

2 ej^o
43



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

**INTRODUCCION AL TELEPROCESO Y
CONCEPTOS TEORICOS DEL DISEÑO
DE REDES DE COMUNICACION
DE DATOS**

T E S I S
QUE PARA OBTENER EL TITULO DE
LICENCIADO EN MATEMATICAS
P R E S E N T A :
JOSE MAURICIO TORRES ROMERO

MEXICO, D. F.

1986



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

- Conversión de Señales : Modulación	20
Demodulación	
- Convertidores de Señales	20
- Canal y Liga de Comunicación	20
- Modos de utilización de una facilidad de comunicación: Transmisión Simplex	20
Transmisión Semi-duplex	
Transmisión Duplex	

CAPITULO III

MODOS DE TRANSMISION	22
- Definición	22
- Modos de Transmisión: En Serie	22
En Paralelo	
- Submodos de la transmisión en Serie:	23
Transmisión Asíncrona	
Transmisión Síncrona	
Transmisión Isócrona	

CAPITULO IV

TIPOS DE ENLACE	27
- Tipos de Enlace: Punto a punto	27
Multipunto	
- Funciones a ejecutarse en un Enlace:	28
Invitación (Polling)	
Direccionamiento	

CAPITULO V

INTERFASES

-Definición, señales que se manejan y funciones básicas.	31
- Ejemplo de interfases para los distintos modos de transmisión:	31
Interfase en serie RS-232	33
Interfase paralela Centronix	38

CAPITULO VI

PROTOCOLOS

- Definición	39
- Categorías de protocolos:	40
Orientados a caracter	
Orientados a contador de byte	
Orientados a bit	
- Ejemplo de protocolos de cada una de las categorías:	
6.2 BSC (Binary synchronous, Orientado a caracter)	42
6.3 HDCCMP (Digital Data Communication Message Protocol, Orientado a contador de byte)	48
6.4 SDLC (Synchronous Data Link Control, Orientado a bit)	52

CAPITULO VII

CAUSAS, TIPOS Y TRATAMIENTO DE ERRORES	58
- Definición y categorías de Ruido	58
- Métodos de detección de errores:	60
VRC (Vertical Redundancy Check)	61
LRC (Longitudinal Redundancy Check)	62
CRC (Cyclic Redundancy Check)	63

CAPITULO VIII

COMPONENTES Y FUNCIONES A EJECUTARSE DENTRO DE UN SISTEMA DE TRANSMISION DE DATOS	70
8.2 Procesadores de comunicación	70
8.3 Procesamiento de las funciones de comunicación	74
8.4 Switcheo de Mensajes	76
8.5 Control del manejo de la red	78
8.6 Control de errores y diagnósticos	79
8.7 Software de comunicaciones:	80
Métodos de acceso	
Emuladores	
8.8 Emuladores	82
8.9 Módems	88
8.10 Multiplexores (Multiplicadores)	93
8.11 Concentradores	97

CAPITULO IX

PUNTOS A SER CONSIDERADOS DENTRO DE LA ARQUITECTURA DE UNA RED

- Arquitecturas organizadas	94
- Trayectorias de acceso	100
- Funciones que debe proporcionar una red de tele- proceso	100
- Arquitectura SNA	103

CAPITULO X

CONCEPTOS DE TEORIA DE REDES Y GRAFICAS 106

- Factores que influyen en el diseño de una topología	106
---	-----

10.2 Definiciones : 109

Gráfica, Adyacencia, Trayectoria, Diámetro,
circuito, conectividad, Arbol Generador,
Flujo Factible

10.3 Trayectorias disjuntas 114

CAPITULO XI

DISEÑO DE REDES CENTRALIZADAS 118

11.1 Conexiones Multipunto.	118
11.2 Algoritmo de Esau-Williams	121
11.3 Algoritmo de Prim	125

CAPITULO XII

DISEÑO DE REDES GENERALIZADAS.	127
- Problemas intermedios dentro del diseño de una red	127
- Algoritmos de Solución	127
12.2 Algoritmo de la Adición	131

CAPITULO XIII

DISEÑO DE REDES DISTRIBUIDAS	134
13.1 Algoritmo de Saturación de corte	134
13.2 Algoritmo de Trayectorias cortas	138
13.3 Algoritmo Especifico	138

APENDICE	140
----------	-----

PROGRAMAS

A.1 Programa Emulador de Terminal VT100 para Equipos Burroughs B-25	140
A.2 Algoritmo de Flujo Máximo	171

BIBLIOGRAFIA	176
B.1 Artículos	177
B.2 Publicaciones	180
B.3 Libros	181

INDICE DE FIGURAS

Fig.	2.1	Componentes Básicos de un Sistema de Comunicación de Datos.	20.1
Fig.	2.2	Tipos de Señales	20.1
Fig.	2.3	Modos de Utilización de un Canal de Comunicación	20.2
Fig.	3.1	Formato de Mensaje en la Transmisión Asíncrona	23.1
Fig.	3.2	Formato de Mensaje en la Transmisión Síncrona (BSC)	23.1
Fig.	4.1	Configuración de un Enlace Multipunto	27.1
Fig.	4.2	Configuración de Enlace Punto a Punto	27.1
Fig.	5.1	Interfase RS-232 de 25 Patas	31.1
Fig.	6.1	Formato de un Mensaje en BSC	43.1
Fig.	6.2	Formato de un Mensaje en DDCMP	43.1
Fig.	6.3	Formato de un Mensaje en SDLC	43.1

Fig. 6.4	Flujo de la Transmisión en BSC	47.1
Fig. 6.5	Flujo de la Recepción en BSC	47.2
Fig. 7.1	Config. del Reg. de Chequeo de Bloque	63.1
Fig. 7.2	Reg. de chequeo de bloque inicial	63.1
Fig. 7.3	Reg. de Chequeo de bloque al tener como entrada un uno.	63.1
Fig. 7.4	Reglas de las Compuertas OR-Exclusivas	63.1
Fig. 8.1	Configuración Típica de una Red de Comunicación de Datos.	70.1
Fig. 8.2	Partes Esenciales de un Controlador de Comunicaciones.	72.1
Fig. 8.3	Partes Esenciales de un Modem FM	90.1
Fig. 8.4	Técnicas de Modulación	91.1
Fig. 8.5	Esquema de Multiplicación por División de Frecuencia.	91.1
Fig. 8.6	Arquitectura de un Modem por División de Frecuencia (FDM)	94.1
Fig. 8.7	Esquema TDM	94.1

Fig. 8.8	Arquitectura de un TDM	94.2
Fig. 10.1	Gráfica con Peso que contiene cuatro A-H cortes	112.1
Fig. 10.2	Flujo Factible de la Fig. 10.1	112.1
Fig. 10.3	Ejemplo del Algoritmo del Flujo Máximo	114.1
Fig. 10.4	Ejemplo de Conectividad por Flujo máximo	117.1
Fig. 11.1	Gráfica Ejemplo en Redes Centralizadas	124.1
Fig. 11.2	Pasos en el Algoritmo de Esau-Williams	124.1
Fig. 11.3	Resultado del el Algoritmo de Prim	124.1
Fig. 12.1	Ejemplo de Red Centralizada	131.1
Fig. 12.2	Inicialización en el Alg. de la Adición	131.1
Fig. 12.3	Resultado Final del Alg. de la Adición	131.1
Fig. 13.1	Algoritmo de Perturbación	136.1

PROLOGO

El proposito principal de esta tesis es introducir al estudiante del area de Computacion de las carreras de Matematicas, Física y Actuaría en el área de comunicación de datos, los componentes y el diseño de un sistema de comunicación de datos y, a la vez, servir como material de apoyo para un seminario a nivel licenciatura de esta materia. Dicha area relativamente nueva, ha alcanzado un auge bastante grande debido a las necesidades que surgen del volumen de información que debe ser manejada en cualquier tipo de organización, del crecimiento continuo de este volumen, de compartir y transportar información entre las distintas estaciones de trabajo que no se encuentran necesariamente en la misma localidad debido al crecimiento de las organizaciones y de la descentralización que surge de este crecimiento, así como de las nuevas tecnologías que han sido desarrolladas y que tienden a crear equipos cada vez mas complejos y compactos. Todo lo anterior promueve el tener distribuidos equipos inteligentes que puedan procesar la información de acuerdo a las necesidades de las distintas estaciones de trabajo, y a la vez compartir y explotar una fuente coman de información que se encuentre almacenada en el computador central de lo que se llama una red de comunicación de datos o bien red de teleproceso.

Dentro de los primeros seis capítulos se definen los elementos involucrados en esta área, así como los distintos procedimientos y conceptos que se manejan. El primer capítulo define el concepto de comunicación de datos, sus posibles utilidades y ventajas, así como algunos de los elementos que llevan a la necesidad de la comunicación de datos; los capítulos II y III describen aquellos conceptos básicos que necesitan conocerse para introducirse en el área. Los capítulos IV, V y VI describen los procedimientos, procesos y elementos involucrados en un sistema de comunicaciones de datos. Dentro del capítulo VII, se presentan los procedimientos básicos para la detección y manejo de errores dentro de este tipo de sistemas. En el capítulo VIII se describen y analizan distintos tipos de dispositivos utilizados en un sistema, desglosando sus partes y funciones que cada una de ellas realiza, así como los procedimientos internos que manejan. Por último los capítulos del IX al XII, que constituyen los capítulos más importantes, se centran en los elementos necesarios a considerar para un buen diseño de un sistema de comunicación de datos, así como de los algoritmos que se tienen para el diseño de los distintos tipos de redes y configuraciones.

CAPITULO I

INTRODUCCION

El término "Comunicación de Datos" se refiere al proceso de transmitir datos de un punto a otro a través de lo que se conoce como una facilidad de comunicación de datos. la cual corresponde a las líneas de transmisión independientemente de la tecnología utilizada (como puede ser cable, radio, satélite, microondas etc.) capaces de llevar señales de datos, opuestos a aquéllos que transmiten señales de voz.

Dos elementos importantes de las facilidades de comunicación son las facilidades mismas de la línea y el alcance del sistema de comunicación de datos. En un principio las únicas facilidades disponibles eran las líneas telefónicas, cables marinos y sistemas de transmisión por microondas. Todos éstos estaban diseñados básicamente para la comunicación de grado de voz. Hoy en día existen sistemas de microondas específicamente diseñados para la transmisión de datos, que se conoce como redes de valor agregado, cuya única característica adicional es la facilidad de efectuar tráfico de datos con exactitud y eficiencia. Se cuenta además con una amplia gama de facilidades de transmisión de datos que han sido desarrollados con el fin de incrementar la exactitud, disponibilidad y eficacia de la transmisión y a la vez reducir el costo de la comunicación de datos.

Otro punto importante que se mencionaba es el alcance de la facilidad de comunicación que se refiere a las restricciones que la limitan y que dependen de la naturaleza de la misma y muchas veces de los límites que imponen los proveedores comerciales de éstas.

De esta manera se puede definir una "Red de Comunicación de Datos" como un sistema compuesto del equipo necesario para efectuar la transmisión de datos entre distintos dispositivos de proceso a través de una facilidad de comunicación. Este equipo consiste a su vez de una serie de dispositivos que a través de los capítulos posteriores se irán mencionando.

Dentro de los principales usos de una red se pueden nombrar los siguientes:

- Descentralización de las operaciones de manejo de información.
- Entrada de datos de distintas fuentes que no necesariamente se encuentren dentro de una misma área.
- Transmisión remota de datos (Remote Job Entry) la cual consiste en transmitir trabajos y/o instrucciones de proceso a estaciones de procesamiento remoto.
- Acceso a programas, paquetes y bases de datos remotas.
- Desarrollo de paquetes comunes para distintos equipos compatibles.

Y muchos usos más que traen como consecuencia la versatilidad, disponibilidad y confiabilidad de la información, así como la reducción de los costos y tiempos necesarios para la transferencia de información entre distintas estaciones remotas.

Las mayores ventajas de contar con un sistema de comunicación incluyen:

- Una relación más favorable costo/rendimiento.
- Mayores posibilidad de crecimiento y explotación del equipo e información
- Desarrollo de nuevas aplicaciones que repercuten en la manera de efectuar el manejo de información.

En general se puede decir que dos grandes motivos han estado presentes para que los sistemas de computadoras sean centralizados en redes. El primero de éstos es el que existe un gran número de organizaciones que cuentan con una cantidad substancial de computadoras en operación, cada una localizada y explotada independientemente de la otra. Por ejemplo, una compañía con varias sucursales a las cuales se les ha asignado un equipo para realizar procesos por separado que en determinado momento deben compartir sus resultados con las otras localidades con el fin de obtener una integración que corresponda a la información total de la compañía. Esto implica poner en disponibilidad para su uso los programas, datos y otros recursos para todas las estaciones, de manera que aún estando a grandes distancias,

el sistema en su totalidad pueda ser visto como un sistema local. esto es, un sistema de recursos compartidos y de procesos distribuidos.

Un aspecto importante que se observa al armar una red es la facilidad de poder contar con recursos alternativos en caso de fallas del equipo o líneas de comunicación que forman el sistema, pudiendose contar así con equipo o facilidades de respaldo que puedan substituir al que falla, de tal manera de mantener ininterrumpidos los procesos y manejos de información que se efectúan en esta red.

Otro punto también importante es el costo relativo de las computadoras y de los elementos de comunicación. Hasta antes de la década de los setentas el costo de las computadoras era mayor comparado con el de las facilidades de comunicación. Actualmente el costo de las pequeñas computadoras o microprocesadores es relativamente bajo lo que permite que la información sea analizada en el lugar donde fue generada y sólo sea compartida la información más importante. reduciendo así el costo de la explotación y manejo de la información. Esto último nos da otra razón para la utilización de redes: el que esté presente una relación superior costo/rendimiento brindada por las pequeñas computadoras sobre los grandes equipos.

CAPITULO II

CONCEPTOS DE UN SISTEMA DE COMUNICACION DE DATOS

Un sistema de comunicación de datos consta de tres componentes básicos:

- La fuente cuya función es originar la información.
- El medio que es la trayectoria a través de la cual fluye la información y que corresponde a la facilidad de comunicación.
- El receptor o destino que es el elemento que acepta la información.

Estos tres componentes se ilustran en la figura 2.1 en la cual la fuente corresponde a un computador central y el receptor a una terminal.

Varios terminos son utilizados para las trayectorias entre los dispositivos de proceso de datos en un sistema de comunicación, dentro de los que se incluyen: canal de comunicación, líneas de comunicación, facilidad de comunicación y liga de comunicación.

Estas facilidades están diseñadas en su gran mayoría para la transmisión orientada a grado de voz y se clasifican en dos grupos:

- Líneas conmutadas (conocidas también como líneas switcheadas, servicio de intercambio o líneas "dial-up").
- Líneas privadas (o no switcheadas, dedicadas o atadas).

Cada uno de estos dos grupos es a su vez dividido en subgrupos en términos de la velocidad de transmisión o amplitud de banda, disponibilidad de servicio y arrendamientos especiales ofrecidos por los distribuidores. En términos de velocidades de transmisión se tiene 3 categorías:

- Líneas de baja velocidad o líneas de grado de subvoz que son aquellos cuya amplitud de banda es la necesaria para la transmisión de voz y que pueden transmitir datos a velocidades menores o iguales a 300 bps (bits por segundo)
- Líneas de velocidad media o de grado de voz las cuales soportan velocidades que oscilan entre los 1200 y 9600 bps. esta última categoría es la más utilizada en comunicaciones tanto interactivas como tipo batch.
- Líneas de alta velocidad que soportan velocidades superiores a los 9600 bps.

Hasta la década de los sesentas el tráfico de datos se efectuaba sobre los canales básicos que consistían en las redes públicas de teléfonos. Hoy en día se cuenta con distintos tipos de medios de transmisión usados para comunicación de datos dentro de los que se encuentran:

- Hilos abiertos (generalmente telefónicos): que consisten de un par de cables diseñados para transmisión dentro de ciertas amplitudes de banda. Este medio presenta problemas

como pueden ser el cruce de línea, interferencia electromagnética, pérdida por atenuación y otros problemas de esta naturaleza.

- Cable: que está formado por dos o más hilos abiertos.
- Cable coaxial: que está compuesto de un cilindro conductor que contiene un hilo, separados por un medio aislante. Este tipo de cables permiten transmitir a mayores frecuencias que un hilo abierto, contándose así con un rango mayor de datos transmitidos en un período de tiempo.
- Radio de alta frecuencia: a diferencia de los anteriores no transmite sobre un medio metálico sino que las señales son enviadas sobre el espacio abierto en una porción específica de espectro de frecuencia, permitiéndose así un mayor número de señales por unidad de tiempo.
- Sistemas de radio de microondas que utilizan el rango de frecuencia de radio más alto por lo cual se requiere de un equipo más sofisticado para la transmisión y recepción.
- Satélites de comunicación que son una forma de transmisión de microondas a través de satélite y se utiliza para transmisiones a larga distancia.
- Sistemas transportadores que no corresponden a un dispositivo físico sino a una técnica la cual utiliza varios medios de transmisión por medio de técnicas de multiplicación (que en capítulos posteriores se describirán). Estos sistemas acomodan un número de canales sobre una línea física colocando cada canal en un nivel distinto de frecuencia.

Por otra parte, debido a que la mayoría de los canales de comunicación están diseñados para transmitir señales de voz (señales analógicas o continuas) que pueden tomar un número infinito de valores de amplitud, y como las

computadoras trabajan con señales digitales (o discretas) que pueden tomar únicamente un número finito de valores de amplitud (generalmente dos)(Fig. 2.2), esto trae como consecuencia que las señales digitales deban ser convertidas a señales analógicas (proceso llamado modulación) antes de que sean transmitidas por el canal de comunicación, y a la vez deban ser convertidas estas señales analógicas en señales digitales (proceso llamado demodulación) al llegar al receptor. Esto obliga a contar con un dispositivo que efectúe estas funciones conocido como convertidor de señal, data set, modem (MÓDULADOR-DEMÓDULADOR) o adaptador de línea. Estos además de efectuar su función básica pueden contar con otras características como son:

- Respuesta automática a llamadas en canales conmutados.
- Selección de velocidades de transmisión menores durante períodos de interferencia.
- Uso de un canal conmutado en caso de que el canal dedicado no funcione adecuadamente, etc.

El término canal de comunicación se refiere a la trayectoria entre los convertidores de señal u otros dispositivos que efectúen las funciones de este. Este canal incluye todo el equipo de transmisión de datos. El término línea de comunicación es utilizado para referirse a la trayectoria de la computadora hacia el equipo terminal, incluyendo convertidores de señales y la porción de control de comunicación de la terminal o computadora receptora.

Los modos de utilización de un canal o métodos de transmisión disponibles son (fig. 2.3):

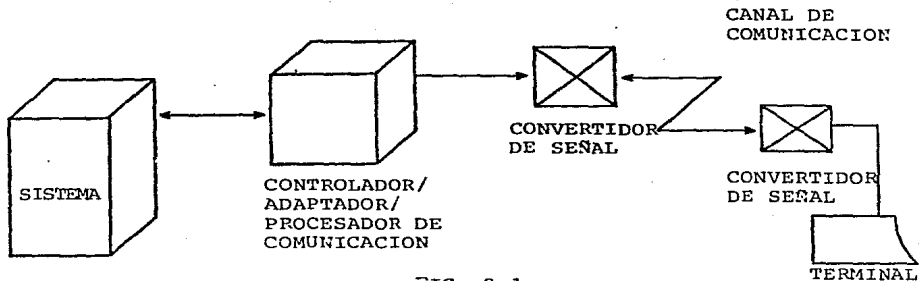
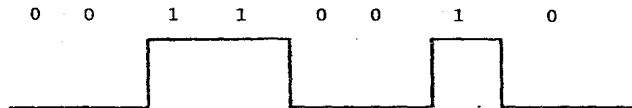


FIG. 2.1
COMPONENTES BASICOS DE UN SISTEMA
DE COMUNICACION DE DATOS



SEÑAL DIGITAL



SEÑAL ANALOGICA

FIG. 2.2
TIPOS DE SEÑALES

- Simplex en el cual la información fluye en una sola dirección de canal.
- Semiduplex (half-duplex) donde el flujo puede ser en ambas direcciones pero en una sola a la vez.
- Duplex (full-duplex) en el que se tiene un flujo en ambas direcciones pudiendo ser simultáneo en ambas.

Otros términos utilizados dentro de la comunicación son:

- "Baud Rate" (Tasa de Bauds) que es el número de cambios ocurridos por incremento de unidad de tiempo entre los convertidores de señal.
- "Bit Rate" (Tasa de Bits) que se refiere al número de bits por incremento de unidad de tiempo en la entrada y salida de los convertidores de señales.
- "Data Rate" (Tasa de Datos) que corresponde a la velocidad en que opera la terminal o el computador central en una unidad de tiempo .

Y una serie de términos adicionales que se irán describiendo dentro de los capítulos subsecuentes, conforme vayan apareciendo.

CAPITULO III

MODOS DE TRANSMISION

Los modos de transmisión son, como su nombre lo indica, métodos que se refieren básicamente a la forma en que se transmiten los datos. Los bits de datos binarios son comúnmente transferidos entre dispositivos electrónicos por cambios de corriente o voltaje. Esta transferencia puede realizarse en SERIE sobre una simple, o en PARALELO sobre varias líneas a la vez. Puede ser síncrona, en la cual el tiempo exacto de salida o llegada de cada bit de información es predecible, o puede ser asíncrona en cuyo caso los datos son transferidos de una manera no uniforme.

En la transmisión en paralelo, cada bit de un conjunto que representa un carácter tiene su propio cable. Adicionalmente se tiene otro cable llamado el "strobe" o reloj, el cual notifica al receptor cuando todos los bits están presentes en su cable respectivo, de modo que los voltajes en los cables pueden ser simplificados.

En la transmisión en serie, los bits que representan el carácter son mandados por un cable simple, uno después de otro.

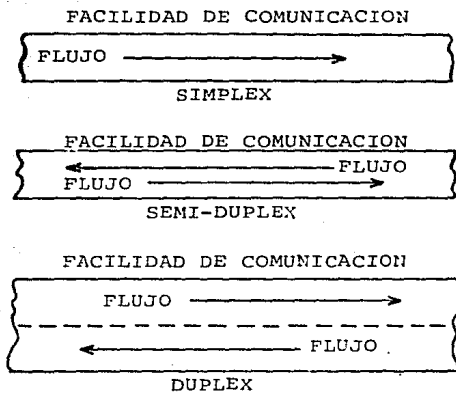


FIG. 2.3
MODOS DE UTILIZACION DE UN
CANAL DE COMUNICACION

En las computadoras y otros dispositivos digitales de alta velocidad, los datos son transmitidos en paralelo debido a la proximidad de los receptores con respecto a los transmisores.

Sin embargo, cuando la distancia entre los dispositivos se incrementa, la cantidad de cables necesarios para la transmisión en paralelo no sólo resulta más costosa, sino que complica la estructura de los manejadores de línea y dispositivos que utilizará el receptor para interpretar las señales. Por ello la transmisión en serie es generalmente utilizada donde el costo del medio de comunicación es suficientemente alto para justificar la relativa complejidad del sistema de transmisión y recepción. En estos casos el sistema de transmisión recibe un carácter en paralelo, lo transmite en serie sobre la línea y al recibirlo lo reensambla en paralelo.

Dentro de la transmisión en serie existen tres submodos de transmisión:

- Asíncrona, síncrona e isócrona.

En la transmisión asíncrona, también conocida como transmisión de arranque/parada, es enviado un carácter a la vez y en cualquier momento. Cada carácter transmitido tiene un bit llamado de arranque que lo precede y uno o dos bits llamados de parada que lo suceden. A partir de estos tipos de bits cada carácter logra la sincronía con el receptor por medio de sus propios bits de arranque y parada (Fig. 3.1). El bit de arranque es una señal que se utiliza para informar al dispositivo receptor que comience a analizar la señal de datos de entrada a velocidad fija para que el carácter pueda

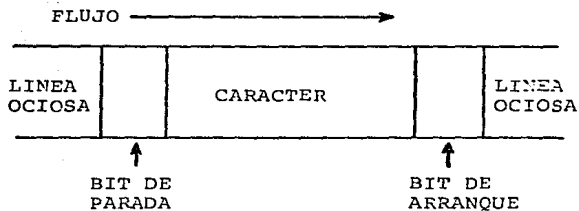
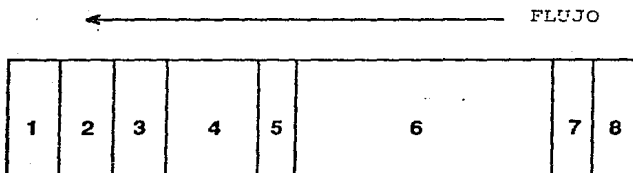


FIG. 3.1
 FORMATO DE MENSAJE EN LA
 TRANSMISION ASINCRONA



- 1, 2.- CARACTERES DE SINCRONIA (SYNC)
- 3.- PRINCIPIO DE ENCABEZADO (SOH)
- 4.- ENCABEZADO
- 5.- CARACTER DE PRINCIPIO DE TEXTO (STX)
- 6.- TEXTO
- 7.- CARACTER DE FIN DE TEXTO O BLOQUE (ETX/ETB)
- 8.- CARACTER DE CONTROL DE BLOQUE (BCC)

FIG. 3.2
 FORMATO DE MENSAJE EN LA TRANSMISION
 SINCRONA
 (BSC)

interpretarse en su estructura de caracter adecuado. El bit de parada, que sigue a los bits de datos, informa al dispositivo receptor el fin de un caracter y restaura al dispositivo para que reconozca el siguiente bit de arranque.

De esta forma la sincronización de dispositivos se restablece al recibir cada caracter.

Este modo de transmisión de arranque/parada, es utilizado generalmente en los mecanismos de teclado con los que el operador envía caracteres a lo largo de la línea, a intervalos más o menos aleatorios cuando oprime las teclas. Así puede haber un período indeterminado de tiempo entre un caracter y el siguiente.

Al tenerse una transmisión continua entre los dispositivos con tiempos regulares, la transmisión síncrona brinda una utilización más eficiente de la línea por medio de transmisiones a altas velocidades de bloques de caracteres. En este caso, tanto el dispositivo emisor como el receptor operan simultáneamente y se resincronizan después de la transmisión de un bloque de caracteres. Esta sincronización se establece definiendo un caracter de sincronización, comúnmente llamado "sync", el cual es elegido de tal manera que sea significativamente diferente a cualquier otro caracter que sea transmitido regularmente. Este caracter de sincronización es mandado un número de veces predeterminado entre los dispositivos transmisor y receptor (Fig. 3.2). El tamaño de bloques de transmisión de datos a menudo se relaciona con la naturaleza física del medio transmisor y receptor. En muchos dispositivos la sincronización se controla por medio de un oscilador, la

máquina receptora tiene que quedar exactamente en fase con el oscilador de la máquina transmisora, lo cual se logra enviando un patrón de caracteres de sincronización al principiar el bloque. Otros sistemas fijan un límite superior a la longitud de los bloques, aunque esto no ocurre siempre, porque pueden enviarse caracteres de sincronización a mitad del bloque. Por ejemplo, los equipos BSC (Binary Synchronous) de IBM, insertan dos caracteres de sincronización en el texto a intervalos de un segundo.

Debido a que en la transmisión síncrona no necesitan ser enviados los bits de parada y arranque para la sincronización de la comunicación, sino que todos los bits de cada carácter son usados para la transmisión de datos, se tiene como consecuencia poder eliminar aproximadamente el veinte por ciento de la utilización de la línea que se perdía en la comunicación asíncrona. Sin embargo, el patrón de caracteres proporcionado por los bits de arranque y parada se pierde, lo que lleva a la necesidad de contar con otro método para determinar qué grupos de bits constituyen un carácter. Típicamente, las unidades de recepción síncrona son puestas en un modo de búsqueda de sincronización, ya sea por Software o por Hardware, al iniciar una transmisión o bien cuando un fracaso en un dato ha ocurrido y el Hardware o Software determinan la necesidad de resincronización. En el procedimiento de búsqueda de sincronización, el hardware cambia un bit en el registro de cambio del receptor (el bit que llega), con él compara el contenido de este registro con el del registro en el cual se almacena el registro de sincronización, y si no se obtiene resultado positivo repite el proceso. Si el resultado de la comparación es positivo el receptor empieza un cambio de

bits y enciende una banda de 'caracter disponible' cada ocho bits. Algunos sistemas de comunicaciones requieren que el receptor identifique dos caracteres de continuación sucesivos antes de encender la bandera de 'caracter disponible', esto con el fin de tener una mayor certeza de que la sincronización se ha obtenido adecuadamente.

Por último, el tercer submodo de transmisión en serie es conocido como transmisión isócrona, la cual combina las características de la transmisión síncrona y asíncrona. Al igual que la transmisión asíncrona, se requiere que cada caracter tenga un bit de arranque y otro de parada, y a la vez, al igual que en la transmisión síncrona, el receptor y el transmisor están sincronizados. En este tipo de transmisión todos los períodos de no transmisión consisten de uno o más intervalos de un caracter, manteniendose así una sincronización. Esta sincronización permite tener mayor precisión entre el dispositivo transmisor y el receptor en comparación con la transmisión asíncrona además de poderse obtener mayores velocidades.

