

6 Ejemplar 2
Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS



UNAMNET

RED UNIVERSITARIA DE TELEPROCESO

T E S I S

PARA OBTENER EL TÍTULO DE
A C T U A R I O
P R E S E N T A

MARCO ANTONIO BENDIMEZ SERRANO

MEXICO, D. F.

6112

1979



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas

Tesis Digitales

Restricciones de uso

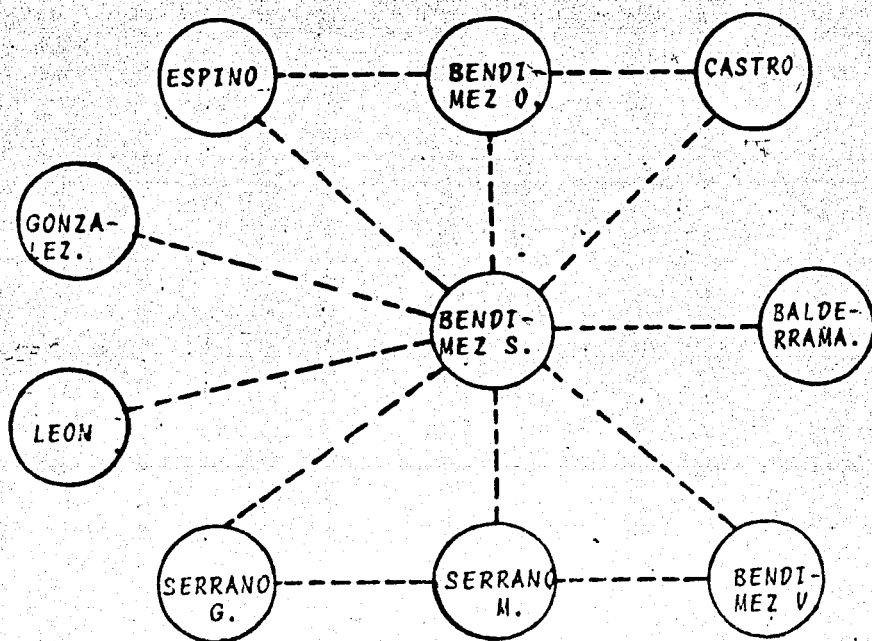
DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta 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.

A LAS FAMILIAS:



A MIS AMIGOS.

I N D I C E

CAPITULO 1.	REDES DE COMPUTO	1
1.1	REDES DE COMPUTADORAS	2
1.1.1	REDES DE COMUNICACION DE COMPUTADORAS	2
1.1.2	RED DE COMPUTADORAS	5
1.1.3	RED DE TERMINALES	15
1.2	ARQUITECTURA	17
1.2.1	TOPOLOGIA	17
1.2.2	COMPOSICION	20
1.2.3	CONTROL	20
1.2.3.1	CONEXION INICIAL	20
1.2.3.2	CONTROL DE FLUJO	21
1.2.3.3	ENRUTAMIENTO	22
1.2.3.4	MONITOREO Y MANTENIMIENTO	23
1.2.4	MEDIDA	23
1.2.5	CONFIABILIDAD	24
1.2.6	TRANSPARENCIA	24
1.2.7	TAMAÑO	24
1.3	COMPONENTES DE UNA RED	24
1.3.1	COMPONENTES BASICOS DE HARDWARE	25
1.3.1.1	CIRCUITOS	25
1.3.1.2	TERMINALES	26
1.3.1.2.1	TERMINALES BATCH	26
1.3.1.2.2	TERMINALES INTERACTIVAS	27
1.3.1.3	ADAPTADORES DIGITALES Y ANALOGICOS	27

1.3.1.4	PROCESADORES DE LA RED	28
1.3.1.4.1	PROCESADOR DE COMUNICACION DE DATOS	28
1.3.1.4.2	CONCENTRADORES	29
1.3.1.4.3	PROCESADOR SWITCHHEADOR	30
1.3.2	COMPONENTES BASICOS DE SOFTWARE	31
1.3.2.1	MINICOMPUTADORAS COMO PROCESADORES DE COMUNICACION DE DATOS	32
1.3.2.1.1	PROCESADORES DE COMUNICACION	33
1.3.2.1.2	CONCENTRADORES	35
1.3.2.2	PROCESOS DE CONTROL DE COMUNICACION	35
1.4	CONTROL DE ERRORES	36
1.5	ENCABEZADO DE INFORMACION	38
1.6	PROTOCOLOS	39
1.6.1	PROTOCOLO IMP-A-IMP	40
1.6.2	PROTOCOLO IMP-A-ANFITRIONA	41
1.6.3	ANFITRIONA-A-ANFITRIONA	42
1.6.4	PROTOCOLOS NIVEL USUARIO	42
1.7	APLICACIONES Y SERVICIOS	43
1.7.1	ACCESO	44
1.7.2	BACKUP	44
1.7.3	RECURSOS COMPARTIDOS	45

CAPITULO 2.	COMUNICACION DE DATOS EN LA B-6700	47
2.1	HARDWARE	47
2.1.1	TERMINALES	48
2.1.2	LINEAS Y MODEM	48

2.1.3	ADAPTERS	48
2.1.4	PROCESADOR DE COMUNICACION DE DATOS	49
2.1.4.1	ADAPTER CLUSTER	50
2.2	FUNCIONES DEL DCP	53
2.3	INTERFACE LOGICA CON EL SISTEMA	55
2.4	INTERFACE LOGICA CON LA RED DE COMUNICACION	56
2.5	TABLAS DE ESTACIONES Y LINEAS DEL DCP	56
2.5.1	VECTOR DE LINEA	56
2.5.2	TABLAS DE LINEA	57
2.5.2.1	ESTADOS DE LINEA	58
2.5.2.2	VECTOR DE INFORMACION	60
2.5.2.3	TABLAS DE LA ESTACION	62
2.6	CONTROL LOGICO DE LINEA	63
2.7	DESCRIPCION GENERAL DEL SOFTWARE DEL SISTEMA DE COMUNICACION DE DATOS	64
2.7.1	PROGRAMA FUENTE DEL LENGUAJE DEFINIDOR DE LA RED	65
2.7.2	COMPILADOR NDL	66
2.7.3	CONTROLADOR DE COMUNICACION DE DATOS (DCC)	66
2.7.4	PROGRAMA OBJETO DEL DCP	69
2.7.5	SISTEMA CONTROLADOR DE MENSAJES (MCS)	69
2.7.6	COMPILADOR DCALGOL	70
2.8	INICIALIZACION	72
2.9	ENTRADA DE UNA ESTACION AL SISTEMA DE DATACOM	73
2.10	SALIDA DEL MENSAJE A LA ESTACION A	77

CAPITULO 3	PROTOCOLOS Y DISCIPLINAS	81
3.1	REQUERIMIENTOS DE MODULARIDAD PARA PROTOCOLOS JERARQUICOS	87
3.1.1	ALGUNAS OBSERVACIONES ACERCA DE LA ESTRUCTURA DE UN ESTADO MODULAR EN LA JERARQUIA DE PROTOCOLOS	90
3.1.2	ALGUNAS VENTAJAS DE LA ESTRUCTURA MODULAR	95
3.1.3	MODIFICACIONES EN PROTOCOLOS MODULARES JERARQUICOS	95
3.2	VERIFICACION Y CORRECCION	99
3.3	CONFIABILIDAD Y EFICIENCIA	100
3.3.1	CARACTERISTICAS DE LOS MEDIOS DE TRANSMISION	101
3.3.1.1	RETARDOS	102
3.3.1.2	DUPLICACION	104
3.3.1.3	PERDIDA Y DAÑOS	104
3.3.1.4	ORDEN	105
3.3.1.5	TAMAÑO DEL PAQUETE	105
3.3.1.6	ANCHO DE BANDA	106
3.4	PROTOCOLOS	106
3.4.1	PROTOCOLO BASICO	108
3.4.2	CODIGO	112
3.5	EJEMPLOS DE DISCIPLINAS DE LINEA	126
3.5.1	DISCIPLINA DE LINEA POLL	126
3.5.2	DISCIPLINA DE LINEA SELECTION	132
3.5.3	DISCIPLINA DE LINEA FAST SELECT	138

3.5.4	DISCIPLINA DE LINEA BROAD CAST SELECT	143
3.5.5	DISCIPLINA DE LINEA GROUP SELECT	149
3.5.6	DISCIPLINA DE LINEA GROUP POLL	154
3.5.7	DISCIPLINA DE LINEA POINT-TO-POINT CONTENTION (LINEAS DEDICADAS)	162
3.5.8	DISCIPLINA DE LINEA POINT-TO-POINT SWITCHED CONTENTION	169
3.5.9	DISCIPLINA DE LINEA POINT-TO-POINT MODO BATCH	179
3.5.10	DISCIPLINA DE LINEA SECUENTIAL SELECT	186
3.5.11	DISCIPLINA DE LINEA POINT-TO-POINT NO CONTENTION (SOLO ENVIA)	192
3.5.12	DISCIPLINA DE LINEA POINT-TO-POINT NO CONTENTION (SOLO RECIBE)	196
CAPITULO 4.	PROGRAMA DE APLICACION	202
CAPITULO 5.	CONCLUSIONES	219
	VOCABULARIO	221
	INDICE DE FIGURAS	225
	INDICE DE CUADROS	227
	INDICE DE DIAGRAMAS	227
	REFERENCIAS	228

1. REDES DE COMPUTO

Antes de dar características de las diferentes redes que existen, nos ubicaremos en qué es una red de computadoras y qué una red de terminales.

Una red de computadoras es un conjunto de computadoras y terminales interconectadas por un sistema de comunicación.

Una red de terminales es un conjunto de terminales conectadas a una computadora que llamaremos anfitriona.

El término terminal, en ambos párrafos, va desde un simple dispositivo capturador de datos, teletipos, panta-

llas de video hasta una "entrada-de-trabajos-remotos" (RJE).

1.1 RED DE COMPUTADORAS

Dentro del area de redes de computadoras existen dos diferentes tipos de redes que son:

- a) Red de comunicación de computadoras
- b) Red de computadoras

En ambos casos hay características comunes como que la red es la responsable del sistema de comunicación de mensajes.

En donde hay diferencia es en el manejo de los recursos de cómputo.

Para el tipo de redes de comunicación de computadoras, la responsabilidad del manejo de los recursos recae sobre el usuario; mientras que en el segundo caso el usuario puede depender del auxilio de un sistema operativo de la red, en la adquisición y manejo de los recursos.

1.1.1 REDES DE COMUNICACION DE COMPUTADORAS

La característica básica de una red de comunicación de computadoras es que el usuario ve la red como una colección de diferentes sistemas de cómputo con servicios

variados y capacidades. Dentro de estos diferentes sistemas de cómputo disponibles en la red, el usuario deberá elegir explícitamente el sistema de cómputo con el cual quiere trabajar. El usuario establecerá la conexión a través de la red con el sistema elegido. Ya hecho el enlace, él podrá comunicarse como si fuera un usuario local al sistema anfitrión.

Hay muchos y variados recursos a disposición del usuario, en este tipo de redes. Estos recursos pueden incluir subrutinas y paquetes de utilizarla como son: simuladores, paquetes de estadística, paquetes de investigación de operaciones, etc, también tienen a su disposición hardware especializado y bases de datos.

Para poder acceder estos recursos, el usuario de la red deberá determinar el sistema donde estos recursos residen y familiarizarse con los comandos del sistema anfitrión al que se enlazó, ya que no existe una interface común de comandos en todas las máquinas para dar facilidad al usuario.

Un ejemplo de este tipo de red es la red TYMNET con la que el CONACYT está enlazado. La fig(1.1) ilustra la forma en que un usuario ve la red.

SISTEMA DE COMPUTO A

SISTEMA DE COMPUTO B

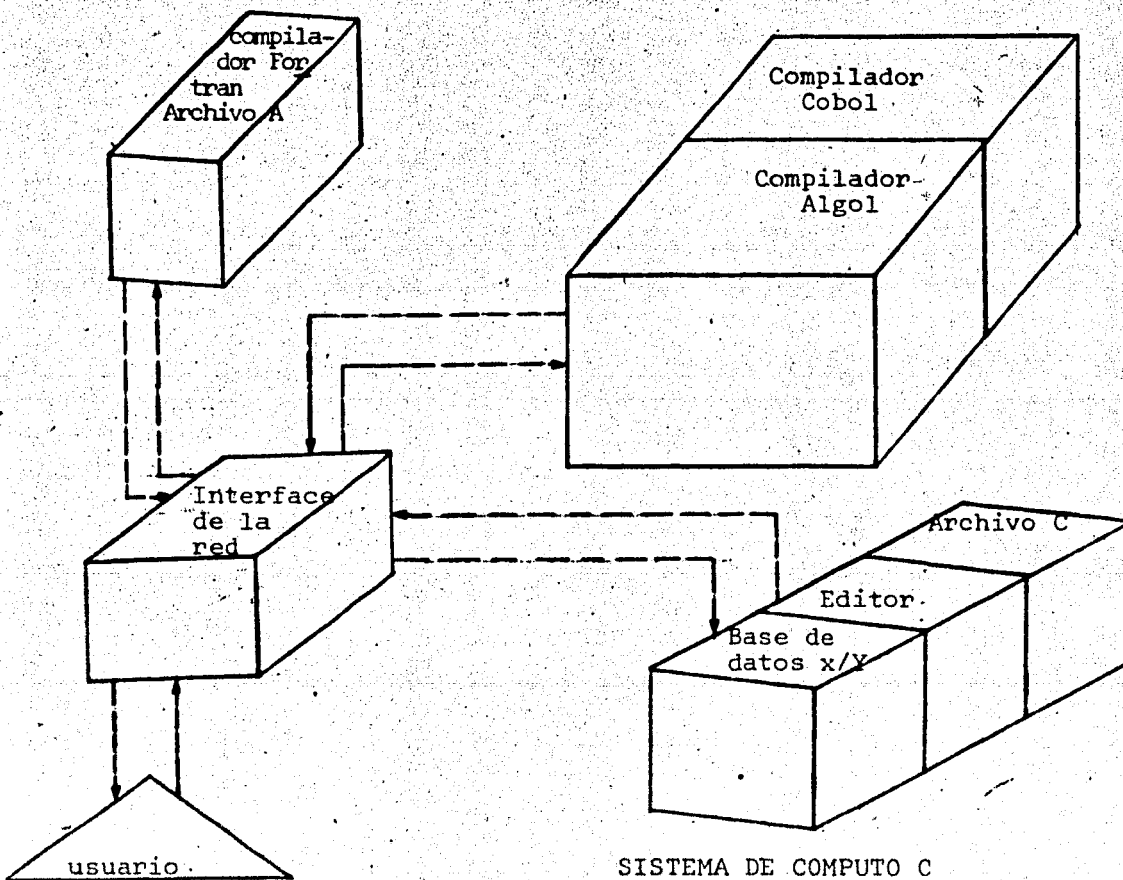


Fig. 1.1 VISTA DEL USUARIO, EN UNA RED DE COMUNICACION DE COMPUTADORAS

1.1.2 RED DE COMPUTADORAS

El usuario de una red de computadoras, al igual que el usuario de la red de comunicación de computadoras, tiene a su disponibilidad una gran variedad de recursos de cómputo; pero en contraste con el otro usuario; él ve la red entera como un gran sistema de cómputo. El usuario no necesita diversos sistemas operativos y protocolos para poder utilizar los variados recursos disponibles en la red, ya que los procesos para la ejecución de un trabajo, son transparentes al usuario. Es por tanto innecesario que el usuario conozca el sitio de cada recurso requerido; la esencial característica de la red de computadoras es que existe un sistema operativo de la red, que toma decisiones por el usuario.

Por ejemplo: si un usuario quisiera compilar un programa, el sistema operativo de la red elegiría a qué sistema de cómputo le corresponde o puede hacerlo. En caso de que por algún motivo ese sistema de cómputo se cayera, el sistema operativo de la red, automáticamente enviaría esa compilación que fue pedida por el usuario a otro sistema de cómputo que pueda llevarla a cabo, sin que el usuario se entere. La fig 1.2 muestra como verla un usuario a este tipo de redes.

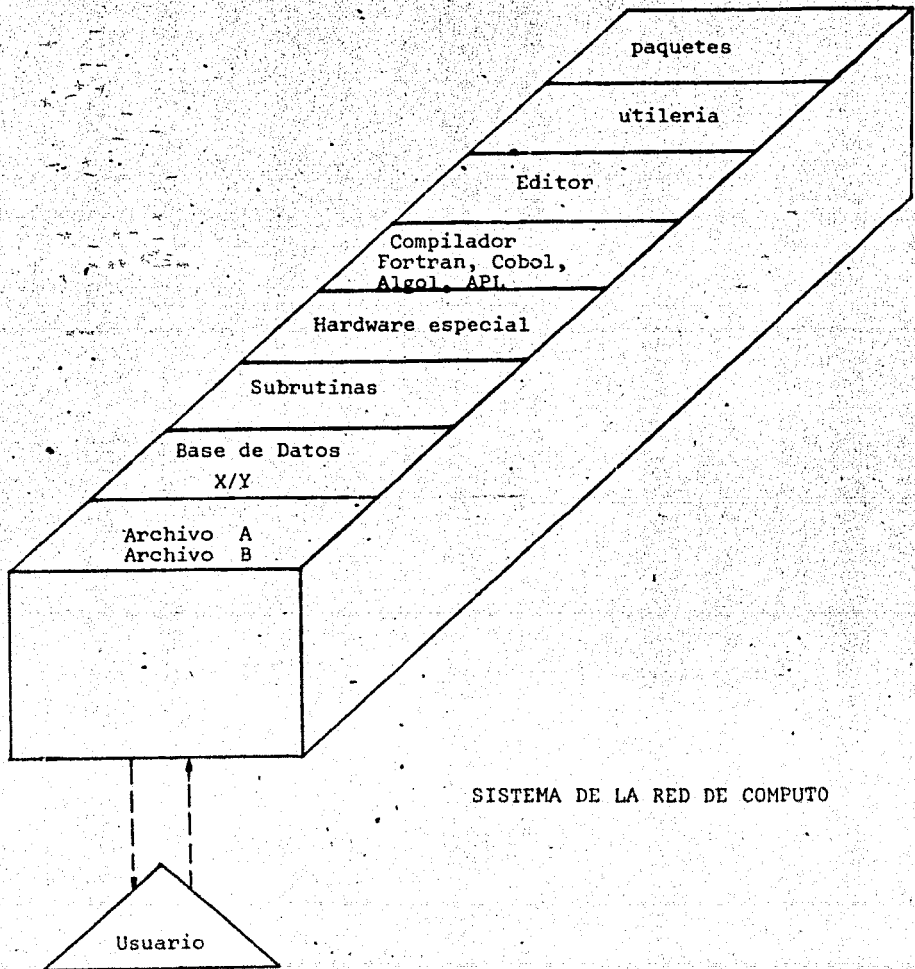


FIG. 1.2 VISTA DEL USUARIO EN UNA RED DE COMPUTADORAS

Un ejemplo de redes de comunicación de computadoras es la red TYMNET, desarrollada por TYMSHARE, INC.

En TYMNET hay más de 20 computadoras anfitrionas, ésto incluye PDP-10_s, XDS-940_s así como una IBM-370.

TYMNET es una red sofisticada de comunicación de datos que emplea 80 procesadores de comunicación (TYSMATS), diseminados en U.S.A., para acceder 26 computadoras localizadas en centros de cómputo de N.Y., Englewood Cliffs, N.J., Houston, Tex., California.

Los procesadores de comunicación llamados TYSMATS usan computadoras VARIAN 620 modificadas; 23 de estas sirven como "base de computadoras", así denominadas (base TYSMATS), cada una asociada directamente con su propia unidad de procesador central (CPU) y las otras 57 forman nodos remotos (remote TYSMATS) a través de los cuales, las terminales pueden entrar al sistema.

Cualquier concentrador nodal (TYSMAT) puede tener tantas como 200 a 300 terminales asociadas con él, pero no más de 31 terminales a un tiempo.

Una terminal, al incorporarse a la red, consigue un número local para llamar al nodo más cercano.

El usuario debe explícitamente designar la computadora que procesará su trabajo; además, el usuario deberá ser autorizado para poder entrar al sistema de cómputo que él eligió.

Como algunos recursos especiales no están en todas las computadoras anfitrionas, el usuario deberá conocer donde se encuentran estos recursos para poder accederlos.

Estos recursos especiales pueden ser compiladores, simuladores, intérpretes, subrutinas, paquetes, etc.

TVMNET tiene algunas ventajas con algunas redes de comunicación de computadoras, ya que tiene un lenguaje llamado TYMEX que sirve para poder trabajar con las PDP-10_s y las XDS-940_s sin ningún problema, además, el usuario tiene la opción de poder trabajar con los comandos normales de la máquina (i.e. DEC. sistema 10 para PDP-10 y Frenle para la XDS-940).

Un serio defecto de TVMNET, como de algunas otras redes de este tipo, es que los back-up no pueden salir donde está el usuario, sino que el usuario tiene que hablar a TVMNET y pedir su back-up. Entonces la copia del archivo se transfiere manualmente al segundo sistema de cómputo y de ahí al usuario.

La red de TYMNET es considerada como una red de switcheo en línea en el sentido virtual, ya que cuando un usuario se conecta a un nodo particular, se le asigna una ruta o canal virtual, a través de la red al CPU apropiado.

La ruta es asignada por un programa supervisor soportado en algunos de los CPU_s, este supervisor hace el papel de seleccionador de ruta que seguirá la transmisión, tratando de elegir apuntadores que no estén muy cargados o que tengan líneas dañadas para evitar lentitud y errores.

El supervisor es un programa que corre en tiempo compartido en una XDS-940; además de seleccionar canales o rutas elabora diagnósticos y guarda estadísticas.

En caso de fallas, el supervisor proporciona "back-up" de toda la red en forma detallada para su restauración. Estos back-up contienen todas las tablas de permutación de asignación de canales, por cada línea; otros ejemplos de redes de comunicación de computadoras son: ARPANET, ALOHA, NASDAQ, OCTOPUS, DECNET.

Ejemplo de red de computadora:

Un ejemplo de red de computadoras es la red DCS (DISTRIBUTED COMPUTER SYSTEM).

Esta red fue desarrollada en la Universidad de Cali-

fornia en IRVINWE.

DCS consiste de varios procesadores, cada uno de los cuales tiene por hardware una interface con un circuito cerrado unidireccional, de una linea de comunicaciones (fig 1.3)

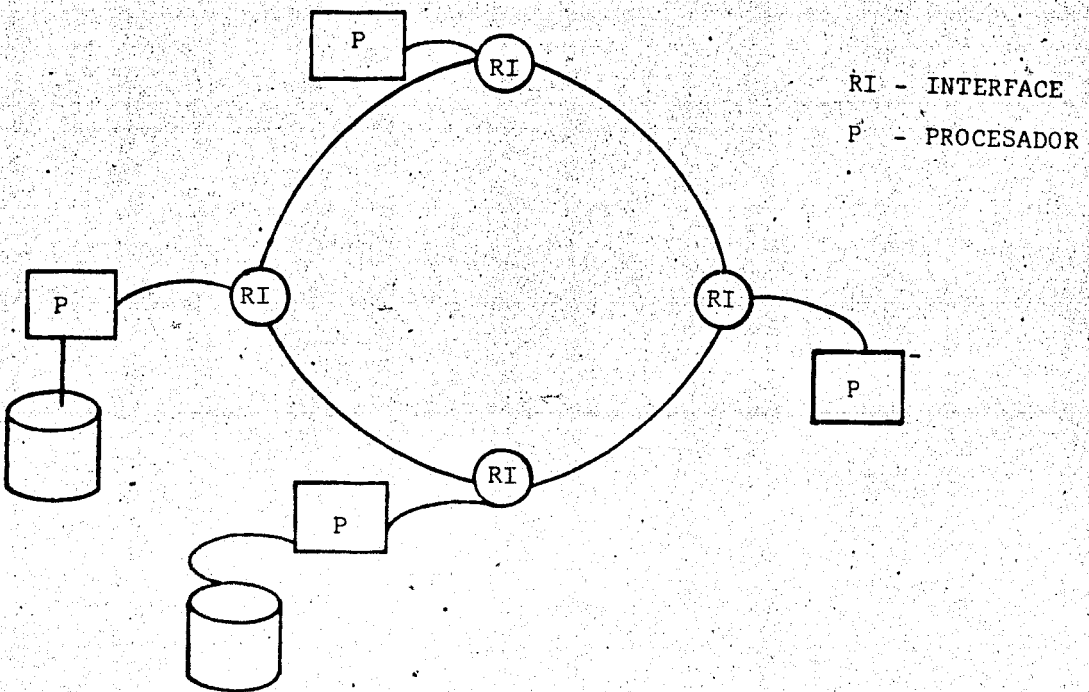


FIG. 1.3 SISTEMA DISTRIBUIDO DE COMPUTADORAS

La interface de hardware, conocida como anillo de interface, "decide" si un mensaje que pasa a través de la linea esta direccionado a un proceso localizado y si es

así envía el mensaje.

DCS se diseñó para que los recursos de cómputo se pudieran añadir con un pequeño incremento en el costo de software y hardware. El software del sistema está orientado a procesos, ya que toda actividad se implementa por procesos dentro del sistema.

Las comunicaciones proceso-a-proceso se ejecutan como las comunicaciones computadora-a-computadora. Esto es, que un proceso que está en una computadora, puede comunicarse con un proceso conocido, sin importar si ese proceso se está ejecutando en la misma computadora o en una computadora remota.

La comunicación se implementa como mensajes direccionados a un único proceso identificado, dentro del sistema operativo en donde reside. Cada procesador tiene un modelo de transmisión de mensajes. Estos modelos dirigen el mensaje a los anillos de comunicación, cuando el mensaje está dirigido a un proceso no local.

Para mantener la estructura de procesos orientados al sistema operativo de la red, cada recurso es controlado por un proceso correspondiente.

Los recursos con los cuales ejecuta funciones idénticas, están cada uno asociado con procesos separados; todos son direccionados con el mismo nombre. Para distinguir entre varios recursos idénticos, llamados dentro del sistema, DCS provee nombres a los procesos con una dirección, para distinguirlos entre los demás.

En este caso, cada requerimiento de recursos de hardware y software es representando con un mensaje direccionado al nombre del proceso. Consecuentemente el usuario que requirió hardware o software es ignorante de la localización de los recursos que necesita.

En caso de que existan idénticos recursos de hardware o software en varias partes de la red, el mensaje es recibido por todos (Poll) los correspondientes procesos.

Al recibir este mensaje, transmitirán una "oferta" al usuario que les pidió ese recurso vía mensaje direccionado; acompañando la respuesta irá el nombre único del proceso. El usuario puede entonces aceptar la oferta que le sea más atractiva. Este mecanismo activa una medida del nivel de carga del sistema, pues la oferta estará reflejando el costo prevaleciente de proveer los servicios necesitados; como cada almacenador periférico es un recurso de la red, la localización de un archivo creado es determinado por la

estrategia oferta oferta descrita previamente. Consecuentemente el sistema deberá ser hábil para la localización de un archivo, sin la intervención del usuario.

Los archivos del sistema DCS son un conjunto de proce sos, ya que los usuarios accederán archivos enviando mensajes a dicho sistema. El sistema de archivos responderá con la acción necesaria para arreglar los procesos remotos para ejecutar la operación requerida.

Por ejemplo, asumiremos que un archivo dado está localizado en el sitio B y el usuario requiere la lectura de ese archivo. El usuario mandará un mensaje a la red preguntando al sistema de archivos por el archivo que necesita, el sistema de archivos en el lugar A responderá vla mensaje direccionado al usuario informando el nombre de su proceso con el cual es manejado el archivo requerido. El usuario y el proceso estarán en comunicación directa para ejecutar la lectura deseada.

Notese que la localización del archivo es transparen te al usuario.

Actualmente DCS consiste de mini-computadoras SUE y dos VARIAN-620_S; cada una tiene por interface un anillo de 2.3 megabits. Hasta este momento, el procesador de coman-

dos, las rutinas de distribución de mensajes, varios paquetes de edición y una serie de rutinas de clasificación están disponibles para su uso en la red; además de la facilidad de acceder archivos por el nombre mnemónico de los mismos.

Como DCS se diseñó para estudiar el concepto de un sistema de cómputo distribuido, no hay planes para expandir las facilidades de software. Actualmente el equipo de investigadores de DCS construyen una interface que permita a terminales y a dispositivos de almacenamiento, residir directamente en el anillo (fig 1.4), ya que hasta el momento estos dispositivos se comunican a través de los cuatro procesadores.

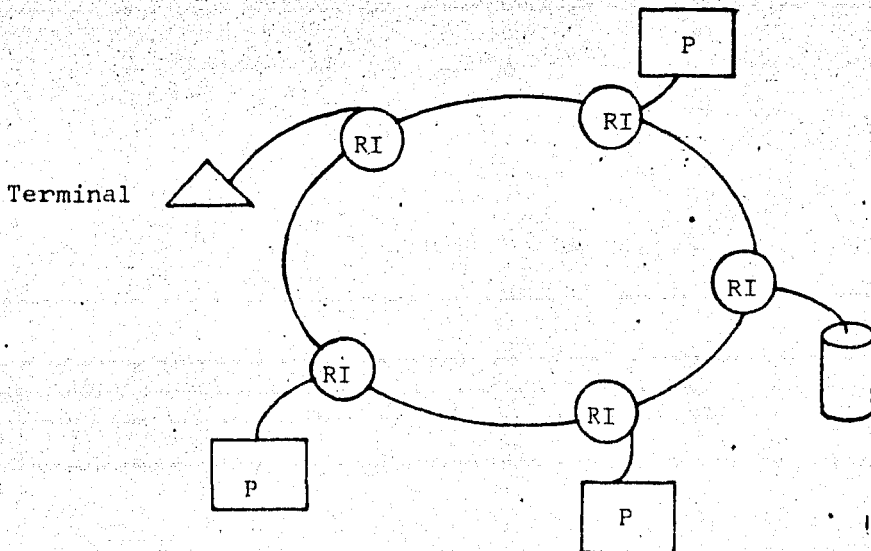


FIG 1.4 PLAN FUTURO DEL SISTEMA DE COMPUTADORAS (DCS)

El proyecto DCS intentó proporcionar una interfaz común de usuario para todos los componentes en el anillo, vía la estructura de procesos de DCS y mecanismos de comunicación. El sistema operativo de la red determina en qué lugar se encuentra el recurso necesitado, ya sea que se trate de un proceso de hardware, un archivo específico o bien una rutina especial de software. El sistema proporcionará al usuario la comunicación que necesita con el proceso pedido, todo esto a través de mecanismos uniformes de comunicación y procesos estructurados del DCS que mantengan la transparencia de operación.

