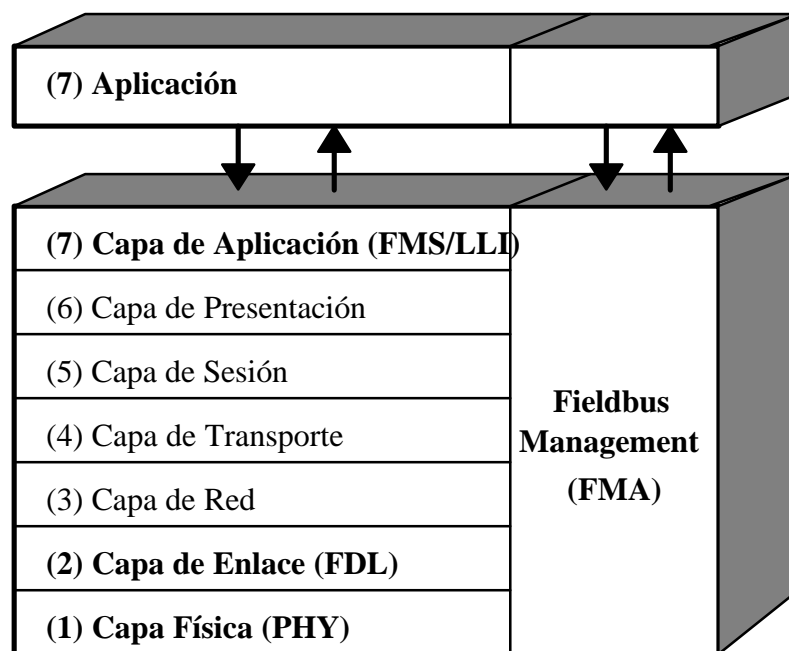


PROFIBUS (PROcess Field BUS) es una normativa Alemana relativa a los buses de proceso y de campo. El estándar PROFIBUS especificado en la norma "DIN 19.245, partes 1 y 2". En la parte 1, se describen las características funcionales, eléctricas y mecánicas para un bus de campo con una transmisión serie. El objetivo de este esfuerzo de normalización es poder conectar en una red local autómatas programables y equipos de campo como sensores, actuadores, transmisores, máquinas herramientas de control numérico, etc de diferentes fabricantes (sistema abierto).

La arquitectura PROFIBUS está modelada de acuerdo con el modelo de referencia ISO/OSI, restringido a 3 capas:

- Capa de aplicación (compuesta de las subcapas FMS y LLI, Layer 7)
- Capa de enlace (FDL Layer, Layer 2)
- Capa física (PHY Layer, Layer 1)

La especificación de las dos capas inferiores está descrita en "DIN 19.245, parte 1", mientras que la capa de aplicación lo está en "DIN 19.245, parte 2". Las capas 3 a 6 del modelo de referencia ISO/OSI no están implantadas en el modelo PROFIBUS, estado su funcionalidad transferida a la capa de aplicación.



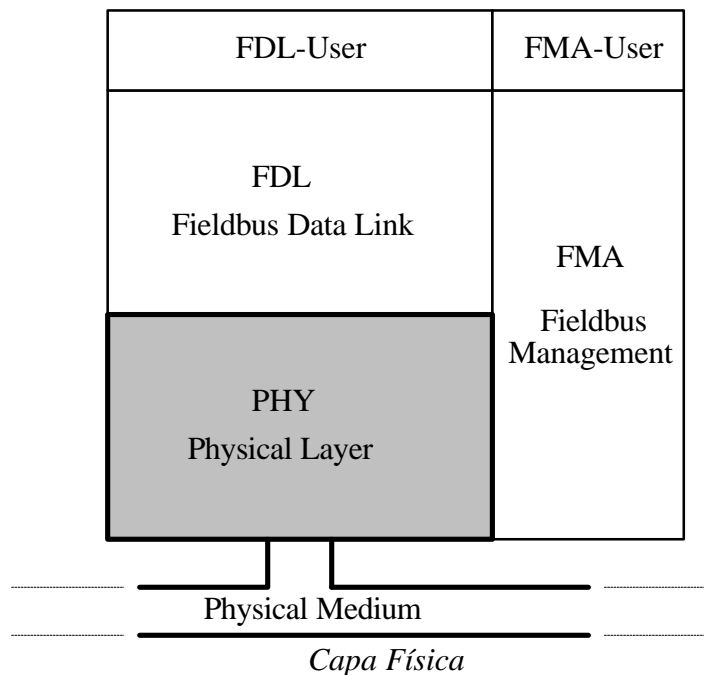
*PROFIBUS en el modelo de Referencia ISO/OSI*

La elección del medio de transmisión y del interface físico de bus vendrá determinado principalmente por el área de aplicación del sistema de bus de campo. Se han considerado los costes de compra e instalación, además del grado de fiabilidad en la transmisión. Por lo tanto, el estándar PROFIBUS prevee la posibilidad de especificar diferentes tipos de interfaces físicos. De hecho, la especificación de PROFIBUS está basada principalmente en estándares ya existentes. Pero, con el fin de satisfacer los requerimientos de varias áreas de aplicación (automatización de procesos de fabricación, edificios inteligentes, etc) se especificó el interface físico del estándar EIA RS-485.

Las capas 1 y 2 se encargan de asegurar la correcta transmisión de los datos, independientemente de su significado. Las características más importantes de estas dos capas, según la norma, son:

- Velocidades de transmisión: 9.6, 19.2, 93.75, 187.5 y 500 KBaudios.
- Cableado: par trenzado apantallado de acuerdo con el estándar Americano EIA RS-485 (ISO DP 8482). Ambos extremos del cable deben incorporar resistencias terminales de bus. Los cables se conectan a los dispositivos por medio de conectores sub-D de 9 pines.
- Número máximo de estaciones: 127 (32 sin utilizar repetidores).
- Distancias máximas alcanzables con cable de 0.22 mm. de diámetro:
  - hasta 93.75 KBaudios: 1200 metros
  - 187.5 KBaudios: 600 metros
  - 500 KBaudios: 200 metros
- Las estaciones pueden ser activas (maestros) o pasivas (esclavos).
- Medio de acceso: híbrido
  - maestro-esclavo
  - pase de testigo entre las estaciones maestras
- Acceso al medio determinístico, asegurándose así los tiempos de respuesta.
- Comunicaciones por medio de conexiones de tipo bidireccionales, multicast o broadcast.
- Servicios:
  - con o sin reconocimiento
  - cíclicos o acíclicos