En resumen se tienen dos modos de transmisión:

- En Serie
- En Paralelo

Y dentro de la transmisión en serie se tienen tres submodos: transmisión asíncrona, síncrona e isócrona,

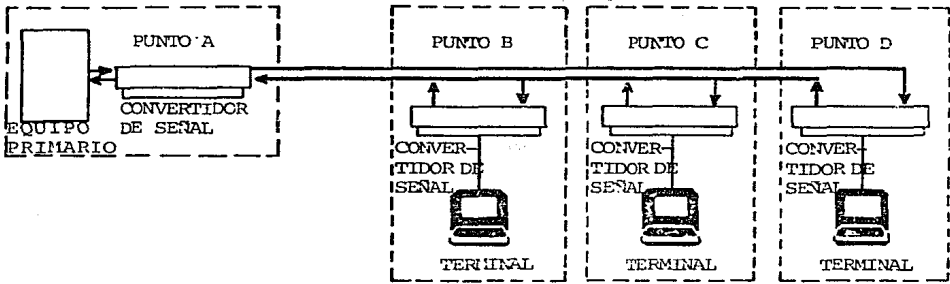
CAPITULO IV

TIPOS DE ENLACES

El contacto entre el computador central y un sistema remoto puede ocurrir de dos maneras dependiendo del tipo de línea de conexión entre ellos. Estas dos maneras corresponden a conexión en multipunto (en líneas no conmutadas) y conexión punto a punto (en líneas no conmutadas y conmutadas).

Una línea punto a punto conecta una terminal o computadora directamente a otra terminal o computadora (fig. 4.1). Esto trae como consecuencia que sean los canales en que es más fácil controlar la comunicación. Este control es generalmente efectuado por medio de la siguiente secuencia:

- El computador central chequea primero que la terminal se encuentre lista para recibir datos. Si es así, ésta responde con un reconocimiento indicando su estado al computador central, pudiendo ser esta respuesta positiva o negativa. Así, hasta no recibir la respuesta, el computador central normalmente indica un error si la respuesta fue negativa, o bien si la terminal no responde. En caso de que la respuesta sea positiva el computador central manda el dato.



LINEA DE ENLACE MULTIPUNTO

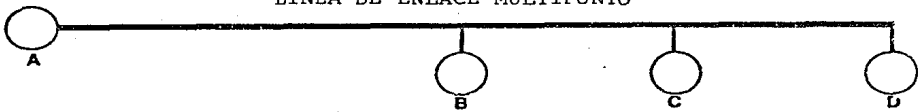


FIG. 4.1
CONFIGURACION DE UN ENLACE
MULTIPUNTO

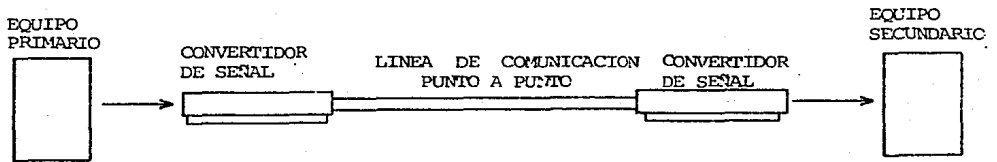


FIG. 4.2
CONFIGURACION DE UN ENLACE
PUNTO A PUNTO

- Si la terminal recibe el dato de manera correcta, normalmente manda una respuesta positiva de regreso al computador, el cual a su vez tiene la opción de seguir mandando datos o indicar el fin de la transmisión.

En un sistema conectado por medio de una línea multipunto (fig 4.2) los datos son enviados y recibidos bajo el control del computador central. Para poder mandar o recibir datos la estación remota, conocida como estación tributaria, debe ser seleccionada por el computador central el cual puede ganar acceso a una de estas estaciones por medio de una invitación ("polling") o bien seleccionando la dirección específica de la estación respectiva ("addressing"). Así, en un canal multipunto se tiene una estación de control que efectúa el proceso de selección de las estaciones conocidas como tributarias.

Cuando la estación de control manda datos a una estación tributaria en un canal multipunto, los datos se encuentran disponibles a todas las estaciones conectadas en el canal, por lo que cada transmisión debe ser controlada para que la estación correcta reciba el dato. De esta manera a cada estación le es asignada una dirección y la estación de control selecciona el destino correcto mandando la dirección de la estación antecediendo al dato. Este proceso es llamado direccionamiento o selección.

Una vez que la estación es seleccionada y está lista para recibir datos, la estación de control envía éstos por el canal, recibiendo los únicamente la estación seleccionada, independientemente de que se encuentren disponibles para todas las estaciones.

Este proceso de direccionamiento es efectuado por medio de la siguiente secuencia:

- Una vez seleccionada la estación tributaria, ésta debe responder al direccionamiento, que en caso de ser una respuesta positiva indica que la terminal está lista para recibir datos.
- Una vez recibidos los datos, responde de nuevo al computador central el cuál tiene la opción de continuar la transmisión o indicar el fin de ésta.

En la otra manera de control que se mencionaba y que corresponde a la invitación, la estación de control efectúa una invitación de comunicación sobre las estaciones tributarias para checar sus requerimientos de transmisión de información. Esta invitación no es necesario que se lleve a cabo en secuencias sobre las estaciones tributarias, sino que puede ser definida una lista en la que las estaciones poco utilizadas son checadas con menor frecuencia, provocando así mayor eficiencia de operación y utilización de la línea.

Este proceso de invitación puede ser visto de la siguiente manera:

- El computador central checa cada terminal en la línea empezando por la primera ya sea en la línea o bien la primera definida en la lista, enviando la dirección correspondiente.
- Si la estación respectiva no tiene requerimientos de transmisión de datos responde negativamente a la invitación.

y el proceso continúa en secuencia sobre las estaciones o sobre la lista.

- En caso contrario se tendrá una respuesta positiva pudiéndose efectuar la transmisión, siendo este proceso continuo sobre todas las estaciones.

Otra situación que debe ser controlada es la falla a una respuesta sobre un requerimiento o invitación, la cual puede ser provocada por situaciones como son el que la estación se encuentre apagada o que la línea no se encuentre disponible. En este caso se esperará un tiempo relativamente corto para enviar nuevamente la invitación sobre esta terminal o bien para continuar con la siguiente estación después de un determinado número de intentos. Esta espera es conocida como "tiempos fuera" los cuales varían desde un segundo hasta varios minutos dependiendo de si la estación se encuentra recibiendo o transmitiendo. Estos tiempos fuera pueden ser controlados automáticamente por hardware o bien por funciones de programa.

CAPITULO V

INTERFASES

Una interfase es un estandar para la interconexión de equipo obtenidos de distintos proveedores y que contienen características diferentes. Estos se conectan a través de una interfase adecuada aprovechando las características comunes de los distintos equipos.

La necesidad de homogenizar la interconexión de equipos surge del hecho de que en los primeros tiempos de la comunicación de datos, dos o tres compañías proveedoras de facilidades de comunicación establecían los estandares de las características eléctricas de los modems comerciales que se fabricaban y con ello provocaban que las compañías manufactureras de equipo terminal tuvieran problemas para la elaboración de sus equipos.

Para resolver este problema, la asociación de industrias electrónicas de los Estados Unidos en cooperación con compañías manufactureras de equipo terminal y de modems, desarrollaron un estandar para la interfase entre el equipo terminal y el equipo de comunicaciones empleado en el intercambio de información en serie, estableciendo el estandar físico y lógico. Esta interfase es conocida como "interfase RS-232" (fig. 5.1), e incluye:

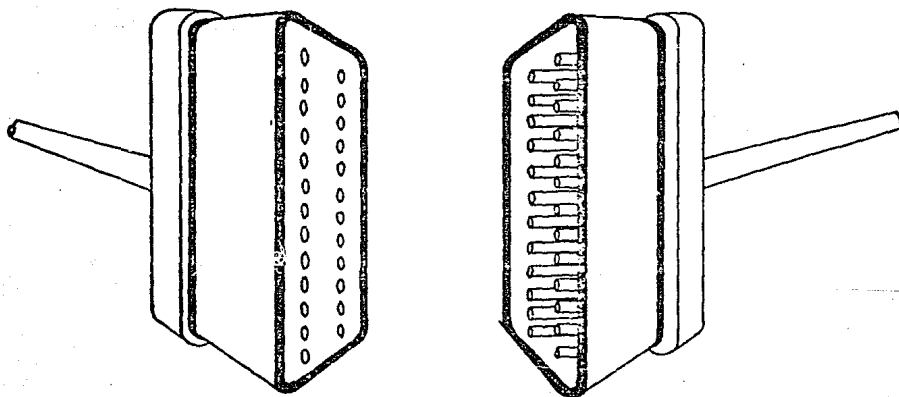


FIG. 5.1

INTERFASE RS-232 DE 25 PATAS

- 1.- Las características de la señal eléctrica.
- 2.- Las características mecánicas de la interfase.
- 3.- Descripción funcional de las características mecánicas de la interfase.
- 4.- Descripción funcional de los circuitos de intercambio.

La tabla 5.1 muestra los circuitos de esta la interfase.

TABLA 5.1.

Circuito de Intercambio.	A	B	C	D	E	F	G	H	I	J	K	L	M	N
AA Tierra de protección	X	X	X	X	X	X	X	X	X	X	X	X	X	X
AB Tierra de señal	X	X		X	X	X	X	X	X	X	X	X	X	0
BA Transmisión de datos	X	X		X	X	X	X	X	X	X	X	X	X	0
BB Recepción de datos			X	X	X	X		X		X		X	X	0
CA Petición para transmitir.	X		X		X				X		X			0
CB Preparado para transmitir.	X	X		X	X	X		X		X		X	X	0
CC Conjunto de datos listo.	X	X	X	X	X	X	X	X	X	X	X	X	X	0
CD Terminal de datos lista.														0
CK Indicador de llamada	S	S	S	S	S	S	S	S	S	S	S	S	S	0
CF Detector de señal de línea recibida.			X	X	X		X	X	X					0
CG Detector de calidad de señal.														0
CH/CI Selector de velocidad de señalización de datos. (DTE)/(DCE)														0
DA/DS Tiempo del elemento transmisor de señal.	T	T		T	T	T		T		T	T	T	T	0
DD Tiempo del elemento receptor de señal.				T	T	T		T		T	T	T	T	0
SBA Datos transmitidos secundarios.								X		X	X	X	X	0
SBB Datos recibidos secundarios.							X	X		X	X	X	X	0
SCA Requerimiento de envío secundario.											X	X		0
SCB Borrado de envío secundario.								X		X	X	X	X	0
SCF Detector secundario de señal de línea recibida.							X	X		X	X	X	X	0

donde en las celdas se indica:

- (O) Es especificado por el proveedor.
- () Opcional.
- (S) Circuitos de intercambio adicionales requeridos para servicio switchado.
- (T) Circuitos de intercambio adicionales requeridos para canales síncronos.
- (X) Circuitos de intercambio básico (todos los sistemas)

Las columnas indican:

- A.- Sólo para transmisión.
- B.- Sólo para transmisión.(*)
- C.- Sólo para recepción.
- D.- Duplex (*)-Semi-Duplex
- E.- Duplex
- F.- Canal primario sólo transmisión(*)/Canal secundario sólo recepción.
- G.- Canal primario sólo recepción/Canal secundario sólo transmisión.
- H.- Canal primario sólo transmisión/Canal secundario sólo recepción.
- I.- Canal primario sólo recepción/Canal secundario sólo transmisión.
- J.- Canal primario sólo transmisión (*)/Canal secundario semi-duplex.
- K.- Canal primario sólo transmisión/Canal secundario semi-duplex.
- L.- Canal primario duplex/Canal secundario duplex (*) Canal primario semi-duplex/Canal secundario semi-duplex.
- M.- Canal primario duplex/Canal secundario duplex.
- Z.- Especial (Circuitos especificados por el proveedor).

(*) Indica que debe incluirse CA (Petición para transmisión).

Definiendo el conjunto de señales que permiten a dos o mas dispositivos poder establecer la comunicación y la sincronización necesaria. De Este conjunto definido de señales un subconjunto corresponde a las señales enviadas por el dispositivo que en determinado momento de la comunicación funge como transmisor y que a su vez son requeridas por el receptor en otro subconjunto de estas señales. Por ejemplo, la señal de transmisión es enviada por el equipo transmisor hacia el equipo receptor que la recibe en la parte correspondiente de la interfase a esta señal. De igual manera la señal de "Preparado para transmitir" que es enviada por el transmisor para indicar al equipo receptor que se encuentra listo para transmitir una respuesta, o bien tiene un requerimiento de transmisión, es recibida por el receptor en la parte correspondiente de la interfase a la señal de "Petición para Transmitir".

Por ejemplo en el caso de uno de los modems comerciales distribuidos en México como lo es el modem SISCO M 24, diseñado para transmisión sincrónica en líneas conmutadas y privadas a velocidades de 1200 ó 2400 BPS's (cualidades que pueden ser configuradas a través de cuatro microswitches), cada señal llevada a través de la interfase RS-232 por medio de la cual se conecta este modem al equipo terminal tienen la siguiente función en cada "Pin" o pata (un "Pin" es una pata del conector):

<u>No. Fin</u>	<u>Dirección</u>	<u>Función.</u>
1	-----	Tierra de protección.
2	Entrada	Transmisión de datos.
3	Salida	Recepción de datos.
4	Entrada	Petición para transmitir
5	Salida	Preparado para transmitir
6	Salida	Aparato para datos preparado.
7	-----	Tierra de señal.
8	Salida	Detector de señal.
9	-----	- V (*)
10	-----	- V (*)
11	Entrada	Petición para transmitir (Canal de retorno)
12	Salida	Detector de señal (Canal de retorno)
13	Salida	Canal de retorno preparado
14	Entrada	Transmisión de datos (Canal de retorno)
15	Salida	Reloj interno para señal de transmisión.
16	Salida	Recepción de datos (Canal de retorno)
17	Salida	Reloj interno para señal de recepción.
20	Entrada	Terminal de datos preparada.
22	Salida	Indicador de llamada.
23	Entrada	Selector de velocidad binaria.
24	Entrada	Reloj externo para señal de transmisión.

(*) Indica voltaje que generalmente oscila entre -7 y -12 volts dependiendo del equipo involucrado.

Cabe hacer notar que cada una de estas funciones tiene asignada una dirección, que indica si el flujo va o viene del equipo, por lo cual dependiendo del tipo de conexión y de los requerimientos del equipo es necesario invertir en las puntas de la conexión algunas señales que son inversas una de la otra, como son:

- Transmisión y recepción.
- Petición para transmitir y preparado para transmitir.
- Aparato de datos preparado y terminal de datos preparado.

Otro estándar de interfase para transmisión en paralelo es la interfase conocida como Centronics, con la cuál se obtienen las ventajas de este modo de transmisión dentro de las que se incluyen:

- Eliminación de la necesidad de codificadores y decodificadores.
- No se requiere de un reloj para regir la velocidad
- Se cuenta con una sincronización implícita.

Esta interface tiene la siguiente configuración:

No. PinFunción1
No. PinStrobe de datos
Función

3	Bit de datos 1
3	Bit de datos 2
4	Bit de datos 3
5	Bit de datos 4
6	Bit de datos 5
7	Bit de datos 6
8	Bit de datos 7
9	Bit de datos 8 (si se utiliza)
10	Reconocimiento (acknowledge)
11	Ocupado (busy)
12	Transmisión fuera de línea
13	5 Volts
14	Tierra
15	Libre
16	Tierra
17	Tierra de chasis
18	5 volt
19	Tierra de unión
20	Tierra de unión
21	Tierra de unión
22	Tierra de unión
23	Tierra de unión
24	Tierra de unión
25	Tierra de unión
26	Tierra de unión
27	Tierra de unión
28	Tierra de unión
29	Tierra de unión
30	Tierra de unión
31	5 Volts
32	5 Volts
33	Tierra
34	Libre
35	Libre
36	Libre

CAPITULO VI

PROTUCULUS

Un protocolo se define como un conjunto de reglas utilizadas para la operación de un sistema de comunicación. Estas reglas están enfocadas básicamente para resolver los siguientes puntos:

- El enmarcamiento de las componentes de la transmisión como son:

. La determinación de que grupos de ocho bits constituye un carácter.

. Cuales grupos de caracteres forman el mensaje.

- La detección de errores por alguno de los métodos que en el capítulo posterior se describen, el control de errores recibiendo los mensajes correctos y haciendo el requerimiento para la retransmisión de los mensajes erróneos, así como la numeración de los mensajes para eliminar mensajes duplicados o evitar pérdida de estos.

- Manejo de transmisiones en modo transparente (como son transmisiones de código binario, esto es transmisiones codificadas), dentro de las cuáles pueden formar parte del mensaje de datos los patrones de los caracteres de control utilizados para realizar los dos puntos anteriores, evitando que estos patrones sean identificados dentro del mensaje como los caracteres de control respectivos.

- Efectuar el control de línea, que consiste en determinar cual estación transmite y cual estación recibe en el caso de transmisión semi-duplex o líneas multipunto.
- Control de tiempos fuera determinando qué hacer en el caso de que el flujo de mensajes cese totalmente.
- Inicialización de la transmisión en una línea de comunicación o sistema de transmisión que ha estado ocioso, esto es, no efectuando flujo de datos.

En sí los protocolos pueden ser divididos en tres grandes categorías de acuerdo a las técnicas utilizadas para el enmarcamiento de los mensajes:

- Protocolos orientados a carácter, los cuales utilizan un conjunto de caracteres de control que efectúan cada uno una función de enmarcamiento o delimitación de las partes que forman el mensaje, por ejemplo el carácter STX (start of text) para indicar el principio de un mensaje, y otro carácter, como es ETB (end of text block), el cual antecede al carácter de chequeo de bloque) para indicar el fin de un mensaje o bloque de texto. Un ejemplo de este tipo de protocolo es BSC (binary synchronous) de IBM.

- Protocolos orientados a un contador de byte, los cuales incluyen un carácter especial de principio de bloque seguido por un contador que indica cuantos caracteres siguen en la porción de datos del mensaje y alguna información de control tal como aquellos mensajes que han sido recibidos correctamente. Un ejemplo de este tipo es DDCMP (Digital Data Communication Message protocol) de Digital Equipment Corporation.

- Protocolos orientados a bit en donde los bits del mensaje son delimitados por un caracter especial de bandera, por ejemplo 01111110, el cual especifica que nunca se presentarán seis unos consecutivos en la transmisión de la bandera. Un ejemplo de este tipo es SDLC (Synchronous Data Link Control) de IBM.

A continuación se describen en detalle los tres ejemplos que se indican en cada uno de los tipos de protocolos, BSC, BDCMP y SDLC.

BSC**BYNARY SYNCHRONOUS**

BSC fué diseñado en 1968 principalmente para la transmisión entre computadoras IBM, terminales batch y videos de despliegue, pasando a ser uno de los protocolos mas utilizados hoy en día en sistemas de transmisión de datos. Como antes se mencionaba es un protocolo orientado a caracter que utiliza una serie de caracteres especiales para determinar el enmarcamiento de un mensaje y efectuar así las funciones de un protocolo.

A grandes rasgos la transmisión a través de BSC tiene las siguientes características:

El mensaje consiste de uno o más bloques de texto de datos, principalmente para proporcionar mayores facilidades y eficiencia en el control de errores. El texto de datos es el cuerpo del mensaje y es identificado por el caracter STX (Start of text), seguido de un bloque de texto. Adicionalmente cada bloque de texto, excepto el último, es seguido por un caracter ETB (end of transmission block) o un caracter de bloque intermedio ITB (intermediate block), y el último bloque es seguido por el caracter ETX (end of text). Opcionalmente el texto del mensaje puede ser precedido por un encabezado que contiene información auxiliar como puede

ser control de estaciones, prioridades, etc., pertenecientes al bloque inmediato. Este encabezado es identificado por un caracter SOH (start of heading) que indica el principio de este.

Al terminar la transmisión de cada bloque de mensajes y antes de continuar la transmisión, el receptor chequea la precisión de la transmisión. La figura 6.1 muestra el formato de un mensaje en BSC (siguiendo el flujo de datos de izquierda a derecha y donde CCB es el caracter de control de bloque para chequeo de errores, que se expone en el capítulo VII).

En el caso de transmisión en modo transparente, el texto es delimitado con DLE ETX y DLE ETB (o DLE ETB), donde DLE (Data Link Escape) es un caracter de escape que indica que el siguiente caracter es de control.

El conjunto de caracteres de control utilizados por BSC para establecer el control de la liga de datos, está formado por los siguientes caracteres especiales:

- SYN (Synchronous Idle)

Este caracter es utilizado para establecer y mantener la sincronización y como caracter de relleno en ausencia de datos u otro caracter a ser transmitido. Generalmente son utilizados como patrón de sincronización dos caracteres SYN contiguos al principio de cada transmisión.

- SOH (Start of Heading).

Este caracter se utiliza para anteceder a un encabezado.

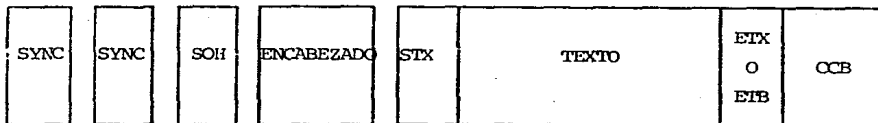


FIG. 6.1
FORMATO DE UN MENSAJE EN BSC

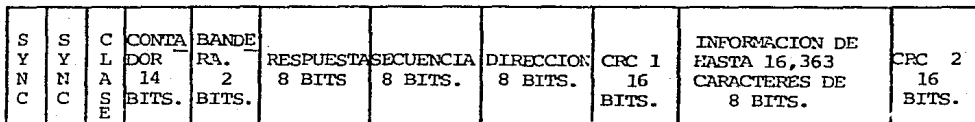


FIG. 6.2
FORMATO DE UN MENSAJE EN DDCMP

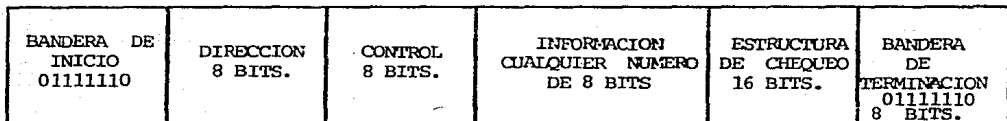


FIG. 6.3
FORMATO DE UN MENSAJE EN SDLC

- STX (Start of Text).

Es el caracter que antecede a un bloque de texto.

- ETB (End of Transmission Block)

Indica el final de un bloque de caracteres, iniciado con un caracter SOH o STX e inmediatamente después de este caracter le siguen el caracter de chequeo de bloque. Adicionalmente, al enviarse un ETB, el transmisor requiere respuesta del receptor indicando su estado por medio de uno de los siguientes caracteres: ACK0, ACK1, NAK, WACK o RVI.

- ITB (End of Intermediate Transmission Block).

Este caracter es utilizado con propósitos de detección de errores para dividir un mensaje. El caracter de chequeo de bloque sigue al caracter ITB. Después del primer bloque intermedio, los bloques intermedios sucesivos no necesitan ser precedidos por un caracter STX o SOH, excepto en modo transparente donde lo precede DLE STX, o bien cuando un bloque intermedio es únicamente encabezado y el siguiente es texto. El bloque debe comenzar con STX.

- ETX (End of Text).

Termina un bloque de caracteres iniciados con STX o SOH. El caracter de chequeo de bloque sigue al ETX. Al enviarse el ETX se espera respuesta del estado del receptor.

- EOT (End of Transmission).

Indica el final de la transmisión de un mensaje que puede contener uno o más bloques, incluyendo al texto y los encabezados respectivos. Este caracter causa una restauración de las estaciones en la línea, por lo cual es utilizado también como respuesta a un requerimiento de

transmisión cuando no se tiene nada que transmitir o bien para indicar un mal funcionamiento del sistema, siendo una señal de aborto.

- ENQ (Enquiry).

Es utilizado en una transmisión repetida de la respuesta a un bloque de mensaje si la respuesta fue perdida o no recibida cuando se esperaba. ENQ es también utilizado para una declaración de la línea en conexiones punto a punto o para indicar el final de un reconocimiento o secuencia de selección (que posteriormente definiremos).

- ACK0/ACK1 (Affirmative Acknowledgment).

Estos caracteres son utilizados como respuesta para indicar que el bloque anterior fue aceptado sin error y que el receptor está listo para recibir el siguiente bloque. ACK0 es utilizado en bloques pares y ACK1 en nones.

- WACK (Wait-Before-Transmit Positive Acknowledgment).

Este caracter es enviado por la estación receptora para indicar que temporalmente no está lista para recibir. Adicionalmente funciona como un ACK positivo para el bloque recibido o para la selección (en líneas multipunto). Por otra parte la respuesta normal del transmisor a un WACK es ENQ, siendo válidos también EOT o DLE EOT. El receptor al recibir este caracter, continúa mandando WACK hasta estar listo para recibir.

- NACK (Negative Acknowledgment).

NACK indica que el bloque previo fue recibido erróneamente y que la estación receptora está lista para recibir la retransmisión del mismo.

- DLE (Data Link Escape).

Corresponde a una secuencia que es usada como prefijo de los caracteres de control del protocolo, cuando la transmisión es efectuada en modo transparente.

- RVI (Reverse Interrupt).

Es utilizado como un ACK positivo, pero adicionalmente es un requerimiento para la terminación de la transmisión actual debido ya sea a que un mensaje de más alta prioridad que el transmitido llega al receptor, o bien, en un ambiente multipunto, la estación de control (actuando como un receptor) necesita comunicarse a otra estación en la línea. No pueden ser transmitidos RVI consecutivos, excepto cuando son una respuesta a un ENQ.

- TTD (Temporary Text Delay).

Esta secuencia (STX ENQ) es mandada por la estación transmisora cuando se encuentra en estado de transmisión de mensaje pero no está listo para transmitir y desea retener la línea. Por su parte la estación receptora responde con NACK y espera a que se inicie de nuevo la transmisión. La secuencia de TTD es mandada generalmente después de la mitad del tiempo de espera que la estación receptora tiene para recibir el mensaje. Si la estación transmisora aún no está lista, la secuencia de TTD no puede ser repetida una o más veces.

- DLE EOT (Secuencia de Desconexión para líneas Switcheadas).

Esta secuencia es transmitida generalmente cuando se ha completado el intercambio de mensajes. La transmisión puede ser hecha desde la estación que llama o la llamada.

La figura 6.4 muestra el diagrama de flujo de la transmisión en el protocolo BSC. En este diagrama aparecen cinco estados de transmisión en modo transparente y dos en modo normal.

La figura 6.5 muestra el flujo de los estados durante el proceso de control de recepción, teniéndose seis estados de los cuales los estados 1 y 3 son utilizados para la recepción en forma normal, los estados 4, 5 y 6 para recepción en modo transparente y el estado 2 para transición entre recepción ordinaria y recepción en modo transparente.

