

UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA  
DIVISION DE ESTUDIOS DE POST-GRADO



DISEÑO DE UNA RED DE COMUNICACIONES PARA LA  
MONITORIZACION DE DISTURBIOS ELECTRICOS  
EN SISTEMAS DE POTENCIA

POR

MARCO ANTONIO ESCOBAR VERA

TESIS

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS  
DE LA INGENIERIA EN TELECOMUNICACIONES

CD. UNIVERSITARIA

SEPTIEMBRE DE 1998

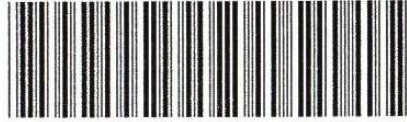


UANL  
FIME  
TESIS

DISEÑO DE UNA RED DE COMUNICACIONES PARA LA MONITORIZACION DE DISTURBIOS ELECTRICOS EN SISTEMAS DE POTENCIA

TE  
Z5853  
.M2  
FIME  
1998  
E8

SEPT.  
1998



1020123505

UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA  
DIVISION DE ESTUDIOS DE POST-GRADO



DISEÑO DE UNA RED DE COMUNICACIONES PARA LA  
MONITORIZACION DE DISTURBIOS ELECTRICOS  
EN SISTEMAS DE POTENCIA

POR

MARCO ANTONIO ESCOBAR VERA

TESIS

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS  
DE LA INGENIERIA EN TELECOMUNICACIONES

UNIVERSITARIA

SEPTIEMBRE DE 1998

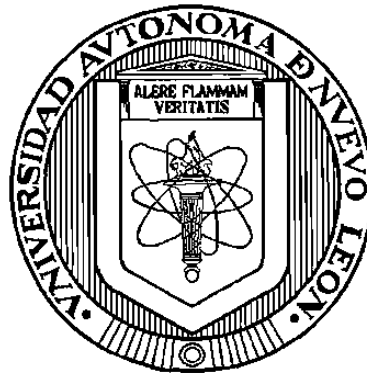
TM  
25853  
.M2  
FINE  
1998  
E8

0119-64560



FONDO  
TESIS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN  
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA  
DIVISIÓN DE ESTUDIOS DE POST-GRADO



DISEÑO DE UNA RED DE COMUNICACIONES PARA  
LA MONITORIZACIÓN DE DISTURBIOS ELÉCTRICOS  
EN SISTEMAS DE POTENCIA

POR:

MARCO ANTONIO ESCOBAR VERA

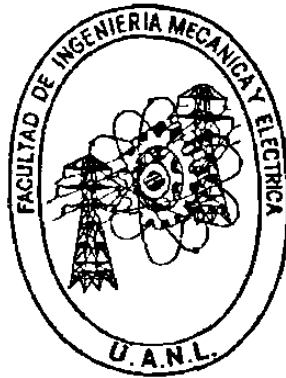
TESIS

EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS  
DE LA INGENIERÍA EN TELECOMUNICACIONES

C.D. UNIVERSITARIA

SEPTIEMBRE 1998

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN  
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA  
DIVISIÓN DE ESTUDIOS DE POST-GRADO



DISEÑO DE UNA RED DE COMUNICACIONES PARA  
LA MONITORIZACIÓN DE DISTURBIOS ELÉCTRICOS  
EN SISTEMAS DE POTENCIA

POR:

MARCO ANTONIO ESCOBAR VERA

TESIS

EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS  
DE LA INGENIERÍA EN TELECOMUNICACIONES

C.D. UNIVERSITARIA

SEPTIEMBRE 1998

***Con mucho amor y cariño dedico este trabajo...***

***A mis queridos padres, Eduardo Escobar Nakamura y Guillermina Vera Hernández, quienes constantemente me han apoyado y motivado con valiosos consejos para seguir adelante.***

***A mi hermana, Ana Lilia, por su cariño, apoyo, comprensión y confianza que siempre me ha tenido.***

***A mis queridos tíos:***

***Elías Paz Hernández y Carmela Cristiani de Paz, quienes con su incondicional amor y comprensión en las buenas y en las malas perpetuamente han estado pendientes de mi y de quienes me siento muy orgulloso.***

***Jorge Elías Paz Cristiani y Rosa Gómez de Paz, quienes con su gran ayuda y confianza depositada en mi conservan un lugar muy especial en mi memoria.***

***Ernesto Paz Cristiani y María del Refugio Lozano de Paz, quienes con su incomparable cariño y afecto he disfrutado de momentos muy gratos.***

***Isaías Castro y Yolanda Paz de Castro, quienes con su apoyo y comprensión siempre me he sentido seguro.***

***Claudio López y Concepción Vera de López, quienes desde niño siempre tengo presentes mi corazón por su cálido amor y fuerte confianza.***

***A todos mis primos y sobrinos, con quienes he disfrutado y convivido momentos tan bellos.***

***A la Familia De Luna Medina, quienes desinteresadamente me han brindado su ayuda, confianza y apoyo.***


***A todos mis amigos, con quienes he convivido y aventurado momentos tan placenteros en las distintas etapas de mi vida.***




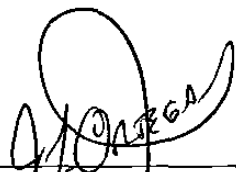
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN  
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA  
DIVISIÓN DE ESTUDIOS DE POST-GRADO


Los miembros del comité de tesis recomendamos que la tesis DISEÑO DE UNA RED DE COMUNICACIONES PARA LA MONITORIZACIÓN DE DISTURBIOS ELÉCTRICOS EN SISTEMAS DE POTENCIA realizada por el Ing. Marco Antonio Escobar Vera, matrícula 728176 sea aceptada para su defensa como opción al grado de Maestro en Ciencias de la Ingeniería en Telecomunicaciones.

El Comité de Tesis

  
Asesor  
Dr. José Antonio de la O Serna

  
Coasesor  
Dr. Juan Arturo Nolazco Flores

  
Coasesor  
M. C. José Luis Ortega Soria

  
Vo.Bo.  
M.C. Roberto Villarreal Garza  
División de Estudios de Postgrado

San Nicolás de los Garza, N.L., Septiembre de 1998

## AGRADECIMIENTOS

*Al Dr. José Antonio de la O Serna, por su paciencia, comprensión y asesoría en la elaboración de esta tesis.*

*Al Dr. Salvador Acha Daza y al Ing. Raúl Alvarado Escamilla, por proporcionarme los medios para realizar las pruebas de este trabajo.*

*Al Ing. Gilberto Ordoñez Pineda, por facilitarnos el hardware y software de adquisición de datos así como los modems para llevar a cabo este proyecto.*

*Al Dr. José Luis Martínez Flores y al Ing. Rodolfo Castillo, por prestarme sus computadoras para llevar a cabo las pruebas de comunicación serial.*

*A mi compañero y amigo Rodolfo Ruben Treviño Martínez, por sus comentarios y su paralela colaboración a este trabajo.*

*A los miembros del jurado, por sus valiosas aportaciones en la revisión de esta tesis.*

*A todos los que participaron directa o indirectamente con sus comentarios y contribuciones para la realización de este trabajo, en especial al Dr. Ernesto Vásquez Martínez y al Dr. Héctor Jorge Altuve Ferrer.*

*Al Consejo Nacional de Ciencia y Tecnología y a la Universidad Autónoma de Nuevo León.*

*A mis padres, por sus consejos de aliento en las situaciones más difíciles y por su apoyo moral y económico.*

## RESUMEN

### DISEÑO DE UNA RED DE COMUNICACIONES PARA LA MONITORIZACIÓN DE DISTURBIOS ELÉCTRICOS EN SISTEMAS DE POTENCIA

*Publicación No. \_\_\_\_\_*

*Marco A. Escobar Vera, M.C. de la Ingeniería en Telecomunicaciones  
Universidad Autónoma de Nuevo León, 1998.*

*Profesor Asesor: Dr. José Antonio de la O Serna*

En la presente tesis se describe el diseño e implementación de una red de comunicaciones para la monitorización de disturbios en líneas de transmisión.

El desarrollo de esta red surge de la necesidad de concentrar la información de los disturbios electromecánicos registrados en diferentes puntos de medición para su estudio y análisis, con la posibilidad posterior de detección, corrección o eliminación de factores que las producen.

La red permite a una computadora central, usando líneas telefónicas como medio de transmisión, almacenar registros de disturbios de cinco terminales, las cuales tienen instalado un *software* de adquisición de datos. Dicha computadora estará ubicada en el CENACE (Centro Nacional de Control de Energía) de Monterrey, mientras que las terminales se encuentran en puntos frontera de la red de distribución eléctrica del Área Noreste de Comisión Federal de Electricidad. La computadora también monitoriza las acciones de las terminales, realizando tareas tales como: sincronización de relojes, envío de comandos y cambio remoto de la configuración del sistema de adquisición de datos.

Para la descripción del funcionamiento de la red, se especifica su capa física. Se presenta la operación de la capa de enlace y la de aplicación, mostrando el formato, la estructura y la descripción general de los mensajes, así como la interacción y tráfico de los mismos. Para mostrar la lógica de conmutación de la capa de enlace, se muestran los diagramas de transición de estado del monitor central y de las terminales.

Por último, se muestra el panorama general del estándar PROFIBUS para procesos de automatización, con la finalidad de tener una referencia que permita evaluar la red propuesta.



# ÍNDICE

<i>Agradecimientos</i>	<i>iv</i>
<i>Resumen</i>	<i>v</i>
<i>Índice de figuras</i>	<i>x</i>
<i>Índice de tablas</i>	<i>xi</i>
<b>CAPÍTULO I. INTRODUCCIÓN</b>	<b>1</b>
1.1 Antecedentes	2
1.2 Objetivos de la tesis	5
1.3 Estructura de la tesis	6
<b>CAPÍTULO II. CAPA FÍSICA</b>	<b>8</b>
2.1 Interfase de la Capa Física	8
2.2 Conexión	9
<b>CAPÍTULO III. CAPA DE ENLACE</b>	<b>11</b>
3.1 Descripción	11
3.2 Formato general de los mensajes	12
3.3 Tránsito de los mensajes	15
3.4 Establecimiento de la comunicación	18
3.5 Estados del transmisor y el receptor	19
3.5.1 Estado 0: Inicio	20
3.5.2 Estado 1: Envío de mensaje	20
3.5.3 Estado 2: Recepción de mensaje	20
3.5.4 Estado 3: Procesamiento del mensaje	21
3.5.5 Transiciones de los dispositivos	21

## ÍNDICE (continuación)

3.6 Funciones para el manejo de los datos	22
3.6.1 Captura de los datos	22
3.6.2 Recepción de los mensajes	23
3.6.3 Transmisión de los mensajes	25
<b>CAPÍTULO IV. CAPA DE APLICACIÓN</b>	<b>28</b>
4.1 Descripción	28
4.2 Operación del monitor central	29
4.3 Operación de las terminales	30
4.4 Descripción de los mensajes	31
4.5 Transferencia de los registros de disturbios	34
4.6 Sincronización de las terminales	37
4.6.1 Sincronización con GPS's	38
4.6.2 Sincronización por medio del monitor central	39
4.6.2.1 Proceso 1: Sincronía automática por el monitor	41
4.6.2.2 Proceso 2: Sincronía por el usuario en el monitor	41
4.6.2.3 Proceso 3: Petición de sincronía por la terminal	41
4.7 Configuración de la tarjeta de adquisición de datos	41
4.8 Transferencia de texto	42
<b>CAPÍTULO V. EVALUACIÓN</b>	<b>43</b>
5.1 Campo de aplicación y propósito de PROFIBUS	43
5.2 Comparación entre el estándar PROFIBUS y la red de comunicaciones presentada en esta tesis	44

## ÍNDICE (continuación)

<b>CAPÍTULO VI. CONCLUSIONES</b>	<b>48</b>
6.1 Conclusiones	48
6.2 Aportaciones	50
6.3 Recomendaciones para trabajos futuros	51
<b>REFERENCIAS</b>	<b>52</b>
<b>APÉNDICES</b>	
A: Registros del UART	54
B: Recomendación V.32 Bis	62
C: Listado del programa de la red de comunicaciones	64
D: Estándar RS-232C/V.24	98
E: Estándar PROFIBUS	110
<b>RESUMEN AUTOBIOGRÁFICO</b>	<b>137</b>

