

**Diseño de alto nivel**

# **Controlador DNP3 para la CIAA**

**Versión 0**

**Preparado por Amaury Jiménez**

**gridTICs**

**Septiembre de 2016**

# Índice

1.	Introducción.....	5
1.1	Objetivo.....	5
1.2	Definiciones .....	5
1.3	Acrónimos y abreviaciones.....	5
1.4	Destinatarios .....	5
2.	Nivel de implementación .....	6
2.1	Tabla de implementación .....	7
3.	Capa de aplicación.....	9
3.1	Puntos y tipos de puntos .....	9
3.2	Índices, grupos y variaciones .....	10
3.2.1	Índices .....	10
3.2.2	Grupos .....	10
3.2.3	Variaciones .....	10
3.3	Objetos DNP3 .....	11
3.3.1	Estático .....	11
3.3.2	Eventos .....	11
3.3.3	Clases .....	11
3.4	Inicio de la remota.....	11
3.5	Estructura de los mensajes .....	12
3.5.1	Estructura general de los fragmentos.....	12
3.5.2	Encabezado de solicitud de Aplicación .....	12
3.5.3	Encabezado de respuesta de Aplicación.....	12
3.5.4	Fragmento de solicitud de Aplicación .....	13
3.5.5	Fragmento de respuesta de Aplicación .....	13
3.6	Tabla de estados de fragmento de la remota .....	13
3.7	Diagrama de estados de fragmentos de la remota .....	16
3.8	Reglas de Fragmentos .....	16
4.	Capa de Pseudo Transporte .....	18
4.1	Encabezado .....	18
4.1.1	Campo FIN:.....	19
4.1.2	Campo FIR.....	19
4.1.3	Campo número de secuencia.....	19
5.	Capa de Enlace.....	19
5.1	Descripción de la capa de enlace.....	20
5.2	Servicios .....	20
5.3	Modelo de transacción .....	20
5.4	Formato de la trama.....	20
5.4.1	Encabezado .....	21
5.4.2	Bloque de datos .....	21

5.4.3	Campos CRC.....	23
-------	-----------------	----

# Historial de revisiones

Nombre	Fecha	Descripción de los cambios	Versión
Amaury Jiménez	23/09/16	Emisión inicial	0

# 1. Introducción

## 1.1 Objetivo

El objetivo de este documento es definir las principales entidades funcionales y arquitectura a implementar para el controlador DNP3 de la CIAA.

## 1.2 Definiciones

**Entrada:** Valores que son medidos, leídos o generados por el dispositivo y son reportados por una remota al maestro. Ejemplos:

Nivel de líquido en un tanque

Estado de un interruptor

La suma calculada de la potencia en las tres fases de una línea

**Maestro:** Cualquier entidad que emite requerimientos para obtener datos o ejecutar comandos usando DNP3.

**Punto:** Entidad física o lógica unívocamente identificable. El término “punto” aplica a entradas analógicas, digitales y contadores como así también a salidas analógicas y digitales.

**Remota:** Cualquier entidad localizada en el campo que recibe requerimientos para enviar datos o ejecutar comandos usando DNP3.

## 1.3 Acrónimos y abreviaciones

ASCII American Standard Code for Information Interchange

CIAA Computadora Industrial Abierta Argentina

DNP Distributed Network Protocol

EPA Enhanced Performance Architecture

IEC International Electrotechnical Commission

IED Intelligent Electronic Device

IEEE Institute of Electrical and Electronics Engineers

RTU Remote Terminal Unit

SCADA System Control and Data Acquisition

## 1.4 Destinatarios

Este documento está dirigido a los desarrolladores del controlador, usuarios finales y a las entidades responsables del financiamiento y control de la ejecución del proyecto.

## 2. Nivel de implementación

Se implementará el Nivel 1 (DNP3-L1). Este nivel de implementación representa la expresión más sencilla de DNP3 para comunicación entre un maestro y un IED típico.

El Nivel 1 está basado en el sondeo de Clases. Una remota Nivel 1 debe aceptar requerimientos para:

- LECTURAs de Objetos de Datos por Clase
- LECTURAs de objetos de Salidas Digitales, si estas existen en la remota.
- LECTURAs de variaciones específicas de objetos del grupo 0 (Atributos del Dispositivo).
- Operaciones de control sobre Salidas Digitales, si existen en la remota. Si no existiesen, la remota puede responder OBJECT UNKNOWN
- COLD\_RESARTs

Una remota Nivel 1 puede podría respuestas espontáneas de algunos objetos, pero esta capacidad debe ser configurable.

## 2.1 Tabla de implementación

**Table 14-2—Level 1 implementation (DNP3-L1)**

DNP3 OBJECT GROUP & VARIATION			REQUEST Master may issue Outstation shall parse		RESPONSE Master shall parse Outstation may issue	
Group num	Var num	Description	Function codes (dec)	Qualifier codes (hex)	Function codes (dec)	Qualifier codes (hex)
1	1	Binary Input—Packed format			129 (response)	00, 01 (start-stop)
1	2	Binary Input—With flags			129 (response)	00, 01 (start-stop)
2	1	Binary Input Event—Without time			129 (response) 130 (unsol. resp)	17, 28 (index)
2	2	Binary Input Event—With absolute time			129 (response) 130 (unsol. resp)	17, 28 (index)
2	3	Binary Input Event—With relative time			129 (response) 130 (unsol. resp)	17, 28 (index)
10	0	Binary Output—Any Variation	1 (read)	06 (no range, or all)		
10	2	Binary Output—Output status with flags			129 (response)	00, 01 (start-stop)
12	1	Binary Command—Control relay output block (CROB)	3 (select) 4 (operate) 5 (direct op)	17, 28 (index)	129 (response)	echo of request
			6 (dir op, no ack)	17, 28 (index)		
20	1	Counter—32-bit with flag			129 (response)	00, 01 (start-stop)
20	2	Counter—16-bit with flag			129 (response)	00, 01 (start-stop)
20	5	Counter—32-bit without flag			129 (response)	00, 01 (start-stop)
20	6	Counter—16-bit without flag			129 (response)	00, 01 (start-stop)
22	1	Counter Event—32-bit with flag			129 (response) 130 (unsol. resp)	17, 28 (index)
22	2	Counter Event—16-bit with flag			129 (response) 130 (unsol. resp)	17, 28 (index)

Table 14-2—Level 1 implementation (DNP3-L1)