**CAPA FISICA:**

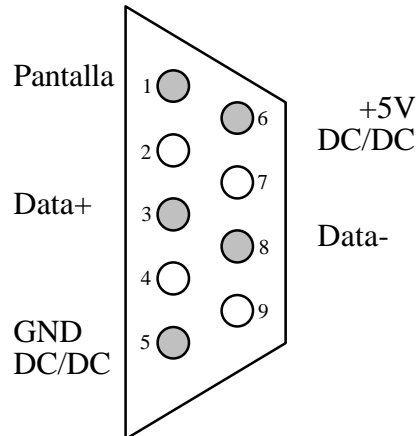


La transmisión de datos se realiza de manera *simétrica* de acuerdo con el estándar Americano **RS-485**. Las versiones más simples utilizan únicamente *par trenzado apantallado*, puesto que:

- Permite operar en ambientes con EMI a elevadas velocidades de transmisión
- Su instalación es sencilla y su coste bajo.

Si no se esperan fuertes interferencias electromagnéticas (EMI), puede emplearse par trenzado sin apantallamiento.

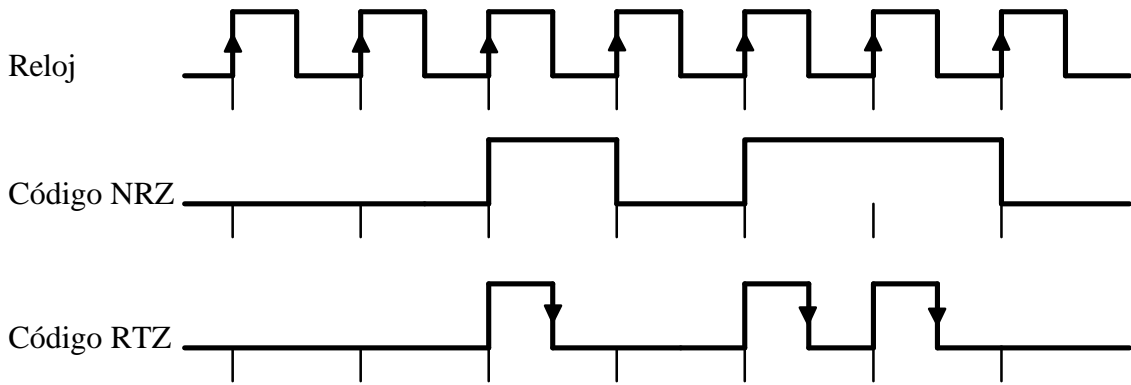
La conexión de los equipos al bus se realizará por medio de un conector sub-D de 9 pines con la siguiente asignación:



*Asignación de polos en el conector sub-D de 9 pines*

A diferencia del extendido estándar RS-232 (V.24), en el que se utilizan *niveles* absolutos de tensión para reconocer la información digital, en la transmisión RS-485 la información se reconoce por la *polaridad* de la tensión recibida. De este modo, es posible emplear velocidades de transmisión muy elevadas (hasta 10 Mbit/s) en ambientes con interferencias electromagnéticas.

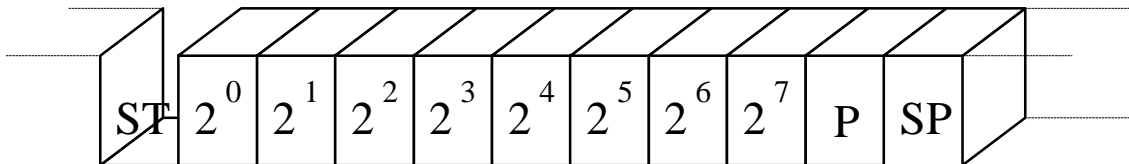
La codificación utilizada para los bits de información es la **NRZ** (Non-Return-to-Zero), muy utilizada en tecnologías digitales. Cada bit consiste de un pulso rectangular, cuyo ancho coincide con el periodo de reloj utilizado. Una tensión no nula identifica al '1' lógico, mientras que una tensión nula identifica al '0' lógico. En la codificación RTZ (Return-To-Zero), el '1' lógico no se identifica por el nivel de tensión, sino por el flanco negativo que aparece a mitad de periodo. Por ello, en este código, el ancho de pulso es la mitad del periodo del reloj.



*Códigos NRZ y RTZ*

En PROFIBUS se utiliza una técnica de transmisión asíncrona, en la que los relojes del emisor y receptor no están sincronizados, y la transmisión está orientada a caracteres. Cada carácter consiste de un total de 11 bits:

- 1 bit de comienzo ('Start bit', ST) que es siempre un '0' lógico.
- 1 bit de parada ('Stop bit', SP) que es siempre un '1' lógico.
- 1 bit de paridad ('Parity bit', P).
- 8 bits de datos ( $2^0 \dots 2^7$ ).



*Estructura de un carácter UART (Universal Asynchronous Receiver/Transmitter)*

Para asegurar la correcta recepción de los datos, los bits transmitidos deben ser muestreados en la mitad del periodo de duración del bit después de su comienzo. Esto requiere que el reloj del receptor duplique como mínimo el del transmisor. El estándar PROFIBUS permite desviaciones en las frecuencias de los relojes menores del 0.3%.

Entre dos mensajes transmitidos debe esperarse un tiempo definido en la norma como **Idle Time**. Este tiempo se caracteriza por la presencia de un '1' lógico en el bus. Su duración mínima ha de ser la equivalente al envío de tres caracteres, esto es, 33 periodos de bit.