## ÍNDICE DE FIGURAS

1.1	Distribución geográfica de los nodos fronterizos de la red de distribución de CFE en el Área Noreste	4
3.1	Formato general de mensajes	13
3.2	Probabilidad de detección para errores dobles en los bits de una columna de 255 octetos.	14
3.3	Diagrama de transición de estado de la capa de enlace del transmisor	19
3.4	Diagrama de transición de estado de la capa de enlace del receptor	19
4.1	Proceso de transferencia del archivo de disturbio	35
4.2	Intento de establecimiento de conexión con el monitor central	36
4.3	Procesos de sincronización	40
D.1	Disposición de registros del UART	102
D.2	Diagrama a bloques del transmisor UART	104
D.3	Diagrama a bloques del UART receptor	106
D.4	Registro de desplazamiento de transmisión del UART	107
D.5	Entrada de datos serie	107
E.1	Arquitectura del protocolo PROFIBUS	111
E.2	Áreas de aplicación de PROFIBUS	111
E.3	Repetidores en topología Bus lineal	115
E.4	Pines de salida, vista frontal del conector macho y vista trasera del conector hembra respectivamente	116
E.5	Alambrado de interconexión	117
E.6	Bus Terminador	117
E.7	Interface entre PHY y FDL en relación al modelo OSI	119
E.8	Codificación de las señales TxD, RxD y RTS	120
E.9	Redundancia de la Capa Física y el Medio	121
E.10	Rotación del token sobre el anillo lógico	124
E.11	Diagrama del estado FDL	126
E.12	Formato y estructuras de las tramas	131



## ÍNDICE DE TABLAS

4.1	Descripción y uso de los mensajes	31
4.2	Contenido de mensajes	32
4.3	Contenido del campo de datos	33
4.4	Comparación entre dos transferencias de un mismo archivo con diferente tamaño en la trama	33
5.1	Familia PROFIBUS	44
5.2	Comparación del estándar PROFIBUS contra la red propuesta en esta tesis	45
A.1	Mapa de bits del LCR	55
A.2	Divisores de la razón de baudios bajo la referencia de dos relojes	56
A.3	Mapa de bits del LSR	57
A.4	Mapa de bits del MCR	58
A.5	Mapa de bits del MSR	59
A.6	Mapa de bits del IER	60
A.7	Mapa de bits del IIR	61
A.8	Condiciones y prioridades de interrupción	61
D.1	Definiciones de los pines y las señales de la interfase RS-232C/V.24	99
D.2	Asignación de los puertos E/S e interrupciones para el correspondiente puerto serie	100
D.3	Direcciones de los datos y registro de control que manipulan las funciones del UART	102
D.4	Configuración del puerto de serie	103
E. 1	Parámetros de operación del administrador FMA1/2	127

## CAPÍTULO I

### INTRODUCCIÓN

En la actualidad existe cada vez más la necesidad de comunicar computadoras a través de redes de datos confiables. Estas redes por lo regular son complejas y costosas ya que se diseñan para transportar grandes volúmenes de datos. También requieren de mantenimiento intensivo y generalmente los equipos de comunicación de distintos fabricantes son incompatibles entre sí.

En esta tesis se presenta el diseño e implementación de una red de datos para transmitir la información de las señales de voltaje y corriente de las oscilaciones electromecánicas registradas en subestaciones del área noreste de la Comisión Federal de Electricidad. El problema de la detección y registro de dichas oscilaciones constituye el tema de otro trabajo paralelo de investigación, por lo que aquí se supone que se dispone ya de tales registros. El proyecto inicial pretende conectar cinco Unidades Terminales Remotas (UTR) a un monitor central ubicado en el Centro Nacional de Control de Energía (CENACE) en Monterrey.

Dados los medios de comunicación disponibles actualmente en las subestaciones y la baja probabilidad de los eventos a comunicar se optó por una configuración punto a punto con enlaces conmutados a través de modems. Esta solución, que es la más sencilla, permite optimizar los canales de comunicación al no requerir de líneas dedicadas en los enlaces, y por otra parte es expandible ya que se le puede agregar tantos nodos como se requieran. Sin embargo, la topología de esta red impide la comunicación simultánea de todos sus nodos. Esta limitación sería significativa en el caso de que se desearan las señales de los nodos en

donde no se registro la oscilación. Además, la comunicación para el registro simultáneo de datos en todos los nodos necesita de canales de fibra óptica los cuales no están disponibles actualmente. Con el fin de detallar la elaboración de este trabajo a continuación se explican los antecedentes del proyecto, los objetivos y la estructura de la tesis.

## **1.1 ANTECEDENTES**

Los disturbios que se presentan en los sistemas eléctricos de potencia constituyen un tema de interés en el que muchos ingenieros e investigadores se encuentran trabajando actualmente. Un disturbio eléctrico puede definirse como cualquier perturbación anormal que modifica el comportamiento del sistema eléctrico. Los disturbios son ocasionados principalmente por descargas atmosféricas, movimientos telúricos, fuertes vientos o accidentes, y provocan situaciones de alarma para el personal encargado del control y restablecimiento del sistema. Las señales de corriente y de voltaje emanadas de dichos disturbios constituyen una riquísima fuente de información para el ingeniero de diseño y mantenimiento de la red de distribución. En particular, del análisis de las oscilaciones electromecánicas se puede conocer la estabilidad transitoria del sistema eléctrico de potencia, sobre todo si esas señales son capturadas en los diferentes nodos de la red de transmisión. Por otra parte posibilitan el modelado dinámico aplicando algoritmos de identificación.

Existen actualmente equipos registradores que permiten almacenar la información de los transitorios provocados por un disturbio. Estos equipos, basados en microprocesadores, poseen una capacidad de procesamiento y almacenamiento de datos. Son equipos que están orientados específicamente a las compañías eléctricas y a las grandes industrias, ofrecen un medio de detección y registro de las señales eléctricas durante la producción del disturbio. También brindan la oportunidad de analizar posteriormente los registros almacenados. Los registros de datos son almacenados en archivos de longitud variable. La información guardada en cada archivo contiene datos de prefalla, falla y de postfalla. Estos registradores, al momento del disturbio en el sistema, pueden ser inicializados de manera programada o

bien a través de un botón externo. Además de la capacidad de cómputo, los equipos están dotados de canales de comunicación [20, 21].

La principal desventaja de estos equipos es su costo, y muchas veces usan protocolos cerrados por lo que crean una dependencia hacia el fabricante cuando requieren mantenimiento o cuando se desea modificar sus algoritmos internos. En particular, ofrecen los datos fasoriales aplicando sus propios algoritmos sobre las señales digitales, lo cual impide el acceso directo a éstas.

Otra fuerte limitación de dichos equipos es que su *software* está programado en un memoria fija (*EPROM*), por lo que es difícil de modificar. Esta falta de flexibilidad impide variar inteligentemente el procesamiento de datos y los modos de comunicación adaptándolos a las circunstancias propias del evento.

El presente proyecto surge de la necesidad de estudiar y analizar los datos de las oscilaciones electromecánicas que aparecen en un disturbio. El proyecto piloto consiste en instalar unidades terminales remotas en cinco subestaciones del Área Noreste de la Comisión Federal de Electricidad (CFE). Cada una de ellas consta de una computadora provista de una tarjeta de adquisición de datos, la cual posee el algoritmo de detección de disturbios y el *software* para el registro de señales trifásicas de voltaje y de corriente. Para transportar automáticamente esta información a un monitor central ubicado en el Centro Nacional de Control de Energía de Monterrey (CENACE), se diseñó e implementó la red de datos que constituye el tema central de esta tesis [1].

Las UTR serán ubicadas en las subestaciones y plantas generadoras claves del Área Noreste: Río Escondido en Coahuila, Río bravo y Altamira en Tamaulipas, San Jerónimo y Monterrey en Nuevo León. La distribución geográfica de estos lugares se muestra a continuación en la Fig. 1.1.



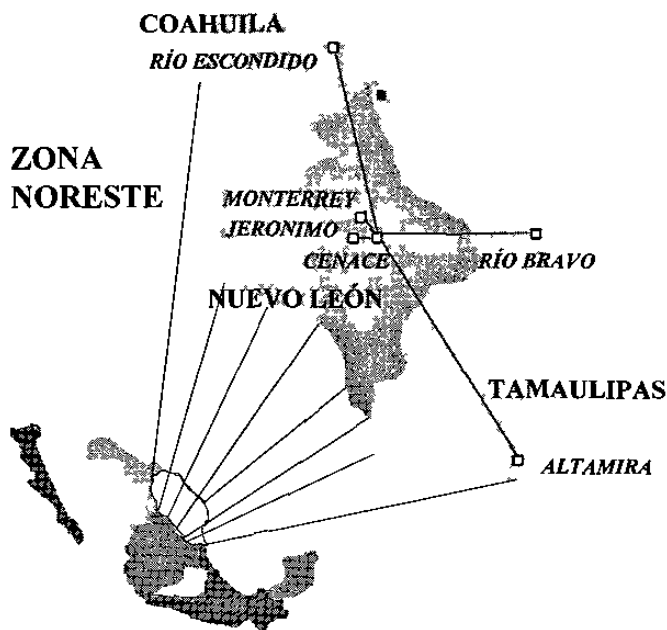


Fig. 1.1 Distribución geográfica de los nodos fronterizos de la red de distribución de CFE en el Área Noreste.

Como se puede apreciar en la Fig. 1.1 los nodos de la red se encuentran a muchos kilómetros de distancia. De ahí la necesidad de conectarlos vía módem con el monitor utilizando los puertos serie de las computadoras. El desarrollo y prueba del *software* de comunicación en todas sus etapas es la materia de la presente tesis. Aún cuando su principal aportación se halla dado en el diseño y codificación de los protocolos de comunicación (capa de enlace y de aplicación) también se utiliza y describe el *software* para el manejo de los puertos seriales, el cual es de uso comercial.

Los protocolos de comunicación se diseñaron de acuerdo a las estructuras de datos requeridas por la información y los comandos necesarios para transportarla. Los datos más importantes y voluminosos son los datos de las señales trifásicas de voltaje y de corriente.

Estas se muestrean tomando dieciséis muestras por ciclo con 16 bits cada una lo que significa para las ocho señales una tasa binaria de transmisión de 122,880 bps. Aún cuando las señales de los disturbios no son periódicas, se aplica un algoritmo de compresión de datos que reduce dicha tasa en un 60% en promedio, esta reducción disminuye en la medida que las señales contengan más segmentos periódicos.

En el desarrollo de la presente red surgió la idea de utilizarla para incorporarle los equipos de control y medición existentes actualmente en las subestaciones. Dado que la tendencia de dichos equipos es el uso del estándar PROFIBUS para comunicarse, en el capítulo V se presenta una descripción consisa sobre dicho estándar de la que se concluye la inconveniencia de utilizarlo en la presente red ya que eso requeriría el uso de *hardware* especializado, aumentando su complejidad y costo. Por otra parte, dado que el estándar PROFIBUS se orienta hacia el control de procesos de automatización, constituiría un cuello de botella para la comunicación requerida en la transmisión de señales de disturbios.

## 1.2 OBJETIVOS DE LA TESIS

Los objetivos principales de la tesis son:

- Diseñar e implantar una red de comunicaciones para el monitoreo y la concentración de datos de las señales trifásicas de voltaje y de corriente en disturbios registrados en las subestaciones de un sistema eléctrico de potencia.
- Dicha red deberá activarse automáticamente al momento de presentarse un disturbio en cualquiera de sus nodos. Para ello se utiliza procesadores de señales capaces de detectar el disturbio.

- Una vez que la información transmitida se registre en el Centro de Control podrá ser accesada a través de *internet* para el estudio y el análisis de los factores que suscitan los disturbios y poder así corregirlos o eliminarlos.
- Elaborar el *software* de comunicaciones acompañado de documentación detallada a fin de poder realizar las futuras modificaciones que otras aplicaciones requieran.
- Desarrollar un algoritmo interno para que el monitor sincronice las diferentes UTR's conectadas a la red, empleando tarjetas de sincronía. Esto con el fin de evitar el elevado costo de los equipos de sincronización por medio de satélites.
- Presentar el panorama general del estándar PROFIBUS empleado en procesos de automatización a fin de evaluar la red tratada en esta tesis

### **1.3 ESTRUCTURA DE LA TESIS**

En la tesis se contemplan seis Capítulos, referencias bibliográficas y apéndices. El contenido breve de cada Capítulo se describe a continuación:

En el primer Capítulo, se introduce y describe la importancia de la tesis, presentando sus objetivos principales y describiendo la estructura general de la misma.

