



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

## **APLICACIÓN DE CONTROL DE ENLACES DE DATOS SOBRE REDES DISTRIBUIDAS**

**Rodolfo Fernando de León Miranda**

Asesorado por el Ing. Mario Alberto Miranda

Guatemala, agosto de 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**APLICACIÓN DE CONTROL DE ENLACES DE DATOS SOBRE REDES  
DISTRIBUIDAS**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**RODOLFO FERNANDO DE LEÓN MIRANDA**

ASESORADO POR EL ING. MARIO ALBERTO MIRANDA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO ELECTRÓNICO**

GUATEMALA, AGOSTO DE 2005

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**FACULTAD DE INGENIERÍA**



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Ing. Enrique Edmundo Ruiz Carballo
EXAMINADOR	Ing. Julio César Solares Peñate
EXAMINADOR	Ing. Marvin Marino Hernández Fernández
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

**HONORABLE TRIBUNAL EXAMINADOR**

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**APLICACIÓN DE CONTROL DE ENLACES DE DATOS SOBRE REDES  
DISTRIBUIDAS,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 8 de abril de 2005.

Rodolfo Fernando de León Miranda

## **RECONOCIMIENTO A:**

- Profa. Gloria Miranda, por su apoyo y comprensión en todo momento, privilegios que me permitieron hacer realidad esta meta.
- Ingeniero Mario Miranda, por la ayuda proporcionada a través de mi carrera y los consejos brindados durante esta experiencia del trabajo de graduación.
- Mi familia y amigos, por la colaboración desinteresada que me han brindado.

# ÍNDICE GENERAL

<b>ÍNDICE DE ILUSTRACIONES.....</b>	<b>VI</b>
<b>GLOSARIO.....</b>	<b>IX</b>
<b>RESUMEN.....</b>	<b>XVI</b>
<b>OBJETIVOS.....</b>	<b>XVII</b>
<b>INTRODUCCIÓN.....</b>	<b>XVIII</b>

## **1. CLASIFICACIÓN DE LOS PROTOCOLOS**

1.1	Comunicación de datos.....	1
1.2	Orientación de los Protocolos.....	3
1.2.1	Protocolo orientado a carácter.....	3
1.2.2	Protocolo orientado a conteo de byte.....	3
1.2.3	Protocolo orientado a bit.....	3
1.3	Funciones de los protocolos.....	3
1.3.1	Control de transferencias de datos.....	4
1.3.2	Revisión de error y recuperación.....	4
1.3.3	Utilización de línea.....	4
1.3.4	Sincronización.....	5
1.3.5	Facilidad de transparencia de comunicación.....	5
1.4	Capas de comunicación de interconexión de sistemas abiertos.....	5
1.5	Desarrollo de DNP.....	8
1.5.1	Elementos de una red distribuida.....	8
1.5.2	Estándares usados como base para DNP3.....	9
1.6	Capas de DNP3.....	10

1.6.1	Relación Maestro-Esclavo.....	10
1.6.2	Software DNP3.....	10

## **2. CAPA DE ENLACE DE DATOS**

2.1	Capa de enlace DNP3.....	13
2.2	Formato de trama.....	14
2.2.1	Formato de trama FT3.....	14
2.3	Campos de encabezado de trama .....	15
2.3.1	Inicio.....	15
2.3.2	Longitud.....	15
2.3.3	Control.....	15
2.3.4	Dirección destino.....	18
2.3.5	Dirección fuente.....	18
2.4	Datos de Usuario.....	19
2.5	Campos de revisión de redundancia cíclica.....	19
2.6	Códigos de Funciones de enlace de datos.....	21
2.6.1	Reajuste.....	21
2.6.1.1	Transacción primaria.....	22
2.6.1.2	Transacción secundaria.....	23
2.6.2	Reajuste para procesos de usuario.....	23
2.6.3	Prueba.....	24
2.6.4	Datos de usuario.....	24
2.6.4.1	Transacción primaria.....	24
2.6.4.2	Transacción secundaria.....	25
2.6.5	Datos de usuario sin confirmación.....	25
2.6.5.1	Transacción primaria.....	26
2.6.5.2	Transacción secundaria.....	26
2.6.6	Requerimiento de estado de enlace.....	26

2.7	Servicios y responsabilidades de enlace de datos.....	26
2.8	Capa física de interfaz.....	28
2.8.1	Descripción.....	28
2.8.2	Características de la capa física.....	29
2.8.2.1	Configuraciones de línea.....	29
2.8.3	Modos de Transmisión.....	30
2.8.4	Capa Física.....	30
2.9	Procedimientos de Transmisión.....	31
2.9.1	Reajuste de enlace secundario.....	31
2.9.2	Reajuste de procesos de usuario.....	32
2.9.3	Envío Confirmación de datos de usuario.....	32
2.9.4	Envío sin respuesta esperada.....	35
2.9.5	Envío/NACK.....	36
2.9.6	Requerimiento/Respuesta.....	37

### **3. FUNCIONES DE TRANSPORTE**

3.1	Encabezado de transporte.....	40
3.2	Campos de encabezado de transporte.....	41
3.2.1	Bit final.....	41
3.2.2	Bit de inicio.....	41
3.2.3	Secuencia.....	42
3.3	Ensamblado de trama.....	42
3.4	Transmisión de Mensajes.....	43
3.5	Servicios de transporte y Responsabilidades.....	45

### **4. CAPA DE APLICACIÓN**

4.1	Formato de mensajes de requerimiento y respuesta.....	48
-----	---	----

4.1.1	Formato de requerimiento de aplicación.....	51
4.1.2	Formato de respuesta de aplicación.....	52
4.2	Campos de Mensaje DNP.....	53
4.2.1	Encabezados de aplicación.....	53
4.2.1.1	Encabezado de requerimiento y respuesta.....	53
4.2.1.2	Control de aplicación.....	53
4.3	Control de flujo de información.....	54
4.4	Códigos de función.....	57
4.5	Encabezado de objeto.....	60
4.6	Librería de datos-objeto .....	61
4.6.1	Reglas de elementos de información.....	61
4.6.1.1	Tipos de datos.....	62
4.6.1.2	Tamaño de dato.....	62
4.6.1.3	Posición de bit.....	62
4.6.2	Reglas y estructura de librería.....	63
4.6.2.1	Estructura de librería.....	63
4.6.2.2	Numeración de punto. ....	64
4.6.2.3	Objetos y variaciones.....	65
4.6.2.3.1	Entradas y salidas binarias.....	66
4.6.2.3.2	Contadores.....	68
4.6.2.3.3	Entradas y salidas análogas.....	69
4.6.2.3.4	Objetos de fecha y tiempo.....	71
4.6.2.3.5	Definición de los objetos de clase.....	72
4.6.2.3.6	Objeto de archivo.....	72
4.6.2.3.7	Objetos de dispositivo.....	73
4.6.2.3.8	Objetos de aplicación.....	74
4.6.2.3.9	Objeto de numeración alternada.....	74

## **5. CRITERIOS E INTERPRETACIONES DE LAS CAPAS DEL PROTOCOLO APLICADO A MEDICIONES Y MONITOREO DE UNIDADES TERMINALES REMOTAS**

5.1	Niveles de implementación.....	78
5.1.1	Implementación de nivel 1.....	78
5.1.2	Implementación de nivel 2.....	78
5.1.3	Implementación de nivel 3.....	79
5.2	Requerimientos de usuario.....	79
5.3	Direccionamiento.....	82
5.4	Eventos.....	83
5.4.1	Clases de datos y eventos.....	83
5.4.2	Configuración de eventos.....	84
5.5	Alarmas.....	84
5.6	Configuración remota.....	84
5.6.1	Respuestas de error.....	84
5.6.2	Perfil de dispositivo.....	85
5.6.3	Archivos de configuración.....	87
5.6.4	Consideraciones del dispositivo esclavo.....	88
5.6.5	Funciones de aceptación.....	90
5.7	Notas de aplicación.....	92
<b>CONCLUSIONES.....</b>		<b>102</b>
<b>RECOMENDACIONES.....</b>		<b>104</b>
<b>BIBLIOGRAFÍA.....</b>		<b>105</b>
<b>ANEXO.....</b>		<b>106</b>



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1	Modelo simplificado para las comunicaciones.....	1
2	Modelo de referencia OSI.....	6
3	Unidades de datos que intervienen en la formación de una red OSI.....	8
4	Capas de software DNP.....	11
5	Formato FT3.....	14
6	Octeto de control.....	15
7	Formato de dirección y fuente.....	18
8	Orden de CRC.....	21
9	Reajuste de enlace de secundario.....	31
10	Reajuste de procesos de usuario.....	32
11	Envío desde la estación A / Confirmación desde la estación B.....	33
12	Envío desde la estación B / Confirmación desde la estación A.....	33
13	Múltiples Tramas de envío desde la estación A confirmación desde la estación B.....	34
14	Operación del bit de conteo de trama FCB Primaria-Secundaria.....	34
15	Operación del bit de conteo de trama FCB Primaria-Primaria.....	35
16	Envío sin respuesta esperada.....	36
17	Envío desde la estación B / NACK desde la estación A.....	37
18	Trama de requerimiento / Respuesta y uso del bit DFC.....	38
19	Estructura de un TPDU.....	40
20	Definiciones de los bits de TH.....	41
21	Ensamblado de datos de tres tramas de datos.....	43
22	Transmisión de un trama de mensaje.....	44

23	Fragmentación de un mensaje de aplicación de múltiples tramas.....	45
24	Contexto EPA.....	47
25	Secuencia de mensaje de la capa de aplicación (Maestra y remota).....	49
26	Formato de requerimiento de aplicación.....	51
27	Formato de respuesta de aplicación.....	52
28	Encabezado de requerimiento y abajo el encabezado de respuesta.....	53
29	Campo de control de aplicación.....	54
30	Encabezado de objeto.....	60
31	Estructura del campo calificador.....	60
32	Posición de bits.....	63
33	Sistema de comunicación.....	94
34	Ventana de interfaz gráfica con los procesos involucrados en las solicitudes y respuestas con una imagen de su equivalente en Hexadecimal.....	100
35	Ventana de interfaz con el reporte de los datos objeto solicitados.....	101
36	Referencia para el control de enlace de datos DNP, lista válida de campos.....	109
37	Vista general de un APDU en el contexto DNP.....	110

## TABLAS

I	Valores de campo de códigos de función del octeto de control enviado desde la estación primaria.....	17
II	Valores de campo de códigos de función del octeto de control enviado desde la estación secundaria.....	18
III	Códigos de función de transferencia.....	57
IV	Códigos de función de control.....	57

V	Códigos de función congelados.....	58
VI	Códigos de función de transferencia.....	58
VII	Códigos de función de aplicación.....	58
VIII	Códigos de función de configuración.....	59
IX	Códigos de función de respuesta, reservados y tiempo de sincronización.....	59
X	Tipos de datos.....	62
XI	Valores definidos para el campo de las unidades.....	75
XII	Mensajes de error.....	84
XIII	Mapa de objetos y variaciones para los datos solicitados en la prueba.....	97
XIV	Tabla del subconjunto de nivel de implementación DNP.....	111



## GLOSARIO

<b>ACK</b>	Es un carácter de control de ocho bits, del juego de caracteres ASCII, utilizado durante la transmisión de un mensaje indicando que la última comprobación de paridad fue válida.
<b>ADCCP</b>	Advanced Data Communications Control Procedure. Protocolo estándar dentro de los EEUU.
<b>Analizar el mensaje</b>	Significa que un dispositivo puede quebrar el mensaje en piezas y cada una consiste de un encabezado o datos correspondientes.
<b>Ancho de banda</b>	Es un indicador de la cantidad de datos que pueden transmitirse en determinado período de tiempo por un canal de transmisión.
<b>BISYNC</b>	(Binary Synchronous Communications). Protocolo de comunicación en la cual los datos son transmitidos en una serie de bytes, el protocolo usa caracteres de control que tiene especial significado.
<b>Bit</b>	Una contracción de dos términos del inglés Binary digIT.
<b>Bloque de Caracteres</b>	Es un número de datos contiguos que forman un mensaje o parte de él.

<b>Buffer</b>	Dispositivo de memoria de almacenamiento temporal diseñado para compensar la diferencia de rangos de datos entre el dispositivo transmisor y el dispositivo receptor.
<b>Carácter</b>	Es un byte con una identidad. Un grupo de bits, usualmente siete u ocho bits, que representan una simple letra, dígito, símbolo especial, o código de control en un esquema codificado como el ASCII.
<b>Código binario</b>	Es un juego de bits que forman parte de un sistema de conversión que determina los estados de la señal.
<b>Collision Avoidance/detection</b>	Técnica basada en detección de colisión. Cada nodo en la red escucha primero y transmite solamente cuando las líneas están libres. Para evitar una colisión, cada nodo espera por una cantidad de tiempo aleatoria antes de intentar transmitir cuando las líneas están ocupadas.
<b>Concentrador</b>	Dispositivo que conecta varios enlaces de comunicaciones con grupo menor de circuitos para obtener transmisiones relativamente económicas.
<b>Configuración</b>	Inicializar un dispositivo, para una operación en forma particular. El dispositivo se puede configurar usando mecanismos como por ejemplo: ROM, dip switches, jumpers de hardware.
<b>CRC</b>	(Cyclic Redundancy check). Es un método de detección de error basado sobre una transformación de los valores de los bits en un paquete de datos o trama.

<b>DCE</b>	(Data Communications Equipment). Es un dispositivo capaz de comunicarse con el apropiado DTE, y provee acceso al apropiado tipo de línea.
<b>DDCMP</b>	(Digital Data Communications Message Protocole). Protocolo orientado a byte; protocolo síncrono de enlace de datos.
<b>Dial Up</b>	Es una línea no dedicada de comunicación en la cual una conexión puede ser establecida por el marcado de un número, o código asociado con el destino. Un ejemplo común de una línea dial up es la línea de telefonía pública
<b>Direccionamiento</b>	Identificación única de un número (o locación) para la fuente o destino de los datos.
<b>Distancia Hamming</b>	Es un código de corrección el cual trabaja por la incersión de bits extra en una locación predeterminada de transmisión. Matemáticamente, el espacio y los valores de estos bits hace posible determinar si un error ha ocurrido, dónde está el error, y cómo corregirlo.
<b>DTE</b>	(Data Terminal Equipment). Dispositivo que puede comunicarse con un DCE.
<b>Esclavo(a)</b>	Dispositivo que reúne datos o realiza operaciones de control en respuesta a un requerimiento del maestro y envía un mensaje de repuesta.
<b>Frame Relay</b>	Estándar para transmisión de datos solamente. Provee intercambio de paquetes dejando la revisión (check) y monitoreo a protocolos

de alto nivel. Tiene alto throughput y bajos retardos. Es también eficiente haciendo el máximo uso del ancho de banda disponible

**FSK** (Frequency Switch Key). Técnica de modulación digital para representar estados binarios modulando la frecuencia de la portadora.

**HDB3** High Density Bipolar 3 Zeros; técnica de codificación digital.

**HDLC** (High-level Data Link Control). Protocolo internacional definido por la ISO, está orientada a bit, protocolo sincrónico que provee corrección de error en la capa de enlace de datos.

**Interfaz** Una interfaz de software es una conexión entre dos programas o dos elementos de programa, como procedimientos o funciones. Una interfase de hardware, es una conexión entre dos dispositivos, una interfase de hardware requiere especificaciones físicas, eléctricas y funciones que determinan cómo los dispositivos se conectan y comunican.

**IP** (Internet Protocol), Protocolo de Internet.

**Loop** Es un enlace cerrado (lazo) de comunicaciones unidireccional que conecta un equipo consigo mismo, pasando a través de otros dispositivos dependientes del primero.

**Maestro(a)** Dispositivo que inicia requerimientos para reunir datos o realizar controles.

<b>NACK</b>	Es un carácter de control de 8 bits, del juego de caracteres ASCII, utilizado durante la transmisión de un mensaje indicando que la última comprobación no valida.
<b>Objeto</b>	Este campo considerado un elemento de información descrito en la capa de aplicación.
<b>Octeto</b>	Objeto de datos de 8 bits sinónimo de byte.
<b>OSI</b>	(Open System Interconnection). Modelo de siete capas de comunicación entre dispositivos compatibles.
<b>Paquetes</b>	Bloque de bytes, los cuales consisten de encabezado, datos y cola. En una red por capas los paquetes creados en un nivel pueden ser insertados en otra cabecera/cola que envuelve al nivel inferior.
<b>Polling</b>	Método para obtener datos, y consiste en que se revisa periódicamente una dirección de un dispositivo para ver si hay algún dato nuevo. En otras palabras, es como si se le preguntara al dispositivo cada cierto tiempo ¿tiene algo nuevo para enviarme?. Si hay alguna información debe proceder a leerla y si no simplemente continúa con otras actividades.
<b>Requerimiento</b>	Este campo describe objetos, códigos de función y calificadores que un dispositivo esclavo debe ser capaz de analizar
<b>Señal</b>	Una señal eléctrica toma la forma de un cambio en voltaje o corriente a través del tiempo. La señal es descrita por niveles, o

amplitudes, que el voltaje o corriente alcanza y por el patrón con el cual este nivel cambia a través del tiempo.

<b>Servidor</b>	Generalmente, un servidor es una entidad que provee algún tipo de servicio de red. El servidor puede estar en hardware, como servidor de archivo en una red, o software, como protocolo a nivel de red para nivel de transporte de cliente. Los servicios se pueden acceder a los archivos o servicios. Los servidores proveen sus servicios a otras máquinas (estaciones de trabajo) en la red o para otros procesos.
<b>SNA</b>	(Systems Network Architecture). Arquitectura diseñada para habilitar cualquier máquina IBM comunicarse con cualquier otra. En particular SNA fue desarrollado para habilitar la comunicación de varias máquinas con tramas de IBM.
<b>Software</b>	Programas en una computadora o microcomputadora.
<b>Throughput</b>	Comportamiento de velocidad de la línea de un circuito virtual durante su uso normal y pico.
<b>Trama</b>	Es un término usado para un paquete de datos, particularmente para un paquete en la capa de enlace de datos del modelo OSI.
<b>Unidad Terminal Remota</b>	Una pieza de equipo localizada a una distancia de la estación maestra para monitor y control de estado de equipo exterior, y para comunicar la información de retorno a la estación maestra. Abreviado como RTU.

## RESUMEN

El propósito de este trabajo es describir las características de un protocolo abierto como lo es DNP3, que se ha desarrollado en el modelo por capas para cumplir con los estándares propuestos por entidades internacionales en una arquitectura de desempeño elevado, expuesta a través de esta documentación.

El tema inicial se enfoca en una reseña de los tipos de protocolo según su orientación, y las ventajas de los controles de enlaces de datos sobre otras técnicas por su flexibilidad en la facilidad de transparencia, además que se hace mención del modelo de referencia por capas de sistemas abiertos interconectados, exponiendo las bases sobre el cual se apoya el protocolo ubicando los parámetros que están involucrados en DNP3.

La capa de enlace de datos expone las diferentes combinaciones de bit que formarán los octetos que guían el proceso del envío y recepción de tramas, acentuando los conceptos de requerimiento y respuesta. En la capa de transporte, se describe la información de control que se añade a las tramas a fin de transmitir y recuperar los mensajes enviados y recibidos.

La capa de aplicación se compone de la descripción de los mensajes DNP y se presentan los datos objetos, y los cambios generados en los mismos que implican eventos que agrupados según las funciones, son utilizados en los niveles de implementación del protocolo proporcionando criterios para el desempeño de los requerimientos de los usuarios y considerarlas como herramientas para la interacción de los dispositivos en la adquisición de datos.



## **OBJETIVOS**

### **General**

- Dar a conocer la aplicación de las capas DNP utilizadas en la comunicación de datos para la industria de potencia, que está dirigida hacia la automatización de procesos remotos.

### **Específicos**

- Dar a conocer los elementos básicos de este protocolo debido a su aplicación en la red eléctrica nacional.
- Analizar las capas que componen su estructura, y características que presenta.
- Proporcionar criterios técnicos que brinden información que permita considerarse como herramientas selectivas.



## INTRODUCCIÓN

Para tomar la mejor decisión en cualquier circunstancia, una herramienta idónea es la información; tener información es considerado un activo para una empresa y trasciende hasta límites personales, es por eso que la transmisión de datos tiene una función importante tomando en cuenta que el hombre, siempre ha querido estar comunicado llevando el desarrollo de las comunicaciones a un nivel en el que actualmente el flujo de información es gobernado por computadoras y dispositivos basados en lenguajes de programación y por qué no decir, lenguaje-maquina.

Un protocolo es un lenguaje que proporcionará las reglas por las cuales los dispositivos intercambian datos unos con otros y, si se trata de protocolos de transmisión de datos, existen para una amplia y general aplicación, ya sea hipertexto o archivos grandes. La adquisición de datos entre dispositivos que forman una red, parece tener como parte fundamental el uso de protocolos que sean diseñados para optimizar la transmisión de datos, empleando comunicaciones seriales y que es utilizado en las grandes compañías como Microsoft, Petroleras y compañías eléctricas, Fábrica de aviones, entre otras.

Una compañía eléctrica tiene centrales de operaciones que monitorean el estado de todo dispositivo eléctrico. En el centro de operaciones, computadoras de gran capacidad almacenan los datos entrantes y los despliegan sobre una interfaz entendible para los operadores humanos. En el caso más general se podría describir que uno o más ordenadores están situados en las subestaciones para recolectar los datos para la transmisión al centro de operaciones. Las computadoras en las subestaciones reúnen datos tales como: datos de entrada binarios usado para dispositivos de dos estados, datos

de entrada analógicos, contadores de entrada de datos, archivos de configuraciones, sincronización, tiempo y fechas, historial de datos, por mencionar las más usuales.

Este trabajo estudiará las características del protocolo que provee las reglas para que las computadoras de las subestaciones y las centrales de operaciones comuniquen datos y comandos de control.

Básicamente, el término de servidor es un dispositivo o programa que tiene datos o información que alguien más quiere. Un cliente es un dispositivo o programa que requiere información del servidor. Una computadora en una subestación es un servidor y se denomina esclavo, y en el centro de operaciones o cliente se entenderá que corresponde a maestro, dada la terminología que comprenderá este protocolo.

Esta investigación describirá las relaciones cliente-servidor en las capas del protocolo, debido a que en el ámbito nacional se encuentra una aplicación de éste para los requerimientos o respuestas en la red eléctrica.

Un protocolo es conocido como protocolo de línea o controles de enlace de datos, razón por la cual las capas DNP en este trabajo se estudiarán para describir sus características y funciones de servicio.

Además, los sistemas de comunicaciones son imperfectos, en este trabajo se describirán las características de este protocolo para proporcionar una fácil transmisión y una descripción básica de las configuraciones donde el protocolo puede ser usado.

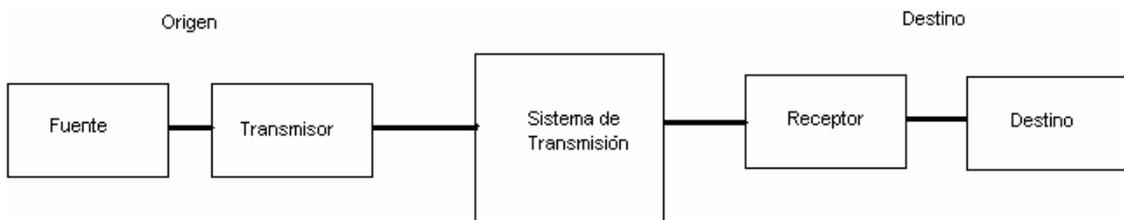
# 1. CLASIFICACIÓN DE LOS PROTOCOLOS

## 1.1 COMUNICACIONES DE DATOS

En el campo de las computadoras (procesamiento de datos) y el de las comunicaciones de datos (la transmisión y sistemas de conmutación), no hay grandes diferencias. En efecto, la tendencia de lo anterior se puede observar entre las industrias de las comunicaciones y de las computadoras, desde la fabricación de componentes hasta la integración de sistemas, que transmiten y procesan todo tipo de datos e información.

El objetivo principal de todo sistema de comunicaciones es intercambiar información entre dos puntos. Los elementos clave para un modelo de sistema de comunicación consisten en: La fuente, que es el dispositivo que genera los datos a transmitir. El transmisor, que los transforma y codifica (la información); el sistema de transmisión puede ser desde una sencilla línea de transmisión hasta una compleja red. El receptor acepta la señal que proviene del sistema de transmisión y la transforma para que pueda ser manejada por el dispositivo destino, que toma los datos del receptor.

Figura 1. Modelo simplificado para las comunicaciones



El modelo parece sencillo pero implica una serie de tareas que debe realizar un sistema de comunicaciones; para comenzar, la utilización del sistema de transmisión se refiere a la necesidad de hacer uso eficaz de los recursos utilizados en la transmisión, los cuales típicamente se suelen compartir en una serie de dispositivos de comunicación. Para que un dispositivo pueda transmitir información tendrá que hacerlo a través de la interfaz con el medio de transmisión. Una vez que la interfaz está establecida se necesitará la generación de la señal y sincronizar el receptor y la fuente. En todos los sistemas de comunicación es posible que aparezcan errores, la señal se distorsiona de alguna manera antes de alcanzar su destino, por lo que se necesita de una detección y corrección de errores. Cuando cierto recurso se comparte por más dispositivos, el sistema fuente deberá indicar la identidad del destino (direccionamiento). En ciertas situaciones en las que el intercambio de información se vea interrumpido por algún fallo, es necesario un proceso de recuperación que permita continuar con el intercambio sin que se vea afectado el contenido de la información.

El formato de mensajes es relacionado al acuerdo que debe existir entre las dos partes respecto a la forma de los datos intercambiados, como por ejemplo, el código binario usado para representar los caracteres. Además, es necesario que el sistema tenga medidas de seguridad o control, ya que el emisor debe asegurarse de que sólo el destino deseado reciba los datos, y que los datos provienen del supuesto emisor.