Otros ejemplos de redes de este tipo son: DCN (DISTRIBUTED COMPUTER NETWORK) y parte de la red ARPA que tiene el nombre de RSEXEC (RESOURCE SHARING EXECUTIVE).

1.1.3 RED DE TERMINALES

Las características básicas de una red de terminales, son parecidas a las de la red de computadoras, porque el usuario ve al sistema como un todo, donde no necesita saber donde se encuentra por ejemplo el compilador Fortran, pero también son parecidas a las de una red de comunicación de computadoras porque en el momento que se caiga el sistema anfitrión se caerá el sistema de comunicación de datos (caso particular: red de comunicación de computadoras de topología estrella);

obviamente en este tipo de red no tenemos las facilidades ni las ventajas de una red de computadoras, ni de una de comunicación de computadoras, ya que existe en la red una sola máquina anfitriona, con sus limitaciones en hardware y software.

Es claro que una red de terminales es una sub-red de una red de computadoras.

Un ejemplo de red de terminales es la red de la Universidad (fig 1.5). Más adelante hablaremos a fondo de este tipo de red, por ser la unidad en una red de computadoras; y al ver un ejemplo de la comunicación entre una terminal y la computadora, nos dará una gran idea de lo que son las comunicaciones entre computadoras.

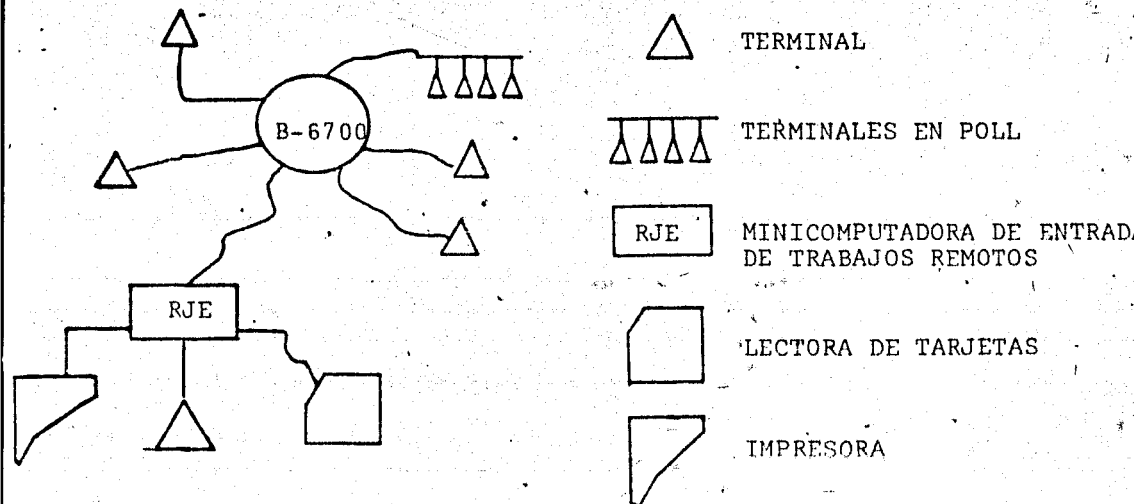


FIG. 1.5 ESQUEMA DE LA RED DE LA UNIVERSIDAD DE MEXICO

1.2 ARQUITECTURA

Aparte de las diferencias antes mencionadas existe otra que es su arquitectura.

La arquitectura consta de topología, composición, mecanismos de control, tipo de canal, estrategia a seguir, medida, tamaño, confiabilidad y transparencia.

1.2.1 TOPOLOGIA

Existen tres tipos principales que son:

- a) centralizada o estrella*
- b) distribuida*
- c) anillo, la cual es una variante de una configuración distribuida*

a) CENTRALIZADA

Es un simple sistema de tiempo compartido, cada canal de comunicación sale de la computadora central a nodos, multiplexores y concentradores (fig 1.6)

El problema principal en este tipo de red sucede cuando

la computadora central tiene una carga del sistema, la comunicación de la red es interrumpida hasta que la computadora central se habilita.

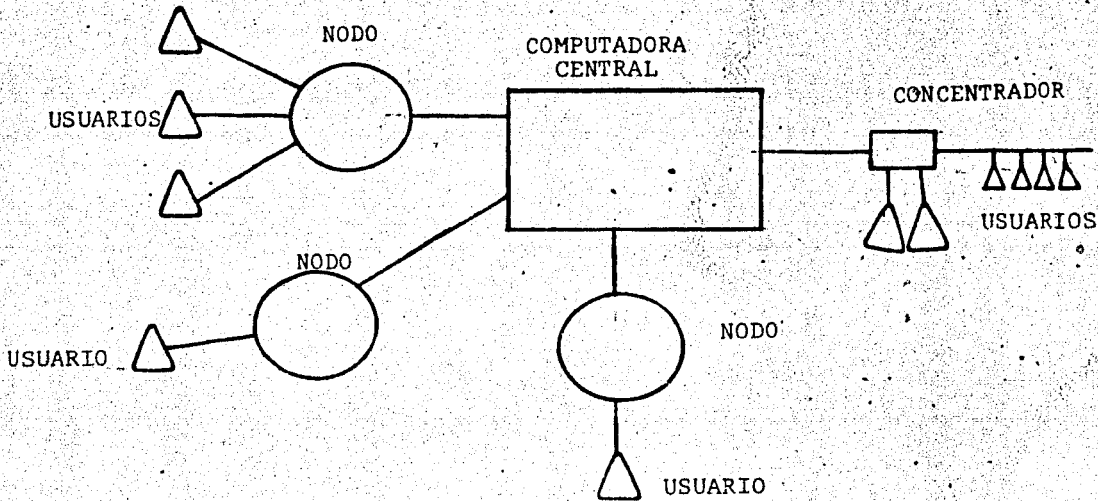


FIG. 1.6 RED CENTRALIZADA

b) DISTRIBUIDA

Las redes de tipo distribuida no tienen una computadora central, sino que todo Nodo tiene conexiones a otros nodos (fig 1.7).

Este tipo de red tiene una gran ventaja sobre las centralizadas, ya que aunque se llegara a caer el sistema de un nodo podríamos llegar a la computadora anfitriona por

otra ruta.

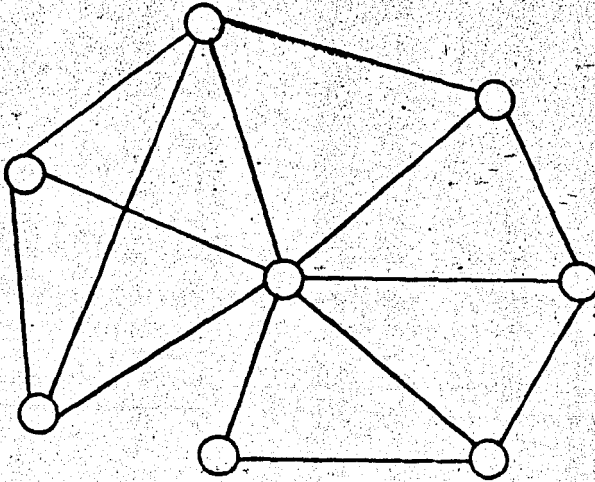


FIG. 1.7 RED DISTRIBUIDA

c) ANILLO

Un caso especial de red distribuida es la de anillo, donde todo nodo es conectado a otros dos nodos (fig 1.8)

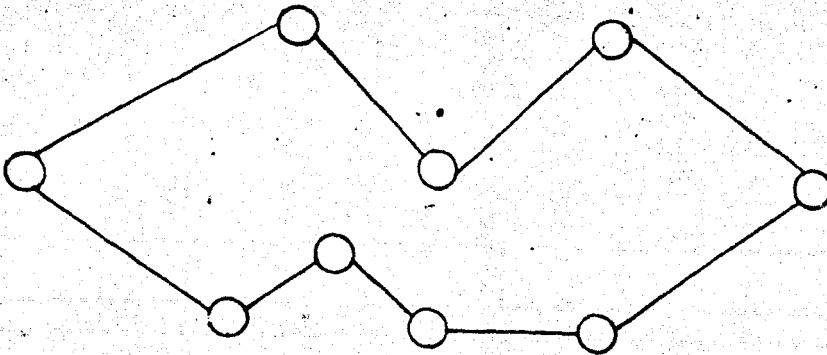


FIG. 1.8. RED DISTRIBUIDA TIPO ANILLO

La computadora central tiene una calda del sistema, la comunicación de la red es interrumpida hasta que la computadora central se habilita.

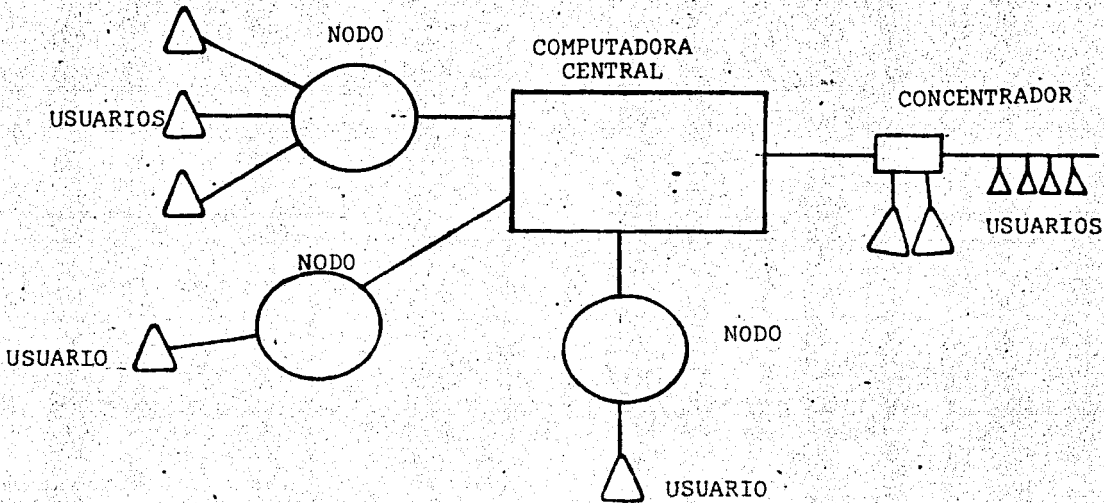


FIG. 1.6 RED CENTRALIZADA

b) DISTRIBUIDA

Las redes de tipo distribuida no tienen una computadora central, sino que todo Nodo tiene conexiones a otros nodos (fig 1.7).

Este tipo de red tiene una gran ventaja sobre las centralizadas, ya que aunque se llegara a caer el sistema de un nodo podríamos llegar a la computadora anfitriona por

otra ruta.

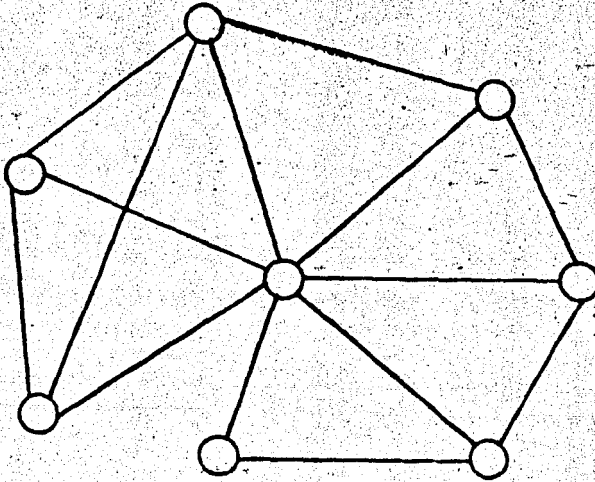


FIG. 1.7 RED DISTRIBUIDA

c) ANILLO

Un caso especial de red distribuida es la de anillo, donde todo nodo es conectado a otros dos nodos (fig 1.8)

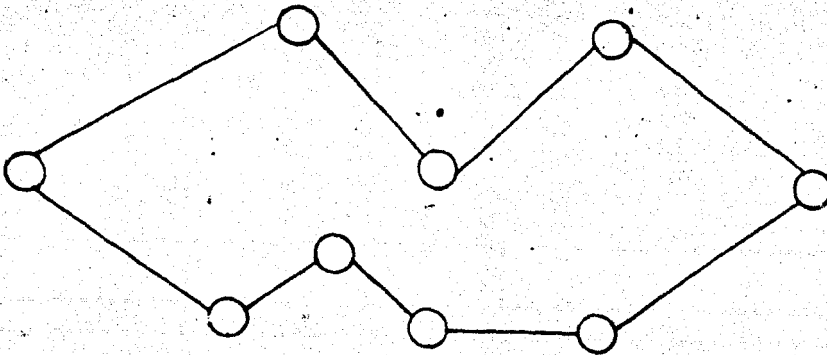


FIG. 1.8. RED DISTRIBUIDA TIPO ANILLO

1.2.2 COMPOSICION

La composición de una red puede ser heterogénea u homogénea dependiendo de la similitud de los nodos o de las computadoras unidas. Igualmente, si todos los nodos de comunicación (switching computers) son esencialmente idénticos, la red puede ser básicamente heterogénea aunque los sistemas de cómputo conectados son de diferentes tipos o manufacturas.

1.2.3 CONTROL

Las funciones de control de la red a todos los niveles, pueden distribuirse entre varios componentes de la red.

Estas funciones de control incluyen el establecimiento de la conexión inicial, control de flujo, enrutamiento, monitoreo de la red para detectar fallas e indicar la necesidad de servicios de mantenimiento, administración de la red, cobro de servicios, estadísticas, etc.

1.2.3.1 CONEXION INICIAL

Cuando un usuario que está en una terminal, quiere conexión con un programa que está en una computadora de la red, debemos responsabilizar a un componente de la red para

que establezca la conexión inicial entre los dos procesos.

Este tipo de control puede centralizarse en la red (aún cuando la topología de la red sea distribuida), de tal modo que cada proceso que desee comunicarse con otro, que esté en la red, deberá establecer una conexión de la computadora en donde está aquél con el cual desea comunicarse. Otra posibilidad es permitir las interfases con la red, de los procesos involucrados para que se responsabilicen de establecer la conexión inicial, distribuyendo entonces esta función a todos los nodos de la red.

1.2.3.2 CONTROL DE FLUJO

El tráfico de flujo entre los componentes de comunicación individual (por ejemplo computadoras de switches), es típicamente controlado por procedimientos de control de línea. Para el control de flujo entre procesos puede involucrarse software especial en las computadoras, en las que estos procesos están corriendo. Una alternativa es usar el controlador regular de software de la máquina anfitriona, en particular para controlar la comunicación interproceso. Esto se puede hacer si todas las computadoras y sistemas son similares o si los dispositivos de "front-end" efectúan una función de mapeo entre la red y el controlador de software. Una tercera alternativa es tener una computadora con

troladora que se responsabilice del control de flujo, aceptando entradas de todos los procesos en computadoras remotas que desean comunicación.

1.2.3.3 ENRUTAMIENTO

Cuando un mensaje puede atravesar más de una trayectoria en una red a partir de la fuente a su destino, el software de comunicación debe efectuar una función de enrutamiento, si la red es un circuito switchado; la conexión física está establecida en cada punto de switcheo y permanece así establecida mientras dure la conexión. El software en este caso es mínimo y puede, por ejemplo, controlar una unidad automática de llamadas.

Si se establece un circuito virtual, entonces el software de comunicación residente en algún componente o componentes de la red, debe seleccionar la trayectoria óptima para el establecimiento del circuito. Esto puede hacerse comparando las condiciones del circuito físico posible y seleccionando aquél que tenga menor tráfico y errores. En una red de switcheo de mensajes, cada mensaje puede enrutarse dinámicamente o puede subdividirse en paquetes y cada uno de ellos se enruta dinámicamente.

1.2.3.4 MONITOREO Y MANTENIMIENTO

El monitoreo y mantenimiento de una red puede hacerse por hardware y software distribuido en la red.

En la actualidad, el monitoreo es realizado generalmente por uno o varios centros de control.

Esto se hace por medio de chequeos lógicos remotos automatizados que incluyen juicios del estado de fallas y de gradación de nodos y canales.

1.2.4 MEDIDA

Es posible generar estadísticas para medir el funcionamiento interno de la red de comunicaciones, por ejemplo, longitud de colas, tamaño de mensajes y enrutamientos, a fin de conocer las fallas del sistema y corregirlas.

Tomando en cuenta el tráfico de mensajes entre nodos, este puede ser analizado para determinar la distribución de facilidades incluyendo lugar y número de computadoras anfitrionas, computadoras de switcheo, dispositivos de interface, y salidas de los circuitos de comunicación.

1.2.5 CONFIABILIDAD

La confiabilidad es uno de los aspectos más importantes de la red, ya que se debe proporcionar un servicio de comunicación sin errores y que tenga varias vías de comunicación a seguir en caso de falla en alguna de éstas.

1.2.6 TRANSPARENCIA

La red debe dar un servicio de comunicación muy eficiente en el tiempo de respuesta, para que el usuario tenga la idea que está en el lugar mismo en donde se encuentra la anfitrión.

1.2.7 TAMAÑO

El tamaño de la red estará dado generalmente por el número de nodos, computadoras o terminales que la formen.

1.3 COMPONENTES DE UNA RED

Existen dos componentes básicos en una red que son el hardware y software.

1.3.1 COMPONENTES BASICOS DE HARDWARE

En esta parte veremos el hardware de la red sin incluir las computadoras anfitrionas, ya que en otro capitulo entraremos más en detalle al hardware de estas.

LOS COMPONENTES BASICOS DE HARDWARE SON:

- 1) circuitos
- 2) terminales
- 3) adaptadores digitales y analógicos
- 4) procesadores de la red

1.3.1.1 CIRCUITOS

El circuito es una de las partes básicas para la construcción de una red.

Hay circuitos que interconectan los nodos de una red, circuitos que conectan terminales con la red, y circuitos que proveen el sendero de comunicación entre procesos.

Consideraremos primero el circuito que físicamente conecta los nodos en una red. El rango de capacidad disponible va desde 110 BPS (bits por segundo) para líneas telefóni

cas a 250,000 BPS para banda ancha (ver punto 3.3.1.6).

El segundo tipo de circuito en la red es el usado para enlazar una terminal, estación remota de entrada de trabajos, sistemas de minicomputadora, etc, con la interface de la red, posiblemente un multiplexor o un concentrador. Este circuito, parte de un canal de baja velocidad de un multiplexor o concentrador, pudiendo operar a una máxima velocidad igual al del circuito conectado al nodo.

Generalmente el circuito de alta velocidad es compartido con otros circuitos de baja velocidad.

1.3.1.2 TERMINALES

Hay varios tipos de terminales que son:

2.1) Terminales Batch

2.2) Terminales Interactivas

1.3.1.2.1 TERMINALES BATCH

Cuando hablamos de terminales BATCH estamos incluyendo subsistemas como impresoras, lectoras de tarjetas, perforadoras de tarjetas y dispositivos de cinta magnética. Esto es, terminales soportadas por batch remoto y por RJES.

1.3.1.2.2 TERMINALES INTERACTIVAS

Las terminales interactivas son generalmente equipadas con un teclado y una pequeña impresora. Opcionalmente estas incluyen un subsistema para cinta (magnética o de papel). Una pantalla puede reemplazar o complementar a la impresora.

Tal terminal soporta interactivamente aplicaciones de conversación, demanda, requerimiento y transacciones.

1.3.1.3 ADAPTADORES DIGITALES Y ANALOGICOS

Estos adaptadores están compuestos de circuitos designados para comunicación de voz, usados para la transmisión de datos.

El adaptador deberá estar entre la terminal y la línea de transmisión proviendo modulación y demulación.

El nombre MODEM es una contracción de la descripción funcional.

La función del MODEM es la transformación de la señal digital de la terminal a una señal analógica, para transmitir a la red y hacer la transformación inversa para cuando la señal viene de la red.

Estos MODEM pueden estar síncronos o asíncronos y es tar "simplex" o "duplex".

1.3.1.4 PROCESADORES DE LA RED

Los procesadores de la red que veremos son los que están fuera de la computadora anfitriona como ya lo hablamos especificado.

Los procesadores a los que nos referiremos, son:

4.1 PROCESADOR DE COMUNICACION DE DATOS

4.2 CONCENTRADORES

4.3 PROCESADOR SWITCHEADOR

1.3.1.4.1 PROCESADOR DE COMUNICACION DE DATOS

En muchas redes de comunicación, la ejecución de tareas es relativamente simple, pero altamente repetida y puede crear considerable demanda de tiempo de computadora. Muchas veces resulta más económico ejecutar estas funciones por una computadora pequeña separada llamada un procesador de "front-end".

Este procesador de comunicación de datos está orientado generalmente, como su nombre lo dice, a actuar como en-

terface entre otras computadoras o una terminal y un enlace de comunicación.

1.3.1.4.2 CONCENTRADORES

El concentrador es un DISPOSITIVO programable que tendrá funciones parecidas a un multiplexor (i.e. controlará entradas y salidas de información), además podremos hacer otro tipo de funciones, como poderle dar algunos comandos para que nos de cierta información.

El concentrador básicamente envolverá operaciones de almacenador y despachador, en donde un "BUFFER" en el concentrador, es utilizado tanto para acumular caracteres que lleguen de una terminal, hasta que haya una terminación condicional, como cuando es llamado este a una señal de fin de mensaje, Los mensajes entonces son transmitidos sobre un circuito de alta velocidad en la red.

Los procedimientos de control de errores y reensamble de mensajes pueden implementarse en el concentrador como también reconexión rápida de terminales y conversión de código.

El concentrador puede reformatear el dato en orden y transmitirlo a la interface de la red a variadas velocidades.

des y transmitir con distintos protocolos a los diferentes circuitos de terminales.

1.3.1.4.3 PROCESADOR SWITCHEADOR

Las computadoras de switcheo son también relevantes en la discusión del hardware de comunicación.

Una forma de establecer conexión de un usuario a una particular computadora, es que el usuario marque un número telefónico para hacer esta conexión, el número será diferente para cada computadora con la cual el quiere conectarse (el número puede también ser dado a una computadora con teléfono usando una unidad de llamada automática "coupler unit").

Si un usuario, no obstante, está telefoneando a un número o está conectado a una línea arrendada sin hacer caso con cuál computadora o computadoras él desea comunicación, entonces una computadora switchheadora deberá emplearse en la arquitectura de la red para que dirija la comunicación a la computadora anfitriona que este más libre.

La computadora de switcheo ejecuta la función de almacenador y despachador; cuando un mensaje es recibido, almacena solo cuando la salida de la línea está ocupada o

llena por otra comunicación; en el momento que queda libre retransmite la información que almacenó.

La técnica que se sigue es que cada mensaje contiene una identificación del dispositivo que transmite y una dirección para recibir procesos, permitiendo a la computadora de switcheo enrutar mensajes desde cualquier fuente a cualquier destino. La función de switcheo puede también ser ejecutada mapeando el puerto de entrada de la computadora de switcheo al puerto apropiado de salida.

Además de la flexibilidad inherente dentro del switcheo por software, permite por su tecnología almacenar y despachar mensajes para la utilización óptima de la línea, como también el poder mezclar mensajes dentro de procesos hacia diferentes computadoras anfitrionas y terminales en la red. Por otra parte, las computadoras de switcheo pueden ejecutar otras funciones tales como el poder concentrar terminales y servir de interface entre computadoras de servicio.

1.3.2 COMPONENTES BASICOS DE SOFTWARE

Enfocaremos esta parte al software de comunicación de la red; especialmente a los componentes en el que el software reside (al igual que en hardware el software de co

comunicación de la anfitriona. Lo mencionaremos en el capítulo que sigue).

Estos componentes son:

- 1) MINICOMPUTADORAS COMO PROCESADORES DE COMUNICACION DE DATOS
- 2) FUNCIONES DE CONTROL DE SOFTWARE

1.3.2.1 MINICOMPUTADORAS COMO PROCESADORES DE COMUNICACION DE DATOS

El utilizar minicomputadoras como procesadores de comunicación de datos, es uno de los más significativos logros en comunicación por su relativo bajo costo y su flexibilidad.

Las minicomputadoras han sido empleadas como procesadores de comunicación, facilitando la computación central y en una red de comunicaciones, permitiendo funciones de control de comunicación para la computadora central.

Como concentradores remotos, las minicomputadoras están siendo utilizadas en lugares donde existe una gran proporción de terminales que quieren entrar a la red pero son de muy baja y variada velocidad.

Este tipo de minicomputadora permite la conexión de muchas terminales que se quieren conectar a la red, con la ventaja que ella podrá estar conectada a un canal de alta velocidad y los usuarios que estén en las terminales, tendrán un buen tiempo de respuesta como si estuvieran conectados directamente a la computadora anfitriona.

El minicomputador puede controlar lectoras e impresoras de alta velocidad dando, de este modo, mayor facilidad al cómputo central.

Las minicomputadoras pueden también programarse para trabajar como computadoras de switch de mensajes en redes grandes, soportando el trabajo de conexión y ruteo de mensajes desde múltiples fuentes al destino correcto.

1.3.2.1.1 PROCESADORES DE COMUNICACION

Los procesadores de comunicación son empleados para manejar y controlar dispositivos periféricos de la computadora central y como administradores del control de comunicación entre terminales y programas dentro de la computadora central.

El procesador de comunicación es responsable de las siguientes tareas:

- i) controlar la línea de todos los circuitos de comunicación entre la computadora central y terminales remotas.
- ii) el control de la interface por medio de un canal de Memoria a la computadora central.
- iii) formateo de mensajes para convertirlos a un formato compatible con la computadora o dispositivo destino.
- iv) chequeo y corrección de errores para garantizar la transmisión.
- v) multiplexar dispositivos para asegurar el eficiente uso de canales y circuitos.
- vi) control de dispositivos, permitiendo flexibilidad en la conexión de diferentes periféricos a la central.
- vii) tener un reloj, en el cual asume toda sincronía y asincronía de las terminales.

Las mayores ventajas del procesador de comunicación de datos son las de cargar fuera del procesador de la computa

dora central trabajo y la de proveer flexibilidad en el número y la variedad de dispositivos que pueden ser conectados.

En los capítulos que siguen hablaremos más a fondo de este tipo de procesador.

1.3.2.1.2 CONCENTRADORES

En contraste con los multiplexores, el concentrador permite llevar una estadística de la manera de cómo se comparte el ancho de banda del circuito de comunicaciones, usado por cada dispositivo.

Los concentradores remotos son responsables del control de terminales, de algún dispositivo periférico, manejo de BUFFER, y de dar facilidad para el control de errores entre terminales o dispositivos y la computadora central.

1.3.2.2 PROCESOS DE CONTROL DE COMUNICACION

Los procesos para el control de comunicación son convenciones para controlar el flujo de mensajes entre procesadores de comunicaciones.

Dentro de una red, los procesos para control de comuni

cación pueden ejecutar lo siguiente:

- i) Identificación de la estación fuente y destino
- ii) checar permiso para enviar mensajes
- iii) avisar a una terminal que puede enviar
- iv) arbitrar cuando múltiples estaciones o dispositivos requieren algún canal de comunicación.
- v) proveer recuperación por error
- vi) identificar un dispositivo o un proceso
- vii) identificar un mensaje cuando múltiples mensajes están saliendo o llegando por procesos de comunicación.
- viii) atención cíclica a todas las líneas que estén conectadas.

1.4 CONTROL DE ERRORES

Para asegurar que una transmisión ocurra sin errores,

es necesario establecer un conjunto de convenciones entre enviar y recibir para detectar y corregir errores.

La forma más simple, es checar paridad en cada carácter transmitido. Esto es también llamado chequeo redundante vertical y es usado para proveer protección contra errores de bit en caracteres.

El chequeo longitudinal Redundante (LRC) es provisto para checar mensajes enteros. Este está hecho para computar un bit de paridad por cada posición de bits de todos los caracteres en el mensaje.

La forma más efectiva de verificar es el chequeo redundante cíclico (CRC) donde una división polinomial es utilizada para las cuerdas de los mensajes y el resto usado como dígito(s) de chequeo. El mismo cálculo es hecho en el dispositivo que recibe y el resultado comparado. Cuando una estación recibe un mensaje y ésta libre de error, ésta informa al que le envió que recibió bien, utilizando unos caracteres de control del Código ASCII. Este carácter es el ACK que significa que recibió bien el mensaje.

Si una estación receptora detecta un error, ésta transmitirá el carácter de control NAK que es el mensaje indicador de que el mensaje llegó con errores. Al recibir este

caracter de control, el que habla transmitido enviara de nuevo el mensaje hasta que él reciba el ACK como indicador de que su mensaje fue recibido bien.

Algunos sistemas en vez de utilizar un NAK (i.e. el acuso de recepción de mal estado del mensaje) han establecido unos "tiempo fuera" convencionales.

1.5 ENCABEZADO DE INFORMACION

Los mensajes están compuestos de datos y de control. El control de información está contenido dentro de la porción del mensaje llamado encabezado. En los encabezados de información, lo siguiente es lo típico:

- i) identificación de la estación origen
- ii) identificación del dispositivo procesador y receptor
- iii) la prioridad del mensaje
- iv) información de ruta concerniente a la distribución del mensaje una vez que ya se tiene identificado el destino.

1.6 PROTOCOLOS

Hasta ahora la discusión básica ha estado en los procedimientos elementales del control de comunicación usados para controlar un enlace de comunicación entre dos procesos.

Dentro de una gran red compleja de computadoras, es deseable construir sobre estos procedimientos básicos y formar niveles superiores de protocolos para controlar el flujo de mensajes entre procesos que están distribuidos en muchas computadoras y dispositivos que pueden estar interconectados por la red.

Un protocolo es esencialmente un conjunto de convenciones entre procesos de comunicación sobre el formato y contexto de mensajes para ser intercambiados.

Para hacer la implementación y uso más conveniente dentro de los protocolos de altos niveles de una sofisticada red, pueden usarse bajos niveles de protocolos dentro de un estrato.

Un ejemplo de protocolos en computadoras es la estructura de protocolos de la red ARPA.

La red ARPA usa una excelente estructura por estratos

que a continuación mencionaremos.

1.6.1 PROTOCOLO IMP-A-IMP

Las comunicaciones de una sub-red de la red ARPA está compuesta de procesadores de interface de mensajes (IMP_S) y circuitos de alta velocidad.

Las IMP_S son minicomputadoras que actúan como almacenador y despachador (store-and-forward) de mensajes y como procesador de comunicación de datos para computadoras anfitrionas. El nivel más bajo de protocolos en la red ARPA es el de protocolo IMP-A-IMP, que provee la comunicación relativa entre IMP_S .

Este protocolo maneja el control de flujo, detección y corrección de errores de una manera similar a los procesos básicos de control ya discutidos. Otra función básica de este protocolo es la del Ruteo de mensajes. Las rutinas de ruteo dentro de cada IMP accionan para transmitir un mensaje a lo largo de la ruta en la cual el tiempo total estimado que utilice en llegar un mensaje a su destino, sea lo mínimo.