En el segundo Capítulo, se presenta la descripción de la capa física de la red, constituida por el puerto serial. Su conexión y funcionamiento son de vital importancia para el diseño y operación de la red.

En el Capítulo tres, se describe la capa de enlace, mostrando los diagramas de estado del transmisor y receptor así como sus correspondientes transiciones. Además se describe el

formato general de los mensajes necesarios para la aplicación, el cual ayuda a entender la operación de la red. Las funciones más relevantes de esta capa son presentadas en lenguaje C.

En el Capítulo cuatro se describe la operación de la capa de aplicación. Para esto se explica la operación del monitor central y las terminales, se describen los mensajes necesarios para conducir la información, así como también su estructura y contenido. También se describe el proceso de transferencia de archivos y de texto, el manejo de la sincronización entre terminales y el proceso para configurar la tarjeta de adquisición de datos.

En el Capítulo cinco se presenta la arquitectura de red del estándar PROFIBUS (*Process Field Bus*), con el objetivo de realizar un estudio comparativo con la red realizada en este trabajo.

Para finalizar, en el Capítulo cinco se presentan las conclusiones de la tesis, se dan a conocer las aportaciones y se muestran las recomendaciones para trabajos futuros en esta área.



## CAPÍTULO II

### CAPA FÍSICA

La capa física constituye el primer nivel de cualquier tipo de red en el modelo de referencia OSI (*Open Systems Interconnection*) [2,3,7,9,23]. La comprensión de su funcionamiento es fundamental para entender las diferentes fases de los algoritmos que gobiernan la operación de la red. En este capítulo se presentan los aspectos más relevantes del funcionamiento de la Capa Física, constituida en nuestro caso por los puertos seriales de las computadoras conectadas a red. En particular se explica cómo se establece, mantiene y desactiva la conexión física del monitor con los UTR's de la red, principales funciones de toda capa física.

#### 2.1 INTERFASE DE LA CAPA FÍSICA

Las comunicaciones llevadas a cabo entre el monitor y los UTR's se establecen a través de la interfase RS-232C/V.24. El RS-232C, definido por EIA (*Electronic Industries Association*), y el V.24, establecido por el CCITT (*Comite Consultive International Telegraphic and Telephonic*), son un estándar eléctrico para la conexión de componentes como módems, impresoras y computadoras, el cual se presenta detalladamente en [2,3,4,5,6,7,23]. Dicho estándar especifica la secuencia de control de la línea de comunicación para configurar, mantener y desactivar el nivel físico de interconexión en la transferencia de datos. Estas características comprenden las secuencias del protocolo de diálogo (*handshaking*) para inicializar las acciones de control y verificar que hayan sido

efectuadas. La descripción y uso de las señales del interfase RS-232 así como la operación del puerto serie se presenta detalladamente en el apéndice D.

Las limitaciones del interfase RS-232 son la velocidad de transmisión y la distancia entre la PC y el módem; las cuales no deberán ser superiores a 20 kbps y 15 metros respectivamente. Estas limitaciones no afectan la operación de la red, ya que la tasa de transmisión de las líneas telefónicas está limitada por 14.4 kbps si se desea una transmisión de buena calidad, y la distancia entre cada PC y su módem será siempre inferior a 2 mts.

## 2.2 CONEXIÓN

Para llevar a cabo la conexión entre la PC y el módem se utilizaron los conectores DB-25 de la siguiente manera: los pines 2 y 3 de la PC manejan las señales de transmisión y recepción respectivamente, y se conectan con los del mismo número del módem. La señal DTR del pin 20 es utilizada por la PC para activar al módem, y la señal CD del pin 8 indica a la PC que se ha establecido conexión con el lado remoto (detección de portadora). La PC transmite los datos por el pin 2 mientras que el módem lo hace por el pin 3. El pin 7 (GND) es común para los dos.

La interconexión entre módems se rige bajo la recomendación V.32 bis, esta recomendación, establecida por el CCITT, contempla características físicas adecuadas para operar la comunicación de datos punto a punto empleando líneas conmutadas de dos hilos, con velocidades de transmisión comprendidas entre los 600 y 14400 bps. Estas velocidades se pueden establecer de dos maneras: de forma manual y de forma automática. El establecimiento de dicha velocidad depende de la razón señal/ruido presente en la línea telefónica. Las características generales de la dicha recomendación se presentan en el apéndice B.

En este trabajo, con el fin de establecer la velocidad de transmisión más adecuada, se realizaron cerca de 15 pruebas con las líneas telefónicas disponibles en lugares locales del área metropolitana de Monterrey, obteniendo un buen desempeño en la recepción de los datos a una velocidad de 9600 bps. Esta velocidad para comunicación a larga distancia puede ser inferior si la calidad de la línea así lo requiere.

### CAPA DE ENLACE

En este capítulo se describe la capa de enlace necesaria para controlar y coordinar la comunicación entre los equipos situados a ambos extremos del circuito físico de comunicaciones. Para ello se define una serie de reglas que administran el flujo de datos entre dos entidades y constituyen su protocolo. Dichas reglas tienen como propósito principal el proveer medios funcionales para la conexión lógica a través de la línea, controlando a su vez el direccionamiento, secuenciamiento y detección de errores en los datos. Con el objeto de presentar la operación de la capa de enlace de la red, en las secciones siguientes se describe su protocolo, el formato general de los mensajes, el tránsito e intercambio de los mismos, los diagramas de estado del transmisor y el receptor. Al final se presentan las funciones de programación más importantes para el manejo de los datos.

#### 3.1 DESCRIPCIÓN

En la actualidad existen en el mercado gran variedad de protocolos de enlace, los cuales caracterizan cada sistema de comunicación; unos son estándares ya definidos (HDLC, Bisync, ARPANET, LAPB, X.25, etc.) [2,3,7,8,9], algunos se ajustan al modelo de comunicación que nos interesa y otros se encuentran en estudio. Estos últimos surgen por la creciente aparición de nuevos medios de comunicación tales como las redes digitales de servicios integrados (*ISDN*), redes satelitales, enlaces con fibras ópticas, entre otros. Estos protocolos prometen ser más fiables, más adaptables y tienden a tener una mayor compatibilidad entre equipos.

Cabe destacar que el protocolo que aquí se presenta no corresponde a un estándar definido, aunque posee ciertas características similares. Su diseño está basado en un protocolo de parada y espera (*Stop and Wait*). En el protocolo primero se plantearon las necesidades de comunicación, estudiando y definiendo el conjunto de mensajes requeridos para la solicitud y el envío de los datos. En seguida, para el control del flujo de los datos, se definieron sus correspondientes contadores y temporizadores. Por último, como parte fundamental de este trabajo, se elaboró el programa de comunicaciones para establecer, mantener y desactivar la interconexión del enlace, con la posibilidad de detectar errores de transmisión en el intercambio de los datos.

Considerando los medios de comunicación actualmente disponibles y la baja probabilidad de eventos a comunicar se optó por una red punto a punto con enlaces conmutados, empleando modems y líneas telefónicas como medio de transmisión. Esta decisión se tomó ya que resulta la mejor opción para optimizar el costo de los canales de comunicación además de evitar la compra de equipo periférico para redes. Cabe destacar que la renta de un sistema digital empleando alguna tecnología de transmisión como *ATM* o *frame relay* garantiza una mayor velocidad y confiabilidad en los datos. Sin embargo, esto resulta demasiado costoso cuando los flujos de información no son continuos o constantes.

A continuación se describe el formato general de los mensajes.

### **3.2 FORMATO GENERAL DE LOS MENSAJES**

La información transita en forma de mensajes cortos constituidos por una trama de dos campos: una cabecera y un cuerpo de datos. En el Capítulo 4 se describe detalladamente el contenido de cada mensaje y se explica su función correspondiente. El formato general de la trama se muestra a continuación en la Fig. 3.1. La cabecera contiene tres octetos, el primero define el tipo de mensaje, el segundo la cantidad de datos, y el último contiene el control de errores para esta cabecera (*Check Sum 1*).

CABECERA DE LA TRAMA			CÁMPO DE DATOS	
Tipo de mensaje (1 octeto)	Cantidad de datos (1 octeto)	Control de cabecera (1 octeto)	Datos de información (1-255 octetos)	Control de datos (1 octeto)

Fig. 3.1 Formato general de mensajes

El contenido del campo de datos depende del tipo de mensaje. Los mensajes de control no portan información por lo que carecen de campo de datos. Cuando éste se requiere, su carga útil en octetos es variable de 1 a 255 y termina con un comprobante para el control de errores de transmisión (*Check Sum 2*).

La información contenida en los octetos de control de errores se obtiene mediante la simple suma de los octetos precedentes en cada campo y de este resultado solo se toma el octeto menos significativo. Lo anterior resulta más eficiente que emplear el operador "OR EXCLUSIVO" sobre los octetos precedentes. Para evaluar dicha eficiencia consideremos una matriz de  $n \times b$ , donde  $n$  representa la cantidad de octetos presentes en el campo de datos y  $b$  el número de bit del octeto, donde  $b$  va de 1 a 8. Si consideramos cuatro octetos continuos ( $n=4$ ), el cálculo del *Check Sum* para el mensaje a transmitir será respectivamente:

	$b$							
$n$	8	7	6	5	4	3	2	1
1	0	0	0	1	1	0	0	1
2	1	1	0	1	1	1	0	0
3	0	1	0	1	0	1	1	1
4	1	1	1	1	0	1	1	1
OR EXC →	01100101							

	$b$							
$n$	8	7	6	5	4	3	2	1
1	0	0	0	1	1	0	0	1
2	1	1	0	1	1	1	0	0
3	0	1	0	1	0	1	1	1
4	1	1	1	1	0	1	1	1
10 →	01000011 ← SUMA							

Si al recibir el mensaje ocurren dos errores en los bits de una columna que en su origen tenían valores *contrarios* (0 y 1), la detección para ambos métodos no será posible, es decir:

	$b$							
$n$	8	7	6	5	4	3	2	1
1	0	0	0	1	1	0	0	1
2	1	1	0	1	1	1	0	1
3	0	1	0	1	0	1	1	1
4	1	1	1	1	0	1	1	1
OR EXC →	01100101							

	$b$							
$n$	8	7	6	5	4	3	2	1
1	0	0	0	1	1	0	0	1
2	1	1	0	1	1	1	0	1
3	0	1	0	1	0	1	1	1
4	1	1	1	1	0	1	1	1
10 →	01000011 ← SUMA							

No hay detección

Sin embargo, si los dos bits dañados originalmente tenían valores *iguales* (0 y 0 o 1 y 1), la detección únicamente será posible con la suma, es decir:

<i>n</i>	<i>b</i>							
	8	7	6	5	4	3	2	1
1	0	0	0	1	1	0	0	0
2	1	1	0	1	1	1	0	0
3	0	1	0	1	0	1	1	1
4	1	1	1	1	0	1	1	0
OR EXC →	01100101							

bits  
cambiados

<i>n</i>	<i>b</i>							
	8	7	6	5	4	3	2	1
1	0	0	0	1	1	0	0	0
2	1	1	0	1	1	1	0	0
3	0	1	0	1	0	1	1	1
4	1	1	1	1	0	1	1	0
1 0 →	01000001							

bits  
cambiados  
← SUMA  
Detección

Si consideramos una columna de  $N$  bits, donde  $r$  representa la cantidad de errores en los bits de la columna. El total de cambios posibles en los bits de la columna para errores dobles ( $r = 2$ ) son:  $TC = \frac{N!}{r!(N-r)!} = \frac{N!}{2!(N-2)!}$ . Los cambios dobles en los bits con valores *contrarios* son:  $CC = Z(N-Z)$ , donde  $Z$  representa la cantidad de bits con estado lógico cero presentes en la columna. Dado que los cambios dobles con bits *contrarios* no se pueden detectar, la probabilidad de NO detección es  $CC/TC$ , o bien la probabilidad de detección es:  $1 - CC/TC$ . En nuestra situación cuando se transmiten tramas con la máxima capacidad de datos ( $n = 255$  octetos), las probabilidades de detección sobre una columna serán como en la Fig. 3.2.