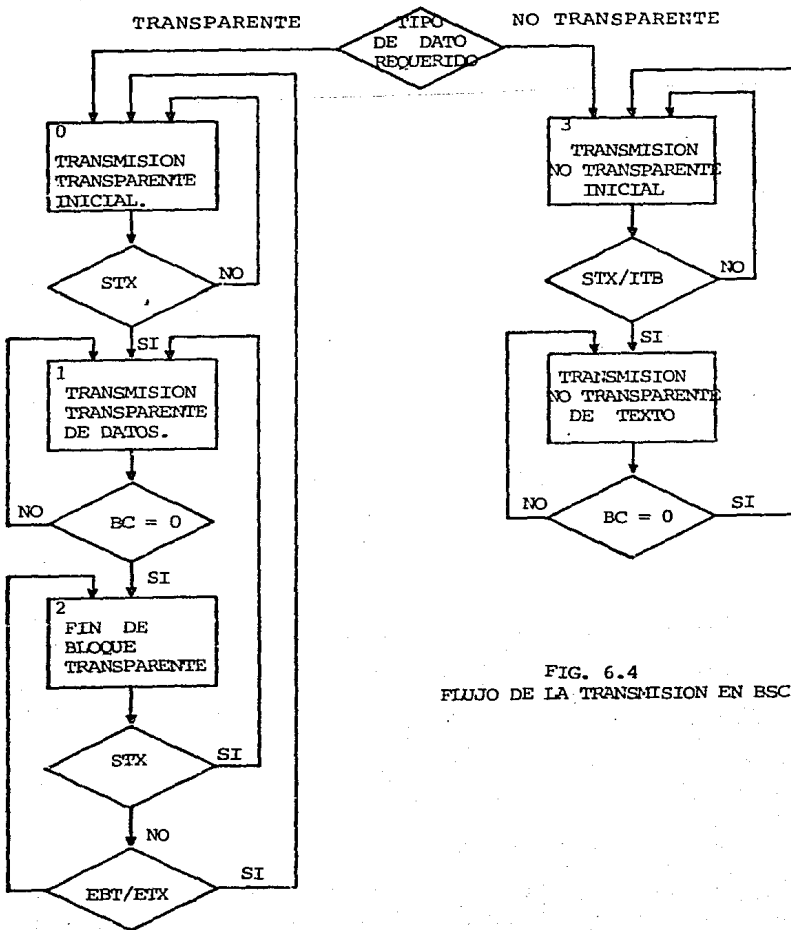
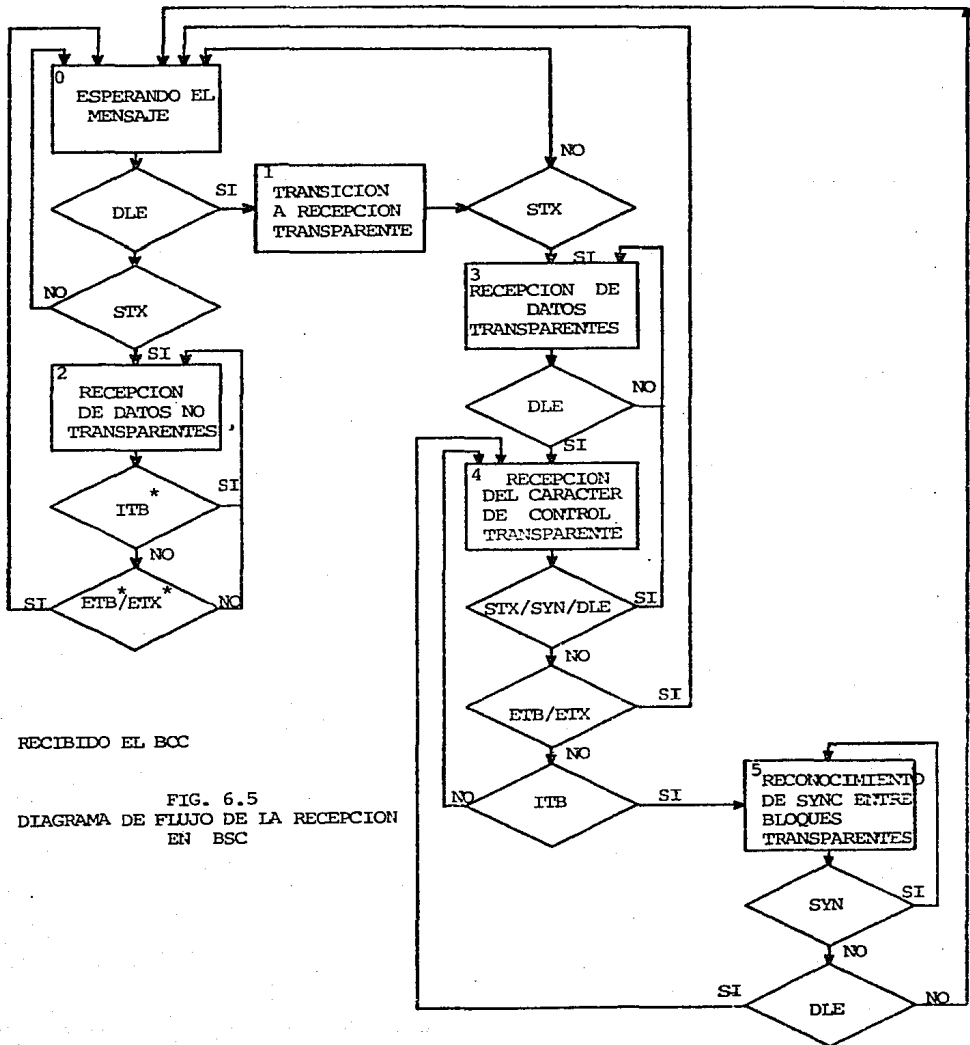


FIG. 6.4
FLUJO DE LA TRANSMISION EN BSC



DDCMP

DIGITAL DATA COMMUNICATION MESSAGE PROTOCOL

DDCMP es un protocolo diseñado para uso general por Digital Equipment Corporation, puede ser utilizado para transmisiones síncronas o asíncronas, semiduplex o duplex, en serie o paralela, punto a punto o multipunto y acepta hasta 255 mensajes de datos por uno solo de ACK, siendo más utilizado en transmisiones síncronas, en serie, semiduplex o duplex.

La figura 6.2 muestra el formato de mensajes de DDCMP. El mensaje, de igual manera que en el protocolo BSC, es dividido en 2 partes:

- Un encabezado conteniendo información de control y que constituye la parte más importante del mensaje ya que contiene los dos elementos más importantes que caracterizan a DDCMP y que son:
 - Información de la numeración de secuencia del mismo.
 - El contador de carácter (a diferencia de BSC esta parte no es opcional y además contiene su propio bloque de chequeo de redundancia ciclica -CRC1 en la figura-).

Como se puede observar en la figura 6.2, el encabezado está formado por la clase, el contador, la bandera, la respuesta, la secuencia, la dirección y el carácter de chequeo de redundancia, mientras que el texto se forma por hasta 16,363 caracteres de 8 bits de información y un segundo carácter de chequeo de redundancia correspondiente al texto.

Los componentes del encabezado en el formato de DDCMP corresponden a:

- La clase.- Es el indicador de la clase del mensaje, teniéndose tres clases:

- Datos
- Control
- Mantenimiento.

- El contador y la bandera.- Son utilizados para mensajes de datos y mantenimiento, e indican el número de caracteres que continúan en el encabezado y el número en el cuerpo. En control de mensajes, los primeros ocho bits son usados para designar qué tipo de mensaje de control es y los últimos seis bits son generalmente llenados con ceros, excepto para el caso en que el mensaje es un NACK, donde estos últimos seis bits son usados para especificar la razón del NACK.

El campo de dos bits contiene la sincronización rápida y banderas de selección. La sincronización rápida es utilizada para indicar al receptor que el mensaje estará seguido de caracteres de sincronía para reestablecer esta, mientras que las banderas de selección son utilizadas para indicar que el mensaje actual será el último a transmitir. Esto es utilizado en configuraciones semi-duplex

o multipunto donde los dispositivos necesitan ser llamados y liberados.

El campo de respuesta contiene el número del último mensaje recibido correctamente y es utilizado en mensajes de datos y en mensajes de control de reconocimiento negativo e positivo.

El campo de secuencia es utilizado en mensajes de datos y en mensajes de control tipo REF (que posteriormente se explican). En mensajes de datos contiene el número de secuencia del mensaje asignado por el transmisor, mientras que en mensajes REF es utilizado como parte de la información que pregunta por los mensajes recibidos correctamente (posteriormente se explica a detalle).

El campo de dirección es utilizado para identificar la estación tributaria en sistemas multipunto. En sistemas punto a punto generalmente es asignado en uno e ignorado.

Por último existen dos mensajes de contador que corresponden a un "inicio" o "reconocimiento de inicio" que se utilizan para inicializar los contadores de mensaje, relojes y otros contadores, siendo el reconocimiento de inicio el que indica que esto ha sido concluido.

Durante el intercambio de mensajes entre dos estaciones bajo el protocolo DDCMP, cada mensaje enviado es numerado secuencialmente empezando en 1 y siguiendo la numeración sucesiva módulo 256. Esta numeración es llevada independientemente en cada estación en una conexión multipunto, esto es, una estación puede estar transmitiendo los mensajes 11, 12, 13, mientras la otra los 3, 4 y 5.

ASI, cuando una estación transmite un mensaje le asigna el número de mensaje correspondiente colocándolo en el campo de secuencia del encabezado. Adicionalmente va llevando el contador que indica el número de mensajes que ha transmitido y un contador del número de mensaje recibido. Este último contador es actualizado cuando un mensaje es recibido con un número de mensaje mayor en una unidad que el del contador, y el contenido del contador de mensaje recibido es mandado a la otra estación en el mensaje, dentro del campo de respuesta del encabezado, para indicar que se recibió. En el caso de recibir un mensaje con error, el receptor manda un reconocimiento negativo (NACK) al transmisor. El hecho de incluir el número de secuencia del último mensaje recibido correctamente, provoca que DDCMP no requiera de un NACK o ACK por cada mensaje recibido, sino al indicar un NACK para un cierto mensaje, eso indica que los anteriores ha sido recibidos correctamente.

Cuando una estación receptora recibe un mensaje fuera de secuencia, no responde a tal mensaje, y la estación transmisora lo detecta al expirar la pausa de respuesta para recibir el ACK. Entonces el transmisor manda un mensaje REP (Replay), el cual contiene el número de secuencia del mensaje más reciente no reconocido. Si el receptor sí recibió correctamente ese mensaje responderá con ACK, en caso contrario manda un NACK y el último número de mensaje que recibió correctamente. A esto el transmisor responderá con la retransmisión de los mensajes a partir del indicado en el NACK.

SDLC.

SYNCHRONOUS DATA LINK CONTROL

Dentro de este protocolo son diferenciadas las estaciones involucradas en una liga de comunicación de datos. Se le llama "Estación primaria" a aquélla que tiene la responsabilidad de la liga de datos y lleva el control, y a la otra estación se le conoce como "Estación secundaria". Sin embargo no necesariamente la estación primaria debe iniciar la transferencia de información, sino que puede ser iniciada por la estación secundaria.

La estructura básica del formato de SDLC se muestra en la figura 6.3, donde la longitud del campo de información no se encuentra restringida, pudiendo ser incluso cero, y como máximo el límite que determinen las condiciones del canal y del receptor.

De las dos banderas que delimitan la estructura de SDLC, la bandera de comienzo sirve como referencia para la posición de los campos de dirección y control, así como para inicializar el chequeo de error de transmisión. La bandera de terminación sirve para indicar que los 16 bits anteriores corresponden a la estructura de chequeo.

Debido a que la secuencia que determina las banderas (01111110 por ejemplo) puede ser enviada en cualquier parte de la estructura, los procedimientos de SDLC requieren que un cero binario sea insertado por el transmisor después de cada sucesión de seis unos continuos en este caso. Por su parte, el receptor remueve este cero.

El campo de dirección determina el número de estación secundaria a la cual el comando de la estación primaria será enviado.

El campo de control, por su parte, puede contener uno de los siguientes tres formatos:

- Formato de transferencia de información
- Formato de supervisión
- Formato no secuenciado.

siendo el bit P/F (Poll/Final) el único elemento en común de estos tres formatos.

Una estructura con bit P (Poll-reconocimiento) encendido es mandada de una estación primaria a una estación secundaria para autorizar la transmisión, mientras una estructura con bit F (final) encendido, es mandada a la estación secundaria en respuesta a un reconocimiento. Por lo general, una estación primaria manda un determinado número de estructuras a una estación secundaria, cada una teniendo cero en el bit P/F, hasta que la estación primaria está lista para recibir la respuesta de la estación secundaria y mandar en ese momento una estructura con un uno en el bit P/F. Por su parte la estación secundaria responde

con estructuras que contienen cero en el bit P/F, hasta completar a su respuesta al reconocimiento, después de lo cual responde con una estructura que tiene uno en el bit P/F.

El formato de transferencia de información es utilizado para la transmisión ordinaria de datos y es el único de los tres formatos que utiliza la numeración de secuencia de estructura. Así, cada estructura transmitida es numerada de tal manera que la estación secundaria puede indicar a la primaria, en caso de necesidad de retransmisión por causas de errores por ejemplo, a partir de cuál estructura comenzar la retransmisión. El contador que es llevado por la estación primaria para numerar las estructuras es conocido como Ns, mientras que el contador de la estación secundaria para indicar hasta qué estructura se ha recibido sin errores se conoce como Nr. Se tiene entonces la siguiente estructura para este formato:

N2 P/F Ns 0 1

El bit que contiene cero indica que el campo de control se encuentra en formato de información. Los 3 bits de Ns indican el número de mensaje que se está enviando. El bit P/F está encendido si la estación está concluyendo su reconocimiento o bien en respuesta a éste y Nr es el número de mensaje que se esperaba recibir.

El formato de supervisión es utilizado en conjunción con el formato de transferencia de información, para controlar esta transferencia en el último formato. Aquí,

los dos primeros bits enviados, 1 y 0, indican que el campo de control se encuentra en formato de supervisión. Los siguientes dos bits indican cuál de los tres siguientes comandos sigue:

- RR (Receptor listo)
- RNR (Receptor no listo)
- REJ (Reejecución)

Un comando RR es enviado, por una estación primaria ó por una secundaria, para indicar que las estructuras anteriores a la Nr-1 han sido recibidas correctamente, estando la estación lista para recibir más. RNR tiene las mismas características el comando RR, a excepción de que indica una condición de ocupada temporalmente en la cuál no pueden ser aceptadas más estructuras que requieran de espacios en un buffer. Por su parte REJ es un comando de respuesta que es enviado para requerir transmisión o retransmisión, indicando también recepción correcta hasta la estructura Nr-1.

Por último el formato no secuenciado es utilizado para establecer modos de operación, inicialización de estaciones, etc. Como su nombre lo indica, este formato no hace uso de Ns o Nr ya que no lleva una secuencia de las estructuras. Los primeros dos bits enviados indican el formato no secuenciado, siguiendo el bit P/F. Los cinco bits restantes son utilizados para indicar comandos y respuestas que corresponden a alguno de los siguientes:

- NSI (Información no secuenciada) indica que la información es enviada separada de cualquier mensaje secuenciado anterior.

- RQI (Requerimiento para inicialización). Este comando es transmitido por la estación receptora para indicar a la transmisora que envíe un comando SIM.
- SIM (Establecimiento de modo de inicialización) es enviada por una estación transmisora e inicia procedimientos especificados por el sistema para inicializar una estación secundaria, poniendo en cero los contadores Ns y Nr de ambas estaciones.
- SNRM (Establecimiento de modo de respuesta normal) Este comando es enviado por la estación primaria e inhibe a la estación secundaria para no enviar hasta que exista un requerimiento de la estación primaria. Inicializa también los contadores Nr y Ns de ambas estaciones en cero, permaneciendo la estación secundaria en este estado hasta recibir un DISC o SIM, y respondiendo a este comando con NSA.
- RQL (Requerimiento en línea) es transmitido por una estación secundaria para indicar que está desconectada.
- DISC (Desconectar).- Este comando pone a la estación secundaria fuera de línea, la cual responde con NSA y permanece en este estado hasta recibir un SNRM o SIM.
- NSA (Reconocimiento no secuenciado).- Esta respuesta corresponde a la esperada al enviar los comandos SNRM, DISC o SIM.
- CMDR (Reejecución de comando) es la respuesta enviada por una estación secundaria cuando recibe un comando inválido. Una estructura conteniendo CMDR en el campo de control contiene a continuación un campo de información en formato de transferencia de información que comprende el contador Ns actual de la estación, el contador Nr y cuatro bits que indican:

- 1.- Un comando inválido.
- 2.- Un campo de información asociado con un comando que se suponía contenía uno.
- 3.- Un campo de información demasiado grande que provoca sobreflujo de un buffer.
- 4.- El Nr no coincide con el Ns enviado .

- ORP (Reconocimiento de Respuesta Opcional) que es una invitación de transmisión de la estación secundaria direccionada.

Una diferencia con los protocolos anteriores es la complejidad para calcular el carácter de chequeo de bloque. Esta complejidad radica primero en que la estación transmisora comienza con un valor de residuo. Este valor es pre-multiplicado por X^{16} y dividido por el polinomio $X^{16}-X^{12}-X^5-1$ enviando el complemento del residuo, empezando por el bit de mayor orden.

Otra diferencia importante es la manera prematura de terminar una liga de datos, conocida como aborto, la cual es efectuada por la estación primaria, que envía ocho unos consecutivos, seguido de al menos otros siete unos o bien una bandera que borra el CRC del receptor.

CAPITULO VII

CAUSAS, TIPOS Y TRATAMIENTO DE ERRORES.

Dentro de todos los medios eléctricos de transmisión de información deben ser considerados los efectos del ruido. Ruido es cualquier señal no deseada que puede ser originada por falsos contactos, cortos en los medios, cambios de voltajes, perturbaciones naturales, etc.

Existen varias categorías dentro de las que se puede clasificar el ruido y distorsión de una línea :

- Ruido blanco o Gaussiano que es el fondo o estática conocida en radios y teléfonos. Debido a su naturaleza es inevitable, ya que se debe a una agitación térmica de electrones. Sin embargo este tipo de ruido por lo general no es muy problemático a menos que su nivel sea muy elevado.
- Ruido de impulsos (conocido como agujas) que es la categoría de ruido más frecuente en los medios de transmisión de datos, y que durante la comunicación de voz se escucharía como un "click". Generalmente con una duración de una centésima de segundo durante una transmisión de datos, puede alterar un grupo de bits de datos provocando errores. Por ejemplo, a razón de 150 bps, una aguja de una centésima de segundo cambia en promedio uno o dos bits, mientras que a 4800 bps cambia 48 bits.

- Líneas cruzadas que ocurren cuando una línea toma parte de la señal que va por otra. Este tipo de ruido se presenta entre pares de líneas que llevan señales separadas, en líneas multiplicadas que llevan muchas señales discretas, en enlaces de microondas en que una antena recoge una pequeña porción reflejada de la señal de otra antena en la misma torre y en general en circuitos telefónicos de alambrado fijo que corren paralelos unos a y otros que se encuentren demasiado próximos entre sí sin estar eléctricamente balanceados.

- Ruido de intermodulación que es un tipo especial de cruce. Las señales de dos líneas independientes se intermodulan y forman una señal que cae dentro de una banda de frecuencias que difiere de ambas entradas. En una línea multiplicada, varias señales distintas se simplifican juntas y las ligeras variaciones en el equipo pueden provocar ruidos de intermodulación o bien, un convertidor de señal mal ajustado puede transmitir un tono de frecuencia intenso cuando no está transmitiendo datos, provocando este tipo de ruido.

- Ruido de amplitud que comprende un cambio repentino en el nivel de potencia. Su efecto depende del tipo de modulación que utilice el convertidor de señal.

- La pérdida de línea que es una causa catastrófica de errores y transmisión incompleta. Sus causas pueden ser pérdida de la señal portadora, fallos del equipo conmutador o fallos que provoquen una línea abierta o corto circuito.

- La atenuación es la pérdida de potencia que sufre la señal al pasar del dispositivo transmisor al receptor. Depende de la distancia entre éstos y las características del medio y trae como consecuencia la necesidad de amplificadores de señal.

Existen otras categorías de errores menos comunes, siendo la característica más importante a considerar en todas las que se mencionaron la relativa larga duración de las distorsiones como anteriormente se mencionaba, una distorsión de una centésima de segundo en una transmisión a 4800 bps causa una variación en 48 bits de datos; cuando el ruido causa que un bit sea recibido con error, generalmente causa que un gran número de bits de datos sean afectados). Generalmente, períodos de alto porcentaje de error se encuentran separados por intervalos relativamente largos en los que se tiene bajo nivel de ruido y bajo porcentaje de error en la recepción, por lo que el promedio de error en un intervalo de una hora es generalmente un error en 100 000 bits transmitidos.

Una de las técnicas más simples utilizada en sistemas de transmisión asíncrona para determinar cuándo los bits correspondientes a un carácter han sido recibidos apropiadamente consiste en buscar que todos los caracteres transmitidos tengan un número impar, o par, de unos para lo cual se aumenta un bit a cada carácter con un cero o uno dependiendo del número de unos que tenga aquél. Por ejemplo el carácter 0101100 se transmitiría como 00101100 debido a que tiene un número impar de ceros, y el carácter 11001100 se transmitirá como 111001100. Este bit adicionado como último bit del carácter es conocido como el bit de paridad. Bajo la filosofía de marcar un número impar de unos se le llama paridad impar. De igual manera podemos asumir que se busque un número par, por lo cual se checaría una paridad par y el bit adicionado sería uno o cero para obtener así un número par de unos. A este método de chequeo se le conoce

como Chequeo Vertical de Redundancia o VRC. (Vertical Redundancy Check). En un sistema de chequeo de paridad, el transmisor calcula el estado del bit de paridad y lo adiciona al caracter durante la transmisión. Por su parte, el receptor calcula de igual manera el estado del bit de paridad y lo compara con el bit de paridad recibido. Si estos no coinciden el receptor sabe que al menos un bit se ha modificado, y por lo tanto el caracter ha sido recibido con error.

Sin embargo, considérese el siguiente caso:

El transmisor envía la secuencia 101101100, y se checa paridad impar. Suponiéndose que los bits cuatro y cinco de derecha a izquierda son recibidos erróneamente, se recibiría el caracter 101110100. La paridad sigue siendo impar y todo parece estar bien después de un error doble. De aquí podemos deducir que la detección de errores por medio de paridad sólo es efectiva en el caso de que el número de bits afectados sea impar, ya que de otra forma el error no es detectado, independientemente de que se cheque paridad par o impar.

Otra técnica de detección de errores consiste en enviar después de un determinado número de caracteres, otro caracter conocido como caracter de chequeo, que contiene uno o cero en una determinada columna, si el número de unos de los caracteres transmitidos considerados dentro de la misma columna es par o impar, dependiendo del tipo de paridad que se cheque (supongamos que impar). Adicionalmente cada caracter, incluyendo el caracter de chequeo, contiene el bit de paridad de la técnica anterior. Por ejemplo, supóngase

que se checa paridad impar y que se envía la siguiente secuencia:

101101100	Caracter 1
110101111	Caracter 2
001110101	Caracter 3
111100010	Caracter 4
100010111	Caracter 5
010111100	Caracter de chequeo

el tercer bit (de derecha a izquierda) es uno debido a que el número de unos en el tercer bit de los cinco caracteres anteriores es un número par.

Supóngase que el segundo y tercer bits del caracter 1 se reciben erróneamente. Esto implica que el caracter de chequeo que calcula el receptor es 010111010 que no coincide con el caracter de chequeo enviado, con lo cual se detecta un error.

Considérese ahora el caso en que el primer y tercer caracter, por ejemplo, son recibidos con errores en las columnas dos y tres. En este caso, el caracter de chequeo obtenido por el receptor y el recibido coinciden, sin embargo, existe un error. Similarmente que en el chequeo de paridad de un caracter donde no se detectaba un error doble, un chequeo de paridad por columnas conocido como Chequeo Longitudinal de Redundancia o LRC, falla cuando un error doble se presenta en una misma columna.

Los sistemas más efectivos de detección de errores dentro de los sistemas de comunicación de datos son aquellos que utilizan el método de chequeo de Redundancia Cíclica (CRC-Cyclic Redundancy Check). En este método el cálculo efectuado es generalmente hecho en un registro de corrimiento de sección múltiple, al cual se alimenta el dato entrando a una compuerta OR-exclusiva. A su vez la salida de esta compuerta se alimenta de nuevo a otra localizada entre las secciones del registro de corrimiento. A este registro de corrimiento se le conoce como Registro de Chequeo de bloque (BCK-Block Check Register) del que se ilustra una de sus posibles configuraciones en la Fig. 7.1.

Dentro de los métodos de chequeo de redundancia cíclica se encuentran CRC-12, CRC-16 y CRC-CCITT (Comite Consultatif Internationale de Telegraphic et Telephonie) en los cuales la posición y cantidad de las compuertas OR varía .

En una compuerta OR-exclusiva se tienen las siguientes reglas (fig 7.4):

- la salida es cero si ambas entradas son iguales (cero o uno)
- la salida es uno si las entradas difieren.

Dentro de un sistema de transmisión el registro de corrimiento es inicializado en ceros. Cada bit que es transmitido por la línea de comunicación es llevado al punto A, de la fig.7.1. Por ejemplo supongamos que seguimos la lógica de la fig. 7.2, inicializaríamos como se indica en la figura inicializando el registro en ceros.

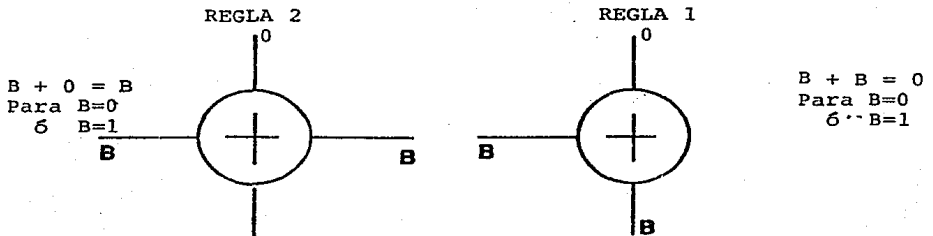
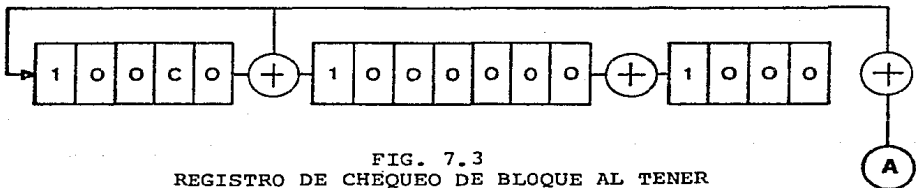
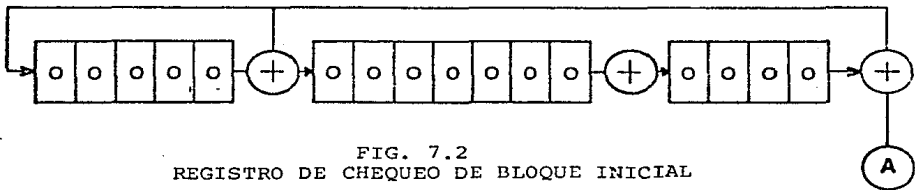
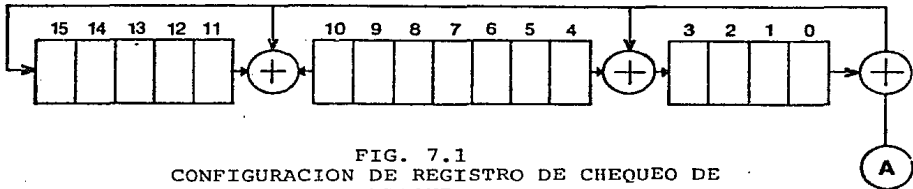


FIG. 7.4
REGLAS DE LAS COMPUTERTAS OR - EXCLUSIVAS

Si suponemos ahora que el primer bit transmitido es un uno, tendríamos el resultado de la figura 7.3.

Y así podríamos ir aplicando el método a los bits consecutivos posteriores. Cabe hacer notar que el efecto de un bit ya aplicado permanece durante un período relativamente largo de tiempo. Esta técnica no solo es aplicada al bit por el transmisor, sino que el receptor aplica la misma técnica a cada bit recibido. Una vez concluida la transmisión del mensaje, el transmisor manda el contenido del registro de corrimiento resultante de la transmisión hacia el receptor, el cual le aplica el método sobre el registro que él obtuvo. Considérense las reglas de la compuerta OR-exclusiva ilustradas en la figura 7.4:

Como el primer bit del carácter del registro de corrimiento es mandado al punto A del registro del receptor y en principio estos registros deben ser iguales, la compuerta OR-exclusiva debe obedecer la regla 1 y por tanto producir un cero, el cual afecta en las siguientes compuertas - debido a la regla 2 - corriendo una posición a la derecha los bits contenidos en el registro y dejando un cero en el último bit de la izquierda de cada sección del registro del receptor. Esto sucede para cada bit del registro de corrimiento transmitido. Al terminar eventualmente de transmitir este registro, en el caso de no haberse presentado fallas en la transmisión, el registro de corrimiento del receptor debe contener cero en todas sus posiciones, independientemente de la posición y cantidad de las compuertas.

Existen protocolos, como lo son SDLC o HDLC, que inicializan el registro con un valor pre-establecido y terminan con un valor distinto de cero.

En comparación con otras técnicas de detección de errores, el chequeo de Redundancia cíclica no requiere de un bit extra, como en el caso del chequeo de paridad. Sin embargo, el transmisor envía generalmente dos veces el contenido del registro de corrimiento al final de cada bloque de caracteres. Esta explicación de la técnica puede ser presentada de una manera formal, expresada en términos matemáticos, de la siguiente manera:

Considérese que un mensaje en esta técnica consiste de un determinado número de bits transmitidos en el bloque que define el mensaje y un determinado número de bits para la transmisión del registro de corrimiento. Sean:

n = número total de bits transmitidos.
 K = número de bits del mensaje.
 $n-K$ = número de bits del o de los registros de corrimiento transmitidos (generalmente se transmite dos veces).

El código del mensaje es derivado de dos polinomios que son la representación algebraica de dos palabras binarias, el polinomio generador $G(X)$ y el polinomio de mensaje $M(X)$.

El polinomio generador es el tipo de código utilizado (CRC-12, CRC-16 ó CRC-CCITT) y el polinomio de mensaje es la cadena de bits de datos seriados. Los polinomios son generalmente representados por una cadena en términos de potencias de X de la forma:

$$X^{2n} - X^{2n-1} - \dots - X - 1$$

Por ejemplo el polinomio $X^{25} - X^{23} - 1$ representa el código binario 101001 (la ausencia de un término se indica con cero y la presencia con un uno).

