores, indicando cuándo paran y cuándo expiran, en caso de que eso ocurra.

Observaciones sobre el flujograma:

Flechas en azul para T1, en marrón para T2 y en rojo para T3.

Flecha terminada con una cruz, indica que expiró, sino, indica que paró normalmente