La utilización de un **Idle Time** presenta dos ventajas importantes:

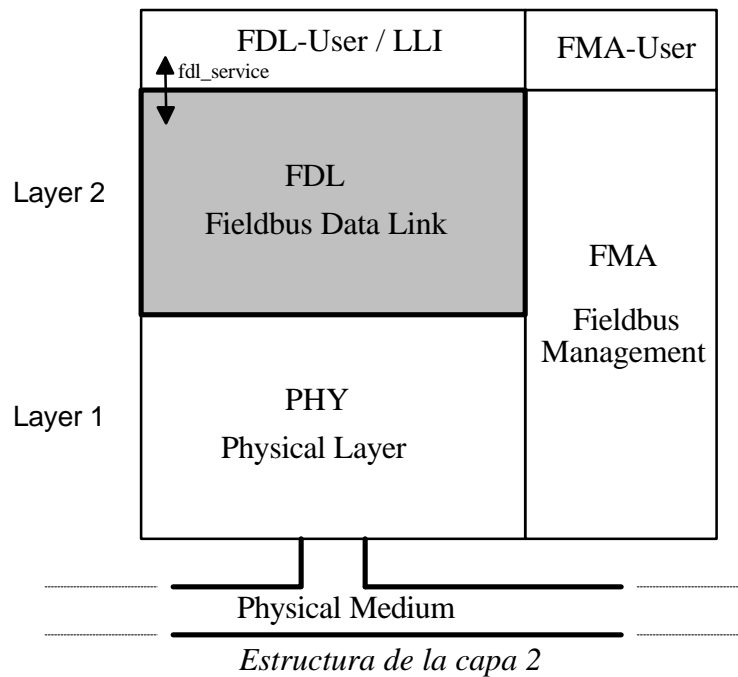
- Las estaciones pasivas activarán sus receptores sólo cuando haya transcurrido este tiempo de espera. Tras él, chequearán la dirección de destino del siguiente mensaje y desconectarán el receptor si no va dirigido a ellas. Este mecanismo de escucha 'selectiva' reduce la carga del protocolo, de modo que se libera mayor capacidad de procesamiento para el programa de aplicación de dichas estaciones.
- Las estaciones que acaban de transmitir pueden deshabilitar su transmisor durante este tiempo de espera. Tras la desconexión, la resistencia terminadora de bus se encarga de mantener éste en el '1' lógico.

Todas las estaciones activas monitorizarán el **Idle Time** por medio de temporizadores controlados por hardware. Un '0' lógico detectado en este periodo (causado, por ejemplo, por ruidos) será interpretado como un error que hará que el temporizador se redispere.

En el futuro se planea utilizar el estándar PROFIBUS en instalaciones con alto riesgo de explosión. No obstante, las especificaciones necesarias todavía no han sido realizadas. También será ventajoso utilizar otras técnicas de transmisión, como la fibra óptica, que permitirán alcanzar mayores distancias y utilizar otras topologías. En este área, existen ya realizaciones comerciales [como la red SINEC L2FO ('Fiber Optics') de SIEMENS].

## CAPA DE ENLACE:

En la terminología PROFIBUS, a la capa 2 se la conoce como la capa FDL (Fieldbus Data Link), capa de enlace.



Aunque la unidad mínima de información en PROFIBUS es el carácter, el protocolo utilizado está orientado a paquetes de información (mensajes). Estos mensajes consisten en una cadena ordenada de caracteres enviados de acuerdo con las especificaciones de la capa física.

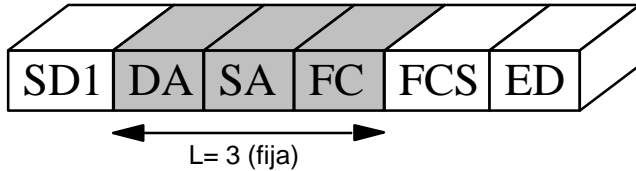
Para desarrollar los servicios de transferencia de datos, es preciso definir unos formatos de mensaje estandarizados en los que, además de buscar la máxima eficiencia en la codificación, se asegura una transmisión de datos fiable. Por esta razón, se incluye en los mensajes información redundante que inevitablemente aumenta la carga de protocolo.

En la figura siguiente se muestran tres ejemplos de formatos de mensajes en PROFIBUS. El primero de ellos se presenta un mensaje de llamada ('*call frame*') sin campo de datos, que es el mensaje más corto posible. Un mensaje siempre empieza por un delimitador de inicio (**SD**, 'Start Delimiter'), que contiene el código del formato de mensaje. A continuación aparece la dirección de destino (**DA**, 'Destination Address') y la dirección de origen (**SA**, 'Source Address') que identifican al receptor y transmisor del mensaje, respectivamente. Tras ellos sigue el control de mensaje (**FC**, 'Frame Control'), por el cual se le indica al receptor el tipo de mensaje; en él también se incluye la prioridad del mensaje y el control de la información que evita, por ejemplo, la pérdida del mensaje. Al final del mensaje aparece la secuencia de verificación de mensaje (**FCS**, 'Frame Check Structure'), que sirve para verificar la consistencia de los datos recibidos, seguido del delimitador de final de mensaje (**ED**, 'End Delimiter'). La estación receptora reconoce la

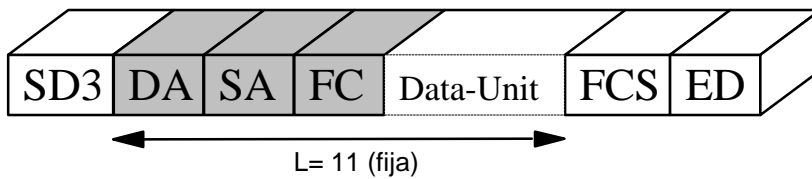
recepción del mensaje transmitiendo un pequeño mensaje de reconocimiento (1 único caracter).

Los ejemplos siguientes son autoexplicativos. El último de ellos puede utilizar la máxima longitud de mensaje en la capa 2 (255 caracteres), de los cuales 246 son información.

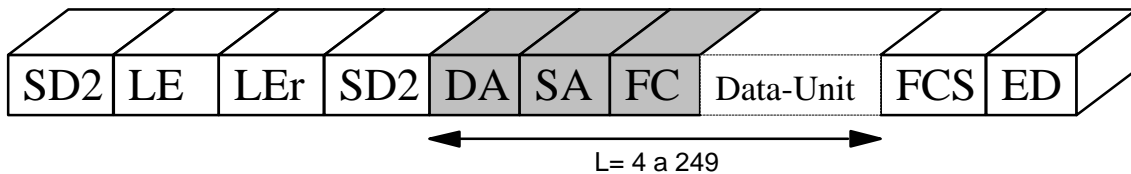
Formato con longitud del campo de información fija y sin datos



Formato con longitud del campo de información y datos fija



Formato con longitud del campo de información variable



- SD1..SD4: Start Delimiter (byte de inicio)
- LE, LEr: Length byte (byte de longitud)
- DA: Destination Address byte (byte de dirección de destino)
- SA: Source Address byte (byte de dirección de origen)
- FC: Frame Control byte (byte de control del mensaje)
- Data-Unit: Frame Check Sequence (byte de chequeo)
- FCS: End Delimiter (byte final)
- ED: Longitud del campo de información
- L:

*Ejemplos de mensajes PROFIBUS con HD=4*

Aunque PROFIBUS dispone de protección contra los errores de transmisión, éstos no pueden ser totalmente eliminados, ya que pueden ocurrir por la existencia de transmisores defectuosos, tierras con alta impedancia, reflexión de la señal, etc. Los componentes de la UART son capaces de detectar algunos de estos errores:

- error de "frame": el bit de stop de un caracter no fue reconocido.
- error de "overrun": se ha sobrescrito un caracter recibido con el siguiente antes de que aquel pudiera ser almacenado.