Así, dado un polinomio de mensajes $M(X)$ y un polinomio generador $G(X)$ el objetivo es construir un polinomio de código de mensaje $C(X)$ que sea divisible por $G(X)$, el cual se obtiene bajo el siguiente algoritmo (tomándose en cuenta las definiciones anteriores de n , k , y $n-k$) :

- 1.- Multiplicar $M(X)$ por X^{2n-k}
- 2.- Dividir el resultado del producto por el polinomio generador $G(X)$.
- 3.- Descartar el cociente y sumar el residuo $R(X)$ al producto:

$$C(X) = X^{2n-k} * [M(X)] - R(X)$$

La división es efectuada en álgebra binaria, de donde el resultado es siempre de longitud un bit menor que la del divisor. El residuo es el registro de corrimiento, por lo que la longitud del registro es un bit menos que la del polinomio generador.

Por ejemplo:

Considérense los siguientes valores:

$$M(X) = 110011 \quad (X^{**5}-X^{**4}-X-1)$$

$$G(X) = 11001 \quad (X^{**4}-X^{**3}-1)$$

$M(X)$ contiene seis bits de datos y $G(X)$ contiene cinco. Supongamos que el registro de corrimiento tiene cuatro bits, entonces :

Paso 1: obtener $X^{n-k} * [M(X)]$

$$\begin{aligned} X^{n-k} * [M(X)] &= X^4 * (X^{**5}-X^{**4}-X-1) = \\ &X^{**9}-X^{**8}-X^{**5}-X^{**4} = \\ &1100110000 \end{aligned}$$

Paso 2: Obtener $R(X) = Q(X) G(X) - X^{n-k}[M(X)]$ con $Q(X)$ el cociente de dividir $X^{n-k}[M(X)]$ entre $G(X)$.

$$\begin{array}{r} 10001 \\ \text{-----} \\ 11001 1100110000 \\ 11001 \\ 10000 \\ 11001 \\ 1001 \end{array}$$

De donde $R(X) = 1001$

Paso 3: Obtener $C(X)$

$$\begin{aligned} C(X) &= X^{n-k} [M(X)] - R(X) \\ 1100110000 - 1001 &= 1100111001 \end{aligned}$$

$C(X)$ es transmitido y el receptor divide $C(X)/P(X)$. Si no existió error en la transmisión $R(X)=0$. Si $R(X)$ es distinto del polinomio cero, esto indica que hubo un error.

Dentro de una configuración típica de Hardware de comunicación, la configuración del registro está basada en el código CRC a ser utilizado. El número de pasos es igual al grado del polinomio generador. El número de OR-exclusivos está también en función del polinomio.

CRC-12 es aplicado para sistemas síncronos que usan caracteres de seis bits. La acumulación del carácter de chequeo de bloque es de 12 bits, y el polinomio generador es:

$$X^{12}-X^{11}-X^3-X^2-X-1$$

(que es el resultado del producto de los factores primos $(X-1)$ y $(X^{11}-X^2-1)$), teniéndose así detección de errores en ráfagas de hasta 12 bits de longitud.

CRC-CCITT es el estandar para calcular el carácter de chequeo de bloque en sistemas de ocho bits por carácter y la acumulación del carácter de chequeo de bloque es en 16 bits.

El polinomio generador $G(X) = X^{16}-X^{12}-X^5-1$ proporciona detección de errores de ráfagas hasta de 16 bits y adicionalmente, más del 99% de detección de error en ráfagas de longitud mayor a 12 bits.

Por último CRC-16 es aplicado en sistemas síncronos de ocho bits por carácter, con una acumulación de 16 bits del

caracter de chequeo de bloque y un polinomio generador $G(X)$
= $X^{16} - X^{15} - X^2 - 1$.

CAPITULO VIII

COMPONENTES Y FUNCIONES A EJECUTARSE DENTRO DE UN SISTEMA DE TRANSMISION DE DATOS

Existe una gran diversidad de equipo que es utilizado en un sistema de comunicación de datos. Tradicionalmente la configuración de una red de comunicación de datos está compuesta por los elementos que se indican en la fig. 8.1

Dentro de este capítulo serán descritas las características y funciones de las principales partes que se ven involucradas en esta configuración típica.

PROCESADORES DE COMUNICACIONES

La principal característica de los procesadores de comunicaciones o controladores de comunicaciones como también se les conoce, radica en que liberan al procesador central de consumir tiempo en tareas para el manejo y control de los procesos correspondientes a la comunicación de datos. Así, esta liberación de memoria y tiempo proporciona una mayor utilización de los recursos del procesador central en otro tipo de aplicaciones más críticas.

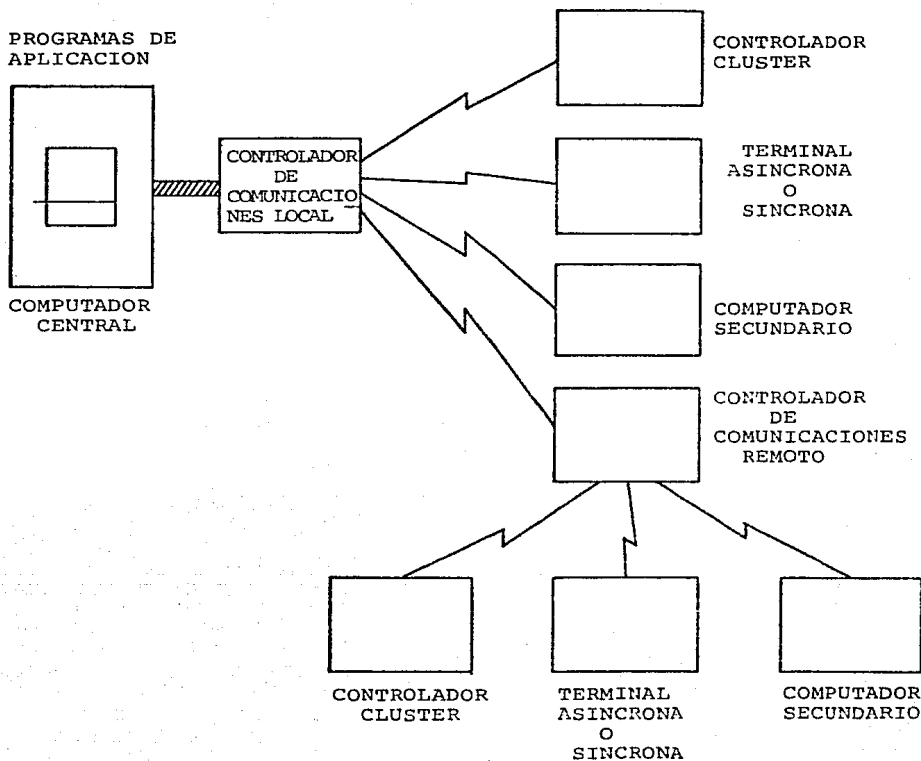


FIG. 8.1
CONFIGURACION TIPICA DE UNA RED DE COMUNICACION
DE DATOS

Adicionalmente la eficiencia de estos controladores se ve incrementada debido a la facilidad de soportar una gran variedad de dispositivos remotos que a su vez hacen uso de distintos protocolos, distintos códigos de transmisión, distintas velocidades de transmisión, etc.

Así, los procesadores de comunicación ofrecen capacidades superiores de entrada/salida para dispositivos remotos y además proporcionan:

- Facilidades de almacenamiento de datos.
- Conversión y formateo de código.
- Funciones de control de línea, terminales y red.
- Facilidades de programación.

En una configuración típica su principal función es de servir de controlador del tráfico, enrutando los mensajes recibidos de terminales remotas, examinando su contenido para efectuar posibles conversiones de código, determinando su destino correcto y finalmente transmitirlo o retransmitirlo para procesos subsecuentes. Estas constituyen las características básicas de un controlador de comunicaciones programable. Además, puede contar con otras más sofisticadas como son:

- Recuperación y control de errores.
- Rutinas de control de acceso.
- Monitoreo de todo el tráfico de comunicación de datos.

- Control del rendimiento de la red.

La identificación y el diagnóstico del tráfico de la línea y del funcionamiento de terminales es en sí lo que se conoce como una función de vigilancia secundaria de un controlador de comunicaciones. Es esta función la que permite detectar fallas tanto en la línea como en los dispositivos y efectuar los procedimientos necesarios para corregirlas.

Dentro de los distintos tipos de controladores y configuraciones de los mismos, la diferencia básica se encuentra en el tamaño de la red que soportan y la diversidad de dispositivos que le pueden ser conectados. De esta manera, un procesador de comunicaciones puede ser visto como una microcomputadora de propósito dedicado con capacidades de soporte de software, el cual se encuentra colocado entre la interfase del computador central (ya que son conectados a canales byte o block-multiplexor) y la interfase de la línea, del lado del equipo terminal. La capacidad de soporte de software permite al controlador ofrecer una variedad de rutinas programadas (macros) para soportar distintas disciplinas de líneas, permitiendo en algunos casos contar además con almacenamiento secundario por medio de otros dispositivos y un panel o consola para monitoreo y control.

La figura 8.2 muestra las componentes comunes de un controlador de comunicaciones. El adaptador es la unidad Hardware que permite conectar el controlador al computador central a través de un canal paralelo byte ó block-multiplexor, de alta velocidad. A continuación se encuentra

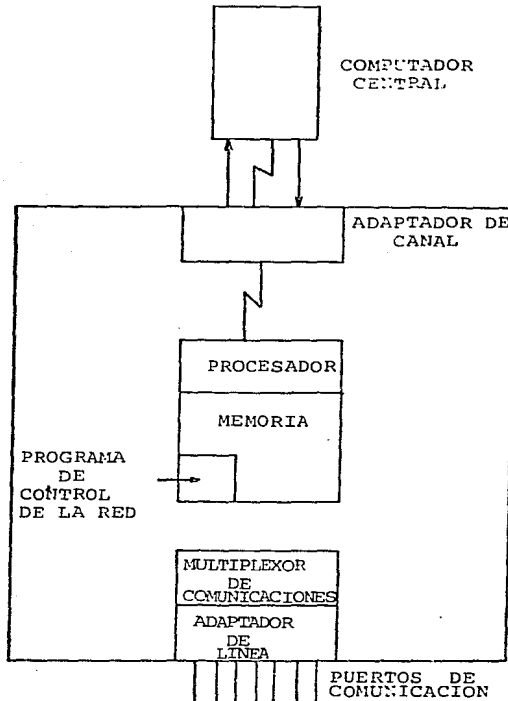


FIG. 8.2
PARTES ESENCIALES DE UN CONTROLADOR DE COMUNICACIONES

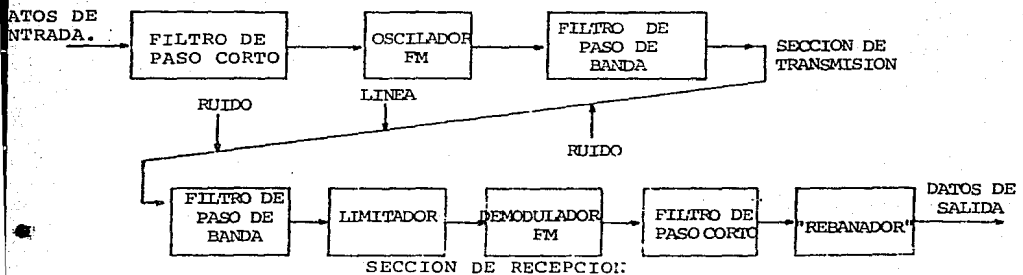


FIG. 8.3
PARTES ESENCIALES DE UN MODEM FM

el procesador dedicado y la memoria dentro de la cual está el programa para el control de la red. Por su parte el multiplicador de comunicaciones distribuye los recursos del procesador entre las líneas de salida de acuerdo a procedimientos de direccionamiento, prioridades, etc. El adaptador de línea efectúa la interfase de cada línea con las facilidades de la red, como son transmisión Semi-duplex o duplex, modems, unidades de auto-llamada, puertos RS-232 e interfases comunes.

Por último se tienen los puertos para la conexión de terminales por medio de convertidores de señal.

También pueden ser clasificadas las funciones que efectúa un controlador o procesador de comunicaciones en cuatro categorías de acuerdo a las necesidades y requerimientos que una red de comunicación de datos debe satisfacer. Estas cuatro categorías de funciones son:

- Procesamiento de las funciones de comunicación.
- Switcheo de mensajes y enrutamiento.
- Control del manejo de la red.
- Control de errores y diagnósticos.

PROCESAMIENTO DE LAS FUNCIONES DE COMUNICACION.

Como anteriormente se mencionaba, la función más importante de los procesadores de comunicación consiste en descargar al computador central del proceso de las funciones de manejo y control de comunicación de datos, efectuando así las funciones de control de mensajes, de terminales, líneas y preprocesos. Dentro de este tipo de funciones efectuadas por el procesador se encuentran, siendo las más importantes:

- Inicialización de la comunicación de datos.- Esta función consiste de una invitación a las terminales para transmitir, un direccionamiento a éstas para recibir un llamado para líneas conmutadas, etc. Generalmente el orden en el cual las líneas deben ser servidas en la red es controlado por medio de una lista de líneas que especifica este orden, y otra lista de estaciones para cada línea indicando la secuencia de servicio de esa línea, o bien, esa secuencia es activada por demanda.

- Conversión de código.- Permite comunicar terminales que no utilizan el mismo código (por ejemplo ASCII o EBCDIC), siendo el controlador el encargado de la conversión.

- Conversión de velocidad.- Debido a que existe un rango muy amplio de velocidades que oscila entre 30 y 19200 BPS's, esta función permite comunicar terminales a diferentes velocidades.

- Edición/Validación.- Es utilizada para prevenir mensajes formados erróneamente o mensajes direccionados para que pasen por toda la red.

SWITCHEO DE MENSAJES Y ENRUTAMIENTO.

Esta función puede ser ejecutada conjuntamente por el controlador de comunicaciones junto con el procesador central. Permite que los mensajes recibidos de terminales remotas puedan ser analizados, comprimidos, almacenados y enrutados a su destino a tiempo.

Dentro de las subfunciones efectuadas dentro de esta función se encuentran:

- Switcheo de mensajes.- Permite que el procesador de comunicaciones se encargue del enrutamiento de las distintas terminales remotas, utilizando el concepto de "almacena y hacia adelante", en el cual el controlador recibe los mensajes de las localidades remotas, analiza su contenido y almacena los datos para transmitirlos al procesador central.

- Enrutamiento. - Aquí se asocian los mensajes con su destino correspondiente, el cual puede ser terminales, otra computadora etc. Este enrutamiento puede ser llevado a cabo explícitamente (por encabezados en el mensaje), implícitamente (basado en parámetros como son el origen del mensaje), o una combinación de ambos.

- Ensamblado Caracter/Mensaje.- Debido a que los dispositivos generalmente transmiten en serie a baja velocidad, y el procesador en modo paralelo, esto hace

necesario ir almacenando los caracteres de un mensaje antes de poder ser procesados y viceversa al recibir. Además, se debe remover información de control en el mensaje.

CONTROL DEL MANEJO DE LA RED.

El procesador de comunicaciones ayuda al manejo del control de la operación de la red, manteniendo un registro del tráfico de ésta (por ejemplo, el número de mensajes transmitidos, estadísticas de error, etc). Algunos dispositivos también cuentan con diagnósticos de red para detectar y localizar fallas de línea.

Las subfunciones en este caso comprenden:

- Concentración de líneas remotas.- Debido a la relativa baja velocidad de transmisión de los dispositivos remotos, el controlador se encarga de concentrar las transmisiones y enviarlas a altas velocidades al procesador, todo esto en un dispositivo llamado concentrador (que posteriormente se define) el cual usa sus periféricos de almacenamiento para almacenar los datos transmitidos, evitando así retardos injustificados en los puntos terminales y promoviendo una optimización del flujo de datos.
- Supervisión de la red.- Esta función es efectuada generalmente por un conjunto de comandos que permiten habilitar, deshabilitar y modificar en general el funcionamiento de la red.
- Manejo de colas.- Se encarga de la administración y manejo de colas efectuando las funciones propias de estos procesos como son eliminar, anexar, mover entre colas, etc.

CONTROL DE ERRORES Y DIAGNOSTICOS.

Ya que existe una alta probabilidad de error en la transmisión debido a la naturaleza de los dispositivos de comunicación, es necesario contar con capacidades de detección de error (discernirlos y clasificarlos de acuerdo a su tipo y origen) y contar también con procedimientos de recuperación, los cuales sean ejecutados por el controlador mismo.

Las subfunciones que se tienen son:

- Chequeo de errores.- Los errores pueden tener su origen en muchos puntos del sistema de transmisión, como puede ser: en fallas de la línea, mal funcionamiento de equipo (modems, dispositivos, etc). De esta manera surgen distintas técnicas de detección y recuperación de errores (que se explican en el Capítulo VII.
- Diagnósticos.- Dentro del control de errores se incluye la identificación y diagnóstico de líneas y fallas de equipos terminales pudiéndose así detectar el origen de los errores.

SOFTWARE DE COMUNICACIONES.

El software de comunicaciones consiste de un conjunto de programas con un fin específico y que pueden ser básicamente de dos tipos:

- Software que proporcione una interface de los programas de aplicación de los usuarios con los recursos de comunicación y Software que efectúe el control de la red y terminales (conocido como métodos de acceso). En sí, el método de acceso de comunicaciones es un programa ofrecido por los distribuidores de equipo, diseñado para el control de la comunicación del procesador central con los dispositivos remotos. Dentro de los métodos de acceso más conocidos y que en un principio establecen los estándares, se encuentran los métodos distribuidos por IBM y que comprenden:

- BTAM (Basic Telecommunications Access Method) Fue el primer método de acceso desarrollado por IBM y proporciona facilidades básicas para comunicación de datos y para invitación (polling) y direccionamiento de terminales. Sin embargo no soporta todas las funciones de control de mensajes requeridos en un sistema de comunicación.

- QTAM (Queue Telecommunications Access Method). Incorpora la catalogación y alojamiento de recursos, y colocación de mensajes en disco, así como un lenguaje macro para acceder las funciones de control de red.

- TCAM (Telecommunications Access Method) Ofrece un programa de control de mensajes para efectuar todas las funciones de control de la red y recepción de mensajes.

- VTAM (Virtual Telecommunications Access Method) Brinda las facilidades de TCAM, pero adicionalmente opera entre particiones múltiples y establece la interface con una gran variedad de programas de aplicación.

Actualmente con el concepto SNA (System Network Architecture), de IBM, fue desarrollado ACF/VTAM (Advanced Communications Function/Virtual Telecommunications Access Method) que soporta equipos y software de la llamada cuarta generación.

EMULADORES

Dentro del Software más importante desarrollado para comunicaciones se tienen a los emuladores. Un emulador es un programa desarrollado para equipos inteligentes (como son microprocesadores o miniprocesadores) que da acceso a un puerto de comunicación en serie simulando las características hardware que controlan los elementos involucrados en la comunicación entre dispositivos y dentro de las que se incluyen:

- Modos de transmisión.
- Protocolos de comunicación.
- Control de dispositivos.
- Direcciónamiento e invitación (polling).
- Manejo de distintos códigos.
- Manejo e interpretación de estructuras de comunicación etc.

De esta manera pueden ser intercomunicados dispositivos que operan con distintas características como puede ser distintos Sistemas operativos, distintos códigos y protocolos de comunicación etc, de tal manera de poder adicionar a esta intercomunicación características que no se limiten a funcionar como dispositivos "tontos", esto es, como simple equipo terminal, sino dando a la conexión características "inteligentes" como en el caso de la

conexión de un microprocesador con un equipo mini o macro, de tal manera que el microprocesador funcione como terminal del otro equipo, brindando a la comunicación capacidades como pueden ser la capacidad de transferencia de información entre los distintos equipos independientemente de las características operativas de éstos.

Un ejemplo de esto es la conexión de los equipos Burroughs B-25 bajo el sistema operativo BTOS y PDP-11/40 operando bajo el sistema RSX-11M. En esta conexión el equipo de microprocesador B-25 emula una terminal VT-100 del equipo PDP-11/40. Adicionalmente existe la posibilidad de transferencia de información almacenada en los distintos dispositivos de ambos equipos independientemente de sus diferencias.

Para dicha conexión es necesario considerar las siguientes características Hardware:

El equipo B-25 cuenta con dos puertos de transmisión en serie de comunicación RS-232 hembra manejándose las siguientes señales básicas en el conector para una conexión local (siendo esenciales únicamente aquellos marcados con *, para conexión local).

TERMINAL DEL CONECTOR	FUNCION
1	Tierra de protección
* 2	Transmisión de datos
3	Recepción de datos

TERMINAL DEL CONECTOR	FUNCIÓN
* 4	Petición para transmitir (RTS)
5	Preparado para transmitir (CTS)
* 6	Aparato de datos preparado (DSR)
* 20	Terminal de datos preparado (DTR)

A dicho puerto será conectado el puerto para terminal proveniente del equipo PDP-11/40 el cual maneja las mismas señales en los pines correspondientes.

En caso de que el puerto para terminal sea un conector 'macho' puede ser efectuada la conexión directamente; si no, es necesaria una extensión, macho-macho con correspondencia uno a uno en sus pines, esto es, no es necesario invertir señales opuestas.

Una vez tomadas en cuenta estas consideraciones, es necesario desarrollar el Software correspondiente en ambos equipos por llevar a cabo la comunicación y transferencia de información.

En el equipo de microprocesador es necesario el emulador que cuente con las siguientes 3 opciones dentro de su operación, las cuales son llamadas al oprimir la tecla indicada entre paréntesis.

1.- Transmitir (<F1>)

En la cual será transferida información de los dispositivos del equipo B-25 hacia el disco del equipo PDP 11/40.

2.- Recibir (<F2>)

Recibir información proveniente de la PDP hacia disco rígido o diskette del equipo B-25.

3.- Terminar (<FIN>)

Para terminar el emulador y volver al estado de microprocesador, esta opción permite entrar y salir a ambos equipos para efectuar chequeos o procesos intermedios en la comunicación.

Para establecer la comunicación el emulador considera al dispositivo del puerto de transmisión en serie como un archivo secuencial en el cual es chequeado en cada ciclo el arribo de caracteres para su proceso y el envío de cada caracter teclado en el microprocesador.

Por otra parte, en lo correspondiente al equipo PDP, no es necesario ningún Software para establecer la comunicación como terminal con el microprocesador, siendo necesario solo para la recepción y el envío de información entre dispositivos. Este Software receptor y emisor debe estar sincronizado de alguna manera para evitar el riesgo de pérdida de información debido a las posibles diferencias de velocidades de operación. A velocidades superiores a los 2400 bps, es necesario suspender la transmisión cada determinado número de caracteres para poder recibirlos correctamente y en su totalidad. Esto se logra por medio del envío de los caracteres XOFF, para detener la transmisión temporalmente, y XON para reiniciarla. Estos caracteres son

válidos en la transmisión asíncrona para controlar el flujo de información entre los dispositivos enlazados.

Considerado todo lo anterior en el apéndice se encuentra el programa correspondiente para el equipo B-25, desarrollado en Pascal del Sistema Operativo RTOS V.4.0., a través del cual se logra la comunicación con el equipo Digital.

Este programa en su estructura general define al puerto de comunicación del equipo B-25 como un archivo secuencial de tal manera de tener una secuencia de chequeo en la cual se verifica en cada ciclo de esta secuencia si han llegado caracteres del equipo Digital hacia el puerto para ser procesados, o bien, si han sido tecleados caracteres en la terminal (equipo B-25), Cada uno de los caracteres tecleados es enviado conforme son dados hacia el puerto de comunicación para ser procesados en el equipo Digital el cual, en el caso de tener definido ese puerto de terminal como un puerto con "eco", regresara el caracter de tal manera de ser desplegado en el video, en caso contrario, no podrá ser visto el caracter tecleado.

Por su parte los caracteres que llegan al puerto de comunicación son analizados de tal manera que si corresponden a un caracter de escape que define un atributo de video, es transformado a través de una tabla en la correspondiente secuencia del equipo B-25. En caso contrario, el caracter simplemente es desplegado en el video. De esta manera se tiene un chequeo constante tanto en el puerto como en el teclado de tal manera de permitir la continuidad en la comunicación.

En la parte de transmisión y recepción de archivos entre los distintos equipos es tomada en cuenta la diferencias de velocidades entre los procesos de transmisión y recepción a través de poner un tiempo de retardo entre cada bloque transmitido.

Por el lado del Equipo PIP se requiere de un programa que reciba y transmita lo que el Equipo Burroughs B-25 requiera y que debe ser arrancado simultáneamente o desde el mismo emulador una vez que ha sido elegida alguna opción de transferencia de datos.

Este programa tiene que leer o escribir desde o hacia el puerto de terminal en la cual se encuentra conectada el equipo Burroughs B-25. Para esto debe ser abierto este puerto con la filosofía de un archivo secuencial que en Fortran de PDF sería:

```
CALL ASSIGN (1,'TT4')
```

Suponiendo que el puerto donde se encuentra conectado el Equipo Burroughs B-25 es el correspondiente a la terminal 4.

MODEMS

El término MODEM surge de las funciones que cumple este dispositivo ya que para enviar datos digitales binarios sobre una facilidad de transmisión, el modem efectúa la función que MÓdula la señal digital en una señal equivalente analógica, y a la vez en el otro extremo, funciona como receptor que se encarga de DEModular la señal analógica recibida para convertirla en la señal digital original.

Los modems o convertidores de señal pueden ser clasificados en tres categorías:

- La primera y la más común de estas categorías comprende a los MODEMS que son conectados a equipo terminal por medio de cables que contienen las interfases adecuadas en sus extremos para conectar el equipo y cuyas características básicas se mencionan en el párrafo anterior.
- La segunda categoría está compuesta por los acopladores telefónicos o simplemente acopladores. Un acoplador es un modem con entrada y salida audible, por lo cual la transferencia de datos puede ser llevada a cabo colocando la bocina y audifono telefónico en una cuna propia del acoplador.
- La tercera categoría es conocida como de modems de distancia limitada o modems de corto alcance. Los modems en

esta categoría operan similarmente a los de las otras pero no cuentan con el refinamiento del equipo incluido en las categorías anteriores en los métodos utilizados para la modulación-demodulación. De esta manera operan en sistemas de corta distancia donde el nivel de ruido e interferencia es mínimo. Generalmente no están equipados con ecualización automática o adaptable, y funcionan en líneas relativamente cortas que tienen características altas de rendimiento, lo cual hace que los modems de este tipo sean simples y baratos.

Dentro de las características de los modems, los proveedores ofrecen un amplio rango que incluye distintas velocidades de transmisión, transmisión duplex o semiduplex, asíncrona o síncrona, compatibilidad con facilidades de línea, distintas interfases, etc.

Una de las características más importantes es las velocidades de operación, las cuales oscilan, en modems de baja velocidad, de 300 BPS's (bits por segundo) hasta modems de alta velocidad que operan a 1.5 Mega BPS's

Los modems diseñados para operar a velocidades menores de 2000 BPS's están generalmente diseñados para transmisión asíncrona, y aquellos que operan a velocidades mayores de 2000 BPS's son utilizados para transmisión síncrona. Adicionalmente el flujo de información tiene una fase precisa y una relación de fase con un reloj de señal que puede ser proporcionado por el dispositivo al que se conecta o comúnmente por el modem mismo. Dentro de la transmisión, el modem utiliza el reloj de señal para examinar el flujo de señal digital a una instancia deseada, y por otra parte, al

recibir, el modem deriva una señal de reloj de los datos. De esta forma la terminal receptora puede recuperar con exactitud el flujo de datos.

En general, los modems pueden ser utilizados en líneas conmutadas o líneas de red privada. Para líneas de grado de voz privadas los modems de 9600 BPS's pueden ser utilizados para transmisión en líneas punto a punto y multipunto. Sin embargo, debido a que los modems de 9600 BPS's requieren de un complejo proceso de modulación para operar a esta velocidad, es necesario que las facilidades de la línea incluyan un bajo nivel de ruido para así poder el modem reproducir con exactitud el flujo de datos.

En general, las funciones de los modems deben ser ejecutadas tomando en cuenta una serie de restricciones de las facilidades de comunicación. Estas restricciones incluyen niveles de señal, ancho de banda limitado, niveles de ruido y otras características que hacen que deban ser desarrolladas distintas técnicas para poder sobreponer a estas.

Una de las técnicas más utilizadas es la modulación de frecuencia (FM), la cual constituye una de las más complejas para modems asíncronos de mediana velocidad (menor a 1800 BPS's). La fig. 8.3 muestra las partes esenciales de un modem FM. El filtro "Low-Pass" remueve las componentes de alta frecuencia del pulso digital de entrada. Los pulsos controlan la frecuencia del oscilador FM. Un uno binario causa que sea mandada una frecuencia de 'marca', mientras que un cero provoca una frecuencia de 'espacio'. El filtro de paso de transmisión de banda limita el espectro de frecuencia al ancho de banda de la facilidad de transmisión.

En el otro extremo de la facilidad, el filtro de paso de recepción de banda remueve el ruido e interferencia fuera de la banda de interés. El limitador mide y descompone la señal para la entrada al demodulador FM donde la señal original es recuperada en un proceso discriminatorio del demodulador FM. Los filtros "Low-Pass" remueven y desechan las señales de alta frecuencia o ruido y la salida de estos es alimentada al "rebanador", que ejecuta una función de "rebanado" para producir pulsos digitales nuevamente.