Para lograr un intercambio de datos entre dos o más puntos se requiere que interactúen los elementos y por así decirlo, hablen el mismo idioma, es decir: qué se comunica, cómo se comunican y cuándo se comunican; para esto debe seguir una serie de convenciones mutuamente aceptadas por las aplicaciones. Este conjunto de convenciones se denomina protocolo, o sea un conjunto de reglas que gobiernan el intercambio de datos.

## **1.2 ORIENTACIÓN DE LOS PROTOCOLOS**

Los protocolos pueden ser divididos en tres categorías, de acuerdo con las técnicas de envío de mensajes usadas. Éstas son orientadas a carácter, orientado a conteo de byte y orientado a bit.

### **1.2.1 PROTOCOLO ORIENTADO A CARÁCTER**

Está orientado a caracteres especiales para indicar comienzo del mensaje y para indicar el fin del bloque. El protocolo orientado a carácter básico es el IBM BISYNC.

### **1.2.2 PROTOCOLO ORIENTADO A CONTEO DE BYTE**

Usan un encabezado, el cual incluye un carácter especial de inicio seguido de un conteo que indica cuáles son los caracteres que siguen en la porción de datos que vienen y alguna información de control; un ejemplo de este tipo es el DDCMP.

### **1.2.3 PROTOCOLO ORIENTADO A BIT**

Estos protocolos están orientados a bit en cuanto a organización y formato, y se les conoce de varias maneras, tales como: HDLC, ADCCP y ofrecen un alto nivel de flexibilidad, adaptabilidad, confiabilidad y eficiencia de operación. Los protocolos orientados a bit difieren muy poco entre si.

## **1.3 FUNCIONES DE LOS PROTOCOLOS**

### **1.3.1 CONTROL DE TRANSFERENCIA DE DATOS**

Estructura el software que controla la interconexión y su organización, se orienta a niveles, con esto cada nivel inferior ofrece ciertos servicios a los niveles superiores y oculta la implantación de estos servicios. En realidad, un nivel que tiene que transmitir un mensaje a un destino puede agregar encabezados para identificar el contenido y el destino, otro nivel puede imponer el tamaño del mensaje y dividirlo en paquetes y de esta manera lograr que las entidades relacionadas se comuniquen en niveles. Esta jerarquía de protocolo tiene reglas y convenciones que controlan la interconexión, o sea un protocolo de capas. El servicio que se provee a una capa superior puede ser orientado a conexión o servicio sin conexión; en el primero, la conexión impone un orden en el cual son enviados los mensajes, el segundo, el camino que toma cada uno es independiente.

### **1.3.2 REVISION DE ERROR Y RECUPERACIÓN**

Se transmite información redundante que permita deducir que debería ser un mensaje transmitido y transmitir solamente la información suficiente para detectar un error.

### **1.3.3 UTILIZACIÓN DE LÍNEA**

La línea de transmisión o canal de transmisión puede tener capacidad de transmitir información en un sólo sentido (de forma simplex), también podría transmitir en ambas direcciones, pero sólo una cada vez (half-duplex), o puede transmitir en ambas direcciones al mismo tiempo (full-duplex), utilizando así el canal de comunicación de las tres que existen.

### **1.3.4 SINCRONIZACIÓN**

La transferencia síncrona permite al transmisor enviar un bloque entero de caracteres cada cierto intervalo de tiempo, exista o no información en ese instante. En la asíncrona, el transmisor envía un carácter en el momento en el que éste está listo, es un auto mensaje, pues cada carácter transmitido posee información que indica al receptor cuándo un carácter comienza o cuándo termina.

### **1.3.5 FACILIDAD DE TRANSPARENCIA DE COMUNICACIÓN**

Es muy provechoso que si el mismo protocolo pudiera ser usado, no importando si el canal de comunicación fuera asíncrono o serie sincrónica, o paralelo, manejando así la transparencia de datos que es un factor que más afecta la facilidad de transparencia.

## **1.4 CAPAS DE COMUNICACIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS**

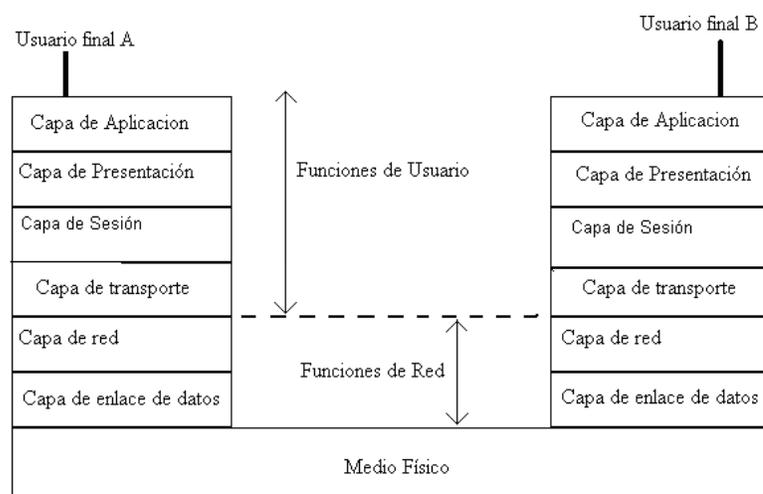
Los dispositivos de comunicación tales como unidades terminales, dispositivos electrónicos y computadoras se conectan a redes, y los datos a intercambiar se transfieren por la red de un dispositivo a otro. Por lo tanto, la transferencia de datos implica organizar la tarea en capas independientes. En la capa de acceso a la red, está relacionada con el intercambio de datos entre el dispositivo y la red a la que está conectado, aquí el software dependerá de la red que se use basado en los estándares, de manera que el resto del software de comunicaciones que esté por encima de la capa de acceso a la red no tendrá que ocuparse de las características específicas de la red que se use. La capa de transporte realiza un intercambio de datos de manera segura, o sea que lleguen a la aplicación destino y además que lleguen en el mismo orden en que fueron enviados. Estos procesos deben proporcionar dicha seguridad y son

independientes de la naturaleza de las aplicaciones. Finalmente, la capa de aplicación contiene la lógica necesaria para admitir las aplicaciones de usuario.

Cada programa de aplicación y cada dispositivo de comunicación pueden requerir diferentes métodos de acceso y diferentes protocolos de comunicación, la intención de una comunicación por capas es que los protocolos se desarrollen de forma tal que realicen las funciones de cada una de las capas. La arquitectura de sistemas de red (SNA: System Network Architecture) de IBM, fue una de las primeras comunicaciones por capas que se desarrollaron.

El modelo de referencia OSI (Open System Interconnection) se desarrolló por la Organización Internacional de Estandarización (ISO: internacional Standard Organization), que rápidamente se ha convertido en un estándar internacional para la comunicación por capas.

Figura 2. Modelo de referencia OSI



La capa o medio, físico está relacionada con la transmisión de un flujo de bits sobre el medio físico; involucra parámetros tales como niveles de voltaje de señales eléctricas y la duración de cada bit en el medio; también con las características mecánicas, eléctricas y de procedimientos para establecer, mantener y desactivar el enlace físico (RS-232C, V.35, etc.).

La capa de enlace, provee transferencia confiable de datos a través del enlace físico, envía bloques de datos (tramas) con la sincronización necesaria, control de errores y control de flujo (HDLC, BISYNC, etc.).

La capa de red, permite transferir los datos en forma transparente, seleccionando una ruta a través de la red. Ésta es responsable de establecer, mantener y terminar las conexiones (X.25, Frame-Relay, IP).

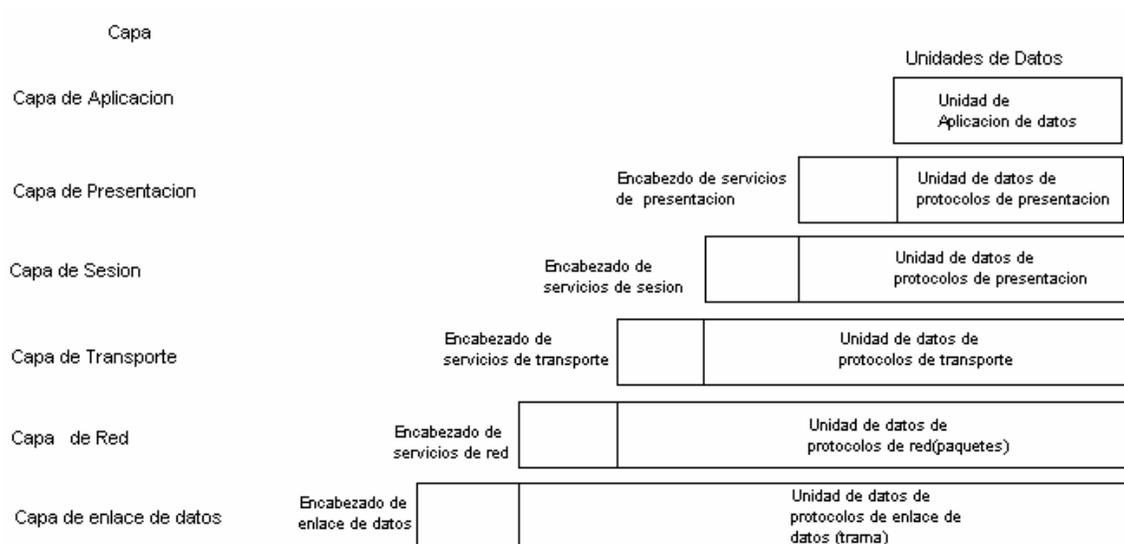
La capa de transporte, asegura transferencia confiable y transparente de datos entre puntos terminales, provee recuperación de errores y control de flujo entre extremos terminales, optimiza la utilización de los recursos empleados, desligando a los usuarios de los detalles de la transferencia misma.

La capa de sesión, es responsable del control de la comunicación entre procesos de aplicación, establece, maneja y termina las conexiones (sesiones) entre procesos cooperantes.

La capa de presentación, ejecuta transformaciones de datos, de tal forma de presentar una interfaz estándar y finalmente, la capa de aplicación, en donde los datos son manejados por el usuario, por lo que éstos ya son denominados datos de usuario.

Cada capa recibe un bloque de datos de la capa superior y añade información de control.

Figura 3. Unidades de datos que intervienen en la formación de una red OSI



## 1.5 DESARROLLO DE DNP

DNP (Distributed Network Protocol) fue desarrollado en 1993 buscando definir las especificaciones como también los acreedores de las mismas y se convirtieron en el grupo de usuarios DNP3 (Distributed Network Protocol Version 3 users group), un grupo compuesto de empresas de servicio público y proveedores que utilizan el protocolo, y que se ha convertido en un protocolo abierto y público para asegurar la interoperabilidad y ser lo más cercano posible a los estándares.

### 1.5.1 ELEMENTOS DE UNA RED DISTRIBUIDA

Para que dos aplicaciones se comuniquen seguirá el conjunto de convenios o protocolos para el intercambio de datos. En términos generales, los elementos que están involucrados en la comunicación incluyen las aplicaciones de usuario, computadoras que se consideran como estaciones maestras, estaciones esclavas y servidores, unidades de terminales remotas (RTUs: Remote Terminal Units), dispositivos electrónicos

inteligentes (IEDs: Intelligent Electronic Devices), medidores y redes. El protocolo de red distribuida (DNP) define estos elementos que usan el enlace de datos para transferir mensajes entre estaciones primarias (origen) y estaciones secundarias (destino), además que para este protocolo, los elementos pueden ser combinados (tanto primarias como secundarias). Una estación primaria controla el enlace de datos; por lo que esta estación transmite tramas de comando a las estaciones secundarias de las cuales recibe tramas de respuesta. Si el enlace es multipunto, la estación primaria es la responsable de mantener sesiones independientes con cada estación. La estación secundaria actúa como una esclava de la estación primaria, responde a los comandos que llegan desde la estación primaria, y mantiene una sesión con la primaria y no tiene responsabilidad del control de enlace.

Entonces los dispositivos (tales como computadoras, RTU's, IED's y concentradores de datos) se conectan a redes, y los datos a intercambiar se transfieren por la red de un dispositivo a otro. Para lograrlo, DNP3 se basa en estándares usados, que resultan en un protocolo que consta de capas compuestas de funciones de red para lograr el control de enlace de datos y establece un esquema cliente-servidor para realizar entre los elementos involucrados, comandos y requerimientos desde las aplicaciones.

### **1.5.2 ESTÁNDARES USADOS COMO BASE PARA DNP3**

Los estándares IEC 870-5-1 y IEC 870-5-2 puestos por la Comisión Electrotécnica Internacional (IEC: International Electrothechnical Commision), el comité técnico No. 57 (Technical Committee) para transmisión de datos en sistemas de telecontrol, son una base para el desarrollo del DNP3, primordialmente de la capa de enlace de datos. El modelo ISO -OSI es soportado por este protocolo y especifica las capas físicas, de enlace de datos y de aplicación.

## **1.6 CAPAS DE DNP3**

### **1.6.1 RELACIÓN MAESTRO -ESCLAVO**

El esclavo es un dispositivo que tiene datos o información que alguien más quiere, el maestro es un dispositivo o programa que requiere información del esclavo. La relación maestro-esclavo comprende dos capas de software: La capa de usuario de alto nivel, y el software del protocolo, que se sitúa en un nivel bajo para requerimientos de transmisión o respuestas. El maestro usa los valores en su base de datos para propósitos específicos. El objetivo del maestro es mantener su base de datos actualizada. Esto lo hace acompañado por requerimientos de envío al esclavo, solicitándole que devuelva la información de la base de datos del mismo (polling). El esclavo responde al requerimiento del maestro, transmitiendo el contenido de su base de datos.

### **1.6.2 SOFTWARE DNP3**

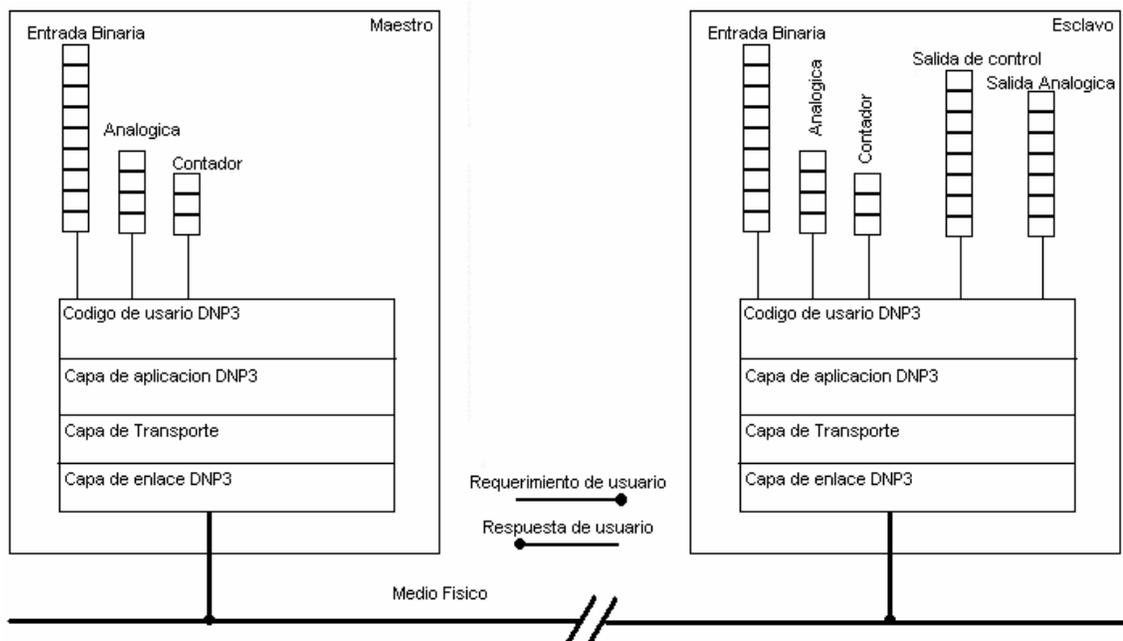
El software está compuesto por capas, las cuales se distinguen como: la capa de enlace con la función de enviar y recibir paquetes que son llamados tramas. La capa de transporte fragmenta mensajes largos en tramas de menor tamaño para que la capa de enlace lo transmita, o cuando en el receptor rearme las tramas en largos mensajes. La capa de aplicación trabaja junto con la capa de transporte y está desarrollada para aplicaciones en sistemas de SCADA (Supervisoy Control And Data Acquisition) y DA (Distribution Automation).

Los elementos en una red distribuida, como las computadoras en estaciones remotas y computadoras en estaciones maestras, se comunicarán e intercambiarán datos y comandos de control, y estos elementos distribuidos compartirán entonces información que es reunida para las estaciones maestras y serían:

- \* Datos de entrada binarios, que son utilizados para monitorear dispositivos de dos estados.
- \* Datos de entrada analógicos, que son convertidos en voltajes, corrientes, niveles de reservas de agua, temperatura.
- \* Datos de contadores de entrada para reportar el consumo de energía.
- \* Archivos que contengan datos de configuración, intercambio de datos de sincronización, fecha y archivos de datos en historiales.

El software DNP3 está organizado en capas para tener una comunicación confiable. Además, la relación maestro-esclavo da una vista simplificada de las bases de datos y los procesos de software involucrados.

Figura 4. Capas de software DNP



En la figura 4, una serie de bloques en la parte superior del maestro representan su base de datos y dispositivos de salida. Los diferentes tipos de datos son organizados como arreglos. Un arreglo de valores de entradas binarias representa estados de dispositivos booleanos físicos o lógicos. Los valores en el arreglo de entrada analógico representan cantidades de entrada que el esclavo ha medido o calculado.

Un arreglo de contadores representa valores de conteo, como son los kilowatt-hora, éstos siempre están incrementándose (aunque éstos alcancen un máximo y entonces regresa a cero y empieza a contar de nuevo). Las salidas de control son organizadas dentro de un arreglo representando físicamente o lógicamente puntos de encendido-apagado, aumento-disminución, disparo-cierre.

El arreglo de las salidas análogas representa cantidades analógicas físicas o lógicas, como aquellas usadas para referencias. El maestro también tiene una base de datos similar para los tipos de entrada de datos (binario, analógico y contadores).

## **2. CAPA DE ENLACE DE DATOS**

### **2.1 CAPA DE ENLACE DE DATOS DNP3**

El principal propósito de la capa de enlace de datos DNP es doble: primero, la capa de enlace de datos debe proveer la transferencia de información o unidad de datos de servicios de enlace (LSDU: Link Service Data Unit), a través de la capa física del estándar ISO-OSI. Esto significa que los datos de usuario proporcionados por capas superiores (LSDU) deben ser convertidos en una trama o unidad de datos de protocolo de enlace (LPDU: Link Protocol Data Unit) y enviadas a la capa física para la transmisión. Inversamente, los LPDUs individuales recibidos por la capa de enlace de datos deben rearmarlos en un LSDU y pasarlo a capas superiores, además esta capa provee sincronización de trama y control de enlace. Segundo, el enlace de datos provee indicaciones de otros eventos tales como estado del enlace.

Una red física real es transparente a la aplicación usando el enlace de datos, porque la capa de enlace de datos es la responsable de conectar y desconectar, de cualquier red física, sin la interacción de capas superiores, (por ejemplo: la capa de aplicación). Es decir que el enlace de datos (dada una dirección de una estación destino) podrá conectar el circuito físico correcto sin el control proporcionado por capas superiores. De esta manera, el medio físico es totalmente transparente al enlace de la capa de servicios de usuario, porque prácticamente el modelo de referencia OSI soporta ambos sistemas como el orientado a conexión y sin conexión. La implementación de DNP3 de la capa de enlace de la IEC maneja ambos sistemas.

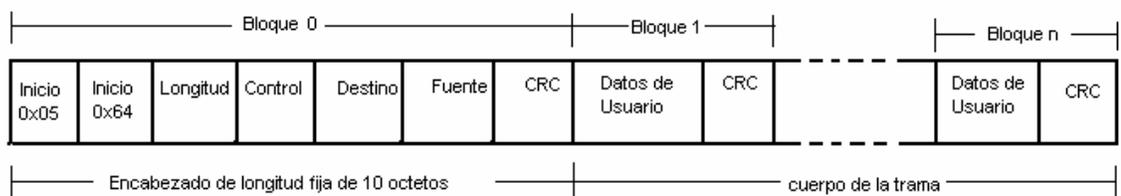
## 2.2 FORMATO DE TRAMA

La capa de enlace envía y recibe paquetes que son llamados tramas. Una trama DNP3 consiste de un encabezado y sección de datos. El encabezado especifica el tamaño de la trama, qué estación debería recibir la trama, qué dispositivo envía la trama e información de control de enlace de datos. La sección de datos es comúnmente la carga útil y contiene los datos que han sido pasados de capas superiores.

### 2.2.1 FORMATO DE TRAMA FT3

Un formato FT3 es definido como un bloque de encabezado de longitud fija seguido por bloques de datos opcionales. Cada bloque tiene un chequeo de redundancia cíclica (CRC: Cyclic Redundancy Check) de 16 bits anexado a él. Los campos de encabezado consisten de dos octetos de inicio, un octeto de longitud, un octeto de control, una dirección de destino y un campo de longitud fija opcional para datos de usuario, especificado por la IEC. Para esta implementación el campo opcional está definido como una dirección fuente.

Figura 5. Formato FT3



## 2.3 CAMPOS DE ENCABEZADO DE TRAMA

### 2.3.1 INICIO

El campo de inicio es en longitud, de dos octetos. El primer octeto es 05 hexadecimal y el segundo es 64 hexadecimal.

### 2.3.2 LONGITUD

El campo de longitud es un octeto en longitud y especifica la cantidad de octetos de usuario en la trama. El tamaño de los campos de CONTROL, DESTINO Y FUENTE están incluidos en esta cantidad. El valor mínimo de este campo es cinco y el valor máximo es 255.

### 2.3.3 CONTROL

El campo de control contiene el sentido de la trama, tipo de trama y el control de flujo de información.

Figura 6. Octeto de control



Si la estación A es definida como la estación maestra designada, y la estación B no es una estación maestra, entonces la estación primaria es la generadora del mensaje. La estación secundaria es la estación destino, de lo anterior podemos describir cada bit de control:

DIR (Dirección física de la transmisión): El bit de dirección indica la dirección física de la trama con relación a la estación maestra designada.

1= estación A hacia la estación B

0= estación B hacia la estación A

PRM (Mensaje primario): El bit de mensaje primario indica la dirección de la trama en relación a la estación iniciante.

1= trama desde la primaria (estación iniciante).

0= trama desde la secundaria (estación respondiente).

FCB (bit de conteo de trama): El bit de conteo de trama es usado para supresión de tramas perdidas y duplicadas de la misma estación secundaria. Este bit se dispara (toggled) para cada servicio de envío-confirmación completado, que es iniciado por la misma estación primaria y dirigido a la misma estación secundaria. Inicialmente, antes de comunicarse con la estación secundaria o después de una falla de comunicación, la estación primaria (en ambas estación maestra y remota) debe reajustar el enlace de datos por cada enlace de datos secundario con que desee comunicarse. Cada estación secundaria, después de un inicio de enlace o falla de transacción, no debe aceptar cualquier mensaje de envío-confirmación con el FCV puesto, sin que un comando de reajuste haya sido recibido o un mensaje de confirmación enviado.

FCV (bit de conteo de trama válido): El bit de conteo de trama válido habilita el funcionamiento del bit FCB.

1= el bit de conteo de trama es válido, indica a una estación secundaria que el estado de el FCB debe ser revisado (check) contra el estado de el FCB de la última trama enviada con FCV puesto.

0= el bit de conteo de bit es ignorado.

DFC (bit de control de flujo de datos): Es usado para prevenir el sobre-flujo de buffers en una estación secundaria. La estación secundaria retorna este bit puesto a uno si un

envío (SEND) de datos de usuario a la estación secundaria podría causar a los buffers del enlace de datos, un sobre-flujo. La estación primaria debe interrogar a la estación secundaria, usando un requerimiento con respuesta (REQUEST-RESPOND) de estado del enlace (LINK STATUS) hasta que el DFC es retornado con el valor de cero. En este punto, la estación primaria puede continuar con el envío de datos de usuario.

RES (reservado = 0).

**CÓDIGO DE FUNCIÓN:** Define el tipo de trama, y como el enlace de datos manejará la trama. La definición de los valores ubicados en este campo difiere entre estaciones primaria y secundaria. Las siguientes tablas definen los códigos implementados y los estados FCV asociados.

Tabla I. Valores de campo de códigos de función del octeto de control, enviado desde la estación primaria (PRM=1)

Código de función	Tipo de trama	Función de servicio	bit FCV
0	Se espera SEND-CONFIRM	RESET del enlace remoto	0
1	Se espera SEND-CONFIRM	Reset del proceso de usuario	0
2	Se espera SEND-CONFIRM	Función de TEST para el enlace	1
3	Se espera SEND-CONFIRM	Datos de usuario	1
4	Se espera SEND-CONFIRM	Datos de usuario sin confirmación	0
5,6,7,8		No usados	
9	Se espera REQUEST-REPLY	Requerimiento de estado de enlace	0
10,11,12,13,14,15		No usados	

Tabla II. Valores de campo de códigos de función del octeto de control, enviado desde la estación secundaria (PRM=0)

Código de función	Tipo de trama	Función de servicio
0	Confirmación	ACK , conocimiento positivo
1	Confirmación	NACK, mensaje no aceptado enlace ocupado
2,3,4,5,6,7,8,9,10		No usado
11	Respuesta	Estado de enlace DFC=0, o DFC=1
12,13		No usado
14		Servicio de enlace sin funcionamiento
15		Servicio de enlace no usado o implementado

### 2.3.4 DIRECCIÓN DESTINO

El campo de dirección-destino es de dos octetos en tamaño, y especifica la dirección de la estación a donde es dirigida la trama. El primer octeto de la dirección es el octeto de bajo orden y el segundo octeto es el de alto orden. Además la dirección 0xFFFF es definida como una dirección para todas las estaciones. Todas las estaciones aceptarán las tramas con la dirección-destino puesto a este valor.