Estas estimaciones aparecen dentro de las tablas de ruteo del IMP y son actualizadas dinámicamente de acuerdo

con retardos estimados con los vecinos IMP_S .

1.6.2 PROTOCOLOS IMP-A-ANFITRIONA

Una computadora conectada a la red ARPA que esté dando servicio de cómputo a los usuarios de la red, es llamada computadora anfitriona.

El siguiente nivel de protocolo en la red ARPA es el protocolo IMP-A-ANFITRIONA, que es provisto para la transmisión de mensajes entre anfitriona e IMP_S , combinado con el protocolo IMP-A-IMP y creando una ruta de comunicación virtual entre computadoras anfitrionas.

El protocolo IMP-A-ANFITRIONA es un conjunto de procesos que permiten que una computadora anfitriona transmita mensajes a otras anfitrionas que están en la red y además recibir información del estado de los mensajes transmitidos. Por otro lado, estas convenciones obligan que se realice un uso eficiente de la capacidad de comunicación permitida sin cerrar la salida a otras computadoras anfitrionas.

1.6.3. ANFITRIONA-A-ANFITRIONA

El siguiente nivel de protocolos es el protocolo anfitrión-a-anfitrión que permite a una anfitrión iniciar y mantener comunicación entre procesos que se están ejecutando en computadoras diferentes.

Cuando un proceso (i.e. un programa de usuario) que esté corriendo en una computadora requiera comunicación con un proceso que está corriendo en una computadora remota, hará que el supervisor local inicie y mantenga el enlace de comunicación bajo el protocolo anfitrión-a-anfitrión.

Este protocolo utiliza los niveles bajos de protocolos dentro de la implementación y mantenimiento, para iniciar el enlace de comunicación entre procesos, computadoras remotas y controla el flujo de mensajes entre estos procesos.

1.6.4 PROTOCOLOS NIVEL USUARIO

Los protocolos discutidos hasta ahora, han sido designados para el manejo de procesos. Este manejo es por medio de un conjunto de primitivas comunicaciones para la ayuda de los pormenores del sistema de operación y comunicación.

Para crear los protocolos de mayor nivel se ha desarrollado un conjunto de protocolos orientados a usuarios que están siendo extendidos. Tales protocolos incluyen:

- a) el protocolo de conexión inicial, el cual puede ser usado por todos los procesos para iniciar un enlace de comunicación con los procesos remotos;
- b) los protocolos para transferencia de archivos, que son utilizados para el traslado de archivos grandes desde una computadora a otra;
- c) protocolos de entrada remota de trabajos, utilizados para la comunicación entre dispositivos y procesos de la variedad batch remoto;
- d) protocolos "telnet". El protocolo "telnet" es el que está extensamente soportado dentro de la red, ya que define una red virtual de terminales permitiendo que todas las terminales en la red tengan una interface similar para procesos dentro de cualquier anfitrión. Esto permite la conexión de cualquier terminal, pues podrá ser soportada por la red.

1.7 APLICACIONES Y SERVICIOS

Las aplicaciones que podemos tener en una red son muy variadas y además su implementación en la mayoría de los ca

so es muy simple.

1.7.1 ACCESO

El más inmediato uso obvio de la red es para el vasto y económico acceso a recursos no disponibles localmente, tales recursos pueden incluir mayor capacidad de cómputo por la gran variedad de máquinas, software y bases de datos que hay en la red. Las redes hacen posible una gran reducción en el costo de acceso de estos recursos permitiendo al usuario participar de las facilidades de comunicación.

Tanto el uso de las líneas de baja como de alta velocidad proporcionan al usuario ahorro económico. Si una persona que quiere servicio de cómputo no tiene acceso a una computadora local, ésta persona, al tener una red, estará en posibilidad de solucionar su problema, ya que sin la necesidad de estar en el lugar donde está la computadora, el usuario podrá tener los recursos de una computadora como si estuviera en el sitio de ésta.

1.7.2 BACKUP

La red puede ser utilizada como BACKUP, ya que podemos tener copias de archivos de suma importancia en otras instalaciones, para que en caso de que la computadora local falle, se pueda seguir trabajando sin ningún problema.

1.7.3 RECURSOS COMPARTIDOS

Los beneficios de acceso y creación de BACKUP derivan de la habilidad del usuario para aprovechar mejor la red; pero existen otro tipo de recursos que tienen un alto costo para que un solo usuario los pague, tales como el desarrollo y el uso de paquetes de software. Pero si se está conectado en una red, los usuarios podrán utilizar paquetes o software desarrollado por otro usuario que esté en alguna computadora remota y obviamente los paquetes que tenga la computadora en donde estamos los podrán utilizar los usuarios que están en sitios remotos.

Un ejemplo palpable es el del CONACYT que puede utilizar grandes bases de datos que han sido creados por otros usuarios o instituciones a un costo relativamente bajo.

También se podrá compartir capacidad de cómputo, dispositivos especiales, líneas de comunicación, compiladores, sistemas operativos, programas en biblioteca, etc y utilizar el sistema de cómputo más adecuado.

Otro de los servicios es, la tele-conferencia que se refiere al uso de redes de computadoras para comunicación

personal entre grupos de personas dispersos.

2. COMUNICACION DE DATOS EN LA B-6700

En esta parte hablaremos de los elementos físicos y lógicos (software y hardware) de la red de terminales de una computadora B-6700.

Como decíamos en el capítulo uno, una red de terminales es un conjunto de dispositivos conectados a una computadora que la llamaremos anfitriona.

2.1 HARDWARE

Las partes que componen el hardware de la red son: terminales, líneas, modems, adapter y el procesador de comunicación de datos.

2.1.1 TERMINALES

El término terminal va desde un simple dispositivo capturador de datos, teletipos, pantallas de video hasta una "entrada-de-trabajos-remotos" (RJE).

2.1.2 LINEAS Y MODEMS

La conexión de una terminal remota al sistema de comunicación de datos es usualmente realizada por línea telefónica. Para distancias cortas (conexión de terminales en el mismo edificio donde se encuentra la anfitriona) generalmente se hacen conexiones directas (i.e. sin modem). Cuando la terminal está fuera de un radio mayor a trescientos metros, es usado un modem para dar confiabilidad en la comunicación (modem, dispositivo modulador y de modulador de la transmisión, ver capítulo 1).

2.1.3 ADAPTERS

El adapter es la primera parte del procesador de comunicación de datos a donde llega la información, ya que la línea telefónica o la línea directa, va conectada al adapter. En un procesador de comunicación de datos se puede tener hasta 256 adapters asociados a él. El propósito

principal del adapter, es el de tomar bits desde la línea y acumularlos para formar caracteres.

2.1.4 PROCESADOR DE COMUNICACION DE DATOS (DCP)

Cuando un caracter (ASCII, BAUDOT, etc) ha sido acumulado en el adapter, éste notificará al DCP. El DCP es un procesador pequeño que corre en paralelo al procesador del sistema B-6700. El DCP tiene la habilidad para acceder, de los adapters, la información correspondiente, a través de grupos de 16 adapters llamados clusters o adapter cluster y además tiene la capacidad de acceder a memoria principal.

El procesador de comunicación de datos es la cabeza de la red de comunicaciones. En la B-6700 es posible conectar hasta cuatro de estos procesadores a cada multiplexor (procesador de entrada y salida).

El DCP es programable y contiene los registros y circuitos necesarios para llevar a cabo las funciones de comunicación como elección, identificación de códigos de control, delimitadores y todas las funciones de chequeo de línea en forma independiente al procesador central. El diseño modular de la red de comunicación de datos, permite que cada línea sea adaptada para servir virtualmente a al-

gán dispositivo de comunicación de datos. Por lo tanto, el DCP está diseñado para procesos continuos en tiempos compartidos, aplicaciones y procesos requeridos en comunicación, operaciones en tiempo real, cómputo remoto y requerimientos remotos.

El procesador de comunicación de datos es una computadora de propósito especial que contiene suficiente lógica y registros para ejecutar todas las funciones básicas asociadas con el envío y recibo de datos (fig 2.1)

El DCP accesa a la memoria principal a través de palabras de interface sobre el multiplexor; todos los DCP'S que estén sobre un solo multiplexor comparten el cable de memoria de ese multiplexor. Sólo el multiplexor o el DCP pueden utilizar este cable de memoria a un tiempo, este uso es compartido entre los demas DCP'S.

2.1.4.1 ADAPTER CLUSTER

El adapter cluster puede ser conectado a dos procesadores de comunicación de datos, cuando es deseable tener facilidad de intercambio de clusters entre los DCP'S.

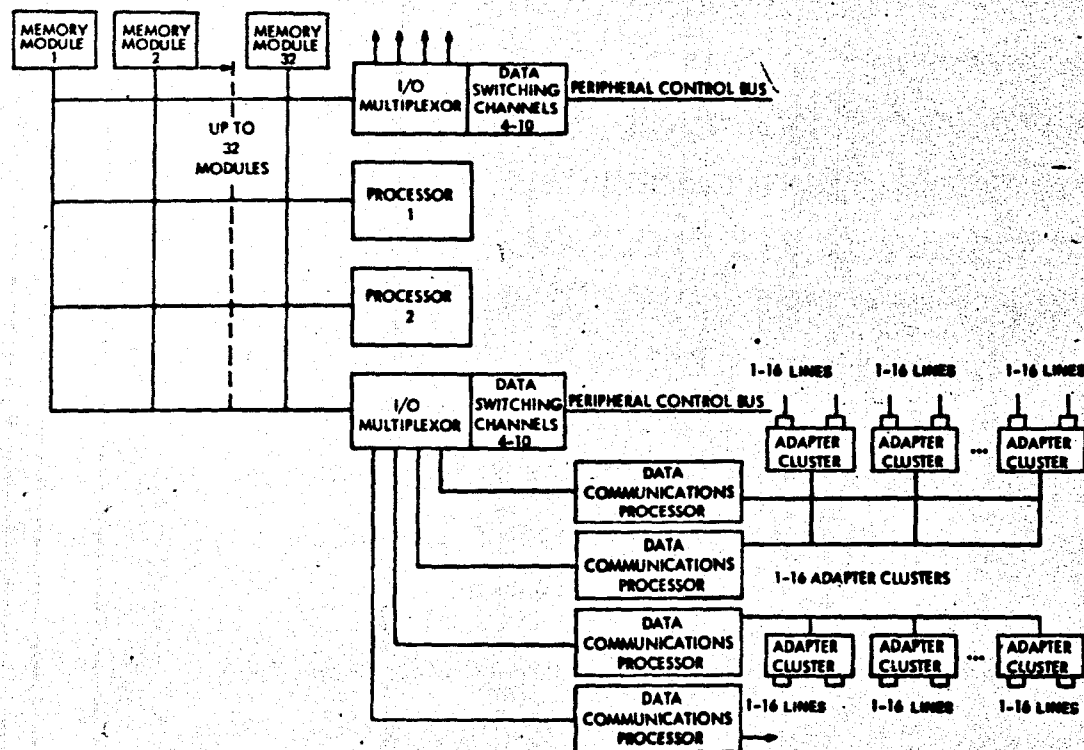


FIG 2.1 CONFIGURACION DEL SISTEMA B-6700 INCLUYENDO
COMUNICACION DE DATOS (TOMADA DEL DATACOM (39))

El DCP ejecuta instrucciones residentes en una memoria local que es opcional o dentro de la memoria principal del sistema. Estas instrucciones monitorean al adapter cluster y al procesador central para saber cuándo requieren atención. Esto origina datos, requerimientos de control, e información para ser pasada a memoria principal desde el cluster o desde la memoria principal al cluster.

Estas instrucciones son también usadas para control de información a los cluster; i.e., escritura o recepción de datos, y respuesta a entrada de llamadas. Estas técnicas permiten al DCP manejar diferentes disciplinas de control de líneas, rapidez y formato de caracteres. El DCP contiene: un pequeño arreglo de registros de intercomunicación, una unidad lógica aritmética simple, una memoria rápida (scratchpad memory) de 8 palabras y posiblemente una memoria local opcional. El DCP está diseñado para ser microprogramado y el repertorio de instrucciones consiste en gran parte de 2 y 3 instrucciones direccionadas, operando sobre caracteres de 8 bits y ejecutando en un solo reloj sencillo. La organización propia de un caracter dentro de una "half-word" (3 caracteres) permite transferencia eficiente de "half-word" a lo largo del DCP. Un diagrama de bloque del procesador de comunicación de datos está dado en la fig 2.2.

2.2 FUNCIONES DEL DCP

El procesador de comunicación de datos es el responsa
ble de mantener la interface para la comunicación en la
red. Esto incluye:

- a) almacenamiento de bits, caracteres y palabras, y
acumulación de mensajes.
- b) mantenimiento de la interface física para la lí-
nea, ya sea conexión directa o vía modem (depende
rá también del adapter).
- c) manejo de procesos estandar de la línea incluyen-
do elección (usando listas y mecanismos de priori
dad, flexible para el usuario por definiciones
dentro de la red).
- d) inserta y suprime caracteres de control y encabe-
za caracteres para dispositivos estandar.
- e) detección y toma de decisiones a la acción a se-
guir en caso de una interrupción break, wru, des-
conexión y error de paridad.
- f) pone el control en donde sea requerido, como en
fin-de-mensajes, regreso-de-espacios y suprimir-
líneas.

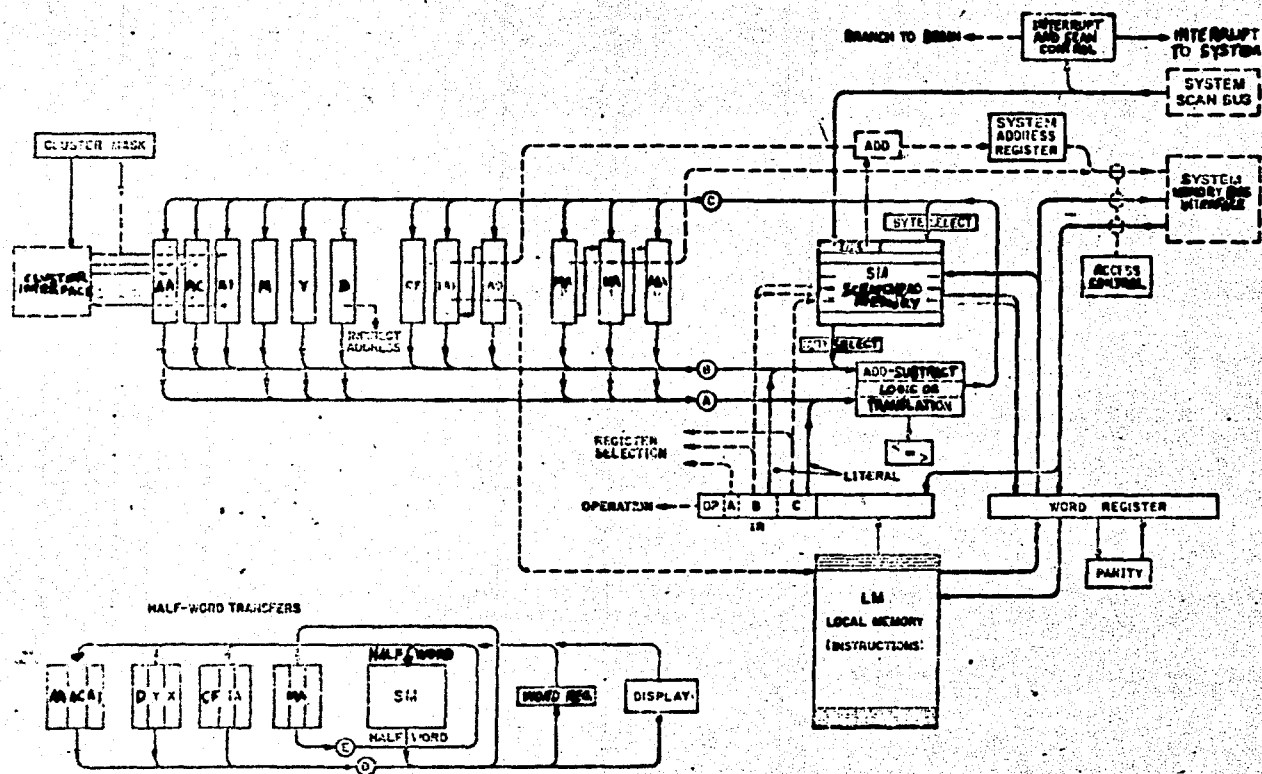


FIG 2.2 DIAGRAMA DE BLOQUE DEL DCP (TOMADA DEL DATACOM (39))

2.3 INTERFACE LOGICA CON EL SISTEMA

Como interfaz lógica para el sistema principal, está el controlador de comunicación de datos (DCC), éste presenta al DCP como una agenda. Los mensajes, que contienen los requisitos para el DCP, son ordenados por el DCC dentro de una lista enlazada llamada cola de solicitud (request). La primera entrada a esta cola es direccionada por una sola palabra la cual contiene una llave que indica qué cola solicita.

Uno u otro procesador (el del sistema principal o el del DCP) están listos para modificar la cola; la primera llave es para prevenir conflictos de actualización.

El DCP describe el resultado al DCC a través de la lista enlazada llamada cola de resultados (Result). Esta cola es usualmente una por cada requeridor (i.e. una por cada estación) pero no necesariamente. La memoria para almacenamiento de entrada para tablas y colas, es arreglada dinámicamente. El DCP localiza áreas de espacios disponibles y da una lista de ellos para los propósitos del DCC.

2.4 INTERFACE CON LA RED DE COMUNICACION

El procesador de comunicación de datos recibe caracteres de las estaciones enviadoras en varios códigos y formatos. El DCP los acumula, los chequea y los pasa al controlador de comunicación de datos (DCC) como mensajes a través de la cola de "result". El DCP interpreta y traduce código a caracteres de control, pero generalmente no interpreta el contexto de información (texto del mensaje) que es transmitido.

2.5 TABLAS DE ESTACIONES Y LINEAS DEL DCP

Cada DCP usa un conjunto de tablas de línea y tablas de estación. Las tablas contienen: información y parámetros para modificaciones, control de selección del DCP para lecturas/escrituras y direccionamiento en memoria del espacio en tablas para aparejar la estructura de la red de comunicación de datos con el DCP. Los espacios en tablas consisten de tres niveles que son: vector de línea, tablas de línea y tablas de estación.

2.5.1 VECTOR DE LINEA

El vector de línea, es un vector informador para las tablas de línea y es indexado por cada línea direccionada

(cluster/adapter). Cada palabra de dicho vector, contiene el direccionamiento absoluto en memoria de la apropiada tabla de línea, como también las siguientes banderas:

- i) bandera de sincronía/asincrónica
- ii) bandera de switcheo.- Es puesta verdadera para líneas que establecen conexión con varias terminales a través de una red telefónica; falsa para conexión directa o líneas privadas.
- iii) bandera telefónica.- Verdadera para líneas conmutadas asociadas con una unidad automática de llamadas (ACU).
- iv) bandera de conectada.- Verdadera para líneas conmutadas a una(s) estación(es) remota(s) cuando ya han sido enlazadas.
- v) auto-respuesta.- Una opción para líneas conmutadas, que el DCP responderá llamadas en forma automática.

2.5.2 TABLAS DE LINEA

Cada tabla de línea tiene dos partes que son: estados de línea que contienen información adicional y un vector de información para las tablas de estación, adheridas a esa línea.

2.5.2.1 ESTADOS DE LINEA

Los estados de línea contienen:

- i) bandera de control/normal.- Es verdadera para control y falsa para normal. El modo control es para solicitudes un tanto especiales, que son absolutos e inmediatamente inicializados por el DCP como las solicitudes del sistema B-6700, el cual controla la línea.*

Dentro del modo normal, el DCP está controlando la línea y hace colas de requerimientos para la B-6700, además inicializará el "Request" y funciones para la estación sobre la línea asignada.

- ii) bandera de ocupado.- Es verdadera si el DCP está dentro de un proceso de ejecución de alguna función para la línea.*
- iii) bandera de lista para escribir.- Es verdadera si la función de la línea está temporalmente suspendida aguardando un requerimiento de escritura del sistema, para un subsecuente bloque o segmento de texto.*
- iv) bandera lista para reconocer.- Es verdadera si la línea funcionando está temporalmente suspendi-*

da y aguardando un requerimiento, para reconocer una entrada de información de la estación, con la opción "logicalack" prendida.

- v) bandera de lista.- Es verdadera mientras no esté ocupada.
- vi) Índice de control de tipo.-

0 para líneas con una estación

1 para líneas con varias estaciones

- vii) Retardos.- Para líneas con varias estaciones, la unidad de tiempo usada para espacios de selección.
- viii) Tiempo Usado.- Para líneas con varias estaciones es el tiempo acumulado para inicio de funciones dentro del paso prevaleciente sobre la estación.
- ix) Tipo de adapter.- El tipo de código enviado del adapter al cluster/adapter.
- x) Tiempo.- Para líneas con varias estaciones, una estimación de la suma de tiempos acumulados por funciones iniciadas, dentro del paso prevaleciente sobre la estación.
- xi) Inicio de tiempo de retardo.- La suma del tiempo de retardo después de mandar al adapter un inicio de transmisión, levantando el nivel del "request-send" para el modem, y antes presentando el primer byte de transmisión en la línea.

- xii) Inicio de retardos de recepción.- La suma del tiempo de retardos después de mandar al adapter la terminación de transmisión, bajando el nivel del "Request-send" al modem y antes mandando al adapter, empezar a acumular bytes en la línea.
- xiii) Tiempos fuera.- Valor de omisión de la línea cuando no hay recepción de bytes.
- xiv) Dirección de la unidad automática.- Para líneas de teléfonos conmutados el cluster/adapter tiene la dirección de la unidad automática de llamadas.
- xv) Estación.- El índice de la estación (relativo a la línea), para la estación con la cual la línea está siendo ocupada, lista para escribir, lista para reconocer, o lista.
- xvi) Estaciones.- El número de estaciones presentes en la línea.

2.5.2.2 VECTOR DE INFORMACION

El papel del vector de información para las tablas de estación es indexar por un número de estación relativo a la línea, cada palabra del vector de estación y la siguiente información adicional.

- i) Prioridad.- Para estaciones, la frecuencia de atención será siempre "habilitado".

- ii) Tipo.- Un índice apuntando al apropiado "request" en el programa de la red (para lectura/escritura en la estación).
- iii) Tiempo.- Variedad de prioridad (frecuencia); especificación del número de unidades de tiempo antes de iniciar un mandato de habilitación.
- iv) bandera de encolado.- Es verdadera si la cola de la estación contiene cualquier solicitud para ser iniciada.
- v) bandera de habilitada.- Es verdadera si el DCP está por iniciar una entrada lógica a la estación cuando la cola de la estación está vacía o cuando un proceso de "terminate enableinput" (instrucción en el programa de NDL) está siendo ejecutada.
- vi) bandera de no lista.- Si es verdadera entonces inicia, de cualquier estación, la función de inhibido. Una estación tendrá la bandera prendida cuando por medio del "request" detecta una terminación de error o cuando sucede una condición de error no etiquetada.
- vii) Logicalack.- Opción que causa suspensión (usualmente) de una entrada lógica cuando hay trabajo en la línea, y durará hasta que el sistema mande que puede continuar.

2.5.3 TABLAS DE LA ESTACION

Las tablas de la estación contienen:

- i) *caracteres de control.*- Los caracteres especiales de entrada, usados para distinguir los mensajes de control para el sistema.
- ii) *caracter de fin de mensaje*
- iii) *caracter de regreso de espacio*
- iv) *caracter para suprimir línea*
- v) *caracter de dirección de la estación.*
- vi) *cola de estación-consiste de direcciones absolutas en memoria del primero y el último mensaje enlazado en la lista de la cola. La cola de mensajes contiene la solicitud para ser iniciada la estación*
- vii) *Índice de espacios.*- Esto consiste en tener un índice de espacios disponibles así como también el índice del acomodo de mensajes. El índice de espacios es usado por la función de entrada cuando la estación fue habilitada.
- ix) *bandera de error.*- Si la bandera está prendida, se retendrá temporalmente la estación hasta que un espacio sea localizado.
- x) *número de transmisión*

- xi) número de secuencia e incremento
- xii) toggles
- xiii) tallys
- xiv) retransmisión.- Si la bandera se prende, se re-
tendrá la estación temporalmente el mensaje que
había transmitido
- xv) inicia transmisión.- Esto consiste en poner el
valor con el cual la retransmisión es inicializa-
da, para después ser copiada dentro del espacio
de mensajes.

2.6 CONTROL LOGICO DE LINEA

El control lógico de línea es determinado, cuando es especificamos las funciones de control, para alguna estación sobre una línea particular. Tal control lógico de línea es invocado dentro del sistema operativo del DCP en diferentes instantes. El control lógico de línea es llamado cada vez que el DCP recibe una entrada en la cola de "request" desde el DCC. Cuando un requerimiento de modo control es tomado de la cola "request", la línea implicada es marcada como en control y puesta como ocupada, el "request" es inicializado inmediatamente.

Cuando una solicitud en una línea es encontrada en modo normal, ésta es primero enlazada dentro de la propia

cola de estación. Si la cola de estación previamente contenía otra solicitud de modo normal, entonces la solicitud más reciente esperará su turno dentro de la cola de la estación. Si el estado de la estación indica "lista", entonces la estación es marcada como "lista", y ésta, esperará control de línea. Si la línea está lista, entonces el "request" es iniciado. Una excepción es el caso de un requerimiento de acuso positivo de recibo (ACK), el cual espera la línea para estar en estado normal y lista para reconocer a una estación particular. Este requerimiento de lista para reconocer, es iniciado si y solo si la línea no está switchheada al estado de control por haber encontrado error en la línea.

2.7 DESCRIPCION GENERAL DEL SOFTWARE DEL SISTEMA DE COMUNICACION DE DATOS.

Los componentes del software del sistema de comunicación de datos consiste de lo siguiente:

- 1) lenguaje definidor de la red (NDL) programa fuente
- 2) compilador NDL
- 3) controlador de comunicación de datos (DCC)
- 4) Código objeto del DCP
- 5) Sistema controlador de mensajes (MCS)
- 6) compilador DCALGOL

2.7.1 PROGRAMA FUENTE DEL LENGUAJE DEFINIDOR DE LA RED

El programa fuente del lenguaje definidor de la red (NDL) es el que describe la red de comunicación de datos (DATACOM) física, lógica y funcionalmente. Este es escrito en Algol como NDL y es usado por el compilador NDL para crear tablas de datos conteniendo características físicas y lógicas del sistema de DATACOM y un conjunto de instrucciones que envuelven procedimientos para el manejo eficiente del sistema de DATACOM. El NDL es lo primero empleado en comunicación de datos, ya que cuando el usuario ha determinado las características físicas y lógicas de este sistema, el NDL es usado para describir la naturaleza exacta de estas características. Por ejemplo, el programador de NDL deberá describir el número y tipos de dispositivos remotos que son conectados a la red, el número y tipos de líneas de transmisión involucrados, y el equipo usado para interface de la red con el procesador central. También, usando NDL, el programador describe cómo desea el sistema a operar. Por ejemplo, se describe cómo los dispositivos remotos tienen prioridad sobre otros, como también la descripción del tipo de comunicación (dentro del diálogo) con el sistema central, y qué acciones son tomadas en caso de ocurrir circunstancias anormales. El NDL es re-empleado solo cuando sea necesario hacer cambios en el sistema de DATACOM (por ejemplo, adicionar estaciones,

cambiar modelo de dispositivos, o agrandar operación de procesos).

2.7.2 COMPILADOR NDL

Mientras que el DCC y un MCS son requeridos para operar una red de comunicación de datos, el compilador de NDL es usado para generar las tablas y códigos del DCP con el cual se crea el control con el cual la red funciona.

El programa fuente de NDL deberá proveer al compilador de NDL, la información con la cual el suministrará el archivo apropiado informador de la red (NIF) y el archivo de código de la red (DCPCODE) para operar al DCP y sus subcomponentes de la red (fig 2.3). Si bien un programa de NDL puede contener hasta 11 secciones discretas (ver fig 2.4), éste funcionalmente consiste de dos piezas interdependientes de información que son: la descripción de la red y el programa del DCP.