DNP3 OBJECT GROUP & VARIATION			REQUEST Master may issue Outstation shall parse		RESPONSE Master shall parse Outstation may issue	
Group num	Var num	Description	Function codes (dec)	Qualifier codes (hex)	Function codes (dec)	Qualifier codes (hex)
30	1	Analog Input—32-bit with flag			129 (response)	00, 01 (start-stop)
30	2	Analog Input—16-bit with flag			129 (response)	00, 01 (start-stop)
30	3	Analog Input—32-bit without flag			129 (response)	00, 01 (start-stop)
30	4	Analog Input—16-bit without flag			129 (response)	00, 01 (start-stop)
32	1	Analog Input Event—32-bit without time			129 (response) 130 (unsol. resp)	17, 28 (index)
32	2	Analog Input Event—16-bit without time			129 (response) 130 (unsol. resp)	17, 28 (index)
40	0	Analog Output Status—Any Variation	1 (read)	06 (no range, or all)		
40	2	Analog Output Status—16-bit with flag			129 (response)	00, 01 (start-stop)
41	2	Analog Output—16-bit	3 (select) 4 (operate) 5 (direct op)	17, 28 (index)	129 (response)	echo of request
			6 (dir. op, no ack)	17, 28 (index)		
50	1	Time and Date—Absolute time	2 (write)	07 (limited qty = 1)		
51	1	Time and Date CTO—Absolute time, synchronized			129 (response) 130 (unsol. resp)	07 (limited qty) (qty = 1)
51	2	Time and Date CTO—Absolute time, unsynchronized			129 (response) 130 (unsol. resp)	07 (limited qty) (qty = 1)
52	1	Time Delay—Coarse			129 (response)	07 (limited qty) (qty = 1)
52	2	Time Delay—Fine			129 (response)	07 (limited qty) (qty = 1)



Table 14-2—Level 1 implementation (DNP3-L1)

DNP3 OBJECT GROUP & VARIATION			REQUEST Master may issue Outstation shall parse		RESPONSE Master shall parse Outstation may issue	
Group num	Var num	Description	Function codes (dec)	Qualifier codes (hex)	Function codes (dec)	Qualifier codes (hex)
60	1	Class Objects— Class 0 data	1 (read)	06 (no range, or all)		
60	2	Class Objects— Class 1 data	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
60	3	Class Objects— Class 2 data	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
60	4	Class Objects— Class 3 data	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
80	1	Internal Indications— Packed format	2 (write)	00 (start-stop) index=7		
No Object (function code only)			0 (confirm)			
No Object (function code only)			13 (cold restart)			
No Object (function code only)			23 (delay measurement)			

### 3. Capa de aplicación

#### 3.1 Puntos y tipos de puntos

En DNP3, un punto es una entidad física o lógica unívocamente identificable. El término “punto” aplica a entradas analógicas, digitales y contadores como así también a salidas analógicas y digitales.

Un “tipo de punto” es un medio para categorizar puntos que tienen características relacionadas, funcionalidad similar, y relación al hardware o espacio lógico.

DNP3 modela cada tipo de punto como un arreglo independiente de puntos. Cada punto es identificado por su índice en el arreglo. En la figura se ilustran los cinco tipos básicos de puntos en una remota DNP3.

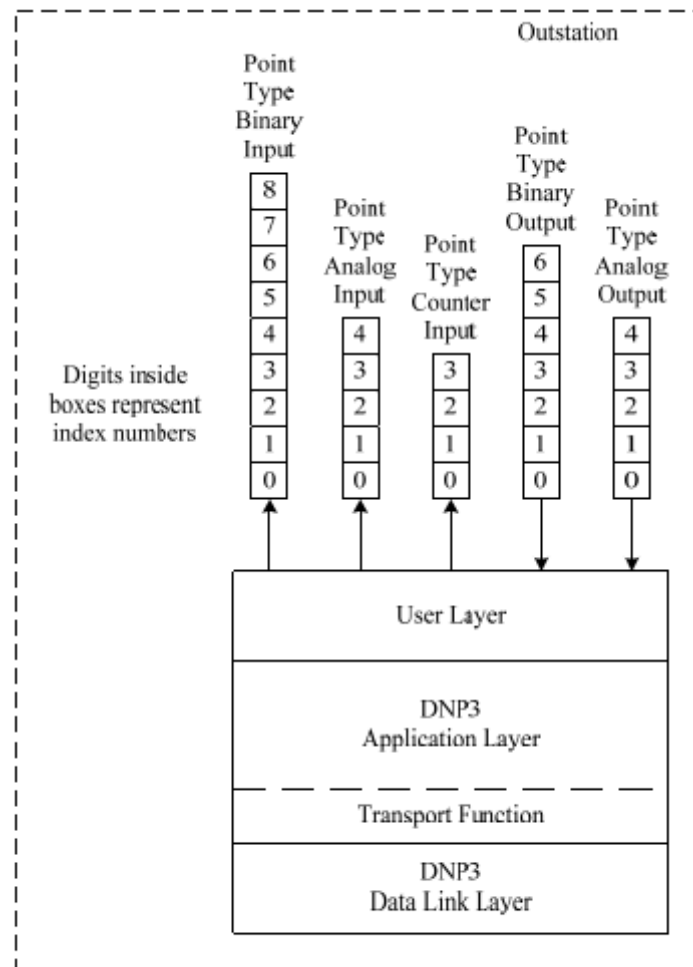


Figure 4-2—Point type arrays

## 3.2 Índices, grupos y variaciones

### 3.2.1 Índices

DNP3 utiliza índices para identificar puntos del mismo tipo. El índice es equivalente a un número de elemento dentro de un arreglo. Los índices DNP3 comienzan en cero.

### 3.2.2 Grupos

Los grupos proporcionan un medio de clasificación del tipo o tipos de datos dentro de un mensaje. Cada miembro de un grupo comparte un tipo de punto y método de generación o creación del dato.

Los números de grupo también son usados para especificar el tipo de datos necesario para reportar valores temporales, control de archivos, caracteres de terminal virtual y otra información.

### 3.2.3 Variaciones

DNP3 ofrece una gama de formatos de codificación para muchos tipos de datos. Las opciones son conocidas como variaciones. Cada grupo tiene un conjunto independiente de variaciones.

## 3.3 Objetos DNP3

Un objeto DNP3 es una representación codificada de datos pertenecientes a un punto, u otra estructura, formateada de acuerdo a su número de grupo y variación para su transporte en un mensaje.

Un mensaje puede contener múltiples objetos, cada uno representando el valor de un punto en un instante de tiempo dado, o el comando a ser emitido en un punto de salida.

### 3.3.1 Estático

En DNP3, el término estático se refiere al valor actual de un punto, el cual es el medido, calculado u obtenido más recientemente.