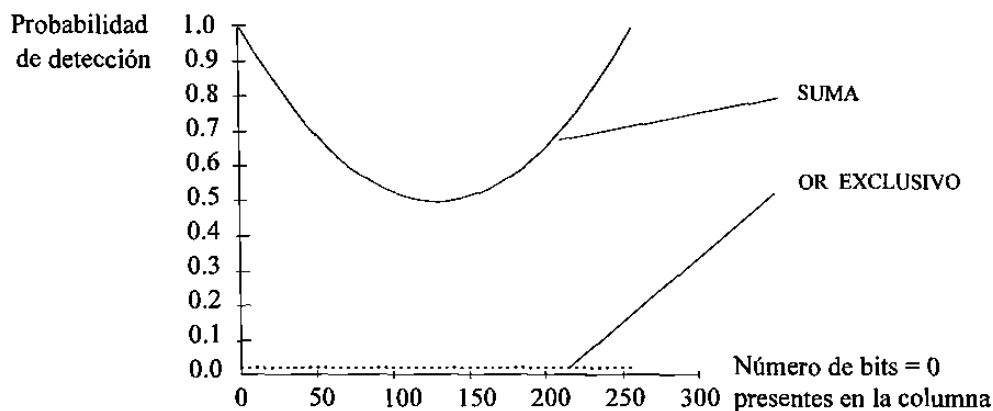


Fig. 3.2 Probabilidad de detección para errores dobles en los bits de una columna de 255 octetos.

Como se puede apreciar el método de la suma ofrece mayor posibilidad para detectar errores pares en los bits de una columna. La detección resulta siempre exitosa cuando los bits

dañados sean impares o no se encuentren sobre una misma una columna. Este método de detección ofrece un considerable desempeño comparado con el método CRC o polinomial donde el tiempo de procesamiento es aún mayor [3,7,9]. Con el propósito de entender el funcionamiento de la capa de enlace, a continuación se explican los detalles del tránsito de mensajes sin importar su contenido.

### 3.3 TRÁNSITO DE LOS MENSAJES

Los mensajes, provenientes de la capa física, pasan a través de la capa de enlace hacia la de aplicación y viceversa. Las fronteras de la capa de enlace están constituidas por dos interfases que le permiten controlar y dirigir la información en ambos sentidos. Ambas poseen una memoria intermedia (*buffer*) e indicadores para el control del tráfico. La primera, llamada interfase externa por colindar con la capa física, recibe los mensajes provenientes del puerto serie. Se ha mencionado anteriormente que la capa física se encarga de registrar ahí los octetos recibidos. La de enlace copia el contenido de la interfase externa en la interna y examina en ésta el contenido del mensaje para comprobar su correcta recepción. En caso de existir error, solicita su retransmisión mediante el comando apropiado. Cuando el mensaje es correcto y completo, la capa de enlace indica a la de aplicación que hay un mensaje disponible para su ulterior procesamiento. A su vez, la capa de aplicación avisa a la de enlace cuando la memoria queda libre. La capa de enlace cuenta con los medios para restaurar la memoria de la interfase externa o la capa física misma.

El envío de una respuesta se activa automáticamente con la recepción completa del mensaje recibido. Por esa razón, el inicio de una sesión de transferencia de información lo realiza una de las dos computadoras del enlace. Para enviar mensajes, la capa de aplicación los registra en el *buffer* de la interfase interna, y la de enlace se encarga de desplazarlos uno por uno a la interfase externa, indicando a la de aplicación cuando el *buffer* se libera. La capa de aplicación indica a la capa de enlace cuando un mensaje está disponible en la interfase interna, y puede restaurarla en caso de detección de error.



Los mensajes, al ser enviados por el medio de transmisión, pueden dañarse debido a la presencia de interferencias electromagnéticas aleatorias en el medio ambiente [2,8,9,10]. Para evitar la incorrecta recepción de un mensaje se utiliza el octeto de control de error en cada campo. Al detectar que el mensaje recibido no concuerda con su comprobante de control, se envía un comando de retransmisión denominado *Nack*.

Cabe destacar que el protocolo aquí presentado no cuenta con un algoritmo para corregir errores en la recepción. Esta corrección sería justificable si el flujo de datos fuera en un solo sentido de tal forma que no pudiesen solicitarse retransmisiones. Además, la corrección de errores ofrece mayores ventajas cuando hay más de dos estaciones en una red, de tal manera que cuando una estación este realizando la verificación y posible corrección en el mensaje, el *token* siga circulando [7]. Esto optimiza tiempo en la operación de la red haciendola más eficiente.

Dado que las fallas físicas del medio de transmisión no pueden ser detectadas por la capa física, la capa de enlace se encarga de ello. Con ese propósito se definió un mensaje de confirmación de recepción y un plazo para recibirlo después de una transmisión. En caso de no recibir esta confirmación durante el plazo designado, la capa de enlace detecta la falla física en la línea [22].

Para controlar el tráfico de los mensajes, la capa de enlace utiliza dos contadores y un temporizador. Esto permite fijar un plazo para detectar condiciones de error por tardanza o algún daño en el medio. El temporizador denominado *Temporizador de entrada* (TE), se encarga de medir el lapso de tiempo entre el fin de una transmisión completa y la recepción correcta de su respuesta. Detecta fallas en la línea cuando se vence el plazo de recepción de respuestas. Cuando esto sucede la capa de enlace transmite de nuevo el mensaje. Si esto persiste más de tres intentos, se declara línea interrumpida, la capa de enlace regresa a su estado inicial y se suspende la comunicación. El límite de tiempo del temporizador es de 0.3 segundos. Este límite de tiempo se fijó bajo pruebas realizadas con dos computadoras

ubicadas una en Escobedo y otra en Monterrey. La prueba consistió en medir el lapso de tiempo que dura en transmitirse una trama y recibir su reconocimiento. Este lapso comprende el tiempo de transmisión, la duración del mensaje en el medio, el tiempo de procesamiento en el lado remoto y el tiempo que tarda en llegar el reconocimiento. El tiempo obtenido de la prueba fue alrededor de los 100 ms pero para tener mayor seguridad para transmisiones a larga distancia se decidió triplicar este tiempo.

El contador denominado *Contador por Agotamiento de Espera (CAE)*, cuenta el número de retransmisiones efectuadas. Este contador se incrementa cada vez que se agota el plazo del temporizador de entrada. El límite de este contador es de 3. Si se recibe respuesta (mensaje correcto ó comando *Nack*) dentro de este límite, el contador se restaura y el proceso continua normalmente, de lo contrario se ignora el mensaje se notifica dicho evento a la capa de Aplicación y se corta la conexión.

No obstante, dado que en las retransmisiones puede existir un número aleatorio de mensajes incorrectos (dañados en el medio), se decidió definir otro contador. Este contador denominado *Contador de Errores (CE)* mide el número de retransmisiones del mismo mensaje por recepción errónea. Dicho contador se incrementa cada vez que se recibe el comando *Nack*. El límite de este contador es de 3. Si se recibe un mensaje correcto dentro de este límite, el contador se restaura y la operación continua normalmente, en caso contrario se ignora el mensaje, se notifica a la capa de Aplicación y se corta la conexión.

Para obtener la eficiencia del canal se realizó un análisis similar al tratado en [7] para el rendimiento del protocolo y del cual se tomaron los siguientes parámetros:

H = Bits en la cabecera de la trama = 44  
 D = Bits de datos por trama = 2805  
 C = Capacidad del canal = 9600 bps

E = Probabilidad de que un bit se dañe =  $1 \times 10^{-9}$   
 U = Utilización del canal  
 T = Intervalo de temporización

Donde:

$$U = \frac{D}{H+D} \frac{(1-E)^{H+D} (1-E)^H}{\frac{1+TC}{H+D}}$$

El uso del canal para  $T=100$  ms,  $T=200$  ms y  $T=300$  ms es respectivamente 0.5881, 0.5633 y 0.5442. En la sección siguiente se explica el intercambio de los mensajes para establecer el enlace de comunicación entre ambas computadoras.

### 3.4. ESTABLECIMIENTO DE LA COMUNICACIÓN

El inicio de la comunicación lo puede realizar cualquier computadora: el monitor o las terminales. Para este propósito el monitor y cada terminal cuentan con los registros telefónicos de cada nodo. Estos registros permiten realizar la marcación automática a donde se requiera establecer la comunicación. En las terminales este proceso puede producirse cuando el operador lo requiera o cuando se produzca un disturbio. En el monitor solamente cuando el operador lo solicita.

El enlace se establece al detectar la señal portadora del lado remoto, la secuencia normal de mensajes entre el monitor central y las terminales es la siguiente:

- Con el objeto de comprobar el correcto estado de la línea, se especificó un comando de inicio de diálogo denominado TST. Este comando se envía al lado remoto y se confirma que la línea funciona adecuadamente cuando se recibe el acuse de recibo (comando *Ack*). Una vez efectuado esto, el monitor o la terminal se declara en línea para después enviar o recibir cualquier tipo de mensaje.
- La terminal y el monitor procesan cada mensaje recibido y contestan con la respuesta correspondiente. Como se ha dicho en caso de no recibir respuesta, se envía nuevamente el mensaje. El CAE cuenta los intentos en caso de que la ausencia de respuesta persista. El enlace se declara fuera de línea cuando se vence el CAE.
- Otra manera de dar de baja el enlace es mediante el contador CE. Cuando se detecta un error binario en un mensaje se descartan los octetos recibidos y se envía el acuse de recibo negativo (comando *Nack*), el cual a su vez activa la retransmisión del mensaje recibido

erróneamente. El CE mide la cantidad de comandos *Nack* recibidos consecutivamente y si exceden el límite del CE, se declara de baja el enlace.

- Si la comunicación no se cortó por los motivos anteriores, para terminar la sesión de comunicación se envía el mensaje del fin de comunicación, el cual da de baja el enlace.

### 3.5 ESTADOS DEL TRANSMISOR Y EL RECEPTOR

Cuando se establece el enlace, el dispositivo que tiene la iniciativa de establecer el diálogo se convierte automáticamente en transmisor mientras que el que recibe el mensaje TST pasa a ser el receptor. Con el propósito de explicar la operación de la capa de enlace cuando se transfieren los datos entre el monitor y la terminal, a continuación en las Figuras 3.3 y 3.4 se muestran los diagramas de transición de estado tanto para el receptor como para el transmisor.

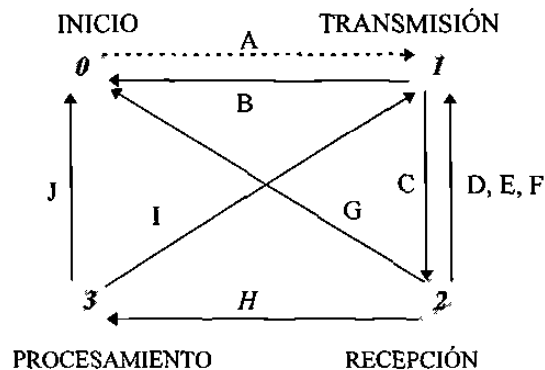


Fig. 3.3 Diagrama de transición de estado de la capa de enlace del transmisor

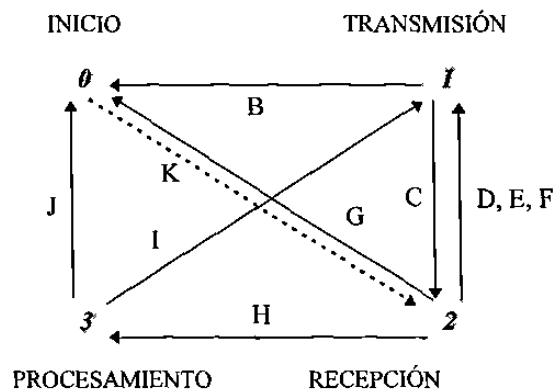


Fig. 3.4 Diagrama de transición de estado de la capa de enlace del receptor

Como se puede observar en las Figuras anteriores, la capa de enlace se puede encontrar en cuatro estados: inicio, transmisión, recepción y procesamiento. Las transiciones entre ellos dependen del control de errores explicado anteriormente y de la secuencia de los mensajes. La descripción de cada estado se define a continuación.

### **3.5.1 Estado 0: Inicio**

En este estado, el controlador (terminal o monitor) espera una condición de inicio. Esta se produce automáticamente al empezar un diálogo. En el transmisor, el mensaje de inicio se carga en la memoria del flujo de salida, se restaura el temporizador y los contadores, y se pasa al Estado 1 (ver Fig. 3.3). Por otro lado, el receptor espera el mensaje inicial, su recepción declara operacional la interfase externa, y se transita al Estado 2 (ver Fig. 3.4).