### 2.3.5 DIRECCIÓN FUENTE

El campo de dirección-fuente es dos octetos en tamaño y especifica la dirección de la estación de la trama originada. El primer octeto de la dirección es el octeto de bajo orden y el segundo octeto es el alto orden.

Figura 7. Formato de la dirección destino y fuente.



## 2.4 DATOS DE USUARIO

Los bloques siguientes al encabezado pueden contener desde uno a 16 octetos de datos de usuario. Cada bloque de datos de usuario tiene redundancia (CRC-16 bit) añadido a él. Cuando más de 16 bloques de datos de usuario siguen el encabezado (bloque cero), cada bloque debe contener 16 octetos de datos, excepto para el último que contendrá el sobrante.

## 2.5 CAMPOS DE REVISIÓN DE REDUNDANCIA CÍCLICA

En los sistemas de redes, un rol importante de la capa de enlace de datos es convertir el enlace físico entre dos entidades confiable, esto se logra incluyendo información redundante en cada trama transmitida, se incluye solamente la suficiente redundancia para hacer posible detectar errores y entonces disponer la retransmisión de las tramas dañadas. La revisión de redundancia cíclica es ampliamente usada en un esquema de detección de errores, basada en el bit de paridad en las aplicaciones de transmisiones seriales. Los bits de los datos a ser transmitidos son los coeficientes del polinomio. Como ejemplo: la cadena de bits 1101011011 tiene 10 bits, representando un polinomio de 10 términos, entonces tenemos que:

$$\begin{aligned}M(x) &= 1 \cdot X^9 + 1 \cdot X^8 + 0 \cdot X^7 + 1 \cdot X^6 + 0 \cdot X^5 + 1 \cdot X^4 + 1 \cdot X^3 + 0 \cdot X^2 + 1 \cdot X^1 + 1 \cdot X^0 \\M(x) &= X^9 + X^8 + X^6 + X^4 + X^3 + X + 1\end{aligned}\quad (\text{ecuación 1})$$

Para calcular el CRC de un mensaje, otro polinomio llamado polinomio generador  $G(x)$  es escogido. Dos octetos de chequeo de redundancia cíclica son añadidos a cada bloque en una trama. Los campos de inicio, longitud, control, destino y fuente son todos incluidos cuando se calculan el CRC para el encabezado.

Los dos octetos de CRC son generados del siguiente polinomio y son invertidos antes de colocarlos en el bloque de transmisión.

$$G(x) = X^{16} + X^{13} + X^{12} + X^{11} + X^{10} + X^8 + X^6 + X^5 + X^2 + 1 \quad (\text{ecuación 2})$$

El algoritmo usado se describe a continuación: Se asume un módulo de dos aritméticas (suma y división). Un bloque de mensaje (M) de k-bits es transmitido junto con otros bloques (k es 64 para el encabezado, 128 para los bloques de datos de usuario, para el último bloque k es de ocho a 128).

Una palabra de check (F) de CRC 16-bits es invertida a nivel bit ( $F'$ ) y añadido a M. Juntos M y  $F'$  son sumados así  $T' = 2^{16}M + F'$  y  $T'$  será transmitido, además se define que  $T = 2^{16}M + F$ . La secuencia de check (CRC) es un patrón (P) de 17 bits definido por la forma polinomial. El algoritmo de CRC requiere que cuando T es dividido por P en el receptor el residuo sea cero.

Si el residuo no es cero entonces el bloque está errado. Además, el residuo (R) de  $2^{16}M/P$  es usado como F en el bloque o sea que:  $2^{16}M/P = Q + R/P$  (ecuación 3). Q es el cociente. Esto se puede demostrar, el residuo es cero, si asumimos que  $T=2^{16}M+R$  entonces  $T/P=(2^{16}M+R)/P$ . Si sustituimos en la ecuación tres entonces  $T/P = Q + (R + R)/P = Q$ , la suma de R a sí mismo da como resultado cero.

A continuación se describe el procedimiento de transmisión y recepción:

Para transmitir un bloque:

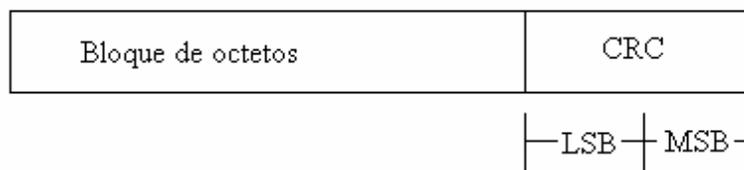
1. Tener el bloque de datos de usuario  $M$  con  $k$  bits de datos.
2. Multiplicar  $M$  por  $2^{16}$  para obtener  $2^{16}M$ .
3. Dividir este número por  $P$  (17 bits) para obtener  $R$  (16 bits).
4. Invertir  $R$  a nivel de bit para obtener  $R'$ .
5. Agregar  $R'$  a  $2^{16}M$  y transmitir como un bloque ( $T'$ ).

Procedimiento para recibir el bloque:

1. Recibir el bloque ( $T'$ ) ( $k + 16$  bits).
2. Invertir  $R'$  (16 bits) en  $T'$  ( $k + 16$  bits) y obtener  $T$  ( $k + 16$  bits).
3. Dividir  $T$  por  $P$  (17 bits) para obtener el residuo.
4. Si el residuo no es cero, entonces hay un error en el bloque, si no el bloque es correcto.

Usando la clase de formato de trama FT3 y CRC, la trama tiene una distancia Hamming de seis. El siguiente diagrama muestra el orden de los 16 bit de CRC de la palabra de check con respecto a cualquier bloque (datos de usuario o encabezado).

Figura 8. Orden de CRC



## **2.6 CÓDIGOS DE FUNCIONES DE ENLACE DE DATOS**

### **2.6.1 REAJUSTE**

Este código de función es usado para sincronizar una estación primaria y una secundaria, aún más que las transacciones de tramas envío-confirmación (SEND-CONFIRM). Hasta una recepción y una respuesta a un comando de RESET, la estación secundaria se iniciará aceptando los mensajes primarios de esa estación primaria con el bit FCB puesto. El comando reajuste (RESET) sólo habilita la comunicación en una dirección, es decir, desde la estación primaria a la secundaria. Esto es porque una transacción completa sólo garantiza que el transmisor de la estación primaria y el receptor de la estación secundaria están comunicándose. La estación primaria debe enviar este código de función cuando lo desee, para comunicarse primero con la estación secundaria o después que una falla de comunicación ha sido reconocida por la estación primaria. Cuando una estación secundaria tiene un reinicio o cuando se ha reconocido una falla de comunicación por la secundaria, la estación secundaria se considerará sin RESET, en este caso, la estación secundaria no aceptará mensajes de la estación primaria hasta que ésta haya recibido una respuesta a el comando de RESET de la estación primaria.

El comando RESET también sincroniza el bit FCB entre las estaciones primaria y secundaria. La estación secundaria, después de completar la transacción de RESET, pondrá el bit FCB a uno en el siguiente mensaje (con FCB válido) para la estación secundaria.

#### **2.6.1.1 Transacción primaria**

- Se hace un número configurable de intentos.
- Enviar una trama RESET con FCB=0, FCB=x, PRM=1, DIR=x.

- Esperará el período predeterminado de una trama ACK desde la estación secundaria.
- Si la trama ACK es recibida, el estado del FCB es uno (ejemplo: La siguiente trama enviada a la secundaria con FCV válido debería tener FCB=1) y salir del loop.
- Si la trama no es recibida entonces regresar al inicio del loop y reintentarlo.
- Final del loop.
  
- Si un ACK fue recibido, entonces la transacción es considerada completa y la estación secundaria puede considerarse en línea (o activa). Una indicación (IND: Data Link Indications) positiva puede ser retornada al enlace de datos de usuario. Por el contrario, la estación secundaria podría considerarse fuera de línea y una indicación (IND) negativa debería ser enviada al enlace de datos de usuario.

### **2.6.1.2 Transacción secundaria**

Después de un inicio o después de una falla, entonces:

- Se espera la recepción de una comando de RESET con FCV=0, FCB=x, PRM=1, DIR=x.
- Se responde con una trama de confirmación ACK (DFC=x, PRM=0, DIR=x). El estado de FCB (valor esperado de FCB en la siguiente trama recibida con FCV válido), debería ser puesto a uno. Una IND positiva puede ser enviada al enlace de datos de usuario.
- Durante la operación normal, si un comando RESET con FCV=0, FCB=x, PRM=1 y DIR=x es recibido, entonces la transacción actual (si hay) puede ser abortada (posiblemente con una IND negativa enviada al enlace de datos de usuario).
- En tal caso, se responde con una trama de confirmación ACK (DFC=x, PRM=0, DIR=x). El estado FCB (valor esperado de FCB en la siguiente trama recibida

con FCV válido) debería ser puesto a uno. Una IND positiva puede ser enviada al enlace de datos de usuario.

### **2.6.2 REAJUSTE PARA PROCESOS DE USUARIO**

Este código de función es usado para un reajuste (RESET) del proceso de enlace de datos de usuario. Sobre la recepción, por una estación secundaria, una IND debería ser enviada al enlace de datos de usuario. El enlace de datos de usuario puede usar esta indicación para reajustar su estado interno. Si es aceptado por el enlace de datos de usuario, una trama de confirmación ACK es enviada en respuesta de lo contrario, una trama de confirmación NACK es enviada en respuesta.

### **2.6.3 PRUEBA**

Un comando de prueba (TEST) es usado para evaluar el estado del enlace de datos de la estación secundaria. Sobre la recepción, por una estación secundaria, este comando revisa el valor del bit FCB en el mensaje primario y lo compara contra el estado de FCB (FCB esperado) por la estación primaria. Si los FCBs no igualan, entonces la estación secundaria debe enviar la última trama secundaria de confirmación. De lo contrario, una trama de confirmación ACK debe ser enviada en respuesta y el estado esperado de FCB debe estar disparado. La estación secundaria también pone el bit DFC de acuerdo a la respuesta.

### **2.6.4 DATOS DE USUARIO**

La función de datos de usuario es usada para enviar datos de confirmación a la estación secundaria. Antes que las comunicaciones inicien, la estación secundaria debe haber tenido un RESET de acuerdo con lo anteriormente expuesto. La trama enviada

contiene los datos de usuario desde el enlace de datos usuario de la estación primaria, que es pasado al enlace de datos de usuario de la estación secundaria.

#### **2.6.4.1 Transacción Primaria**

Se hace el número configurable de intentos:

-Envía una trama de datos de usuario con FCB=1, PRM=1, DIR=x y el FCB puesto al estado del FCB para la estación secundaria (siguiente estado FCB esperado).

-Esperará el período predeterminado para una trama ACK o NACK de la estación secundaria.

-Si la trama es de NACK, entonces se espera una cantidad predeterminada de tiempo hasta que el enlace secundario no esté ocupado o usando el requerimiento de ESTADO DE ENLACE e ir al inicio del loop para reintentar.

-Si una trama ACK correcta es recibida, está disparado el estado de FCB (ejemplo: En la siguiente trama enviada a la secundaria con FCB válido debe tener FCB opuesto) y salir de loop.

-Si la trama no es recibida entonces ir al inicio del loop y reintentar.

-Si una ACK fue recibida, entonces la transacción es considerada completa y la estación secundaria puede ser considerada en línea. Una indicación (IND) positiva puede ser retornada al enlace de datos de usuario.

-Por el contrario, una indicación (IND) negativa debe ser enviada al enlace de datos de usuario y la estación secundaria es considerada fuera de línea o en línea, dependiendo sobre la interpretación de la falla del enlace de datos de usuario.

#### **2.6.4.2 Transacción secundaria**

-Hasta la recepción de una trama de datos de usuario con FCB=1, PRM=1, DIR=x y FCB puesto a el estado de FCB esperado, se ejecuta lo siguiente: Si los datos del

enlace de datos de usuario están listos para aceptar los datos de usuario, entonces responde con una trama de confirmación ACK (DFC=x, PRM=0, DIR=x) si no responde con una trama NACK (el mismo orden de bits como ACK) y salir de este loop.

## **2.6.5 DATOS DE USUARIO SIN CONFIRMACIÓN**

Esta función es usada para enviar datos de usuario a la estación secundaria sin necesidad de confirmación. De esta manera, el ancho de banda del sistema puede ser completamente utilizado si los datos de usuario son de baja prioridad. La trama enviada contiene los datos de usuario del enlace de datos de usuario primario, que es pasado al enlace de datos de usuario de la estación secundaria. A continuación se describen los procedimientos de transmisión:

### **2.6.5.1 Transacción primaria**

Envía una trama de datos de usuario sin confirmación con PRM=1, DIR=x, FCV=0, FCB=x. Anunciando una IND positiva al enlace de datos de usuario.

### **2.6.5.2 Transacción secundaria**

Recibe una trama de datos de usuario sin confirmación como el anterior, y envía una IND positiva con el dato al enlace de datos de usuario.

## **2.6.6 REQUERIMIENTO DE ESTADO DE ENLACE**

Este comando es usado para requerimientos del estado del enlace de datos del secundario. Una estación secundaria responderá a este requerimiento con una trama de confirmación (CONFIRM) del estado de enlace (LINK STATUS), con el bit DFC

puesto a uno si el enlace de datos está ocupado o el enlace de datos de usuario no puede aceptar cualquier dato de usuario, cero está indicando que el enlace de datos no está ocupado y el enlace de datos de usuario puede aceptar más datos de usuario.

## **2.7 SERVICIOS Y RESPONSABILIDADES DEL ENLACE DE DATOS**

A continuación se describen los servicios proporcionados por el enlace de datos y sus funciones.

El enlace de datos es responsable de realizar las siguientes funciones:

- \* Realizar los reintentos de mensajes.
- \* Sincronización y manejo del bit FCB en la palabra de control.
- \* Ajustar y limpiar el bit DFC, basado en la disponibilidad sobre el buffer.
- \* Empaquetar los datos de usuario en el formato de trama y transmitir los datos a la capa física.
- \* Desempaquetar las tramas que son recibidas desde la capa física en datos de usuario.
- \* Controlar todos los aspectos de la capa física.
- \* Realizar procedimientos de anulación/detección de colisiones (Collision avoidance/detection) para asegurar la fiable transferencia de los datos sobre el enlace físico.
- \* Responder a todas las tramas válidas (códigos de función) recibidos desde la capa física.

El enlace de datos es responsable de proveer los siguientes servicios:

- \* Intercambio de las unidades de servicios de datos (SDUs: Service Data Units) entre el par de enlace de datos DNP.
- \* Notificación de error al enlace de datos de usuario.
- \* Secuencia de las SDUs.
- \* Priorizar la entrega de la SDU.

\* Calidad de entrega de la SDU.

La prioridad de entrega puede ser rápida o normal, para indicar una alta o baja prioridad de requerimiento. La calidad de entrega puede ser del tipo envío-sin respuesta o envío-confirmación para indicar si, o no, un reconocimiento de mensaje es requerido. La notificación de error se podrá dar al enlace de datos de usuario, cuando una respuesta a un requerimiento no ha sido recibida.

## **2.8 CAPA FÍSICA DE INTERFAZ**

A continuación se describe el enlace de datos DNP3 hacia una interfaz física. La interfaz física describe los servicios necesarios que cualquier capa física debe proporcionar a manera de acomodarse al enlace de datos DNP3.

### **2.8.1 DESCRIPCIÓN DE LA CAPA FÍSICA**

La capa física que es recomendada para el enlace de datos, está orientada a una capa física serial asíncrona soportando ocho bits de datos, un bit de parada, sin paridad y niveles de voltaje RS-232C y señales de control. El estándar V.24 describe al DTE, el cual es usado para comunicarse con un DCE y es usualmente modulado usando FSK; éste tipo de conexión de circuito, a una red pública (PSN: Public Switching Network) o líneas privadas arrendadas pueden ser usadas. En ambos casos el modem debe cumplir como mínimo con las definiciones del estándar V.24 para DCE.

La capa física debe proveer cinco servicios básicos: enviar, recibir, conectar, desconectar, y el estado. El servicio de “envío” convierte los octetos de datos a bits seriales de datos para la transmisión entre el DTE y DCE. Éste debe proveer el apropiado control de señales a manera de comunicarse con el DCE asignado. El servicio

de “recepción” debe ser capaz de aceptar datos desde el DCE y por consiguiente proveer de la correcta señalización al DCE, a manera de recibir datos y no ruido. Los servicios de “conectar” y “desconectar” proveen la conexión y desconexión de la PSN (si se aplica). El servicio de “estado” debe ser capaz de retornar el estado del medio físico. Como mínimo el servicio debe indicar si el medio está o no ocupado.

Los requerimientos de la capa física pueden ser enviados en cualquier tiempo, después que la capa física ha sido iniciada y configurada con todos los parámetros relevantes.

## **2.8.2 CARACTERÍSTICAS DE LA CAPA FÍSICA**

### **2.8.2.1 Configuraciones de línea**

Indiferentemente de la capa física usada, hay dos topologías que son la directa y bus serial.

La topología directa tiene dos nodos físicos con conexión directa uno con otro. Ésta es mayormente referida como un punto a punto y puede ser un cable físico directo de punto a punto, los dos nodos pueden utilizar radio, modem o conexión dial up a través de una PSN.

La topología de bus serial tiene más de dos nodos físicos con cada nodo conectado al mismo canal o línea de comunicación, con cada otro nodo en la red de bus serial. Ésta es referida también como multi-drop. Así puede haber un nodo ubicado para estar en control con la red física, y este nodo transmite a múltiples nodos y recibe de múltiples nodos. Todos los otros nodos en el bus reciben desde el nodo maestro y transmiten hacia el nodo maestro. El enlace de datos soporta múltiple-maestro, múltiple-esclavo y comunicaciones punto a punto.

En comunicaciones punto a punto, todos los dispositivos actúan como enlaces de datos esclavos y la anulación de colisiones está activa, como ningún dispositivo tiene una alta prioridad y todos pueden transmitir espontáneamente.

En una configuración múltiple-maestro, los dispositivos maestros tienen mayor prioridad que los dispositivos esclavos. Sin embargo, la prioridad está asignada entre los maestros.

### **2.8.3 MODOS DE TRANSMISIÓN**

La capa física soporta transmisión/recepción de datos en modo serial. Generalmente, la unidad de datos transferida será de ocho bits en longitud. La transmisión puede ser asíncrona, síncrona.

### **2.8.4 CAPA FÍSICA**

El propósito del enlace de datos para la interfaz de la capa física, es permitir que el enlace de datos envíe o reciba un mensaje hacia o desde otro enlace de datos. Para cumplir con esto, el enlace de datos debe ser capaz de controlar cuando una transmisión de datos toma lugar, detectar la presencia de datos en el circuito físico de comunicación y usar el control de señalización de línea para el control del circuito físico. Además, la estación maestra (o dispositivo de mayor prioridad), necesita estar apto para tomar el control del circuito de comunicación y bloquear la transmisión de otras estaciones.

En una conexión de tipo directa, la estación primaria (estación iniciante) puede sólo comunicarse con una estación. Si este circuito es de cuatro hilos (alambre de cobre), entonces se usará un procedimiento full-duplex (transmisión y recepción) y no habrá colisión de mensajes en el circuito. Sin embargo, si el circuito es de dos alambres,

entonces un procedimiento half-duplex (transmisión o recepción) será usado. En este caso, una colisión puede ocurrir si ambas estaciones intentan transmitir datos al mismo tiempo. Una conexión directa o dial up PSN es típicamente de dos alambres, pero el circuito desde una estación al modem es de cuatro alambres (circuito full-duplex) y debe ser usado en un criterio full-duplex. El modem dial up debe usar señalización, o sea, CTS para mantener inactivo el transmisor después de un acertado RTS.

En una topología multi-drop, la estación maestra designada puede actuar como estación primaria para varias estaciones secundarias. En este caso, sí es posible la colisión en circuitos de dos y cuatro alambres.

En un circuito de dos alambres, la estación maestra designada puede chocar con el mensaje de cualquier otra estación, y la estación esclava puede chocar con otra al mismo tiempo.

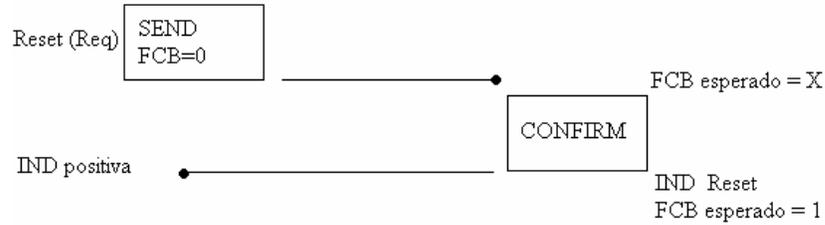
En un circuito de cuatro alambres, los mensajes de la estación maestra no pueden chocar con los mensajes de la estación esclava, pero los mensajes de la estación esclava pueden chocar una con otra.

## **2.9 PROCEDIMIENTOS DE TRANSMISIÓN**

### **2.9.1 REAJUSTE DE ENLACE SECUNDARIO**

Se designa una estación primaria que envía una trama SEND-CONFIRM RESET a la estación secundaria. La estación secundaria recibe el mensaje y responde con una trama de confirmación ACK.

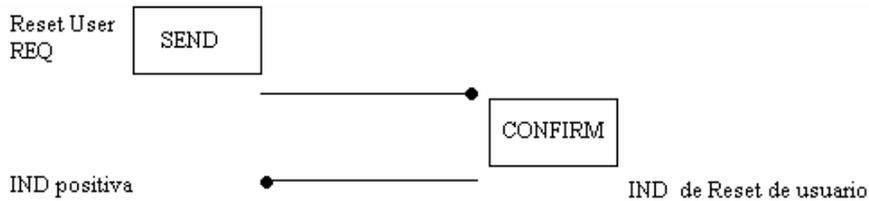
Figura 9. Reajuste de enlace secundario.



### 2.9.2 REAJUSTE DE PROCESOS DE USUARIO

Una estación primaria envía una trama de RESET de procesos de usuario con SEND-CONFIRM a la estación secundaria, la estación secundaria recibe el mensaje y responde con una trama de confirmación ACK.

Figura 10. Reajuste de procesos de usuario

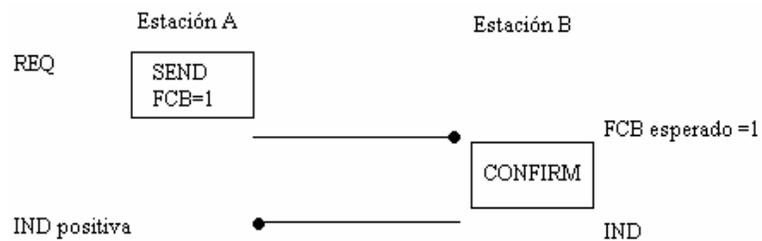


### 2.9.3 ENVÍO-CONFIRMACIÓN DE DATOS DE USUARIO

La estación maestra designada actúa como una estación primaria que envía una trama SEND-CONFIRM a una estación no maestra, que actúa como estación secundaria. En la primera trama FCB es válido después que el enlace secundario fuera reajustado, entonces FCB=1 en la trama SEND. La estación secundaria espera que FCB sea uno (ya que FCB es válido) y envía una trama CONFIRM. La estación maestra, hasta que recibe la confirmación, asume que el mensaje fue correctamente recibido y hace una

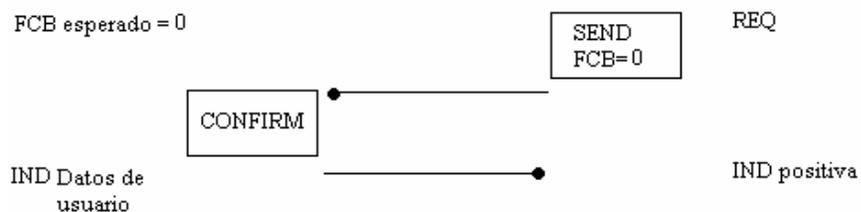
indicación de enlace de datos de usuario (IND) de la estación maestra del suceso, cuya representación está en la figura 11.

Figura11. Envío desde la estación A / Confirmación de la estación B



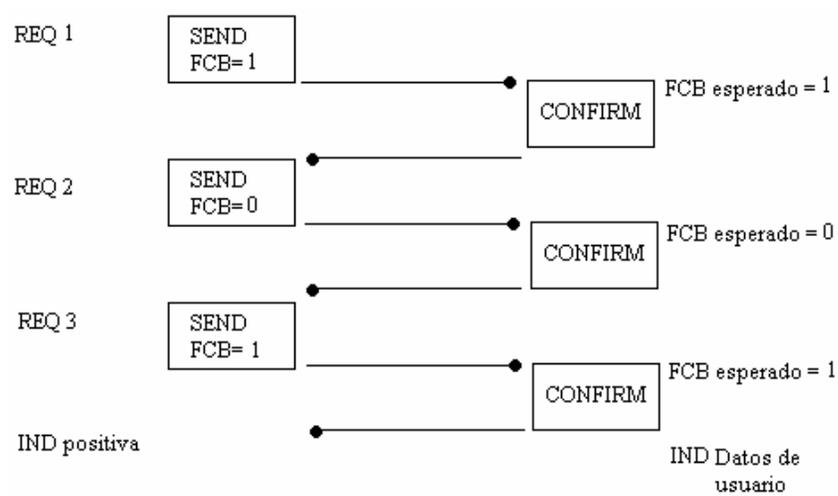
En la figura 12, una estación no maestra actúa como una estación primaria que envía una trama SEND-CONFIRM a una estación no maestra designada, actuando como estación secundaria. La segunda trama, después de que el enlace secundario ha sido reajustado FCB=0 en la trama SEND, la secundaria espera que FCB=0, como es la segunda trama recibida después de que el enlace fue reajustado (RESET). Una trama CONFIRM es enviada en respuesta. La estación no maestra, hasta que recibe la indicación de la confirmación del suceso, hace una indicación al enlace de usuario (IND) de la estación no maestra.