Para asegurar una transmisión libre de errores, PROFIBUS dispone como mecanismo de seguridad con distancia Hamming 4 (Hd = 4). La distancia Hamming indica en cuantas unidades binarias difieren dos caracteres reconocidos como válidos por la UART (así, la introducción de un bit de paridad en los caracteres UART incrementa en 1 la distancia

Hamming). En el caso de  $H_d = 4$ , es posible detectar y corregir un error de 1 bit en un carácter, mientras que un error de 2 bits es posible detectarlo pero no corregirlo.

Se logra  $H_d=4$  enviando un FCS ("Frame Check Sequence") con cada mensaje, que representa la columna paridad del campo de información, de longitud L. Se genera haciendo la suma sin acarreo de los caracteres transmitidos. Puesto que los bits SD y ED no se incluyen en esta suma, su distancia Hamming frente a los demás es 5.

En PROFIBUS no se utiliza la corrección de errores. Cuando se detecta un error en un mensaje, se descarta el mismo y se repite la transmisión.

Según se comentó anteriormente, en PROFIBUS existen dos tipos de estaciones: activos y pasivos.

Las estaciones activas (maestros) pueden:

- enviar por iniciativa propia datos a otras estaciones
- solicitar datos de otras estaciones

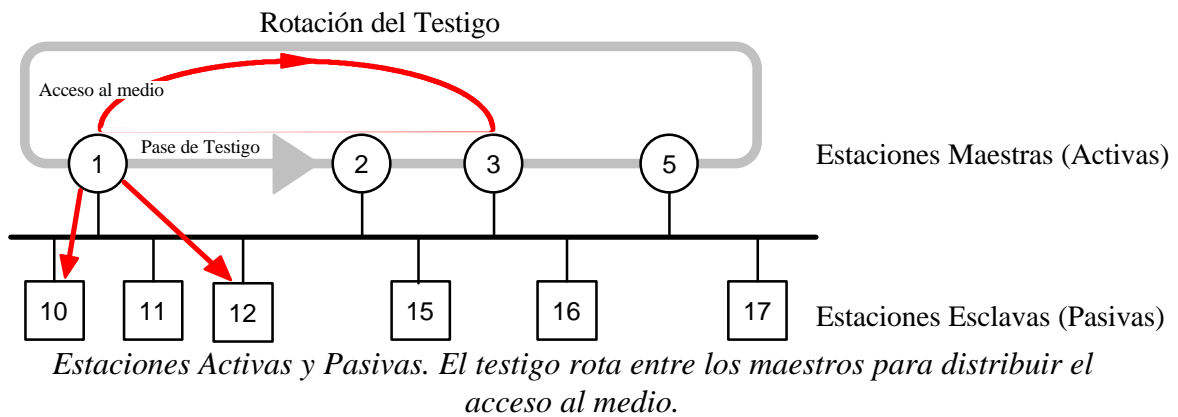
Las estaciones pasivas (esclavos):

- NO pueden intercambiar datos excepto en el caso de que una estación activa les autorice a ello.

La característica activo/pasivo de una estación depende del equipo considerado. En general, los elementos de campo simples (estaciones de mando de motores, sensores, etc) serán pasivos, mientras que los equipos "inteligentes" (autómatas programables, máquinas herramientas con control numérico, etc) serán activos. También es posible que una misma estación se pueda configurar como activa o pasiva en el bus.

Para evitar que todas las estaciones activas accedan al mismo tiempo al bus, toda estación que desee enviar una información debe esperar hasta que se reciba una autorización de acceso al bus. Dicha estación recibe la autorización en forma de un telegrama especial, el "testigo" (token):

- El testigo pasa automáticamente de una estación activa a la siguiente, siguiendo el orden de direcciones de las estaciones.
- La transmisión del testigo se efectúa según un anillo lógico, es decir, la estación activa presente cuya dirección sea la más elevada reenvía el testigo a la estación activa presente cuya dirección sea la más baja.
- Cada estación activa "conoce" las direcciones de las restantes estaciones activas y controla de manera cíclica el campo de direcciones entre ella y la estación activa presente que le sigue. Durante este control se detecta:
  - El añadido de una estación activa o pasiva, o
  - La supresión de una estación pasiva.
- Si se detecta que se ha añadido una nueva estación activa, ésta última recibe inmediatamente el testigo.



Con respecto al diagrama anterior:

- Las estaciones 1, 2, 3 y 5 son activas. El testigo se transmite de la siguiente forma:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 1 \rightarrow 2 \dots$$

- Un ciclo de rotación de testigo se compone aquí de 4 transmisiones sucesivas del testigo:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 1$$

- Las estaciones 10, 11, 12, 15, 16 y 17 son pasivas
- Las direcciones de estación 0, 4, 13, 14 y 18 a 126 no están ocupadas.

Un ciclo de rotación del testigo dura un cierto tiempo. El tiempo máximo de ciclo de rotación de testigo se especifica en la norma por medio del parámetro **Token-Rotation-Time (TRT o  $T_{TR}$ )**.