### **3.5.2 Estado 1: Envío de mensaje**

El controlador espera en este estado hasta que termina la transmisión completa del mensaje y guarda una copia de éste. A la vez, se inicializa el temporizador y se pasa al estado 2. Si se transmite el mensaje de fin de enlace se pasa al estado 0.

### **3.5.3 Estado 2: Recepción de mensaje**

En este estado, se espera la completa recepción del mensaje. Una vez recibido se restaura el temporizador y se procede a la verificación de datos con su correspondiente detector de errores (*Check sum*). Si existe error, se restaura la interfase externa, se pasa al Estado 1 y se transmite el comando *Nack*.

Cuando se recibe el comando *Nack*, se incrementa el CE, se pasa al estado 1 y se retransmite el mensaje guardado. El CE se restaura si se recibe el mensaje correcto dentro del límite de este contador. Cuando se alcanza el límite del CE se pasa al Estado 0.

Si el plazo del temporizador se agota sin haberse completado la recepción de la respuesta, se incrementa el CAE, se pasa al estado 1 y se transmite el mensaje guardado. Si se cumple el límite del CAE se pasa al Estado 0. Al recibir la respuesta correcta y sin error antes de vencerse el plazo de recepción se pasa al estado 3 y se indica a la capa de aplicación que ha llegado un mensaje a procesar.

### 3.5.4 Estado 3: Procesamiento del mensaje

La maquina espera aquí hasta que el mensaje recibido sea procesado completamente por la capa de aplicación. Esto se produce cuando se vacía completamente la memoria de la interfase interna. Si el mensaje recibido es un fin de sesión se pasa al Estado 0.

En este estado se atienden las peticiones de transmisión de la capa de aplicación. Si hay un mensaje a enviar, éste se desplaza de la capa de aplicación a la capa de enlace para iniciar su transmisión y se pasa al estado 1.

### 3.5.5 Transiciones de los dispositivos

Las transiciones entre los estados de las Figs. 3.3 y 3.4 son las siguientes:

- A) Inicialización de TE, CAE y CE, mensaje cargado en la memoria intermedia de salida.
- B) Envío del mensaje de fin de sesión.
- C) Transmisión completa; inicio del TE.
- D) Mensaje no recibido dentro del TE. Se trasmite el mensaje guardado.
- E) Recibo de *Nack*; incremento del CE. Se trasmite el mensaje guardado.
- F) Error en la estructura del mensaje. Se trasmite el comando *Nack*.
- G) Vencimiento de algún contador.
- H) Mensaje completo recibido sin error.
- I) Mensaje disponible puesto en la parte de transmisión.
- J) Recibo del mensaje de fin de sesión.
- K) Transito automático del receptor.

A continuación se presenta y se describe la operación de las funciones y subrutinas de programación más importantes de la capa de enlace. Dichas funciones se programaron en lenguaje “C y C++”.

### 3.6 FUNCIONES PARA EL MANEJO DE DATOS

Para el acceso y manejo de los datos de entrada y de salida se definieron tres arreglos. Dos de estos constituyen las interfases externa e interna, el tercero es el buffer de salida. En estos se procesa el flujo de la información. Estos se definen a continuación:

- Arreglo *bufentrada*. Tiene capacidad para almacenar 260 octetos obtenidos directamente del puerto serie.
- Arreglo *bufproces*. Almacena 260 octetos extraídos del vector anterior para revisar y procesar los datos.
- Arreglo *bufsalida*. Se emplea para cargar y transmitir los mensajes de la capa de aplicación. Su capacidad es de 260 octetos.

La capacidad de los vectores, se estableció tomando en cuenta el mayor número de datos contenidos en el campo de datos (3 cabecera + 255 datos + 1 detector de errores). El acceso a los datos contenidos en los vectores anteriores se lleva a cabo a través de las siguientes funciones.

#### 3.6.1 Captura de los datos

Con el objetivo de monitorear la entrada de datos por el puerto serie se creó una función denominada “*Interrupciones()*”, esta se muestra a continuación.

```
void interrupt Interrupciones(...)
{ disable();
  if ( ( inportb( dpuerto + IIR ) & RX_MASK ) == RX_ID )
    { if( fin_buf == TBUFFER ) fin_buf = 0;
      bufentrada[ fin_buf++ ] = inportb( dpuerto + RXR );
    }
  outportb( ICR, EOI );
  enable();
}
```

La función anterior indica, identifica y carga la entrada de datos del puerto serie. La indicación (interrupción) ocurre automáticamente cuando el bit 0 del registro IIR (ver características en el Apéndice A) cambia a cero. La identificación de entrada de datos se realiza cuando el bit 1 es cero y el bit 2 es uno. Al identificar esto, el octeto presente en el registro RXR se carga a las casillas del arreglo *bufentrada* indexada por la variable *fin\_buf*. En este arreglo, la carga de octetos se realiza de manera sucesiva conforme los datos van llegando. Una vez que se llega a su capacidad máxima se reinicializa el apuntador de manera que la información se cargue cíclicamente.

Por otra parte, para extraer los datos del vector *bufentrada* y copiarlos al vector *bufproces* con el propósito de revisarlos y procesarlos posteriormente, se creó la función “*puerto\_com>>*” mostrada a continuación.

```

serial &serial::operator>>( unsigned char &ch )
{ if( fin_buf == ini_buf )
  { ch = 255;
    return *this;
  }
  ch = bufentrada[ ini_buf++ ];
  if( ini_buf == TBUFFER ) ini_buf=0;
  return *this;
}

```

Como se puede observar la rutina anterior es una típica rutina de copiado de arreglos.

### 3.6.2 Recepción de los mensajes

En el programa principal cuando se carga el primer octeto de un mensaje al vector *bufentrada*, este octeto se compara con los tipos de mensajes definidos. Si el octeto corresponde a uno de ellos, los datos restantes del vector *bufentrada* son copiados al vector *bufproces* a través de la siguiente función denominada “*Recepción()*”.



```

Recepción( unsigned char ent )
{ int r = 1, rx_datos = 0, NDT = 2; bufproces[ 0 ] = ent;
  if( fin_buf != ini_buf )
    { puerto_com>>ent;
      bufproces[ 1 ] = ent; /* Determina el número de datos a copiar */
      if(ent) NDT = ent + 3;
    }
  while( !rx_datos )
    if( fin_buf != ini_buf )
      { puerto_com>>ent;
        bufproces[ ++r ] = ent;
        if( r == NDT) rx_datos = Rev_datos( NDT );
      }
  return rx_datos;
}

```

Esta función emplea las funciones: *Interrupciones()* y *puerto\_com>>*. Cuando se confirma que el primer octeto corresponde a uno de los mensajes, este dato se copia en la primer casilla del arreglo *bufproces*. Después se copia en la casilla siguiente el segundo dato, el cual determina el número de datos restantes a copiar, incluyendo el o los octetos de chequeo. La copia de estos datos se realiza consecutivamente sobre dicho vector.

Una vez copiados los octetos del mensaje completo, estos pasan a ser revisados. Para llevar a cabo esta revisión se elaboró la función “*Rev\_datos(NTD)*”. Donde NDT define el número de los datos a revisar en el vector *bufproces*. La función se presenta a continuación.

```

Rev_datos(int NTD)
{ unsigned int Rck1, Rck2 = 0, a = 21, i;
  Rck1 = ( int )( bufproces[ 0 ] + bufproces[ 1 ] ) & 0xFF;
  if( Rck1 == ( int )bufproces[ 2 ] ) /* Revisión de la cabecera */
    { a = 6;
      if( ( int )bufproces[ 1 ] ) /* Revisión del cuerpo de datos */
        { for(i = 3 ; i < NTD ; i++) Rck2 = ( ( int )bufproces[ i ] + Rck2 );
          if( (Rck2 & 0xFF) != ( int )bufproces[ NTD ] ) a=21;
        }
    }
  return a;
}

```

En esta función se realizan dos operaciones de suma binaria: una para el contenido de la cabecera y la otra para el campo de los datos. El resultado de cada suma se conjuga con el valor hexadecimal FF para quedarse con el octeto menos significativo del valor entero obtenido. El octeto resultante de cada suma se guarda en las variables *Rck1* y *Rck2* respectivamente. Después, estas variables son comparadas con los correspondientes comprobantes para el control de errores recibidos en el mensaje; *Rck1* con *bufproces[2]* y *Rck2* con *bufproces[NTD]*. Si la comparación es igual, la función regresa un valor de 6 (*Ack*) ratificando que el mensaje recibido fue correcto. En caso contrario, regresa un valor de 21 (*Nack*) señalando que el mensaje recibido es inválido. Cuando el mensaje no tiene campo de datos únicamente se revisa la cabecera.

### 3.6.3 Transmisión de los mensajes

Cuando la capa de aplicación dispone de un mensaje, los datos del mismo son puestos en el arreglo *bufsalida*. Después, en este arreglo, la capa de enlace crea el correspondiente comprobante para el control de errores tanto para la cabecera como para el campo de datos. Por último, el mensaje completo es enviado al puerto serie. La creación del comprobante de control de errores se realiza con las siguientes funciones denominadas *Check1()* y *Check2()*.

```

Check1()
{ int Ck1 = 0;
  Ck1 = bufsalida[ 0 ] + bufsalida[ 1 ];
  return( Ck1&0xFF );
}

Check2()
{ unsigned int i, Ck2 = 0;
  for(i = 3; i < ( bufsalida[1] + 3 ) i++)
  Ck2 = bufsalida[ i ] + Ck2;
  return ( Ck2 & 0xFF );
}

```

En estas funciones se llevan acabo dos sumas binarias sobre los datos cargados en el arreglo *bufsalida*; una para la cabecera y otra para el campo de los datos. El resultado de cada suma se conjuga con el valor hexadecimal FF para quedarse con el octeto menos significativo del valor entero obtenido. En el protector de errores de la cabecera (función *Check1()*) se suman los octetos de las casillas *bufsalida[0]* y *bufsalida[1]* y el resultado se guarda en

*bufsalida[2]*. Para el protector del campo de datos (función *Check2()*) se realiza la suma de los octetos contenidos desde la casilla *bufsalida[3]* hasta la casilla *bufsalida[bufsalida[1]+3]* y el resultado se guarda en la casilla siguiente (*bufsalida[bufsalida[1]+4]*). Una vez ya formado el mensaje completo se procede a enviarlo. Para el envío de cada octeto se creó la función “*puerto\_com<<*” mostrada a continuación.

```

serial& serial::operator<<( unsigned char x )
{ disable();
  while ( ( inportb( dpuerto + LSR ) & 0x30 ) != 0x00 );
  outportb( dpuerto + TXR, x );
  enable();
  return *this;
}

```

La función copia la primer casilla del vector *bufsalida* en una variable denominada *x*. Posteriormente, verifica que los bit 4 y 5 del registro LSR estén en cero lógico para confirmar que no exista desconexión en la línea y que el registro TXT esta vacío (disponible). Cuando esto sucede la variable *x* se copia al registro TXT y se envía el contenido de este registro por el puerto serie. Para enviar secuencialmente los octetos del mensaje completo contenidos en el vector *bufsalida* se elaboró la siguiente función denominada “*Transmisión()*”.

```

void Transmisión()
{ int i, rtx = 0;
  RTX:
  for( i = 0; i<bufsalida[ 1 ] + 3; i++ ) puerto_com<< bufsalida[ i ];
  if( bufsal[ 1 ] ) puerto_com<< bufsalida[ i ];
  i=0;
  /* Inicio de rutina de espera */
  while( fin_buf == ini_buf && (bufproces[ 0 ] >= M12 && bufproces[ 0 ] <= M15 ) )
  { delay( 10 );
    i++;
    if( i == LAP_RTX * 100 ) /* Lapso de retransmisiones (TE)*/
      { if( rtx++ < NUM_RTX ) goto RTX; /* Intentos de retransmisión (CAE)*/
        else { sal_enlace = 0;
              break;
            }
          }
    }
  }
}

```

El envío de los octetos del mensaje se lleva a cabo por medio de una rutina sucesiva de transmisión empleando la función “*puerto\_com<<*”. La rutina envía cada octeto contenido en las casillas del vector *bufsalida* por el puerto serie. La casilla *bufsalida[1]* define la cantidad de octetos a enviar. Cuando todos los octetos del mensaje son enviados, empieza una rutina de espera para la respuesta del lado remoto. Es aquí donde se inicializa el TE y se restablece el CAE, mencionado en la sección 3.5.1. Esta rutina establece lapsos de 10 milisegundos a la espera de alguna interrupción por entrada de algún dato en el puerto serial. Cuando los lapsos acumulan un tiempo de 3 segundos (*LAP\_RTX*), los datos alojados en el vector *bufsalida* son transmitidos nuevamente. Si se alcanzan 3 intentos (*NUM\_RTX*) de retransmisión el enlace se da de baja.