Figura 12. Envío desde la estación B / Confirmación desde la estación A



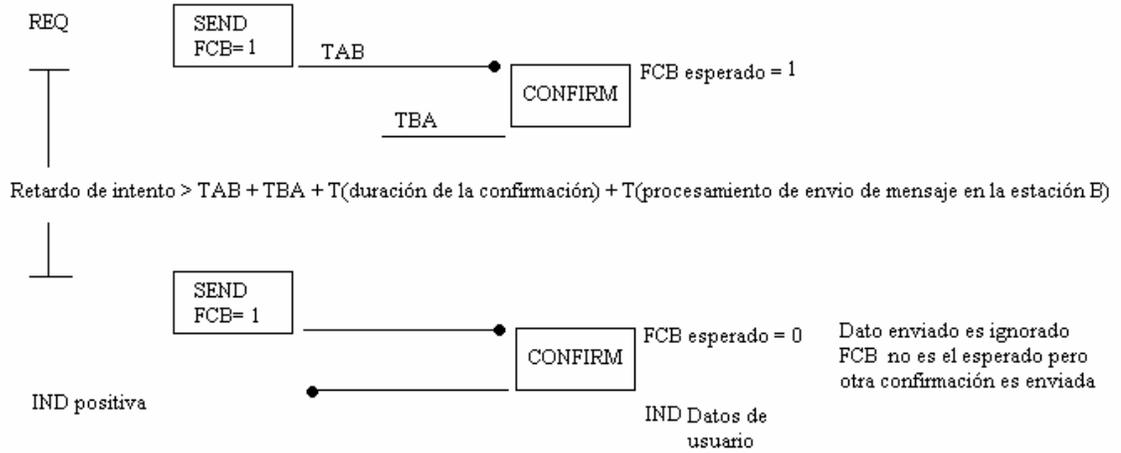
En la figura 13, una estación maestra envía tres tramas consecutivas a la misma estación no maestra.

Figura 13. Múltiples tramas de envío desde la estación A, confirmación desde la estación B



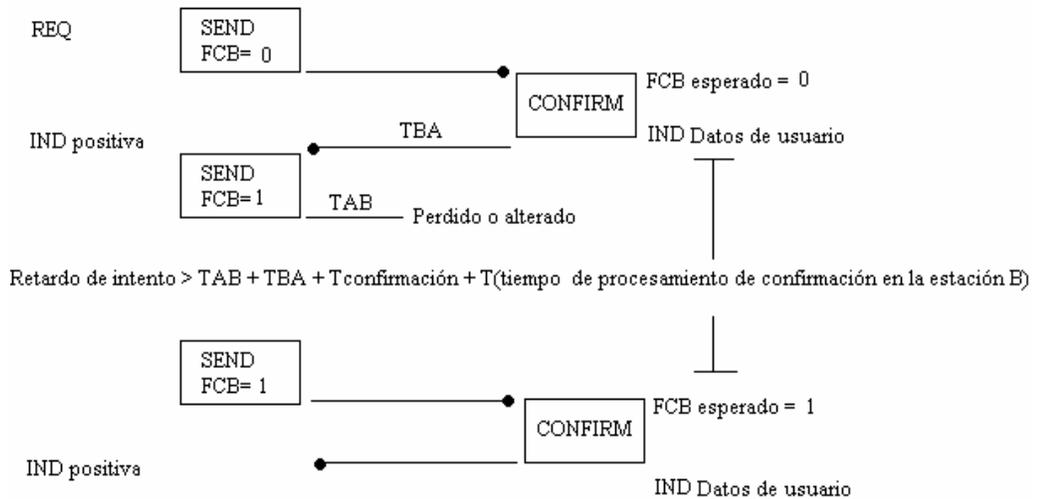
En la Figura 14, la estación maestra designada actúa como primaria, envía un mensaje de trama a la estación secundaria no maestra. Este ejemplo ilustra qué pasa cuando una trama de confirmación de la estación secundaria es perdida o alterada.

Figura 14. Operación del bit de conteo de trama FCB Primaria-Secundaria



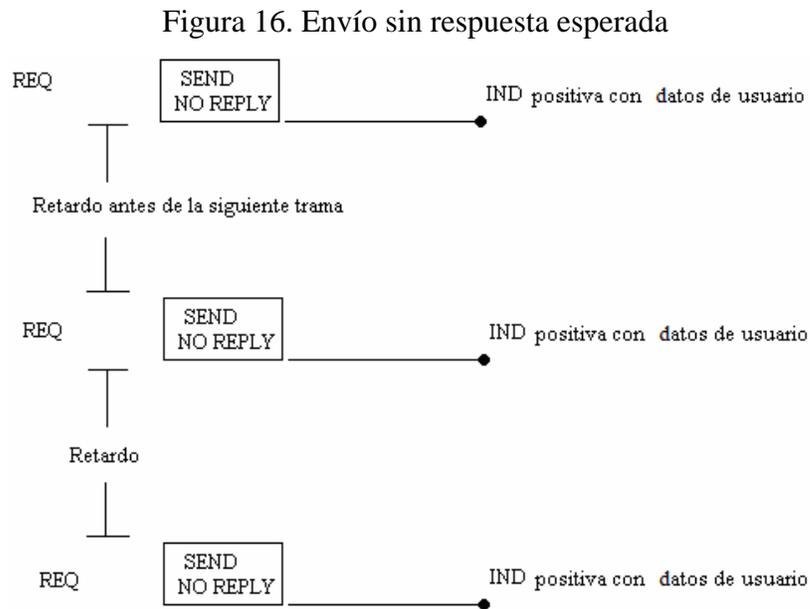
En la figura 15, la estación maestra designada actúa como primaria y envía dos tramas de mensajes a la estación secundaria no maestra. Ambas estaciones, maestra y no maestra, actúan como estaciones primarias y pueden reintentar tramas de envío.

Figura 15. Operación del bit de conteo de trama FCB Primaria-Primaria



## 2.9.4 ENVÍO SIN RESPUESTA ESPERADA

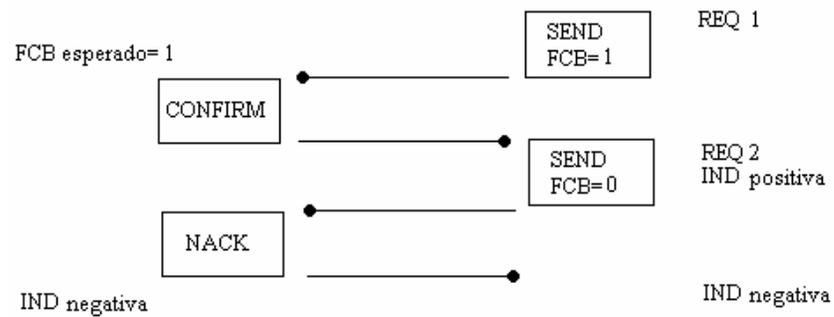
En la figura 16, la estación primaria maestra o no maestra envía tres tramas a la secundaria maestra o no maestra. Con la finalización de transmisión de la trama de envío, la estación primaria hace una indicación al enlace de datos de usuario. La estación secundaria, con la recepción de una trama válida, indica datos disponibles al enlace de datos de usuario.



## 2.9.5 ENVÍO / NACK

En la figura 17, una estación no maestra envía una trama a la maestra secundaria. Con la recepción de la primera confirmación, la primaria indica los sucesos al enlace de datos de usuario. La estación primaria envía una segunda trama a la secundaria. La maestra secundaria decide que no puede aceptar cualquier trama, en este momento envía una trama de retorno NACK. La primaria, después de recibir este NACK, abandona la transacción y envía una indicación negativa al enlace de datos de usuario.

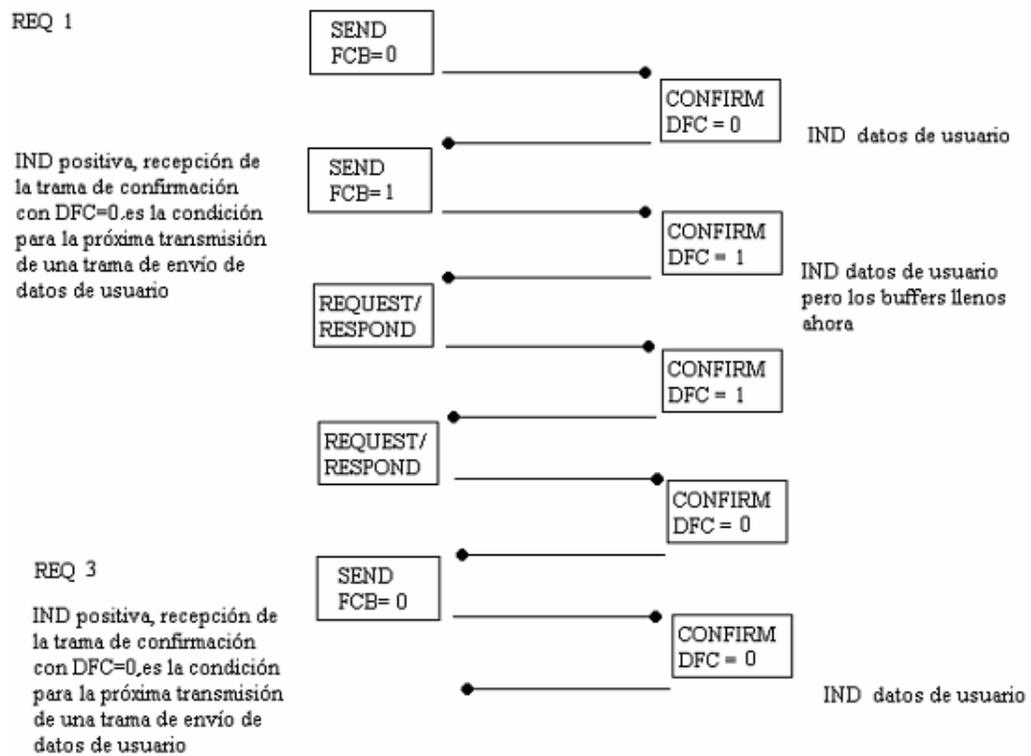
Figura 17. Envío desde la estación B / Una trama NACK desde la estación A



### 2.9.6 REQUERIMIENTO / RESPUESTA

En la figura 18, la estación primaria envía tramas consecutivas a la estación secundaria. Cuando la estación secundaria no puede recibir alguna trama, los mensajes de confirmación contienen el bit DFC puesto. La estación primaria, hasta que tenga la recepción de la confirmación, deja de enviar tramas de datos a la estación secundaria, pero a cambio, periódicamente hará requerimientos del estado de la secundaria por el envío de una trama de requerimiento-respuesta. La secundaria responderá al requerimiento con el estado actual del bit DFC. Si la secundaria está lista para recibir más datos, el DFC retornado será cero, o de lo contrario el DFC retornado será uno. Cuando la estación primaria reconoce que el DFC=0 en la trama de respuesta, la transmisión de tramas de envío continuará.

Figura 18. Trama de Requerimiento / Respuesta y uso del bit DFC



### 3. FUNCIONES DE TRANSPORTE

En este capítulo se describen las funciones de la capa de transporte, la cual actúa como una capa de pseudo transporte a la capa de enlace de datos de DNP. La función de la capa de pseudo transporte es específica, sólo para aquellos mensajes que son más largos que una unidad de datos de enlace de protocolo (LPDU: Link Protocol Data Unit), entre las estaciones primaria y secundaria. Este proceso se describe de la siguiente manera: La capa de pseudo-transporte toma una TSDU (datos de usuario) y lo fragmenta en varias secuencias de TPDUs: Transport Protocol Data Units (cada una con información de control del protocolo de transporte (TPCI: Transport Protocol Control Information)). Cada TPDU es enviada a la capa de enlace de datos como una unidad de datos de servicio de enlace (LSDU) para la transmisión. Esto también trabaja en la forma inversa. La capa de pseudo transporte recibe múltiples TPDUs de la capa de enlace de datos y las ensambla en una TSDU.

Las LSDUs son fragmentos de datos de usuario, las cuales son suficientemente pequeñas para colocarlas en el formato de trama FT3. Cuando una estación primaria transmite un mensaje a una estación secundaria, la función de transporte es partir el mensaje en LSDUs. Esta función se suma al octeto de encabezado de la capa de transporte (TH: Transport Header), en el inicio de los fragmentos de datos de usuario que contienen la información para la estación secundaria para reconstruir el mensaje completo, además todos los mensajes de la capa de pseudo-transporte tienen un encabezado de transporte (TH). La estación secundaria revisa el octeto TH en la recepción de cada LSDU para la correcta secuencia y construye un mensaje TSDU para las capas superiores.

El encabezado de la capa de transporte (TH) contiene información que puede identificar la primera trama, la última trama y dar de cada trama un número de secuencia de seis bits. Esta información es requerida para reconstruir un mensaje y también para proteger contra capas superiores de pérdida de dirección o mensajes incompletos.

### 3.1 ENCABEZADO DE TRANSPORTE

Después que el enlace de datos recibe una trama completa, los datos están presentes para las funciones de transporte en un formato mostrado a continuación. El campo de encabezado de transporte (TH) es suprimido antes que la trama sea combinada con otras tramas pertenecientes al mismo mensaje.

Figura 19. Estructura de un TPDU



Cuando una aplicación requiere la transmisión de un mensaje largo, el mensaje es partido en pequeños fragmentos y lo suficiente para llenar una trama de enlace de datos DNP. El máximo tamaño de un fragmento es de 249 octetos de datos de usuario. El TH es sumado a la cabeza del fragmento y el máximo número de octetos para ser tramados se convierte en 250 octetos.

- \* Cantidad de datos máxima del enlace de dato                    +255    octetos
- \* Cantidad de datos del encabezado del enlace de datos        -5      octetos
- \* Encabezado de transporte    -1      octetos
- \* Datos de usuario de aplicación                                      =249    octetos

## 3.2 CAMPOS DE ENCABEZADO DE TRANSPORTE

A continuación se describe el octeto del encabezado de transporte, teniendo las definiciones de los bits.

Figura 20. Definiciones de los bits de TH



### 3.2.1 BIT FINAL

El bit final (FIN) indica que esta trama de datos de usuario es la última de una secuencia, la cual compone un mensaje de usuario completo.

- FIN = 0      Más tramas siguen  
          1      Trama final de una secuencia.

### 3.2.2 BIT DE INICIO

El primer bit (FIR) indica que es la primera trama de una secuencia de tramas que componen el mensaje completo. Cuando una estación secundaria recibe una trama con el bit FIR puesto, todas las tramas recibidas previamente, sin terminar la secuencia, son descartadas. La primera trama de una secuencia puede tener cualquier secuencia de 0 a 63. Si la trama es recibida sin el bit FIR puesto y no hay una secuencia de mensaje

en proceso, entonces la trama es ignorada. Si un mensaje de usuario completo es solamente una trama en longitud, entonces ambos bits FIR y FIN están puestos.

FIR = 1      Primera trama de una secuencia.  
         0      No es la primera trama de una secuencia.

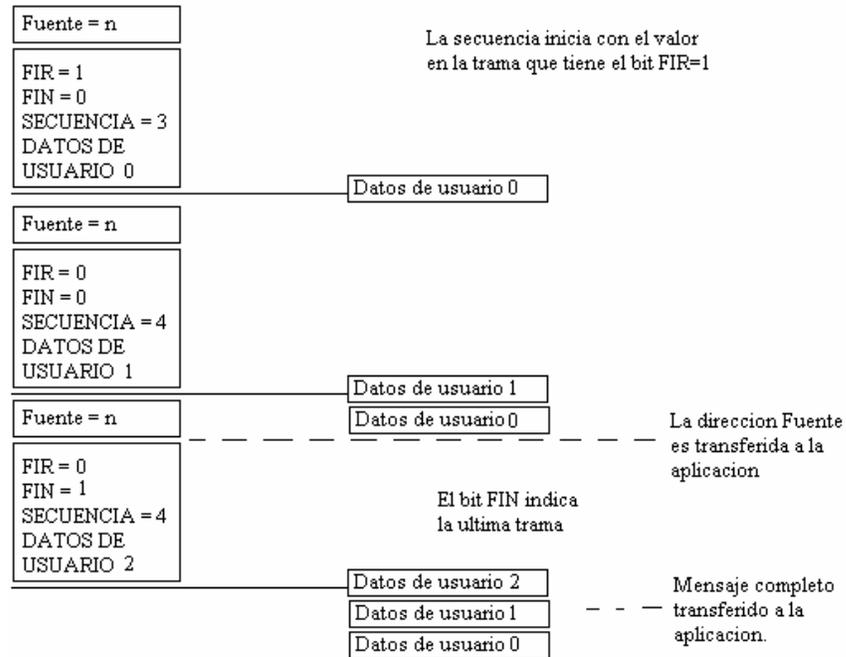
### **3.2.3 SECUENCIA**

El número de secuencia de la trama es usado para revisar que cada trama está siendo recibida en secuencia. Esto protege contra pérdida de tramas o tramas duplicadas. Todos los mensajes de usuario parten con una secuencia especificada en la primera trama, la cual tiene el bit FIR puesto (cada mensaje puede iniciar con cualquier número de secuencia entre 0 y 63). Después del número de secuencia 63, el siguiente número en la secuencia será 0. El número de secuencia se incrementa para cada trama enviada o recibida de la misma dirección, perteneciente al mismo mensaje y se reajusta en el inicio de un nuevo mensaje. El número de secuencia no tiene un incremento a través de los límites de los mensajes, como por ejemplo: cualquier número de secuencia es válido cuando el bit FIR es puesto.

## **3.3 ENSAMBLADO DE TRAMA**

A continuación se ilustra en la figura 21, el ensamblado de un mensaje de tres tramas. La primera trama del mensaje se identifica por el bit FIR puesto en el campo TH. La última trama es identificada por tener el bit FIN puesto en el campo TH.

Figura 21. Ensamblado de datos de tres tramas de datos



### 3.4 TRANSMISIÓN DE MENSAJES

La figura 22, ilustra la transmisión de un mensaje de una trama usando la trama de servicio Envío-Confirmación. La figura 23, ilustra la transmisión de un mensaje de múltiples tramas usando la trama de servicio Envío-Confirmación.

Figura 22. Transmisión de una trama de mensaje

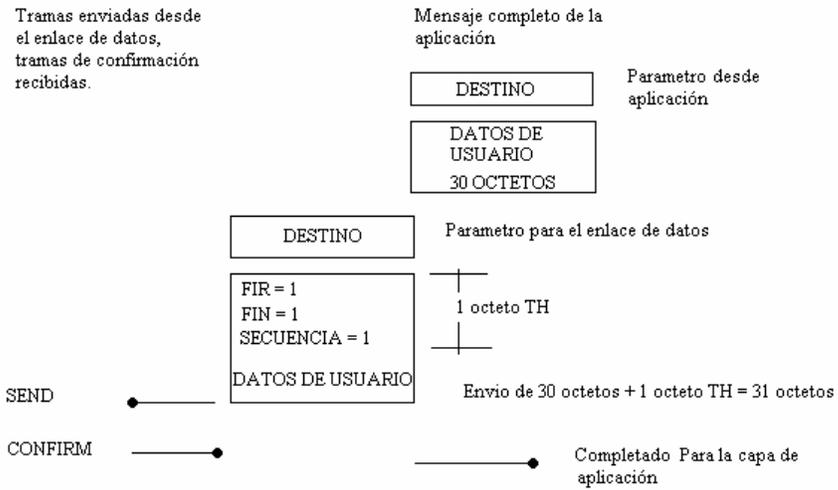
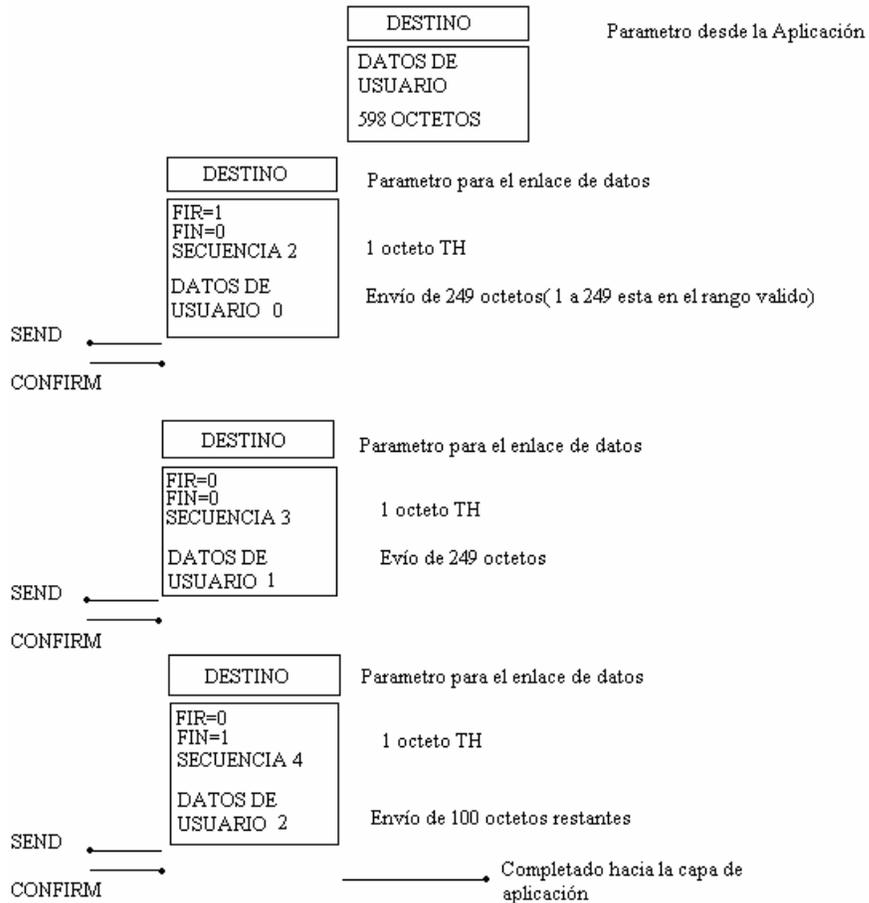


Fig. 23 Fragmentación de un mensaje de aplicación de múltiples tramas.



### **3.5 SERVICIOS DE TRANSPORTE Y RESPONSABILIDADES**

Los requerimientos de comunicación de la capa de red y de la capa de aplicación son satisfechos por servicios primitivos (transacciones que se invocan desde o hacia la capa, a través de puntos de accesos al servicio como el requerimiento, generado por el usuario para invocar una función; servicio de indicación, generado por la capa suministradora de servicios; servicio de respuesta, generada por el usuario para completar la función y servicios de confirmación, generada por el suministrador de servicios para completar la función invocada) de la capa de pseudo-transporte.

La capa de pseudo transporte es responsable de realizar las siguientes funciones:

- \* Empaquetar los datos de usuario sobre múltiples tramas (más de una) del enlace de datos, en el formato de trama definido (FT3) y usando los servicios del enlace de datos para la transmisión de datos.
- \* Desempaquetar múltiples tramas que son recibidas desde el enlace de datos transformados en datos de usuario.
- \* Control de todos los aspectos del enlace de datos, excluyendo la configuración del enlace de datos.

La capa de pseudo-transporte es responsable de proveer los siguientes servicios:

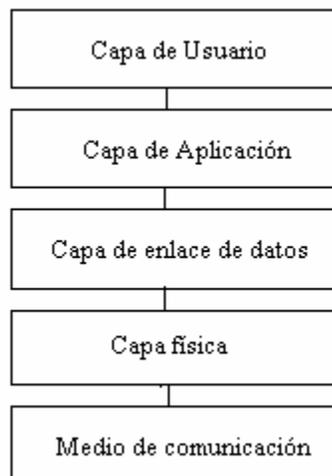
- \* Intercambio de SDUs entre capas de pseudo transporte DNP3 iguales.
- \* Notificación de error al usuario de transporte, cuando una respuesta a un requerimiento no ha sido recibido.
- \* Secuencia de SDUs.
- \* Entrega de SDU priorizada, que puede ser puesta como normal o rápida para indicar una prioridad de requerimiento baja o alta.

- \* Calidad de entrega de SDU ajustada en envío sin repuesta o envío-confirmación para indicar si, o no, el mensaje requiere de conocimiento.

## 4. CAPA DE APLICACIÓN

El modelo ISO-OSI especifica siete niveles o capas, la IEC especifica un modelo simplificado que consiste en la capa física, enlace de datos y aplicación solamente y es la denominada Arquitectura de Desempeño Elevada (EPA: Enhanced Performance Architecture).

Figura 24. Contexto EPA



La interfaz de capa de usuario a la capa de aplicación implica que el usuario no tiene necesidad de saber de otros elementos del sistema de comunicación, excepto por la interfaz de la capa de aplicación. La capa de usuario hace uso de la capa de aplicación para “enviar/recibir” mensajes “SCADA/DA” completos “hacia/desde” una estación maestra o estación remota (outstation).