2.7.3 CONTROLADOR DE COMUNICACION DE DATOS (DCC)

Cada estación es controlada por un MCS en todo momento, además las estaciones según su objetivo serán controladas por diferentes MCS. Pero todo movimiento físico de información entre el sistema central y el DCP está contro-

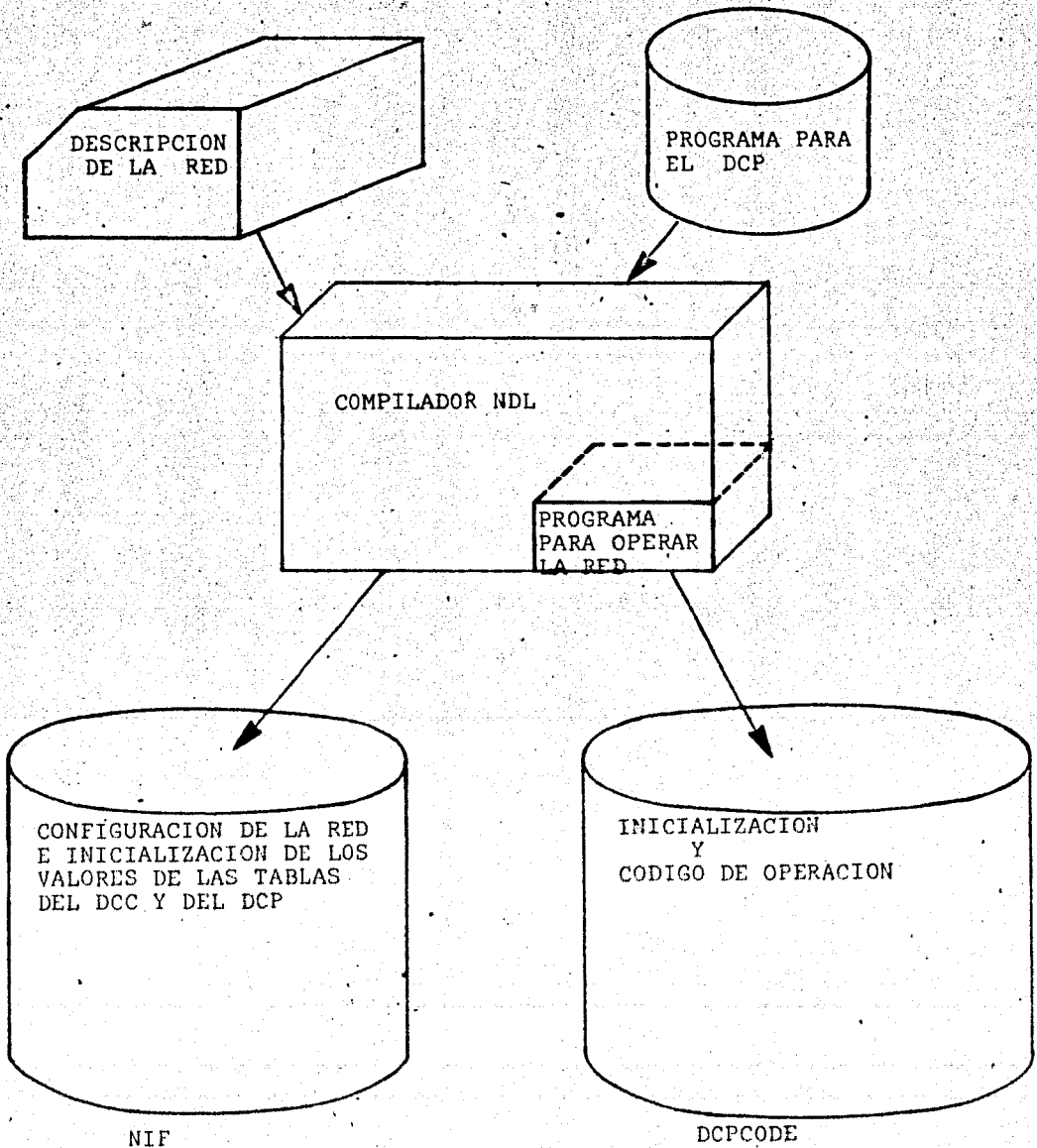


FIG. 2.3 GENERACION DEL AMBIENTE DE OPERACION DEL DCP

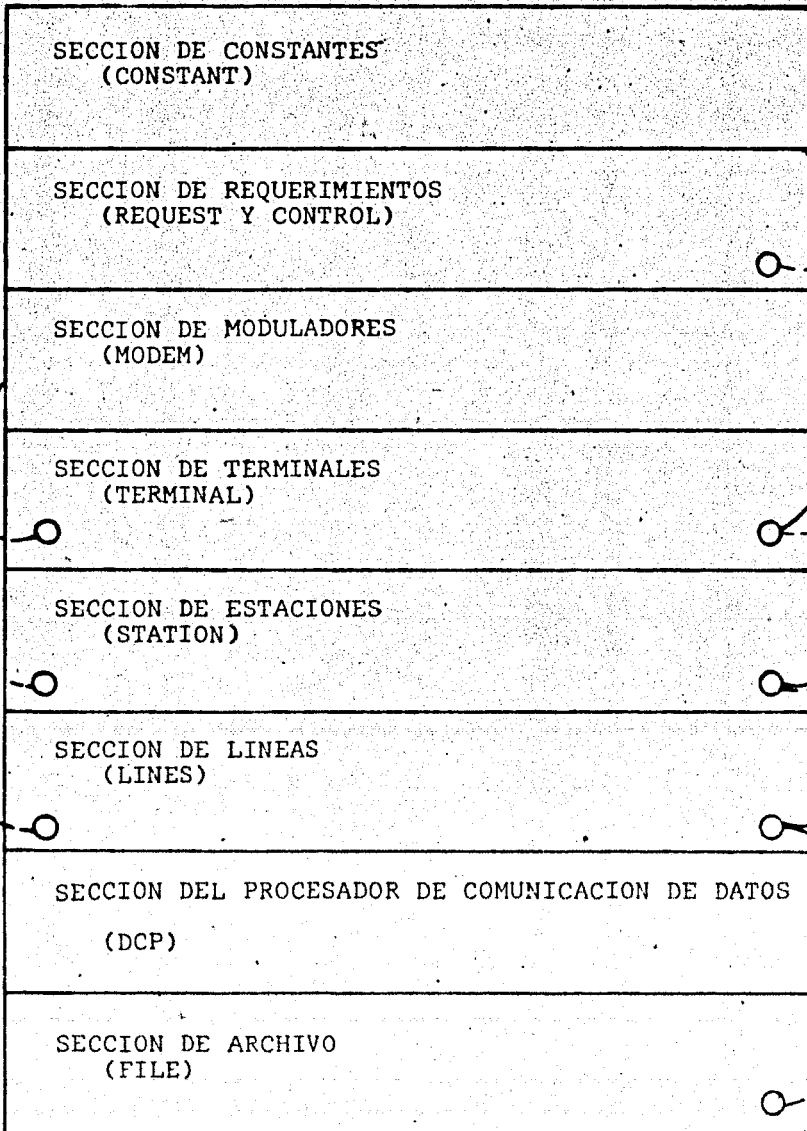


FIG. 2.4 REFERENCIAS DE INTERSECCIONES DEL PROGRAMA DE NDL.

lado por el DCC. El DCC, de esta manera, actúa como intermediario entre varios DCP_s de un lado y con varios MCS y programas de aplicación por el otro lado. El DCC es también responsable de checar la validez del paso de comandos desde los MCS_s a través del DCC a los DCP_s.

2.7.4 PROGRAMA OBJETO DEL DCP.

El programa objeto del DCP es el programa administrador que comanda la operación del DCP. Las funciones ejecutadas por este programa, que dependen del programa fuente del NDL. Algunas de estas funciones son las que siguen:

- i) multiplexeo de operaciones de línea
- ii) multiplexeo de requerimientos del sistema
- iii) proveer control de línea
- iv) proveer la distribución y colección de caracteres
- v) detección de errores (por ejemplo, paridad y formatos) y ejecución de acciones de corrección descritos en el programa fuente del NDL.

2.7.5 SISTEMA CONTROLADOR DE MENSAJES (MCS)

Un MCS es un programa escrito en un lenguaje de alto nivel (DCALGOL), que es un Algol orientado a comunicación de datos y es de propósito especial. La función primaria

de un MCS es, como su nombre lo dice, controlar el flujo de mensajes de comunicación de datos entre la terminal y el sistema principal. La información del DCP, tal como entrada de terminales y estados de información, es transmitida al MCS vía el DCP, la salida a una terminal o cambios en la red, son ejecutados por el MCS invocando unas funciones intrínsecas llamadas DCWRITE. Por lo tanto las funciones que realiza el MCS son:

- i) ejercitar control lógico sobre estaciones
- ii) ejecutar funciones de entrada de las estaciones
- iii) manejo de mensajes
 - interviniendo
 - reformateando
 - conmutando
- iv) inicialización y control de trabajos
- v) control de archivos
- vi) manejo de errores secundarios

2.7.6 COMPILADOR DCALGOL

El compilador DCALGOL está orientado para verificar sintaxis y semántica del lenguaje DCALGOL, el cual se hizo para la construcción de MCS.

B-6700

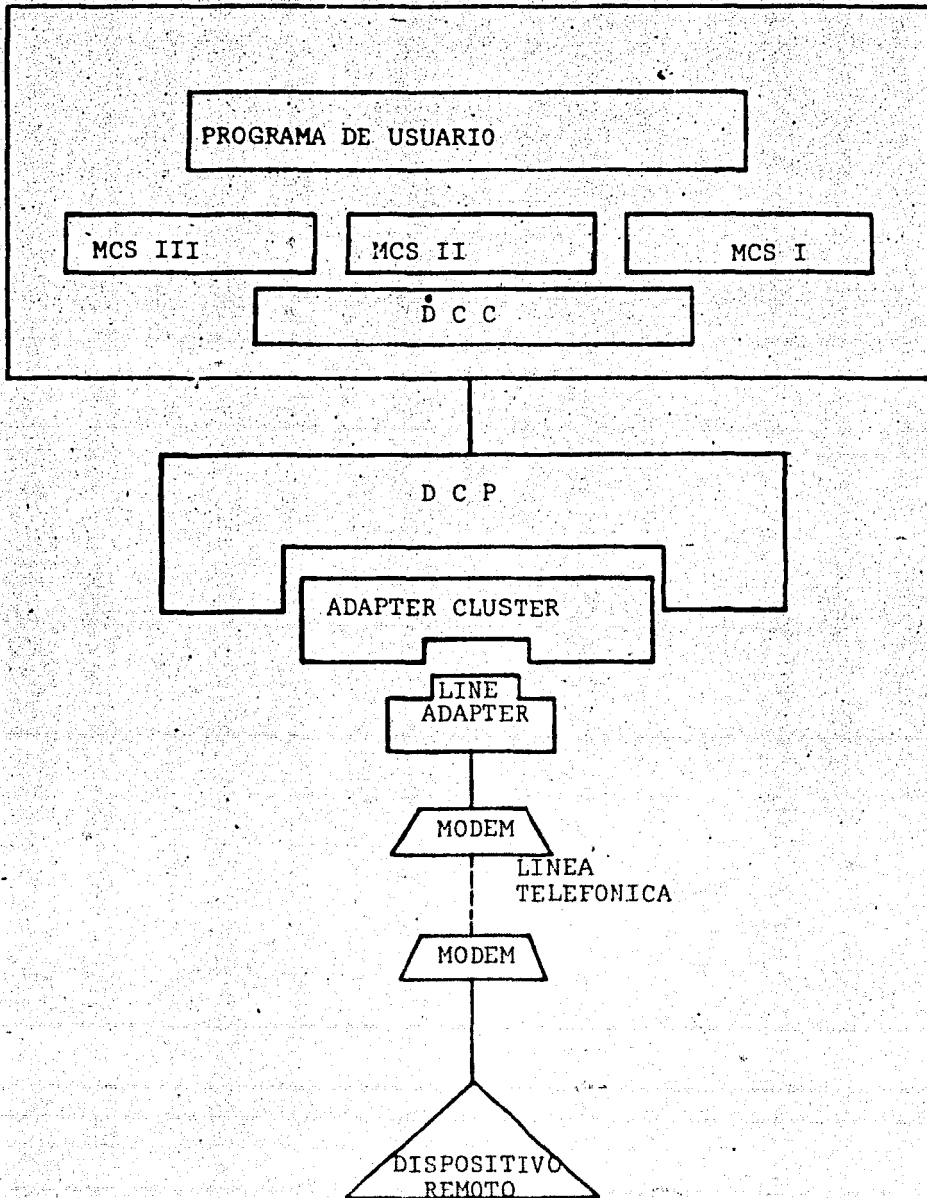


FIG. 2.5 SUBSISTEMA DE COMUNICACION DE DATOS EN LA B6700

2.8 INICIALIZACION

El proceso de inicialización del DATACOM es dado desde consola de operación cuando el operador envía al sistema el comando para inicializar los DCP_s. El archivo NIF es abierto y leído, las tablas y colas de estaciones son construídas en memoria principal.

También el archivo del DCPCODE es abierto y leído dentro de memoria principal (puede estar en memoria local). El DCC coloca los requerimientos de cluster utilizados, dentro de la cola de "request" y coloca la dirección de las colas y tablas establecidas dentro del bloque de código temporal en memoria principal. El DCC ahora ejecuta un comando, llamado SCAN-OUT/INITIALIZE para el DCP. La segunda palabra del comando SCAN-OUT contiene la primera dirección del bloque de código temporal del DCP. El DCC espera que el DCP responda al comando SCAN-OUT vía la cola de "Result"

En la respuesta al SCAN-OUT/INITIALIZE, el DCP accesa el bloque de código temporal a la memoria principal (conteniendo la inicialización del código) y empieza a ejecutar las instrucciones allí encontradas. Este código limpia las banderas de omisión, limpia el registro "mask" del cluster, y causa que el código permanente del DCP sea leído fuera de memoria principal y almacenado en la memoria local del DCP

(en caso que se haya especificado en el programa del NDL). El control es ahora transferido a la porción de código permanente llamado el "continuity loop".

La fig. 2.6 describe el sistema de DATACOM inicializado.

2.9 ENTRADA DE UNA ESTACION AL SISTEMA DE DATACOM

Como declamos anteriormente, ciertos atributos lógicos son asignados para cada estación por el programador de NDL. Por ejemplo, para que el sistema reconozca a esta estación, deberá estar el atributo "ENABLEINPUT" igual o verdadero en el programa de la red, para que sea reconocida como estación que puede meter información (fig 2.7, evento 1), los caracteres representados por bits son transmitidos uno a uno en un tiempo (series de bits). Como cada bit llega al line adapter (evento 2), éste es retenido hasta que sea posible pasarlo al adapter cluster; esto ocurrirá cuando la palabra del adapter cluster la "SCRATCHPAD MEMORY" asociada con su línea (i.e. palabra 0 para línea 1 en el ejemplo), aparezca dentro del registro asociado al "Buffer" (BAR).

Mientras el registro "SCRATCHPAD MEMORY" palabra 0 (para línea 1 del ejemplo) está dentro del BAR, el primer bit ingresado de la estación A es empacado justificándose a la derecha en el campo C1 (evento 3). Los bits restantes que

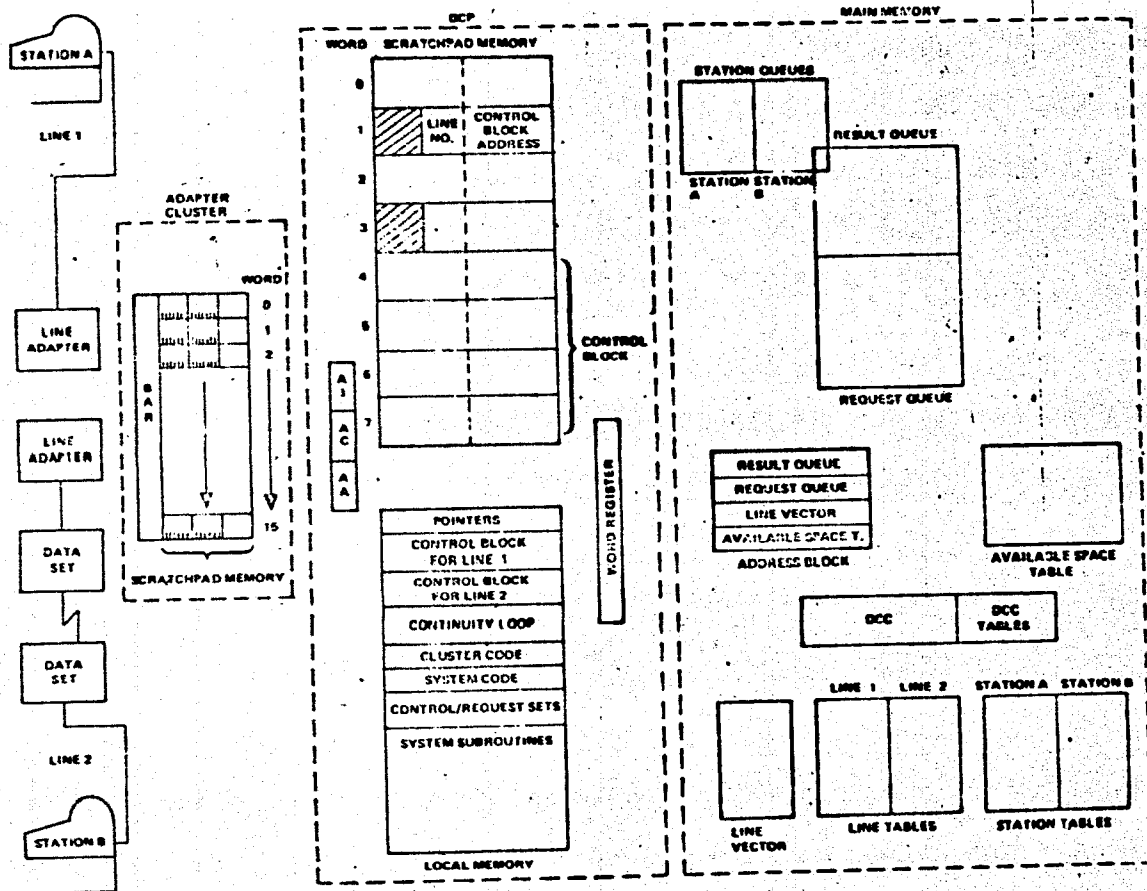


FIG 2.6 SISTEMA DE DATACOM INIZIALIDO
(TOMADA DEL DATACOM (40))

llegan son entonces acumulados dentro del campo C 1 cuando este disponible. Cuando un caracter entero es acumulado en el campo C 1, el adapter cluster mueve el caracter entero al campo C 2 (evento 4) para retenerlo hasta que el DCP esté listo, para ello esta transferencia también toma lugar cuando la palabra está dentro del BAR. Mientras tanto el adapter cluster empezará la acumulación de bits del próximo caracter dentro del campo C 1 que está vacante (evento 3). Cuando cualquier caracter del DCP brinca al campo C 2 en cualquier palabra de la memoria de "SCRATCHPAD" del adapter cluster, el adapter cluster mandará el comando de "necesita atención el cluster" (CAN) al DCP.

El "continuity loop" del DCP, en este momento, está esperando el comando de "necesita atención el sistema" (SAN) o un CAN (el SAN tiene prioridad mayor al CAN). Cuando el CAN es notado por el DCP, el DCP ejecuta una operación de lectura en el cluster; con esto pone información de control y la dirección en el registro A del DCP. La información de control es usada para indicar generalmente, la naturaleza de la entrada del caracter (por ejemplo, texto o control) y la información de la dirección son usadas por el DCP para acceder el propio bloque de control en la memoria local del DCP. Este bloque de control (para la línea 1 del ejemplo) es entonces copiado desde la memoria local dentro de las palabras 4 a la 7 de la memoria "SCRATCHPAD" del DCP (evento 5)

Una porción de este bloque de control contiene la dirección de la porción del código objeto del DCP que ha sido especificado (en el programa de NDL) como el "Request" para esta línea. Este procedimiento para el ejemplo, envuelve apuntadores a caracteres de control y copia textual de caracteres dentro de la palabra del Buffer (palabras 6 de la memoria "SCRATCHPAD", evento 6).

En el tiempo que el primer caracter es almacenado en el bloque de control, el DCP accesa memoria principal y desenlaza un mensaje del fondo común de las áreas aprovechables de mensajes. La dirección de esta área, más un índice dentro del primer espacio de la palabra disponible, es copiado dentro del bloque de control.

Cuando los 6 primeros caracteres completos de la palabra textual son almacenados del adapter cluster de la estación A, dentro del bloque de control apropiado, la palabra entera es transferida, vía la palabra del registro del DCP, dentro del primer espacio de palabra del área de mensaje obtenido para esta palabra (evento 7).

El índice de la palabra es entonces incrementado en uno, en preparación para la transferencia de la próxima palabra. Cuando el mensaje es completado (i.e. cuando la últi-

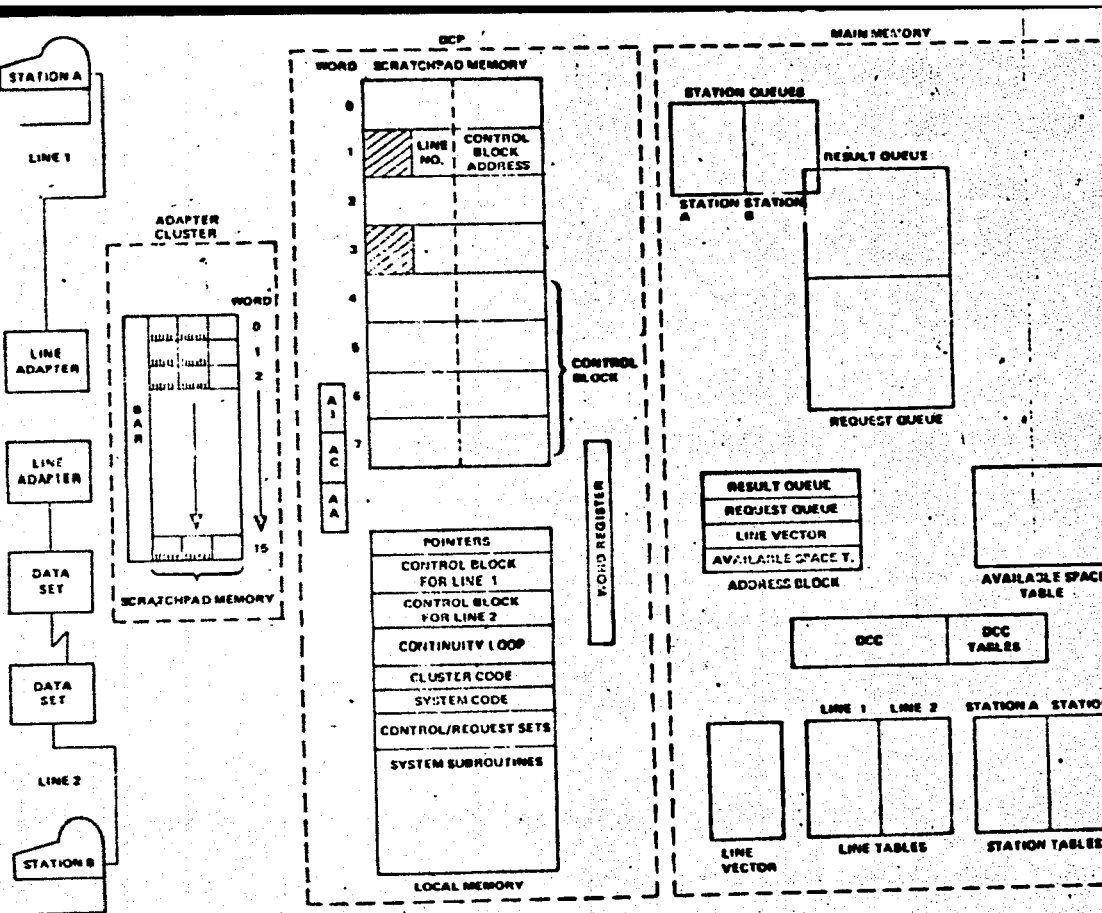
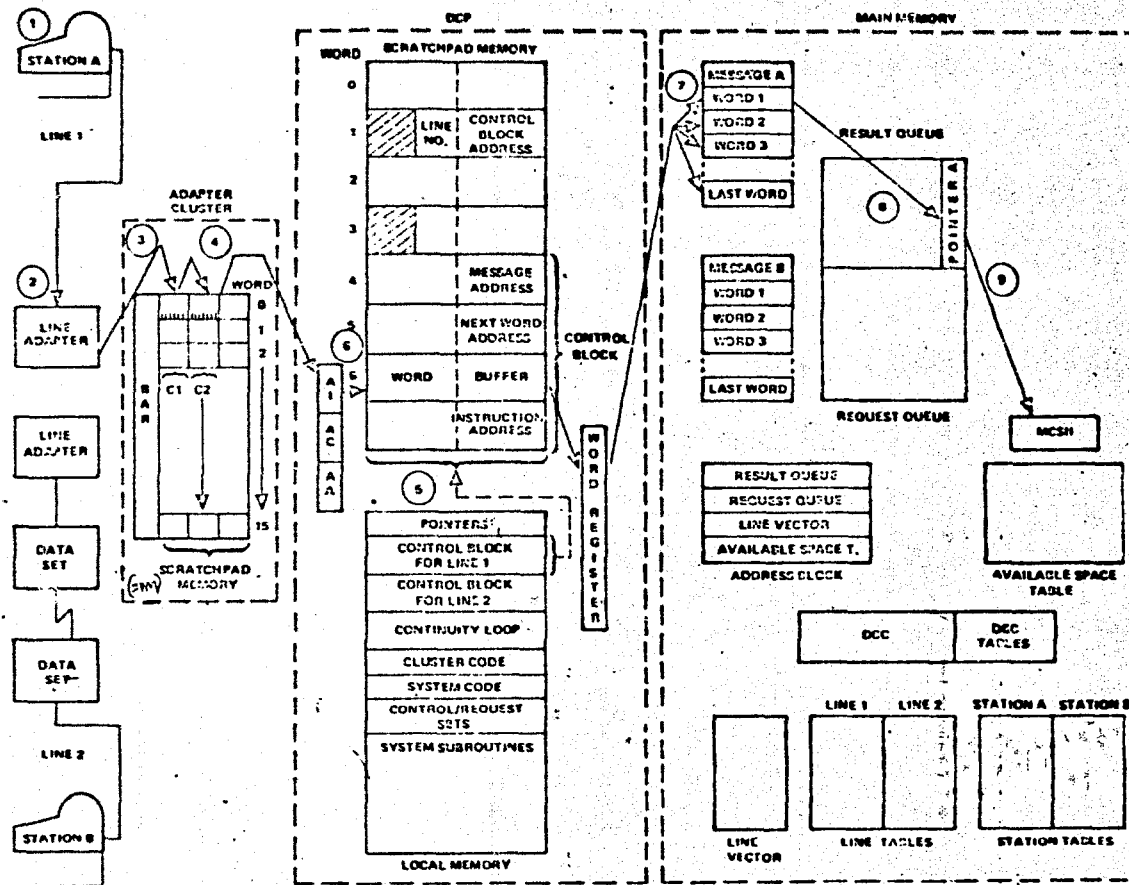


FIG 2.6 SISTEMA DE DATACOM INIZIALIDO
(TOMADA DEL DATACOM (40))



IFG 2.7 ENTRADA DE UN MENSAJE DE LA ESTACION A
(TOMADA DEL DATACOM (40))

al DCP. El "continuity loop" del DCP nota el SAN (dando es te prioridad sobre cualquier CAN) y procede a desenlazar el mensaje de la cola "Request" y enlazará ahora sobre la cola de estación correspondiente (evento 2).

El DCP ahora empieza a extraer el mensaje una palabra a un tiempo (evento 3), a la palabra "buffer" del DCP, que es la palabra 6 del bloque de control Línea 1 (evento 5). Como en la entrada, el DCP tiene el bloque de control propio disponible dentro de esta memoria de "SCRATCHPAD" (evento 4), para cuando la palabra llegue. Desde la palabra buffer, la palabra es transferida un caracter a un tiempo a la apropiada palabra de la memoria de "SCRATCHPAD" del adapter cluster (palabra cero para la estación A) cuando la palabra pase a través de BAR (evento 6). Dentro de esta dirección, el adapter cluster recibe el caracter dentro del campo C 2 y pasa al campo C 1 para distribuir bits al adapter de línea. Cuando el campo C 2 queda vacante, el adapter cluster pone una señal de CAN en el contexto "byte request" al DCP hasta que el adapter cluster pueda recibir otro caracter dentro del campo C 2 en el mismo tiempo que este distribuye un bit desde el campo C 1.

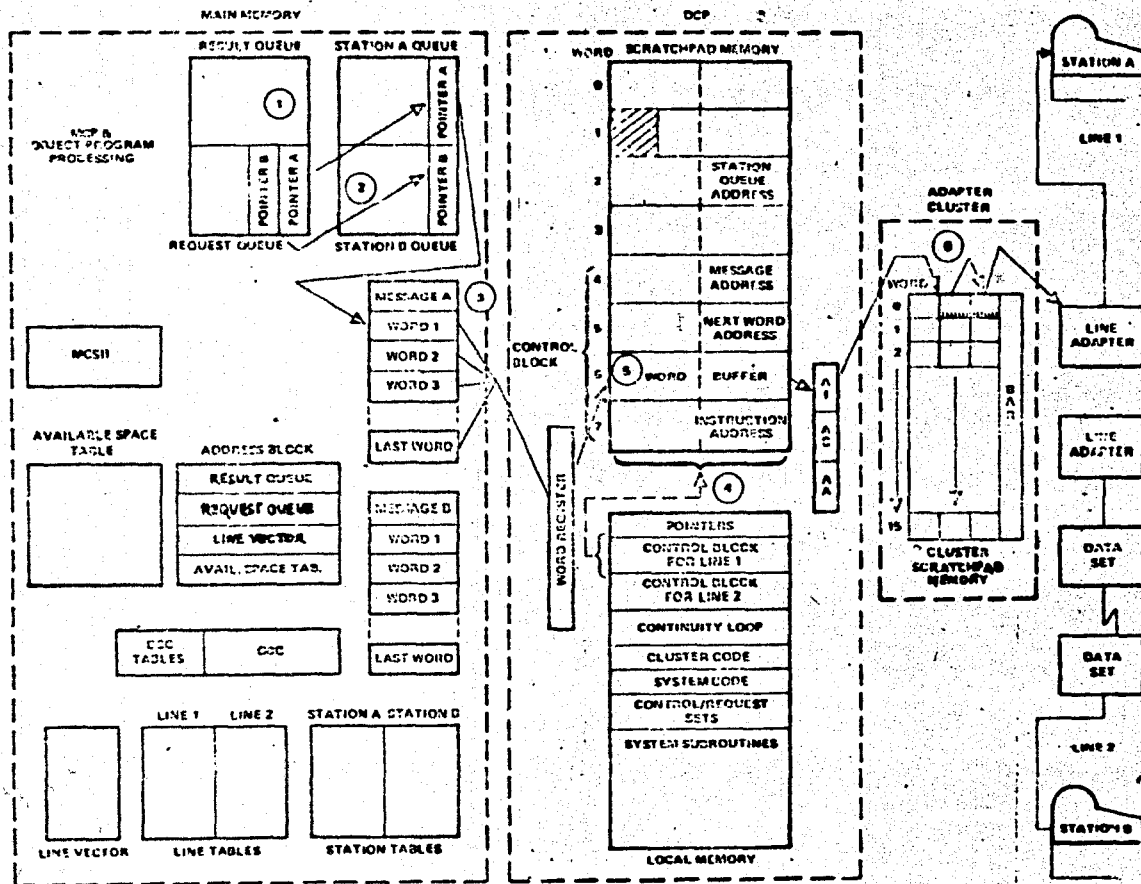


FIG 2.8 SALIDA DE UN MENSAJE PARA LA ESTACION A
(TOMADA DEL DATACOM(40))

3. PROTOCOLOS Y DISCIPLINAS

Al conjunto de reglas que gobiernan la comunicación entre dos o más dispositivos se les llama protocolo de comunicación.

Los protocolos de comunicación juegan un papel importante en redes de computadoras y terminales. Ellos definen reglas de comunicación entre todos los tipos de entidades dentro de una red; una entidad puede ser un usuario, un proceso, un programa de aplicación, un sistema de archivos.

Los protocolos también definen el control de flujo de mensajes entre las entidades para evitar saturar los

BUFFER. Una tercera función es detectar y corregir los errores que pueden ocurrir durante la transmisión de mensajes.

Los protocolos para una red compleja deberán estar en forma jerárquica, para una mejor comunicación y control.

Consideremos la ilustración de la figura (3.1); Esta consiste de dos computadoras, computadora A y computadora B, conectadas vía un canal físico de comunicación.

Asumimos que el proceso 1 (en computadora A) quiere enviar un mensaje al proceso 2 (en computadora B). ¿ En qué casos el proceso envía mensajes al extremo virtual 1? El extremo virtual 1 establece un canal virtual entre sí mismo y el extremo virtual 2, entonces divide el mensaje entre un número de paquetes y los envía uno por uno sobre el canal virtual establecido.