### CAPA DE APLICACIÓN

En este capítulo se trata el nivel superior del modelo de referencia OSI constituido por la capa de aplicación. Bajo esta capa es donde se llevan a cabo las funciones específicas de comunicación que constituyen el sistema. Para explicar la operación de esta capa, se presenta en las secciones siguientes la operación del monitor central y de las terminales, se muestra la descripción de los mensajes así como también la estructura del contenido de cada uno de ellos. Por otra parte se explica el proceso de transferencia de los registros de disturbios, la sincronización entre terminales, la medición remota de voltaje y corriente en las líneas de transmisión, el proceso para configurar la tarjeta de adquisición de datos, y por último la transferencia de texto.

Las capas de red, transporte, sesión y presentación, no se definen en este trabajo ya que no son utilizadas. Esto se debe a que no se requiere de direccionamiento por ser una topología punto a punto.

#### 4.1 DESCRIPCIÓN

Como se ha visto, las capas tratadas en los capítulos anteriores existen para brindar soporte a la de aplicación. La aplicación se compone de procesos cooperantes que se intercomunican mediante el uso de protocolos definidos en la capa de enlace. Estos procesos de aplicación constituyen la fuente y el destino último de los datos intercambiados. La capa de aplicación se ocupa de las funciones u operaciones del usuario final. Este último puede ser el operador o el programa en la unidad de control.

En un instante dado, solamente puede haber un mensaje en el enlace. Si un mensaje sale de la capa de aplicación, el envío del siguiente tiene que esperar hasta que la respuesta del anterior sea recibida completamente. Controlando los mensajes de esta manera, se puede detectar rápidamente una falla en el enlace. Si se recibe un aborto de la capa de enlace (tres intentos sin recibir respuesta), significa que el receptor no está respondiendo. Lo que significa que no hay continuidad en la línea, que el módem esté apagado o que exista una interrupción parcial de energía en la estación de trabajo.

La secuencia de sondeo (*Polling sequence*) se diseñó para comunicarse rápidamente con cualquier dispositivo, asegurando la rápida comunicación cada vez que se detecte un disturbio en las líneas de la subestación. A continuación se presentan las funciones disponibles de operación tanto del monitor central como de las terminales.

#### **4.2 OPERACIÓN DEL MONITOR CENTRAL**

Las operaciones disponibles en el monitor son las siguientes:

- Establecer comunicación con la terminal deseada. Para este propósito, el monitor realiza la llamada correspondiente a través del registro telefónico ya mencionado.
- Consultar los parámetros de operación de las tarjetas de adquisición de datos de las UTR's. Estos parámetros podrán ser reconfigurados bajo límites ya preestablecidos. A través de cambios en la configuración, se pueden cambiar remotamente los tipos de algoritmos de detección de disturbios.
- Detener o activar el proceso de detección de disturbios en cada terminal.
- Consultar las mediciones presentes de voltaje y de corriente en la línea de transmisión.
- Sincronizar el propio reloj del sistema o el de la terminal. Esto puede ser realizado por el operario o bien de modo automático.

- Establecer el dialogo entre operadores, para enviar y recibir mensajes en texto.
- La comunicación finaliza cuando el operario o la aplicación así lo determinan, dando de baja la conexión.

Utilizando el método de sondeo secuencialmente con todas las terminales, es posible contar con un estado global de operación y enviarles comandos cada vez que se desee. De esta manera se monitoriza la actividad de todas las terminales y se ordenan cambios cuando son necesarios.

El registro telefónico de todos los nodos está en el archivo "LISTERM.DAT", el cual se encuentra en el subdirectorio "C:\DETECTOR\REGFALLA". Cuando inicia el programa de comunicaciones los registros de dicho archivo se cargan a un arreglo de variables con las cuales se realizan las marcaciones del módem. Los registros contienen el nombre y el teléfono de cada terminal.

### **4.3 OPERACIÓN DE LAS TERMINALES**

La operación de las terminales es la siguiente:

- En caso de que se detecte un disturbio en las líneas de una subestación, éste se registra en el disco duro e inicia el programa de comunicación automáticamente.
- Una vez realizado lo anterior, se avisa de la producción del disturbio al monitor. Posteriormente éste solicita los registros del disturbio y estos le son enviados.
- Cuando los registros han sido completamente transferidos, el enlace se da de baja.
- El operador también puede comunicarse al monitor para solicitar comunicación textual o solicitar una sincronización del monitor si lo desea. Al terminar la comunicación el operador da baja el enlace.

La detección de un disturbio requiere la más alta prioridad, motivo por el cual al finalizar la comunicación con el monitor central, la terminal continua detectándolos. A continuación se describe cada uno de los mensajes utilizados en la red.

#### 4.4 DESCRIPCIÓN DE LOS MENSAJES

Para el diálogo entre el monitor y las terminales se definieron 17 mensajes clasificados en peticiones y envíos. En la Tabla 4.1 se presenta la descripción y uso de ellos.

Tabla 4.1 Descripción y uso de los mensajes

<i>Mensaje</i>	<i>Descripción</i>	<i>Uso</i>
1	Envío de prueba de sincronía del monitor	Sincronización
2	Peticion de sincronía de la terminal	
3	Envío de sincronía	
4	Envío de la hora	
5	Peticion de la hora	
6	Peticion de la configuración de la tarjeta	Configuración de la tarjeta de adquisición de datos
7	Envío de configuración de la tarjeta	
8	Envío de reconfiguración de la tarjeta	
9	Peticion de la medición actual en la línea	Medición en la línea de transmisión
10	Envío de la medición actual en la línea	
11	Peticion de transferencia de texto	Transferencia de texto
12	Envío de aviso de falla presente	Transferencia de archivos de falla
13	Envío de datos del archivo de falla	
14	Peticion de la próxima trama del archivo	
15	Envío de fin del archivo de falla "N"	
16	Envío de fin de los archivos pendientes de falla	
17	Envío de fin del enlace	Fin de la comunicación

Como se aprecia en la Tabla anterior, para realizar el proceso de sincronización se emplean los mensajes 1,2,3,4 y 5. Los mensajes 6 y 7 se utilizan para la configuración de la tarjeta de adquisición de datos. La medición presente en la línea de transmisión requiere de los mensajes 9 y 10. El mensaje 11 se destina para la petición de la transferencia de texto.



Los mensajes 12,13,14,15 y 16 son empleados para la transferencia de los registros de disturbios y el 17 es para finalizar el enlace. En la Tabla 4.2 se presenta el contenido de la trama de cada mensaje.

Tabla 4.2 Contenido de mensajes

Cabecera de la trama			Campo de datos	
Tipo de mensaje	Cantidad de datos	Control de la cabecera	Información	Control de los datos
1	0	1	*	*
2	0	2	*	*
3	4	7	Datos	Control
4	4	8	Datos	Control
5	0	5	*	*
6	0	6	*	*
7	8	15	Datos	Control
8	8	16	Datos	Control
9	0	9	*	*
10	32	42	Datos	Control
11	0	11	*	*
12	9	21	Datos	Control
13	255	12	Datos	Control
14	0	14	*	*
15	1-256	Control	Datos	Control
16	1-256	Control	Datos	Control
17	0	17	*	*

Nota: Las celdas que contienen '\*' carecen de datos por tanto no tienen uso.

Como puede verse en la Tabla 4.2, la máxima cantidad de datos contenidos en la trama es de 255, este valor corresponde al mayor valor contenido en el octeto número 2 de la cabecera de datos.

El valor y la posición del octeto de control de datos es variable y depende del contenido de los datos de cada mensaje. A continuación en la Tabla 4.3 se especifica la estructura de datos de cada mensaje y el contenido de cada uno de sus campos.

Tabla 4.3 Contenido del campo de datos

Tipo de mensaje	Estructura	Campos	Cantidad de Octetos	Tipo de Variable
3 4	Tiempo del sistema: - Modificar - Actual	Tiempo del BIOS.	4	long
7 8	Configuración de la tarjeta: - Actual - Modificar	Canales en operación. Sensibilidad del umbral. Tiempo de grabación de postfalla. Hora de verif. de los reg. pend.	1 4 2 1	char float int char
10	Medición de las líneas:	Línea 1: Voltaje (A,B,C,N). Corriente (A,B,C,N). Línea 2: Voltaje (A,B,C,N). Corriente (A,B,C,N).	2/fase 2/fase 2/fase 2/fase	int int int int
12	Aviso del disturbio	Const. % por trama del registro. Número de registro. Número de registros pendientes. Número de nodo.	4 3 1 1	float char u. char u. char
13	Datos del registro	Datos.	256	u. char
15	Datos del fin del registro 'N'	Datos.	1-256	u. char
16	Fin de registros pendientes	Datos.	1-256	u. char

En el archivo a transferir, con la finalidad de aumentar la cantidad de datos en la trama para reducir la cantidad de tramas a enviar, se realizó una prueba con un archivo binario de 130 Kbs. Para ello, en la cabecera se emplearon dos octetos para indicar la cantidad de datos en la trama. La transferencia se realizó a una velocidad de 9600. Los resultados obtenidos se muestran en la Tabla 4.4.

Tabla 4.4 Comparación de dos transferencias de un mismo archivo con diferente tamaño en la trama

Maxima cantidad de datos puros por trama	255	1,024
Datos para el control de la trama	4	5
Relación de eficiencia por trama (datos puros / datos totales)	0.9845	0.9951
Tramas empleadas	510	127
Tiempo de transferencia (segs)	144.65	143.11
Tiempo promedio por trama (segs)	0.28	1.12
Tiempo mínimo para retransmitir una trama inválida (segs)	0.3	1.2
La probabilidad de que una trama se transmita correctamente es: $(1 - P)^n$ , donde P es la tasa de error por caracter (errores independientes) y n el numero de caracteres por trama.	$P=10^{-6}$ $P=10^{-5}$ $P=10^{-4}$ 0.9997 0.9974 0.9744	$P=10^{-6}$ $P=10^{-5}$ $P=10^{-4}$ 0.9989 0.9897 0.9022

Como se puede apreciar en la Tabla anterior la máxima cantidad de datos es de 1,024, sin embargo la probabilidad de error en la transmisión también aumenta. Por otra parte, con mensajes más largos se requiere también aumentar el límite de tiempo del temporizador de entrada, debido al mayor tiempo de procesamiento empleado por trama del lado receptor. Los resultados de las pruebas efectuadas entre Monterrey, Escobedo y San Nicolás, mostraron que la trama más corta es más eficiente. Por tal razón se determinó utilizar un solo octeto para el número de datos por trama.

#### **4.5 TRANSFERENCIA DE REGISTROS DE DISTURBIOS**

La detección de disturbios eléctricos en las subestaciones se lleva a cabo por un programa que utiliza algoritmos de procesamiento de señales. Éste programa, instalado en las terminales, también controla la operación de la tarjeta de adquisición de datos y se encarga de establecer la comunicación con el monitor central cuando se requiere.

Cuando un disturbio se detecta, los datos de éste así como la hora, fecha y duración se guardan en un archivo. El nombre de este archivo se designa con el número del registro presente seguido del lugar donde ocurrió el disturbio. Al terminarse la grabación se escribe su número de registro en otro archivo denominado "FALLASPE.DAT", en el subdirectorío "C:\DETECTOR\REGFALLA", el cual contiene los archivos no transmitidos. Posteriormente el programa ejecuta la rutina de comunicaciones para enlazarse con el monitor central y transferir el archivo. Al terminar la transferencia, se borra el número de registro del archivo "FALLASPE.DAT".