El valor seleccionado para el **TRT** debe ser respetado incluso en presencia de un gran volumen de datos a transferir. Esto se consigue con los principios que se describen a continuación:

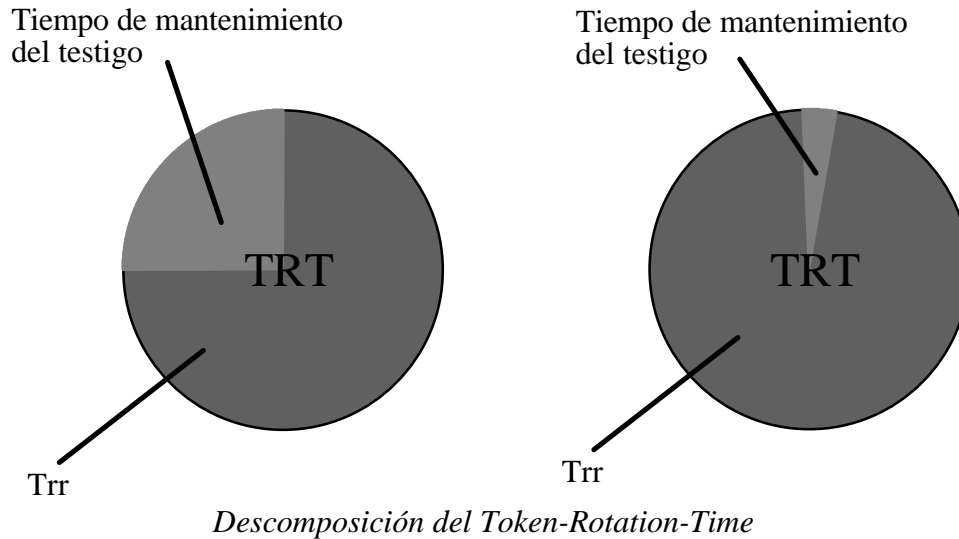
Las conexiones entre las diferentes estaciones de la red pueden ser de **alta** prioridad o de **baja** prioridad. La prioridad de una conexión influye sobre el tratamiento de los mensajes que deben ser emitidos por la conexión considerada.

Cada estación activa mide el tiempo durante el cual ella no está en posesión del testigo; [dicho tiempo representa para ella el **Real-Token-Rotation-Time ( $T_{RR}$ )**] y lo compara con el **TRT** parametrizado.

El tratamiento de los mensajes a emitir depende del resultado de esta comparación y la prioridad de las conexiones, de modo que:

- Si  **$T_{RR}$**  es inferior al **TRT**, todas las órdenes de emisión y recepción existentes son ejecutadas hasta que finalice el **TRT** o bien se agoten todas las órdenes en espera: en primer lugar las órdenes cuyas conexiones son de alta prioridad y después las órdenes cuyas conexiones tienen baja prioridad.
- Si  **$T_{RR}$**  es igual o superior al **TRT**, sólo se elabora **una** orden cuya conexión tenga prioridad alta. Las órdenes con prioridad baja sólo se elaborarán en los siguientes ciclos, cuando  **$T_{RR}$**  sea inferior al **TRT**.

Estos dos casos se ilustran de forma gráfica en la siguiente figura, en la cual el **TRT** se representa por el círculo completo.



Los servicios de baja prioridad se realizan de acuerdo con la siguiente secuencia:

- Procesamiento de la Lista de Consulta (Poll List)
- Procesamiento de los mensajes de baja prioridad (servicios acíclicos)
- Almacenamiento de la lista de estaciones activas (Live List)
- Actualización del GAP (a lo sumo una dirección de la GAP list)

Esta secuencia no es fija, sino que puede ser modificada bajo ciertas condiciones de operación. Por ejemplo, el procesado de la Lista de Consulta se hace de manera segmentada, de modo que si no queda disponible tiempo de testigo, no se pueden procesar servicios acíclicos. Por esta razón el control FDL debe asegurar el procesamiento de todas las peticiones de baja prioridad, incluyendo la actualización del GAP, antes de comenzar un nuevo ciclo de consulta.

A continuación se enumeran los principales parámetros de bus que han de inicializarse obligatoriamente:

Nombre	Rango de Valores	Significado
<b>TS</b>	0 a 126	Address of this Station
<b>Baud_rate</b>	9.6 , 19.2, ... 500 kbit/s	Baud Rate in this PROFIBUS
<b>Medium_red</b>	single o redundant	Single/Redundant Media available
<b>HW-Release</b>	ASCII String	Hardware Release Number
<b>SW-Release</b>	ASCII String	Software Release Number
<b>T<sub>SL</sub></b>	1 ... 65535 bit times	Slot Time
<b>min T<sub>SDR</sub></b>	1 ... 65535 bit times	smallest Station Delay Time
<b>*) max T<sub>SDR</sub></b>	1 ... 65535 bit times	largest Station Delay Time
<b>*) T<sub>OUT</sub></b>	0 ... 255 bit times	Transmitter fall Time / Repeater switch Time

*) <b>T<sub>SET</sub></b>	1 ... 255 bit times	Setup Time
*) <b>T<sub>TR</sub></b>	1 ... 16.777.215 bit times	Target Rotation Time
*) <b>G</b>	1 a 100	GAP Update Factor
*) <b>in_ring_desired</b>	true o false	Request entry into or exit out of the logical Token Ring
*) <b>HSA</b>	2 ... 126	Highest Station Address in PROFIBUS
*) <b>max_retry_limit</b>	1 ... 8 (preferible 1)	Maximum number of retries
*) sólo posible en Estaciones Maestras		

*Principales parámetros de bus*

**TS** : Dirección de la estación en el bus (no pueden existir dos estaciones con la misma dirección en el bus). Como es lógico, este parámetro es específico de cada estación.

**Baud\_rate** : Velocidad de transmisión en Kbits/segundo. Una vez seleccionada, se define el *bit time* ( $t_{bit}$ ) como el tiempo necesario para la transmisión de un bit de información, es decir:  $t_{bit} = 1 / \text{Baud\_rate}$  (en bits/segundo). Este parámetro debe ser igual en todas las estaciones.

**Medium\_red** : Se indica aquí si existe o no redundancia.

**T<sub>SL</sub>** : Tiempo durante el cual el transmisor de un mensaje espera el reconocimiento por parte del destinatario. Cuando se sobrepasa, se repite o no la transmisión de acuerdo con el valor de **max\_retry\_limit**. Este parámetro debe ser igual en todas las estaciones.

**min T<sub>SDR</sub>** : Mínimo tiempo que debe esperar el destinatario de un mensaje para enviar el reconocimiento al transmisor del mismo. Este parámetro debe ser igual en todas las estaciones.

**max T<sub>SDR</sub>** : Mínimo tiempo que debe esperar el transmisor antes de enviar un nuevo mensaje. Este parámetro debe ser igual en todas las estaciones maestras.