Generalmente una línea telefónica de grado de voz tiene un ancho de banda nominal de 3K Hz. Modems diseñados apropiadamente pueden transmitir en estas líneas hasta 9600 BPS's. Sin embargo velocidades mayores no son posibles bajo esta técnica debido a que sólo aproximadamente 2400 Hz de las líneas tienen características adecuadas de transmisión.

De aquí que existen dos maneras para que los modems puedan operar a velocidades superiores:

- Optimizar las técnicas de modulación para que los datos puedan ser transmitidos en un promedio de dos señales por cada HZ de ancho de banda usado.
- Empacar dos o más bits de información por cada señal.

Tres de las técnicas de modulación más simples utilizadas son por amplitud, frecuencia y fase, que se muestran en la figura 8.4.

Otros tipos de características y configuraciones son ofrecidas por los distintos proveedores, dentro de las que se encuentran:

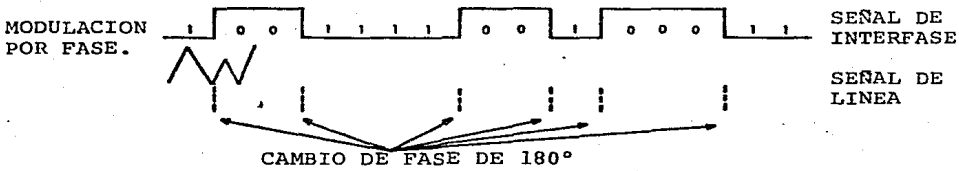
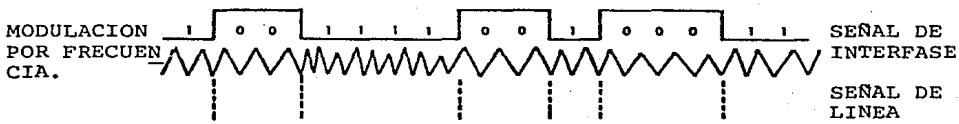
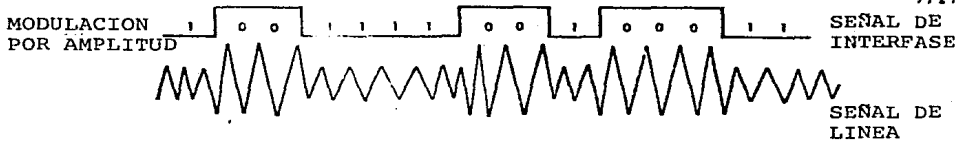


FIG. 8.4
TECNICAS DE MODULACION

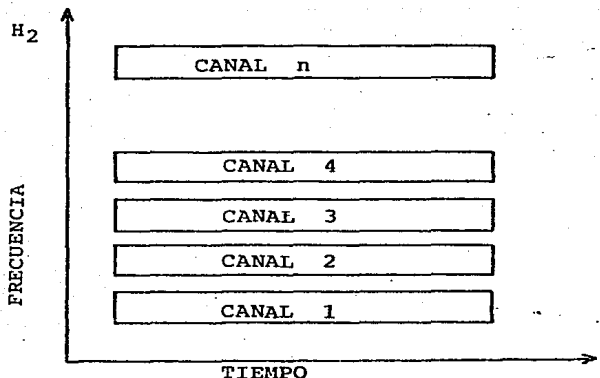


FIG. 8.5
ESQUEMA DE MULTIPLEXION POR DIVISION DE FRECUENCIA

- Operación Multipuerto.- Esta característica permite que un modem que transmite a 9600 BPS's, por ejemplo, pueda operar cuatro flujos de datos a 2400 BPS's por el lado de las terminales, permitiendo así que una línea sea compartida. Esencialmente el modem efectúa una función de multiplicación.

- Control de velocidad.- Que permite que el procesador central controle la velocidad de operación. Esto hace posible que en fallas de línea las velocidades sean disminuidas y también así el promedio de error.

- Ecualizadores de Adapte Automático.- Esta característica permite que modems de alta velocidad puedan operar en líneas no condicionadas o condicionadas mínimamente, lo cual es importante para líneas multipunto o conmutadas.

- Capacidades de prueba.- Permitiendo rastrear el origen de una falla ya sea en el modem, la línea o la terminal.

- Línea conmutada de respaldo.- Operando en línea privada, permite tener una línea conmutada de respaldo, para casos de falla de línea o distancias grandes.

- Respuesta Automática.- Característica que permite al modem responder automáticamente a un llamado telefónico.

MULTIPLEXORES
(MULTIPLICADORES)

Un multiplicador es un dispositivo de comunicación que permite a un número de terminales de baja velocidad compartir una línea de alta velocidad. Su uso es justificado por factores económicos que dependen de la relación de menor costo de los multiplicadores con respecto a las líneas.

Los multiplicadores tienen dos funciones básicas:

- La primera de éstas, llamada precisamente multiplicación, en donde un número de líneas de baja velocidad provenientes de subcanales separados son combinadas en un canal de comunicación de alta velocidad.
- La segunda función se refiere a la capacidad de separar canales de alta velocidad en un número de líneas de baja velocidad.

Existen varias técnicas de multiplicación dentro de las que se encuentran:

- Multiplicación por división de frecuencia (FDM)
- Multiplicación por división de tiempo (TDM)
- Multiplicación por división de tiempos estáticos.

En los multiplicadores de división de frecuencia (FDM) el ancho de banda de la facilidad de comunicación es dividida en subcanales separados (Fig. 8.5).

Cada subcanal FDM activa una trayectoria de comunicación de datos en la parte de baja velocidad del FDM. La parte de alta velocidad integra los datos de los subcanales de alta velocidad, usando una facilidad de comunicación completa para transmitir todos los datos sobre una sola línea telefónica de grado de voz.

La figura 8.6 ilustra la arquitectura de un FDM. En esta figura se tiene una tarjeta de canal por cada subcanal y terminal de baja velocidad, más equipo común para la terminación de las líneas de alta velocidad. La parte de transmisión de la tarjeta de canal establece la interfase con la señal eléctrica de su terminal y la convierte en tonos cambiantes. Al recibir, hace esta conversión aceptando esos tonos cambiantes de frecuencia y transformándolos para constituir la entrada de la terminal.

Por otra parte, los multiplicadores de división de tiempo (TDM), dividen la facilidad de comunicación en subcanales definidos por tiempo (Fig. 8.7). Estos subcanales son llamados de rebanada de tiempo y son utilizados para transmisión de datos desde y hacia terminales de baja velocidad. Un subcanal por cada terminal es conectado al TDM.

La figura 8.8 muestra la arquitectura de un TDM. De igual manera que en el caso anterior se tiene una tarjeta de canal de baja velocidad por cada línea que termina en el

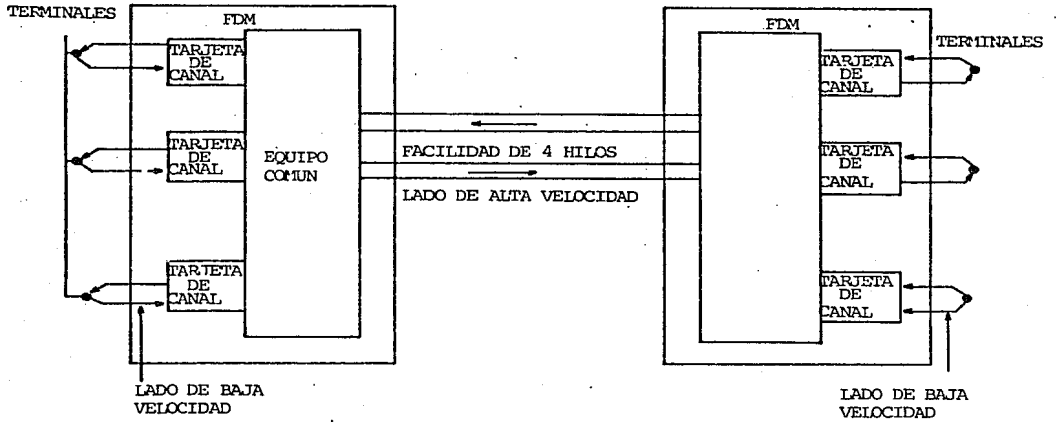


FIG. 8.6
ARQUITECTURA DE UN MODEM POR DIVISION DE FRECUENCIA

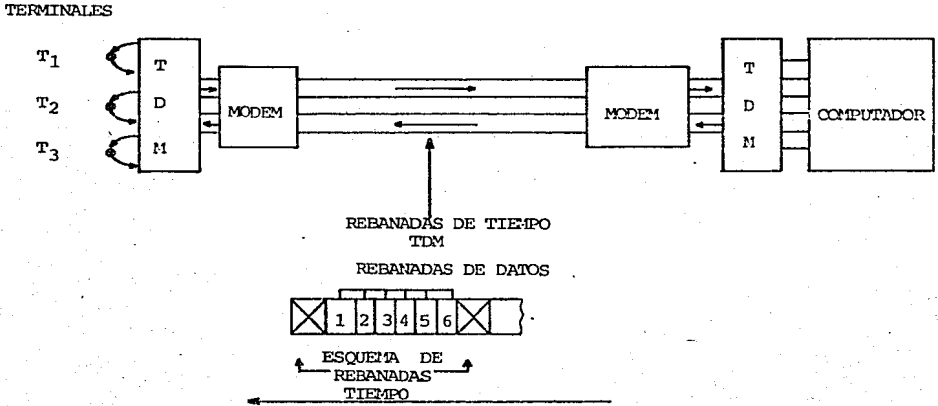


FIG. 8.7
ESQUEMA TDM

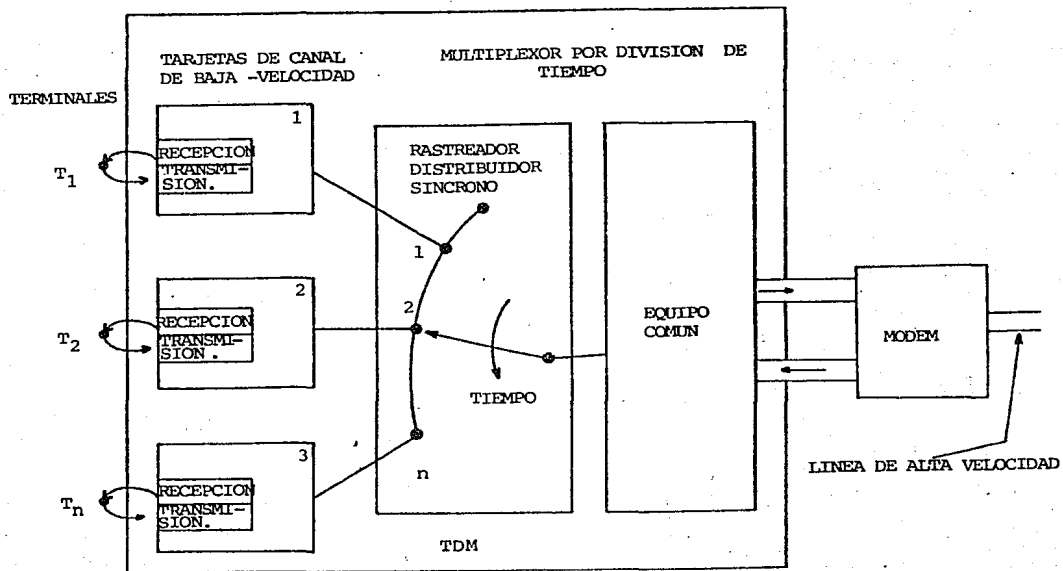


FIG. 8.8
ARQUITECTURA DE UN TDM

TDM. Esta toma los datos y señales de control de las terminales y proporciona capacidades de almacenamiento en registro, brindando buffers de bit o carácter y poniendo o aceptando datos de la rebanada de tiempo correspondiente a la terminal.

En el lado de baja velocidad del TDM, la conexión puede ser hecha directamente a una computadora, terminal, modem u otro multiplicador.

El rastreador/distribuidor es el camino desde el cual los datos son tomados de la tarjeta de canal para la integración en la línea de alta velocidad y por el cual los datos de las rebanadas de alta velocidad son transferidos a la tarjeta de canal. Rastrea y extrae los datos para salida a la línea de alta velocidad y distribuye los datos recibidos de ésta.

El control común es el cerebro del TDM. Proporciona las funciones lógicas para ensamblar y desensamblar las rebanadas de tiempo del TDM bajo condiciones normales o de falla. Así mismo contiene los mecanismos para trabajar con las rebanadas de tiempo y la interfase del modem.

Las señales de datos, control y reloj aparecen también en esta interfase, siendo estas señales de tiempo del reloj, ya sea del modem o de la línea de datos digital, el principal recurso de tiempo del TDM.

La última técnica de multiplexión que se mencionó consiste de los multiplicadores de división de tiempo estático (STDM), los cuales dividen una facilidad de

comunicación en subcanales definidos por tiempo, similarmente a un TDM. Así, el lado de alta velocidad del SDM es muy similar al de un TDM, encontrándose la diferencia del lado de baja velocidad. El TDM brinda una rebanada de tiempo (o subcanal) para cada terminal o línea de baja velocidad conectado a él. El flujo de datos en cambio lo maneja por contención de las líneas de baja velocidad haciendo la asignación de estos datos a líneas de alta velocidad buscando un mínimo de tiempo de espera.

Un STDM incrementa la utilización de la línea tomando más líneas de baja velocidad que puedan ser llevadas al mismo tiempo a la línea de alta velocidad. El número de líneas conectadas a un STDM se obtiene a partir de los tiempos activos, cuando las terminales están transfiriendo datos. En caso de que un número de terminales estén ociosas y que se tenga un gran número de terminales que son candidatas a un TDM, entonces un STDM es más eficiente y resulta ser la mejor opción.

CONCENTRADORES

A diferencia de los multiplicadores que surgen de las nuevas tecnologías desarrolladas, los concentradores son producto del concepto de comunicación de datos y computadoras. La función básica de los concentradores es de compartimiento de línea. Esta función puede ser efectuada por el Hardware, o bien por el Software a través de un emulador, como actualmente es efectuado en equipos de microprocesadores. Los concentradores permiten la contención entre sus terminales de baja velocidad compartiendo un pequeño número de canales de salida en demanda, con un número mayor de canales de entrada de menor velocidad.

Los concentradores difieren de los multiplicadores en los siguientes aspectos:

- La operación de un concentrador no es transparente, esto es, el mensaje de control de línea puede ser eliminado.
- Un mensaje de entrada puede ser reformateado antes de ser transmitido.
- La capacidad de soporte de SOFTWARE programable permite el acomodo de diferentes protocolos de línea, velocidades de datos, formatos control de error, conversión de códigos y detección de código automático.

En comparación a la relativa sofisticación de los concentradores, los multiplicadores son dispositivos Hardware esencialmente inflexibles. Los multiplicadores están basados en conceptos obtenidos de una derivación de canal estático, a través del cual las frecuencias de banda o la porción de tiempo en un canal de comunicación compartido son asignados en una forma predeterminada y fija. Usualmente no existe contención para un canal compartido, por tanto no necesita de consideraciones de encolamiento.

CAPITULO IX

PUNTOS A SER CONSIDERADOS DENTRO DE LA

ARQUITECTURA DE UNA RED

Hoy en día las redes de computadoras varían desde un procesador sencillo que soporte a una o dos terminales hasta complicadas interconexiones entre decenas de unidades de proceso de varios terminales que son interconectadas una a la otra y a su vez a cientos de terminales de distintos tipos y características, por medio de multiplicadores, concentradores o controladores. Esto trae como consecuencia pensar en una arquitectura organizada por niveles.

A continuación se listan las funciones básicas que la red debe proporcionar para obtener una filosofía de arquitectura organizada.

Entre las funciones básicas que debe contemplar cualquier red de computadores esta la de proporcionar trayectorias de acceso por la cuales un usuario final instalado en alguna localidad geográfica pueda tener acceso a otro usuario de otra localidad. Estos dos usuarios finales pueden corresponder a un usuario de terminal y un programa de aplicación remota que este usuario invoca, a dos programas de aplicación interactuando uno con el otro, a un programa de aplicación actualizando un archivo remoto, etc.

Por una trayectoria de acceso se puede pensar en la secuencia de funciones que hace posible a un usuario final no sólo la conexión física con otro, sino también comunicarse con él a pesar de las diferentes características de equipo y comunicación que se puedan presentar tales como diferencias de velocidades, formatos, códigos, interfases etc.

Existe distintas formas de caracterizar una red. Una de ellas es examinando las características de topología del entrelazado de las líneas de transmisión que conectan las distintas estaciones. Otra forma importante es examinando el total de funciones que la red debe proporcionar para brindar trayectorias de acceso efectivas entre los usuarios finales. Dentro de estas funciones se encuentran:

- Estar seguro de que se cuenta con los recursos para la transmisión física (facilidades de comunicación) y que estos recursos contienen las características adecuadas para la transmisión de datos. Además contar con los dispositivos para utilizar estas facilidades para comunicación de datos como son convertidores de señal.
- Garantizar que el flujo de datos recibido corresponde al flujo de datos transmitido chequeando las funciones de protocolos de control de la liga de datos, cuyos métodos de chequeo de errores se explican en el capítulo VII. Estas funciones de protocolo pueden ser llevadas a cabo en enlaces asíncronos o síncronos.
- Checar la naturaleza y cantidad del tráfico de datos que determinan las características del enlace, tipo de línea, etc, permitiendo, por ejemplo que varios usuarios

finales compartan una sola línea. De esta manera si cada usuario final fuese a mandar un flujo de datos a intervalos constantes, una configuración punto a punto para ese usuario será la solución apropiada, mientras que en una configuración con estaciones múltiples atadas a una línea, asignar a cada flujo de datos ciertos campos de dirección de liga y caracteres de control que son utilizados para seleccionar al usuario correspondiente es la solución. Para estaciones raramente utilizadas, conexiones en línea conmutada son la solución, y así sucesivamente, dependiendo de las características de la estación se puede determinar el tipo de enlace.

- Proporcionar buffers para colocar en ellos los datos o mensajes que van llegando hasta que éstos pueden ser atendidos, y de igual manera con los mensajes que son emitidos hasta poder ser enviados por la línea de transmisión. Sin embargo, ciertas limitaciones como son tamaños máximos de buffers, tiempos de respuesta, métodos de chequeo de error, traen como consecuencia la necesidad de segmentar o empaquetar el flujo de datos que se emite, en elementos o bloques de tamaño razonable a las capacidades del equipo y de la línea. Posteriormente, el promedio de flujo de los bloques salientes tiene que ser regular para evitar un sobreflujo en los buffers de la estación receptora, lo cual puede ser logrado enviando caracteres de control de flujo.

- Brindar una manera en la que los usuarios finales utilicen todas las funciones anteriores de tal manera que se establezca un diálogo con los usuarios de otras trayectorias de acceso.

- Y por último el conjunto de funciones que efectúan la activación y desactivación de los componentes de la trayectoria de acceso, las cuales proporcionan los parámetros requeridos para su operación y efectúan el manejo de las recuperaciones en caso de error.

Este conjunto de funciones de activación y desactivación es conocido como control de regla y puede estar centralizado en un nodo o bien descentralizado residiendo en varios nodos, sin existir un nodo dominante. En sí, las funciones de control de red que son requeridas para manejar la trayectoria de acceso pueden ser clasificadas en las siguientes fases:

- Establecimiento de la transmisión eléctrica entre nodos de la trayectoria. Por ejemplo, para líneas conmutadas debe ser marcado el número correspondiente para establecer el enlace.
- Asignar las direcciones de liga de datos de las estaciones, designando quién es estación primaria y quién es secundaria.
- Establecer y actualizar las tablas de enrutamiento que indican cada nodo hacia donde mandan los mensajes.
- Establecer y actualizar los directorios de todos los usuarios en la red y proporcionar la conversión nodo-dirección.
- Proporcionar una interfase para el administrador de la red. Esta debe incluir funciones de determinación de problemas, tales como reporte de errores, pruebas, rastreo de líneas y medición de rendimiento.

La definición de las funciones que una red de computadoras y sus componentes deben ejecutar corresponden a

su arquitectura, en la cual un elemento importante es el protocolo utilizado, que como se mencionaba en un capítulo anterior, consiste de tres elementos esenciales:

- La sintaxis, que es la estructura de los comandos y respuestas en algún campo formateado (bits de encabezado) o cadena de caracteres que forman la semántica.
- La semántica, que es el conjunto de requerimientos a ser usados, acciones a ser ejecutadas y respuestas a ser regresadas por algún dispositivo.
- La sincronización que especifica el orden de los eventos.

Una de las estructuras más importantes que definen un protocolo entre distintos dispositivos que se comunican es SNA (Systems Network Architecture) la cual releva a los usuarios del sistema de comunicación de muchos de los requerimientos de manejo de control y recursos de la red permitiéndoles concentrarse en sus funciones de aplicación. SNA define formalmente las funciones y responsabilidades de los componentes de un sistema de comunicación, pudiendo los distintos dispositivos SNA ejecutar funciones que eran ejecutadas antes por el procesador central y dentro de las que se encuentran:

- El manejo de los canales de comunicación y control de dispositivos.
- Formateo de datos.
- En algunos casos, la ejecución de programas de aplicación.

Intentando de esta manera estandarizar asíf un ambiente diverso que incluye variedad en:

- Hardware (CPU, terminales, controladores etc.)
- Facilidades de comunicación (líneas telefónicas, satélite)
- Software (sistemas operativos, programación de control, microcódigo)

Todo esto conjuntado en un concepto lógicamente consistente.

SNA reconoce básicamente dos tipos de elementos:

- Programas de aplicación
- Usuarios finales.

Estos últimos pueden estar dispersos geográficamente dentro del ambiente y definiendo la relación entre los elementos del sistema de una manera estandarizada permitiendo que cualquier usuario pueda ser dinámicamente enlazado a cualquier función de aplicación y esta a su vez a otra función de aplicación.

Dentro de una estructura SNA se definen niveles de expansión teniéndose en el nivel más alto al procesador central, siguiendo en jerarquía el controlador de la red, posteriormente el controlador de acumulación (cluster) o de terminales y por último, en el nivel más bajo estaciones inteligentes.

En el nivel intermedio la diferencia básica entre un nodo terminal y el controlador inteligente o controlador cluster, es que la terminal es un procesador de entrada/salida diseñado para servir a un sólo usuario a la vez, mientras que el cluster es un procesador de entrada/salida diseñado para servir a múltiples usuarios a

la vez, los cuales pueden compartir recursos como son dispositivos de almacenamiento secundario, dispositivos de Entrada/Salida, canales de Comunicación, procesador, etc..

Existen varios puntos claves a considerarse en el análisis de un cluster:

- Todos los usuarios de procesos de aplicación ejecutados en el cluster deben, por definición, ser procesos interactivos entre los programas del usuario en el cluster y los usuarios de la terminal.
- La instrumentación de productos en el cluster es tal, que esta aplicación puede variar extensivamente y puede ser lógicamente asociada con funciones de aplicación ejecutándose en tiempo real en cualquier otro nivel superior o inferior de la jerarquía SNA.
- Múltiples programas de usuario pueden ser ejecutados concurrentemente en el controlador cluster teniendo cada programa su propia identificación lógica en el sistema SNA.
- Los programas del usuario pueden ser ejecutados sin depender de las otras partes del sistema SNA.
- El controlador cluster efectúa las funciones de control de dispositivos de menor nivel, control de comunicaciones y control de tareas, manteniendo la integridad del sistema SNA.

Por último el controlador de la red corresponde a un procesador de comunicaciones, delegando el manejo de red y control de enrutamiento al controlador de terminales o cluster.

CAPITULO X

CONCEPTOS DE TEORIAS DE REDES Y GRAFICAS.

El enfoque de esta parte es brindar al lector los conceptos básicos de teoría de gráficas a manera de introducirlos en esta área con el fin de estudiar posteriormente los algoritmos y heurísticas para el diseño de la topología de una red de computadoras. Esta topología tiene como finalidad diseñar la red para obtener el máximo rendimiento al menor precio.

El problema que es la base de este capítulo puede ser planteando en los siguientes términos.

"Dadas las localizaciones del procesador central y los distintos nodos terminales y tomando en cuenta las capacidades de las facilidades de comunicación y la asignación de flujos de información, diseñar una topología de red que minimice el costo y a la vez incremente el rendimiento".

Primero se definirán las partes involucradas en la red asignándoles un nombre que los identifique de tal manera que puedan ser asociadas con los términos de teoría de gráficas más adecuadamente.

Los dos dispositivos finales que se encargan de transformar y procesar la información y que corresponden al computador central y a las terminales, para este análisis son dispositivos equivalentes a los que se les llama localidad o estación.

Se definen dos matrices que intervienen en estos procesos:

- Matriz de Tráfico la cual muestra el promedio de flujo de paquetes o bloques de información que son enviados de una estación i a otra estación j , (llamese T a esta matriz). Obviamente al estar diseñando una red, las entradas de esta matriz son desconocidas, de donde se inicializan con un valor estimado que sea proporcional al producto de la estimación de tráfico de estas dos estaciones dividido entre la distancia entre ellas.
- Matriz de costos cuyas entradas corresponden al costo mensual de las líneas o medios de la conexión de la estación i a la estación j .

En general el costo de una línea depende de la distancia y velocidad soportada por el medio. De esta manera deben ser consideradas las líneas que contemplen los dos puntos o características más importantes para optimización del rendimiento y que corresponden a:

- La confiabilidad del medio
- El mínimo tiempo de respuesta que se pueda brindar.

Estos dos puntos pueden ser ajustados al diseñar la red tomando en cuenta la topología misma de ésta, las

capacidades o características de las facilidades de comunicación y la asignación del flujo que corresponden a los algoritmos de enrutamiento de la información.

Considérese como ejemplo una configuración en la cual se tienen n estaciones, lo que daría un total de $n(n-1)/2$ líneas potenciales a ser consideradas, las cuales pueden tenerse o no dependiendo del análisis, capacidades y necesidades, dando un total de $2^{n(n-1)/2}$ topologías potenciales, lo cual para estructuras con 10 o más estaciones se vuelve un número bastante considerable que no puede ser analizado exhaustivamente.

Lo anterior trae como consecuencia la necesidad de desarrollar heurísticas o estrategias que permitan tener un punto de partida más cercano a la solución óptima y métodos que permita desarrollar una aproximación más real al objetivo final.

El primer problema que se encuentra dentro del diseño de redes consiste en que las estaciones no se encuentran distribuidas de una manera uniforme dentro de un área, teniéndose distintas características en cada localidad.

DEFINICIONES

Una gráfica se define como una pareja $G=(N,A)$ donde N y A son dos conjuntos. N corresponde a un conjunto de elementos llamados nodos o vértices que representan los puntos de la gráfica, y A es el conjunto de arcos, ligas o brincos de la forma vu , donde u y v son vértices.

Se dice que dos nodos son adyacentes si existe un arco que directamente los conecte, esto es, dos nodos u y v son adyacentes si existe x en A de tal manera $x=uv$. Este concepto de adyacencia nos lleva a definir dos tipos de gráfica al darle un "sentido" a los arcos que la determinan. Teniéndose así las gráficas direccionadas en las cuales solo es posible ir de un nodo u a un nodo v en el sentido que lo indique la flecha del arco. Por ejemplo, un canal simplex en que la información fluye en un solo sentido. Por otra parte se tienen las gráficas no direccionadas en donde si dos nodos son adyacentes se puede ir en cualquiera de los dos sentidos del arco, como es el caso de dos estaciones conectadas por una línea semi-duplex. De esta manera se le llama gráfica dirigida a una gráfica direccionada, y una gráfica no dirigida a una gráfica no direccionada.

Si un nodo es adyacente a sí mismo, al arco que determina su adyacencia se le conoce como un "loop" o ciclo propio.

Otra forma de representar una gráfica es por medio de una matriz conocida como matriz de adyacencia (M) y cuyo

elemento $M(i,j)$ es distinto de cero si el nodo i es adyacente al nodo j . Se dice distinto de cero ya que existen gráficas en que a cada arco le es asignado un valor que representa ya sea la distancia entre los nodos que une, o bien el costo del arco, o en el caso de una red, la cantidad de información promedio que fluye. Generalmente se tiene que $M(i,j) = 1$ si i y j son adyacentes, si no $M(i,j) = 0$. Obsérvese que si esta gráfica representa una estructura enlazada por líneas semi-duplex o duplex $M(i,j) = M(j,i)$. Si en la gráfica no se presentan ciclos propios, la diagonal de la misma será ceros.

Otro concepto importante es el grado o valencia de un nodo el cual representa el número de arcos que entran (o terminan en el caso de gráficas direccionadas) al nodo. Uno de los principales resultados de este concepto es conocido como teorema de Euler, el cual plantea:

" La suma de los grados de los nodos de una gráfica es igual al doble del número de los arcos de la misma ".

A partir del concepto de grado se define una gráfica regular de grado g como una gráfica en la cual todos sus nodos tienen grado g . De esta definición y del teorema de Euler se tiene el siguiente resultado:

" No existen gráficas regulares de grado impar con un número impar de nodos ".