Los paquetes son transmitidos sobre un canal físico del extremo físico 1 al extremo físico 2, y de ahí al extremo virtual 2.

Los mensajes son reconocidos y enviados al canal virtual establecido por medio de los identificadores que llevan en una parte del contexto.

Por cada paquete recibido, el extremo virtual 2 envía de regreso un acuso-de-buen recibo (ACK) al extremo virtual 1 sobre el mismo canal.

Sólo hasta cuando este ACK llega al extremo virtual 1, éste envía el próximo paquete.

El mecanismo de canal virtual tiene dos funciones:

- i) Permitir al proceso 1 establecer un número de comunicaciones simultáneas y lógicas con el proceso 2 (una sola comunicación por canal virtual).
- ii) Controlar el flujo del paquete sobre el canal virtual y también sobre el canal físico.

En el ejemplo de arriba, nosotros podemos observar tres estratos de protocolos que son:

- i). Protocolos de canal físico
- ii) Protocolos de canal virtual
- iii) Protocolos entre procesos

NOTACION

La figura (3.2) muestra los dos lados jerárquicos de

protocolos con "I" estratos.

Cada estrato consiste de dos entidades, llamadas pro tocolos de extremo. $E_{\ell,i}$ denota el protocolo de extremo en el lado izquierdo estrato L_i y $E_{r,i}$ denota el protoco lo de extremo derecho de estrato L_i , donde $i = 1, 2, \dots, I$.

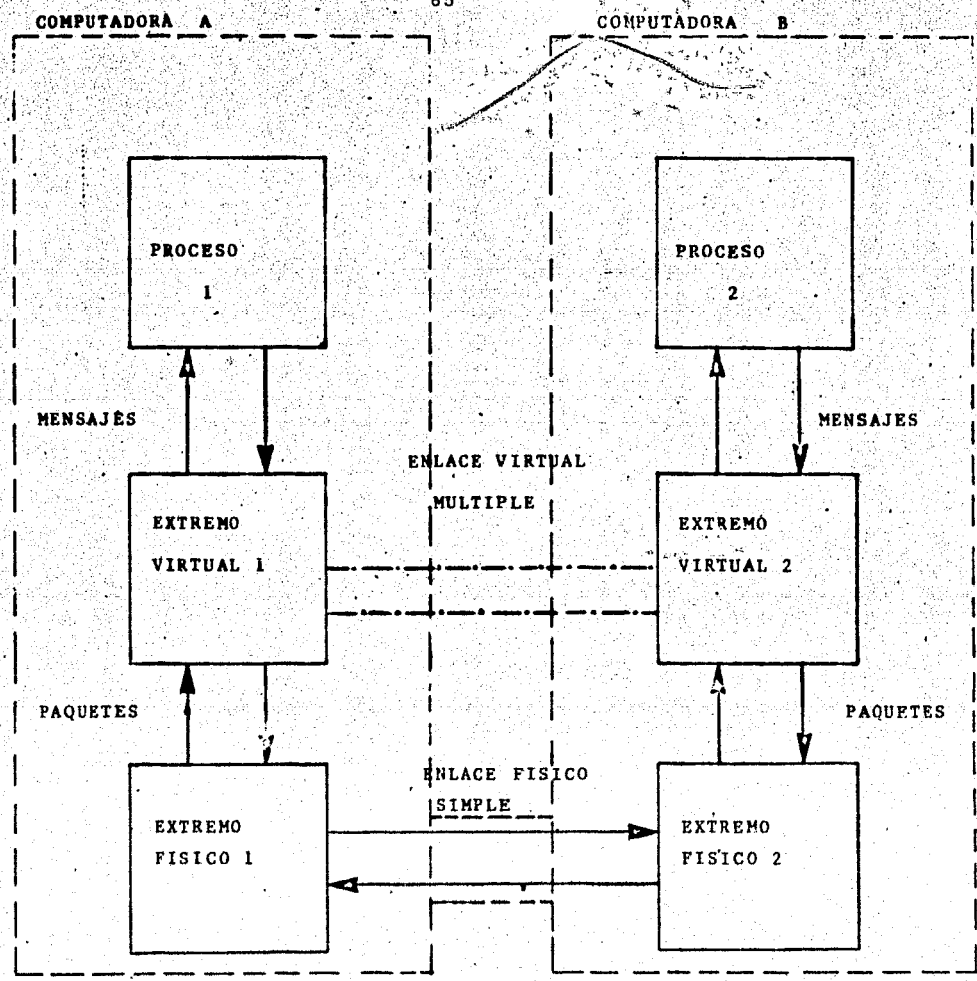


FIG 3.1 ESTRUCTURA DE UN ESTRATO TRIPLE Y JERARQUICO DE DOS LADOS

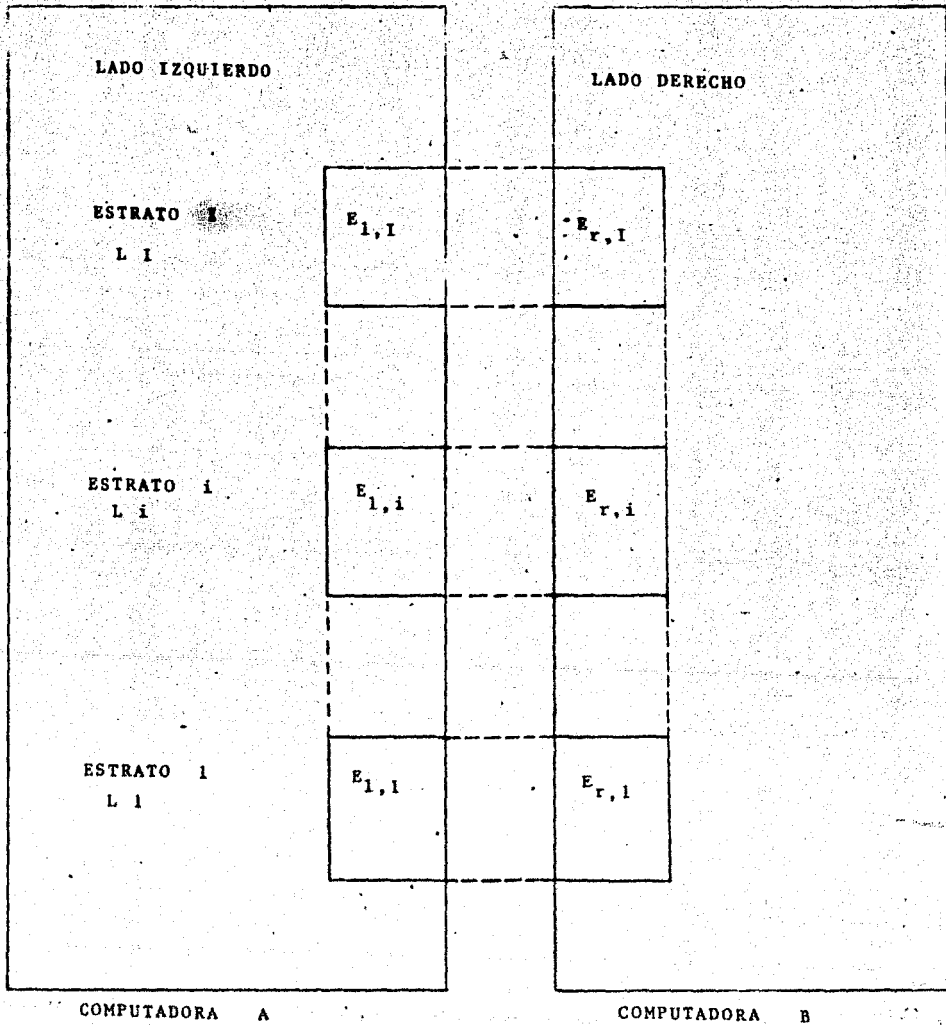


FIG 3.2 NOTACION USADA PARA REFERENCIAR LOS COMPONENTES
PROTOCOLOS JERARQUICOS

3.1 REQUERIMIENTOS DE MODULARIDAD PARA PROTOCOLOS JERARQUICOS

La modularidad es uno de los factores a ser observados mientras planeamos y diseñamos jerarquías de protocolos.

Para que un protocolo sea modular deberá tener las siguientes partes:

- i) *Que la interface entre sucesivos estratos deberá ser tan simple como sea posible; ésto sugiere dos buffer partidos, con direcciones opuestas, entre cualquiera de los dos extremos de protocolos adyacentes, como se muestra en la fig (3.3).*
- ii) *Cada protocolo de extremo consiste de un número de bloques funcionales. El número de bloques en cada extremo depende de la complejidad funcional de cada extremo.*
- iii) *La interacción entre sucesivos estratos son tiempos independientes (i, e, asíncronos). Las actividades dependientes de tiempo, tales como procesos de "TIMEOUT", son permitidos dentro de la comunicación interna, pero no lo son en comunicaciones entre-estratos. Este requerimiento es*

impuesto para simplificar los procesos de modificación de estructuras, de cualquier estrato dentro de las jerarquías de protocolos, con mínimos efectos sobre los otros estratos.

- iv) Los mensajes generados por cualquier protocolo de extremo dentro de las jerarquías de protocolos, son destinados sin cambios para el otro protocolo de extremo dentro del mismo estrato. Esto implica que la comunicación entre los dos protocolos de extremo, dentro del mismo estrato, es como si los dos extremos estuvieran conectados directamente. Todos los estratos de protocolos de niveles abajo los convierte transparente para su comunicación.

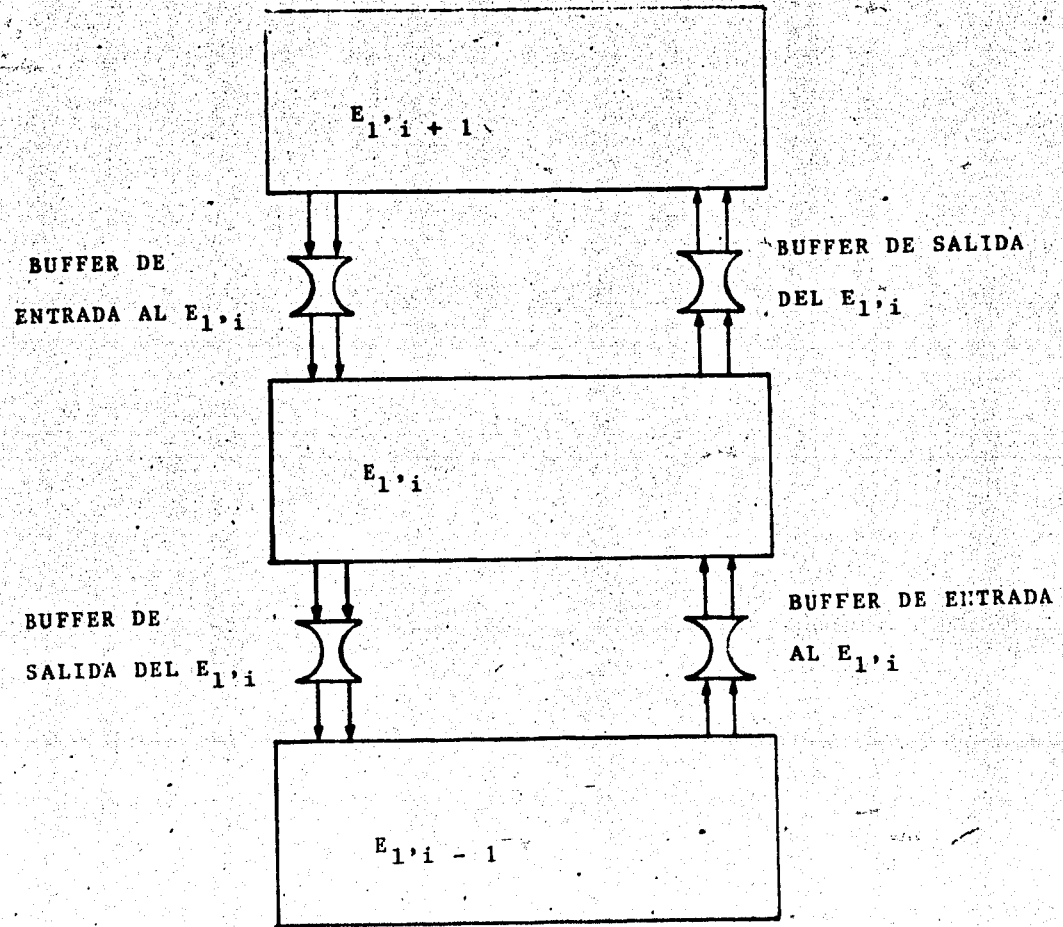


FIG 3.3 LA INTERFACE ENTRE SUCCESIVOS PROTOCOLOS DE ESTRENO

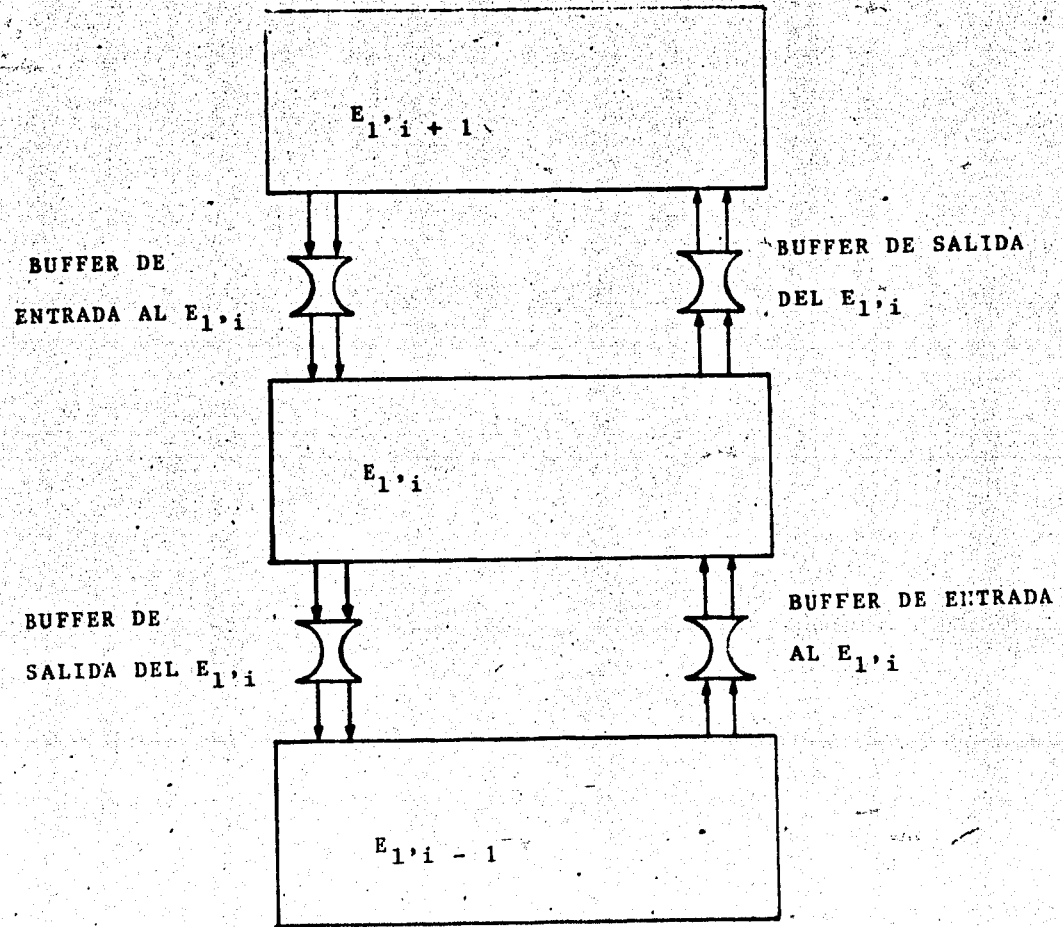


FIG 3.3 LA INTERFACE ENTRE SUCCESIVOS PROTOCOLOS DE ESTRENO

La restricción de transparencia para protocolos modulares jerárquicos es útil en la comprensión y verificación de algunos requisitos propios en las jerarquías de comunicación. En particular, lo requerido puede ser provisto (o desprovisto) basándose sólo sobre la estructura de estratos e independientemente de la estructura o comunicación de cualquier estrato (bajo o alto), en las jerarquías de protocolos, observese que, proviendo una comunicación de estratos para ser correcto, no será necesario que el estrato pueda ser apto en las jerarquías de protocolos.

3.1.1 ALGUNAS OBSERVACIONES ACERCA DE LA ESTRUCTURA DE UN ESTRATO MODULAR EN LA JERARQUIA DE PROTOCOLOS.

Consideremos el i -ésimo estrato L_i dentro de una jerarquía de protocolos; éste consiste de dos extremos que son $E_{l,i}$ y $E_{r,i}$, donde incluimos funciones de transmisión de mensajes entre los extremos de arriba $E_{l,i+1}$ y $E_{r,i+1}$. Asumiremos que los mensajes transferidos entre cualquier protocolo de extremo será estrictamente half-duplex.

Siempre que estos mensajes son transmitidos (o un conjunto de mensajes sucesivos) desde un extremo a otro, el otro extremo no podrá enviar cualquier mensaje en la otra dirección hasta que el reciba todos los mensajes transmiti-

dos a él.

Las figuras (3.4) y (3.5) muestran una propuesta fuera de línea para la estructura interna de ambos $E_{l,i}$ y $E_{r,i}$; estas figuras representan la función de bloque en cada extremo, y la dirección de la orilla representa el flujo del mensaje entre los bloques y el Buffer.

Asumimos que el mensaje viene de $E_{l,i+1}$ como aparece en el Buffer 1 y entra a $E_{l,i}$ mostrado en la figura (3.4).

Este orden contrasta con la secuencia de mensajes previamente recibidos por $E_{l,i}$, y es examinado por la verificadora de secuencia de $E_{l,i}$. Si esto está en el orden propio, éste es transferido a un inmediato Buffer 5, de otra forma éste es quitado y un mensaje de error susceptible es puesto dentro del Buffer 6, tal que el operador (o un proceso de recuperación-de-error) quede enterado del error. Además la función de verificación de secuencia es para asegurar que el mensaje recibido de un protocolo de extremo superior esté en la secuencia correcta.

Por ejemplo, si una de las funciones de $E_{l,i}$ y $E_{r,i}$ es hacer conexiones virtuales arriba (y abajo), entonces el mensaje recibido por $E_{l,i}$ es hacia abajo. Una conexión virtual deberá ser precedida por otro mensaje para lograr arri

ba la conexión. La regla de la verificación de secuencia es vital, especialmente durante el desarrollo y "debugging" de la jerarquía de protocolos.

Cuando el mensaje recibido aparece en el Buffer 5, uno de los Bloques $a, \dots, \delta b$, queda activo, dependiendo del tipo de mensaje (i.e. que tipo de mensaje es: $m_a, \dots, \delta m_b$). Sin pérdida de generalidad, asume que el mensaje recibido es de tipo m_a y también el bloque "a" queda activado. Estando activado el bloque "a" envía mensajes (o secuencias de sucesivos mensajes) al correspondiente bloque, bloque a^1 , dentro de $E_{r,i}$, mostrado en la figura (3.5), ruta Buffer 2^1 y 3^1 . Este se hace en dos direcciones por comunicación half duplex entre bloque "a" y bloque " a^1 ". Durante el curso de la comunicación, el contenido del mensaje recibido m_a , es transmitido al bloque a^1 ; y en el extremo del bloque a^1 envía m_a al protocolo de extremo superior $E_{r,i+1}$ ruta Buffer 4^1 .

Un bloque de bloques " x ", $\dots$, " y " en $E_{l,i}$, y el correspondiente bloque de bloques " x^1 ", $\dots$, y " y^1 " dentro de $E_{r,i}$ son usados para transferir mensajes en la otra dirección, i.e., de $E_{r,i+1}$ a $E_{l,i+1}$.

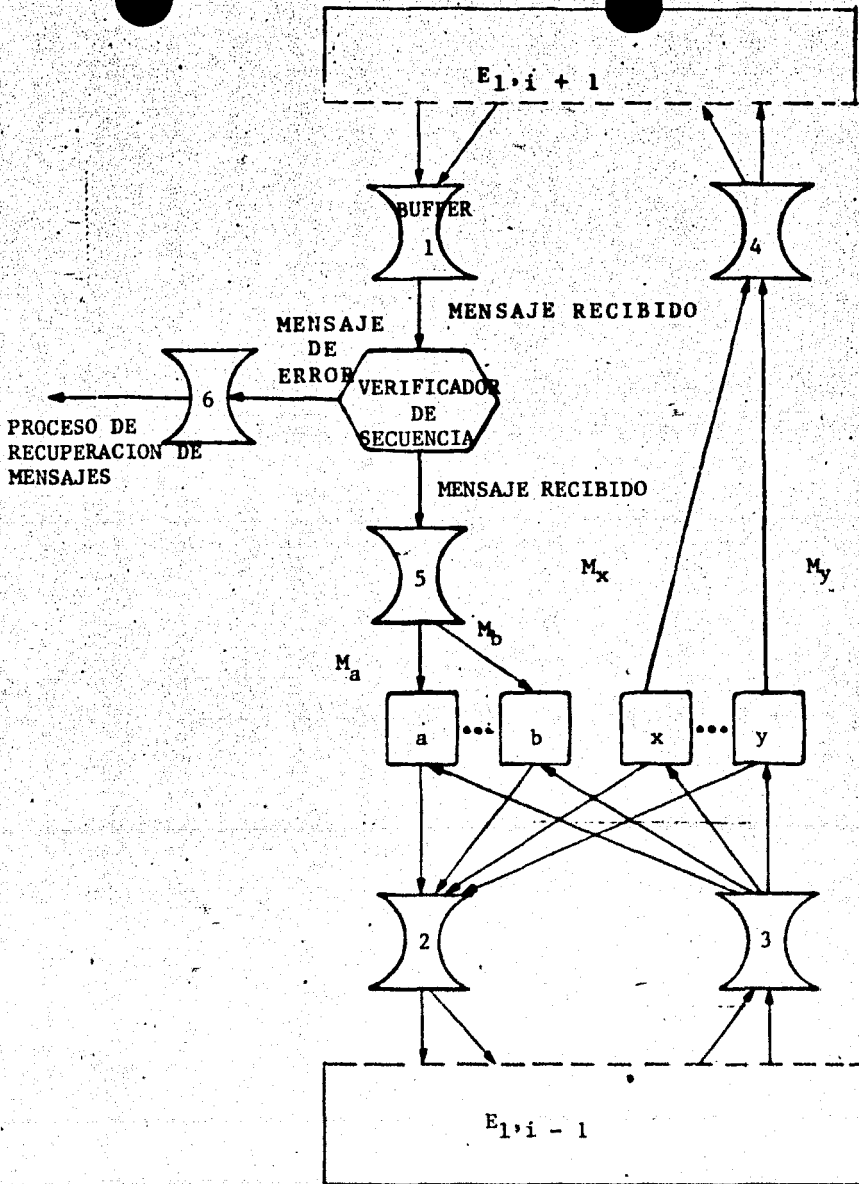


FIG 3.4 LA ESTRUCTURA PROPUESTA PARA UN PROTOCOLO DE EXTREMO $E_{1,i}$ MODULAR

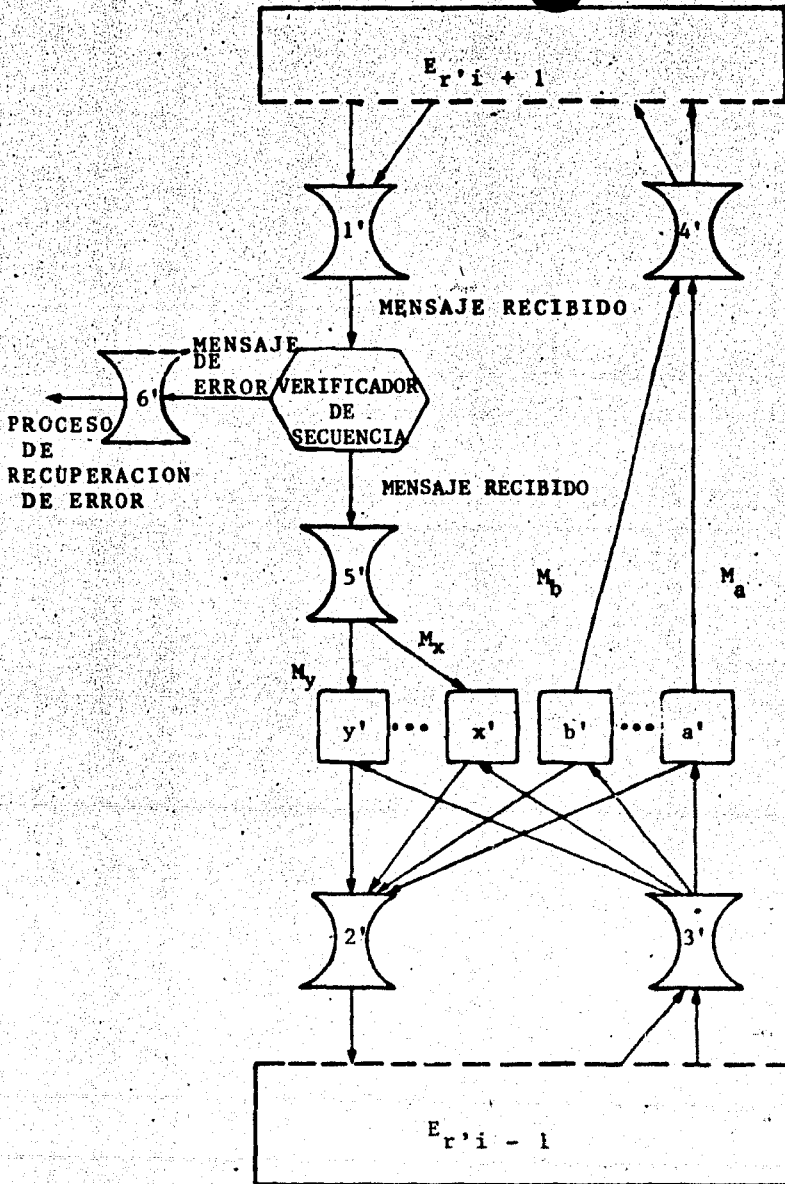


FIG 3.5 ESTRUCTURA PROPUESTA PARA UN PROTOCOLO DE EXTREMO $E_{r,i}$ MODULAR

3.1.2 ALGUNAS VENTAJAS DE LA ESTRUCTURA MODULAR

En esta parte daremos algunas ventajas obtenidas con la estructura modular mencionada antes:

- i) La estructura aumenta la flexibilidad de la estructura jerárquica y de la interacción entre sus diferentes componentes. Este aumento en flexibilidad significa evitar mayor esfuerzo y tiempo en el proceso de diseño y desarrollo de protocolos jerárquicos modulares.
- ii) Provee un camino uniforme para detectar errores de secuencia de mensajes como es tan común que ocurra. Esto es útil en pruebas y monitoreo de protocolos jerárquicos.
- iii) Por su estructura, simplifica la tarea de modificar cualquier estrato en la jerarquía de protocolos, cuando surja la necesidad de hacer cambios.

3.1.3 MODIFICACIONES EN PROTOCOLOS MODULARES JERARQUICOS

La facilidad de modificar la estructura de los protocolos propuestos, viene del hecho de que todo protocolo de extremo consiste de un número de bloques funcionales. Para modi

ficar una de las funciones de los protocolos de extremo, necesitaremos sólo modificar ese bloque sin alterar los demás de ese extremo.

Consideremos el protocolo de extremo $E_{l,i}$ mostrado en la figura (3.4). Existen cuatro tipos de modificaciones que pueden necesitarse:

- i) Cambiar la respuesta de un extremo a uno de los mensajes aceptables, tal como m_a .
- ii) Hacer que el extremo rechace un tipo de mensaje aceptable y que mande m_a .
- iii) Hacer que el extremo acepte y responda a un nuevo tipo de mensajes, y envíe m_c , si llegan en el orden adecuado.
- iv) Modificar la secuencia con la cual, el extremo acepta el mensaje del protocolo de extremo superior.

Para la primera modificación, tenemos sólo que cambiar el bloque "a", dentro de $E_{l,i}$ y el correspondiente bloque "a¹" dentro de $E_{r,i}$. Para el segundo tipo de cambio, el bloque "a" dentro de $E_{l,i}$ y el bloque "a¹" dentro de $E_{r,i}$ son quitados. Para adicionar el tipo de mensaje m_c al

conjunto de mensajes de tipo aceptable a la máquina, el bloque "c" es adicionado a $E_{1,i}$ como se muestra en la fig (3.6), un correspondiente bloque "c¹" es adicionado a $E_{r,i}$ y el verificador de secuencia en ambos extremos son cambiados.

Para el último tipo de modificación, sólo el verificador de secuencia es cambiado.

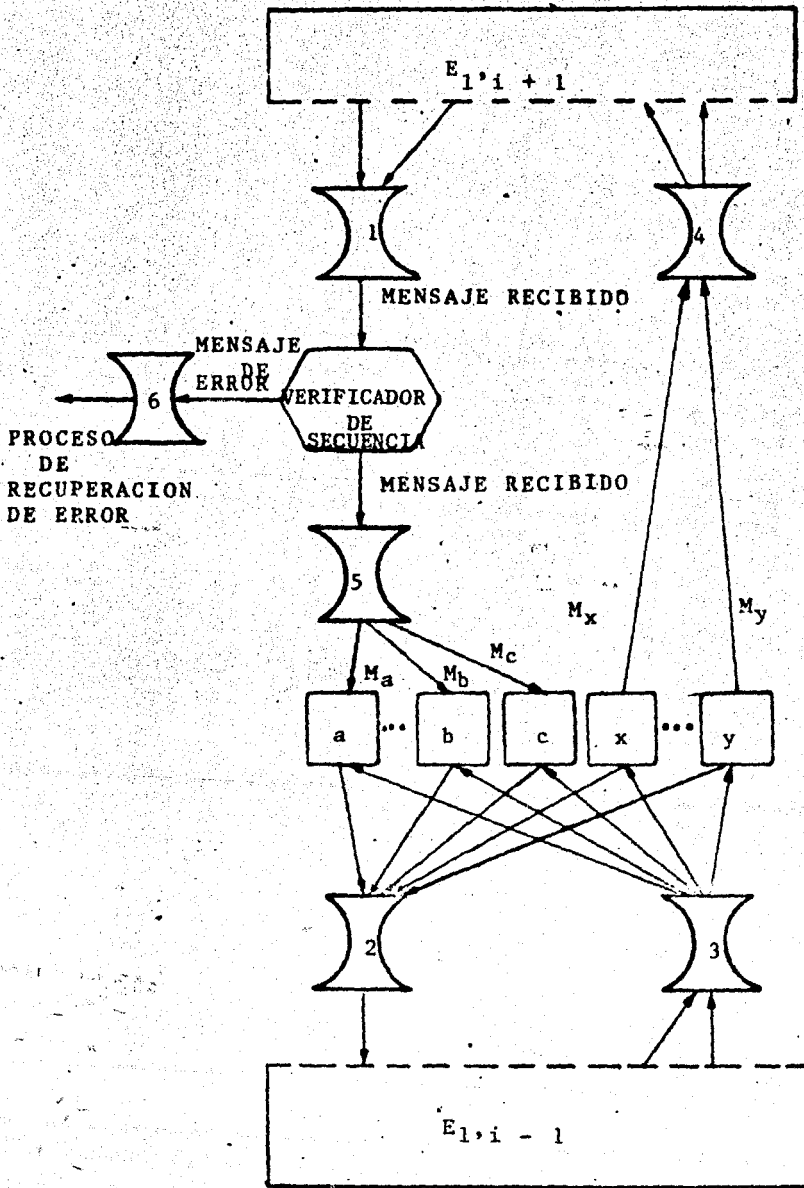


FIG 3.6 LA ESTRUCTURA DE $E_{1,i}$ DESUPUES DE LA MODIFICACION, PARA ACEPTAR EL TIPO DE MENSAJE m_c .