**T<sub>QUI</sub>** : Tiempo que una estación transmisora debe esperar tras enviar un mensaje para activar su receptor. Este parámetro debe ser igual en todas las estaciones maestras.

**T<sub>SET</sub>** : Mínimo tiempo de espera entre la recepción de un reconocimiento y el envío de un nuevo mensaje por parte del transmisor. (Este parámetro debe ser igual en todas las estaciones maestras).

**T<sub>TR</sub>** : Tiempo predeterminado para que el testigo pase por todas las estaciones del anillo lógico. Este parámetro debe ser igual en todas las estaciones maestras.

**G** : Después de transcurrido el tiempo " $G * T_{TR}$ ", el área de direcciones libres entre dos estaciones activas debe ser chequeado por la estación con dirección más baja para detectar si existen una nueva estación que desea entrar en el anillo lógico. Este parámetro debe ser igual en todas las estaciones maestras.

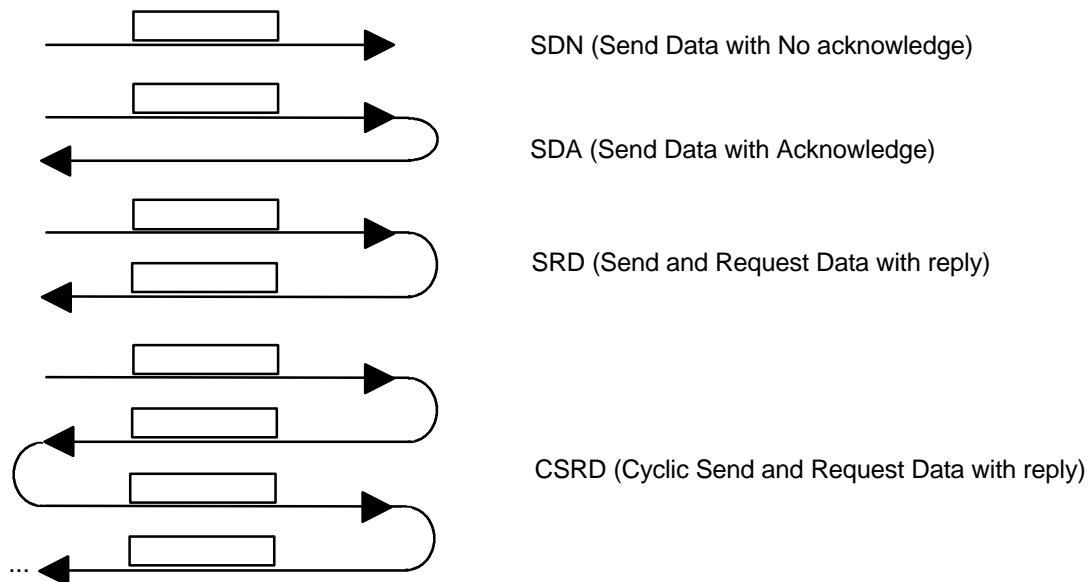
**in\_ring\_desired** : Indica si la estación va a entrar en la red PROFIBUS como esclava o como maestra. Este parámetro es específico de cada estación.

**HSA** : Especifica la dirección más alta permitida para una estación maestra en el bus. Las direcciones de estaciones esclavas pueden ser mayores que **HSA**. Este parámetro debe ser igual en todas las estaciones maestras.

**max\_retry\_limit** : Especifica el número máximo de veces que el transmisor repetirá el envío de un mensaje cuando no reciba el reconocimiento del mismo por parte del destinatario. Este parámetro debe ser igual en todas las estaciones maestras.

Además de controlar el acceso al medio y el tiempo de rotación de testigo, en la capa FDL existen una serie de servicios de transferencia de datos disponibles para el usuario a través de dicha capa. PROFIBUS ofrece tres servicios de transferencia de datos para envíos y recepciones *acíclicas*:

- **SDA**      Send Data with Acknowledge
- **SDN**      Send Data with No acknowledge
- **SRD**      Send and Request Data



*Servicios disponibles en la capa FDL*

El servicio **SDA** es un servicio básico por el cual la estación activa (la estación que tiene actualmente el derecho de acceso al bus) puede enviar un mensaje a una estación pasiva y recibir inmediatamente la confirmación. Esta última sólo puede reconocer los datos recibidos o reaccionar ante los datos recibidos transmitiendo otros datos ella misma.

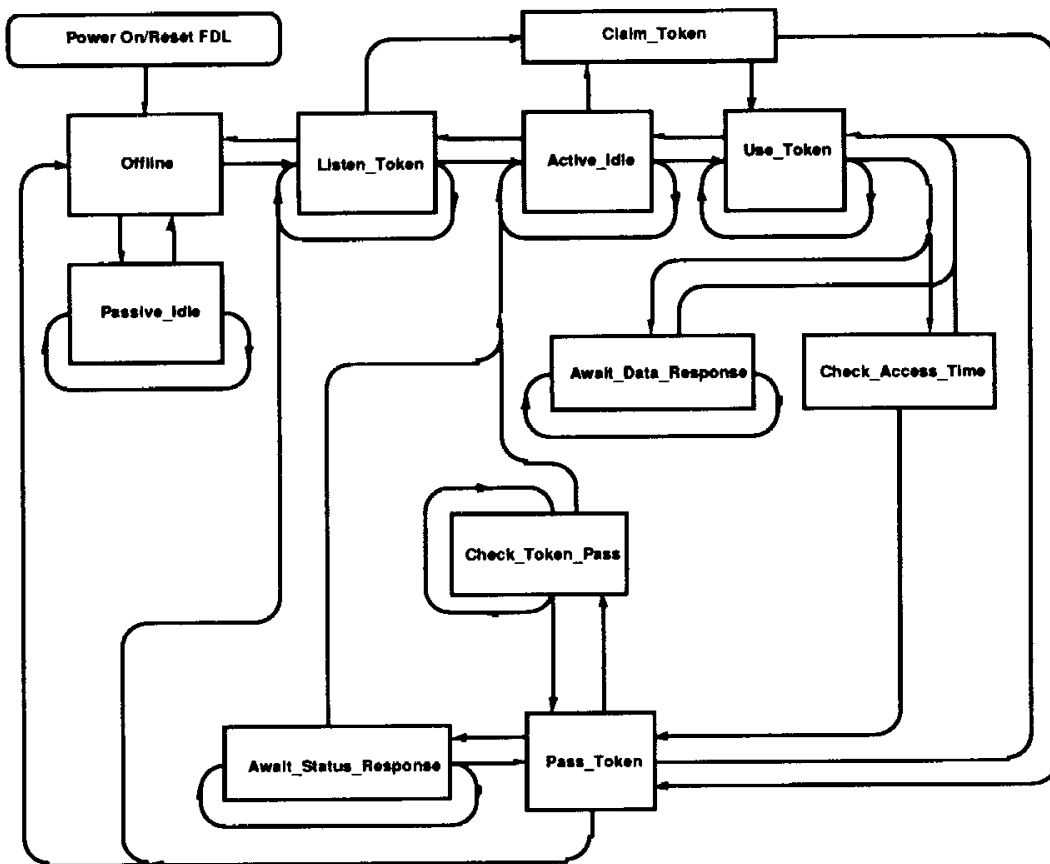
El servicio **SDN** se utiliza principalmente en mensajes de tipo '*broadcast*' o '*multicast*' desde la estación activa a varias (o todas las demás) estaciones, razón por la cual no se recibe reconocimiento. Todos los demás servicios se basan en una conexión recíproca entre un *iniciador* (la estación que posee el testigo) y un *respondedor* (cualquier estación que no tenga el testigo), y requieren bien un reconocimiento (Acknowledge) bien una respuesta (Response). Este comportamiento, importante para el funcionamiento en tiempo real del sistema de bus, se denomina de **respuesta inmediata**.