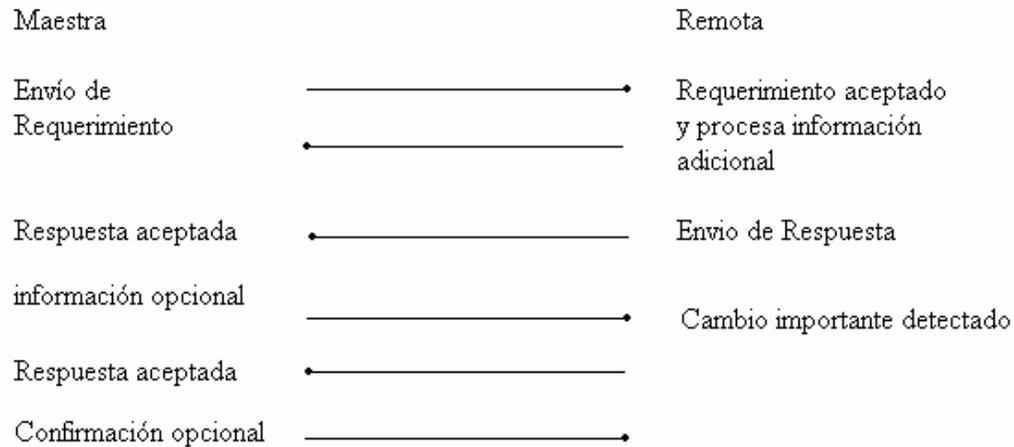
Estructuralmente, la PDU (Protocol Data Unit: Unidad de datos de protocolo) de la capa de aplicación cumple la descripción de la IEC de un APDU (Application Protocol Data Unit). Los usuarios envían unidades de datos de usuario de aplicación a la capa de

aplicación donde son convertidos en ASDU (Application Service Data Unit: unidad de aplicación de datos de servicio). Los datos de usuario de aplicación están convertidos en múltiples ASDUs. Cada ASDU es entonces prefijado por la APCI (Application Protocol Control Information: información de aplicación de protocolo de control), el cual es entonces empaquetado como una APDU. Cada APDU, que es parte de múltiples APDUs, es referida como un fragmento y hay una restricción, donde cada fragmento contiene datos objetos completos solamente, y que el código de función de la APCI es idéntico en cada fragmento del mismo mensaje o múltiples APDU. O sea que no habrá fragmentación de información de objetos entre APDUs y la misma operación debe ser requerida para cada objeto en el mensaje. Esto asegura que cada fragmento por si mismo puede ser procesado y también implica que cada ASDU contiene solamente datos objeto completos. A la inversa, la capa de aplicación recibe varias APDUs (una a la vez), donde se remueve el APCI para obtener la ASDU y ensamblar las ASDUs sobre los datos de usuario de aplicación.

#### **4.1 FORMATOS DE MENSAJE**

En esta especificación la estación maestra es definida como la estación que envía un requerimiento y la estación remota es el dispositivo esclavo, RTU o IED, al cual el mensaje de requerimiento es destinado. Solamente la estación maestra designada puede enviar el mensaje de requerimiento o solicitud de la capa de aplicación y las remotas pueden enviar mensaje de respuesta de la capa de aplicación. Como indica la figura 25, la estación maestra envía un requerimiento de capa de aplicación a la estación remota, la cual retorna una respuesta de capa de aplicación, además que la estación remota puede decidir transmitir datos espontáneamente usando un mensaje de capa de aplicación de respuesta no solicitada.

Figura 25. Secuencia de mensaje de la capa de aplicación (maestra y remota)



Para una maestra, una transacción de “requerimiento/respuesta” con una estación remota en particular, debe ser completada antes que otro requerimiento pueda ser enviado a esa estación remota. Una estación maestra puede aceptar respuestas no solicitadas mientras la transacción de requerimiento está en progreso. Para una estación remota, una transacción de “requerimiento/respuesta” debe ser completada antes que otros requerimientos aceptados o respuestas no solicitadas sean enviadas. Las respuestas no solicitadas pueden ser enviadas antes o después pero no durante la transacción de “requerimiento/respuesta”. Si una estación remota está presente en la mitad de una transacción no solicitada, esta puede condicionalmente aceptar un comando de requerimiento de la estación maestra.

Cuando respuestas no solicitadas son generadas por una estación remota, allí existe la posibilidad que la estación maestra envíe un requerimiento al mismo tiempo que la estación remota envía una respuesta no solicitada. La estación remota puede recibir el requerimiento cuando ésta espera recibir una confirmación de su respuesta no solicitada. La maestra recibe la respuesta no solicitada cuando ésta espera recibir una respuesta a su requerimiento. El proceso de la anterior situación y similares depende del

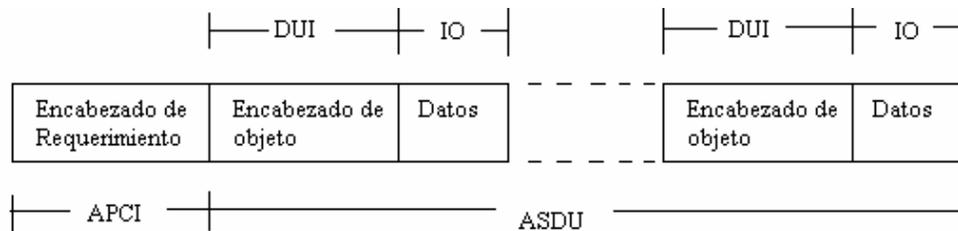
tipo de requerimiento emitida por la estación maestra. La estación maestra generalmente procesará una respuesta no solicitada inmediatamente, siempre que ésta llegue cuando la estación maestra esté esperando una respuesta a un requerimiento previamente emitido. Una confirmación de la respuesta no solicitada es emitida inmediatamente si es requerida por la estación remota (el bit “CON” puesto). La estación remota generalmente procesará un requerimiento inmediatamente, aun si esta esperando por una confirmación de una respuesta no solicitada previa. Todo requerimiento, excepto los requerimientos de lectura (READ) para los datos del sistema (ejemplo: datos de entrada binarios, contador de datos de evento, etc.), Son procesados de esta manera, por lo que este modo de operación es referido como modo de Procesamiento Inmediato. La estación remota no procesará un requerimiento de lectura de la estación maestra, si está esta esperando por una confirmación de una respuesta no solicitada previa. Ésta esperará por la confirmación antes de procesar el requerimiento. La razón de la funcionalidad diferente es para prevenir la pérdida o duplicación de datos de eventos, este modo de operación es referido como modo de Procesamiento después de confirmar.

Además, cada respuesta o requerimiento puede consistir de uno o más fragmentos individuales (el término APDU y fragmento son intercambiables). Cada fragmento, sin embargo, debería ser analizable y entonces ejecutado (porque el código de función es parte de cada fragmento). Por otro lado, existen dispositivos con capacidad de almacenamiento de mensajes limitados y deberían solamente enviar un fragmento de mensaje de requerimiento, cuando una respuesta esperada (de todos los fragmentos enviados) sea más larga que un fragmento. Esto es para asegurar que los dispositivos puedan procesar un requerimiento y construirlo, y más importante enviar una respuesta antes del próximo requerimiento recibido. De lo contrario, mensajes de múltiples fragmentos pueden requerir respuestas de múltiples fragmentos, las cuales pueden requerir más almacenamiento de mensajes que los que tiene los dispositivos.

### 4.1.1 FORMATO REQUERIMIENTO DE APLICACIÓN

El formato de mensaje de requerimiento de aplicación (APDU) se ilustra en la figura 4.3. El APDU es hecho de un bloque APCI que contiene información de control del mensaje y un ASDU que contiene información para ser procesada por la estación receptora. El APCI es usualmente llamado un encabezado de requerimiento, en un mensaje de requerimiento de aplicación. El ASDU, aunque es opcional, es usado cuando el significado del mensaje no es transmitido completamente en el encabezado de requerimiento. El encabezado de requerimiento contiene información sobre cómo ensamblar un mensaje de múltiples tramas y el propósito del mensaje. El encabezado de requerimiento está presente en todos los APDUs de requerimientos de la capa de aplicación. Cada ASDU consiste de uno o mas Identificadores de Unidad de Datos (DUI: Data Unit Identifiers) o encabezados de objeto y objetos de información asociados opcionales (IO: Information Objects) o campos de datos.

Figura 26. Formato de requerimiento de aplicación

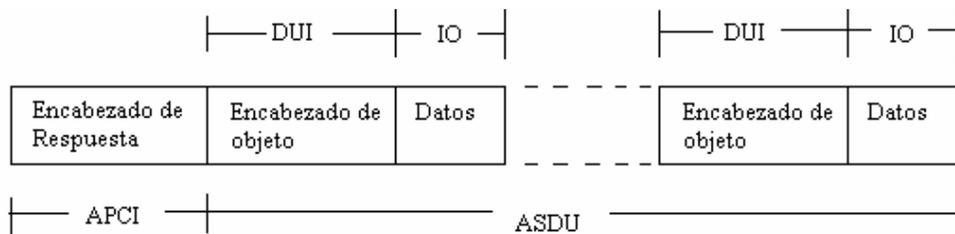


El encabezado de requerimiento identifica el propósito del mensaje y consiste de información de control de protocolo de aplicación (APCI: Application Protocol Control Information). El encabezado de objeto (Object Header) identifica los datos objeto que siguen. Los datos son del tipo de datos objeto especificados en el encabezado de objeto.

#### 4.1.2 FORMATO DE RESPUESTA DE APLICACIÓN

La respuesta de una estación remota a un requerimiento APDU de la capa de aplicación o de una respuesta no solicitada de una estación remota, tiene el formato ilustrado en la figura 27. El formato es idéntico en forma al requerimiento. El APCI es usualmente llamado un encabezado de respuesta, en un mensaje de respuesta de aplicación. El encabezado de respuesta contiene la misma información como el encabezado de requerimiento, más un campo adicional que contiene indicaciones internas de la estación remota. El encabezado de respuesta es siempre parte de la respuesta de aplicación.

Figura 27. Formato de respuesta de aplicación



El encabezado de respuesta identifica el propósito del mensaje y consiste en el APCI, el encabezado de objeto (Object header) identifica los datos objeto que siguen. Los datos son del tipo especificado en el encabezado de objeto.

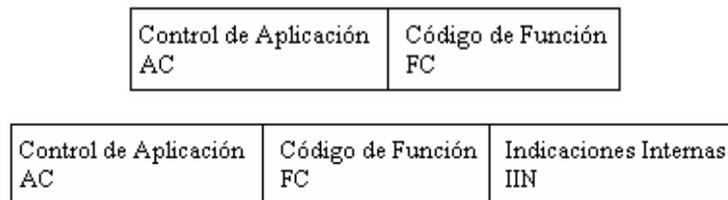
## 4.2 CAMPOS DE MENSAJE DNP

### 4.2.1 ENCABEZADOS DE APLICACIÓN

#### 4.2.1.1 Encabezado de requerimiento y respuesta

El encabezado de requerimiento o APCI tiene dos campos, cada campo es un octeto en longitud. El encabezado de respuesta tiene adicional un campo de indicaciones internas (IIN).

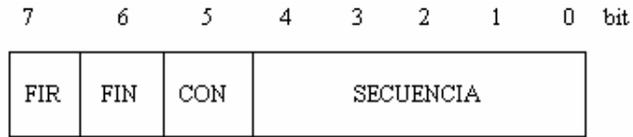
Figura 28. Encabezado de requerimiento y abajo el encabezado de respuesta



#### 4.2.1.2 Control de Aplicación

El campo de control de aplicación tiene de tamaño un octeto. Éste provee la información necesaria para construir un mensaje de aplicación de múltiples fragmentos. Un mensaje de aplicación puede ser empaquetado en fragmentos, con cada fragmento lo suficientemente pequeño para ocupar el buffer de mensaje de la aplicación. El tamaño recomendado del buffer de fragmento es de 2048 bytes a manera de mantener compatibilidad de dispositivos. Cada fragmento tiene un encabezado de aplicación y un apropiado encabezado de objeto, así cada fragmento puede ser procesado como mensajes individuales que se pueden descartar para abrir paso al siguiente fragmento.

Figura 29. Campo de control de aplicación



El bit FIR es puesto a uno, entonces éste indica que el mensaje fragmentado tiene el primer fragmento de un mensaje de aplicación completo. Si el bit FIN es puesto a uno, indica qué fragmento de mensaje es el fragmento final de un mensaje completo. El bit CON (confirmación) puesto a uno, indica que la aplicación transmisora está esperando una confirmación desde la aplicación receptora del fragmento recibido. Una función de aplicación con el código cero es usada en la confirmación del mensaje. La secuencia (SEQUENCE) indica el número de fragmento. Los números de fragmentos del cero al 15 son reservados para requerimientos de la estación maestra y todas las respuestas de las estaciones remotas (pero no las repuestas sin solicitud). Los números de fragmento del 16 al 31 son reservados para respuestas sin solicitud de las estaciones remotas. Para respuestas sin solicitud, cada fragmento consecutivo de una estación remota debe tener un número de secuencia de incremento (el número se salta de 31 a 16). Los requerimientos para una estación remota y respuesta de una estación remota, los fragmentos consecutivos recibidos de o transmitidos a la misma estación remota deben tener un número de secuencia de incremento (el número salta de 15 a cero).

### 4.3 CONTROL DE FLUJO DE INFORMACIÓN

El flujo de requerimientos y respuestas entre la maestra y la remota está controlada por campos en la respuesta y encabezados de requerimiento como lo son los temporizadores de aplicación (timers) y parámetros, por lo tanto los campos, timers y parámetros que controlan el flujo de mensaje son:

- 1 bit de campo CON. “Poniendo/limpiando” este bit “habilita/deshabilita” los mensajes de confirmación de respuesta (CONFIRM). Una respuesta de confirmación es una aplicación de reconocimiento del requerimiento previo o mensaje de respuesta.
- 2 Bits de campo FIR y FIN.
- 3 Campo de número de secuencia. Este número es usado para ensamblar mensajes de múltiples fragmentos e identificar cuáles respuestas corresponden a requerimientos en particular.
- 4 Tiempos fuera (Time out) de respuesta en las aplicaciones de la estación maestra y la estación remota. Éstos especifican qué tanto una aplicación debe esperar para una respuesta o mensaje de confirmación, antes de retransmitir o abortar la transacción. La aplicación puede o no soportar una retransmisión de las transacciones en la capa de aplicación.
- 5 Cantidad de intentos de aplicaciones de la estación maestra y estación remota. Las aplicaciones pueden o no soportar un nivel de intentos de aplicación. Los contadores de intentos especifican cuántas veces un requerimiento es repetido si una respuesta falla, o qué tan seguido una respuesta es retransmitida si una respuesta de confirmación no es recibida.

Las siguientes reglas dictan cómo trabajan los números de secuencia.

- \* El rol del número de secuencia es través de 15 a cero ó de 31 a 16. Cada fragmento de requerimiento sucesivo de la estación maestra DNP, tiene un incremento de número de secuencia. La excepción es para reintentos en los requerimientos. Para un reintento de requerimiento con simple fragmento, el número de secuencia no es incrementado. Para reintentos de requerimiento con múltiples fragmentos, el número de secuencia del primer fragmento del reintento de requerimiento iguala el número de secuencia del último segmento del requerimiento, el cual falló.

- \* Una respuesta de fragmento simple para un requerimiento de fragmento simple, tiene el mismo número de secuencia como el número del requerimiento o respuesta.
- \* Una respuesta de confirmación para un requerimiento o respuesta tiene el mismo número de secuencia como el requerimiento o respuesta.
- \* El primer fragmento de una respuesta de múltiples fragmentos para un requerimiento de fragmento simple, tiene el mismo número de secuencia como el requerimiento. Para fragmentos sucesivos de la respuesta de múltiples fragmentos, la secuencia del número es incrementada.
- \* El primer fragmento de una respuesta de múltiples tramas para un requerimiento de múltiples fragmentos, tiene el mismo número de secuencia como el último fragmento del requerimiento de múltiples fragmentos.

El uso de estos esquemas de secuencia de número, asegura que la estación remota y la estación maestra puedan hacer frente con todas las veces que los mensajes se han perdido o retrasado sobre una red de comunicaciones. Las siguientes reglas son obedecidas por ambas estaciones maestra y remota:

- \* Si el sistema usa un nivel de reintentos de aplicación, cuando una respuesta no es recibida antes de un tiempo fuera, el requerimiento debe ser retransmitido con el mismo número de secuencia.
- \* Si dos mensajes son recibidos con el mismo número de secuencia, esto usualmente significa que la respuesta para el mensaje no fue recibido por la otra estación, en este caso, retransmitir la respuesta.
- \* Si dos respuestas de confirmación son recibidas con el mismo número de secuencia, se ignora la segunda respuesta.

#### 4.4 CÓDIGOS DE FUNCIÓN

Los códigos de función identifican el propósito del mensaje. El tamaño de este campo es un octeto. Hay dos grupos de códigos de función; uno para requerimientos y el otro para respuestas.

Tabla III. Códigos de función de transferencia

Código	Función	Descripción
0	Confirmación	Los fragmentos de mensaje de confirmación usados en ambos requerimientos y respuestas. Para este mensaje no se requiere respuesta.
1	Lectura	El requerimiento especifica objetos de la estación remota, responde con objetos requeridos que están disponibles.
2	Escritura	Almacena objetos especificados en la estación remota, responde con el estado de la operación.

Tabla IV. Códigos de función de control

Código	Función	Descripción
3	Seleccionar	Seleccionar o armar puntos de salida, pero no ajustarlos o producir cualquier acción de salida (controles, puntos de nivel, salidas análogas), responde con el estado de los puntos de control seleccionados. El código de función de operación es requerido para activar estas salidas.
4	Operación	Ajusta o produce las acciones de salida sobre los puntos previamente seleccionados con la función de seleccionar, responde con el estado de los puntos de control.
5	Operación directa	Selecciona y opera o ajusta las salidas especificadas, responde con el estado de los puntos de control.
6	Operación directa - sin reconocimiento	Selecciona y opera o ajusta la salida especificada pero no envía una respuesta al requerimiento.

Tabla V. Códigos de funciones congeladas

Código	Función	Descripción
7	Congelado Inmediato	Copia los objetos especificados para congelarlos en el buffer y responder con el estado de la operación.
8	Congelado Inmediato - Sin reconocimiento	Copia los objetos especificados para congelarlos en el buffer, no responde con un mensaje.

Tabla VI. Códigos de función de transferencia

Código	Función	Descripción
9	Congelar y limpiar	Copia los objetos especificados para congelar el buffer, entonces limpia los objetos, responde con el estado de la operación.
10	Congelar y limpiar - sin reconocimiento	Copia los objetos especificados para congelar el buffer, entonces limpia los objetos, no responde con un mensaje.
11	Congelado con tiempo	Copia los objetos especificados para congelar el buffer en el tiempo especificado a intervalos, responde con el estado de la operación.
12	Congelado con tiempo- sin reconocimiento	Copia los objetos especificados para congelar el buffer en el tiempo especificado a intervalos, no responde con un mensaje.

Tabla VII. Códigos de función de aplicación

Código	Función	Descripción
13	Reinicio en frío	Ejecuta la secuencia de reajuste (reset) deseada, responde con un objeto de tiempo indicando cuanto tiempo la estación remota tiene disponibilidad.
14	Reinicio en caliente	Ejecuta la secuencia parcial de reajuste, responde con un objeto de tiempo, indicando cuanto tiempo la estación remota tiene disponibilidad.
15	Inicializar datos por defecto	Inicializa los datos especificados para levantar los valores iniciales, responde con el estado de la operación.
16	Inicialización de aplicación	Prepara las aplicaciones específicas para correr, responde con el estado de la operación.
17	Inicio de Aplicación	Inicia corriendo las aplicaciones específicas, responde con el estado de la operación.
18	Detener la aplicación	Detiene las aplicaciones específicas, responde con el estado de la operación.

Tabla VIII. Códigos de función de configuración

Código	Función	Descripción
19	Guardar Configuración	Guarda la configuración especificada en una memoria no volátil, responde con un objeto de tiempo, indicando el tiempo hasta que la estación remota esté disponible.
20	Habilitación de mensajes no solicitados	Habilita el reporte espontáneo de los datos objetos especificados, responde con el estado de la operación.
21	Deshabilitación de los mensajes no solicitados	Deshabilita el reporte espontáneo de los datos objetos especificados, responde con el estado de la operación.
22	Asignación de clase	Asigna los datos objetos especificados para una clase particular.

Tabla IX. Códigos de función de respuesta, reservados y de tiempo de sincronización

Código	Función	Descripción
23	Medida de retraso	Permite a la aplicación calcular el camino del retraso (o retraso de propagación) para una estación remota en particular. El valor calculado de este código de función debe ser usado para ajustar el tiempo del día cuando se fija el tiempo de la estación remota
24-120		Reservados
121-128		Reservados para pruebas
0	Confirmación	La confirmación del fragmento del mensaje es usada en ambos requerimientos y respuestas. No hay respuesta requerida para este mensaje.
129	Respuesta	Respuesta a un mensaje de requerimiento
130	Mensaje no solicitado	Respuesta no solicitada que no fue puesta por un requerimiento

Los campos de indicaciones internas (IIN) se definen como un campo de dos octetos, que siguen al código de función en todas las respuestas. Cuando un requerimiento no puede ser procesado debido a la formación de errores o el requerimiento de datos no está disponible, las indicaciones internas (IIN) siempre retornan con los bits puestos de manera apropiada.

## 4.5 ENCABEZADO DE OBJETO

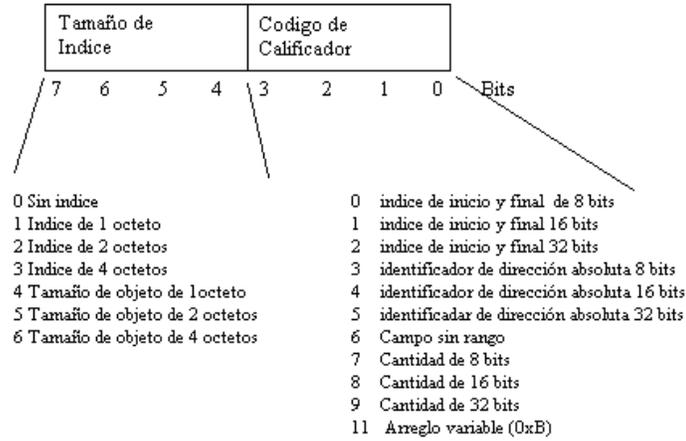
El encabezado de objeto de un mensaje, especifica los datos objeto (IOs), que también están contenidos en el mensaje o son usados para responder a este mensaje. El formato del encabezado de objeto es idéntico para un requerimiento y una respuesta pero la interpretación del encabezado es dependiente sobre si es un requerimiento o una respuesta y qué código de función acompaña el encabezado.

Figura 30. Encabezado de objeto

Objeto	Calificador	Rango
--------	-------------	-------

El objeto (object) especifica el grupo de objeto y la variación de los objetos que siguen el encabezado. El campo de objeto identifica únicamente el tipo o clase de objeto, estructura y el tamaño del dato objeto, este es un campo de dos octetos. El calificador (Qualifier) especifica cómo el rango del campo es interpretado, éste es un campo de un octeto. El rango indica la cantidad de objetos e identifica únicamente los objetos en cuestión. El campo de rango no puede estar presente si el calificador especifica que no hay campo de rango y tiene de tamaño de cero a ocho octetos.

Figura 31. Estructura del campo de calificador



Los objetos son asignados a clases, de las que hay cuatros clases de datos. La clase cero esta reservada para datos objetos estáticos (los datos estáticos reflejan el valor actual de los datos en la estación remota). Las clases uno, dos y tres son reservadas para eventos de datos objetos (Los objetos creados como resultado de los cambios en los datos de la estación remota). Cada evento de objeto puede ser asignado a la clase uno, dos o tres. Los objetos pueden ser agrupados en clases por su prioridad (la prioridad está determinada por el usuario). La clase de dato es usada por la estación maestra para requerir de datos objetos pre-asignados en demanda o base disponible de la estación remota.

Una clase de un encabezado de dato objeto puede ser usado solamente en un requerimiento para indicar a la estación remota qué dato-objeto retorna. La estación remota retornará (en respuesta) los encabezados de objeto de los datos objetos reales y no la clase de encabezado de objeto.

## 4.6 LIBRERÍA DE DATOS-OBJETO

Los elementos de información o datos-objetos de DNP son utilizados en la capa de aplicación, también son usados por dispositivos inteligentes y que son capaces de monitorear, controlar y/o producir diferentes datos, tanto en los niveles de hardware como software. Estos elementos de información son procesados y almacenados como objetos de información y de esta forma empaquetados para la transmisión como unidades de datos de aplicación.

### 4.6.1 REGLAS DE ELEMENTOS DE INFORMACIÓN

Las reglas para la declaración de elementos de información han sido derivadas de la IEC TC 57 870. Los documentos del perfil del dispositivo son usados para indicar el origen exacto y el significado del dato objeto de cada dispositivo de telecontrol.

#### 4.6.1.1 Tipos de datos

Todos los tipos de datos son reconocidos como construcciones estándar, usadas en muchos lenguajes de compiladores. Los elementos de información DNP usan estas construcciones.

Tabla X. Tipos de datos

Tipo de dato	Símbolo	Significado
1. Entero no asignado	UI	Número entero positivo
2. Entero	I	Número entero positivo o negativo
3. Punto fijo no asignado	UF	Número de punto fijo de número positivo
4. Punto fijo	F	Número de punto fijo de número positivo o negativo
5. Real	R	Número positivo o negativo de punto flotante
6. Bitstring	BS	Ensamblado de bits independientes, string de bits de tamaño 1
7. Octectstring	OC	Ensamblado de octetos.

### 4.6.1.2 Tamaño de dato

Cada elemento de información está compuesto de un tipo de dato y un tamaño. El tamaño del dato  $i$ , es nombrado después de la notación del símbolo del tipo de dato, y es un número cardinal que especifica la longitud del campo de dato en bits u octetos. Un ejemplo es:  $BS12$  que especifica un Bistring de 12 bits.

### 4.6.1.3 Posición de bit

En la definición de los objetos de información, los cuales son una combinación de elementos de información, la posición individual de los bits puede ser significativa. La posición del bit de un campo especificado de tamaño de dato  $i$  es nombrado en paréntesis cuadrados  $[p1...pn]$ , donde  $p1$  y  $pn$  son el primero y el último bit del campo.