3.2 VERIFICACION Y CORRECCION

Hay dos propiedades que deberán cumplir todos los protocolos de extremo dentro de la jerarquía modular, y son:

- i) Cada protocolo de extremo es compatible con el protocolo del otro extremo dentro de su estrato.
- ii) Cada protocolo de extremo es conectable con el protocolo de extremo de abajo, dentro de su jerarquía.

Por compatibilidad de dos protocolos de extremo dentro del mismo estrato, cada bloque de mensaje procesado dentro de cada uno de ellos es compatible con el correspondiente bloque de mensajes procesado dentro del otro.

Por ejemplo, para mostrar que $E_{l,i}$ y $E_{r,i}$, en las figs (3.4) y (3.5), son compatibles, necesitamos mostrar que el bloque "a" es compatible con el bloque " a^1 ", ..., el bloque "b" es compatible con el bloque " b^1 ", el bloque "x" es compatible con el bloque " x^1 ", ..., y el bloque "y" es compatible con el bloque " y^1 ".

Aparentemente dos bloques son compatibles, si nunca alcanzan una situación de bloque muerto en donde (por ejemplo) cada uno esté esperando recibir un mensaje del

otro, de modo que la comunicación se quede a la mitad. Para esto se tendrá que desarrollar algoritmos, para probar que un par de bloques son compatibles y para diseñar una pareja compatible para un bloque dado.

Un protocolo de extremo $E_{1,i}$ es conectable a el protocolo de extremo abajo $E_{1,i-1}$ dentro de su jerarquía, si las siguientes dos condiciones son satisfechas:

- i) Todos los tipos de mensajes enviados por $E_{1,i}$ son aceptados por $E_{1,i-1}$.
- ii) Que no haya un camino por el cual $E_{1,i}$ pueda enviar una secuencia de mensajes que no sea aceptable al verificador de secuencia de $E_{1,i-1}$.

3.3 CONFIABILIDAD Y EFICIENCIA

La confiabilidad y la eficiencia pueden ser distinguidas como dos clases principales del objetivo de ejecución. La confiabilidad de los protocolos de comunicación entre procesos, involucra pérdidas o duplicación de datos transmitidos, liberando datos en el mismo orden como fueron suministrados y la inicialización y terminación de la transferencia de datos para continuar operaciones confiables. La eficiencia en forma primaria, se ocupa del "THROUGHPUT" y de los tiempos de retardo

para la comunicación entre procesos remotos.

Esto depende de la operación de protocolos en sus mecanismos tales como: retransmisión, control de flujo, localización de buffer, secuencia y fragmentación.

Estos objetivos definen un lado del diseño de protocolos, mientras las características de los medios de comunicación definen otro.

3.3.1 CARACTERÍSTICAS DE LOS MEDIOS DE TRANSMISION

En esta parte identificaremos las características más importantes de los medios básicos de transmisión o facilidades de comunicación que el protocolo usará para prevenir un aumento del servicio.

Los medios de transmisión deberán ser apropiados para aceptar paquetes o bloques de datos desde una fuente y tener un buen desempeño para liberarlos en su destino.

Las características de los medios de transmisión que veremos son:

- 1) Retardos variables
- 2) Duplicación de paquetes

- 3) Pérdida y demanda de paquetes
- 4) Entrega de salida-de-orden
- 5) Tamaño de paquetes
- 6) Ancho de banda

3.3.1.1 RETARDOS

Los retardos de los medios de transmisión, son la cantidad total del tiempo que transcurre desde que el paquete sale de una fuente hasta su liberación en el destino. Tradicionalmente, los retardos totales son compuestos por retardos de transmisión, o el tiempo requerido para transmitir todos los bits de un paquete en una proporción nominal de los medios de transmisión, más retardos de propagación y el tiempo en tomar un bit para viajar a través del medio de transmisión al destino.

Sobre una línea física dedicada, estos retardos son relativamente constantes. Si el medio de transmisión está repartido entre varios usuarios, como es lo más usual, ellos pueden también estar provocando retardos mientras un paquete espera su turno para ser transmitido.

En las redes de switcheo, que almacenan y despachan paquetes por varios nodos, el paquete experimenta combinaciones de estos retardos en los nodos, mientras compite

por ingresar al nivel de tráfico.

La retransmisión de información entre nodos ocasionados por errores y las diferentes alternativas de rutas de los paquetes, también contribuyen en la variación de retardos experimentados en los paquetes.

Sin la especificación del tráfico total y los algoritmos de ruteo, es difícil detallar la distribución de retardos.

Una importante característica de los retardos en el diseño de protocolos, es el tiempo máximo de propagación o "Tiempo de vida del paquete" dentro de los medios de transmisión, expresada por L .

En un enlace simple de datos y en redes de anillo, el tiempo de vida de un paquete es casi constante y determinado generalmente por la longitud de la línea y por las propiedades físicas de la transmisión.

Dentro de las redes de switcheo de paquetes, con anomalías ocasionales de ruta y otros malos funcionamientos, L podrá ser de magnitudes mayores que la normal o tener bajos tiempos de propagación. Los protocolos deberán siempre ser muy cautelosos de la llegada de paquetes viejos, así como defi-

nir el tiempo máximo de vida deseable del paquete.

Un medio sugerible para llevar a cabo esto, es que un paquete se destruya a sí mismo después de un tiempo específico dentro de la red.

3.3.1.2 DUPLICACION

Un solo paquete suministrado puede ser duplicado por el medio de transmisión y dar más de una copia al destino. Normalmente el medio de transmisión remueve cualquier dupli-cación (por retransmisión interna), pero ciertas fallas de líneas o nodos en momentos críticos pueden resultar que al destino lleguen transmisiones repetidas.

3.3.1.3 PERDIDA Y DAÑOS

La detección de paquetes dañados es un problema bien estudiado en teoría. Algunos expertos en esta área, afirman que el problema de pérdidas de paquetes es por malas técnicas del manejo de código. Por otra parte, sabemos que para tener una alta confiabilidad se requiere de mucho código; esto aumenta el "encabezado" para transmitir un paquete y reduce en forma efectiva el ancho de banda. Normalmente un protocolo, descarta paquetes dañados y espera una retransmisión de ellos. Por lo tanto, cuando hay una pérdida de un

paquete, el protocolo se pondrá en espera de la retransmisión de ellos.

3.3.1.4 ORDEN

Generalmente, los paquetes llegan al destino en diferente orden al orden que fueron suministrados en el medio de transmisión. Esto pasa cuando un paquete es transmitido en una red con múltiples circuitos, de la fuente al destino y existan varias alternativas para el ruteo del paquete. Algunas redes proveen un ordenador de paquetes en el destino, implementando un servicio de secuencia fin-a-fin, con el cual facilita la comunicación de la red.

3.3.1.5 TAMAÑO DEL PAQUETE

Los medios de transmisión normalmente tienen un tamaño máximo por paquete para poder ser aceptado para su transmisión. Si un proceso quiere enviar información que es mayor al tamaño reglamentado, el protocolo fragmentará esa información en pedazos pequeños para su transmisión y reensamblará los pedazos en su debido orden en el destino. Los medios de transmisión podrán fragmentar paquetes en ciertos enlaces (particularmente cuando diferentes redes son conectadas) y obviamente también podrán reensamblar.

La optimización del tamaño del paquete, involucra consideraciones del tipo de línea de comunicación, características de errores, tráfico y ejecución de objetivos.

3.3.1.6 ANCHO DE BANDA

Los medios de transmisión aceptan paquetes en un determinado porcentaje, en base a la demanda. Aunque dentro de una red de switcheo, es usualmente la capacidad de la línea física la que marca ese porcentaje. Sin embargo, los medios de transmisión pueden "deshabilitarse" en determinado tiempo, debido a mecanismos de control cuando hay congestionamiento interno y el porcentaje de bits efectivo ofrecido es además reducido por disposición.

3.4 PROTOCOLOS

Un "acknowledgement" positivo transmitido por protocolos de comunicación entre procesos, consiste de una disciplina que envía, una disciplina que recibe, y un medio de transmisión para enviar paquetes (mensajes, cartas, textos, cuerdas de bits finitas) entre procesos.

La disciplina que envía acepta paquetes de un proceso unido a algún control de información usado por el protocolo, para llevar a cabo la comunicación, pasando el paquete

al medio de transmisión.

Normalmente, la disciplina que envía retransmitirá cada paquete en intervalos determinados por un tiempo de retransmisión, hasta que un "acuse-de-recibo-positivo" (ACK) es recibido. Entonces al proceso se le notifica que el paquete que transmitió puede ser liberado.

Otro parámetro que tendrá que chequear ese medio de transmisión es el tiempo de "abandono", determinado para dejar de retransmitir el paquete y los reportes de posibles fallas para liberar un paquete.

La disciplina receptora recibe paquetes del medio de transmisión y usa el control de información para eliminar duplicación, reensamblar u ordenar fragmentos y liberar paquetes de los procesos dentro de un orden. Consecuentemente, por cada paquete recibido, transmitirá un ACK (mensaje de acuso de recibido bien). El medio de transmisión aceptará paquetes de la disciplina enviadora y los liberará a la disciplina receptora sujeta a retardo, pérdida, duplicación, orden, tamaño y características de ancho de banda discutidos antes.

Como este tipo de protocolo provee comunicación bi-direccional entre dos procesos, una disciplina receptora y una

transmisora son requeridas; cada lado deberá ser inicializado para que la comunicación pueda empezar.

3.4.1 PROTOCOLO BASICO

En esta parte veremos el protocolo sin secuencia, fragmentación y flujo de control.

Disciplina enviadora (ver fig 3.7). Cada paquete suministrado por el proceso fuente es asignado a un único identificador. El paquete es transmitido, y una copia es retenida. En el momento que llegue un cardcter ACK se checa para ver si es válido, y los que vengan en mal estado son descartados. Cuando un ACK referenciando es recibido, la copia retenida es descartada (y el proceso fuente es notificado del proceso), si no es recibido un ACK dentro del tiempo fuera de retransmisión, la copia es de nuevo transmitida y el ciclo repetido. Si el tiempo de "abandono" está siendo exedido, la retransmisión es suspendida (y el proceso envlador notificado).

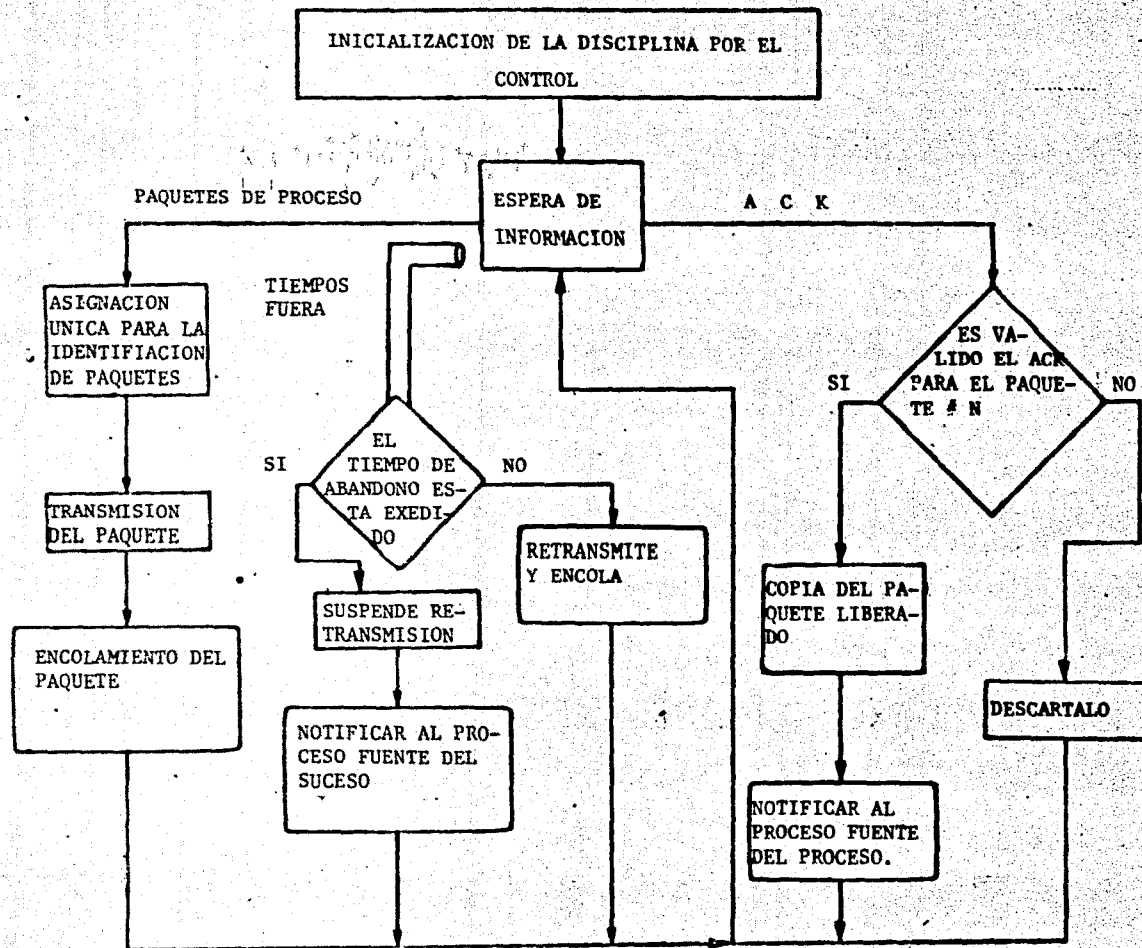


FIG 3.7 PROTOCOLO DE LA DISCIPLINA ENVIADORA

DISCIPLINA RECEPTORA (ver fig 3.8)

Cada paquete recibido del medio de transmisión es checado para ver si no tiene errores de algún tipo.

Si no está dañado, la identificación del paquete es anexado a la lista de identificadores de paquetes recibidos y un carácter ACK es transmitido al proceso fuente. Si el identificador del paquete está en la lista, el paquete es descartado por duplicación. De otra forma, el paquete está libre para el proceso.

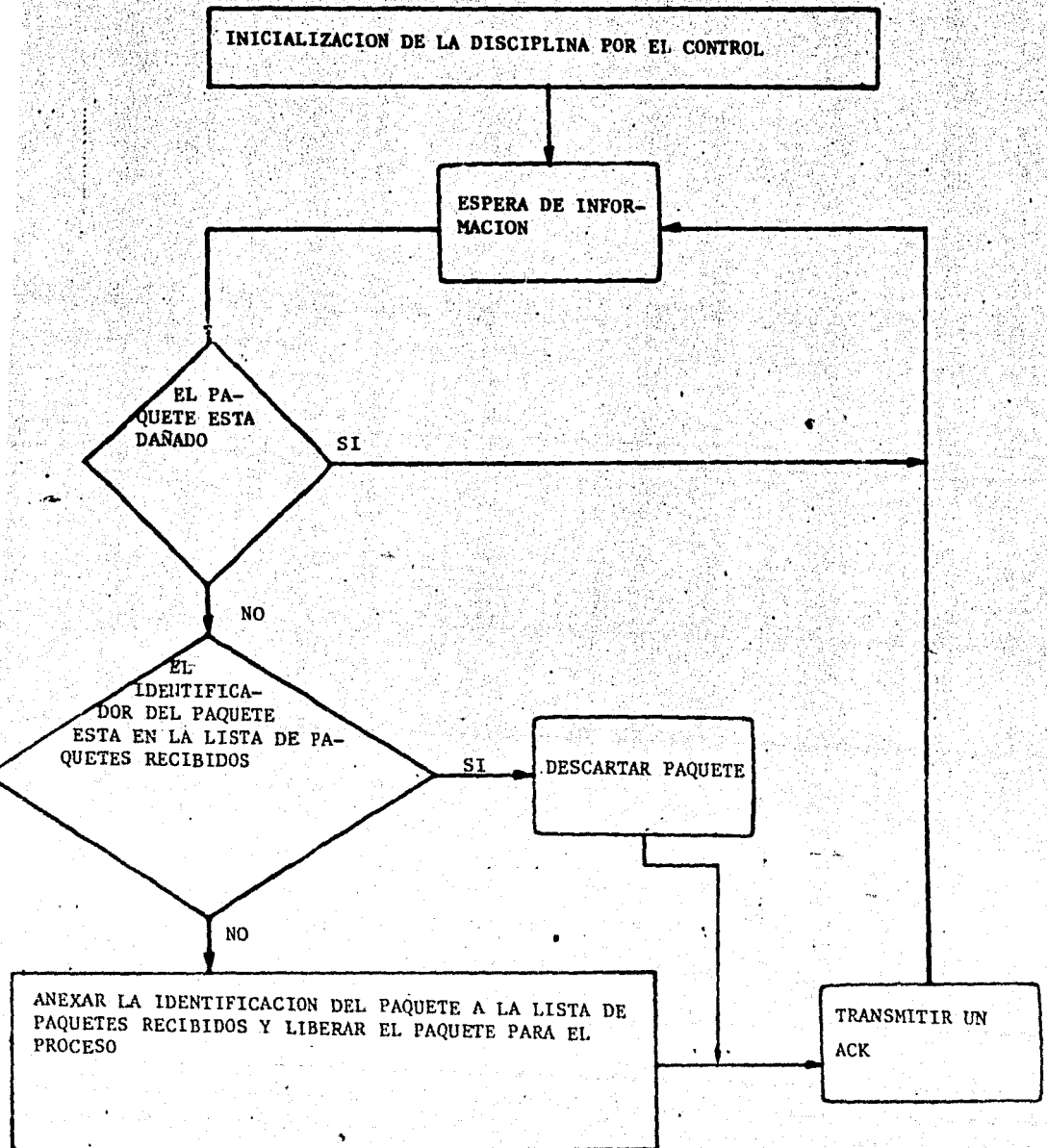


FIG 3.8 PROTOCOLO DE LA DISCIPLINA RECEPTORA

Los protocolos son inicializados cuando ambos lados tienen vacía la lista de paquetes-recibidos y no tienen paquetes a ser enviados. Sobre la definición de protocolos, asumimos que todos los paquetes dañados y acusos-de-recibo-positivo serán detectados. En realidad, no es posible detectar todos los errores de transmisión.

Sin embargo, la probabilidad de que un error no pueda ser detectado es extremadamente pequeña ya que por costo modesto, por el uso de una buena técnica que reconoce código, se puede evitar pérdidas de información.

3.4.2 CODIGO

Antes de ver algunas disciplinas con su protocolo, veremos brevemente los códigos que generalmente se usan para la comunicación de datos y el significado de los símbolos orientados a la comunicación.

CODIGO ASCII

El código de cinco bits es generalmente usado para sistemas de telex; pero para el uso de computación tiene muchas restricciones; por ejemplo, el conjunto de caracteres disponibles a la comunicación está muy limitado.

El código ASCII , dentro de la transmisión de datos, particularmente en computadoras, está muy difundido y aceptado por varias manufacturas; este código ASCII (American standard code for information interchange) está definido sobre siete bits (cuadro 3.1); algunos usuarios transmiten el código sobre ocho bits siendo el último siempre uno; muchos usuarios incorporan el bit ocho como un dígito de paridad.

BIT 4321	CONTROL CHARACTERS		DISPLAYABLE CHARACTER							ESCAPE SEQUENCE				
	000	001	010	011	100	101	110	111	010	011	100	101	110	111
0000	NUL	DLE	SP		P				PRINT		CHAR DEL	BACK TAB	F1	
0001	SOH	DC1 ENTER	!	1	A	O			TAB SET		CHAR INS ON	CURS SENSE	F2	
0010	STX	TOGGLE DC2 FORMAT	"	2	B	R			TAB CLR		CHAR INS OFF	KEYBOARD ENABLE	F3	
0011	ETX	CURSOR DC3 UP	#	3	C	S					SCROLL UP	KEYBOARD DISABLE	F4	
0100	EOT	DC4 HOME	\$	4	D	T					SCROLL DOWN		F5	
0101	ENQ	NAK	%	5	E	U					CLEAR	NEXT PAGE	EOS DEL	F6
0110	ACK OFF LINE	SYN	&	6	F	V					CURS ADDR	PREV PAGE	PRINTER ON LINE	F7
0111	BEL	ETD (XMIT)	'	7	G	W						FORMAT ON	PRINTER OFF LINE	F8
1000	BS	CAN	(	8	H	X					HOME	FORMAT OFF		
1001	HT	EM	)	9	I	Y						PROG ENTRY ON		
1010	LF	SUB	.	:	J	Z					ERM	PROG ENTRY OFF	ON LINE	
1011	VT	ESC	+	:	K	[					EOL	END Protect	OFF LINE	
1100	FF CLEAR	CURSOR FS RIGHT	.	<	L	\					LINE INS			
1101	NEW CR LINE	GS	-	=	M	SET PROTECT					LINE DEL	Start Protect		
1110	SO	RS	.	>	N	A					PAGE EOT ON			
1111	SI	US NEW LINE	/	?	O	-					PAGE EDIT OFF	Video Control		

U S A S C I	EBCDIC	U S A S C I	EBCDIC	U S A S C I	EBCDIC	U S A S C I	EBCDIC	U S A S C I	EBCDIC	U S A S C I	EBCDIC	U S A S C I	EBCDIC	U S A S C I	EBCDIC
00	00 NUL	10	10 DLE	20	40 SP	30	F0 0	40	7C @	50	D7 P	60	79 \	70	97 P
01	01 SOH	11	11 DC1	21	4F !	31	F1 1	41	C1 A	51	D8 Q	61	81 a	71	98 q
02	02 STX	12	12 DC2	22	7F "	32	F2 2	42	C2 B	52	D9 R	62	82 b	72	99 r
03	03 ETX	13	13 DC3	23	7B #	33	F3 3	43	C3 C	53	E2 S	63	83 c	73	A2 s
04	37 ECT	14	3C DC4	24	5B \$	34	F4 4	44	C4 D	54	E3 T	64	84 d	74	A3 t
05	2D ENQ	15	3D NAK	25	60 %	35	F5 5	45	C5 E	55	E4 U	65	85 e	75	A4 u
06	2E ACK	16	32 SYN	26	50 &	36	F6 6	46	C6 F	56	E5 V	66	86 f	76	A5 v
07	2F BEL	17	26 ETB	27	7D '	37	F7 7	47	C7 G	57	E6 W	67	87 g	77	A6 w
08	16 BS	18	18 CAN	28	4D (	38	F8 8	48	C8 H	58	E7 X	68	88 h	78	A7 x
09	05 INT	19	19 EM	29	5D)	39	F9 9	49	C9 I	59	E8 Y	69	89 i	79	A8 y
0A	25 LF	1A	3F SUB	2A	5C *	3A	7A :	4A	D1 J	5A	E9 Z	6A	91 j	7A	A9 z
0B	0B VT	1B	27 ESC	2B	4E +	3B	5E !	4B	D2 K	5B	4A [	6B	92 k	7B	CO {
0C	0C FF	1C	1C FS	2C	6B ,	3C	4C <	4C	D3 L	5C	E0 \	6C	93 l	7C	6A ;
0D	0D CR	1D	1D GS	2D	60 -	3D	7E =	4D	D4 M	5D	5A]	6D	94 m	7D	DO }
0E	0E SO	1E	1E RS	2E	4B .	3E	6E >	4E	D5 N	5E	5F ^	6E	95 n	7E	A1 ~
0F	0F SI	1F	1F US	2F	61 /	3F	6F ?	4F	D6 O	5F	6D _	6F	96 o	7F	07 DEL

CODIGO EBCDIC

En el código EBCDIC (Extended Binary Coded Decimal Interchange Code), cada caracter es representado por ocho bits; los ocho bits son divididos en dos grupos de cuatro dígitos (cuadro - 3.2).

Puesto que cuatro bits dan los números del 0 al 15 decimal, las letras A, B, C, D, E y F son usadas para los números 10, 11, 12, 13, 14 y 15. Esto da un código hexadecimal. Un caracter EBCDIC es por lo tanto un par de dígitos hexadecimales; por ejemplo, C 1 hexadecimal es el código de la letra A; C 1 es 1100 0001 en código binario en 8 bits.

Dentro de estos códigos existen los caracteres orientados a la comunicación, que son los caracteres que utilizaremos como palabras claves en las disciplinas de línea, para realizar la comunicación entre diferentes dispositivos. Estos caracteres son:

CARACTERES PARA COMUNICACION DE DATOS

NOMBRE/FUNCION

NULL	NULL: Este caracter, donde todos sus dígitos binarios
o	son ceros, sirve para completar tiempos de
NUL	relleno.

- SOM *START OF MESSAGE: Este es usado, en conjunción con EOA, EOM y EOT, para mensajes sobre cinta donde el mensaje es enviado automáticamente.*
- SOH *START OF HEADING: Este caracter de comunicación es usado al principio de una secuencia de caracteres, que constituye una "sensible" dirección o información de Rutinas, cuando una secuencia es referenciada con su encabezado. Un caracter STX tiene el efecto de terminar un encabezado.*
- EOA *END OF ADDRESS: Este caracter, junto con SOM, será usado para definir la sección de cintas perforadas dentro de la cual, el código de llamada directa de la dirección está contenido.*
- STX *START OF TEXT: Caracter de control de comunicación que precede una secuencia de caracteres que están o son tratados como una entidad y transmitido eternamente a su destino. Cada secuencia es referida como texto. STX puede ser usado para terminar una secuencia de caracteres principiado por SOH.*
- EOM *END OF MESSAGE: Este puede usarse para separar mensajes individuales que se están enviando en secuencia en una sola transmisión entre dos estaciones (ver SOM).*

NOMBRE/FUNCION

- ETX END OF TEXT: *Caracter de control de comunicación usado para terminar una secuencia de caracteres que fué iniciada con STX y transmitida como una entidad.*
- EOT END OF TRANSMISSION: *Caracter de control de comunicación usado para indicar la conclusión de una transmisión que puede tener contenido uno o más textos y cualquier encabezado asociado.*
- WRU ENQUIRY: *Caracter de control de comunicación usado en el sistema de comunicación de datos como un requeridor de respuesta de una estación Remota. Esta puede ser usada como "Quién eres" (WRU) para obtener identificación y también es usado para saber el estado de disponibilidad de una estación remota.*
- RU ARE YOU: *Este caracter es usado para la confirmación de regreso de respuesta.*
- ACK ACKNOWLEDGE: *Caracter de control de comunicación, enviado por un receptor a un transmisor como una respuesta afirmativa de que llegó bien el mensaje.*

BELL	BELL: <i>Caracter usado cuando hay una necesidad para llamar la atención humana; puede ser una alarma de control o dispositivo de atención.</i>
FE O BS	BACKSPACE: <i>Formato que efectúa control de movimientos del mecanismo de la impresión, retrasando impresiones en una misma línea.</i>
HT/SK O HT	HORIZONTAL TABULATION: <i>Formato que lleva el control de movimientos de los mecanismos de una impresión para las próximas series de posiciones predeterminadas a lo largo de las líneas de impresión. (aplicable también a la función salto en tarjetas perforadas)</i>
LF O LF	LINE FEED: <i>Formato que efectúa control de movimientos de una línea de impresión en un tiempo.</i>
VT O VT	VERTICAL TABULATION: <i>Formato que efectúa control de movimientos del papel para las próximas series de líneas de impresión.</i>
FF	FORM FEED: <i>Formato que efectúa control de movimientos para la posición de impresión.</i>

NOMBRE/FUNCION

- CR CARRIAGE RETURN: Formato que efectúa control de movimiento del mecanismo de impresión para la primera posición impresa en alguna línea de impresión.
- SO SHIFT OUT: Caracter de control que indica que las combinaciones de código que siguen deberán ser interpretadas como fuera del conjunto de caracteres del código estandar hasta que una variación de caracteres sea alcanzada.
- SI SHIFT IN: Caracter de control que indica qué combinación de código deberá interpretarse de acuerdo con el código estandar.
- DC
o
DLE DATA LINK ESCAPE: Caracter de control de comunicación que cambiará el sentido de un conjunto limitado de caracteres consecutivos siguientes. Este es usado exclusivamente para proveer controles suplementarios en Redes de Comunicación de Datos.
- DC1
DC2
DC3
DC4 DEVICE CONTROLS: Caracteres para el control auxiliar de dispositivos asociados con el procesamiento de datos o sistemas de comunicación, especialmente cambiando dispositivos en ON ó OFF.