Una trayectoria es una sucesión de nodos adyacentes, definiéndose la longitud de ésta como el número de arcos que contiene. En general pueden existir varias trayectorias

entre dos nodos. A la mínima trayectoria entre dos nodos se le conoce como una trayectoria geodésica.

El diámetro de una gráfica es la longitud de la geodésica más larga.

Un circuito es una trayectoria que termina en el nodo en el cual empezó, esto es, un circuito es una trayectoria cerrada. Si este circuito contiene a todos los nodos de la gráfica se le conoce como circuito Hamiltoniano.

Una gráfica conectada o de un solo componente es aquella en que para cualquier par de nodos existe al menos una trayectoria que los une.

Un árbol generador de una gráfica es una subgráfica conteniendo todos los nodos de la gráfica y un subconjunto de los arcos de tal manera que para cada par de nodos existe sólo una trayectoria que los une.

Para cada par de nodos se define la longitud del árbol generador de la gráfica y ésta es la suma de las longitudes de los arcos.

Otro concepto importante que es utilizado para obtener la capacidad de acarreo de la información de la red es el concepto de corte. Un corte x - y es un conjunto de arcos que al ser removidos de la gráfica desconectan al nodo x del nodo y . La figura 10.1 muestra de una gráfica en la cual se identifican distintos A-H cortes. En general, en una gráfica pueden existir varios cortes entre dos nodos, cuya cardinalidad puede variar desde uno hasta el número total de arcos de la gráfica .

Se puede observar en la figura que :

C1= [AB,AE]
 C2= [AB,ED,JF,JK]
 C3= [BC,FG,KL]
 C4= [CH,LH]

corresponden a distintos A-H cortes, y al adicionar cualquier arco a estos conjuntos sigue siendo un corte.

Se dice que un corte es mínimo si al recolocar cualquiera de sus elementos en la gráfica ésta se reconecta. Si la gráfica es una gráfica con peso, ésto es, que a cada arco se le asigna un valor, se define la capacidad de corte como la suma de los pesos de cada arco en el corte, y un corte con capacidad mínima es un corte mínimo. Por ejemplo, en la figura anterior, el corte C1 es un corte mínimo.

Otra definición importante que junto con el concepto de corte permite determinar la cantidad del flujo de información (digamos por ejemplo en bits por segundo) en cada línea de una red, es el concepto de flujo factible. Un flujo entre dos nodos x y y , considerando a una como fuente a otra como destino, se dice que es factible si cumple con:

- El nodo fuente no tiene arcos entrantes (la información fluye de él).
- El nodo destino no tiene arcos salientes (no emite datos).
- Ningún arco tiene un flujo mayor a su capacidad.

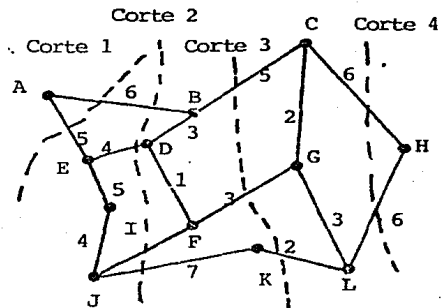


FIG. 10.1
GRAFICA CON PESO QUE CONTIENE CUATRO A-H CORTES

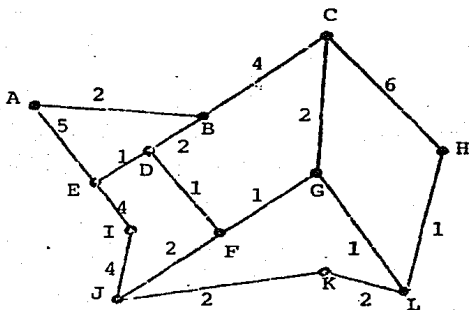


FIG. 10.2
FLUJO FACTIBLE DE LA GRAFICA DE LA FIG. 10.1

- El flujo entrante y el saliente de todo nodo guardan un balance, ésto es, su suma es igual, excepto posiblemente en los nodos fuente y destino.
- El flujo saliente del nodo fuente es igual al flujo entrante del nodo destino.

El flujo corresponde en la gráfica al peso del arco. La fig. 10.2 muestra un flujo factible de la gráfica anterior.

Sin embargo, éste no es necesariamente el flujo factible mínimo, ya que por la condición, se pueden multiplicar todos los flujos por la misma constante.

En sí, el máximo flujo puede ser fácilmente encontrado si todos los cortes son conocidos, debido al siguiente teorema conocido como teorema del "flujo máximo-corte mínimo":

" El flujo máximo entre dos nodos en una gráfica es igual a la capacidad del mínimo corte que separa estos nodos ".

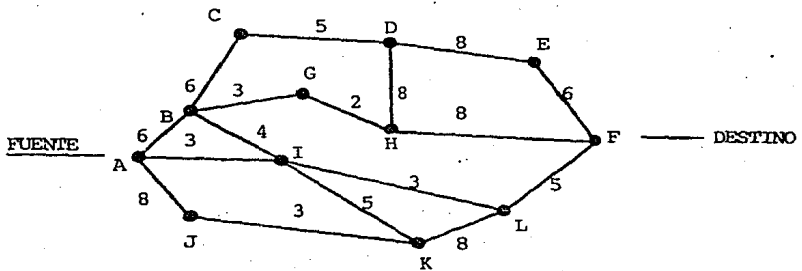
Esto es, siempre existe un flujo factible que satura al corte mínimo.

Sin embargo, independientemente del poder de este teorema, requiere de que se conozca el corte mínimo, por lo cual han sido desarrollados distintos algoritmos para encontrar el flujo máximo. Uno de estos algoritmos es el algoritmo del flujo máximo; del cual será presentado el programa en Pascal que efectúa este algoritmo (sacado de r). Este algoritmo toma una gráfica dirigida con peso y determina el flujo máximo factible entre dos nodos. Para esto se tiene la siguiente definición:

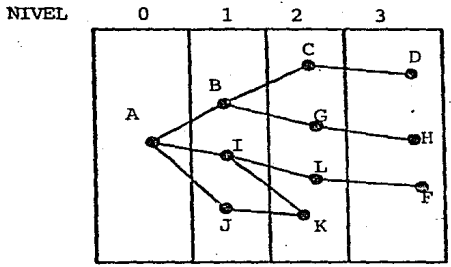
Se dice que un arco es común dirigido al nodo X si un nodo Y tiene un flujo asignado de X a Y menor que la capacidad, y existe algún flujo asignado al arco de Y a X.

Considérese la gráfica de la figura 10.3(a) como base para ejemplificar. En esta gráfica se tienen dos arcos para cada nodo, los cuales son indicados con un arco de doble flecha. Considérese al nodo A como el nodo fuente y al nodo F como el nodo destino. El algoritmo construye una red de niveles del nodo A al nodo F de la siguiente manera:

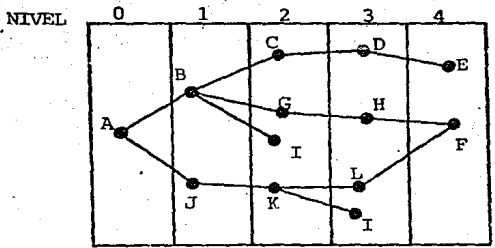
- Colocar al nodo fuente en el primer nivel.
- Cualquier nodo conectado directamente al fuente por un arco común es colocado en el nivel 1, y los conectados al nivel 1 se colocan en el nivel 2 y así sucesivamente hasta colocar al nodo destino en algún nivel. La fig. 10.3(b) muestra la gráfica resultado. Cabe hacer notar que solo arcos que van en el nivel $i-1$ son permitidos, esto es, arcos en niveles anteriores no se permiten en niveles posteriores.
- El siguiente paso de este algoritmo consiste en eliminar aquellos arcos que no terminan en el nodo destino, y que en el ejemplo corresponden a los arcos ABDC, ADGH, IK, IAJK.
- Posteriormente se debe checar cada nodo que queda del paso anterior de tal manera de determinar el incremento del flujo potencial que pueda tomar, el cual para cada arco de un nodo X a un nodo Y es la capacidad de XY menos el flujo asignado a XY más el flujo asignado a YX. El potencial total de salida es la suma de los potenciales de los arcos que se comunican con el nivel más alto próximo. De igual manera se obtiene el potencial mínimo. Dentro de la gráfica de niveles, al nodo que tiene el potencial más pequeño se le



(a)



(b)



(c)

FIG. 10.3
EJEMPLO DEL ALGORITMO DEL FLUJO MAXIMO

conoce como nodo de referencias, y a su potencial como potencial de referencia. En el ejemplo los nodos I y L son nodos de referencia con potencial de referencia 3.

- Se procede a partir del nodo de referencia y colocar un flujo igual al del potencial de referencia siguiendo hacia adelante hacia el nodo destino, utilizando más de un arco si es necesario, y repitiendo este procedimiento recursivamente ya que al poner el flujo en un nodo se sigue hacia adelante con el siguiente hasta llegar al destino que es el único nodo que puede aceptar un extra-flujo. De igual manera se puede empezar en el nodo de referencia e ir eliminando flujo del lado de entrada. Estos dos procesos siempre son posibles ya que cualquier otro nodo (que no sea el de referencia) en la red de niveles tiene cuando mucho este potencial, pero en entrada y salida.

- El proceso continúa, aplicando el mismo procedimiento, pero ahora comenzando con una nueva gráfica de niveles fig. 10.3(c) con otras trayectorias comunes, hasta no poder construir otra gráfica de nivel x , esto es, hasta no haber más trayectorias comunes del origen al destino. Al terminar se tiene que el flujo asignado es el flujo máximo y los arcos que terminan saturados constituyen un corte.

El programa correspondiente a este algoritmo se encuentra en el Apéndice de esta tesis.

TRAYECTORIAS DISJUNTAS.

Es importante dentro de una red de transmisión de datos poder perder algunas líneas sin que con esto se desconecten otros dispositivos. Para esto considérense las siguientes definiciones:

- La conectividad entre dos nodos X y Y es el mínimo de arcos que pueden ser removidos de la gráfica para desconectar X y Y. Esto es, la conectividad es la cardinalidad del corte mínimo.
- Dos trayectorias se dice que son disjuntas si no tienen arcos en común, pudiendo tener nodos en común pero no arcos. Este concepto es importante ya que se tiene el siguiente resultado:

* Si existen n trayectorias disjuntas entre dos nodos X y Y, entonces existe un xy-corte que contiene al menos n arcos*.

Una manera de calcular el número de trayectorias disjuntas entre un par de nodos es utilizando el teorema del flujo máximo y mínimo corte. Primero hay que reemplazar los pesos de los arcos con un peso común de uno y calculando el flujo máximo entre estos dos nodos utilizando uno como fuente y otro como destino. Así, el flujo resultante, que es igual al número de arcos en el corte mínimo, es igual al número de trayectorias disjuntas entre los nodos.

Otro concepto es la conectividad de una gráfica la cual se define como el mínimo de las conectividades de cualquier par de nodos. Esto es, si el mínimo número de conectividad de los nodos es K entonces existen K trayectorias disjuntas en esta, de donde la gráfica es K -conectada.

Para determinar la conectividad de los nodos de una gráfica nuevamente se utiliza el algoritmo del máximo flujo de la siguiente manera:

- Si existe un arco entre dos nodos dados, la gráfica puede soportar la pérdida de cualquier número de nodos intermedios, en cuyo caso si la gráfica tiene n nodos, la conectividad del nodo es solo $n-1$.

- Si no es así, se toma la gráfica original (no dirigida) con sus n nodos y a arcos y se convierte en una gráfica dirigida con $2n$ nodos y $2a-n$ arcos. Cada nodo X es reemplazado por los nodos X y $X1$ con un arco dirigido de X a $X1$ pero sin existir arcos de X a $X1$. Si un nodo Y se encuentra conectado al nodo X en la gráfica original se ponen dos arcos YX y $X1Y$ y en la gráfica modificada. Así, todo arco que llega es ligado a X y los que salen son ligados a Y .

- A los nuevos arcos introducidos les es dado un peso de uno y los arcos originales son removidos (fig 10.4).

- Por último solo resta encontrar el número de trayectorias disjuntas en nodos entre el nodo fuente y el destino aplicando a esta gráfica el algoritmo de flujo máximo. Así, el flujo máximo encontrado es igual al número de trayectorias disjuntas en nodos, ya que cada arco con peso uno puede tener solo una trayectoria pasando por él. Por tanto, el nodo original no puede estar en dos o más trayectorias.

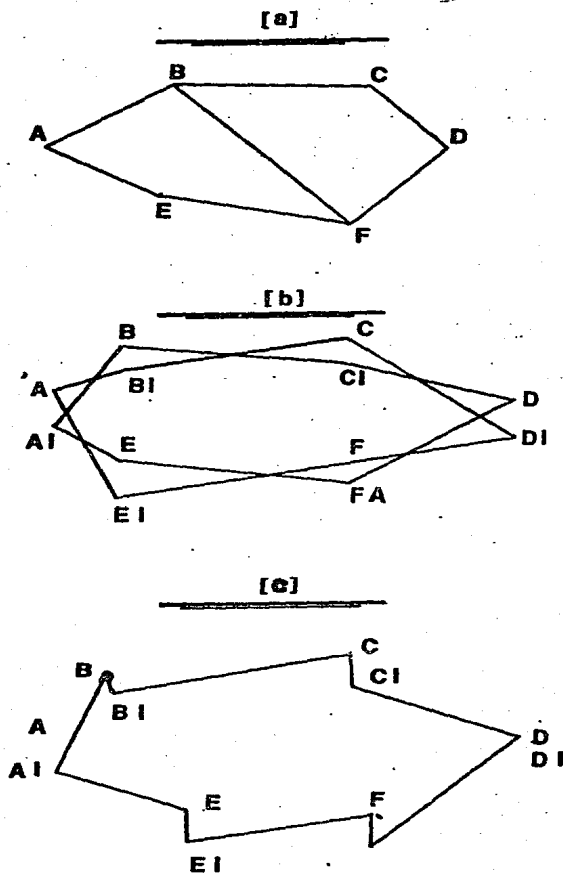


FIG. 10.4
EJEMPLO DE CONECTIVIDAD POR FLUJO MAXIMO

CAPITULO XI

DISERNO DE REDES CENTRALIZADAS

CONEXIONES MULTIPUNTO

Dentro del diseño de redes, algunas de las principales preguntas que se presentan son las siguientes:

Dada una multiplicidad de terminales y un conjunto de localidades de las fuentes de información sobre un área geográfica, con una aproximación del tráfico esperado entre las distintas fuentes, ¿ cómo decidir el número de concentradores necesarios, la posición de éstos, y como deben ser interconectados ?. Posteriormente surge la pregunta de que terminales deben ser conectados a determinado concentrador.

Las respuestas a estas preguntas y todas aquellas que giran alrededor del diseño de redes no son inmediatos y en algunos casos son necesarias heurísticas para resolver o dar una aproximación a la solución.

Uno de los tipos más sencillos de redes es el de redes en sistemas centralizados en el que el flujo de mensajes es hacia adentro de alguna facilidad de proceso central. El modelo de red centralizada que se presentará se aplica a dos problemas importantes:

- El problema antes mencionado, de la organización de las terminales, en el cual las terminales deben ser conectadas en una estructura multipunto a un concentrador específico dependiendo de las características y necesidades de la red.
- El problema de una red centralizada en la cual los concentradores por sí mismos son conectados a una facilidad central de proceso.

En ambos casos un punto importante es la conexión con un costo mínimo del conjunto de enlaces en la red considerándose ciertas restricciones de interés. Una restricción común es el establecimiento de un tiempo de respuesta máximo. Otra puede ser la disponibilidad, la cual, puede ser interpretada de varias maneras: una de éstas es que se cuente con una trayectoria alterna en una ruta o bien, que no más de un número específico de terminales o concentradores sean desconectados si algún enlace de la red falla.

Un algoritmo que proporciona la solución de costo mínimo a este problema de restricciones en el diseño de este tipo de redes, es el algoritmo desarrollado por Chandy y Russell el cual requiere tiempos de proceso de computadora muy altos, particularmente para redes muy grandes. La idea principal para este algoritmo es particionar el conjunto de posibles soluciones en subconjuntos cada vez más pequeños,

siempre chequeando si la cota mínima, basada en una conexión del mínimo árbol generador en ese subconjunto, es alcanzada. Cuando esta cota es alcanzada el algoritmo termina.

Sin embargo existen distintas heurísticas desarrolladas que proporcionan una solución subóptima no muy lejana de la solución óptima y que requieren de un menor tiempo de procesamiento. Entre estas heurísticas se encuentran el algoritmo de Esau-Williams y el algoritmo de Prim.

ALGORITMO DE ESAU-WILLIAMS.

Este algoritmo se encuentra basado en la misma idea de producir un árbol generador mínimo en el cual las restricciones son removidas. De esta manera proporciona generalmente diseños de redes de este tipo que se encuentran más cercanos a la solución óptima.

El algoritmo esencialmente busca aquellos nodos que se encuentran más cercanos al nodo central (en sentido de costo) y los conecta a los nodos vecinos que proporcionen mayor beneficio de costo.

Para esto se define la matriz de costo, que corresponde a una matriz simétrica cuyas entradas se asume que se conocen y representan el costo del establecimiento de las ligas en cualquier par de nodos.

Por ejemplo, supongamos que tenemos 5 nodos y que la matriz de costo es la siguiente:

Nodo/Nodo	1	2	3	4	5
1	-	3	3	5	10
2	3	-	6	4	8
3	3	6	-	3	5
4	5	4	3	-	7
5	10	8	5	7	-

Supongase además que el tráfico generado por unidad de tiempo en cada uno de los nodos fuente (considerando al nodo 1 como la facilidad central) es el siguiente:

$$A_2 = 2, \quad A_3 = 3, \quad A_4 = 2, \quad A_5 = 1$$

Así tendríamos la red inicial ilustrada en la figura 11.1.

El algoritmo Esau-Williams consta de cuatro pasos que son los siguientes y que se seguirán con los datos del ejemplo planteado:

PASO 1.- Obtener los parámetros de intercambio $T(i,j)$ que contienen la diferencia del costo de la conexión del nodo i al nodo j , y del nodo i directamente a la facilidad central:

$$T(i,j) = C(i,j) - C(i,1)$$

Donde $C(i,j)$ es la entrada de la matriz de costo C .

EJEMPLO:

$$\begin{aligned} T(2,4) &= C(2,4) - C(2,1) = 4-3=1 \\ T(4,2) &= C(4,2) - C(4,1) = 4-5= -1 \\ T(5,3) &= C(5,3) - C(5,1) = 5-10= -5 \\ T(3,5) &= C(3,5) - C(3,1) = 5-3=2 \\ T(2,1) &= T(3,1) = T(4,1) = T(5,1)=0 \end{aligned}$$

PASO 2.- Seleccionar el mínimo valor de $T(i,j)$ y considerar la conexión del nodo i al nodo j . En el ejemplo $T(5,3) = -5$ es el mínimo, considerándose así la conexión del nodo 5 al 3.

PASO 3.- Checar si las restricciones se satisfacen. Si es así, efectuar el paso 4 y si no hacer $T(i,j) = \text{infinito}$, y regresar al paso 2 haciendo de nuevo la selección con los valores restantes.

En el ejemplo supongamos que el flujo máximo es 5 y la liga 5-3 tiene $A_5=1$, el cual es menor que el máximo, por lo que la restricción se satisface.

PASO 4.- Adicionar la liga $i-j$ con la etiqueta $i \leftarrow j$ para mostrar que el nodo i está conectado al nodo j , reevaluar las restricciones y actualizar las restricciones de intercambio. Regresar al paso 1.

Con la liga 5-3 se conectan en el ejemplo de la fig. 11.1 $5 \leftarrow 3$, el flujo A_3 es ahora $A_3 = A_3 - A_5 = 4$ y el nodo 5 puede ser conectado a la facilidad central de la liga más larga. Así las funciones de intercambio $T(5,3)$ y $T(3,5)$ pueden ser eliminadas de la lista.

Para concluir con el ejemplo la liga a ser considerada es $T(4,3)=-2$. Si el nodo 4 es conectado con el nodo 3 el nuevo flujo de 3 es $A_4-A_3=6 > 5$. Esto no es válido así que con que $T(4,2)=-1$, la restricción en este caso se satisface $A_2' = A_2 - A_4 = 4 < 5$, y la conexión puede ser hecha.

Repitiendo así sucesivamente se obtiene finalmente la red de la figura 11.2 donde el número entre paréntesis indica el orden en que las conexiones fueron hechas.

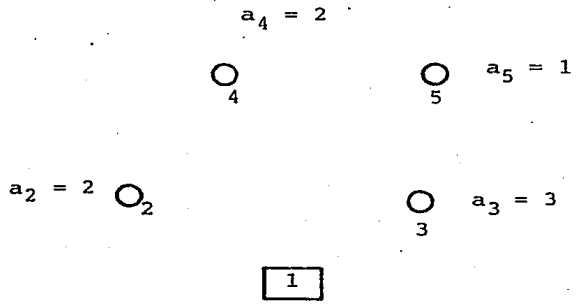


FIG. 11.1
GRAFICA PARA EJEMPLO EN REDES CENTRALIZADAS

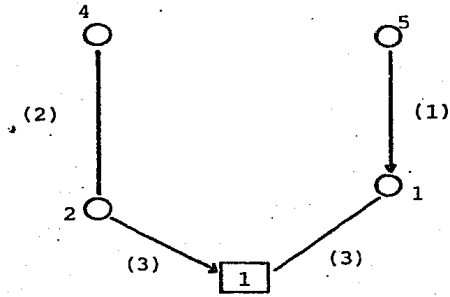


FIG. 11.2
PASOS EN EL ALGORITMO DE ESAU-WILLIAMS

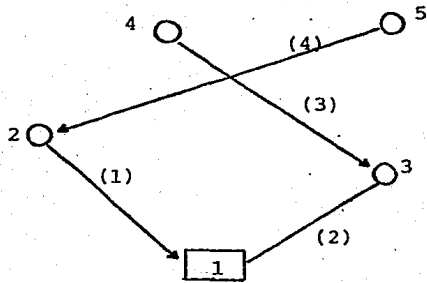


FIG. 11.3
RESULTADO AL APLICAR EL ALGORITMO DE PRIM

ALGORITMO DE PRIM

Este algoritmo, a la inversa que el algoritmo de Esau-Williams, inicialmente selecciona los nodos más cercanos al central, (de nuevo en un sentido de costo) y los conecta a aquéllos que están más cercanos a los que ya se encuentran en la red. Los pasos a seguir dentro de este algoritmo son los siguientes y se ejemplificarán tomando la misma configuración que el algoritmo anterior.

PASO 1.- A cada nodo le es asignado un factor de peso $W(i)$, con $W(1) = 0$ y $W(i) = \langle \text{menos infinito} \rangle$ para i distinta de 1. La función de intercambio $T(i,j)$ se define como $T(i,j) = C(i,j)$ de donde inicialmente $T(i,j) = \langle \text{infinito} \rangle$, excepto para $T(1,j)$, el cual representa el costo de conectar cada nodo al centro o facilidad central. De esta manera el único nodo inicialmente conectado a la red es el nodo central.

PASO 2.- Encontrar el mínimo valor de $T(i,j)$, de esta manera se busca el costo mínimo de la conexión de un nodo a otro ya conectado en la red.

En el ejemplo $T(1,2) = C(1,2) = 3$, $T(1,3) = 3$, $T(1,4) = 5$, $T(1,5) = 10$. Así, los nodos 2 ó 3 pueden ser conectados al nodo 1. Tomese, por ejemplo, el nodo 2.

PASO 3.- Checar si la restricci3n ha sido violada, si no es as3 efectuar el paso 4. En caso contrario, asignar al m3nimo $T(i,j) = \langle \text{infinito} \rangle$ y regresar al paso 2. En el ejemplo $A2 = 2 \times 5$.

PASO 4.- Adicionar la liga $i-j$, estableciendo $W(j) = 0$, reajustando la restricci3n y recalculando de nuevo todos los $T(i,j)$, y regresar al paso 1.

En el ejemplo, repitiendo el algoritmo se tiene que 3-1 es el siguiente a colocar, posteriormente la liga 4-3 conectando el nodo 4 al nodo 3 que ya exist3a en la red. Finalmente 5-2, obteniendo la red de la figura 11.3 en donde se indica entre par3ntesis el orden en que se colocaron.

CAPITULO XII

DISEÑO DE REDES GENERALIZADAS.

En este capítulo serán tratados dos problemas intermedios dentro del diseño de redes y heurísticas de bosquejo para su solución. Estos dos problemas son:

- 1.- ¿Dónde colocar los concentradores, cuántos de éstos deben ser utilizados y qué terminales asociar a cada uno en una configuración de red centralizada?
- 2.- Obtener el costo mínimo de interconexión de concentradores en una red distribuida en la cual, a diferencia de las redes centralizadas en las que el flujo de la información es hacia una facilidad central, el flujo se tiene hacia, probablemente, todas las estaciones.

Las dos técnicas que a continuación se desglosan conjuntamente con el algoritmo anterior pueden ser utilizadas para proporcionar un diseño viable para una red total. El primer paso de ambos algoritmos está enfocado hacia redes centralizadas.

De esta manera el problema puede ser planteado de la siguiente forma:

* Un número n de terminales debe ser conectado, vía probablemente concentradores, a una facilidad central.

Supóngase que este conjunto está determinado por $T(1), T(2) \dots T(n)$. Se cuenta además con un conjunto de localizaciones de concentradores $[S(0) \dots S(M)]$ del cual debe ser extraído un subconjunto con terminales asignadas a cada uno conectando a $S(0)$, teniéndose la posibilidad de tomar una localización de terminal como localización de concentrador si así se desea. Cada concentrador $S(j)$ tiene un costo fijo $f(j)$ consistiendo del costo del Hardware y el costo de la conexión de $S(j)$ a $S(0)$. Llámese "e" a la capacidad máxima de puertos terminales de un concentrador. Así $f(0) = 0$ y $e(j) > n$. Sea $C(i, j)$ el costo de conectar la terminal i al concentrador j , con $i=1, 2, \dots, n$ y $j=1, 2, \dots, m$. El problema consiste en encontrar una función de costo total mínimo Z a través de la definición de los parámetros de conexión binaria.

Sean $X(i, j) = 1$ si $T(i)$ está conectada y $S(j) = X(i, j) = 0$ en caso contrario.

Se dice que un concentrador está abierto si está en uso y cerrado si no. Sea $Y(j) = 1$ si $S(j)$ está abierto o bien si la suma de $X(i, j)$ es mayor que cero con $i=1, 2, \dots, n$, y $Y(j) = 0$ en caso contrario.

El costo total Z a ser minimizado en X es:

$$Z = \sum (Y(i)f(j)) - \sum (X(\sum(x(i, j)), C(i, j))) \quad i=1 \dots n \quad j=1 \dots m.$$

El primer término representa el costo del concentrador y el segundo el costo de la liga de la terminal al concentrador.

La minimización a ser obtenida está sujeta a dos restricciones:

$$1.- \text{Suma } (X(i,j))=1 \quad (i = 1,2,\dots,n).$$

La cual indica que la terminal i debe ser conectada a algún concentrador inclusive $S(0)$.

$$2.- \text{Sum } (X(i,j)) \leq e \quad (j = 1,2,\dots,m)$$

Que indica que no más de e terminales pueden ser conectadas al concentrador j .

El problema es encontrar el conjunto de $Y(j)$ y $X(i,j)$ que minimicen el valor de Z sujetos a las dos restricciones, para lo cual se presentan a continuación dos heurísticas que serán explicadas por medio de un ejemplo en los siguientes términos:

Se tienen seis terminales $T(1), \dots, T(6)$ conectadas a tres concentradores $S(1), S(2), S(3)$, o bien al nodo central $S(0)$ como se muestra en la figura 12.1 en la cual el concentrador $S(1)$ y la terminal $T(2)$ se encuentran en el mismo punto, al igual que $S(3)$ y $T(5)$.

Supóngase que no más de tres terminales pueden ser conectadas a un concentrador, esto es $e=3$, y que el costo de los concentradores es el mismo. Así $f(1)=f(2)=f(3)=2$, mientras que el costo de la liga terminal-concentrador está dado por la matriz C siguiente:

Ti / Sj ---->

			0	1	2	3
.	1	.	2	1	2	4
.	2	.	1	0	1	2
.	3	.	4	1	2	2
.	4	.	1	2	1	2
.	5	.	2	3	2	0
.	6	.	4	4	3	2

ALGORITMO DE LA ADICION

Dentro de este primer algoritmo la inicialización consiste en conectar todas las terminales a $S(0)$; todos los concentradores son cerrados siendo abierto sólo uno a la vez en forma iterativa, de tal manera de proporcionar una disminución en el costo. Así:

$$Z = \sum(c(i,0)) = 14 \quad \text{para } i=1, \dots, 6$$

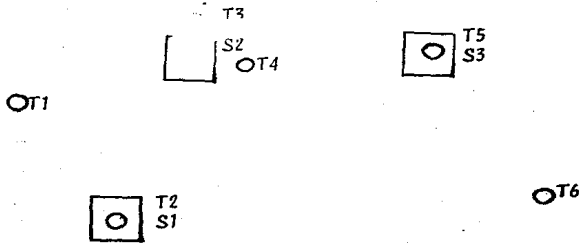
teniéndose como red inicial la ilustrada en la figura 12.2.