### 3.3.2 Eventos

Los eventos DNP3 están asociados con la ocurrencia de un suceso relevante. Ejemplo son los cambios de estado, una medición que cruza un umbral, una entrada analógica cambiando más que su banda muerta, etc.

Una remota almacena información sobre los eventos en estructuras propietarias.

Se mencionan a continuación ejemplos de información relacionada a eventos:

- Tipo de evento (binario, analógico, etc.)
- Valor (on, off, 387, etc.)
- Índice del punto
- Tiempo de ocurrencia del evento
- Clase asignada

### 3.3.3 Clases

- DNP3 utiliza el concepto de clases para organizar datos estáticos y eventos en varias categorías:
- Clase 0: Datos estáticos (puede ser un subgrupo del total de datos de la remota)
- Clases 1, 2, 3: Eventos

## 3.4 Inicio de la remota

Regla 1: Una remota indica que ha sido reiniciada estableciendo en uno el bit de indicaciones internas IIN.7 [DEVICE\_RESTART] en sus respuestas a una encuesta del maestro y en respuestas no solicitadas. Debe además establecer en uno la indicación interna de sincronización de tiempo IIN.4 [NEED\_TIME] si requiere sincronización de tiempo. La remota debe establecer en cero IIN.7 solo cuando recibe una petición "Clear Restart" (un mensaje de escritura con objeto 80, variación 1, índice 7) del maestro. Las peticiones "Clear Restart" no tienen efecto si el bit IIN.7 ya se encuentra en cero.

Regla 2: Una remota, configurada para ejecutar reportes no solicitados, debe inmediatamente transmitir una respuesta no solicitada nula sin objetos. Si una confirmación de Capa de Aplicación no es recibida dentro del tiempo de espera configurado, la remota debe retransmitir la respuesta no solicitada nula. Debe seguir retransmitiendo, ignorando el número configurado de retransmisiones de respuestas no solicitadas, hasta que reciba una

confirmación de Capa de Aplicación del maestro, confirmando la recepción de la respuesta no solicitada nula.

Regla 3: Una remota configurada para ejecutar reportes no solicitados no debe enviar datos en una respuesta no solicitada hasta que reciba un petición del maestro conteniendo el código de función ENABLE\_UN SOLICITED que habilita algunos o todos los puntos para iniciar respuestas no solicitadas. Esto no quiere decir que los puntos no generan eventos, solo que los puntos no pueden iniciar un reporte no solicitado de esos eventos.

Regla 4: Una remota no debe descartar eventos luego del reinicio. Debe guardar toda la información de eventos, incluso si está configurada para reportar respuestas no solicitadas y aún no ha recibido un mensaje enable unsolicited responses del maestro.

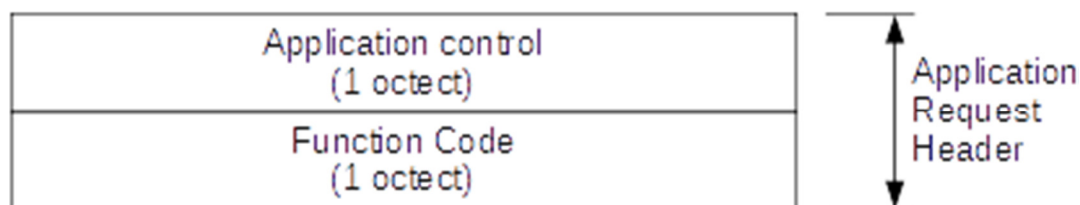
Regla 5: Una remota, configurada para ejecutar reportes no solicitados, debe responder normalmente a cualquier petición que reciba del maestro.

## 3.5 Estructura de los mensajes

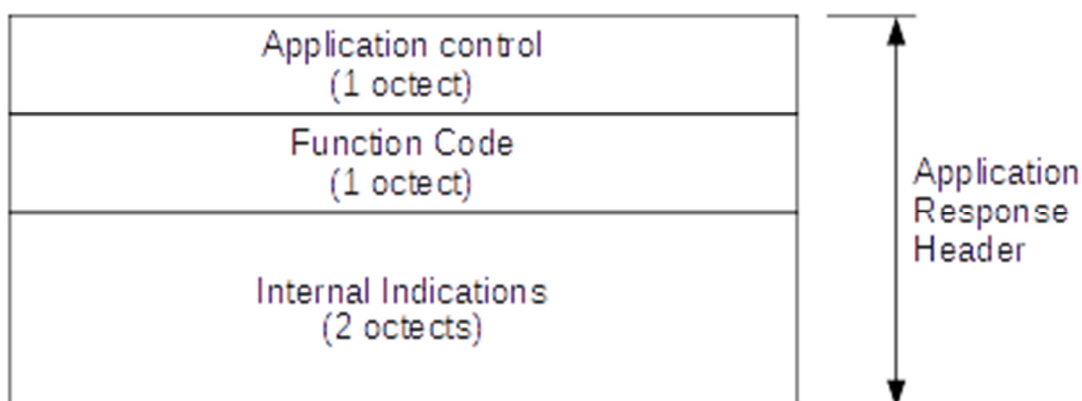
### 3.5.1 Estructura general de los fragmentos



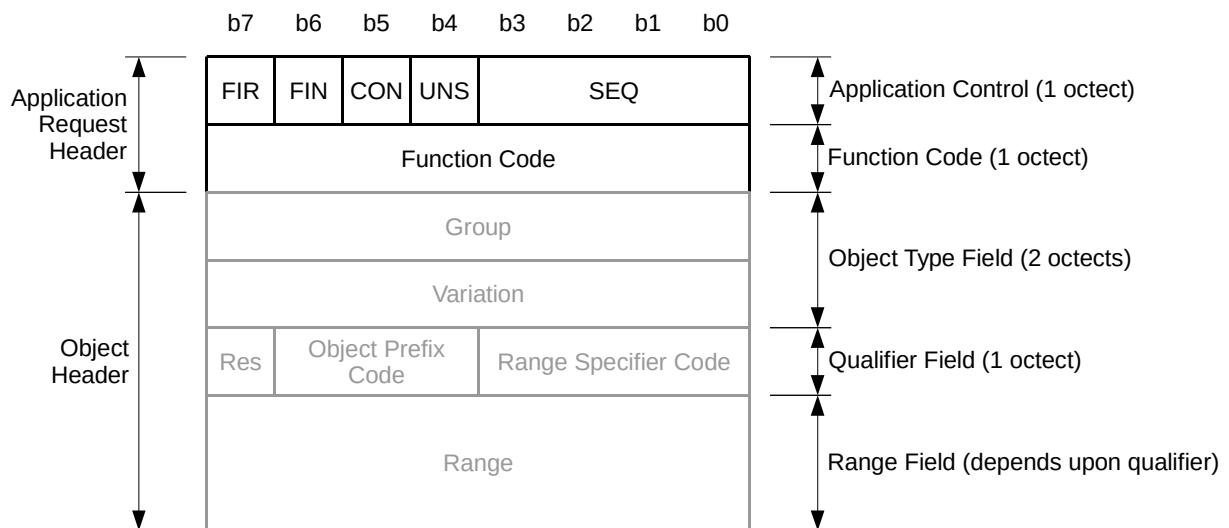
### 3.5.2 Encabezado de solicitud de Aplicación