El servicio **SRD** permite transmitir datos a una estación pasiva y al mismo tiempo solicitar datos de ésta. Los datos solicitados se envían conjuntamente con la respuesta. Como caso especial de este servicio, una estación puede sólo solicitar datos de otra estación enviándole un 'mensaje vacío'.

Además de estos tres servicios acíclicos, las aplicaciones industriales requieren a menudo la realización de transmisiones *cíclicas*. El método de consulta centralizada es adecuado para recoger información de dispositivos de campo sencillos, como sensores y multiplexores de E/S que nunca serán dispositivos controlados por eventos ni tendrán (por razones de coste) la funcionalidad de una estación activa. PROFIBUS ofrece la posibilidad de almacenar una *lista de consulta (Poll List)* en un nivel cercano al hardware y, basándose en el servicio acíclico SRD, implantar una consulta cíclica de las estaciones incluídas en la misma. Además, al estar realizados los procedimientos de consulta en la capa de enlace se optimizan los tiempos de respuesta. Este servicio es el definido como:

- **CSRD** Cyclic Send and Request Data with reply.

#### MAQUINA DE ESTADOS DEL CONTROLADOR DE LA CAPA DE ENLACE



*Diagrama de estados del controlador FDL*

Tras alimentar la estación (Power ON), tanto las estaciones activas como las pasivas entran en el estado *Offline*, en el que deben ejecutar un autochequeo interno. A

continuación cargan los parámetros de comunicación necesarios y se conectan al medio de transmisión.

Después, las estaciones pasivas pasan al estado *Passive\_Idle*, en el que escuchan y reconocen los mensajes dirigidos a ellas (excepto en los mensajes broadcast). No hay más estados a los que pueda pasar una estación pasiva.

Tras el estado *Offline*, las estaciones activas pasan al estado *Listen-Token* cuando están preparadas para entrar en el anillo lógico. El Testigo ("token") es una secuencia de bits particular que puede interpretarse como la cesión del derecho a transmitir datos. En este estado, las estaciones recogen todos los mensajes del bus y recopilan la Lista de Estaciones Activas (LAS) a partir de las direcciones de los mensajes de testigo recibidos.

Tras la recopilación de esta lista, la estación debe esperar a ser direccionada por la estación previa con un mensaje 'Request FDL Status' y así ser "invitada" a entrar en el anillo lógico. Entonces, esta estación reconoce el mensaje enviando 'ready for the ring' y pasa al estado *Active\_Idle*. De este modo la estación entra oficialmente en el anillo.

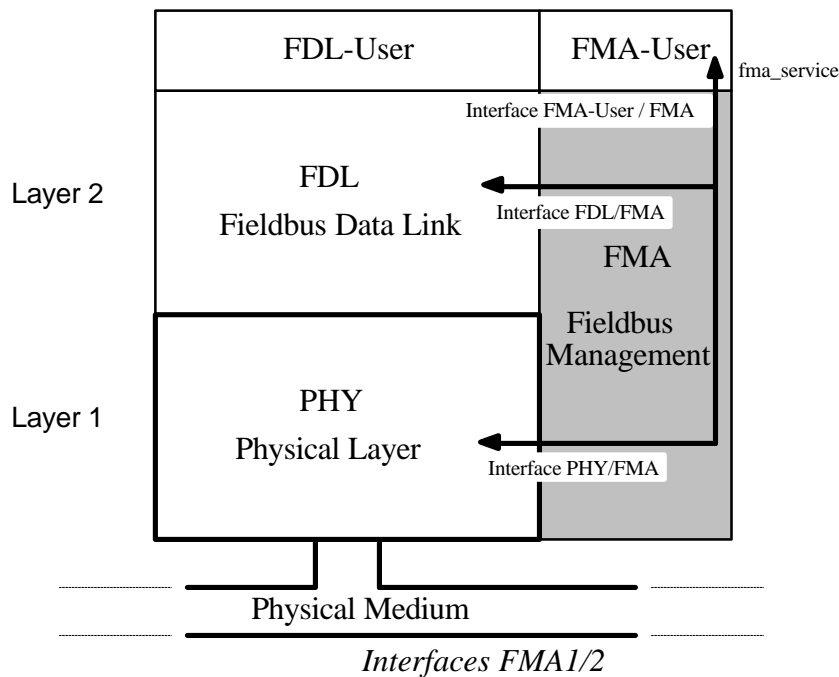
Cuando la estación reciba el mensaje de testigo, pasará al estado *Use-Token*. Este es el estado de una estación activa en el anillo lógico que posee el derecho de acceso al bus (todas las demás estaciones activas estarán en el estado *Active\_Idle*). La estación comprueba el tiempo de testigo que le queda en el estado *Check\_Access\_Time* y ejecuta ciclos de mensajes a través del bus. Si la estación utiliza un servicio con respuesta, pasa al estado *Await\_Data\_Response* en el que espera la respuesta durante un cierto periodo de tiempo.