Figura 32. Posición de bits

BITS	7	6	5	4	3	2	1	0
OCTETS	Data Size $i$							
1	7	6	5	4	3	2	1	0
2	15	14	13	12	11	10	9	8
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
$j$	$8j-1$	$8j-2$	$8j-3$	$8j-4$	$8j-5$	$8j-6$	$8j-7$	$8j-8$

## **4.6.2 REGLAS Y ESTRUCTURA DE LIBRERÍA**

### **4.6.2.1 Estructura de librería**

En la capa de aplicación DNP un objeto es de ocho bits y un campo de variación de 8 bits usados para nombrar el dato objeto. Los objetos de ocho bits se nombran para tipos de datos generales, como los datos binarios estáticos. La variación de este objeto da una representación diferente del mismo punto de dato, tal como el tamaño del objeto, si, o no el objeto tiene una bandera (flag) de información. Hay generalmente cuatro diferentes categorías de datos, con los que cada tipo tiene su esquema:

- \* Objetos estáticos: Objetos con los que se refleja el valor real del punto del campo o punto de software.
- \* Objetos de eventos: Objetos con los cuales son generados como un resultado del cambio de los datos u otro estimulante. Éstos son objetos tipo historiales, reflejando el valor del dato en algún tiempo en el pasado.
- \* Objetos estáticos congelados: Objetos con los cuales se refleja una congelación actual del valor del punto del campo o punto de software. Los datos están congelados como resultado de requerimientos congelados.
- \* Objeto de eventos congelados: Objetos que son generados como el resultado del cambio de datos congelados o algún otro estimulante. Estos son objetos de historial, reflejando el valor del dato cambiado en algún tiempo en el pasado.

### **4.6.2.2 Numeración del punto**

Las siguientes reglas se aplican a la interpretación del número de punto de objeto, en conjunto con objetos y variaciones.

**Regla 1:**

El punto **i** del objeto **x**, variación **y** representan el mismo punto físico como el punto **i**, objeto **x**, variación **z**, donde **y** y **z** son variaciones del objeto **x**. Por ejemplo: un dispositivo tiene 16 contadores funcionando (objeto 20) numerados de cero a 15. El punto cinco puede ser preguntado de cuatro formas diferentes.

- \* Obj 20, var 1, rango 5 retorna el valor del contador funcionando 5 en formato de 32 bits.
- \* Obj 20, var 2, rango 5 reporta la misma información, solamente en formato de 16 bits.
- \* Obj 20, var 3, rango 5 retorna el número de la cantidad acumulada del contador 5 desde la última vez que fue reportado, en formato de 32 bits.
- \* Obj 20, var 4, rango 5 reporta la misma información, pero en formato de 16 bits.

**Regla 2:**

El punto **i** del objeto **x** puede ser reportado en una de varias maneras (ejemplo: Pueden ser contadores de 16 bits o 32 bits). Cuando se reporta como evento, el punto **i** puede ser retornado en cada una de las variaciones para cada objeto. La variación exacta para retornar es una decisión específica de aplicación, sin embargo una aplicación debe reportar solamente un objeto-evento en cualquier variación para cada evento. Cuando se responde un requerimiento para la clase de dato o variación cero del objeto **x**, ésta solamente sería una variación del objeto retornado.

**Regla 3:**

El punto **i** dentro de objetos diferentes de la misma agrupación no son necesariamente únicos, sin embargo, dentro de cada grupo de entrada binaria, salida binaria, entrada analógica, salida analógica y de contadores se aplican las siguientes reglas:

- \* El punto **i** del objeto estático es el mismo punto físico, como el punto **i** en el objeto evento.

\* El punto **i** en el objeto congelado es el mismo punto físico, como el punto **i** en el objeto de evento congelado.

Por ejemplo: Para entradas binarias, el punto **i** en el obj. 1 var. 1 y 2 es el mismo punto como el punto **i** en el obj. 2 var. 1, 2 y 3 (correlación de estático y evento).

**Regla 4:**

Las agrupaciones de objetos con los cuales pueden por definición, o debido a la limitación del dispositivo, tener solamente un punto por objeto o donde el número de punto no es necesario debe usar el rango de número cero o cantidad igual a uno, cuando se está usando un formato de mensaje que requiere un número de punto.

#### **4.6.2.3 Objetos y variaciones**

A continuación se mencionan las especificaciones de codificación de los elementos de información o datos objetos usados en la capa de aplicación, de interés en la estructura de la aplicación, la información de objetos son usados en la capa de aplicación. Es importante considerar que los dispositivos que utilizan la capa de aplicación son capaces de monitorear, controlar y/o proporcionar grandes cantidades de datos, tanto en los niveles de hardware como software, los datos llamados elementos de información son procesados y almacenados como información de “objeto” y estos son empaquetados para la transmisión como unidades de datos. La cantidad de datos objeto procesados o analizados por los dispositivos, son agrupados en tablas de subconjunto de implementación (Véase Tabla XIV de subconjuntos en el anexo), que contiene tanto el objeto como las variaciones que comprende el objeto. A continuación, se describen los grupos de objetos, especificando el significado de los bits que componen el octeto de los datos objeto.

#### **4.6.2.3.1 Entradas y salidas binarias**

Las entradas binarias representan el estado de un punto de entrada digital (hardware o software), usadas para transmitir estos estados en un formato de paquetes, la transmisión de los datos objeto es siempre realizada en octetos completos, con las posiciones no ocupadas de bit puestas a cero.

Además, también las entradas binarias con estado indican el estado del punto de entrada digital, el bit on-line indica que el punto de entrada binaria ha sido leído completamente, el bit de restart indica que el dispositivo de campo que originó el dato objeto está actualmente reiniciándose, el bit de pérdida de comunicación indica que el dispositivo que reporta este dato-objeto ha perdido la comunicación con el originador del dato objeto, el bit de dato local forzado indica que el estado de la entrada binaria ha sido forzado a su estado actual, en el final del dispositivo. El bit de filtro (chatter filter) indica que el punto de entrada binaria ha sido filtrada a manera de remover transiciones innecesarias en el estado del punto. El bit de estado indica el estado actual del punto de entrada binaria.

El cambio de entrada binaria sin tiempo, es usado para representar el estado cambiado de un punto de entrada digital y también indica el estado del punto, el cambio de entrada binaria con tiempo es un objeto usado para representar el estado cambiado de un punto de entrada digital y el tiempo al cual este estado cambió, por lo que el tiempo de ocurrencia indica el tiempo absoluto, al cual el dispositivo de telecontrol ha detectado el cambio de estado. El objeto con cambio relativo de tiempo es usado para representar el estado cambiado de un punto de entrada digital, y el tiempo relativo al cual el estado cambió. El tiempo común de objeto (CTO: common time object) es usado como un tiempo absoluto base para los objetos con cambio de tiempo relativo. El tiempo relativo en cada objeto de entrada binaria es sumado al tiempo absoluto de CTO, para dar el tiempo absoluto al cual el cambio de la entrada binaria fue detectado por el dispositivo.

La salida binaria es usada para controlar un punto digital (hardware o software), el bloque de información de control de relevador de salida contiene una salida digital de parámetros de control. Estos parámetros definen el tipo y duración de la salida digital, específicamente, el campo de código de control (control code) indica la función de control a realizar, el campo de cantidad (count) indica el número de veces que la operación de control debe realizarse en sucesión, el campo sobre el tiempo (on time) especifica la cantidad de tiempo que está activa la salida digital. El campo fuera de tiempo (time out) especifica la cantidad de tiempo que está inactiva la salida digital.

El objeto patrón de bloque de control (PCB) contiene parámetros de control de salida digital para los tipos de patrones de control. Estos parámetros definen el tipo y duración de la salida digital para cada punto afectado. El objeto PCB define todos los parámetros para el subsecuente objeto-patrón máscara que le sigue a este objeto en el mensaje. La combinación de este objeto y los objetos de patrón máscara que siguen especificarán un número de operaciones de control a realizar.

El objeto-patrón máscara es usado para seleccionar entre un rango posible de puntos de control que han sido ejecutados en paralelo. Este objeto es usado conjuntamente con el objeto PCB, para especificar tanto los puntos de control como los parámetros. Si la máscara es puesta activa, entonces los parámetros especificados en el PCB precedente son aplicados a un punto específico para cada patrón-máscara y una operación de control es generada para el punto.

#### **4.6.2.3.2 Contadores**

Un contador binario representa un valor acumulado, éste puede ser pulsos o transiciones acumuladas de un punto de hardware o software, el campo de valor (value) muestra el valor actual del contador en el tiempo de reporte o el último valor reportado

del dispositivo origen. Este valor se incrementa indefinidamente hasta la operación de clear (ésta realiza un reajuste del valor a cero).

El bit de roll-over indica que el valor acumulado ha excedido el valor permitido ( $2^{32} - 1$ , o,  $2^{16} - 1$  según sea de 32 bits o 16 bits). El valor del contador se reajusta a cero hasta que el conteo y el roll-over se resumen como normales.

Un contador congelado (de 32 ó 16 bits) está compuesto de un objeto de información que contiene información sobre el contador que fue congelado previamente. El valor congelado (frozen value) muestra el valor del contador cuando el último conteo congelado fue realizado en el dispositivo origen, el contador acumula pulsos o transiciones de un punto de hardware o software.

Un contador congelado con tiempo de congelado está compuesto de un objeto de información, sobre el contador que fue congelado previamente. El valor congelado (frozen value) muestra el valor del contador cuando el tiempo se considera tiempo de congelado (time of freeze). Un contador congelado delta con tiempo de congelado representa un valor congelado, que se ha acumulado desde el último valor que fue reportado.

El contador de cambio de evento sin tiempo representa un valor de contador que, desde el último reportado, excede la cantidad configurada. El campo de valor general (current value) muestra el valor del contador que generó el evento. Para un contador delta de cambio de evento sin tiempo (también con tiempo), el valor en el contador delta representa que ha excedido una “deadband” (periodo de tiempo sin cambio o banda muerta) configurada.

El contador congelado de evento con tiempo y sin tiempo (ya sea de 32 ó 16 bits) representa un valor de contador congelado que es reportado como un evento. El campo de valor congelado muestra el valor en el tiempo de congelación.

#### **4.6.2.3.3 Entradas y salidas análogas**

Una entrada análoga es un objeto de información usado para representar un punto análogo de hardware o software, que puede representar una señal digitalizada o valor calculado. El campo de valor muestra el valor general de una entrada análoga en el tiempo de reporte o el último valor reportado del dispositivo origen. El campo de bandera (flag) tiene como adicional: el campo fuera de rango (out o frange) que indica que la señal digitalizada o cálculo ha excedido. El valor será de:  $+2^{31} - 1$  o  $-2^{31}$  si éste está sobre el rango o bajo el rango para una entrada análoga de 32 bits; el campo fuera de rango que es excedido para una entrada de 16 bits es de:  $+2^{15} - 1$  positivamente o  $-2^{15}$  negativamente. El campo de referencia (reference check) indica que la señal de referencia usada para digitalizar la entrada análoga no es estable y el valor digitalizado resultante puede no ser correcto.

Una entrada análoga congelada es un objeto de información usado para representar un punto análogo de hardware o software, que podría representar una señal digitalizada o valor calculado. El valor congelado muestra el valor de la entrada análoga en el tiempo del último comando congelado que fue realizado en este punto. La entrada análoga con tiempo de congelado es un objeto de información usado para representar un punto análogo congelado de hardware o software. El valor congelado muestra el valor de la entrada análoga en el tiempo especificado en el tiempo de congelado.

El cambio de evento análogo sin tiempo es un objeto de información usado para representar un punto análogo cambiado de hardware o software, que podría representar una señal digitalizada o valor calculado. El campo de valor general (current value)

muestra el valor general de la entrada análoga en el tiempo de reporte o el último valor reportado del dispositivo origen. Este objeto será reportado solamente si el valor general y el último valor reportado difieren por un valor de banda muerta configurable. El cambio de evento análogo con tiempo es un objeto de información usado para representar un punto análogo cambiado de software o hardware. El valor general (current value) muestra el valor de la entrada análoga, en el tiempo especificado en el campo de tiempo. El campo de tiempo, muestra el tiempo al cual se procesa el evento causado.

Un evento análogo congelado sin tiempo es un objeto de información usado para representar un punto congelado de hardware o software que es reportado como un evento. El campo de valor congelado muestra el valor de la entrada análoga en el tiempo de congelado. El evento análogo congelado con tiempo es un objeto de información usado para representar un punto análogo congelado de hardware o software que es reportado como un evento.

La información de estado de la salida análoga es un objeto que representa el valor actual de una salida análoga, el valor actual de un hardware convertido de digital a analógico. El campo de valor actual contiene el valor general de una salida análoga.

El bloque de salida análoga, es un objeto de información que representa el valor deseado de una salida análoga, de un hardware convertidor de digital a analógico o un punto de software. El valor representado es puramente lógico, como el valor puede ser escalado o manipulado antes que el nivel de salida sea puesto. El campo de valor requerido (request value) indica el estado de una operación de control análoga de la misma manera que el bloque de salida de control de relevador.

#### **4.6.2.3.4      Objetos de fecha y tiempo**

El objeto de fecha y tiempo es un objeto de información que representa el tiempo absoluto del día y fecha. Este objeto podría ser usado para sincronización de tiempo. El objeto de tiempo y fecha con intervalo, representa tanto el tiempo absoluto como un intervalo de tiempo. El tiempo absoluto representa el tiempo base (o de inicio), y el intervalo de tiempo es un desfase positivo de el tiempo base. El intervalo podría aplicarse varias veces al mismo tiempo base, a manera de especificar una secuencia de tiempo periódica.

La fecha y tiempo de CTO es un objeto de información que representa el valor absoluto del día, este objeto puede ser usado en conjunto con otros objetos que contienen tiempos de referencia; este objeto es de tiempo base para un tiempo relativo (de incremento) que puede ser añadido, o que puede ser extraído a manera de obtener otro tiempo absoluto de referencia. El tiempo y fecha CTO no sincronizado es un objeto de información que representa el tiempo diario relativo-absoluto, el tiempo absoluto real puede ser calculado por la recepción de un mensaje, basado en el tiempo de recepción del mensaje que contiene este mensaje. El tiempo relativo-absoluto es el tiempo del día no sincronizado en la estación que envía el objeto.

El objeto de retraso de tiempo ordinario representa un tiempo relativo, que indica un período de tiempo que inicia en el tiempo de la recepción del mensaje. El retraso de tiempo fino es un objeto que se representa como el retraso de tiempo ordinario y además puede ser usado en la sincronización del tiempo para realizar un cálculo de ruta de retraso medida.

#### **4.6.2.3.5 Definición de los objetos de clase**

El objeto de clase cero es un objeto de lugar, estos elementos pueden ser tipos de objeto enteros, una variación específica, ciertos puntos de variación, o cualquier combinación de éstos. El dato especificado por este tipo de objeto es configurable junto con la estación que está respondiendo. El dato de clase cero no es un dato de prioridad.

El objeto de clase uno especifica una clase entre cero o más elementos de información. Estos elementos pueden ser objetos enteros, una variación específica, ciertos puntos de variación o la combinación de éstos. El dato especificado por este tipo de objeto es configurable junto con la estación que responde. La estación que responde no envía objetos de datos clase uno, como no contiene información actual, pero es simplemente un identificador para otros objetos. Típicamente un dato de clase uno tiene mayor prioridad que los datos de clase dos, tres y cero.

Los datos de clase dos son similares a los de clase uno usados para requerir una configuración de grupo, cambios usuales de objetos de información de una estación que responde, los de clase tres son también similares a uno y dos; no contienen información actual, pero son simplemente identificadores para otros objetos.

#### **4.6.2.3.6 Objeto de archivo**

(Objeto: 70, Variación: 1) El identificador de archivo es un objeto de información que representa información acerca de un archivo en un sistema de archivos de red. Este objeto está orientado para ser usado en transferencias de bloques largos de datos que no siguen el formato de un dato-objeto existente. En particular, este objeto se sitúa para subir o bajar archivos de configuración para dispositivos remotos o concentradores de datos. La operaciones de archivo están definidas para este objeto y permiten copiar, borrar, etc. en el archivo. El contenido de un objeto de archivo y el

procedimiento exacto para manejar el archivo no es interpretado por la capa de aplicación, mas bien debe ser realizado por el usuario de la capa de aplicación.

#### **4.6.2.3.7 Objetos de Dispositivo**

Las indicaciones internas (internal indications) son elementos de información usados para convertir estados internos y diagnóstico de resultados de una estación respondiente. Esta información puede ser usada por una estación receptora para realizar recuperaciones de error u otras funciones.

El objeto de almacenamiento es un elemento de información usado para convertir a estados en los buffers internos y áreas de almacenamientos tipos de datos específicos, el campo de grupo indica el grupo (o tipo de dato) al que el campo de estado (status) corresponde; el campo de variación indica la variación del objeto que corresponde al campo de estado. Los campos de grupo y variación juntos especifican el tipo de dato exacto. El campo de estado muestra qué porcentaje del espacio en el buffer es permitido para este tipo de dato generalmente usado. El bit de sobre-flujo (overflow) indica que el espacio del buffer para el tipo de dato especificado, ha sido sobre-utilizado, y los datos objetos se han perdido.

El objeto de perfil de dispositivo se provee para interoperabilidad entre diferentes dispositivo DNP que usan solamente un sub-conjunto (sub-set) de funciones de código y datos objetos de la capa de aplicación. El objeto de registro privado (PRO) es reservado para definiciones específicas de los fabricantes, el objeto consiste de un encabezado fijo para la transferencia transparente de datos y un número de registro único del siguiente objeto.

#### **4.6.2.3.8      Objetos de aplicación**

(Objeto: 90, Variación 1) El objeto de identificación de aplicación es usado para representar una aplicación o un proceso del sistema operativo junto al dispositivo. Este objeto es usado en conjunto con las funciones de aplicación de la capa de aplicación para controlar el software de aplicación. Este objeto no tiene un formato definido, el calificador libre de formato de la capa de aplicación se usa para identificar la aplicación en cuestión.

#### **4.6.2.3.9      Objeto de numeración alternada**

El objeto de punto flotante tanto el corto, como el largo y el extendido; representan un valor calculado o medición científica, el campo de valor mantiene el número de punto flotante actual, el campo de bandera mantiene la información acerca del punto, el campo de las unidades determina las unidades del campo de valor. Estas son unidades científicas o de ingeniería asociadas con la cantidad calculada o medida.

El código binario decimal de corto empaquetado (small- packed binary coded decimal), de empaquetado medio (medium packed) y largo empaquetado (large packed), son objetos de información que representan una secuencia de dígitos BCD. Cada dígito BCD puede representar una variedad de información, desde las salidas de control hasta entradas análogas.

Tabla XI. Valores definidos para el campo de las unidades

0	Voltios (pico-pico)	10	Hz (Frecuencia en ciclos por segundo)
1	Amperios (pico-pico)	11	w (Frecuencia en Radianes)
2	Voltios (RMS)	12	C (Grados celsius)
3	Amperios (RMS)	13	F (Grados fahrenheit)
4	Kw (Potencia real)	14	K (Grados kelvin)
5	KVA (Voltios-Amperios totales)	15	N (Fuerza en newtons)
6	KVAR (Potencia Reactiva)	16	Kg (Masa en kilogramos)
7	KwH (Kilowatt-Hora)	17	m/s <sup>2</sup> (Aceleración)
8	KVARH (KiloVAR-Hora)	18	N/m <sup>2</sup> (Presión)
9	PF (Factor de Potencia)		
19	N*m(Torque)		

## **5. CRITERIOS E INTERPRETACIONES DE LAS CAPAS DEL PROTOCOLO APLICADO A MEDICIONES Y MONITOREO DE UNIDADES TERMINALES REMOTAS.**

Como se ha mostrado anteriormente, estructuralmente, el envío de bloques de datos, conocidos como PDUs desde un punto a otro punto a través de una red, especifican el formato del protocolo de control de aplicación (PCI) que forma el inicio del PDU y los servicios provistos por la aplicación (SDU). Los APDUs intercambiados tienen dos partes, el inicio APCI y el ASDU. Los códigos de función (fijados en el APCI), los objetos y calificadores (fijados en el ASDU) describen el conjunto mínimo que los dispositivos deben analizar para implementar un subconjunto particular de requerimientos DNP en una estación maestra o esclava a manera de satisfacerlos, y se dice que se implementa un nivel (level) del protocolo, el término nivel es escogido para no entrar en conflicto con las clases de datos o capas del modelo OSI. Si al dispositivo se implementa el protocolo de la capa de aplicación DNP3 en el nivel uno, el dispositivo tiene la implementación DNP L-1.

En los sistemas de naturaleza jerárquica se constituyen de dispositivos que pueden ser provistos de múltiples fabricantes. La implementación de niveles de subconjuntos y la comunicación de datos, usando los estándares de datos objeto DNP para varias funciones, ofrece una alternativa para que diferentes fabricantes de dispositivos sean inter-operables.

La definición de subconjuntos (subset), o niveles para la capa de aplicación DNP3 se usa como referencia para determinar compatibilidad entre implementaciones de la capa de aplicación DNP3, y los términos para este conjunto describen que ya se ha

implementado o qué se requiere implementar. El subconjunto de protocolo describe un nivel de implementación que contiene tablas con los siguientes campos:

- \* Objeto: Este campo describe un objeto de la capa de aplicación, el grupo de objeto al que pertenece, la variación y la descripción.
- \* Requerimiento: Este campo describe el conjunto de objetos, códigos de función y calificadores que un dispositivo esclavo debe ser capaz de analizar como parte de un requerimiento entrante.
- \* Respuesta: Este campo describe el conjunto de objetos, códigos de función, y calificadores que un dispositivo maestro debe ser capaz de analizar de una respuesta o respuesta sin solicitud.

El subconjunto describe el mínimo de objetos, códigos de función, y calificadores que los dispositivos deben analizar a manera de implementar el subconjunto. Hay tres niveles de implementación de DNP3 definidos. Cada nivel se concluye sobre el nivel precedente.

Los subconjuntos fueron considerados para cumplir con las siguientes características: Minimizar la complejidad de implementación, minimizar el ancho de banda usado, permitir que todos los datos proporcionados por un esclavo sean recuperados en cualquier nivel, permitir la flexibilidad de diferentes implementaciones pero que sean inter-operables. Los calificadores son una parte compleja de DNP, así el número de calificadores requeridos para ser soportados tanto por dispositivos maestros y esclavos, está limitado.

## **5.1 Niveles de Implementación**

### **5.1.1 Implementación de Nivel 1**

Este nivel de implementación DNP3 representa la más simple comunicación de un maestro y un dispositivo electrónico inteligente (IED). Este caso podría representar una estación maestra o concentrador de datos y un dispositivo final (ejemplo: medidor, relevador, auto-recloser o un controlador de un banco de capacitores).

El subconjunto de nivel uno se basa aplicando polling en la clase de datos. Un esclavo de nivel uno acepta solicitudes de:

- \* Lecturas de objetos de clase de datos.
- \* Lecturas de objetos de salidas binarias y salidas análogas.
- \* Operaciones de control para salidas binarias o salidas análogas.
- \* Escritura a indicaciones internas de reinicio (restart).
- \* Reinicio en frío.
- \* Medidas de retraso y escritura de tiempo y fecha.

El maestro acepta el subconjunto de variaciones de objeto alrededor de un tercio del total del número definido para DNP, que incluye tipos de datos básicos tales como: Entradas binarias y eventos, contadores y contadores de eventos, entradas análogas y eventos, salidas de estado análogas y binarias.

### **5.1.2 Implementación de Nivel 2**

Contiene algunas características más que la implementación DNP3 de nivel uno. Orientada a comunicaciones entre estaciones maestras, concentradores y pequeñas unidades remotas terminales. Típicamente, los puntos de entrada y salida de dichos dispositivos podrían ser locales al dispositivo. Una implementación de esclavo de nivel

dos es la misma que el nivel uno con las siguientes adicionales: Un esclavo de nivel dos acepta solicitudes congeladas sobre contadores binarios, analiza solicitudes de lecturas para “variación 0” de objetos específicos. En el nivel dos se analizan las solicitudes para variaciones uno, dos y tres de cambios de objetos de entradas binarias, analizan solicitudes de contadores congelados (pero no contadores delta congelados), puede enviar respuestas sin solicitud conteniendo datos estáticos.

### **5.1.3 Implementación del Nivel 3**

Este nivel de implementación es para comunicación entre un maestro y un dispositivo esclavo (RTU). El nivel tres usa más rangos de objetos, variaciones, códigos de función y calificadores que el nivel dos. Para esta implementación, el esclavo procesa requerimientos de lectura para varios objetos específicos y variaciones, también procesa más rangos de requerimientos con más códigos de funciones. Soporta la habilitación y deshabilitación de respuestas sin solicitud para las clases de datos excepto la clase cero, soporta también la asignación y reasignación de datos-objeto a clases dinámicamente (Código de función 22).

## **5.2 REQUERIMIENTOS DE USUARIO**

Comúnmente se han implementado protocolos de comunicaciones con derechos de propiedad del mismo fabricante, pero con protocolos abiertos se gana independencia de soluciones específicas de fabricantes específicos.

Un protocolo abierto para ser usado debe reunir una base de requerimientos de usuario (tal es el caso para DNP3). Las características de DNP3 son usadas para reunir dichos requerimientos específicos, para una implementación. A continuación se describe cómo un usuario puede incorporarse sobre la base DNP3 de estaciones remotas en

sistemas SCADA; con los siguientes requerimientos de usuario descritas como características del dispositivo involucrado:

- \* La habilidad de una estación remota para muestrear valores de mediciones en intervalos periódicos, almacenarlos (con estampa de tiempo) por una cantidad razonable de tiempo, y cuando también reporte todos los valores al maestro (SCADA) y en un tiempo después o esperar hasta que el maestro solicite los valores. Este requerimiento es referido como direccionamiento (trending).
- \* La habilidad de una estación remota para mantener una lista de eventos no críticos, que no causen reportes no solicitados inmediatos de la estación remota. Esto es referido como eventos (eventing).
- \* La habilidad de una estación remota de realizar cálculos locales y funciones lógicas. Éstas son definidas por el usuario, preferentemente usando las características de las estaciones remotas que sean compatibles con estándares (Controladores programables, lenguajes de programación).
- \* La habilidad del maestro SCADA para enviar parámetros de configuración a la estación remota.

Una implementación de requerimientos de usuario puede usar un número básico de las características del protocolo, que a continuación se mencionan:

- \* Clases de evento: el protocolo implementa tres niveles de clases (referidas como clase uno, clase dos, y clase tres). Los puntos de datos en una estación remota pueden ser configurados para generar eventos cuando los puntos de datos cambian de valor, estos eventos presentes son configurables con la asignación a cualquiera de las tres clases. El protocolo habilita a la estación maestra la habilidad de leer

eventos de clases individuales, o leer eventos con cualquier combinación de clases con solicitudes de lectura.