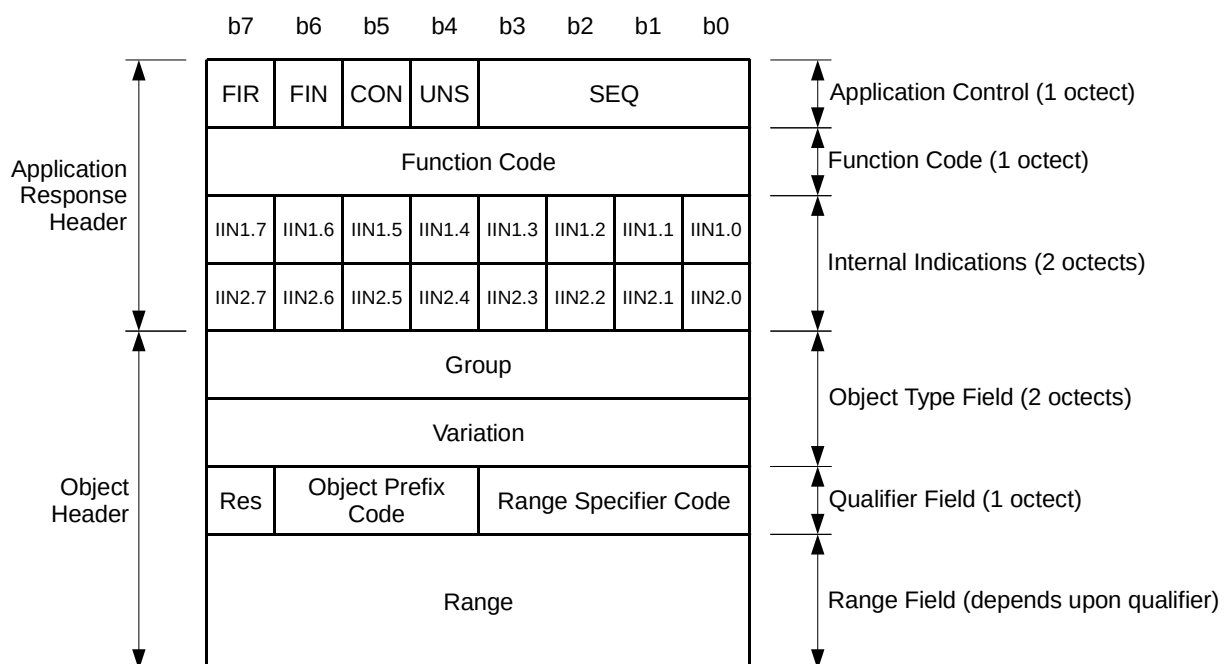
### 3.5.3 Encabezado de respuesta de Aplicación



### 3.5.4 Fragmento de solicitud de Aplicación



### 3.5.5 Fragmento de respuesta de Aplicación



## 3.6 Tabla de estados de fragmento de la remota

El propósito de esta tabla de estados es especificar el comportamiento de una remota con respecto a la recepción y transmisión de fragmentos.

La remota debe examinar los bits FIR, FIN y UNS y el valor SEQ en el byte de control de aplicación de los fragmentos recibidos. Debe además inspeccionar el código de función de Capa de Aplicación. Adicionalmente, necesita recordar el valor SEQ y todos los bytes de los fragmentos de petición que acepta y de los fragmentos de respuesta que transmite.

La remota acepta fragmentos de petición si contienen una petición válida y cumplen los criterios de la tabla, en otro caso, descarta los fragmentos y se mantiene en el mismo

estado. La remota no necesita recordar bytes de control de aplicación o valores de cualquiera de los fragmentos que descarta.

El término petición válida se refiere a un fragmento recibido con un código de función que tiene un valor en el rango de 1 a 128, que está listado en la tabla 6-1 del estándar y que los bits FIR, FIN y UNS tienen los siguientes valores:

- FIR = 1
- FIN = 1
- UNS = 0

La remota debe mantener varias variables locales para cada maestro con el que se comunica.

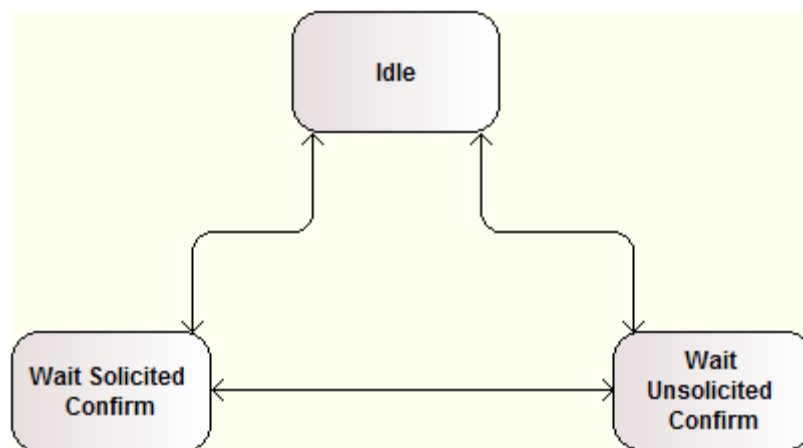
<code>bool FirstValidRequestAccepted</code>	Esta variable booleana es usada para sincronizar el procesamiento de valores SEQ en requerimientos válidos. El valor de esta variable se debe ajustar a falso en el arranque, inmediatamente después del reinicio. Toma el valor verdadero cuando el primer requerimiento válido es recibido.
<code>uint8_t ExpectedConfirmSequenceNumber</code>	ECSN. Si la remota está esperando una confirmación a una respuesta solicitada, el valor de esta variable contiene el número de secuencia que aparece en el octeto Control de Aplicación de la respuesta solicitada esperando confirmación. En otro caso, tiene el valor NothingExpected (NE). Solamente confirmaciones solicitadas que son recibidas con el número de secuencia coincidente con el ECSN son procesadas, las demás son descartadas. Esta variable no tiene ninguna significación para respuestas no solicitadas, aunque su valor puede ser modificado o examinado mientras se espera una confirmación a una respuesta no solicitada.
<code>uint8_t SEQ</code>	El valor del campo SEQ del fragmento de requerimiento válido aceptado más recientemente.
<code>uint8_t LastRxApplicationRequest[]</code>	Bytes de la Capa de Aplicación del fragmento de requerimiento válido aceptado más recientemente.
<code>struct ApplicationControl {     bool FIR;</code>	Valores FIR, FIN, CON y SEQ del