El proceso de la transferencia de un registro se muestra en la Fig. 4.1. Como puede apreciarse, primero se comprime el archivo por medio del compresor de archivos *pkzip.exe*. Después se transmite el mensaje 12 para avisar que el presente archivo será enviado. Al llegar este mensaje al monitor, éste inmediatamente crea un archivo temporal en el que se

registrarán secuencialmente los datos comprimidos. Posteriormente, se inicia la rutina de intercambio de datos con los mensajes 13 y 14 hasta transferir el archivo completo. En dicha rutina, el monitor envía el mensaje 14 para solicitar una trama, al que la terminal contesta con el envío del mensaje 13, el cual contiene la trama de datos. El número de intercambios con estos dos mensajes es variable y depende de la longitud del registro.

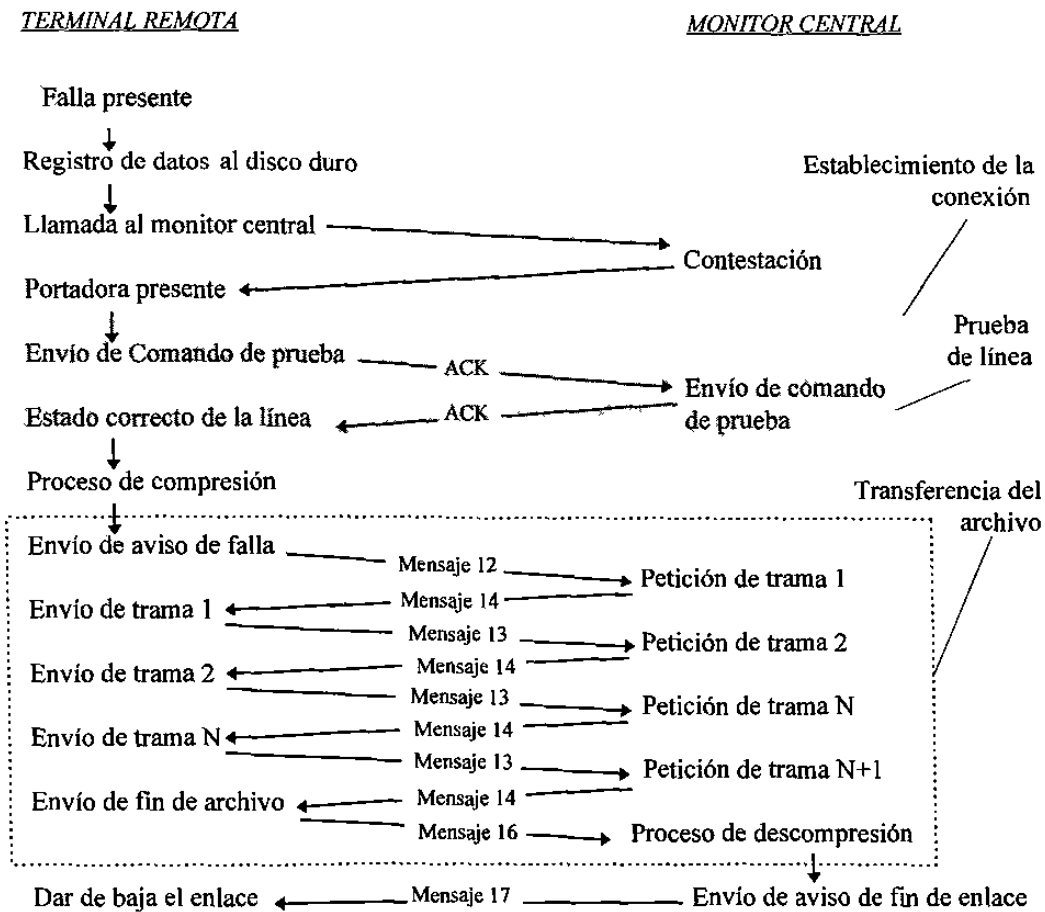


Fig. 4.1 Proceso de transferencia del archivo de disturbio

Con el mensaje 16, la terminal notifica el término de envío de datos. Cuando el monitor lo recibe, éste descomprime la información del archivo temporal creado con el descompresor *pkunzip.exe*, y ésta información se guarda en un subdirectorio denominado

“C:\DETECTOR\FALLAS\”. Posteriormente, se borra el archivo comprimido, y por último se envía a la terminal el mensaje 17 para dar baja el enlace y restablecer el algoritmo de detección.

Si al momento de realizar el enlace con el monitor éste no se encuentra disponible (ver Fig. 4.2), se realizarán tres intentos para conectarse, con lapsos de 30 segundos entre ellos. Si no hay éxito en la conexión, un mensaje aparecerá en la pantalla notificando dicha situación. Los lapsos entre cada reintento y la cantidad de estos son configurables y podrán cambiarse por el operador si se desea.



Fig. 4.2 Intento de establecimiento de conexión con el monitor central

Con la finalidad de que los archivos no transferidos se acumulen, se establece un tiempo fijo al día (definido por el usuario) para que diariamente se revise el archivo “FALLASPE.DAT”, y se transfieren los archivos pendientes. Otra alternativa adicional, es que al presentarse un disturbio se transfieran también todos los archivos pendientes.

En la Fig. 4.1 se presentó el proceso de transferencia para un solo archivo. Si hay que transferir más de un archivo, el fin de cada uno de ellos se marcará con el envío del mensaje 15, el cual activa a su vez la descompresión del precedente y la apertura de un archivo nuevo para recibir los datos del siguiente. La terminación de la transferencia de todos los archivos, se indica por el envío del mensaje 16.

El tiempo para transferir cada archivo depende de su tamaño. Este dependerá del lapso de la grabación y del porcentaje de compresión. El tiempo promedio para cada registro con 10 segundos de grabación de señales oscila entre 55 a 75 segundos (a 9600 bps).

Para disponer de una referencia de tiempo común en todas las terminales es necesario sincronizarlas. Este proceso de sincronización es muy importante ya que las mediciones fasoriales requieren de una base común para comparar las fases. A continuación se presentan algunos procesos de sincronización.

#### **4.6 SINCRONIZACIÓN DE LAS TERMINALES**

La sincronización de las terminales representa un factor muy importante al interpretar los datos grabados. Cuando un disturbio eléctrico se presenta, se introduce en el registro de los datos el instante de tiempo en el que se detectó. Este dato se obtiene del reloj de la terminal local. Cuando un disturbio es detectado simultáneamente en varias terminales, es de vital importancia que estén sincronizadas para hacer correctamente estudios comparativos [11].

Para realizar dicha sincronización, se emplean varios métodos, la mayoría de los cuales requiere de una fuente maestra de tiempo. Dicha fuente puede enviarse por radio AM, vía microonda, fibra óptica o satélite. De esta manera se dispone del mismo tiempo en los equipos registradores [12].

Con miras a la confiabilidad de la sincronización, el empleo de fibra óptica o satélite es el más adecuado actualmente [12]. Sin embargo, un sistema con fibras ópticas es muy difícil de justificar por el alto costo de los equipos para esta única aplicación. El uso de fibra óptica es una buena elección cuando esta ya disponible en todas las terminales [13].

#### 4.6.1 Sincronización con GPS's

El sistema NAVSTAR operado por el departamento de la defensa de los Estados Unidos es un sistema de posicionamiento global (*GPS: Global Positioning Systems*) utilizado para aplicaciones civiles. Este sistema dispone de una constelación de 24 satélites y provee información de posicionamiento preciso en todo el planeta [14, 15]. Estos satélites se encuentran en 6 órbitas no estacionarias (4 satélites/órbita) con alturas de 10,900 millas sobre la superficie de la tierra. El sistema suministra un pulso de tiempo de acceso común, con un microsegundo de precisión sobre cualquier localización de la tierra. Para la adquisición de este pulso se requiere de una antena visible al satélite y un receptor.

La sincronización de las terminales se alcanza por la distribución simultánea de un conjunto de pulsos precisos por medio de GPS's. Para la adquisición del temporizador de pulsos, únicamente uno de los satélites necesita ser visible a la antena de la terminal. Los receptores GPS proveen una señal de 1 pps (un pulso por segundo) y una etiqueta de tiempo, con el año, día, hora, minuto y segundo. Este tiempo puede ser el local, o el UTC (*Universal Time Coordinated*). La señal de 1 pps es usualmente dividida por un oscilador con enganche de fase dentro del número requerido de pulsos por segundo. El tiempo del temporizador del GPS, junto con el número asociado del contador, son asignados al archivo como etiqueta identificadora del inicio del disturbio.

La experiencia adquirida en disponibilidad y dependencia al emplear los pulsos del GPS, ha sido favorable hasta ahora [14]. Sin embargo, cabe también señalar que estos sistemas de sincronización son costosos.

Con la finalidad de obtener una adecuada sincronización en las terminales y evitar el empleo de GPS's, a continuación se presenta un proceso de sincronización para las terminales utilizando los recursos de operación de la red telefónica.

#### 4.6.2 Sincronización por medio del monitor central

Las computadoras (PC's) poseen varios relojes y temporizadores que operan a distintas velocidades y ejecutan funciones diferentes. Algunos de estos están diseñados intrínsecamente dentro del circuito de la computadora, y su operación es independiente del *software* de control. Otros están diseñados para soportar funciones de temporización, y su operación se controla por *software* a través de los servicios básicos de entrada/salida del sistema (*BIOS: Basic Input/Output System*) o bien directamente programando el *hardware* [4,5].

El acceso al temporizador del sistema de la PC es derivado de un oscilador de frecuencia básica de 14.31818 Mhz, el cual es comúnmente usado en los circuitos de televisión. Los pulsos generados de este temporizador se obtienen dividiendo esta frecuencia básica entre un factor de doce y un contador de 16 bits, esto es:

$$\text{Pulsos del} \\ \text{temporizador} = \frac{\text{Frecuencia del oscilador básico}}{(\text{Factor de conversión})(\text{Contador de 16 bits})} = \frac{14.31818 \text{ Mhz}}{(12)(65,536)} = 18.2 \text{ pulsos/seg.} \\ \text{del sistema}$$

Esto muestra que la resolución temporal del reloj interno es de 54 ms. Este lapso es inadecuado para sincronizar las UTR's ya que nuestra aplicación requiere una resolución temporal mínima de 1.04 ms, considerando que la frecuencia fundamental de las señales es de 60 Hz y que éstas se toman a 16 muestras por ciclo, esto es:

$$\text{Intervalo de tiempo} \\ \text{entre muestras} = \frac{1}{(\text{Frec. de la señal})(\text{Frec. de muestreo})} = \frac{1}{(60 \text{ Hz})(16 \text{ muestras/ciclo})} = 1.04 \text{ mseg/muestra} \\ \text{de la señal}$$



La limitación anterior hace necesaria una referencia de tiempo más precisa en cada terminal. La tarjeta de adquisición de datos cuenta con un reloj que controla el muestreo de datos cuya resolución temporal es de 1.3 microsegundos. Sin embargo, el acceso a este temporizador no permite lectura o escritura. Actualmente existen en el mercado tarjetas de sincronización que operan independientemente de algún sistema de sincronía (*GPS*, *GOES*, etc.) y tienen resoluciones cercanas a 1 microsegundo.

Una solución de sincronización consiste en implantar en el monitor y en cada terminal una tarjeta de sincronía modelo GPS-PC [15]. El tiempo de la tarjeta del monitor servirá para sincronizar las tarjetas de sincronía de las terminales a través de la red. De esta manera las terminales podrán registrar disturbios con la misma base de tiempo que tenga el monitor.