- \* Recuperación de datos: un uso eficiente del ancho de banda es indispensable para la mayoría de compañías porque todavía es completamente limitada, DNP3 ofrece varias formas de operación en las remotas:
  - Operación fija, en la cual el maestro no usa “polling” en cualquier esclavo, y todas las comunicaciones están en reporte por excepción sin solicitud (RBE: report-by-exception).
  - Operación de reporte por excepción sin solicitud, en la cual la mayoría de las comunicaciones es sin solicitud, pero el maestro ocasionalmente usa polling para datos de clase cero ó para verificar si su base de datos está actualizada.
  - Operación de reporte polling por excepción sin solicitud, en la cual el maestro realiza polling frecuentemente para datos evento y ocasionalmente para datos clase cero. Muchas veces la respuesta del esclavo al polling de evento contiene pocos datos, así el polling puede ser bastante rápido.
  - Operación de polling estático, en el cual el maestro hace polling solamente para datos de clase cero o a los datos específicos requeridos. Este método es el más simple para implementar, pero puede ser bastante ineficiente si el número de puntos que se necesitan recuperar es alto y los cambios son infrecuentes.
  
- \* Variaciones de datos objeto. DNP3 permite datos de varios tipos, conocidos en los grupos de objetos para ser reportados en varios formatos, conocidas como variaciones, por ejemplo: el cambio de una medida análoga puede ser enviado con una resolución de enteros de 16 ó 32 bits o como un valor de punto flotante. Estos valores pueden ser reportados con o sin la medida de tiempo. Algunas variaciones son comúnmente usadas, y están especificadas como variaciones requeridas para cumplir a la implementación de los niveles de subconjuntos, y aunque no todos los

equipos DNP3 los soportan, se verifica por la referencia de la documentación del perfil de dispositivo DNP3.

### **5.3 DIRECCIONAMIENTO**

El direccionamiento (trending) es usado por un software de aplicación en la estación remota que periódicamente genera “cambios de eventos de entradas análogas” (usando variaciones con tiempo), así, el usuario configura localmente cuál de las entradas de datos de la base análoga de la estación remota está para ser direccionada y el intervalo de generación de evento (configurados como eventos de clase tres). Estas variaciones de eventos no son obligatorias en el subconjunto de definiciones, pero comúnmente en los sistemas SCADA se analizan y procesan eventos.

El direccionamiento de eventos es controlado por el tiempo de flujo actual de los eventos (direccionados) como una cola de eventos por la habilitación y deshabilitación de reportes sin solicitud de la clase de eventos. Los modos de direccionamiento son:

- (a) Direccionamiento de recuperación sin solicitud, por día. Cuando el usuario requiere este modo, la remota está configurada con el tiempo del día en que los eventos direccionados están para ser enviados al Maestro (como ejemplo de la aplicación se puede decir que se refiere a un maestro SCADA), en el tiempo configurado del día la remota ubica los eventos direccionados de clase tres reportándolos en la cola. El maestro debe previamente tener disponible el reporte sin solicitud para eventos clase tres.
- (b) Periódicamente solicitados por el Maestro SCADA: Si un usuario requiere este modo, la remota está configurada para ubicar los eventos direccionados en la cola de reporte de clase tres en intervalo de tiempo regular (típicamente 15 minutos). Sin embargo, el maestro SCADA debe previamente asegurar que el reporte, sin solicitud de eventos de clase tres está deshabilitado (y por defecto en la remota

reiniciar). Los eventos direccionados están ahora disponibles en colas de eventos de clase tres para que el maestro pueda hacer una lectura en los momentos libres.

- (c) Lectura de requerimientos en demanda desde el maestro: Este modo de operación es muy similar a el modo (b), excepto que en este caso el maestro SCADA ha sido disparado por un evento del sistema (el cual puede ser que no esté totalmente relacionado con los puntos de datos en su remota) y llegar hasta la remota y leer los eventos direccionados que estén disponibles. Este modo usa la configuración en la remota como para el modo (b).

## **5.4 EVENTOS**

### **5.4.1 Clase de datos y eventos**

Cuando un dispositivo esclavo reporta objetos evento, están en clase uno, clase dos, clase tres para el objeto evento (de los puntos de datos generados). La clase de dato evento es, sin embargo, un atributo del punto de dato y los objetos evento del punto de dato reportado. Cada objeto evento reportado por el dispositivo esclavo, puede pertenecer a Clase uno, Clase dos o Clase tres, cada objeto estático reportado por el dispositivo esclavo pertenece a la clase cero.

Para el nivel tres en los dispositivos esclavos se permite que los maestros habiliten o deshabiliten el reporte de objetos evento, haciendo solicitudes de la asignación de clase (código de función 22). Para habilitar el reporte de objetos evento para un punto de dato en particular, el maestro especifica en su solicitud una clase uno, dos o tres del encabezado de objeto y el encabezado de un objeto estático para el apropiado punto de dato. Un dispositivo maestro dice al dispositivo esclavo que pare de reportar objetos eventos para un punto de dato particular, por la asignación de objetos evento del punto de dato para clase cero. Entonces la clase cero es la clase de datos

estáticos y los objetos evento no pueden ser clase cero, el esclavo no produce ningún objeto evento para el punto de dato.

#### **5.4.2 Configuración de eventos**

Es la configuración de eventos generados (eventing) desde las entradas binarias asociado a estados de condiciones no críticas (también entradas reales o seudo entradas de programa de usuario) para la clase dos. El maestro SCADA se asegura que el reporte no solicitado está deshabilitado para la clase dos, y entonces los eventos no son reportados hasta que son solicitados por el maestro (típicamente una vez por día, o en demanda si el maestro SCADA ve un evento de sistema que dispara una lectura no solicitada).

### **5.5 ALARMAS**

Las alarmas son realizadas por configuración de usuario, de eventos generados desde las entradas binarias asociadas con condiciones críticas de alarma (también de entradas reales o puntos derivados del programa de usuario) a la clase uno. El maestro SCADA se asegura que el reporte no solicitado es habilitado para la clase uno. Entonces los eventos-alarma son reportados en respuestas no solicitadas al maestro, tan pronto como ellos ocurren.

### **5.6 CONFIGURACIÓN REMOTA**

#### **5.6.1 Respuestas de error**

Como resultado de la definición de niveles de implementación, se hace necesario definir las repuestas válidas a solicitudes en donde ocurre lo siguiente:

- \* No es una solicitud válida para el nivel o,

\* Es para el nivel dado, pero no es aplicable al dispositivo en particular.

Un dispositivo esclavo puede responder a una solicitud del maestro como un error en el requerimiento.

Tabla XII. Mensajes de Error

Respuesta	IIN bit	Significado
Función desconocida	0	El nivel de implementación no soporta esta función, sobre objetos de este grupo y variación
Objeto desconocido	1	El nivel de implementación no soporta este grupo y variación del objeto.
Error de parámetro	2	Los objetos que están de este grupo y variación, son válidos pero no el rango de los puntos especificados
Respuesta nula	r/n	No está especificado el rango de punto y no hay objetos para este tipo. Primeramente usado en respuesta de polling datos eventos.

### 5.6.2 Perfil de dispositivo

Los fabricantes y vendedores deben proporcionar un documento del perfil de dispositivo para cada uno que ellos construyan, implementando DNP3. El documento del perfil de dispositivo claramente identifica cualquier desviación de los subconjuntos de implementación y cualquier otro problema que pueda surgir cuando se determina la compatibilidad del dispositivo con otro. El documento de perfil de dispositivo es dirigido a la capa de aplicación y la capa de control de enlace solamente, la capa física no está dirigida por las diferentes capas y configuraciones. La información mínima que un vendedor debe incluir en el documento de perfil de dispositivo es la siguiente:

- Tabla de implementación: Identifica qué variaciones de objeto y códigos de función y calificadores soporta el dispositivo, tanto en solicitudes como en respuestas.
- Lista de puntos: Esta es una parte opcional del documento, incluye una lista de los puntos de datos provistos por el dispositivo.
- Nombre del vendedor: El nombre de la organización que produce el dispositivo.
- Nombre del dispositivo: El modelo y el nombre del dispositivo, suficiente para distinguirlo de cualquier otro dispositivo de la misma fábrica.
- El nivel DNP más alto soportado: El número más alto de la implementación DNP del subconjunto, al cual el dispositivo cumple enteramente.
- Función del dispositivo: Tanto para el maestro como el esclavo, el maestro envía requerimientos (o solicitudes), mientras que el esclavo envía respuestas. Sin un dispositivo, físicamente realiza ambas funciones, se provee de un documento de perfil de dispositivo separado para cada función.
- Tamaño máximo de trama de enlace de datos: El número mayor de la trama del enlace de datos que transmite el dispositivo, en octetos.
- Tamaño máximo de fragmento: El mayor número de fragmentos de la capa de aplicación que el dispositivo transmite en octetos.
- Enlace de datos máximo: El número de veces que el dispositivo retransmitirá un dato confirmado.
- Reintentos: En el enlace de trama antes de abortar la transacción.
- Reintentos máximos de aplicación: El número de veces que el dispositivo transmitirá una solicitud o respuesta confirmada de aplicación, antes de abortar la transacción.
- Confirmación de aplicación requerida: Si el dispositivo siempre envía mensajes de capa de aplicación con el bit de confirmación, puesto en el campo de control de aplicación.
- Tiempos fuera: Las condiciones de tiempo, bajo la cuales el dispositivo cancela una trama entrante, fragmento, solicitudes o respuestas.

- Operaciones de control: Las operaciones de control permitidas por el dispositivo. Un dispositivo puede soportar una salida análoga o salida de bloque de control de relevador, pero no necesariamente todas las operaciones sobre ese objeto.
- Reportes de cambios de evento de entrada binaria: Una descripción de qué variaciones de objetos reporta un dispositivo esclavo, cuando es preguntado por cambios de evento de entrada binaria con variación cero, o cuando es preguntado por la clase de dato.
- Envío de respuestas sin solicitud: Una lista de las condiciones bajo las cuales un dispositivo esclavo envía respuestas sin solicitud.
- Envío de datos estáticos sin solicitud: Una lista de condiciones bajo las cuales un dispositivo esclavo envía datos estáticos (Clase cero) como respuestas sin solicitud.
- Contador de Objeto/Variación por defecto: El objeto y la variación que un dispositivo esclavo usa para reportar, del contador de objetos, cuando es preguntado por la variación cero o por la clase dato.
- Contador en roll over: El valor en el cual los contadores de dispositivo esclavo están fuera de rango (roll over).
- Envío de respuesta de múltiples fragmentos: Si o no, el dispositivo esclavo envía respuestas de múltiples fragmentos (Los dispositivos maestros no envían requerimientos de múltiples fragmentos).

### **5.6.3 Archivos de configuración**

Una configuración remota se define parcialmente para la aplicación, el requerimiento de programa de usuario para cargar remotamente, puede ser satisfecho por el uso del objeto control de archivos (obj. 70). Sin embargo, el envío limitado de un conjunto de valores de configuración a la remota desde el maestro son usados en la misma según el programa definido localmente. El maestro escribe valores para puntos

de salida en el tiempo de reinicio de la RTU. El programa de usuario usa los valores en las salidas para el cómputo.

#### **5.6.4 Consideraciones del Dispositivo Esclavo**

Para el inicio de un dispositivo esclavo, debe ajustarse a verdadero (1) el bit de indicación interna de Reinicio de Dispositivo junto con el campo de indicación interna de cada subsiguiente respuesta y respuesta sin solicitud. Este bit debe permanecer puesto hasta que el dispositivo maestro lo limpia con una solicitud de escritura.

Si el dispositivo esclavo es configurado para enviar respuestas sin solicitud, este debe inmediatamente enviar una respuesta sin solicitud de inicio. Esta respuesta no solicitada inicial debe contener: el encabezado de respuesta solamente, conteniendo el control de acceso, código de función y los cambios de indicaciones internas, y el encabezado de respuesta más el estado actual de todos sus puntos estáticos (ejemplo: entradas binarias, entradas análogas).

La respuesta sin solicitud inicial puede también contener cualquier objeto evento que el dispositivo esclavo ha generado mientras estaba previamente ejecutándose, pero no tiene el reporte para el maestro. Si tales objetos existen, el esclavo debe reportarlos antes de los datos estáticos. Esta regla permite al maestro reportar cambios de datos en orden cronológico por el simple análisis del mensaje desde el inicio al final. El fabricante del dispositivo debe especificar en el documento de perfil de dispositivo, si el dispositivo envía respuestas sin solicitud, y si lo hace, que incluya los datos estáticos en la respuesta sin solicitud inicial.

Un dispositivo esclavo puede opcionalmente enviar respuestas sin solicitud al dispositivo maestro, el fabricante del dispositivo esclavo debe especificar en el documento de perfil de dispositivo, si el dispositivo puede enviar respuestas sin

solicitud. Este documento también debe especificar cuáles son los objetos, variaciones, calificadores y códigos de función que el dispositivo esclavo puede incluir en las respuestas sin solicitud. Además los dispositivos esclavos deben proveer un modo de operación en la cual el esclavo nunca envía respuestas sin solicitud, y debe ser configurable. Para la Operación de Salidas Binarias todos los dispositivos esclavos deben ser capaces para analizar operaciones sobre objetos de salida de bloque de control de relevador para la operación de salidas binarias. Si un dispositivo esclavo no tiene cualquier punto de salida binaria, este puede responder con las indicaciones internas puestas a Error de parámetro u Objeto Desconocido.

Un dispositivo maestro no debe enviar requerimientos de múltiples tramas, un dispositivo maestro debe aceptar respuestas de múltiples tramas, pero un dispositivo esclavo debe ser capaz de retornar todos sus datos (estáticos y eventos) junto con la respuesta.

Los dispositivos esclavos de todos los niveles deben ser capaces de analizar una solicitud de maestro conteniendo cualquier subconjunto de todos los objetos que el dispositivo esclavo soporta. Cuando hay múltiples objetos en una respuesta todos los dispositivos maestros deben ser capaces de analizar una respuesta de esclavo o respuesta sin solicitud conteniendo un subconjunto de objetos que el dispositivo maestro soporta.

En las confirmaciones y reintentos, todos los niveles de dispositivos esclavos y maestro puede escoger cuando, ellos solicitan confirmación de tramas salientes del enlace de datos. Un dispositivo maestro o esclavo debe transmitir una trama de confirmación al enlace de datos, si ésta es solicitada para hacerlo así por un código de función de envió/confirmación esperada de trama entrante. Un dispositivo esclavo o maestro debe transmitir una respuesta de confirmación de capa de aplicación si es solicitada por medio del bit de confirmación en un fragmento entrante.

Las variaciones con o sin bandera de un dispositivo esclavo se pueden escoger para retornar un objeto conteniendo una bandera (ejemplo: Entrada análoga de 16 bits), cuando el maestro ha solicitado una variación de objeto sin bandera (ejemplo: Entrada análoga de 16 bits sin bandera). Similarmente, un esclavo puede escoger retornar una variación de objeto sin bandera cuando el maestro lo ha solicitado. Si o no, el dispositivo responde con un objeto conteniendo una bandera, es la decisión del dispositivo.

Las operaciones de congelado, intervienen varios códigos de función de congelado. Sólo recibiendo una solicitud congelada, un dispositivo esclavo debe copiar los valores apropiados del contador binario (los contadores corriendo, es decir efectuando el conteo) en un buffer "congelado". El maestro puede leer estos valores congelados como objetos de contadores de entrada congelados. Si una solicitud o requerimiento de congelar y limpiar está siendo procesada, el esclavo debe limpiar los valores de "conteo" inmediatamente después de realizar el congelado. Una solicitud de lectura de objetos de contadores de entrada binaria, siempre retorna el valor de los contadores corriendo.

### **5.6.5 Funciones de aceptación**

Son las funciones que un dispositivo maestro o esclavo deben soportar a manera de cumplir un nivel de implementación definido. El subconjunto de definiciones trata primeramente con la capa de aplicación del protocolo DNP3. Éstos es sin embargo, un requerimiento para un dispositivo a manera de cumplir con un nivel de implementación, éste debe tener implementado los servicios de la capa de datos de enlace y funciones de transporte suficientes para soportar un nivel de implementación. Por ejemplo, todos los subconjuntos requieren que un dispositivo soporte el uso de polling de datos clase cero; si la respuesta de tal polling podría exceder 249 octetos, la segmentación por la función de transporte debe ser soportada.

#### Dispositivos Esclavos:

La manera para la implementación de un nivel particular X de DNP3, el dispositivo debe cumplir con lo siguiente:

- El dispositivo esclavo debe ser capaz de analizar todos los requerimientos del maestro definidos para el nivel X.
- El dispositivo esclavo debe ser configurable a no transmitir cualquier cosa que no sean repuestas del nivel X para solicitudes del nivel X.
- El dispositivo esclavo debe obedecer todas las reglas de implementación definidas como las mencionadas en estas secciones.
- El fabricante debe describir la implementación DNP del dispositivo en un perfil del dispositivo y hacer este documento disponible a los usuarios del dispositivo. Un dispositivo esclavo puede aceptar y responder a solicitudes adicionales no definidas en el nivel X y aún cumplir con el nivel. Este puede responder a dichas solicitudes con datos no definidos en el nivel X.

#### Dispositivos maestros:

Para implementar un nivel particular X de DNP3, el dispositivo debe cumplir con lo siguiente:

- El dispositivo maestro debe se capaz de analizar todas las repuesta del esclavo definidas para el nivel X.
- El dispositivo maestro debe ser configurable para limitar los requerimientos enviados a los dispositivos esclavos con niveles de implementación inferiores que el nivel X. Por ejemplo, un maestro en el nivel tres debe ser configurable, así éste no envía solicitudes tanto a un esclavo de nivel uno como a un esclavo de nivel dos que ellos no pueden analizar, esto no impide al maestro el envió de requerimientos de nivel tres para un esclavo de nivel tres.
- El dispositivo maestro debe obedecer todas las reglas de implementación definidas en las secciones anteriores.

- El fabricante del dispositivo maestro debe describir la implementación DNP del dispositivo en un documento del perfil del dispositivo y tener este documento disponible a los usuarios del dispositivo.

## **5.7 Notas de aplicación**

Los requerimientos de usuarios son desarrollados utilizando las características de DNP3 para cumplir metas específicas de los usuarios, los que pueden también ser provistos para el beneficio de otros usuarios, sin embargo, estas notas de aplicación no contienen todos los detalles de una implementación, pero no por eso menos importante, esta aproximación es una forma del uso de DNP3 para cumplir con los requerimientos de usuarios en las áreas de mediciones y monitoreo de parámetros eléctricos y los sucesos que en éstas se pueden observar a través de dispositivos finales, que están ubicados estratégicamente en la red eléctrica nacional con el propósito de reunir el caudal de información, que facilite a los usuarios ya sean estos los distribuidores, generadores y transportistas de energía eléctrica las revisiones de sus parámetros eléctricos. El sistema SCADA, así como los usuarios y los fabricantes solamente se han citado como referencias, porque se consideran fuera del propósito que pretende cumplir este documento.

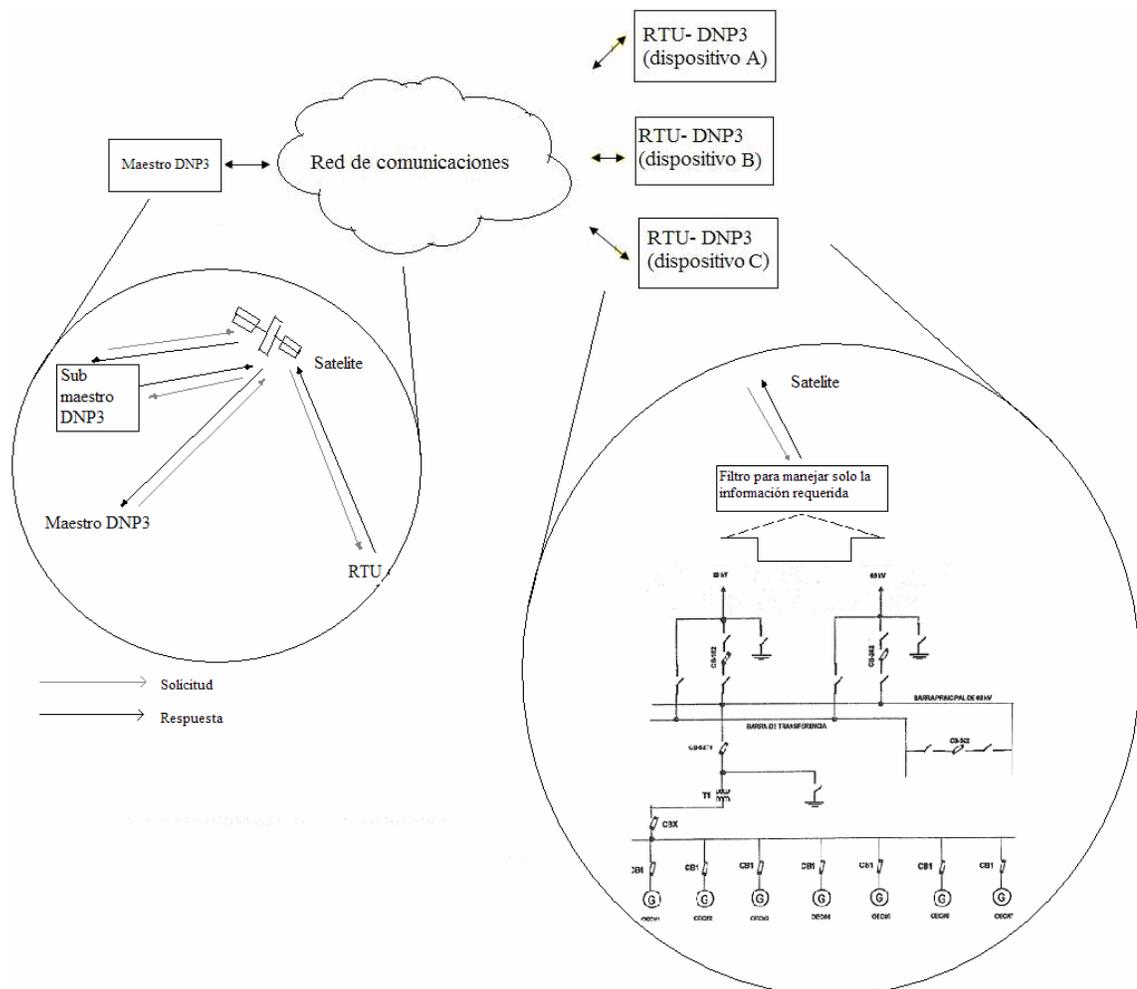
A lo largo de los anteriores capítulos se han reunido las características del protocolo proporcionando los fundamentos de su estructura. Definiendo los distintos conceptos y modelos de intercambio de parámetros de control que perfectamente encajan en un esquema del entorno energético nacional, es decir, se proyecta como una aplicación del reporte de datos, manejados en el protocolo como datos objeto pero que reflejan cantidades reales, como por ejemplo: en las entidades generadoras de energía eléctrica que a través de sus unidades terminales remotas (RTU), evalúan sus estadísticas de los datos provistos por sus instrumentos (Registradores de Eventos), ya sea para el reporte de sus parámetros como el cambio de éstos, por medio de un sistema de

comunicación redundante, que se puede constituir de una computadora que actúa como estación maestra y una RTU, y completar las pruebas de los comandos de solicitudes y respuestas dentro del mapeo de objetos, variaciones, clases y calificadores que comprenden el nivel tres de implementación mencionado anteriormente. En los anexos se proporciona una tabla completa de objetos y sus variaciones junto con los calificadores y los códigos de función manejados, tanto por el maestro como por el esclavo, denominado esclavo debido a que dependerá del nivel implementado según la cantidad de objetos que soporte el dispositivo, aunque se interpreta que si se refiere a una RTU comprende el nivel más alto de implementación, es decir el nivel tres. El sistema de comunicación construido se muestra en la figura 33, el maestro tiene suficiente información en cualquier tiempo, considerando la comunicación de un enlace dedicado, representado en la nube de transporte, sin embargo, se puede tratar de un enlace de última milla de un anillo de fibra óptica en un transporte SDH, un ambiente dial-up a través de una red pública, o a través de un radio enlace. En el sistema de comunicaciones, mostrado generalmente, el maestro puede realizar comandos de solicitud y respuestas a varias unidades, esto significa que para el protocolo se visualizan fronteras o interfaces, básicamente éstas se comprenden entre los equipos de medición y los de transmisión y recepción (red de comunicaciones). Esta flexibilidad de transporte de paquetes de datos sobre terceros que proporcionan los servicios de transporte ofrece una gran rentabilidad en cuanto a las opciones de acceso de los puntos que se necesitan comunicar debido a la gran expansión de éstos a nivel nacional que participan activamente en las demandas de comunicación.

Esta nota de aplicación está referida al caso particular del intercambio de datos a través de solicitudes desde el maestro (ubicado en una entidad con centrales de operaciones) y el reporte desde una RTU (ubicada en una empresa agroindustrial). La comunicación entre el maestro y la RTU es a través de un enlace satelital, que implica un canal de comunicación con un ancho de banda que advierte un número considerado de datos solicitados debido a la saturación del ancho de banda en momentos de muchas

variaciones de los parámetros de monitoreo y medición, es decir, que para que no exista saturación se debe enviar sólo la información requerida por el maestro, para evitar congestión en la red (debido a que podría ser el caso de tener varios esclavos a disposición para solicitudes y respuestas), por lo tanto, se debe definir un mapeo de puntos, es decir qué información es estrictamente la que se requiere. Con el mapa de puntos se programa en el esclavo el envío de estos, de lo contrario el esclavo podría reportar puntos de datos adicionales, aunque estos son también parte de la información, se considera innecesaria para evitar la saturación de los reportes y el mapa de puntos prácticamente establece que sea el menor número de paquetes de datos solicitados.