<pre> bool FIN; bool CON; uint8_t SEQ; }LastTxSolicitedResponse; </pre>	fragmento de respuesta solicitada transmitido más recientemente.
<pre> uint8_t LastTxApplicationSolicitedResponse[] </pre>	Bytes de la Capa de Aplicación del fragmento de respuesta solicitada transmitido más recientemente.
<pre> struct ApplicationControl {     bool FIR;     bool FIN;     bool CON;     uint8_t SEQ; }LastTxUnsolicitedResponse; </pre>	Valores FIR, FIN, CON y SEQ del fragmento de respuesta no solicitada transmitido más recientemente.
<pre> uint8_t LastTxApplicationUnsolicitedResponse[] </pre>	Bytes de la Capa de Aplicación del fragmento de respuesta no solicitada transmitido más recientemente.

La remota requiere tres estados para una recepción adecuada.

- Estado Inactivo: El software se encuentra esperando la llegada de un fragmento o un evento que dispare una respuesta no solicitada. En algunos casos, una solicitud diferida puede estar disponible tras ingresar a este estado como consecuencia de las acciones en otro estado. Cuando esta solicitud existe, debe ser procesada de manera inmediata como si hubiera sido recién recibida. El software arranca en estado Idle.
- Estado Esperar confirmación solicitada (WaitSolCfm): El dispositivo recibió una petición que causó que la remota envíe un respuesta que requiere confirmación (ej: incluye eventos o tiene múltiples fragmentos), y la remota está esperando confirmación.
- Estado Esperar confirmación no solicitada (WaitUnsolCfm): El dispositivo transmitió una respuesta no solicitada y está esperando confirmación del maestro. Debe notarse que la espera de una confirmación solicitada puede ocurrir mientras el software está en el estado WaitUnsolCfm dado a un petición del maestro para la cual la remota se encuentra obligada a establecer en uno el bit CON en la respuesta solicitada.

### 3.7 Diagrama de estados de fragmentos de la remota



### 3.8 Reglas de Fragmentos

- Regla 1: Los dispositivos DNP3 que son capaces de establecer su tamaño máximo de fragmento de transmisión mayor a 2048 bytes deben proporcionar la posibilidad de limitar el tamaño (a través de configuración) a un máximo de 2048 bytes.
- Regla 2: El tamaño de fragmento más pequeño que todos los dispositivos deben aceptar de dos bytes.
- Regla 3: Las remotas deben estar preparadas para recibir fragmentos de al menos 249 bytes, mientras que los maestros deben estar preparados para recibir fragmentos de al menos 2048 bytes.
- Regla 4: Los dispositivos maestros sólo deben enviar peticiones que ocupen un solo fragmento. Cualquier maestro que sea capaz de enviar fragmentos de longitud superior a 249 bytes, debe poder configurarse para restringir el tamaño máximo de las peticiones a 249 bytes.
- Regla 5: Los dispositivos maestro deben aceptar respuestas de múltiples fragmentos.
- Regla 6: Una remota debe ser capaz de devolver todos sus eventos y datos estáticos juntos, dentro de una misma respuesta. Si fuera necesario, debe utilizar múltiples fragmentos para transportar la respuesta completa.
- Regla 7: Cada fragmento debe ser individual y completamente analizable sintácticamente. Un fragmento debe contener solo objetos DNP3 completos y porciones, es decir, los objetos no puede ser divididos en dos o más fragmentos.
- Regla 8: El bit FIR debe ser establecido en uno en el fragmento que comienza el mensaje.
- Regla 9: El bit FIN debe ser establecido en uno en el fragmento que finaliza el mensaje.
- Regla 10: Un mensaje puede consistir en un único fragmento que tiene los bits FIR y FIN en uno.
- Regla 11: Los maestros no deben solicitar confirmación de Capa de Aplicación, es decir, no deben poner en uno el bit CON en las peticiones. Esta práctica es obsoleta.
- Regla 12: Las remotas que reciben un fragmento correctamente formado con el bit CON en uno deben responder inmediatamente con un mensaje de confirmación de Capa de Aplicación (por compatibilidad).



- Regla 13: Los maestros que reciben un fragmento correctamente formado con el bit CON en uno deben responder con un mensaje de confirmación de Capa de Aplicación antes de enviar cualquier otra petición a esa remota.
- Regla 14: Por cada nuevo fragmento de petición que envía (no reenviado), el maestro debe incrementar el número SEQ en uno (modulo 16) a partir del fragmento de petición previa.
- Regla 15: El primer fragmento de un respuesta solicitada (encuesta) debe tener el mismo número SEQ que aparece en el campo SEQ de la petición. Si la respuesta requiere múltiples fragmentos, cada fragmento subsiguiente debe utilizar un número SEQ incrementado en uno (modulo 16) a partir del previo.
- Regla 16: Un maestro puede enviar una retransmisión de un mensaje de solicitud si usa el mismo número SEQ y todos los bytes del mensaje retransmitido son iguales al mensaje original. Hay excepciones donde las retransmisiones están prohibidas, específicamente, si la petición incluye:
  - Códigos de función DIRECT\_OPERATE o DIRECT\_OPERATE\_NR
  - Códigos de función DELAY\_MEASURE o RECORD\_CURRENT\_TIME
  - Código de función WRITE con un objeto de Tiempo Absoluto, g50v1, o con un objeto Último Tiempo Grabado, g50v3
- Regla 17: Una remota no debe retransmitir respuestas solicitadas. (Este requerimiento no afecta la capa de Enlace de Datos, la cual puede retransmitir tramas si es apropiado). Una remota puede reenviar respuestas no solicitadas.
- Regla 18: Un fragmento que contiene un código de función CONFIRM debe usar el mismo número SEQ y el mismo estado del bit UNS del fragmento que está siendo confirmado.
- Regla 19: Las remotas deben ignorar el número SEQ en las peticiones de difusión.
- Regla 20: Los números SEQ enviados por una remota en una respuesta no solicitada son diferentes, y no tienen relación con los números SEQ de las respuestas solicitadas.