ITERACION I

1.- Cada concentrador es checado, uno a la vez, para determinar cual proporciona el máximo mejoramiento sobre el paso de inicialización.

Así se abre $S(1)$ como primero y se conecta las terminales con el mayor decremento en el costo

2.- $C(i,1) - C(i,0) = 3$, para un máximo de $e=3$. En este caso las terminales $T(1)$, $T(2)$ y $T(3)$ conectadas a $S(1)$ resultan proporcionales al mejoramiento en el paso de inicialización. Para estas tres terminales conectadas a $S(1)$ y las otras permaneceran conectadas a $S(0)$ se tiene:



S0 (CPU)
 FIG. 12.1
 EJEMPLO DE RED
 CENTRALIZADA

○ REPRESENTA TERMINAL
 □ REPRESENTA
 CONCENTRADOR

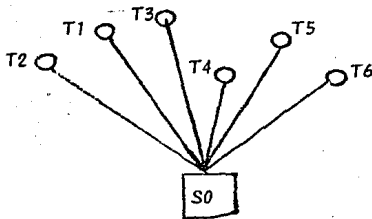


FIG. 12.2
 INICIALIZACION EN EL ALGORITMO
 DE LA ADICION

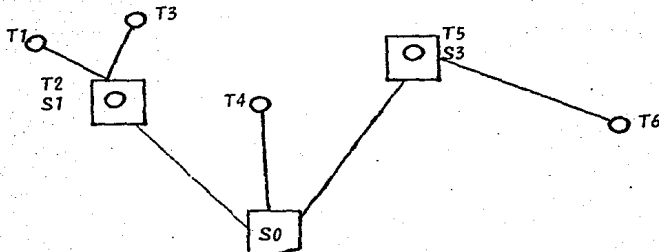


FIG. 12.3
 RESULTADO FINAL DEL ALGORITMO
 DE LA ADICION

$$Z(1) = \sum(c(i,0) f_i - \sum(c(j,i) f_j)) = 11 \text{ para } i=1, \dots, 6, j=1, \dots, 3$$

3.- Ahora trátase de abrir S(2). Aquí las terminales T(3) Y T(6) pueden ser conectadas (ya que otra terminal más conectada a S(2) mejoraría sobre la conexión con S(0) directamente). En este caso se tiene:

$$Z(2) = 13$$

Finalmente con S(3) abierto, se conectan las terminales T(3), T(5) y T(6) que mejoran el resultado obtenido en la inicialización, ya que:

$$Z(3) = 10$$

Con esto todos los concentradores han sido chequeados, siendo Z(3) el costo mínimo, permaneciendo S(3) abierto y la iteración termina obteniéndose el resultado de la figura 12.3.

ITERACION 2

Al permanecer S(3) abierto, S(1) y S(2) son chequeados para ver si abriendo cada uno de éstos se obtiene una mejora. Todas las terminales son tomadas en cuenta incluyendo aquéllas previamente conectadas a S(3). Abriendo S(1) primero con las terminales T(1), T(2) y T(3) conectadas a él, se ganan tres unidades por el corte y se pierden dos por la conexión contra uno. Repitiendo para S(2) se tiene una posible mejora y así sucesivamente se llega a que:

Z=9

Teniendo T(1), T(2) y T(3) conectados a S(1).

CAPITULO XIII

DISEÑO DE REDES DISTRIBUIDAS

ALGORITMO DE SATURACION DE CORTE

Dentro de las restricciones que se encuentran dentro del diseño de una red se tiene un promedio máximo de tiempos de respuesta dentro de la transmisión y algunos criterios de disponibilidad específicos, los cuales son planteados en términos de la conectividad de los nodos, donde un nodo tiene conectividad N si está conectado al menos a otros N nodos. Otro punto importante a considerar es la capacidad de las ligas debido a la relación no lineal que generalmente se presenta entre el costo y la capacidad.

El algoritmo conocido como de Saturación de Corte asume que las capacidades de las ligas son conocidas y que son las mismas para todas éstas en la red, a partir de lo cual encuentra iterativamente la red distribuida de costo mínimo sujeta a las restricciones de tiempos de respuesta y de disponibilidad de la red, y asumiéndose que el tráfico en bits por segundo generado en promedio entre cualquier par de nodos en la red es el mismo para todo par.

Con respecto al tiempo de respuesta, es considerada una subrutina de enrutamiento que encuentra la ruta óptima para mensajes entre todos los pares fuente-destino, después de cada iteración de diseño.

El algoritmo de saturación de corte consiste de los siguientes cinco pasos básicos en cualquier iteración.

1.- Enrutamiento. Dado un diseño de red se encuentra la liga de flujo óptima que minimice los tiempos promedios de respuesta.

2.- Determinación de Cortes Saturados. Una vez determinado el flujo óptimo las ligas son ordenadas y removidas de acuerdo a su utilización. El mínimo conjunto que desconecte la red es llamado un corte saturado.

3.- Paso de adición única. En el cual se adicionan a la red las ligas de costo mínimo que a partir del corte saturado conecten efectivamente los nodos de cualquier lado y relativamente lejanos que fueron removidos a partir del corte y que deben ser ligados. Así se utiliza un criterio de "distancia Z" donde nodos que se encuentran al menos a dos ligas removidas de los nodos del corte son candidatas para posible liga.

4.- Operación de eliminación única. En este paso son eliminadas las ligas de una topología superior conectada. Una liga es removida a la vez en cada iteración. Esta liga elegida corresponde a la más expansiva y menos utilizada de acuerdo al siguiente criterio:

Minimizar

$$E(i) = D(i) \langle C(i) - f(i) \rangle$$

$$\text{-----}$$

$$C(i)$$

Donde $D(i)$ es el costo de la liga, $C(i)$ es la capacidad y $f(i)$ es el flujo.

5.- Paso de perturbación.- Una vez que es alcanzado un rango de flujo adecuado, las ligas de la red son reordenadas utilizando operaciones de adición y eliminación únicas de tal manera de reducir el costo teniéndose el diagrama de flujo de la figura 13.1 para este paso.

En el paso "ES UNA RED DOMINANTE", se checa si el método de perturbación resulta una solución pobre y se tiene una lista de costo y flujo consistente de pares (D_i, R_i) de la solución posible de la red previa. Si una nueva configuración es encontrada con el método de perturbación y tiene pares (D, R) en que el costo es mayor y el flujo menor, entonces la liga que fue eliminada es adicionada de nuevo y se continúa con la iteración.

Un paso adicional en el algoritmo de saturación de corte involucra cadenas de "colapso", las cuales son porciones de la red con una secuencia de nodos en serie. Una red que satisface el criterio de distancia 2 antes mencionado puede contener distintas cadenas. Aquellas cadenas que acarrean tráfico directamente dirigido entre nodos externos a la cadena pueden ser "colapsados" o

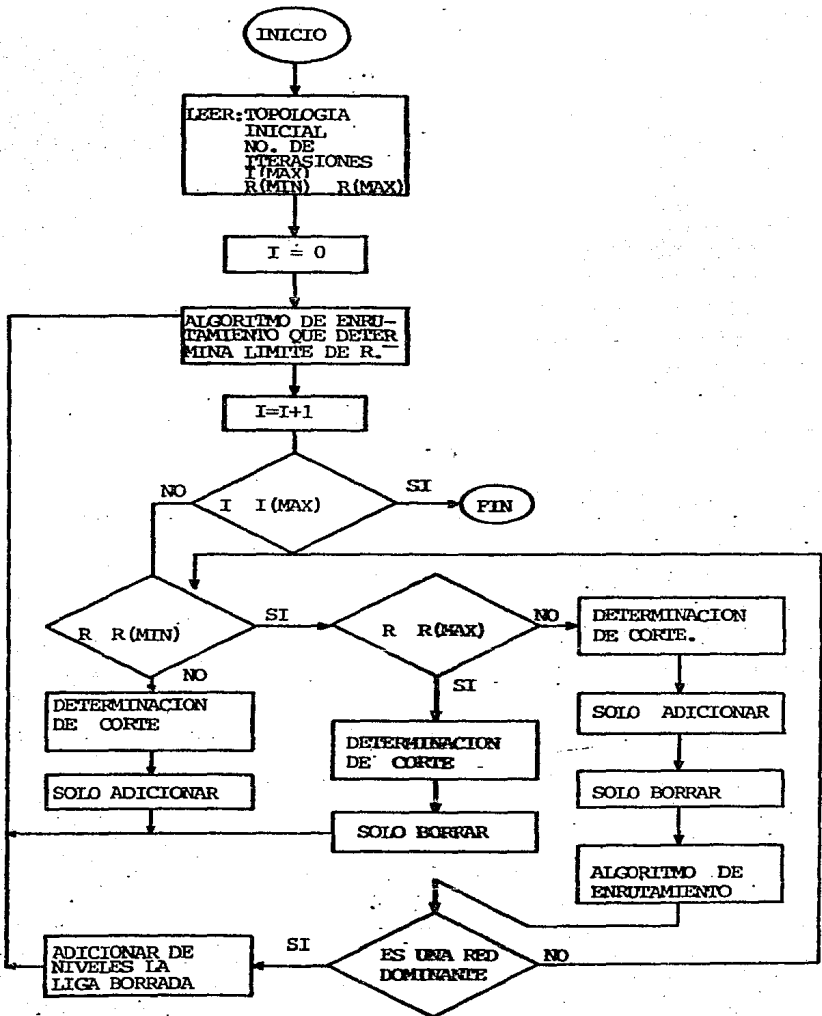


FIG. 13.1
ALGORITMO DE PERTURBACION

reemplazados por una simple liga de tal manera de aumentar la eficiencia del algoritmo.

ALGORITMO DE TRAYECTORIAS CORTAS

Un problema importante dentro del diseño de redes es la determinación de una ruta óptima (bajo ciertas restricciones de costo y distancia de una liga) entre un par de nodos fuente y destino, no teniéndose restricciones, de tal manera de poder checar cada par de nodos por separado.

El siguiente algoritmo tiene dos partes básicas. Dentro de la primera se establece la distancia más corta (en términos métricos) entre un nodo fuente y los nodos restantes de la red. Cada nodo es etiquetado con esa distancia así como el nodo siguiente al frente.

ALGORITMO ESPECIFICO.

1.- Inicialización.

Asignar a todos los nodos etiquetas de la forma $(- , d(i))$ ($-$ representa inicialmente blanco). Se considera un nodo $N(a)$ como nodo fuente con $d(a) = 0$ y $d(i) = \langle \text{infinito} \rangle$ si $i \neq a$.

Adicionalmente se cuenta con la matriz de longitudes L donde $L(i,j)$ representa la longitud o distancia del nodo $N(i)$ a $N(j)$ quedando no definida o con valor cero si los nodos no están conectados.

2.-Etiquetar todos los nodos con la distancia mínima.

Encontrar un par i,j de tal manera que:

$$d(i) - L(i,j) < d(j)$$

si existe este par, cambiar la etiqueta del nodo $N(j)$ a $(N(i), d(i) - L(i,j))$ repitiendo este paso hasta no encontrar un par de nodos que cumplan la condición, después de lo cual, todos los nodos quedan etiquetados con la distancia más corta del nodo fuente $N(a)$ y con el nodo próximo vecino dentro de la trayectoria mínima.

3.- Determinación de la trayectoria más corta.

Para identificar la trayectoria más corta del nodo fuente $N(a)$ a un Nodo destino $N(b)$.

- a) Asignar $i=b$
- b) Identificar $N(k)$ de la etiqueta $(N(k), d(i))$ asignada al nodo $N(i)$. Si $N(k)$ no existe entonces una trayectoria de $N(a)$ a $N(i)$ en la red.
- c) Asignar $i=k$, Si $i=a$, terminar; si no regresar al paso b.

Esta parte del algoritmo trabaja esencialmente hacia atrás a partir del nodo destino $N(b)$, determinando la trayectoria más corta a $N(a)$.

.APENDICE

A.1 Programa Emulador de Terminal VT100 para
Equipos Burroughs B-25.

(*)

Emulador de Terminal Asíncrona VT100
Para Conexión de Equipos Burroughs B-25
Con Equipos Digital Series PDP y VAX

Elaborado por : José Mauricio Torres Romero

*)

(*)

INTRODUCCION:

El Equipo B-25 cuenta con 2 puertos de comunicación
Asíncrona conocidos como [Comm]A y [Comm]B, con una
interfase RS-232C los cuales pueden ser utilizados para
comunicación asíncrona con otras máquinas o dispositivos

para distintos usos como pueden ser: comunicación para emular una terminal asíncrona (como lo es este caso) o bien para conectar dispositivos tales como impresoras en serie.

Estos puertos pueden ser accedidos desde un programa desarrollado en los lenguajes que soporta el equipo B-25 únicamente llamando a las rutinas del Sistema Operativo (BTOS V.4.0) que acceden a la facilidad de este sistema conocida como "Byte Stream", el cual es una secuencia de bytes de 8 bits de la cual es posible leer o en la cual es posible escribir, y de esta manera dispositivos reales como son el video, teclado, impresoras, puertos de comunicación y demás dispositivos reales, emulan un dispositivo conceptual, secuencial orientado a carácter, el cual puede ser leído de forma secuencial.

La filosofía del llamado de las funciones del sistema que son declaradas como funciones EXTERN, consiste en que el valor regresado por la función corresponde al código de error obtenido como resultado de haberse efectuado la función, en el caso de no existir error un valor de cero es regresado. He aquí el uso de las variables ERC, ERC1.

 *)

```
PROGRAM EMULADOR(INPUT,OUTPUT);
```

```
TYPE
```

```
  BSWAT=ARRAY [1..130] OF BYTE;(* Define el área de
                                trabajo del "byte
                                stream". *)
  PTR=ADS OF BYTE;             (* Define el tipo --
                                apuntador a varia-
                                bles *)
  BUFFT=ARRAY [1..1024] OF BYTE;(* Buffer de datos -
                                del "byte stream" *)
  NOMARC=STRING(7);
  PAS=STRING(1);
  CADENA=LSTRING (30);
  ARC=TEXT(132);
```

```
VAR
```

```
  MUCHOS:ARRAY [1..5] OF BYTE;    (* Almacena comandos
```

```

de manejo de video
con varios param. *)
PASS: PAS;
CARGRAF: ARRAY [1..32] OF BYTE; (* Contiene las co-
res correspondencias de
los caracteres gra-
ficos de la VT100 *)
BSWAB: BSWAT;
BUFFA, BUFF: BUFFT;
DISP: NOMARC;
(* Area de trabajo
del "byte stream" *)
(* Buffer de datos -
del "byte stream" *)
(* Contiene el nombre
del puerto (Lcomm)B
*)
I, MODO, LENPAS, MODKBD, LENDIS, ERC, ERC1, ERC2, CONT: WORD;
FH, J, J1, I1, CONT2: INTEGER;
RN1, RN2, REN, REN1, COL, COL1, CAR, C1, C2: BYTE;
COR1, COR2, LPA, LAR, L2, ENDOF, VECE: WORD;
KEYPAD, GRAF, GRAF1, CARSET, KP, CARSET1: BOOLEAN;
AUX, AUX1, AUX2, AUX3, AUX4, AUX5: BOOLEAN;
VARIOS: BOOLEAN;
ARCHIVO: CADENA;
X: LSTRING (10);
ARCHI, Y: ARC;
CAD: LSTRING (132);
TABLAK: SET OF BYTE;
(* Conjunto de Carac-
teres graficos vali-
dos *)

```

```

(*)
-----
-----
-----

```

```

FUNCION QUE SIERRA UN ARCHIVO ABIERTO CON OPENFILE
PARAMETRO: EL ENCABEZADO DEL ARCHIVO
-----
-----
-----

```

```

*)

```

```

FUNCTION CLOSEFILE (FH: INTEGER): WORD;
EXTERN;

```

(* -----

FUNCION ABRE UN ARCHIVO PARA SER UTILIZADO SECUENCIALMENTE
POR SECTORES

PARAMETROS:
FH: CONTENDRÁ EL ENCABEZADO DEL ARCHIVO
AR1: NOMBRE DEL ARCHIVO
LAR: LONGITUD DEL NOMBRE
PAS: PASSWORD DEL ARCHIVO
LPAS: LONGITUD DEL PASSWORD
MD: MODO DE APERTURA (LECTURA Y/O ESCRITURA)

(*)

FUNCTION
OPENFILE(FH:PTR;AR1:PTR;LAR:WORD;PAS:PTR;LPAS:WORD;MD:WORD):
WORD;
EXTERN;

(* -----

FUNCION QUE ABRE UN "BYTE STREAM"

PARAMETROS:
BSWA: APUNTA DOR AL AREA DE TRABAJO
DISP: APUNTA DOR AL NOMBRE DEL DISPOSITIVO
TAM: TAMAO DEL NOMBRE DEL DISPOSITIVO
PASS: APUNTA DOR AL PASSWORD DEL "BYTE STREAM"
LENPAS: LONGITUD DEL PASSWORD
MODO: MODO DE ACCESO LECTURA (Y/O ESCRITURA)
PBUF: APUNTA DOR AL BUFFER DE DATOS
TAM1: TAMAO DEL BUFFER

 *)

FUNCTION
 OPENBYTESTREAM(BSWA:PTR;DISP:PTR;TAM:WORD;PASS:PTR;LENPAS:WO
 RD;
 MODO:WORD;PBUF:PTR;TAM1:WORD):WORD;
 EXTERN;

 (*

 FUNCION QUE DETERMINA EL FORMATO DE LOS DATOS
 (NORMAL/IMAGEN/BINARIO)

PARAMETROS:

AW: APUNTA A LA AREA DE TRABAJO DEL "BYTE STREAM"
 IMAGE: DEFINE EL MODO:
 1 MODO NORMAL
 2 MODO IMAGEN
 3 MODO BINARIO

 *)

FUNCTION SETIMAGEMODE(AW:PTR;IMAGE:WORD) : WORD;
 EXTERN;

 (*

 FUNCION CHECA SI SE HA TECLADO ALGUN CARACTER

PARAMETROS:

MODEKBD: MODO : 1 CHECA SI SE TECLEO ALGUN CARACTER

0 ESPERA A QUE SE TECLEE UN CARACTER
 A: APUNTA A LA VARIABLE DONDE SE DEJARA EL -
 CARACTER LEIDO;

*)

FUNCTION READKBDIRECT(MODEKBD:WORD;A:PTR) : WORD;
 EXTERN;

(*

FUNCION QUE LEE Y ESCRIBE, RESPECTIVAMENTE, UN CARACTER
 AL "BYTE STREAM"

*)

FUNCTION READBYTE(AW:PTR;CAR:PTR) : WORD;
 EXTERN;

FUNCTION WRITEBYTE(AW:PTR;A:BYTE) : WORD;
 EXTERN;

(*

FUNCION CIERRA EL "BYTE STREAM APUNTADO POR AW

*)

```
FUNCTION CLOSEBYTESTREAM(AW:PTR) : WORD;
  EXTERN;
```

```
(*
```

```
-----
PROCEDIMIENTO QUE INICIALIZA LAS CORRESPONDENCIAS
DE CARACTERES GRAFICOS DE VT100 CON BURROUGHS
B-25
-----
```

```
*)
```

```
PROCEDURE INIC;
```

```
BEGIN
```

```

CARGRAFC1J:=32; (* BLANCO *)
CARGRAFC2J:=27; (* DIAMANTE *)
CARGRAFC3J:=127; (* CUADRO *)
CARGRAFC4J:=32; (* NO EXISTE CORRESP *)
CARGRAFC5J:=32; (* NO EXISTE CORRESP *)
CARGRAFC6J:=32; (* NO EXISTE CORRESP *)
CARGRAFC7J:=32; (* NO EXISTE CORRESP *)
CARGRAFC8J:=175; (* SIMBOLO DE GRADOS *)
CARGRAFC9J:=30; (* MAS/MENOS *)
CARGRAFC10J:=32; (* NO EXISTE CORRESP *)
CARGRAFC11J:=32; (* NO EXISTE CORRESP *)
CARGRAFC12J:=238; (* ESQUINA DERECHA
INFERIOR *)
CARGRAFC13J:=240; (* ESQUINA DERECHA
SUPERIOR *)
CARGRAFC14J:=239; (* ESQUINA IZQUIERDA
SUPERIOR *)
CARGRAFC15J:=237; (* ESQUINA IZQUIERDA
INFERIOR *)
CARGRAFC16J:=33; (* LINEAS CRUZADAS *)
CARGRAFC17J:=218; (* LINEA HORIZONTAL *)
CARGRAFC18J:=218; (* LINEA HORIZONTAL *)
CARGRAFC19J:=218; (* LINEA HORIZONTAL *)
CARGRAFC20J:=218; (* LINEA HORIZONTAL *)

```



```

CARGRAF[21]:=218; (* LINEA HORIZONTAL *)
CARGRAF[22]:=224; (* T IZQUIERDA *)
CARGRAF[23]:=227; (* T DERECHA *)
CARGRAF[24]:=221; (* T DE FONDO *)
CARGRAF[25]:=220; (* T DE TOPE *)
CARGRAF[26]:=225; (* LINEA VERTICAL *)
CARGRAF[27]:=229; (* MENOR O IGUAL *)
CARGRAF[28]:=31; (* MAYOR O IGUAL *)
CARGRAF[29]:=223; (* PI *)
CARGRAF[30]:=24; (* DISTINTO DE *)
CARGRAF[31]:=32; (* NO EXISTE CORRESP *)
CARGRAF[32]:=36; (* PUNTO CENTRADO *)
TABLAK:=0,1,2,3,11,12,13,14,15,17,18,19,21,22,23,24,25
,26,28,29,30,31;

```

END;

(*

```

-----
PROCEDIMIENTO QUE ESCRIBE AL "BYTE STREAM" UNA SECUENCIA DE
DOS CARACTERES PRECEDIDOS POR UN ESCAPE
-----

```

*)

PROCEDURE ESCRIBE(A,B:BYTE);

BEGIN

```

ERC:=WRITEBYTE(ADS BSWAB,27);
ERC:=WRITEBYTE(ADS BSWAB,A);
ERC:=WRITEBYTE(ADS BSWAB,B);

```

END;

(*

```

-----
PROCEDIMIENTO QUE DEFINE LAS CORRESPONDENCIAS DE MODO
-----

```

KEYPAD DE LA TERMINAL VT100 PARA SISTEMAS
 ESPECIFICOS COMO LO ES EL EDITOR DE PANTALLA
 "EDT".

 *)

PROCEDURE SECUENCIA;

BEGIN

CASE CAR OF

0:	ESCRIBE (79,82);	(* AYUDA - PF3 *)
1:	ESCRIBE (91,65);	(* CURSOR UP *)
2:	ESCRIBE (79,81);	(* MARCA - PF2 *)
3:	ESCRIBE (79,108);	(* PAS ANT - *)
11:	ESCRIBE (91,66);	(* CURSOR DOWN *)
12:	ESCRIBE (79,109);	(* PROX - *)
13:	ESCRIBE (79,83);	(* BORDE - PF4 *)
14:	ESCRIBE (91,68);	(* CURSOR LEFT *)
15:	ESCRIBE (79,80);	(* MOVER - PF1 *)
17:	ESCRIBE (79,110);	(* SUBIR - *)
18:	ESCRIBE (91,67);	(* CURSOR RIGHT *)
19:	ESCRIBE (79,77);	(* BAJAR - RETURN *)
21:	ESCRIBE (79,113);	(* F1 - 1 *)
22:	ESCRIBE (79,114);	(* F2 - 2 *)
23:	ESCRIBE (79,115);	(* F3 - 3 *)
24:	ESCRIBE (79,116);	(* F4 - 4 *)
25:	ESCRIBE (79,117);	(* F5 - 5 *)
26:	ESCRIBE (79,118);	(* F6 - 6 *)
28:	ESCRIBE (79,119);	(* F7 - 7 *)
29:	ESCRIBE (79,120);	(* F8 - 8 *)
30:	ESCRIBE (79,121);	(* F9 - 9 *)
31:	ESCRIBE (79,112);	(* F10 - 10 *)

END

END;

 (*)

PROCEDIMIENTO QUE INCREMENTA EL NUMERO DE RENGLON DE ACUERDO
AL TAMAÑO DE LA PANTALLA

*)

PROCEDURE INCREREN;

BEGIN

IF REN < 27 THEN
REN:=REN+1
ELSE
REN:=27;

END;

(*

PROCEDIMIENTO QUE BORRA TODA LA PANTALLA

*)

PROCEDURE PBORRA;

BEGIN

WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
WRITE(CHR(255)*'EF');
REN:=0;
COL:=0;

END;

(*

```

=====
PROCEDIMIENTO QUE DETERMINA UN TIEMPO FUERA PARA
RECIBIR UN CARACTER AL MOMENTO DE RECEPCION DEL
EQUIPO DIGITAL AL EQUIPO BURROUGHS
=====

```

*)

PROCEDURE TIEMPO(VAR X:BOOLEAN);

BEGIN

```

    I:=1;
    X:=TRUE;
    WHILE I < 5000 DO
    BEGIN
        ERC1:=READBYTE(ADS BSWAB,ADS CAR);
        IF ERC1=0 THEN
        BEGIN
            I:=5999;
            X:=FALSE;
        END;
        I:=I+1;
    END

```

END;

(*

```

=====
PROCEDIMIENTO QUE ABRE EL PUERTO DE COMUNICACIONES
COMO UN "BYTE STREAM"
=====

```

 *)

PROCEDURE ABREPTO;

BEGIN

ERC:=OPENBYTESTREAM(ADS BSWAB,ADS DISP,LENDIS,ADS
 PASS,LENPAS,*6D6D,ADS BUFF,1024);
 ERC:=SETIMAGEMODE(ADS BSWAB,MOD0)

END;

(*

PROCEDIMIENTO QUE CIERRA EL PUERTO DE COMUNICACIONES
 COMO UN "BYTE STREAM"

 *)

PROCEDURE CIERRA;

BEGIN

ERC:=CLOSEBYTESTREAM(ADS BSWAB)
 END;

(*

PROCEDIMIENTO QUE EFECTUA UNA TRANSMISION DE ARCHIVOS DEL
 EQUIPO BURROUGHS AL EQUIPO DIGITAL PERMITIENDO ASI
 LA TRANSFERENCIA DE INFORMACION, JUNTO CON LA Rutina RECIB,
 ENTRE AMBOS EQUIPOS

*)

PROCEDURE TRANS;

BEGIN

```

CAR:=0;
CONT2:=0;
ARCHIVO:=NULL;
I:=0;
CONT:=0;
WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
WRITE(CHR(255)*'EF');
WRITELN;
WRITE(' Archivo a Transmittir ');
WRITE(CHR(255)*'F'*' '*CHR(22)*CHR(1)*CHR(30)*CHR(1));
WRITE(CHR(255)*'C'*CHR(22)*CHR(1));
J:=0;
WHILE (CAR<>10) AND (CAR<>7) DO
BEGIN
  ERC:=READKBDIRECT(MODKBD,ADS CAR);
  IF (CAR<>10) AND (CAR<>7) AND (ERC <> 602) THEN
  BEGIN
    WRITE(CHR(CAR));
    CONCAT(ARCHIVO,CHR(CAR));
    J:=J+1;
    IF CAR = 8 THEN
    BEGIN
      DELETE(ARCHIVO,J,1);
      J:=J-1;
      DELETE(ARCHIVO,J,1);
      J:=J-1;
    END
  END;
  END;
  WRITELN(CHR(10));
  IF CAR <> 7 THEN
  BEGIN
    LPAS:=1;
    LAR:=J;
    ERC2:=OPENFILE(ADS FH,ADS ARCHIVOCL3,LAR,

```

```

ADS PASS,LPAS,#6D72);
ERC:=CLOSEFILE(FH);
IF ERC2 <> 0 THEN
  WRITELN(' >>>>> Error de Subsistema >>>>> NO Existe tal
Archivo')
ELSE
  BEGIN
    ASSIGN(ARCHI,ARCHIVO);
    RESET(ARCHI);
    ERC:=0;
    WHILE NOT EOF(ARCHI) DO
      BEGIN
        READLN(ARCHI,CAD);
        CONT2:=ORD(CAD[1]);
        FOR I1:=1 TO CONT2 DO
          BEGIN
            FOR J:=1 TO 500 DO J1:=0;
            I2:=ORD(CAD[I1]);
            CAR:=I2;
            ERC1:=WRITEBYTE(ADS BSWAB,CAR);
            END;
            CAR:=I3;
            ERC1:=WRITEBYTE(ADS BSWAB,CAR);
            ERC:=0;
            WHILE ERC = 0 DO
              ERC:=READBYTE(ADS BSWAB,ADS CAR)
            END;
          END;
        IF ERC2=0 THEN
          BEGIN
            WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
            WRITE(CHR(255)*'EF');
            WRITELN;
            WRITELN(' Transmisión completa ')
          END;
        ELSE
          BEGIN
            WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
            WRITE(CHR(255)*'EF');
          END;
        CLOSE(ARCHI);
      END;
    END;
  END;
  (* -----