Tres son los procesos para llevar a cabo la sincronización en las terminales: que el monitor la efectúe automáticamente con todas, que el usuario la realice con alguna terminal y que la terminal la solicite al monitor de manera automática o por el usuario. La manera automática requiere que el usuario defina cada cuantos días se lleve a cabo la sincronización. A continuación en la Fig. 4.3 se presentan estos procesos:

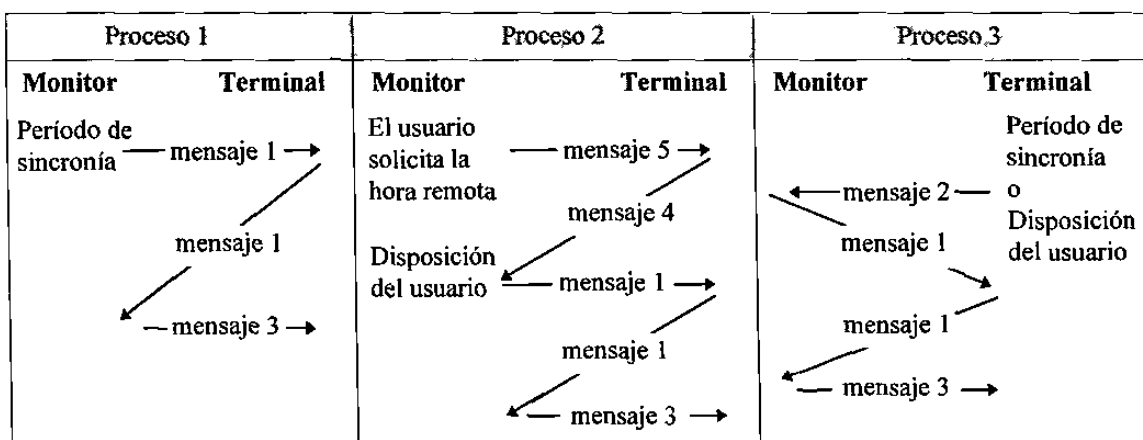


Fig. 4.3 Procesos de sincronización

#### **4.6.2.1 Proceso 1: Sincronía automática por el monitor**

En este proceso, el monitor envía el mensaje 1 a una terminal, guardando el tiempo de la tarjeta de sincronía en ese instante. Cuando la terminal lo recibe ésta lo regresa al monitor. El monitor al recibirlo toma nuevamente el tiempo de la tarjeta y le resta el tiempo anterior. El resultado obtenido es dividido entre 2 para obtener el lapso  $T_m$ , que es lo que tarda el mensaje en llegar a la terminal. Por último, se toma el tiempo de la tarjeta, se le añade el lapso  $T_m$  y el tiempo resultante se envía en el mensaje 3 a la terminal. Cuando ésta lo recibe configura el reloj con este dato. Este proceso supone que el tiempo de procesamiento de los mensajes y el tiempo de configuración del reloj es muy pequeño. Este proceso se repite con todas las terminales.

#### **4.6.2.2 Proceso 2: Sincronía por el usuario en el monitor**

Para realizar este segundo proceso, el usuario desde el monitor envía el mensaje 5 para solicitar a la terminal deseada el tiempo de la tarjeta. Cuando la terminal lo recibe, ésta envía dicho tiempo en el mensaje 4. El monitor al recibirlo, presenta en pantalla el tiempo de la terminal. En caso requerir una sincronización se solicita y el proceso 1 se efectúa con la terminal deseada.

#### **4.6.2.3 Proceso 3: Petición de sincronía por la terminal**

Cuando se cumple el periodo de días para llevar la sincronía en la terminal, ésta la solicita enviando el mensaje 2 al monitor. El monitor al recibirlo realiza el proceso 1. Si el usuario de la terminal desea realizar la solicitud de sincronía, el mensaje 2 se envía cuando él lo requiera.

### **4.7 CONFIGURACIÓN DE LA TARJETA DE ADQUISICIÓN DE DATOS**

Otro uso de la red, es poder controlar la operación de la tarjeta de adquisición de datos de manera remota. Esta aplicación permite al usuario dar de alta o de baja algún canal de la

tarjeta por razones de mantenimiento, establecer el lapso para la grabación del disturbio y definir el umbral de sensibilidad de detección requerido.

Para llevar a cabo esto, el usuario del monitor establece conexión con la terminal que él requiera. Cuando ésta se encuentra en línea, podrá enviar a la terminal el mensaje 6 para solicitar la configuración de la tarjeta de adquisición. Al recibirlo, la terminal envía al monitor su configuración en el mensaje 7. El monitor a su vez presenta en pantalla dichos parámetros. Si el usuario requiere modificar esta configuración, lo podrá realizar a través del mensaje 8. La terminal al recibirlo, configura la tarjeta con los parámetros contenidos en este mensaje.

En los parámetros de configuración de la tarjeta, el lapso de grabación del disturbio es un valor entero comprendido entre 1 y 65534 segundos. El límite de umbral del detector es un valor fraccionario en el rango de 1.0 hasta 2048.0 niveles de sensibilidad. La tarjeta cuenta con dos canales para las líneas de medición éstos pueden ser únicamente activados o desactivados.

#### **4.8 TRANSFERENCIA DE TEXTO**

La transferencia de texto es otra de las aplicaciones que puede ser usada tanto por el usuario del monitor como por los de las terminales. El usuario que requiera utilizar esta aplicación deberá establecer conexión con la PC deseada, ya estando en línea podrá solicitar comunicación textual (*chat mode*) al usuario remoto enviando el mensaje 11. El diálogo opera en forma bidireccional. Si por alguna razón no se encuentra el destinatario, el solicitante podrá generar un timbrado similar al de un teléfono convencional, con el fin de que alguien le conteste. Para finalizar la comunicación, cualquiera de los usuarios lo puede realizar enviando el comando ESC.

### EVALUACIÓN

En este capítulo se presenta el estándar PROFIBUS (*Process Field Bus*), enfocado al campo de la automatización, con el objetivo de realizar un estudio comparativo con la red de comunicaciones tratada en los capítulos anteriores. No es una descripción exhaustiva porque esto daría lugar a un volumen demasiado extenso y nos desviaría de nuestro propósito meramente comparativo. Por tanto, sólo se tratan los aspectos más relevantes para realizar dicha comparación. Estos aspectos se refieren a la operación de las capas de aplicación, de enlace y la física. Cabe destacar que dicho estándar contempla tres protocolos distintos para el tipo de aplicación a emplear: Especificación de mensajes de *fieldbus* (FMS), dispositivos periféricos (DP) y automatización de procesos (PA). Aquí solamente nos referiremos al primer protocolo por ser el que mejor se adapta para el propósito de comparación.

#### 5.1 CAMPO DE APLICACIÓN Y PROPÓSITO DEL ESTÁNDAR PROFIBUS

Los *fieldbuses* seriales son métodos de comunicación para intercambiar información entre sistemas de automatización y dispositivos de campo distribuidos. Miles de exitosas aplicaciones han aprobado que el uso de tecnología *fieldbus* ahorra hasta un 40% en costo de cableado y mantenimiento comparado con la tecnología convencional [16].

PROFIBUS es un estándar de los *fieldbuses* diseñado para arquitectura de redes con aplicaciones en procesos de automatización, operado a través de un simple Bus (canal

principal de interconexión) de interfase con un óptimo comportamiento en tiempo real [17,18].

Los propósitos del estándar son: permitir integrar diversos componentes digitales en un sistema de control distribuido, disminuir el costo de los dispositivos de interconexión que ofrecen multivendedores y garantizar una confiable comunicación en lo que usualmente se llama “*Open Systems Interconnection, OSI*” [2,3,7,9].

El estándar ofrece flexibilidad y permite implementación para diferentes aplicaciones. PROFIBUS tiene tres protocolos para la correspondiente aplicación. A continuación en la Tabla 5.1 se muestra cada protocolo y su aplicación.

Tabla 5.1 Familia PROFIBUS

PROFIBUS-DP	PROFIBUS-FMS	PROFIBUS-PA
Automatización de Fabricas	Automatización y propósitos generales	Automatización de Procesos

En el apéndice E se presenta la arquitectura y descripción de cada protocolo, así como sus respectivas capas basadas en el modelo de referencia OSI. De dicho apéndice se tomaron los aspectos más relevantes para llevar a cabo la evaluación mostrada a continuación.

## **5.2 COMPARACIÓN ENTRE EL ESTÁNDAR PROFIBUS Y LA RED DE COMUNICACIONES PRESENTADA EN ESTA TESIS.**

A continuación en la Tabla 5.2 se comparan algunas características de la red tratada en esta tesis y el estándar PROFIBUS.

Tabla 5.2 Comparación del estándar PROFIBUS contra la red de esta tesis.

Características	Red tratada	Estándar PROFIBUS
Topología	Punto a punto	Bus lineal
Número de estaciones	2 por cada enlace	32 por cada segmento, hasta 127 estaciones con repetidores.
Máximo tamaño de las tramas de datos	258 octetos	266 octetos
Medios de transmisión	RS-232C empleando módems y líneas telefónicas	RS-485, IEC1158-2, fibra óptica
Distancia	Definida por la red telefónica	Para RS-485: 1.2 Kms a 9.6 Kbps. 100 mts a 1.2 Mbps. Para F.O: 24 Kms a 9.6 Kbps. 4.2 Kms a 1.5 Mbps.
Velocidad	Depende de la relación S/N en la línea telefónica	Depende de la longitud y el medio de transmisión empleado
Métodos de arbitraje	Ninguno	<i>token passing</i> y maestra-esclava

De la Tabla anterior se concluyen los siguientes puntos:

- PROFIBUS permite conectar y controlar hasta 127 dispositivos en todo el Bus. Sin embargo, la longitud del Bus está limitada y depende de la velocidad de transmisión de los datos. En la versión 1, utilizando RS-485 como medio de transmisión, la máxima longitud es de 1.2 Km. para operar con velocidades de 9.6 Kbps. El empleo de fibra óptica hace que la longitud se extienda hasta 24 Km. No obstante, en nuestro caso, dicha distancia es insuficiente para interconectar los dispositivos de las subestaciones con el CENACE.
- En PROFIBUS la velocidad de los datos esta definida por la cantidad de estaciones conectadas en el bus y la distancia entre ellas, mientras que en la red tratada dicha velocidad se establece de acuerdo a la relación señal /ruido presente en la línea telefónica.

- PROFIBUS controla el acceso al medio por un método híbrido. Para la comunicación entre EM's emplea el método "token passing", mientras que para la comunicación entre EM y EE utiliza el método "maestra esclava". Cabe destacar que, en la red de esta tesis no se necesitó de ningún método de arbitraje (*token passing, CSMA maestra-esclava, sondeo cíclico, etc.*) para el tránsito de los datos, ya que por ser una comunicación punto a punto no existe direccionamiento entre más de dos estaciones.
- La desventaja de la red propuesta en esta tesis es que únicamente se permite conectar a dos estaciones por enlace. Esto evita que una tercera estación pueda comunicarse con alguna de las dos estaciones que están transfiriendo un archivo o bien que sus usuarios se encuentren comunicando.
- La comunicación múltiple entre estaciones sería conveniente si se deseara la grabación simultánea en todas las estaciones en el instante que un disturbio sea detectado por cualquier estación. Para esto, la estación detectora solicitaría simultáneamente a las terminales restantes la grabación y el envío de sus registros. La finalidad de este registro simultáneo permitiría a los ingenieros saber el comportamiento global del sistema eléctrico cuando al menos un disturbio se presente en una estación. Esta comunicación múltiple obliga el uso de una topología estrella con medios de transmisión rápidos como la fibra óptica.
- PROFIBUS tiene características técnicas y funcionales para interconectar dispositivos digitales con baja operación de ejecución en la comunicación, es decir la comunicación en ellos es frecuente pero el contenido es muy corto, por ejemplo controladores lógicos programables (PLC's), controladores numéricos (CN), transmisores, actuadores, sensores etc. Sin embargo, en nuestra situación dicho estándar presenta poca eficiencia para conectar sistemas o equipos que requieran una comunicación poco frecuente pero voluminosa. Tal es el caso de los detectores de disturbios en las subestaciones.

- La implementación de el estándar PROFIBUS resulta válido para aquellos sistemas que requieran de un control simultáneo de la operación de múltiples dispositivos de automatización, trabajando dentro de un rango de operación definido. La incorporación de este estándar constituiría un cuello de botella para la comunicación requerida en la transmisión de señales de disturbios.



### CONCLUSIONES

#### 6.1 CONCLUSIONES

La capacidad para registrar el evento de un disturbio eléctrico en las líneas de transmisión, permite a los ingenieros estudiar el comportamiento de las oscilaciones electromecánicas del sistema, con la consecuente posibilidad de detección, corrección o eliminación de factores que los producen. Además, los datos obtenidos con el paso del tiempo permiten la elaboración de estadísticas de gran valor en la validación de los modelos eléctricos de la red de distribución.

El empleo de múltiples sistemas registradores de disturbios eléctricos, en distintos puntos de medición de la red de distribución, ha propiciado el diseño de la presente red de comunicaciones para centralizar la información registrada con propósitos de estudio y análisis de dichos fenómenos.

El diseño de dicha red constituye el tema central de esta tesis. Con ella, el problema de la transferencia y concentración de registros queda resuelto satisfactoriamente. La transferencia de cada registro se realiza automáticamente al momento de la detección de cada disturbio. En esta transferencia se emplea un protocolo de parada y espera, con comprobantes para el control de errores binarios en la transmisión.