uint8_t SEQ_UR	Número de secuencia de respuestas no solicitadas
uint8_t SEQ_SR	Número de secuencia de respuestas solicitadas

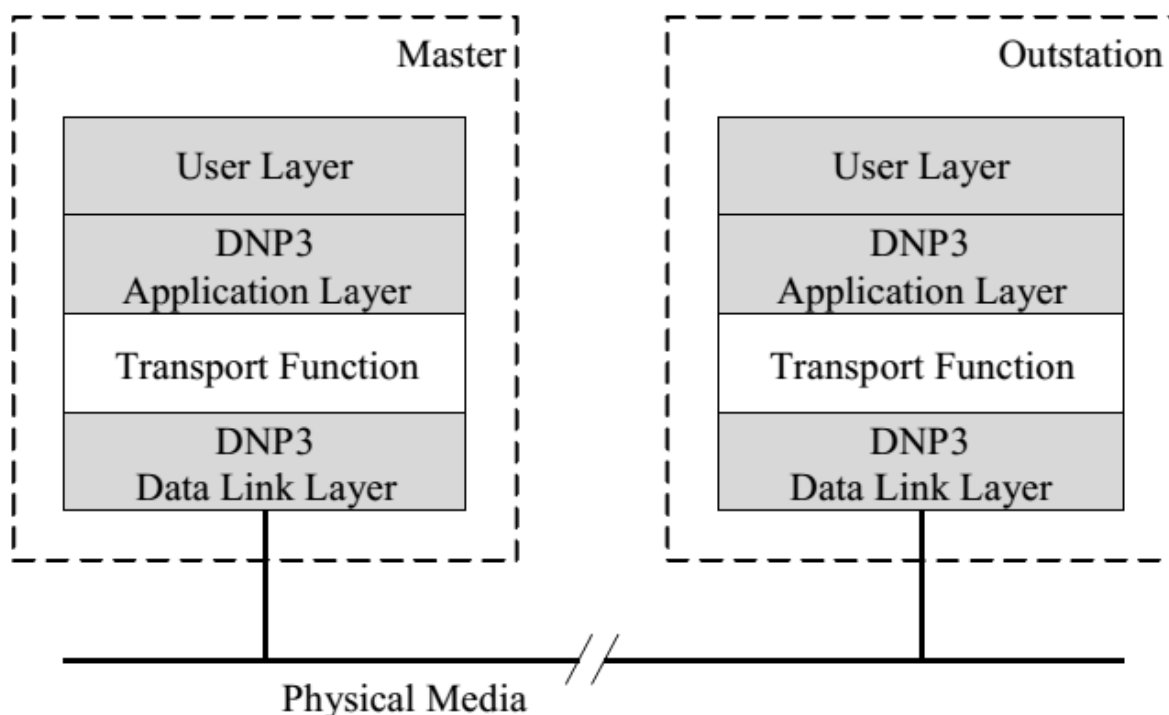
- Regla 21: Una remota que envía respuestas no solicitadas puede elegir cualquier número SEQ para su primer mensaje luego de un reinicio. El maestro debe aceptar estos mensajes.
- Regla 22: Una remota que se reinicia debe ignorar el número SEQ en la primera petición que recibe de un maestro luego del reinicio pero debe ejecutar la petición. A partir de ese momento, la remota debe examinar los números SEQ en las peticiones recibidas.
- Regla 23: Una remota debe pedir una confirmación de Capa de Aplicación cuando envía un fragmento conteniendo objetos de eventos. La recepción del mensaje de confirmación por parte del maestro permite a la remota saber que la información del evento llegó al maestro y que, por lo tanto, la remota puede descartar de sus buffers la información correspondiente al evento.
- Regla 24: Una confirmación de Capa de Aplicación es requerida para cada fragmento un mensaje compuesto por múltiples fragmentos a excepción del último

fragmento. La confirmación de Capa de Aplicación es opcional para el último fragmento de un mensaje compuesto por múltiples fragmentos a menos que haya otra razón por la que la confirmación es obligatoria, tal como si el último fragmento contuviera eventos. La recepción de confirmación significa para la remota que puede enviar el siguiente fragmento. También informa a la remota que puede descartar cualquier información de eventos correspondiente al fragmento confirmado. Las remotas no deben enviar el siguiente fragmento de un mensaje con múltiples fragmentos hasta que el maestro confirme el fragmento previamente enviado.

- Regla 25: La confirmación de Capa de Aplicación es requerida para reconocer la recepción de mensajes de respuestas no solicitadas. Una remota no debe descartar ninguna información o hacer ninguna suposición acerca de lo que el maestro recibió si no recibe un mensaje de confirmación del maestro.
- Regla 26: A veces se requiere que las remotas pidan confirmación de Capa de Aplicación luego de que el maestro envía una petición de difusión. La dirección que el maestro utiliza en la difusión determina la necesidad de confirmación.

## 4. Capa de Pseudo Transporte

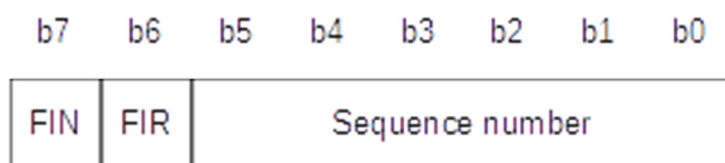
La Función de Transporte es en realidad una subcapa de la capa de aplicación, que se ubica entre esta y la capa de enlace.



El tamaño de un fragmento de mensaje de capa de aplicación puede ser mayor que el número de bytes permitidos en una trama de capa de enlace. La Función de Transporte divide estos fragmentos en unidades de datos del tamaño adecuado para la capa de enlace.

### 4.1 Encabezado

El encabezado de transporte es un único byte. Es además el primer byte del segmento de transporte.



### 4.1.1 Campo FIN:

Cuando este bit se encuentra en uno, indica que transporta el último segmento de un fragmento de aplicación.

### 4.1.2 Campo FIR

Cuando este bit se encuentra en uno, indica que transporta el primer segment de un fragment de aplicación.