NOMBRE/FUNCION

ERR *NEGATIVE ACKNOWLEDGE: Caracter de control de comunicaci^on transmitido por un receptor como una respues*
o *ta negativa al emisor.*

SYN *SYNCHRONOUS IDLE: Caracter de control de comunicaci^on usado por un sistema de transmisi^on sincr^ona en ausencia de cualquier otro caracter para proveer una se^ñal y de esta forma mantener la sincron^oa.*

LEM *LOGICAL END OF MEDIA: Usado para indicar el fin de la informaci^on usable, como un "fin-de-tarjetas".*

ETB *END OF TRANSMISSION BLOCK: Caracter de control de comunicaci^on usado para indicar el final de un bloque de datos para prop^ositos de comunicaci^on.*

SO *Separador de informaci^on*

CAN *CANCEL: Un caracter de control usado para indicar que el dato con el cual este es enviado, est^a en error o est^a rechazado.*

SI *Separador de informaci^on*

NOMBRE/FUNCION

- EM END OF MEDIUM: *Un caracter de control asociado con el envlo de datos, los cuales podrn ser usados para identificar el final fisico del medio o el final de informacin registrada en un medio. (La posicin de este caracter no necesariamente corresponde al final fisico del medio).*
- SUB SUBSTITUTE: *Caracter que podra ser sustituido por otro caracter.*
- ESC ESCAPE: *Caracter de control encaminado a proveer extensi3n de c3digo (caracteres suplementarios), para intercambio de informaci3n.*
El caracter "esc" es asl mismo un prefijo que afecta la interpretaci3n de un n3mero limitado de caracteres consecutivos siguientes.
- FS FILE SEPARATOR, GROUP SEPARATOR, RECORD SEPARATOR y
 GS UNIT SEPARATOR: *Estos separadores de informaci3n podr3n ser usados dentro de datos en forma opcional,*
 RS *la jerarqu3a que tienen para separar est3 dada como*
 US *aparecen a la izquierda.*
- SP SPACE: *Caracter que normalmente no se imprime; usado para separar palabras*

NOMBRE/FUNCION

- BCC *Chequeo del bloque de caracteres. Este caracter es usado para la detección de errores en la paridad.*
- DEOT *Mandato de desconexión. Este caracter es usado como señal de desconexión en circuitos conmutados.*

CARACTERES DEFINIDOS PARA DISCIPLINAS

- AD1, *Dirección 1, Dirección 2. Estos caracteres son usa*
AD2 *dos para direccionar dispositivos.*
- BS# *Número de bloque. Caracter usado para subdividir una transmisión, el BS# consiste de dos caracteres (que son dos números) que identifican el número de bloque secuencial en el mensaje bloqueado.*
- BSL *BOADCAST SELECT. Este caracter es usado para identificar que "éste es un mensaje difundido" a todas las terminales.*
- FSL *FAST SELECT. Este caracter es usado para indicar "ésta es una selección rápida" en una selección secuencial transmitida por la computadora central.*

- GSL GROUP SELECT. Este caracter es usado para indicar "éste es un mensaje para un grupo de estaciones"
- POL POLL. Caracter utilizado para indicar " ésta es una elección".
- SEL SELECT. Caracter utilizado para indicar " ésta es una selección normal"
- SEQ SEQUENTIAL SELECT. Caracter utilizado para indicar que el grupo de terminales remotas está siendo seleccionado para recibir un mensaje direccionado al grupo. La última terminal seleccionada dentro del grupo mandará el acuso de recibo de la selección se cuencial.

XM# Número de transmisión

CONVENCIONES

Las convenciones son usadas para ilustrar los procedimientos de comunicación de datos descritas más adelante.

CONTROL Programa que administra la comunicación de datos

- MAESTRO** *Dispositivo que transmite el dato*
- ESCLAVA(S)** *Dispositivo(s) y/o sistema(s) que está(n) sién
do controlado(s) por el maestro*
- FLUJO DE
CARACTERES** *El lado izquierdo del flujo de caracteres (en
los diagramas) refleja generalmente la función
del maestro y el lado derecho muestra la fun-
ción de la esclava*
- FLECHAS** *El flujo de datos es representado por 
y el flujo lógico es representado por *

3.5 EJEMPLOS DE DISCIPLINAS DE LINEA

3.5.1 DISCIPLINA DE LINEA

"POLL" (ver diag 3.1)

Esta disciplina es utilizada en una red, cuando tenemos en una línea de comunicación varios dispositivos y queremos comunicarnos con alguno en particular.

DISCIPLINA

El primer paso es que la computadora envíe un EOT; con esto activa todas las unidades remotas en el estado de control (prepara a los dispositivos para recibir un "poll" o un "selection"). Luego transmite el domicilio del dispositivo y un poll. Las terminales que están direccionadas chequean el mensaje para ver si es para ella; si el mensaje no le corresponde o si no se encuentra disponible para transmitir datos, este dispositivo enviará un EOT.

Después del caracter poll se envía un ENQ; éste es usado como el final de caracteres en un mensaje poll, cuando es requerida una respuesta de estación.

Ahora pasamos a ver que sucede con la terminal: Si la terminal recibe un mensaje poll inválido, no producirá

ninguna respuesta. Esta falta de contestación hace que el procesador central genere un "time out" e inicie procedimientos de recuperación. La falta de respuesta por lo general, es la resultante de una de las tres condiciones siguientes:

- i) La terminal direccionada no está en línea
- ii) Los caracteres EOT, POLL, y/o ENQ son correctos.
- iii) El domicilio AD1, AD2 a continuación del caracter EOT no es el adecuado y, consecuentemente, impide que cualquier estación en línea reconozca la dirección.

De otra forma, el procesador central recibirá un SOH y con esto iniciamos la recepción, recibirá la dirección de la terminal, el indicador de principio de texto, el texto, control de fin de texto y un BCC que es un checador de paridad; si recibió bien transmitirá un ACK para indicar a la estación que puede continuar enviando más mensajes y el procesador central continuará con la secuencia "polling".

En caso de mala recepción ya sea por alguna irregularidad en la paridad de caracteres, verificación de bloque o de la dirección de la estación, se enviará un NAK; con esto se estará solicitando la repetición del envío. Este proceso puede repetirse un número específico de veces.

Si después de "n" tiempo ó "n" retransmisiones no hay respuesta satisfactoria, se transmitirá una EOT, para dar por terminada la secuencia.

Control

Esclava (hasta que la maestra envia EOT)

Maestra

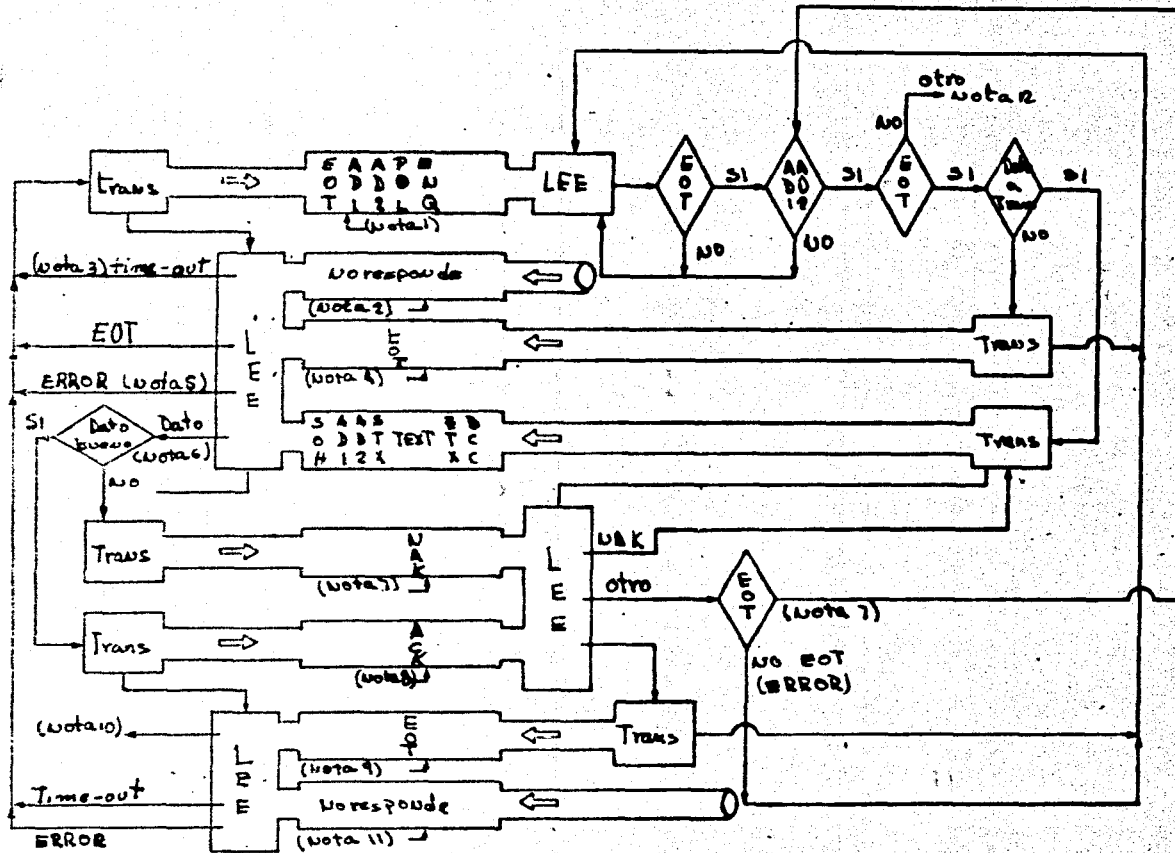


DIAGRAMA 3.1 DISCIPLINA POLL

NOTAS DE LA DISCIPLINA POLLING

- 1) Este EOT podría ser la terminación de una secuencia previa de transmisión
- 2) Si una-esclava no recibe o reconoce un EOT seguido por su dirección (AD1, AD2), no responderá al "poll" Esta condición de no responder causa un "time-out" al maestro.
- 3) El "time-out" en la maestra es causada por la esclava al no responder.
- 4) Si la esclava no tiene datos para transmitir cuando es "poleada", enviará un EOT.
- 5) Existe una condición de error cuando la maestra recibe otra cosa que no sean datos, EOT o Time-out como respuesta a un poll.
- 6) La decisión de cuando los datos son buenos o malos, está en base a que la dirección sea válida en la esclava, paridad en caracteres y en el BCC.
- 7) Si la esclava recibe malos datos, por lo que se dice en la nota (6), transmitirá un NAK para que le repitan la transmisión.
- 8) Un ACK indica que se recibió bien el mensaje e indica que la maestra deberá responder con EOT para completar el ciclo del poll.
- 9) Fin de la transmisión

- 10) Recepción de EOT desde la maestra al control indica terminación normal de un ciclo Poll.
- 11) Si la esclava no recibe ACK, NAK, o EOT, ésta puede retener su mensaje y quedarse callada. Los tiempos fuera de control y la transmisión de un EOT, terminarán la secuencia. El mensaje será retransmitido con el próximo Poll.
- 12) En este punto, la esclava podrá checar otros caracteres de control tales como el SEL, FSL, BSL, o GSL, como también el Poll.

3.5.2 DISCIPLINA DE LINEA

"SELECTION" (ver diag 3.2)

El caso de la disciplina "selection" es muy parecido al "polling" (de hecho estas dos disciplinas se trabajan en conjunto para escribir y transmitir de un dispositivo), ya que con esta disciplina manejaremos varios dispositivos en una misma línea.

DISCIPLINA

El computador transmitirá un EOT desde el procesador central, ésto activará todas las unidades remotas en el estado de control y se prepara a la terminal para un "selection".

Este EOT puede ser la terminación de una secuencia de transmisión anterior. Para minimizar el efecto del ruido, la secuencia de selección deberá seguir inmediatamente. Pero suponiendo que no es la terminación de una secuencia, entonces, el procesador transmite la dirección, la secuencia de selección y el ENQUIRY que es usado como el final de caracteres en una transmisión selection.

Iniciamos recepción.- Si la terminal no está lista para recibir, el procesador central recibirá un NAK de esta

estación; es posible que el procesador central reintente la selección de esa terminal en particular dentro del siguiente ciclo de selección.

Ahora si la terminal recibe un mensaje de control inválido, no responderá. Esta falta de respuesta origina un "time out" en el procesador central y hace que inicie procedimientos de recuperación. Por lo general la falta de respuesta se debe a una de las cuatro condiciones siguientes:

- i) La terminal direccionada no está en línea.
- ii) La dirección a continuación del EOT no es correcta, por lo que impide que cualquier terminal de la línea reconozca el domicilio.
- iii) El caracter ETX es incorrecto. Si el caracter ETX se ha perdido, la terminal direccionada continúa buscando este caracter o bien un EOT. Consecuentemente, el centro de cómputo deberá aplicar uno de los dos métodos siguientes al recuperarse de un "time out".
 - a) Si se desea volver a llamar a la terminal que originó el "time out", se está en condiciones de hacerlo. El EOT inicial regresa a esta terminal para activar la recepción de mensajes en caso de que el ETX haya sido inadecuado.

b) Si se desea dirigirse a otra terminal en la misma línea como la terminal que originó el "time out", puede proceder con la secuencia de procedimiento de comunicación EOT, AD1, AD2, etc, ya que únicamente la terminal direccionada en la secuencia anterior esperará el ETX. El EOT de esta secuencia regresa hacia la terminal que espera ETX y la sitúa en un estado de lista para recepción de mensajes de control.

iv) Los caracteres EOT, SEL, y/o ENQ son incorrectos.

Ahora si el procesador central recibe un ACK entonces transmite el SOH, dirección, el número de transmisión, la secuencia de principio de texto y texto y finaliza transmitiendo el BCC para checar paridad.

De nuevo recibirá un ACK si es que recibió bien y continuará con la secuela del "selection", pero si recibe un NAK quiere decir que la paridad de los caracteres o la verificación del bloque no es válida para la terminal. Entonces el procesador central retransmitirá el mensaje un número específico de veces. Si aún después de ello, la terminal no reconoce el mensaje, el procesador central queda en libertad de cumplir la secuencia con EOT después de grabar el error. También es factible que el procesador central, según opción prevista por lo programado, retenga el mensaje para retransmitirlo durante la siguiente secuencia

de selección de la terminal especificada. Pero si el procesador no recibe una respuesta ACK o NAK a su mensaje, se genera un "time out" y retransmitirá un número específico de veces. Si después de esto no se recibe respuesta alguna, el procesador central produce un "time out", concluye la secuencia con EOT y registra el error. A opción del programa dor, el procesador central puede retener el mensaje para ser retransmitido durante la siguiente secuencia de selección a la terminal específica.

Maestra

Esclava (s)

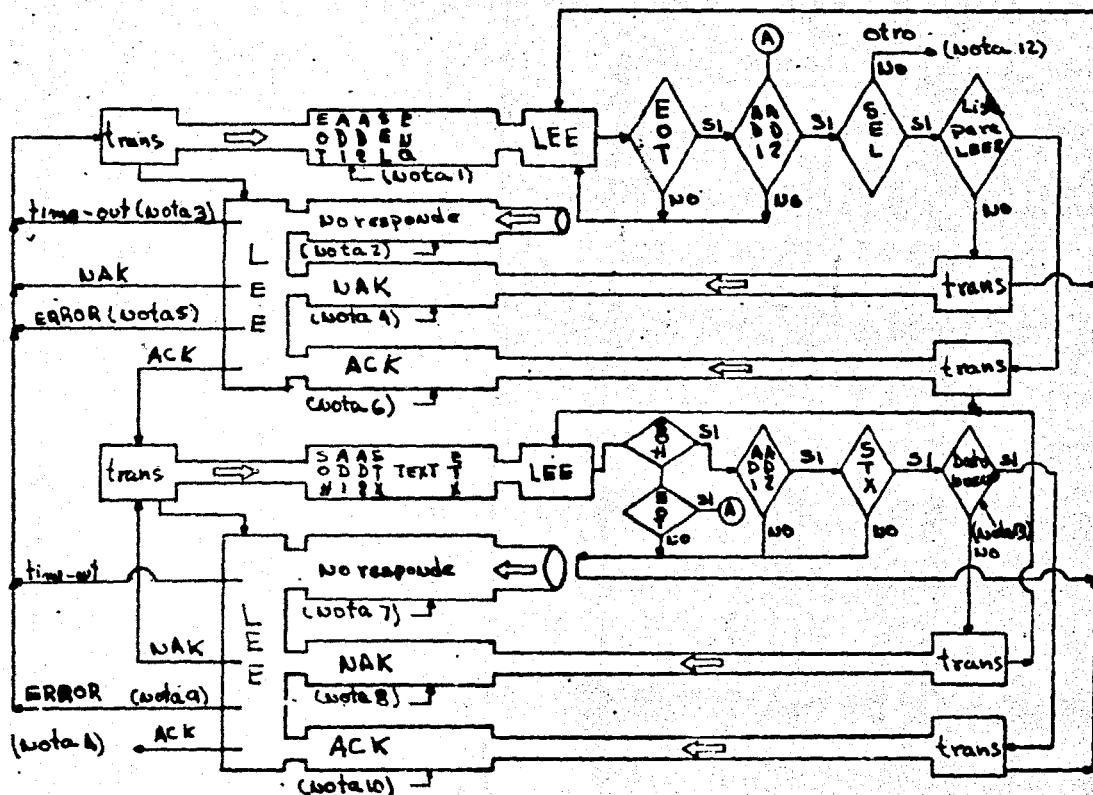


DIAGRAMA 3.2 DISCIPLINA SELECTION

NOTAS DE LA DISCIPLINA "SELECTION"

- 1) Este EOT pudo haber sido la terminación de una pre-
via transmisión
- 2) Si la esclava no reconoce o recibe EOT seguido por
su dirección, no responderá al mensaje "select".
Esta falta de respuesta causa un "time out".
- 3) Los "time out", en la maestra, son causados por fallas
de la esclava al no responder a un "select", con un
caracter ACK o NAK.
- 4) Si la esclava no está lista para recibir, como es
indicado por la transmisión de un NAK, la maestra
reintentará la selección de la esclava en próximas
secuencias.
- 5) Existe una condición de error cuando la maestra re-
cibe otra cosa que no sea un NAK, ACK o un time out.
- 6) Una respuesta de ACK a un mensaje "select", indica
que la esclava está lista para recibir.
- 7) Si la maestra no recibe una respuesta (ACK o NAK) a
su mensaje, hará un "time out" y retransmitirá el da-
to n veces.
- 8) Si hay paridad en caracteres, al checar bloque o la
identificación no es válida para la esclava; ésta
transmitirá un NAK. La maestra podrá retransmitir
n veces.

- 9) Existe una condición de error si la maestra recibe alguna otra cosa diferente a un ACK, NAK o time out como una respuesta a un mensaje de datos. Después de n intentos fallados, un proceso de recuperación de error será ejecutado.
- 10) Cuando la esclava responde a un mensaje de datos con un ACK, Éste es un indicador de que el mensaje fue recibido propiamente. La maestra deberá transmitir un EOT (puede ser el principio de otro mensaje select o poll) para completar el ciclo de selection.
- 11) Recepción de un ACK por el maestro, indica el término del ciclo de selección. Este es el momento para que la maestra inicie otro select o alguna otra disciplina de línea.
- 12) En este punto, la esclava puede checar otro caracter de control tal como POLL, FSL, BSL, o GSL así como el mismo select.
- 13) La decisión sobre qué dato es bueno o malo está en base a que sea válido la dirección de la esclava, caracteres de paridad, y el BCC.

3.5.3 DISCIPLINA DE LINEA

FAST SELECT (FSL) (ver diag 3.3)

Esta disciplina se aplica siempre que se desea transmi
tir un mensaje a una terminal cuyo estado es el de listo para

ra recepción. i.e. se usa cuando el procesador central desea transmitir un mensaje sin probar el estado de recepción de la terminal.

Se utiliza el caracter "S" del código modificado ASCII (X34.4, VII/67).

DISCIPLINA:

Se inicia la transmisión cuando el procesador central transmite un EOT; con esto activamos todas las unidades remotas en el estado de control, sin embargo este EOT puede ser la conclusión de una secuencia anterior de transmisión. Para minimizar el efecto del ruido, puede seguirse inmediatamente con la secuencia de fast select.

Se transmite la dirección, el FSL y el principio de cabeza, se transmite de nuevo la dirección ("you are"), el número de transmisión, el STX para indicar el principio del texto y el texto, después se transmite el fin de texto y el BCC para checar paridad. y se prepara para recibir. Si se recibe un error de mensaje NAK, es porque la paridad del mensaje no correspondió por haber llegado mal; el procesador central retransmitirá la totalidad del mensaje un número específico de veces.

Si la terminal no reconoce el mensaje, el procesador central termina la secuencia con EOT y registrará el error. Según la opción que sea provista por el programador, es posible que el procesador central retenga el mensaje para retransmitirlo durante la siguiente secuencia de selección de la terminal especificada.

Si la terminal no produce ninguna respuesta, el centro de cómputo genera un Time out.

- 1) Si el centro de cómputo desea volver a dirigirse a la terminal que originó el time out, está en situación de hacerlo. El EOT regresa a esta terminal al estado "lista para recibir un mensaje de control" si el ETX era incorrecto.*
- 2) Si el centro de cómputo desea dirigirse a otra terminal en la misma línea como si fuese la terminal que causó el time out, puede proceder con la secuencia de procedimiento de comunicación EOT, AD1, AD2, etc, ya que únicamente la terminal direccionada en el fast select está buscando ETX. El EOT de esta secuencia regresa a la terminal que espera ETX al estado de lista para recibir un mensaje de control. Por último si la terminal recibió bien, enviará el ACK para indicar que el mensaje es correcto y con esto continuará el polling, selection, etc.*

Maestra

Esclava(s)

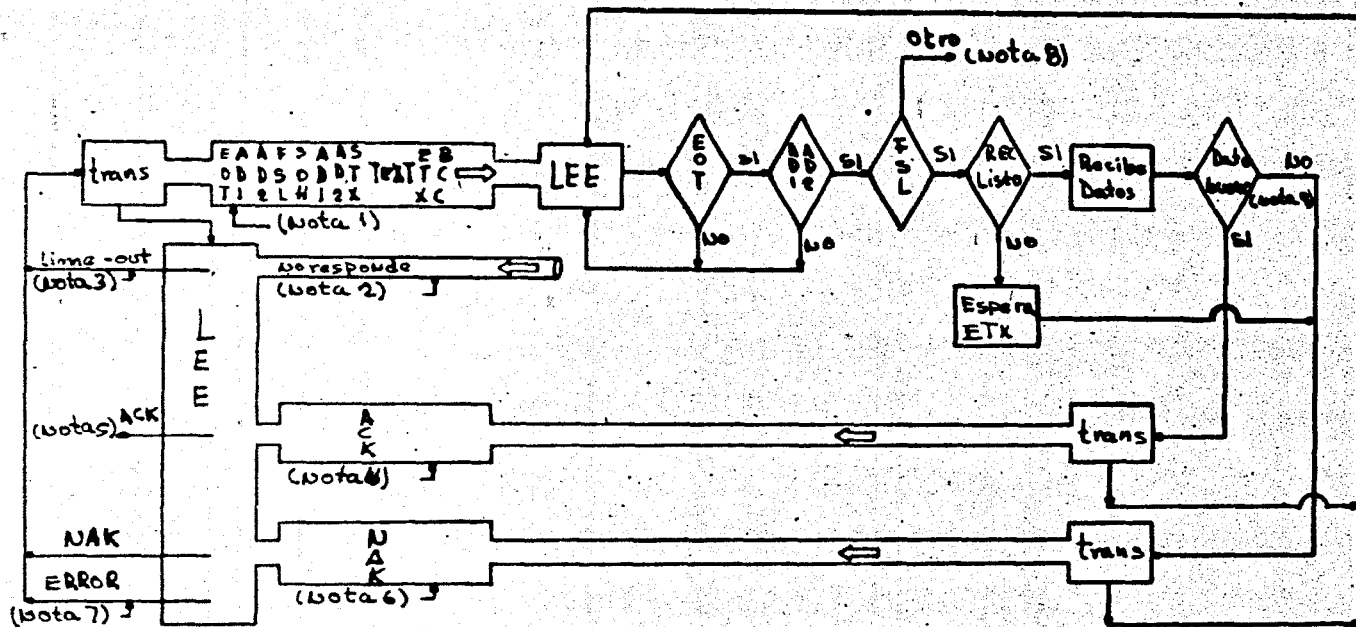


DIAGRAMA 3.3 DISCIPLINA FAST SELECT

NOTAS DE LA DISCIPLINA FAST SELECT

- 1) Este EOT puede ser la terminación de una transmisión previa.
- 2) Si la esclava no recibe o reconoce un EOT seguido por subdirección (AD1, AD2), no responderá al mensaje del fast select. Al no responder genera un time out al maestro.
- 3) El time out es generado al maestro por fallas de la esclava para responder al mensaje fast select con un ACK o NAK.
- 4) Un ACK indica que la esclava está lista para recibir y que el mensaje fue recibido propiamente. La maestra deberá transmitir un EOT para completar un ciclo de fast select.
- 5) Recepción de un ACK por la maestra indica término normal del ciclo fast select.
- 6) Si la esclava no está lista para recibir o si hay paridad de caracteres, chequeo de bloque o la identificación de la esclava no es válida, ella enviará un NAK; la maestra retransmitirá el dato.
- 7) Existe una condición de error si la maestra recibe otra cosa diferente a un NAK, ACK o time out a un mensaje fast select.
- 8) En este punto la esclava puede checar otro tipo de caracteres de control tales como POLL, SEL, BSL, o

GSL y el mismo mensaje fast select.

- 9) La decisión sobre si el dato es bueno o malo está basado en el chequeo de paridad, chequeo de bloque, y que la dirección de la esclava sea válida.
- 10) La esclava deberá esperar recepción del mensaje antes respondiendo con NAK.

3.5.4 DISCIPLINA DE LINEA

"BROADCAST SELECT" (BSL) (ver diag 3.4)

Este procedimiento se utiliza cuando se desea transmitir un mensaje a todas las unidades de una línea. Se emplea un caracter especial "broadcast select" (BSL) del código: ASCII, para indicar a todas las terminales la naturaleza de la transmisión y que todos ellos han de aceptar el mensaje. Este caracter especial es sucedido por el principio del encabezado (SOH) del mensaje. No se lleva a cabo ningún reconocimiento de la parte SELECT de la transmisión. El "broadcast select" suprime el método local de transmisión, ocasionando que la terminal pase automáticamente al método de recepción y acepte el mensaje BSL.

DISCIPLINA:

Se inicia la transmisión cuando se transmite un EOT

para despertar a las terminales que están en estado de control, para minimizar el efecto del ruido, conviene seguir inmediatamente con la secuencia de "broadcast".

Se transmite la dirección, este domicilio identifica a la terminal que reconocerá el mensaje, se transmite "broadcast select" (BSL), el principio de cabeza, la dirección ("you are"), se transmite el número de transmisión, lo cual es opcional, pero si es transmitido, la terminal ignora el carácter XMT, excepto al calcular el BCC. Transmite fin de texto y el checadore de paridad.

Si el mensaje contiene un error de paridad, la terminal transmite NAK, lo que considera el procesador central como una solicitud para volver a emitir la última transmisión. Cuando el procesador central no recibe respuesta de la terminal se presenta un time out. El time out puede ser por alguna de las cuatro condiciones siguientes:

- a) La terminal direccionada no está en línea.
- b) Los caracteres EOT, SOH, BSL y/o STX son incorrectos
- c) El domicilio AD1, AD2, es incorrecto, lo que impide que cualquier terminal, en la línea reconozca la dirección.
- d) El carácter ETX no es correcto. Cuando esto sucede, ninguna de las terminales lo reconoce, y continúa

buscando un caracter ETX o un caracter EOT.

Puesto que todas las máquinas de una línea perciben un mensaje "broadcast select", si un caracter incorrecto ETX es responsable del time out, todas las máquinas estarán tratando de localizar un ETX o un EOT. El EOT inicial del siguiente mensaje de control, regresa a todas las terminales al estado de "listas para recibir mensaje de control;" permitiendo el reconocimiento de un poll, select, fast select, etc, enviados por el centro de computo.

Puesto que todas las máquinas de una línea perciben un mensaje "broadcast select", si un caracter incorrecto ETX es responsable del "time out", todas las máquinas estarán tratando de localizar un ETX o un EOT. El EOT inicial del siguiente mensaje de control, regresa a todas las terminales al estado de "listas para recibir mensaje de control", permitiendo el reconocimiento de un poll, select, fast select, etc, enviados por el centro de cómputo.

Ahora volviendo al punto cuando el procesador central no recibe respuesta de la terminal, se presenta el time out y puede retransmitir el mensaje original, Las retransmisiones se ejecutan un número específico de veces (que puede ser cero) y, si no hay respuesta afirmativa, el procesador central registra el error y concluye la emisión con un EOT.

Maestra

Esclava(s)

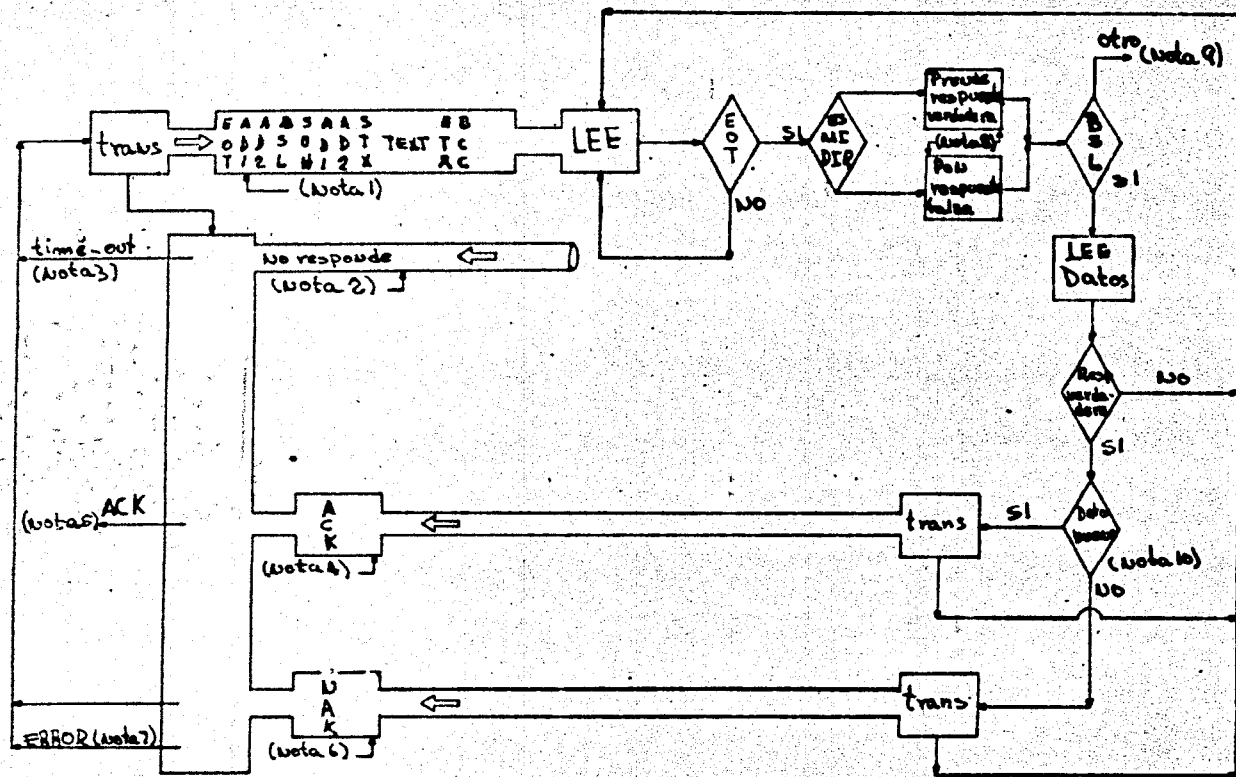


DIAGRAMA 3.4 DISCIPLINA BROADCAST SELECT

NOTAS DE LA DISCIPLINA "BROADCAST SELECT"

- 1) Este EOT puede ser la terminación de una previa secuencia de transmisión
- 2) Si la esclava no recibe o reconoce un EOT seguido por su dirección (AD1, AD2), no responderá a el mensaje broadcast select. Esto causa un time out en la maestra.
- 3) El time out en la maestra es causado por fallas de la esclava para responder con un ACK o un NAK.
- 4) Un ACK indica que la esclava está lista para recibir y que recibió apropiadamente el mensaje. La maestra transmitirá un EOT para completar el ciclo de broadcast select.
- 5) La recepción de un ACK por la maestra, indica término : normal del ciclo broadcast select. En este momento, la maestra puede iniciar otro broad cast select o alguna otra disciplina de línea.
- 6) Si la esclava no estaba lista para recibir o si hubo caracter de paridad, chequeo de bloque o la identificación de la esclava no es válida, ésta enviará un NAK.
- 7) Existe una condición de error cuando la maestra recibe alguna otra cosa diferente a un NAK, ACK o time out de un mensaje broadcast select.