```

 PROCEDIMIENTO QUE VACIA LO QUE QUEDA EN EL BUFFER DEL
 PUERTO DESPUES DE CADA LECTURA EN LA RECEPCION
 DE ARCHIVOS

*)

```

PROCEDURE VACBUF;
VAR CONTX:INTEGER;
BEGIN
  CONTX:=0;
  ERC:=0;
  WHILE ERC=0 DO
  BEGIN
    ERC:=READBYTE(ADS BSWAB,ADS CAR);
    IF ERC=0 THEN
    BEGIN
      IF ((CAR=13) OR (CAR=10)) THEN
      BEGIN
        WRITELN(ARCHI,CAD);
        CAD:=NULL;
      END
      ELSE
        CONCAT(CAD,CHR(CAR));
    END
    ELSE
    BEGIN
      IF CONTX=0 THEN
      BEGIN
        CONTX:=1;
        ERC:=0;
      END
    END
  END
END;

```

(* -----

 PROCEDIMIENTO QUE RECIBE UN ARCHIVO DEL EQUIPO DIGITAL
 AL EQUIPO BURROUGHS

*)

PROCEDURE RECIB;

BEGIN

```

CAR:=0;
ARCHIVO:=NULL;
AUX5:=FALSE;
CAD:=NULL;
J:=0;
CONT:=0;
WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
WRITE(CHR(255)*'EF');
Writeln;
WRITE(' Archivo en donde grabar ');
WRITE(CHR(255)*'F'*' '*CHR(24)*CHR(1)*CHR(30)*CHR(1));
WRITE(CHR(255)*'C'*CHR(24)*CHR(1));
WHILE (CAR<>10) AND (CAR<>7) DO
BEGIN
  ERC:=READKBDIRECT(MODKBD,ADS CAR);
  IF (CAR<>10) AND (CAR<>7) AND (ERC<>602) THEN
  BEGIN
    WRITE(CHR(CAR));
    CONCAT(ARCHIVO,CHR(CAR));
    J:=J+1;
    IF CAR=8 THEN
    BEGIN
      DELETE(ARCHIVO,J,1);
      J:=J-1;
      DELETE(ARCHIVO,J,1);
      J:=J-1
    END
  END
END;
Writeln(CHR(10));

```

```

IF CAR <> 7 THEN
BEGIN
  ERC2:=0;
  ASSIGN(ARCHI,ARCHIVO);
  REWRITE(ARCHI);
  IF ERC2 <> 0 THEN
    WRITELN(ERC,'Error en Apertura de Archivo')
  ELSE
    BEGIN
      ERC:=0;
      WHILE NOT AUX5 DO
      BEGIN
        ERC:=READBYTE(ADS BSWAB,ADS CAR);
        IF ERC = 0 THEN
          BEGIN
            IF ((CAR=13) OR (CAR=10)) THEN
              BEGIN
                C1:=19;
                ERC1:=WRITEBYTE(
                  ADS BSWAB,C1);
                WRITELN(ARCHI,CAD);
                CAD:=NULL;
                VACBUF;
                C1:=17;
                FOR J:=1 TO 50 DO J1:=0;
                ERC1:=WRITEBYTE(ADS BSWAB,C1);
              END
            ELSE
              BEGIN
                C1:=19;
                ERC1:=WRITEBYTE(ADS BSWAB,C1);
                CONCAT(CAD,
                  CHR(CAR));
                VACBUF;
                C1:=17;
                FOR J:=1 TO 50 DO
                  J1:=0;
                ERC1:=WRITEBYTE
                  (ADS BSWAB,C1);
              END
            END
          END
        ELSE TIEMPO(AUX5)
      END;
    END
  IF ERC2=0 THEN
    BEGIN
      WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
      WRITE(CHR(255)*'E'');
    END
  END
END

```

```

...TELN;
...TELN( 'Recepción completa ' )
END;
ELSE
BEGIN
WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
WRITE(CHR(255)*'EF');
END;
CLOSE(ARCHI);
END;

```

```

(*)
=====
=====

```

```

FUNCION QUE REGRESA EL VALOR NUMERICO DE UN PARAMETRO
CONTENIDO EN UNA SECUENCIA DE ESCAPE
=====
=====

```

```

*)

```

```

FUNCTION PARAM:BYTE;

```

```

VAR
AUX,X1:BYTE;
BEGIN

```

```

ENDOF:=48;
AUX:=0;
WHILE ((ENDOF>=48) AND (ENDOF <= 57)) DO
BEGIN
ERC:=READBYTE(ADS_BSWAB,ADS_ENDOF);
IF ((ERC=0) AND (ENDOF>=48) AND (ENDOF <= 57)) THEN
AUX:=(AUX*10)+ENDOF-48;
END;
PARAM:=AUX;

```

```

END;

```

```
(* -----  
-----  
PROCEDIMIENTO QUE COLOCA EL CURSOR EN EL RENBLON A  
COLUMNA B  
-----  
-----
```

```
*)
```

```
PROCEDURE PONCUR(A,B:BYTE);
```

```
BEGIN
```

```
  X:=NULL;  
  CONCAT(X,CHR(255));  
  CONCAT(X,'C');  
  CONCAT(X,CHR(B));  
  CONCAT(X,CHR(A));  
  WRITE(X);  
  X:=NULL;  
  REN:=A;  
  COL:=B;
```

```
END;
```

```
(* -----  
-----  
PROCEDIMIENTO MANDA LAS SECUENCIAS DE ATRIBUTOS  
DE VIDEO COMO SON:  
- VIDEO NORMAL  
- VIDEO INVERSO  
- VIDEO A MEDIO BRILLO  
-VIDEO SUBRAYADO  
-----  
-----
```

```
*)
```

```
PROCEDURE VIDEOATR(X:BYTE);
```

```
BEGIN
```

```
  CASE X OF
```

```
    0: WRITE(CHR(255)*'AA');
    1: WRITE(CHR(255)*'BP');
    4: WRITE(CHR(255)*'BB');
    5: WRITE(CHR(255)*'BH');
    7: WRITE(CHR(255)*'BD');
```

```
  END
```

```
END;
```

```
(*
```

```
=====
PROCEDIMIENTO QUE INTERPRETA LAS SECUENCIAS DE
ESCAPE SOPORTADAS POR TERMINALES VT100
=====
```

```
*)
```

```
PROCEDURE PANTALLA;
```

```
BEGIN
```

```
  X:=NULL;
  ENDFI:=0;
  C1:=0;
  C2:=0;
  CASE CAR OF
    28: PBORRA;
    29: WRITE(CHR(255)*'HF');
    31: WRITE(CHR(255)*'HF');
    25: WRITE(CHR(255)*'HN');
    40: BEGIN
```

```
(* ESC (*)
```

```

ERC:=10;
WHILE ERC <> 0 DO
ERC:=READBYTE(ADS BSWAB,ADS CAR);
CASE CAR OF
48: GRAF:=TRUE; (* ESC ( 0 *)
66: GRAF:=FALSE; (* CARACTERES GRAFICOS *)
(* ESC ( E *)
(* CARACTERES ANSI *)
END;
END;
24: WRITE(CHR(255)*'EF');
23: BEGIN
WRITE(CHR(255)*'EF');
WRITE(CHR(255)*'HN');
END;
17: BEGIN
ERC:=10;
WHILE ERC <> 0 DO
ERC:=READBYTE(ADS BSWAB,ADS C1);
WHILE ERC <> 0 DO
ERC:=READBYTE(ADS BSWAB,ADS C2);
PONCUR(C1,C2);
END;
15: WRITE(CHR(255)*'EL');
35: BEGIN (* ESC# *)
ERC:=10;
WHILE ERC <> 0 DO
ERC:=READBYTE(ADS BSWAB,ADS CAR);
CASE CAR OF
53: WRITE(CHR(255)*'B@'); (* ESC#5
ANCHO SENCILLO *)
54: WRITE(CHR(255)*'BP'); (* ESC#6
DOBLE ANCHO *)
56: BEGIN (* ESC#8
ALINEAMIENTO DE
VIDEO *)
PBORRA;
FOR J:=1 TO 26 DO
BEGIN
FOR J1:=1 TO 79 DO
WRITE('E');

```

```

        WRITELN
    END;
    FOR J1:=1 TO 79 DO
    WRITE('E');
    WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
    REN:=0;
    COL:=0;
    END;
END;
END;

```

```

(* ESC 7
  SALVAR CURSOR

```

```

*)

```

```

55: BEGIN
    REN1:=REN;
    COL1:=COL;
    GRAF1:=GRAF;
    CARSET1:=CARSET;
END;

```

```

(* ESC 8
  RESTAURAR CUR-
  SOR
  *)

```

```

56: BEGIN
    REN:=REN1;
    COL:=COL1;
    GRAF:=GRAF1;
    CARSET:=CARSET1;
    PONCUR(REN,COL);
END;

```

```

(* ESC =
  MODO KEYPAD *)

```

```

61: KEYPAD:=TRUE;

```

```

(* ESC >
  MODO NUMERICO
  )

```

```

62: KEYPAD:=FALSE;

```

```

(* ESC D
  CURSOR ABAJO *)

```

```

68: BEGIN
    IF REN = 27 THEN
    BEGIN
        X:=NULL;
        CONCAT(X,CHR(255));
    END;

```

```

CONCAT(X, 'S');
CONCAT(X, CHR(COR1));
CONCAT(X, CHR(COR2));
CONCAT(X, CHR(1));
CONCAT(X, 'U');
WRITE(X);
X:=NULL;
END;
WRITE(CHR(11));
INCREREN;
END;

(* ESC E
PROXIMA LINEA *)

69: BEGIN
  IF REN = 27 THEN
    BEGIN
      X:=NULL;
      CONCAT(X, CHR(255));
      CONCAT(X, 'S');
      CONCAT(X, CHR(COR1));
      CONCAT(X, CHR(COR2));
      CONCAT(X, CHR(1));
      CONCAT(X, 'U');
      WRITE(X);
      X:=NULL;
    END;
    INCREREN;
    COL:=0;
    FONCUR(REN, COL);
  END;

(* ESC M
CURSOR ARRIBA *)

77: BEGIN
  IF REN=0 THEN
    BEGIN
      X:=NULL;
      CONCAT(X, CHR(255));
      CONCAT(X, 'S');
      CONCAT(X, CHR(COR1));
      CONCAT(X, CHR(COR2));
      CONCAT(X, CHR(1));
      CONCAT(X, 'D');
      WRITE(X);
      X:=NULL;
    END;
    WRITE(CHR(1));
  END;

```



```

IF REN > 0 THEN REN:=REN-1;
END;

```

```

(* ESC Z
IDENTIFICACION
DE TERMINAL *)

```

```

89: BEGIN
CAR:=27;
ERC:=WRITEBYTE(ADS BSWAB,CAR);
CAR:=91;
ERC:=WRITEBYTE(ADS BSWAB,CAR);
CAR:=63;
ERC:=WRITEBYTE(ADS BSWAB,CAR);
CAR:=49;
ERC:=WRITEBYTE(ADS BSWAB,CAR);
CAR:=59;
ERC:=WRITEBYTE(ADS BSWAB,CAR);
CAR:=52;
ERC:=WRITEBYTE(ADS BSWAB,CAR);
CAR:=99;
ERC:=WRITEBYTE(ADS BSWAB,CAR);
END;

```

```

(* ESC c
RESTAURAR A
ESTADO INICIAL *)

```

```

99: BEGIN
PONCUR(0,0);
PBORRA;
CARSET:=FALSE;
GRAF:=FALSE;
END;

```

```

91: BEGIN (* ESC E *)
C1:=PARAM;
C2:=ENDOF;
IF ((C2<>75) AND (C2<>89) AND (C2<>109)) THEN
IF C1>0 THEN C1:=C1-1;
ENDOF:=C1;
IF ((C2<>75) AND (C2<>89) AND (C2<>109)) THEN
IF ENDOF=0 THEN ENDOF:=1;
CASE C2 OF

```

```

(* ESC E p n ?1
RESTAURAR PARAMETROS *)

```

```

63: BEGIN
CAR:=0;
PONCUR(0,0);
PBORRA;
WHILE ((CAR <> 108) AND (CAR <> 109))

```

```

DO
  ERC:=READBYTE(ADS BSWAB,AUS CAR);
END;
(* ESC [PnD
  CURSOR A LA IZQ. *)
68: WHILE ((ENDOF <> 0) AND (COL > 0))
  BEGIN
    WRITE(CHR(14));
    COL:=COL-1;
    ENDOF:=ENDOF-1
  END;
(* ESC [PnB
  CURSOR HACIA ABAJO *)
66: WHILE ((ENDOF <> 0) AND (REN < 27)) DO
  BEGIN
    WRITE(CHR(11));
    INCREREN;
    ENDOF:=ENDOF-1
  END;
(* ESC [PnC
  CURSOR ADELANTE *)
67: WHILE ((ENDOF <> 0) AND (COL < 80)) DO
  BEGIN
    WRITE(CHR(18));
    COL:=COL+1;
    ENDOF:=ENDOF-1
  END;
(* ESC [PnA
  CURSOR ARRIBA *)
65: WHILE ((ENDOF <> 0) AND (REN > 0 )) DO
  BEGIN
    WRITE(CHR(1));
    REN:=REN-1;
    ENDOF:=ENDOF-1
  END;
(* ESC [PsJ *)
74: CASE C1 OF
0: BEGIN
(* BORRA DE POSICION ACTUAL A FINAL DE PANTALLA *)
  WRITE(CHR(255)*'EF');
  PONCUR(REN,COL);
  END;
(* BORRA DE PRINCIPIO DE PANTALLA A
  POSICION ACTUAL *)
1: BEGIN
  RN1:=REN;

```

DO

```

RN2:=COL;
FOR J:=0 TO REN-1 DO
BEGIN
  PONCUR(J,0);
  PONCUR(REN,COL);
  WRITE(CHR(255)*'EL');
END;
FOR J:=0 TO COL DO
WRITE(CHR(32));
PONCUR(RN1,RN2);
END;
(* BORRA LA PANTALLA SIN MOVER EL
CURSOR *)
2: BEGIN
WRITE(CHR(255)*'C'*CHR(0)*CHR(0));
WRITE(CHR(255)*'E');
WRITE(CHR(255)*'B');
PONCUR(REN,COL);
END;
END;
(* ESC [PnK *)
75: CASE C1 OF
(* BORRA DE PRINCIPIO DE LA LINEA
A LA POSICION ACTUAL *)
0: WRITE(CHR(255)*'EL');
(* BORRA DE LA POSICION ACTUAL AL FINAL
DE LA LINEA *)
1: BEGIN
RN1:=REN;
RN2:=COL;
FOR J:=0 TO REN-1 DO
BEGIN
  PONCUR(J,0);
  WRITE(CHR(255)*'EL');
END;
FOR J:=0 TO COL DO
WRITE(CHR(32));
PONCUR(RN1,RN2);
END;
(* BORRA TODA LA LINEA *)
2: BEGIN
RN1:=COL;
PONCUR(REN,0);
WRITE(CHR(255)*'EL');
PONCUR(REN,RN1);
END;
END;
(* ESC [Pn m

```

```

                                ATRIBUTOS DE
                                VIDEO *)
109: VIDEOATR(C1);
                                (* ESC [Pn;Pn *)
59: BEGIN
    C2:=PARAM;
    IF C2>0 THEN C2:=C2-1;
    CASE END OF OF
    109: VIDEOATR(C2);
                                (* ESC [Pn;Pn H
                                POSICIONAMIENTO DE
                                CURSOR *)
    72: BEGIN
        REN:=C1;
        COL:=C2;
        PONCUR(C1,C2);
    END;
                                (* ESC [Pn;Pn ; *)
59: BEGIN
    VECES:=1;
    MUCHOSVECESJ:=C2;
    WHILE ((ENDOF<>63) AND
    (ENDOF<>109) AND (VECES <=5))
    DO
    BEGIN
        C2:=PARAM;
        MUCHOSVECESJ:=C2;
        VECES:=VECES+1;
    END;
    IF ((ENDOF<>63) AND
    (ENDOF<>109)) THEN
    WHILE ((CAR <> 63) AND
    (CAR<>109)) DO
    ERC:=READBYTE(ADS BSWAB,ADS
    CAR);
    CASE END OF OF
    63: BEGIN
        CAR:=0;
        PONCUR(0,0);
        PBORRA;
    END;
    109: BEGIN
        VECES:=VECES-1;
        FOR I2:=1 TO VECES
        VIDEOATR(MUCHOSVEC
        END;
DO
ESJ);

```

```

                                END;
                                END;                                (* ESC [Pn;Pn R *)
114: BEGIN
    COR1:=C1;
    COR2:=C2;
    END;
END;                                (* ESC [Pnc *)
99: BEGIN
    CAR:=27;
    ERC:=WRITEBYTE(ADS BSWAB,CAR);
    CAR:=91;
    ERC:=WRITEBYTE(ADS BSWAB,CAR);
    CAR:=63;
    ERC:=WRITEBYTE(ADS BSWAB,CAR);
    CAR:=49;
    ERC:=WRITEBYTE(ADS BSWAB,CAR);
    CAR:=59;
    ERC:=WRITEBYTE(ADS BSWAB,CAR);
    CAR:=52;
    ERC:=WRITEBYTE(ADS BSWAB,CAR);
    CAR:=99;
    ERC:=WRITEBYTE(ADS BSWAB,CAR);
    END;
    OTHERWISE
    J:=0
    END
END;
OTHERWISE
J:=0
END;
AUX4:=FALSE
END;

(* -----
   PROGRAM A          PRINCIPAL
   ----- *)

BEGIN

```

```

WRITE (CHR(255)*'X'*CHR(0));
WRITE (CHR(255)*'PF');
PBORRA;
INIC;
COR1:=0;
COR2:=28;
DISP:='[COMMJA]'; (* SE ASUME EL PUERTO [COMMJA] *)
LENDIS:=7;
PASS:=7;
LENPAS:=1;
MODKBD:=1;
AUX:=FALSE;
AUX4:=FALSE;
GRAF:=FALSE;
KEYPAD:=FALSE;
MODD:=1;
CONT:=0;
REN:=0;
COL:=0;
ABREPTU;

```

```

(*)
=====
=====

```

SE CHECA SI HAN SIDO TECLEADOS CARACTERES

```

*)
=====
=====

```

```

REPEAT
BEGIN
  AUX1:=FALSE;
  AUX3:=FALSE;
  REPEAT
  BEGIN
    ERC:=READK?DDIRECT(MODKBD,ADS CAR);
    IF ((ERC=0) AND (CAR IN TABLAK)) THEN
      KP:=TRUE ELSE KP:=FALSE;
    IF CAR >192 THEN CAR:=CAR-192;
    IF (CAR = 10) AND (ERC = 0) THEN
      BEGIN
        CAR:=13;

```

```

        AUX3:=TRUE
    END;
    (* LA TECLA <F1> INDICA TRANSMISION DE ARCHIVOS *)
        IF ((CAR = 21) AND (NOT KEYPAD)) THEN
            TRANS
        ELSE
            (* LA TECLA <F2 INDICA RECEPCION DE ARCHIVOS *)
                IF ((CAR = 22) AND (NOT KEYPAD)) THEN
                    RECIB
                ELSE
                    IF (ERC=0)AND (CAR<>4) THEN
                        BEGIN
                            IF ((CAR IN TABLAK) AND KEYPAD
                                AND KP) THEN
                                SECUENCIA
                            ELSE
                                ERC:=WRITEBYTE(ADS BSWAB,CAR)
                        END
                    ELSE
                        IF ERC=602 THEN
                            AUX1:=TRUE
                        ELSE
                            AUX1:=TRUE;
                            IF CAR = 4 THEN
                                BEGIN
                                    AUX1:=TRUE;
                                    AUX1:=TRUE
                                END
                            END
                        UNTIL AUX1;
                        AUX2:=FALSE;
                    END
                END
            END
        END
    (* -----
    -----
    SE CHECA SI HAN LLEGADO CARACTERES POR EL PUERTO
    -----
    -----
    *)

```

```

ERC:=READBYTE(ADS BSWAB,ADS CAR);
IF ERC = 0 THEN
BEGIN
  IF (GRAF AND (CAR=10)) THEN
    CAR:=11;
  IF (GRAF AND (CAR=8)) THEN
    CAR:=14;
  IF CAR = 10 THEN COL:=0;
  IF (CAR = 13) AND AUX3 THEN
  BEGIN
    CAR:=10;
    COL:=0;
  END;
  IF CAR = 27 THEN
    AUX4:=TRUE
  ELSE
    IF AUX4 THEN
      PANTALLA
    ELSE
      BEGIN
        IF (GRAF AND (CAR > 94) AND
          (CAR < 127)) THEN
          BEGIN
            J:=CAR-94;
            CAR:=CARGRAFLJJ;
          END;
          WRITE(CHR(CAR));
          IF ((CAR<>27) AND
            (CAR<>10)) THEN
            COL:=COL+1;
          END;
          AUX3:=FALSE
        END
      ELSE
        AUX2:=TRUE;
      END;
    END
  UNTIL AUX;
  CIERRA;
  WRITE(CHR(255)*'PN')
END.

```


A.2 Algoritmo de Flujo Máximo

```

(*) -----
-----
-----
*)

Program FlujoMaximo(Input,Output);
Const
  N=123;                               (* Número de Nodos *)
  NoRastreado=N;
  NoValido=MaxInt;

Type
  Nodo          = 1..N;
  Xnodo         = -N..N;
  Vector        = Array [Nodo] of Xnodo;
  Ynodo         = 0..N;
  Matriz        = Array [Nodo,Nodo] of Integer;
  Operación     = (Push,Pull);

Var
  NodoRef       : Node;   (* Nodo con menos exceso *)
  PotencialMin  : Real;   (* Exceso de capacidad en *)
  Nivel         : Vector; (* el nodo de referencia *)
  i,j           : Node;   (* Arreglo en que se define *)
  i,j           : Node;   (* la red de niveles *)
  i,j           : Node;   (* Indices *)

(*) -----
-----
-----
*)

Function ValMin(a,b:Real):Real;

```

```

Begin
  If a < b then
    ValMin:=a
  else
    ValMin:=b
End;

```

```

(*) -----
      Función que Recorre la Red de Niveles a partir del
      Nodo i hacia el nodo origen.
-----

```

```

*)

```

```

Procedure Recorre(i:Nodo);

```

```

  Var

```

```

    j:Nodo;
    h:Xnodo;

```

```

  Begin

```

```

    Nivel[i]:=-Nivel[i];

```

```

    h:=Nivel[i];

```

```

    For j:=1 to N do

```

```

      If (j <> i) and

```

```

        (-Nivel[j]=h-1) and

```

```

        ((FEj,i] < CEj,i]) or (F[i,j] > 0)) then

```

```

          Recorre(j);

```

```

  End;

```

```

(*)

```

```

-----
      Función que chequea si es posible construir otra
      Gráfica de Niveles, si si la construye.
-----

```

```

*)

```

```

Function NivelPosible:Boolean;

```

```

  Var

```

```

    i,j:Nodo;

```

```

    k:Ynodo;

```

```

    NivelVacio:Boolean;

```

```

  Begin

```

```

    k:=0;

```

```

    (* Indica que Nivel se ha Construído *)

```

```

For i:=1 to N do (* Se inicializa cada nodo con un
                    valor no válido *)
  Nivel[i]:=NoRastreado;
  Nivel[s]:=k; (* El nodo fuente es puesto en el
                    nivel 0 *)
  Repeat
    k:=k+1;
    NivelVacio:=True;
    For i:= 1 to N do
      If -Nivel[i] = k-1 then
        For j:=1 to N do
          If (Nivel[j]=NoRastreado) and
            ((F[i,j] < C[i,j]) or
             (F[j,i] > 0)) then
            Begin
              Nivel[j]:=-k;
              NivelVacio:=False
            End
        End
    Until (Nivel[t] <> NoRastreado) or NivelVacio;
    NivelPosible:=Not NivelVacio;
    Recorre(t)
  End;

```

```

(*) -----
    Procedimiento para Encontrar el Nodo de Referencia
    a partir del nodo destino t.
    -----
(*)

```

```

Procedure EncNodoRef(i:Nodo);
Var
  j:Nodo;
  h,h1:Xnodo;
  CapEntrada,CapSalida:Real;
Begin
  h1:=Nivel[i];
  CapEntrada:=0;
  CapSalida:=0;
  For j:=1 To N Do
    Begin
      h:=Nivel[j];
      If (h1 = h-1 ) and (j <> s) and
        ((F[j,i] < C[j,i]) or (F[i,j] > 0))Then
        EncNodoRef(j);
    End
  End

```

```

      If h1 = h-1 Then
        CapEntrada:=CapEntrada+(C[i,j]-
FC[j,i])+FC[i,j];
      If h1 = h-1 Then
        CapSalida:=CapSalida+(C[i,j]-FC[i,j])+FC[j,i];
    End;
    If (i <> s) and (i <> t) and
      (min(CapEntrada,CapSalida) < PotencialMin) Then
      Begin
        PotencialMin:=min(CapEntrada,CapSalida);
        NodoRef:=i;
      End
    End;
  End;

```

```

(* -----
   Procedimiento que aumenta el flujo a partir del
   nodo i efectuando un 'Push' o 'Pull' en tomando
   como unidad el PotencialMin.
  ----- *)

```

```

Procedure PushPull(i:Nodo;Unidad:Real;Ope:Operación);
Var
  j,k1,k2,NivelVisto:Ynodo;
  r:Real;
Begin
  j:=0;
  While (Unidad > 0) and (j < N) Do
  Begin
    j:=j+1;
    If Ope = Push Then
    Begin
      k1:=i;
      k2:=j;
      NivelVisto:=Nivel[i]+1
    End
    Else
    Begin
      k1:=j;
      k2:=i;
      NivelVisto:=Nivel[i]-1
    End;
    r:=min(Unidad,C[k1,k2]-F[k1,k2]+F[k2,k1]);
    If (r > 0) and (Nivel[j] = NivelVisto) Then

```

```

      Begin
        Unidad:=Unidad-r;
        FLK1,k2j:=FLK1,k2j+r-min(r,FLK2,k1j);
        FLK2,k1j:=FLK2,k1j-min(r,FLK2,k1j);
        If (J <> s) and (J <> t) Then
          PushPull(j,r,Ope)
        end
      end; end
end;

```

```

(*)
-----

```

PROGRAMA PRINCIPAL

```

(*)
-----

```

```

Begin
  For i:=1 To N Do
    For j:=1 to N do
      FLi,j:=0;
    While NivelPosible Do
      Begin
        PotencialMin:=NoValido;
        EncNodoRef(t);
        PushPull(NodoRef,PotencialMin,Push);
        PushPull(NodoRef,PotencialMin,Pull)
      End
    End
  End.

```

BIBLIOGRAFIA

B.1 ARTICULOS

Computing Communications - A Perspective of the Evolving Environment

L. M. Branscomb

IBM System Journal

Volume Eighteen - Number Two - 1979

And Introduction to Network Architectures and Protocols

P.E. Green

IBM Systems Journal

Volume Eighteen-- Number Two - 1979

Public Data Networks; Their Evolution, Interfase, and Status

J. R. Halsey, L. E. Hardy and L. F. Powning

IBM Systems Journal

Volume Eighteen - Number Two - 1979

SNA and Emerging International Standards

F. P. Carr and D. H. Neal

Volume Eighteen - Number Two - 1979

Routing and Flow Control in Systems Network Architecture

V. Ahuja

IBM Systems Journal

Volume Eighteen - Number Two - 1979

Evolution of Laboratory Communication Network

R. S. Moore

IBM Systems Journal

Volume Eighteen - Number Two - 1979

**Potential Technology Implications for Computers and
Telecommunications in the 1980's**

W. D. Frazer

IBM Systems Journal

Volume Eighteen - Number Two - 1979

Data Communications Facilities

Averbach On ...

Communications Processors

Averbach On ...

Multiplexors

Averbach On ...

Modems and Telephone Couplers

Averbach On ...

Remote Batch Terminals

Averbach On ...

Facsimile Equipment

Averbach On ...

B.2 PUBLICACIONES

Introduction to Data Communications Network Design

IBM

Reprinted, February 1984

IBM Data Communications Fundamentals

IBM

Second Edition, March 1980

Advanced Communications Function Network

IBM

February 1981

General Information - Binary Synchronous Communications

IBM

Third Edition, October 1970

B.3 LIBROS

Computer Networks
Andrew S. Tanenbaum
Prentice - Hall

Technical Aspects of Data Communication
John E. Mc. Namara
Digital Equipment Corporation
Third Printing, June 1978

Computer - Communications Network Design and Analysis
Misha Schwartz
Prentice - Hall

Fundamentos de Comunicación de Datos
Jerry Fitzgerald y Tom S. Eason
Limusa
Primera Edición 1981

Introducción al Teleprocesamiento
James Martin
Miana
Primera Edición, Abril 1975