La estación puede ejecutar ciclos de mensaje hasta que acabe el tiempo de mantenimiento de testigo. Cuando esto ocurre, la estación pasa al estado *Pass-Token* en el que pasa el testigo a la siguiente estación activa del bus. El paso del testigo se monitoriza en el estado *Check-Token\_Pass*. Si no existe un sucesor conocido, se entra en el estado *Await\_Status\_Response*.

En este momento el controlador FDL espera un cierto tiempo por un mensaje de reconocimiento. Si la estación activa no recibe mensaje o es incorrecto, se entra de nuevo en el estado *Pass-Token*. Si ocurre un error, la estación pasa al estado *Listen-Token* e informa de ello a la capa FMA 1/2. En el caso normal (después de pasar el testigo sin que se produzca error), pasa al estado *Active\_Idle* hasta que vuelve a recibir el testigo.

Si una estación activa no recibe el mensaje de testigo por un largo periodo de tiempo, pasa al estado *Claim-Token*, en el que intenta inicializar (si hasta ese momento no había sido una estación activa en el bus) o reinicializar (si ya lo era) el anillo lógico. En el último caso, la lista de estaciones activas (LAS) sigue siendo válida.

FMA1/2



La capa FMA1/2 (Fieldbus Management Administration de las capas 1 y 2) dispone de funciones para el manejo de las capas 1 y 2. Se encarga por tanto de enlazar un usuario de FMA1/2 (por ejemplo, la capa FMA7) con las capas 1 y 2.

Utilizando los servicios FMA, es posible realizar tareas de gestión a los interfaces. Utilizando la primitiva de servicio FMA1/2 *service.request*, la capa FMA7 solicita a la capa FMA1/2 la ejecución del servicio. Sin embargo, sólo es posible ejecutar un único servicio FMA1/2 en cada instante.

Antes de la ejecución de la petición (request), se chequean los parámetros que van a ser transferidos con el servicio para asegurar la compatibilidad. En la capa FMA1/2, los servicios son procesados, adaptados y transferidos a la capa correspondiente. Tras procesar el servicio, esta última capa envía una primitiva *service.confirm.primitive* FMA1/2 que, a su vez, genera la respuesta FMA1/2 *service.confirm* a la capa FMA7. Esta respuesta generalmente incluye información sobre si el servicio se ha ejecutado correctamente o no. Sin embargo, algunos servicios permiten una confirmación inmediata en la capa FMA1/2 antes de ser procesados por la capa correspondiente. Las respuestas FMA1/2 *service.indication* envían los errores y eventos producidos en las capas 1 y 2 o en el bus al usuario de FMA1/2.

Los servicios pueden ser clasificados en dos categorías: servicios locales y servicios remotos. En ambos grupos existen servicios obligatorios (que han de existir en toda estación) y opcionales (que se pueden implantar o no). Tras su ejecución, la mayoría de los servicios generan una respuesta que informa al usuario a través de FMA1/2 si el servicio se ejecutó correctamente, o si por el contrario se ha producido un error.

Los servicios locales definidos son:

- **Reset:** Con este servicio, la capa FMA7 puede reinicializar las capas 1 y 2. Una vez que FMA1/2 recibe este servicio, genera las órdenes de reset para ambas capas, las pasa por el interface correspondiente y espera la confirmación de ambas capas.

- **Set Value:** Con este servicio opcional es posible asignar valores a ciertas variables de las capas 1 y 2. En la primitiva de servicio se incluye el nombre de la variable y el valor deseado. La capa FMA1/2 genera un *FDL* y/o *PHY\_SET\_VALUE.request* y lo transfiere a la(s) capa(s).

- **Read Value:** Con este servicio opcional es posible leer valores de variables en las capas 1 y 2. En la primitiva de servicio se incluye el nombre de la variable. Entonces, la capa FMA1/2 genera un *FDL* y/o *PHY\_READ\_VALUE.request* y lo transmite a la capa correspondiente.

- **Event:** Este servicio informa al usuario de ciertos eventos o errores producidos en las capas 1 y/o 2. Cuando se produce una indicación de error (*PHY\_EVENT.indication* o *FDL\_FAULT.indication*) en una de las capas, se genera una indicación *FMA1/2\_EVENT.indication* para la capa FMA1/2 que a su vez la transmite a FMA7.

- **(R)SAP Activate FMA1/2:** Con este servicio opcional, el usuario puede configurar y activar un SAP. Todas las comunicaciones se llevan a cabo a través de los SAP's (Service Access Points). Contienen las condiciones de los mensajes permitidos y de los formatos de mensaje (direcciones de destino permitidas, longitud de los datos, etc). Siempre que un mensaje llega a un SAP, se chequea de acuerdo con los requerimientos prescritos para el mismo. Si es correcto, se transmite el mensaje; en caso contrario, se genera un mensaje de error. A primera vista, la comunicación a través de los SAP's parece compleja, pero esto permite chequear los mensajes y las líneas de transmisión de modo que es posible evitar conexiones incorrectas. Para enviar respuestas sobre los servicios SRD y CSRD, es preciso establecer un RSAP por medio de este servicio.

- **(R)SAP Deactivate FMA1/2:** Este servicio es complementario del anterior. Permite que el usuario de la capa FMA1/2 desactive un (R)SAP y también todas las conexiones relacionadas con él.

Los servicios remotos definidos son:

- **Ident FMA1/2:** Con este servicio opcional, el usuario de la capa FMA1/2 puede obtener información sobre las versiones de hardware y software. Las estaciones activas pueden preguntar a todas las demás por su identidad, mientras que en las estaciones pasivas se limita a la identidad de la propia estación.

- **LSAP Status FMA1/2:** Este servicio opcional sólo puede ser utilizado en estaciones activas. Permite obtener información acerca de la configuración de un SAP de otra estación de la red para servicios FDL (por ejemplo, SDA, SRD, etc).

- **Live List FMA1/2:** Este servicio opcional permite que el usuario obtenga una lista de todas las estaciones conectadas al bus. Esta lista se compila en la capa FDL generando un servicio *Request FDL Status* con respuesta para cada posible dirección de estación. Consultando todas las posibles direcciones, el usuario obtiene una lista completa de las

estaciones que incluye tanto a las activas como a las pasivas (a diferencia de las estaciones existentes en la LAS (List of Active Stations)).