Figura 33. Sistema de comunicación.



En la figura 33, el submaestro DNP3 supervisa qué datos están siendo solicitados y cuáles son los que el esclavo está reportando, esto con el propósito de analizar lo que está pasando en la red entre esclavos y medidores. Los datos de medición pueden estar filtrados, así sean los puntos de datos estrictamente requeridos. El protocolo es una implementación dedicada a la adquisición de datos, aunque prácticamente la comunicación entre el esclavo, submaestro y maestro contribuye además con otras funciones de maniobras sobre la red entre esclavos y medidores.

Cuando una aplicación para un sistema de comunicaciones se pone en funcionamiento, le preceden en la mayoría de casos una serie de pruebas (en el caso que el proveedor de servicio DNP3 sea diferente) debido a los problemas originados de la interconexión, derivado de la interacción de dispositivos de varios fabricantes lo que implica muchas veces realizar pruebas en el punto de recolección de información (antes que se haga efectiva su conexión y funcionamiento al sistema de comunicación), así establecer:

- La comunicación entre los dispositivos.
- El tipo de información requerida (datos análogos, datos digitales).
- Cantidad de datos (mapa de objetos, variación, clases)

El programa de aplicación para las pruebas es el ASE2000 para DNP3, que en general implica la configuración de los puertos de comunicación de los dispositivos (como se ha mencionado, se refiere a un par de dispositivos DNP3, por así decir, que soportan el protocolo), las solicitudes de datos digitales y datos análogos, además de las clases datos (sean estos ya considerados como datos de clase cero, uno y dos). Específicamente, a continuación se mencionan parámetros que intervienen en la prueba efectuada para un sistema de comunicaciones, interpretando en cuál de la capa del protocolo están situados. La prueba mediante una opción de salvar todas las rutinas de transacciones en un archivo con extensión de captura de los datos, es decir un “Log File”, es una herramienta que sin duda proporciona detalles del intercambio de

solicitudes y respuestas del protocolo para los requerimientos de los usuarios y que permite tener criterios sobre el desempeño del sistema de comunicaciones.

-Capa Física.

- \*Selección del puerto de comunicaciones tanto hacia la RTU como el maestro.

- \*El control de flujo de ASE2000 en el hardware (es decir en la línea del enlace dedicado).

- \*La velocidad configurable, es de 9600 bps, (el tiempo fuera en configurable n segundos)

- \*Tiempo de retraso en las respuestas en segundos (provocada por retrasos en el tramo del enlace)

- \*Se definen los valores de la fuente y el destino (o sea la dirección de la RTU y del maestro respectivamente para la prueba).

- \*Control de Flujo de la RTU (Tx, Rx, Gnd, CTS, RTS).

- \*Pinout RS232 (se refiere a la forma de conexión de los pines, es decir, un cable cruzado).

-Capa de enlace de datos

- \*Valores de destino y fuente de la RTU y el maestro (de hecho la misma de la capa física).

- \*Encabezado del enlace de datos (longitud de octetos).

- \*Control de enlace de datos en el maestro:

  - Reset Link, código de función cero (y confirmación)

  - Link estatus, código de función nueve (con Respuesta)

- \*Control de enlace de datos en la RTU:

  - Reset link, código de función cero

  - Link Status, código de función nueve

-Capa de transporte

- \*Encabezado de transporte

- \*Objetos y variaciones enviados

- \*Secuencia de Tramas

-Capa de aplicación

\*Se definen los objetos y sus variaciones, las clases de objeto, calificadores y Número de puntos (datos de usuario sin confirmar), como por ejemplo: el mapa de objetos mostrado en la Tabla 13 con referencia a los parámetros relevantes para una implementación.

\*Resultado de cambio de valores

\*Secuencia de Fragmento

\*Indicaciones Internas

Obj. 80 var. 1 Indicaciones internas, Calificador 00x.

Tabla XIII. Mapa de objetos y variaciones para los datos solicitados en la prueba

Object	Variation	QUAL RTU	obj var rtu	# of Points
1 binary input	1	00x	1,1	64
1 binary input	1	oox	1,1	5 to 30
30 analog input	2	00x	30,2	96
30 analog input	2	oox	30,2	4 to 26
30 analog input	1	00x	30,2	96
30 analog input	1	oox	30,2	4 to 26
30 analog input	0	00x	30,2	96
30 analog input	0	oox	30,2	4 to 26
60 class data	1 static Class0	oox di y ai	ai 30,2 di 1,2	ai48, di64
60 class data	1 static Class0	n/a	obj unknown	4 to 30
60 class data	2 binary events Class1	17x	2,2 di change	56
60 class data	2 binary events Class1	n/a	param invalid	40 to 60
60 class data	3 analog events Class2	17x	32,2 ai change	2,26,30
60 class data	3 analog events Class2	n/a	param invalid	3 to 40
60 class data	4 class1+class2 Class3	n/a	param invalid	n/a
60 class data	4 class1+class2 Class3			

Fuente: ASE

Una vista general del formato gráfico de la pantalla del programa de la prueba mediante el ASE200 se proporciona en la figura 34, con los parámetros relevantes en la prueba involucrados en el protocolo DNP3.

En esta ventana de usuario de aplicación se puede observar tanto el protocolo en la forma hexadecimal de cada solicitud y respuesta, así también el programa de prueba ofrece una vista con detalle de los principales procesos, resultado del intercambio de las transacciones de la solicitudes (en gris) y sus respectivas respuestas (en gris oscuro), además de proporcionar el tiempo transcurrido entre la solicitud y respuesta (en este caso se refiere al basado en el tiempo de la computadora que se implementó como estación maestra) que proporciona un criterio de ajuste del tiempo de retardo que pueda suceder en cuanto a la comunicación entre el maestro y la Unidad Remota Terminal. Se observa que la estación maestra tiene como valor fuente uno y la estación remota como destino 137.

En la figura 35 se muestra la ventana de la interfaz grafica que ilustra cómo los datos objeto son reportados por la remota después de una solicitud de los mismos por parte de la estación maestra según el mapa de objetos que éste designado para la estación remota en la prueba y que específicamente es una respuesta de datos clase cero para un objeto de entrada analógica (Objeto 30, variación 6, calificador 00 y un número de puntos del 1 al 18). Seguido está la parte de una respuesta de los datos de entradas binarias solicitadas (Objeto 1, variación 2, calificador 00 con un número de puntos de dato de 0 hasta 152). En la figura 35 hay un número de puntos de entrada de datos digitales mayores que el establecido por el mapa de datos, esto se puede interpretar que en un momento dado el flujo de datos reportados por la estación remota podría crear un reporte con una cantidad excesiva de datos que no necesariamente pueden considerarse como prioritarios o críticos, y que podría significar un sobre-flujo de datos sobre el sistema de comunicación que ineficientemente, utilizaría el canal de comunicación en las transacciones de respuestas que espera recibir el maestro de esta estación remota. De lo anterior, una medida correctiva tiene que ser primero ubicada en la configuración del protocolo del esclavo, derivado de que el proveedor del protocolo que puede ser de otro fabricante, el dispositivo esclavo podría tener diferencias entre las variaciones de los datos objeto que soporta, las clases de datos asignadas. Por otra parte en el maestro el

mapa de objetos y variaciones queda definida (por los requerimientos de usuario) la cantidad de datos solicitados al dispositivo esclavo. Por lo tanto aunque el protocolo DNP3 es un protocolo abierto, una desventaja es la gama de dispositivos esclavos que implican la revisión de los datos objetos soportados así como sus variaciones. Sin embargo, éste protocolo así como el programa de aplicación de comunicación, ofrecen una ventaja debido a que están situados bajo estándares internacionales, que tiene su aplicación a nivel nacional.





## CONCLUSIONES

1. La orientación del protocolo a niveles o capas, define los servicios que ofrece y da un acuerdo al formato de los mensajes, así el enlace de datos DNP3 realiza las transacciones de mensajes sin el control de capas superiores y además la red física es transparente a la aplicación usando el enlace de datos, el formato de trama para realizar las transacciones entre primaria y secundaria se vale de los bits de control y los códigos de función.
2. La capa de transporte se especifica para los mensajes que son más grandes que las unidades de datos del protocolo para formar unidades de transporte con la información de control de transporte y satisfacer los requerimientos de comunicación de la capa de red y la capa de aplicación en los puntos de acceso a servicios para completar las funciones invocadas.
3. El formato de mensaje en la aplicación; la estación maestra envía mensajes de solicitud de la capa de aplicación y la remota únicamente envía mensajes de respuesta de la capa de aplicación, y el manejo de las respuestas sin solicitud se maneja con operación de procesamiento inmediato o procesamiento después de confirmar.
4. Los datos objetos son elementos de información utilizados por la capa de aplicación que facilita a los dispositivos esclavos la capacidad de monitorear, controlar y generar diferentes datos-objetos tanto en niveles de hardware o software, así el número de objetos capaz de analizar un dispositivo esclavo determina el nivel de implementación del protocolo; la variación de un dato objeto es una representación del mismo punto de dato pero especifica el tamaño

del dato objeto y si tiene bandera de información, el rango define el punto de dato específico en un conjunto de datos objetos del mismo tipo, el calificador complementa el uso del rango.

5. Los estándares de los datos objetos ofrecen una alternativa para diferentes fabricantes de dispositivos que soportan DNP3 y los subconjuntos de objetos y variaciones minimizan la complejidad de implementación, la utilización del ancho de banda y la recuperación de los datos en cualquier nivel de implementación.

## RECOMENDACIONES

1. El tamaño máximo de los fragmentos o APDU`s debe ser configurable en ambos dispositivos esclavos y maestros, así también elegir en los dispositivos maestros los calificadores y objetos en sus solicitudes y así minimizar la cantidad de ancho de banda requerido; para mantener el estándar los dispositivos no deben transmitir fragmentos largos de la capa de aplicación mayor que 2048 bytes, además, todos los dispositivos deben aceptar fragmentos de la capa de aplicación de 249 bytes o menores, el formato de trama FT3 se utiliza para una interfase física serial y una transmisión y recepción full duplex con conexión directa evitando colisiones de mensajes
2. La habilitación o deshabilitación de la confirmación del enlace de datos debe ser configurable en ambos dispositivos esclavo y maestro, porque el uso de las tramas de confirmación del enlace de datos, sobre un medio físico punto a punto, puede reducir a la mitad la comunicación en la velocidad de la línea.
3. Los dispositivos maestros deben proveer una operación de reporte por excepción para que los dispositivos esclavos reporten los datos eventos, para ciertos objetos evento, aunque éste no necesariamente los utilice o que el dispositivo esclavo necesariamente los reporte.
4. El uso de polling para todos los datos evento y datos estáticos por el maestro hacia un dispositivo que no reporta automáticamente todos sus datos estáticos en una respuesta sin solicitud inicial, se realiza con una solicitud de lectura para todos los datos de clase 1, clase 2, clase 3 y clase 0.



## BIBLIOGRAFÍA

1. Comunicación de datos. <www.cybercursos.net > marzo 2005.
2. DNP users group. <www.dnp.org > marzo 2005.
3. Glenn Berg **MCSE Training Guide: Networking Essentials** Segunda Edición. New Riders 1998.
4. IEC. <www.iec.ch> junio 2005.
5. Kusum Rafiquzzaman **Fundamentals of Digital Logic and Microcomputer Design**. Tercera Edición. Rafe Systems Inc. 2000.
6. Lavares Girón, Jorge Fernando. **Programación del puerto serial de la PC, integrando lenguajes de alto nivel con módulos en lenguaje ensamblador para comunicación de computadoras**. Tesis Ing. Electrónica. Guatemala, universidad de San Carlos de Guatemala. Facultad de Ingeniería. 1998.
7. Mischa Schwartz **Redes de telecomunicaciones protocolos, modelado y análisis**. Addison-Wesley 1994.
8. Rodríguez Rivas, Roberto René. **Recomendaciones para la estandarización de protocolos en la interconexión de redes de datos en Guatemala**. Tesis Ing. Eléctrica. Guatemala, Universidad de San Carlos de Guatemala. Facultad de Ingeniería. 1994.
9. Santos, Mario. **Protocolos y arquitecturas en redes de computadoras**. Tesis Ing. Eléctrica. Guatemala, Universidad de San Carlos de Guatemala. Facultad de Ingeniería. 1982.



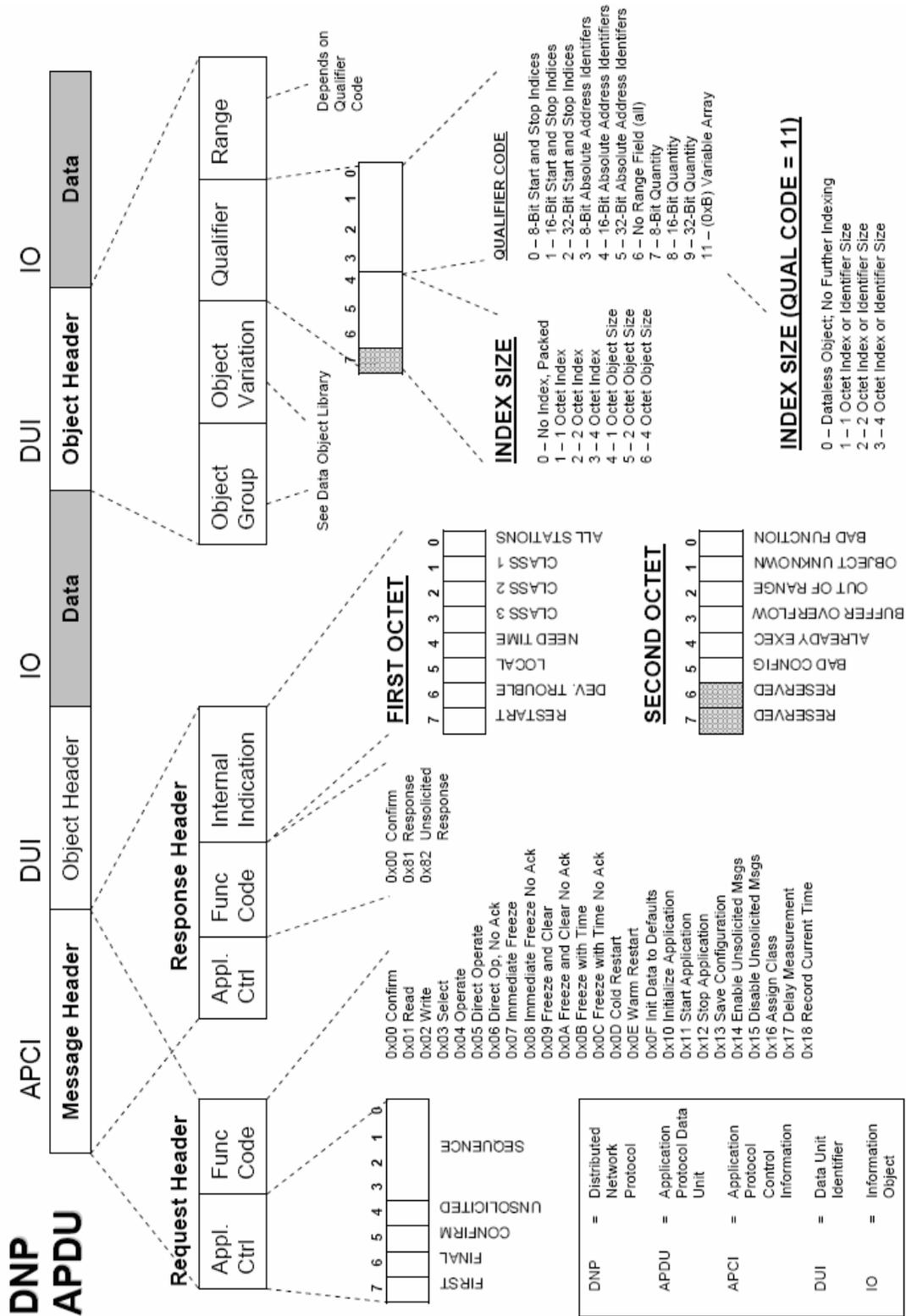
## ANEXO

Figura 34. Referencia para el control de enlace de datos DNP, lista válida de campos de control en valores hexadecimales

From Remote	From Master	Function	Other
00	80	ACK	
01	81	NACK	Failed transaction
0B	8B	Link Status Reply	No flow Control
10	90	ACK	Frame accepted, no more buffers left
11	91	NACK	Frame not accepted, no buffers available
1B	9B	Link Status Reply	No buffers available
40	C0	RESET LINK	
41	C1	Reset User Process	
44	C4	Unconfirmed User Data	
49	C9	Link Status Request	
52	D2	Test Link	FCB = 0
53	D3	Confirmed User Data	FCB = 0
60	E0	RESET LINK	FCB = 1 ignored
61	E1	Reset User Process	FCB = 1 ignored
64	E4	Unconfirmed User Data	FCB = 1 ignored
69	E9	Link Status Request	FCB = 1 ignored
72	F2	Test Link	FCB = 1
73	F3	Confirmed User Data	FCB = 1

Fuente: Páginas de referencia, procedimiento de certificación DNP.

Figura 35. Vista general de un APDU en el contexto DNP



Fuente: Procedimiento de Certificación DNP

Tabla XIV. Subconjuntos de Niveles de Implementación DNP  
Fuente: Subnet Solutions Inc.

DNP OBJECT & VARIATION			REQUEST (master may issue and slave must parse)		RESPONSE (master must parse and slave may support)	
OBJ	VAR	Description	Func Codes (dec)	Qual Codes (hex)	Func Codes (dec)	Qual Codes (hex)
1	0	Binary Input - All Variations	1 (Read) 22 (Assign Class)	00,01 (Start-Stop) 06 (No Range)		
1	1	Binary Input	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
1	2	Binary Input with Status	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
2	0	Binary Input Change - All Variations	1 (Read)	06 (No Range) 07,08 (Limited Qty)		
2	1	Binary Input Change w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
2	2	Binary Input Change with Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
2	3	Binary Input Change with Relative Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
10	0	Binary Output - All Variations	1 (Read)	00,01 (Start-Stop) 06 (No Range)		
10	1	Binary Output				
10	2	Binary Output Status	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
12	0	Control Block - All Variations				
12	1	Control Relay Output Block	3 (Select) 4 (Operate) 5 (Direct Op) 6 (Dir. Op. Noack)	17,28 (Index)	129 (Response)	echo of request
12	2	Pattern Control Block	5 (Direct Op) 6 (Dir. Op. Noack)	17,28 (Index)	129 (Response)	echo of request
12	3	Pattern Mask	5 (Direct Op) 6 (Dir. Op. Noack)	00,01 (Start-Stop)	129 (Response)	echo of request
20	0	Binary Acc - All Variations	1 (Read) 7 (Imed. Freeze) 8 (Imed. Fr. Noack) 9 (Freeze & Clr.) 10 (Fr. & Clr. Noack) 22 (Assign Class)	00,01 (Start-Stop) 06 (No Range)		
20	1	32-Bit Binary Acc	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
20	2	16-Bit Binary Acc	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
20	3	32-Bit Delta Acc	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
20	4	16-Bit Delta Acc	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
20	5	32-Bit Binary Acc w/out Flag	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
20	6	16-Bit Binary Acc w/out Flag	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
20	7	32-Bit Delta Acc w/out Flag	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
20	8	16-Bit Delta Acc w/out Flag	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
21	0	Frozen Accs - All Variations	1 (Read) 22 (Assign Class)	00,01 (Start-Stop) 06 (No Range)		
21	1	32-Bit Frozen Acc	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
21	2	16-Bit Frozen Acc	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
21	3	32-Bit Frozen Delta Acc	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
21	4	16-Bit Frozen Delta Acc	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)

DNP OBJECT & VARIATION			REQUEST (master may issue and slave must parse)		RESPONSE (master must parse and slave may support)	
OBJ	VAR	Description	Func Codes (dec)	Qual Codes (hex)	Func Codes (dec)	Qual Codes (hex)
21	5	32-Bit Frozen Acc with Time of Freeze				
21	6	16-Bit Frozen Acc with Time of Freeze				
21	7	32-Bit Frozen Delta Acc w/ Time of Freeze				
21	8	16-Bit Frozen Delta Acc w/ Time of Freeze				
21	9	32-Bit Frozen Acc w/out Flag	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
21	10	16-Bit Frozen Acc w/out Flag	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
21	11	32-Bit Frozen Delta Acc w/out Flag				
21	12	16-Bit Frozen Delta Acc w/out Flag				
22	0	Acc Change Event - All Variations	1 (Read)	06 (No Range) 07,08 (Limited Qty)		
22	1	32-Bit Acc Change Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
22	2	16-Bit Acc Change Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
22	3	32-Bit Delta Acc Change Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
22	4	16-Bit Delta Acc Change Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
22	5	32-Bit Acc Change Event with Time				
22	6	16-Bit Acc Change Event with Time				
22	7	32-Bit Delta Acc Change Event w/ Time				
22	8	16-Bit Delta Acc Change Event w/ Time				
23	0	Frozen Acc Events - All Variations	1 (Read)	06 (No Range) 07,08 (Limited Qty)		
23	1	32-Bit Frozen Acc Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
23	2	16-Bit Frozen Acc Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
23	3	32-Bit Frozen Delta Acc Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
23	4	16-Bit Frozen Delta Acc Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
23	5	32-Bit Frozen Acc Event with Time				
23	6	16-Bit Frozen Acc Event with Time				
23	7	32-Bit Frozen Delta Acc Event w/ Time				
23	8	16-Bit Frozen Delta Acc Event w/ Time				
30	0	AI - All Variations	1 (Read) 22 (Assign Class)	00,01 (Start-Stop) 06 (No Range)		
30	1	32-Bit AI	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
30	2	16-Bit AI	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
30	3	32-Bit AI w/out flag	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
30	4	16-Bit AI w/out flag	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
31	0	Frozen AI - All Variations				
31	1	32-Bit Frozen AI				
31	2	16-Bit Frozen AI				
31	3	32-Bit Frozen AI w/ Time of Freeze				
31	4	16-Bit Frozen AI w/ Time of Freeze				
31	5	32-Bit Frozen AI w/out Flag				
31	6	16-Bit Frozen AI w/out Flag				
32	0	Analog Change Event - All Variations	1 (Read)	06 (No Range) 07,08 (Limited Qty)		
32	1	32-Bit Analog Change Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
32	2	16-Bit Analog Change Event w/out Time	1 (Read)	06 (No Range) 07,08 (Limited Qty)	129 (Response) 130 (Unsol Resp)	17,28 (Index)
32	3	32-Bit Analog Change Event with Time				
32	4	16-Bit Analog Change Event with Time				
33	0	Frozen Analog Event - All Variations				
33	1	32-Bit Frozen Analog Event w/out Time				

DNP OBJECT & VARIATION			REQUEST (master may issue and slave must parse)		RESPONSE (master must parse and slave may support)	
OBJ	VAR	Description	Func Codes (dec)	Qual Codes (hex)	Func Codes (dec)	Qual Codes (hex)
33	2	16-Bit Frozen Analog Event w/out Time				
33	3	32-Bit Frozen Analog Event with Time				
33	4	16-Bit Frozen Analog Event with Time				
40	0	Analog Output Status - All Variations	1 (Read)	00,01 (Start-Stop) 06 (No Range)		
40	1	32-Bit Analog Output Status	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
40	2	16-Bit Analog Output Status	1 (Read)	00,01 (Start-Stop) 06 (No Range)	129 (Response) 130 (Unsol Resp)	00,01 (Start-Stop)
41	1	32-Bit Analog Output Block	3 (Select) 4 (Operate) 5 (Direct Op) 6 (Dir. Op. Noack)	17,28 (Index)	129 (Response)	echo of request
41	2	16-Bit Analog Output Block	3 (Select) 4 (Operate) 5 (Direct Op) 6 (Dir. Op. Noack)	17,28 (Index)	129 (Response)	echo of request
50	0	Time and Date - All Variations				
50	1	Time and Date	2 (Write)	07 Quantity=1		
			1 (Read)	07 Quantity=1	129 (Response)	07 (Ltd Qty=1)
50	2	Time and Date with Interval				
51	0	Time and Date CTO - All Variations				
51	1	Time and Date CTO			129 (Response) 130 (Unsol Resp)	07 Quantity=1
51	2	Unsynchronized Time and Date CTO			129 (Response) 130 (Unsol Resp)	07 Quantity=1
52	0	Time Delay - All Variations				
52	1	Time Delay Coarse			129 (Response)	07 Quantity=1
52	2	Time Delay Fine			129 (Response)	07 Quantity=1
60	0	Not Defined				
60	1	Class 0 Data	1 (Read)	06 (No Range)		
60	2	Class 1 Data	1 (Read)	06 (No Range) 07,08 (Limited Qty)		
			20 (Enbl Unsol Msg) 21 (Dis Unsol Msg) 22 (Assign Class)	06 (No Range)		
60	3	Class 2 Data	1 (Read)	06 (No Range) 07,08 (Limited Qty)		
			20 (Enbl Unsol Msg) 21 (Dis Unsol Msg) 22 (Assign Class)	06 (No Range)		
60	4	Class 3 Data	1 (Read)	06 (No Range) 07,08 (Limited Qty)		
			20 (Enbl Unsol Msg) 21 (Dis Unsol Msg) 22 (Assign Class)	06 (No Range)		
70	1	File Identifier				
80	1	Internal Indications	1 (Read)	00,01 (Start-Stop)		
			2 (Write)	00 (Start-Stop)		
81	1	Storage Object				
82	1	Device Profile				
83	1	Private Registration Object				
83	2	Private Registration Object Descriptor				
90	1	Application Identifier				
100	1	Short Floating Point				
100	2	Long Floating Point				
100	3	Extended Floating Point				
101	1	Small Packed Binary-Coded Decimal				
101	2	Medium Packed Binary-Coded Decimal				
101	3	Large Packed Binary-Coded Decimal				
		No Object	13 (Cold Restart)			
		No Object	23 (Delay Meas.)			