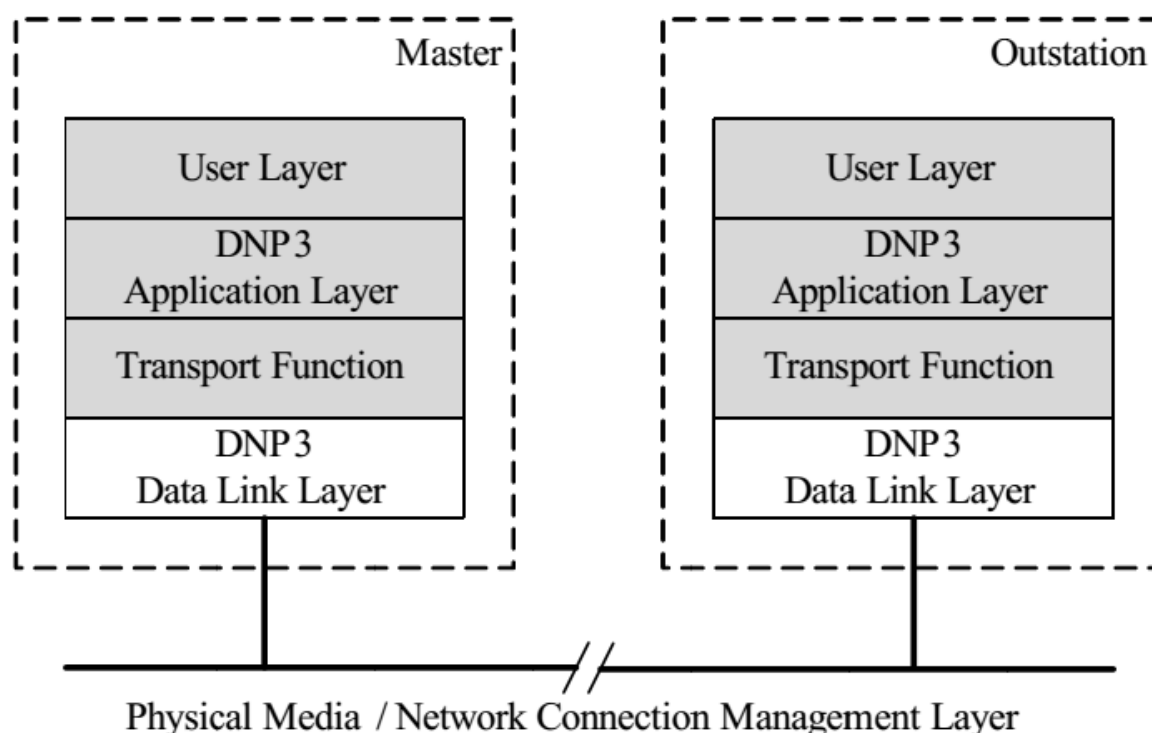
### 4.1.3 Campo número de secuencia

El número de secuencia es un campo de seis bits. Es usado para verificar que los segmentos de transporte son recibidos en el orden correcto y protege contra la duplicación y pérdida de segmentos.

## 5. Capa de Enlace

La capa de enlace brinda la interfaz entre la función de transporte y el medio físico. Los principales aportes de la capa de enlace son el direccionamiento y la detección de errores.

La capa de enlace asume que la comunicación se realiza por un canal representado por un flujo de datos continuo, sin importar el medio de comunicación.



## 5.1 Descripción de la capa de enlace

La capa de enlace tiene dos propósitos principales. En primer lugar, trasladar de manera bidireccional los datos de la capa de aplicación (transporte) al dispositivo destino. Para efectuar esta tarea, la capa de enlace codifica y construye trama.

## 5.2 Servicios

La capa de enlace brinda los siguientes servicios al protocolo:

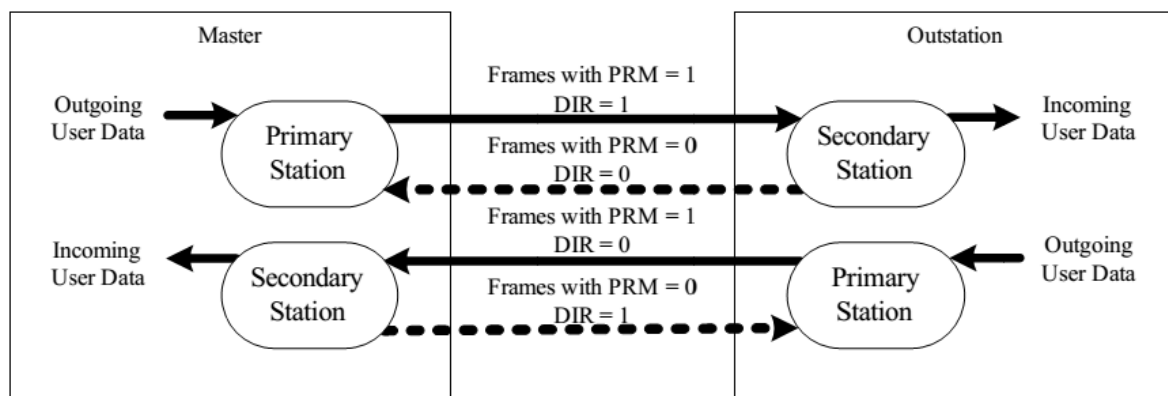
- Encapsulación de los segmentos de transporte en tramas para su transmisión en el canal de comunicación.
- Decodificación de las tramas recibidas en segmentos de transporte.
- Detección de errores.
- Direcccionamiento de origen y destino.

## 5.3 Modelo de transacción

La capa de enlace utiliza transacciones para conducir datos entre dispositivos DNP3 y para efectuar funciones de administración del enlace.

Una transacción consiste en uno o dos mensajes:

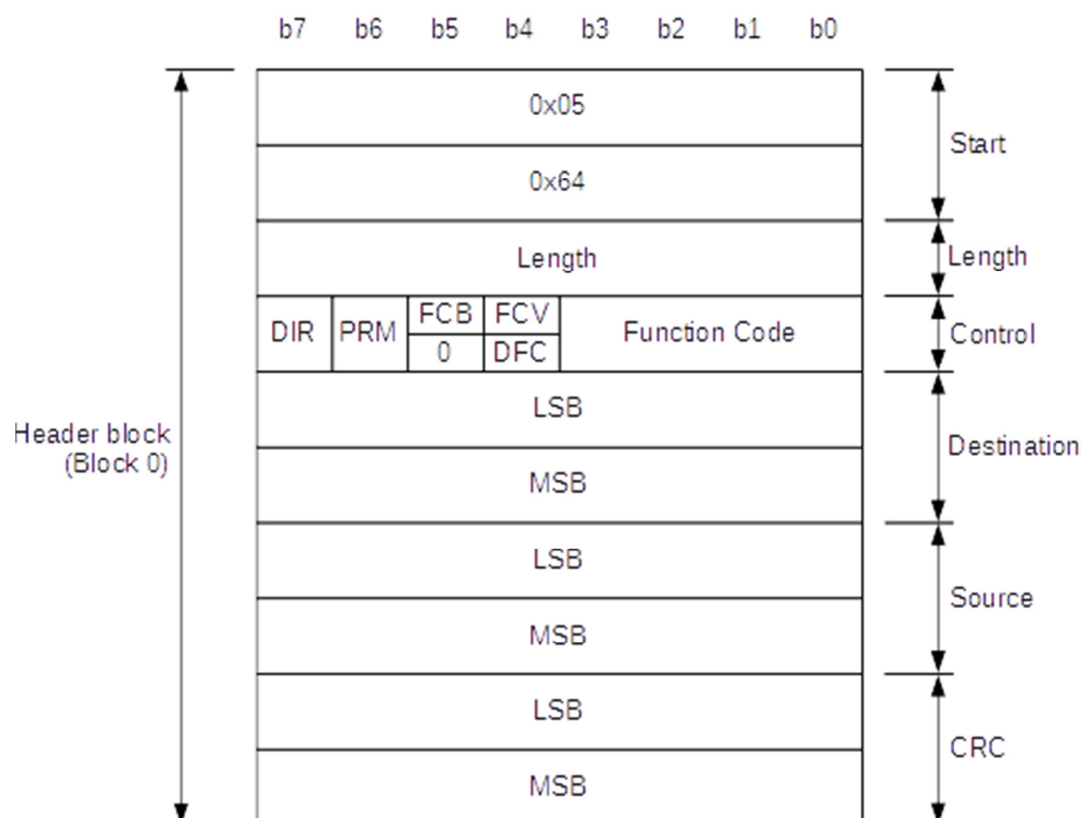
- Un mensaje de solicitud de un dispositivo que inicia la transacción, llamado estación Primaria, a otro dispositivo, llamado estación Secundaria.
- Opcionalmente, un mensaje de respuesta de la estación Secundaria a la Primaria.



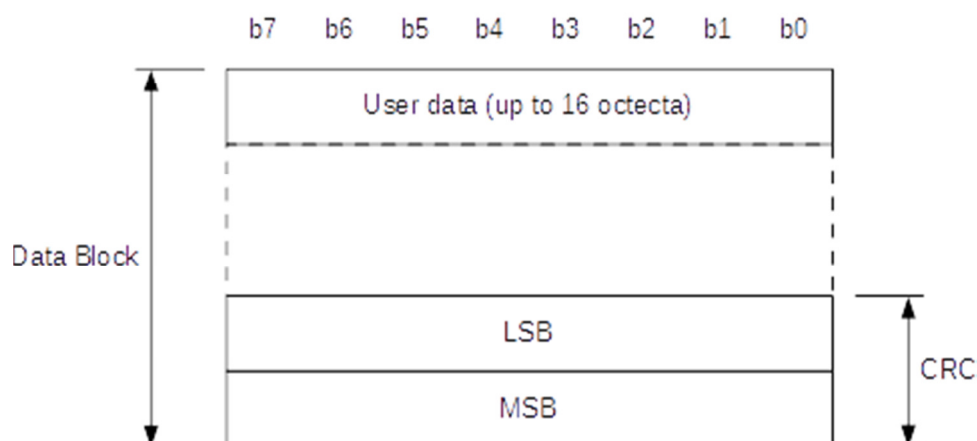
## 5.4 Formato de la trama

Una trama de capa de enlace está compuesta de un bloque de encabezado de longitud fija, block 0, seguido de bloques de datos opcionales. Cada bloque finaliza con un CRC.

### 5.4.1 Encabezado



### 5.4.2 Bloque de datos



#### 5.4.2.1 Campo START

Tiene dos bytes de longitud. El primero es 0x05 y el segundo es 0x64.

#### 5.4.2.2 Campo LENGTH

Especifica la cantidad de bytes, exceptuando CRC, que continúan en el encabezado y en los bloques de datos. El mínimo valor para este campo es cinco, indicado que solo en encabezado está presente y el máximo es 255.

### 5.4.2.3 Campo CONTROL

Contiene información acerca de la dirección de la trama, el iniciador de la transacción, control de errores, control de flujo y función.

#### 5.4.2.3.1 Campo DIR

Este bit es independiente de la estación que origina la trama (Primaria o Secundaria).

- DIR = 1 indica que la trama se originó en un maestro
- DIR = 0 indica que la trama se originó en una remota

#### 5.4.2.3.2 Campo PRM

Indica la dirección de la trama con respecto a la estación que la originó.

- PRM = 1 indica estación primaria.
- PRM = 0 indica estación secundaria.

#### 5.4.2.3.3 Campo FCB

Se utiliza en conjunto con FCV para mantener sincronización para servicios de datos de usuarios confirmados. Este bit sólo es válido cuando la solicitud es enviada de una estación Primaria y con el bit FCV en uno.

#### 5.4.2.3.4 Campo FCV

Indica si la estación Secundaria debe evaluar el bit FCB.

- FCV = 1 indica FCB válido
- FCV = 0 indica ignorar FCB

Campo DFC

Se utiliza para reportar búfers llenos o que la capa de enlaces está ocupada.

#### 5.4.2.3.5 Campo FUNCTION CODE

Identifica la función o servicio asociado a la trama. Depende de si la trama es emitida por una estación Primaria o Secundaria.

##### 5.4.2.3.5.1 Códigos de función – Primario a Secundario

```
typedef enum
{
    CIAA_DNP_PRI_FC_RESET_LINK_STATES = 0,
    CIAA_DNP_PRI_FC_TEST_LINK_STATES = 2,
    CIAA_DNP_PRI_FC_CONFIRMED_USER_DATA = 3,
    CIAA_DNP_PRI_FC_UNCONFIRMED_USER_DATA = 4,
```

```
CIAA_DNP_PRI_FC_REQUEST_LINK_STATUS = 9  
}ciaaDnp_PrimaryFCEnum;
```

#### 5.4.2.3.5.2 Códigos de función – Secundario a Primario

```
typedef enum  
{  
    CIAA_DNP_SEC_FC_ACK = 0,  
    CIAA_DNP_SEC_FC_NACK = 1,  
    CIAA_DNP_SEC_FC_LINK_STATUS = 11,  
    CIAA_DNP_SEC_FC_NOT_SUPPORTED = 15  
}ciaaDnp_SecondaryFCEnum;
```

#### 5.4.2.4 Campo DESTINATION

Es un campo de dos bytes de longitud y especifica la dirección de la estación a la que está dirigida la trama.

#### 5.4.2.5 Campo SOURCE

Es un campo de dos bytes de longitud y especifica la dirección de la estación donde se originó la trama.

### 5.4.3 Campos CRC

Código de redundancia cíclica incorporado en cada bloque de encabezado y datos.