- 8) La esclava prenderá una bandera en base a recepción de su dirección (AD1, AD2) dentro de un mensaje broadcast select. Es responsabilidad de la esclava el ver su dirección para responder (ACK o NAK) al mensaje broadcast select. Las demás esclavas permanecen silenciosas.
- 9) En este punto, la esclava puede checar otros caracteres de control tales como poll, SEL, FSL, o GSL y el mismo mensaje de broadcast select.
- 10) La decisión sobre si un dato es bueno o es malo está en base a caracteres de paridad y chequeo de bloque.

3.5.5 DISCIPLINA DE LINEA

"GROUP SELECT" (GSL) (Ver diag 3.5)

El Group select es una disciplina fast select pero para un grupo de terminales. En el group select se envia el domicilio de grupo i.e. el GSL en lugar del FSL.

El domicilio de grupo (group select) puede ser cualquiera de los convenidos en el diagrama de código ASCII, columnas dos a la siete. Ahora el domicilio en el mensaje es el del grupo, además de la dirección individual de una de las terminales del conjunto. El caracter group select se asigna a las terminales en el momento de instalarlas, simultáneamente con la asignación de domicilio.

DISCIPLINA

Las terminales que tengan un domicilio de grupo de acuerdo con el contenido en la transmisión, aceptan y reciben el mensaje. El group select suprime los métodos local y de transmisión en la terminal, dando lugar a que ésta cambie automáticamente al método de recepción para aceptar el mensaje GSL al tiempo que reciba la información.

Únicamente la terminal AD1, AD2, en el grupo, responde con ACK o NAK según el caso.

Si el mensaje contiene un error de paridad, la terminal transmite un caracter NAK, lo que indica al procesador central que se requiere la repetición del mensaje transmitido.

Ahora, cuando el procesador central no recibe respuesta de la terminal, se presenta un time out y es posible que vuelva a seleccionar al grupo.

La reelección o la repetición de la transmisión, se efectúan el número específico de veces, que bien puede ser cero, hasta que se obtiene una respuesta positiva en el procesador central.

Si el límite de group select es alcanzado sin recibir un reconocimiento afirmativo, ocurrirá un time out en el procesador central, se registra el error y se termina el procedimiento con EOT.

RECUPERACION DE TIME OUT

Cuando no hay respuesta de parte de la terminal, se presenta un time out en el centro de cómputo. La falta de respuesta se debe por lo común a una de las cuatro condiciones siguientes:

a) Los caracteres EOT, SOH, GSL, y/o STX son incorrectos.

b) El domicilio AD1, AD2, es incorrecto, impidiendo que cualquier terminal en línea reconozca los dos caracteres.

c) La terminal direccionada (AD1, AD2, GSL) no está en línea.

d) El caracter ETX no es correcto. Si este caracter se perdió, la terminal direccionada continúa buscando ETX o bien EOT. En consecuencia, el centro de cómputo debe aplicar uno de los dos métodos que se mencionan en seguida de recuperarse de un time out. Estos métodos están basados en la presunción de que dicho time out ocurrió debido a la pérdida del ETX.

i) Si el centro de cómputo desea dirigirse a una de las terminales que forma parte del conjunto que originó el time out, estará en condiciones de hacerlo. El EOT inicia el mensaje de control.

ii) Si el centro de cómputo desea dirigirse a otra terminal en la misma línea, pero que no pertenece al grupo que acaba de originar el time out, puede proceder con EOT, AD1, AD2, etc. El EOT de esta secuencia regresará a cualquier terminal que esté esperando un ETX al estado de "lista para recibir mensajes de control".

Esclava (s)



DIAGRAMA 3.5 DISCIPLINA GROUP SELECTION

NOTAS DE LA DISCIPLINA "GROUP SELECTION"

- 1) El EOT puede ser de la terminación de una previa secuencia de transmisión.
- 2) Si la esclava no recibe o reconoce un EOT seguido de su dirección (AD1, AD2), no responderá al mensaje group selection. Esto causa un time out en el maestro.
- 3) El time out en el maestro, es causado por fallas de la esclava apuntada para responder al mensaje group selection con un ACK o NAK.
- 4) Una respuesta de ACK indica que la esclava apuntada estuvo lista para recibir y que recibió apropiadamente el mensaje. La maestra deberá transmitir un EOT para completar el ciclo de group selection.
- 5) La recepción de un ACK por el maestro indica una normal terminación del ciclo group select. En este momento la maestra está en opción de empezar otro ciclo de group select o alguna otra disciplina.
- 6) Si la direccionada esclava no estaba lista para recibir o si tenía paridad en caracteres, chequeo de bloque, o la identificación de la esclava era inválida enviará un NAK.
- 7) Existe una condición de error si la maestra recibe cualquier otra cosa diferente a un NAK, ACK o time out de un mensaje group select.

- 8) La esclava prenderá una bandera fundamentada sobre recepción de su dirección (AD1, AD2) dentro de un mensaje group select. Es responsabilidad de la esclava ver la dirección para responder (ACK o NAK) al mensaje group select. Las demás esclavas permanecen en silencio.
- 9) En este punto, la esclava puede checar cualquier otro caracter de control tal como POLL, SEL, FSL, á BSL, como también el mensaje del group select.
- 10) Además de checar su dirección, la esclava deberá checar su dirección de grupo. Si una esclava recibe su dirección de grupo pero no su individual dirección, deberá de recibir el mensaje sin responder. Si la esclava recibe su individual dirección sin un group select, es responsable de responder al mensaje; si la esclava no recibe su dirección o la dirección de grupo, ésta no deberá recibir o responder al mensaje.
- 11) La decisión sobre si el dato es bueno o malo está basada sobre paridad en caracteres y el chequeo de bloque.

3.5.6 DISCIPLINA DE LINEA

"GROUP POLL" (opcional) (ver diag 3.6)

Este procedimiento es una opción del método estandar "multipunto". La terminal posee la capacidad de aceptar

AD1, AD2 adicionales que no sean su domicilio particular.

Si estos domicilios de grupos se decodifican durante el procedimiento estandar de poll, la terminal responde intentando transmitir un caracter EOT (indicador de ausencia de tráfico) o bien transmitiendo un mensaje si está en método de transmisión. Este procedimiento se utiliza para reducir el "overhear" en una red de terminales en la cual varias de ellas están localizadas en un sitio y participan de una misma línea de comunicación. El recibir un group poll resultará, en caso de que las terminales no estén listas para transmitir, en una respuesta correspondiente al grupo.

En estas condiciones es posible que el control pase al grupo siguiente. Durante periodos de baja actividad, el control estará en situación de efectuar una lista de polling para determinar el estado de todas las terminales con excepción de un poll en cada ubicación y no uno en cada terminal. Además, si existen varias terminales listas para transmitir en una ubicación, se permite que lo lleven a la práctica en secuencia, como respuesta a un poll, selecting, broadcast, fast select, etc; no se ven afectados por este procedimiento de "poleo" en grupo.

En una concatenación de terminales, si una de ellas es eliminada por cualquier causa, la pérdida de este dispo-

sitivo periférico no afecta el procedimiento aquí descrito.

DISCIPLINA

La transmisión ordenada de datos hacia el procesador central por todas las terminales que posean el mismo domicilio del grupo, se lleva al cabo de la siguiente manera:

Todas las terminales de grupo deben de pertenecer a la misma concatenación, i.e., se transmite EOT, el domicilio, el POLL y el ENQ.

En este momento, la secuencia de "poleo" se apega el mismo formato que el "poleo" normal y usa el caracter normal para "poll". El "group polling" está controlado sólo por domicilios. El domicilio del grupo se asigna a un conjunto de terminales concatenadas. Cada miembro del grupo responde al domicilio del conjunto, como si esta dirección fuera la suya en particular.

Cuando se recibe un "poll" por el grupo direccio nado, cada terminal conectada al modem por concatenación prepara su respuesta activando la línea de la señal "request-to-send". Las terminales que tengan un mensaje listo para ser transmitido, preparan ésto; las terminales que no tengan mensaje, se situardn listas para transmitir un EOT.

En secuencia, cada terminal sin mensaje probará la línea de la señal "request-to-send" y, si se detecta una de estas señales desde otra terminal, aquella que no tenga mensaje cancelará su EOT y esperará a la siguiente EOT enviada por el procesador central.

Todo mensaje enviado en respuesta a un "poll" de grupo, contiene el domicilio individual de la terminal que contesta. Esta bloquea las señales concatenadas "request-to-send" y "clear-to-send", relacionadas con las terminales más remotas considerando el modem. Estas señales permanecen bloqueadas de las terminales más alejadas hasta que la terminal que responde haya completado su transmisión y recibido un caracter ACK o EOT desde el procesador central.

Si el procesador central detecta un error en el mensaje recibido al recibir la respuesta a un "group poll", el caracter NAK se transmite solicitando así que la transmisión sea repetida.

Esta operación puede efectuarse "n" veces (donde n puede ser igual a cero) en cualquier momento; si esta prueba falla se graba el error en el procesador central y se transmite un caracter EOT dando por terminada la secuencia de transmisión. La terminal transmite el mismo mensaje cuando es "poleada" en la siguiente oportunidad.

Si la terminal no recibe ACK, NAK o EOT, puede retener su mensaje y permanecer inactiva.

El procesador central presenta un "time out" y transmite un caracter EOT, terminando así la secuencia de transmisión. El mensaje se transmitirá en la siguiente oportunidad que la terminal sea "poleada".

Cuando no hay otras terminales listas para transmitir, se envía un EOT. Existen algunos sistemas, en donde el número de mensajes permitidos para ser enviado a un grupo, como resultado de un "poll" sencillo, puede estar limitado por un número determinado.

En este caso es posible enviar EOT por medio del procesador central en lugar de ACK, aún si otras terminales listas para transmitir están esperando servicio. La terminal que está esperando ACK, retiene su mensaje hasta que recibe el siguiente "group poll", lo mismo que las demás terminales que estan listas para transmitir.

Cuando el procesador central envía el caracter ACK hacia la terminal que envió el último mensaje, inmediatamente la siguiente terminal, lista para transmitir, iniciará la operación. Pero si al enviar el ACK al procesador central no hay ninguna terminal lista para transmitir, la última

terminal del grupo concatenado tiene la responsabilidad de transmitir un EOT final.

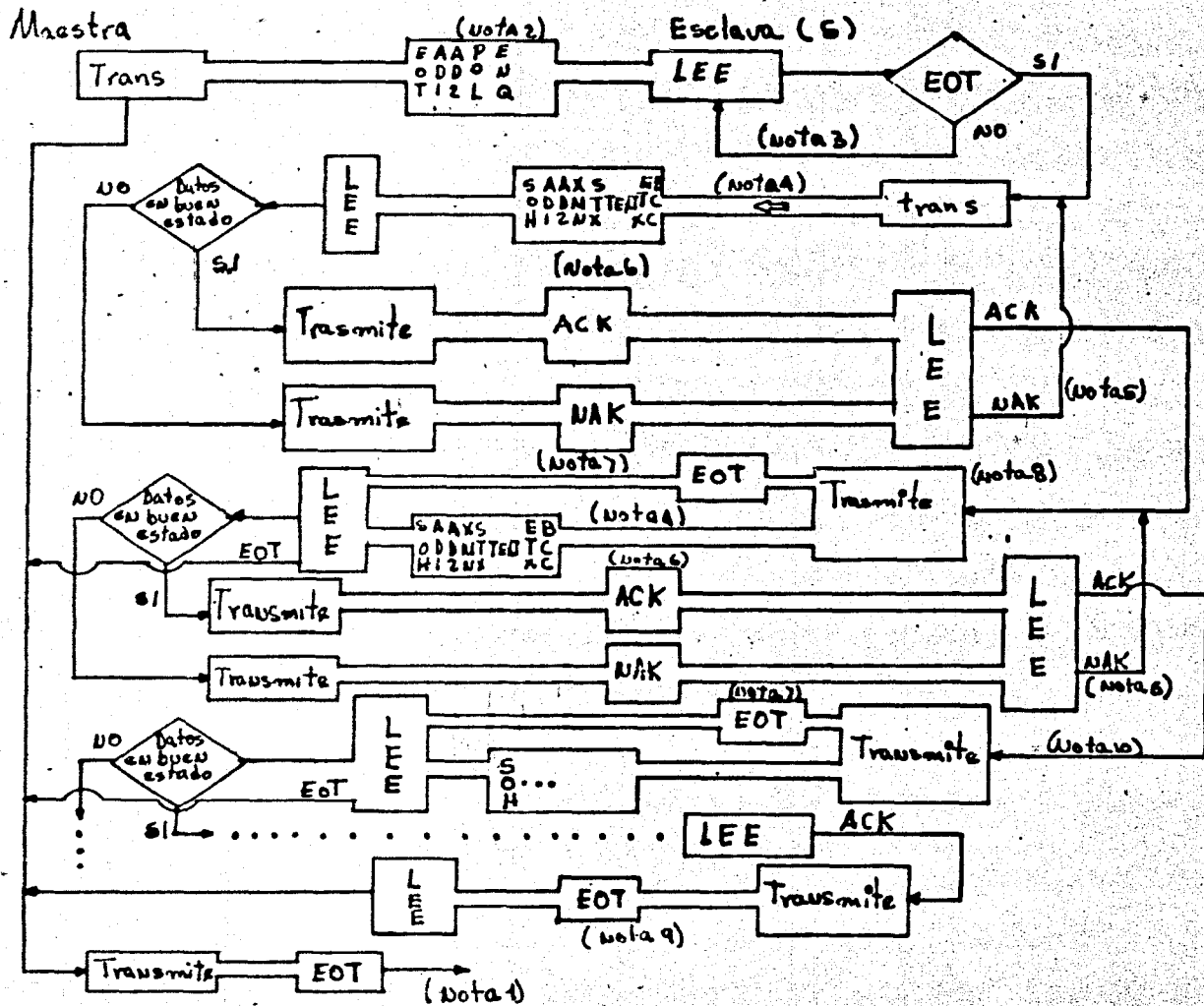


DIAGRAM 3.6 DISCIPLINA GROUP POOL

NOTAS DE LA DISCIPLINA "GROUP POLL"

- 1) Continúa group polling, select, o pasa a control.
- 2) En este procedimiento, la secuencia de "poleo" se apega al mismo formato que al "poleo" normal. El group poll está controlado sólo por domicilios. El domicilio del grupo responde al domicilio del conjunto.
- 3) Al recibir el poll, las terminales que tengan mensaje para transmitir lo transmitirán y las estaciones que no tengan mensajes transmitirán un EOT.
- 4) La respuesta de cada estación va con la dirección individual de cada dispositivo.
- 5) Si el procesador central detecta un error en el mensaje recibido transmitirá el NAK para que la estación retransmita la información.
- 6) Si la terminal no recibe NAK, ACK o EOT, puede retener su mensaje y permanecer inactiva. El procesador central tendrá un time out y transmitirá un EOT.
- 7) Si ninguna terminal está lista para transmitir se envía un EOT.
- 8) Tan pronto como el caracter ACK se recibe desde el maestro, la siguiente terminal que está lista para transmitir iniciará la operación.
- 9) Cuando un ACK se recibe desde el procesador central y no hay terminales listas para transmitir la última

terminal del grupo concatenado asume la responsabilidad de la transmisión del caracter EOT final.

3.5.7 DISCIPLINA DE LINEA

"POINT-TO-POINT CONTENTION"

LINEAS DEDICADAS (ver diag. 3.7)

Este procedimiento se aplica cuando hay dos estaciones en un enlace dedicado "point to point", en el que ninguna estación ha sido designada como la estación maestra.

Ambas estaciones se alternan el estado maestro mediante la transmisión de un ENQ y puede retener el estado siempre que prevalezca la condición de que la otra máquina no esté contendiendo para obtener el estado maestro simultáneamente. Si el estado maestro no se obtiene, lo único que tiene que hacer el operador es reintentar el procedimiento mediante la depresión de la tecla XMT o CR.

El procedimiento de contención determina la relación maestro/esclavo entre ambos puntos.

Una función de conclusión regresa al sistema de la condición de Inactividad.

La condición de inactividad en el enlace comunicador, es la situación subsecuente después de una función de conclusión (EOT) de la transmisión anterior. En esta condición, ninguna estación posee el estado maestro, pero ambas pueden competir por obtenerlo.

Cuando una estación desea transmitir, solicita el estado maestro con sólo enviar un carácter de solicitud ENQ, y empezar un time out, el cual depende del operador. Este reinicia su solicitud después del intervalo del time out, si el estado maestro/esclavo no ha sido definido. Para resolver solicitudes simultáneas emitidas por ambas estaciones que pudiera resultar un mensaje mutilado o ignorado por ambos puntos, el operador deberá reintentar el procedimiento mediante la depresión de la tecla CR o ENTER o XMT, etc.

Cuando se da el caso de una respuesta inválida, una falta de respuesta o recepción de un carácter NAK a un ENQ inicial, la estación que solicita el estado maestro reinicia la solicitud emitiendo un carácter ENQ una vez más (mediante la depresión de la tecla CR o ENTER o XMT). El operador de la estación puede reiniciar este procedimiento cualquier número de veces y, si no obtiene el éxito deseado, puede salir hacia un procedimiento de recuperación de error.

Cuando el punto B está listo para recibir, reconoce

afirmativamente (ACK) el ENQ. El punto A asume entonces el estado maestro y procede con la transmisión del mensaje.

Si la estación maestra recibe una respuesta inválida o no recibe ninguna respuesta a la transmisión de un mensaje, el operador puede enviar un ENQ (que solicita la estación esclava retransmita su último carácter ACK o NAK) oprimiendo la tecla CR o XMT o ENTER según el caso. Este ENQ puede ser "n" veces. Si la esclava había recibido un mensaje inválido y debido a un carácter impropio de control no replicó a la maestra, ahora transmitirá un NAK dando lugar a que la maestra le vuelva a enviar el mensaje de datos. Si la esclava no respondió al mensaje original, volverá a enviar su última respuesta. Si después de enviar la respuesta a la solicitud una vez más, no se recibe un reconocimiento válido, la estación maestra se sitúa en time out y sale hacia un procedimiento de recuperación de error.

La esclava transmitirá NAK cuando reciba un mensaje con error de paridad o de verificación de bloque, con esto solicita la repetición de la transmisión. Este proceso se repite un número específico de veces que puede ser cero, y si las pruebas aún no son satisfactorias, la unidad maestra sale hacia un procedimiento de recuperación de error.

Si el punto A no recibe la condición de estado maestro

o no recibe una respuesta válida a su mensaje, entonces transmite un EOT y regresa al estado inactivo.

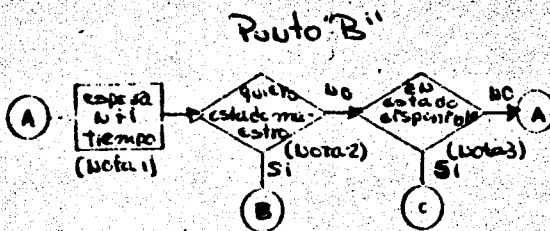
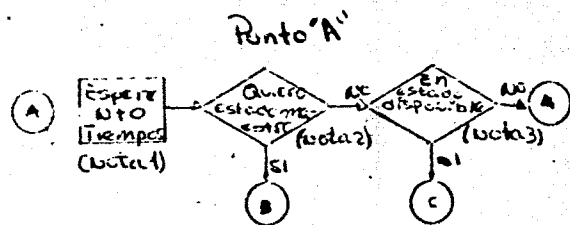


DIAGRAMA 3.7 DISCIPLINA POINT-TO-POINT CONTENTION (1 DE 2)

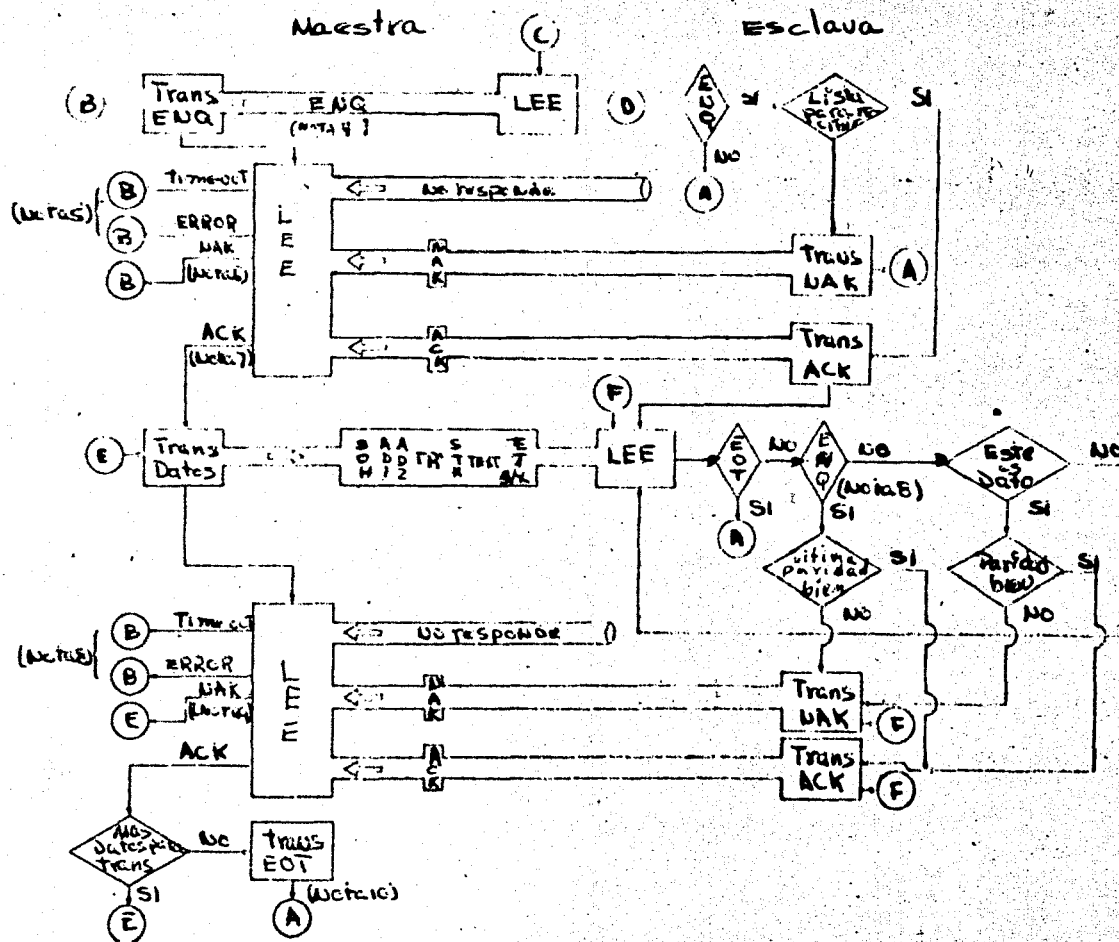


DIAGRAMA 3.7 DISCIPLINA POINT-TO-POINT CONTENTION (2 DE 2)

NOTAS DE LA DISCIPLINA DE LINEA

"POINT-TO-POINT CONTENTION"

- 1) La condición de disponible en un enlace de comunicación, es la continuación de la terminación de una transmisión previa.
- 2) El deseo de poseer estado maestro es usualmente cuando la terminal tiene algún dato para transmitir.
- 3) El reconocimiento cuando una estación ofrece el estado maestro es generalmente cuando se acepta un ENQ.
- 4) Cuando una estación desea el estado maestro, envía un ENQ y empezará una función de time out normalmente automática.
- 5) En caso de no tener respuesta a un inicial ENQ o una contestación inválida, la estación tendrá el estado maestro.
- 6) Cuando se recibe un NAK la estación que quiere el estado maestro puede reiniciar su ofrecimiento enviando un ENQ de nuevo.
- 7) Si la estación B está lista para recibir, ésta puede enviar un prefijo seguido por ACK. (El prefijo puede incluir dirección, estado, etc). La estación A detectará la respuesta afirmativa, asumirá el estado maestro y procederá con la transmisión de mensajes.

- 8) Si la estación maestra recibe una respuesta inválida o no recibe respuesta a un bloque de transmisión, ésta puede enviar un requerimiento de respuesta ENQ.
- 9) Si la esclava recibe un mensaje con paridad en caracteres, chequeo de bloque o fallas en la dirección de la estación, transmitirá un NAK para que le envíen de nuevo el mensaje.
- 10) Cuando una transmisión de maestra a esclava es terminada, la maestra puede reelegirse maestra por enviar un EOT.

3.5.8 DISCIPLINA DE LINEA

"POINT-TO-POINT SWITCHED CONTENTION" (ver diag 3.8)

Este procedimiento se aplica cuando son dos estaciones sobre un enlace conmutado punto a punto, con ninguna estación designada como la estación maestra.

Ambas estaciones contienden por el estado maestro y una puede apoderarse del estado maestro bajo la condición que la otra no se haya apoderado antes de éste.

Diferentes intervalos de tiempos-fuera pueden ser empleados en cada estación para reintentar obtener el estado maestro en simultaneas tentativas; inicialmente no tenemos contention.

Una función contention determina la relación maestra/esclava de dos estaciones. Una terminación hace funcionar el regreso a condición contention.

DISCIPLINA

Cuando una estación quiere obtener el estado maestro, transmite el caracter ENQ y empieza la función time out normalmente automática pero que puede ser manual. Este ENQ puede tener un prefijo opcional (e.g. identidad, dispositivo, etc).

Para resolver ofrecimientos simultáneos por ambas estaciones después de tener el ofrecimiento para el estado maestro, la estación con el intervalo más largo de time out reaccionará a una recepción de un ENQ como si ésta no hubiera querido el estado maestro.

Cada estación reintentará el querer obtener el estado maestro, cuando la relación maestra/esclava esté muerta (i.e. fuera de servicio).

En caso de una respuesta inválida o no respuesta a el inicial ENQ, la estación que quiere el estado maestro reintentará el envío del ENQ para obtener el estado maestro. La estación puede reintentar el querer el estado maestro n

veces (Definido por cada sistema particular y donde n puede ser igual a cero).

Después de n intentos desafortunados de tener el estado maestro, la estación puede salirse al proceso de recuperación del error apropiado.

Si la estación B (a la que se le está enviando el carácter ENQ) no está lista para recibir, enviará un carácter NAK o puede enviar un prefijo opcional en vez del NAK.

Cuando la estación A detecta el NAK, puede intentar de nuevo para conseguir el estado maestro. En caso de que en el número de reintentos que haya tenido no hubiera obtenido el estado maestro, procederá a una desconexión obligatoria a salirse a un error apropiado para un proceso de recuperación.

Pero si la estación B está lista para recibir, ésta puede enviar un prefijo seguido por un ACK (el prefijo puede incluir dirección, status, etc). Cuando la estación A detecta la respuesta afirmativa, asume el estado maestro y procede con la transmisión de mensajes.

Si la estación maestra recibe una respuesta inválida o no responde a un bloque de transmisión, ésta puede enviar

un caracter ENQ pidiendo respuesta. Este puede ser enviado n veces (definido por cada sistema particular y donde n puede ser igual a cero). Nótese que este uso de ENQ, puede resultar pérdida o duplicación de transmisión de bloques. Una forma de evitar este problema es numerando respuestas, numerando bloques o ambos. Si después de repetir el caracter ENQ n veces a válidos acusos de recibo no es recibida respuesta, la maestra hace time out y sale a una recuperación de error de proceso.

Si la esclava recibe un mensaje con paridad en caracteres, checar bloque o falla en dirección de terminal, puede transmitir NAK, para llamar a repetir la transmisión. Esta puede ser repetida n tiempos (definida por cada sistema particular, y n puede ser igual a cero).

Después de n tentativas, la estación maestra puede salir a una recuperación de error de proceso.

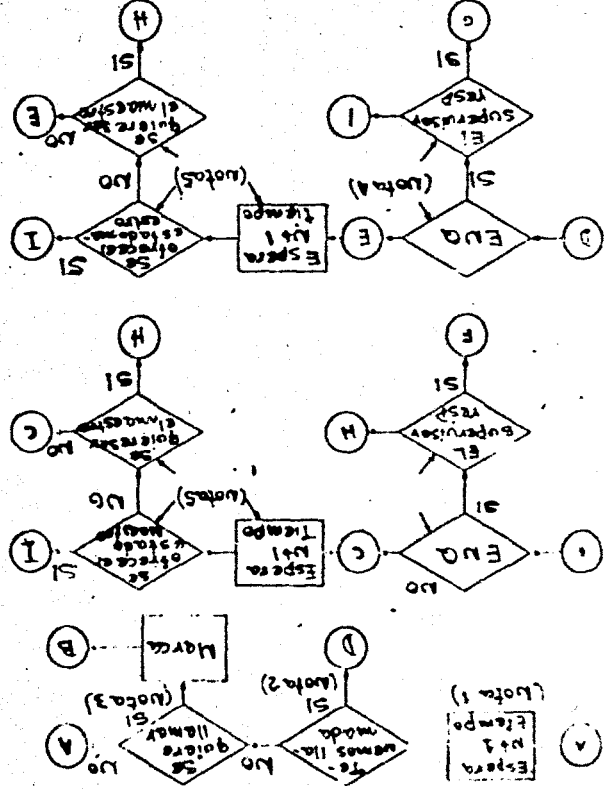
Cuando una transmisión de la estación maestra (la que llama) a la estación esclava es completado, la estación que llama puede dejar el estado maestro a la estación llamada enviando EOT. Luego, intercambiando EOT_s, el estado maestro puede continuar a ser intercambiado entre las dos estaciones hasta que ninguna de las estaciones tenga algo que

transmitir.

En conclusión de la operación de transferencia de mensajes, la estación maestra puede evitar la función "over to you" (EOT) y proceder directamente a la desconexión obligatoria (DEOT).

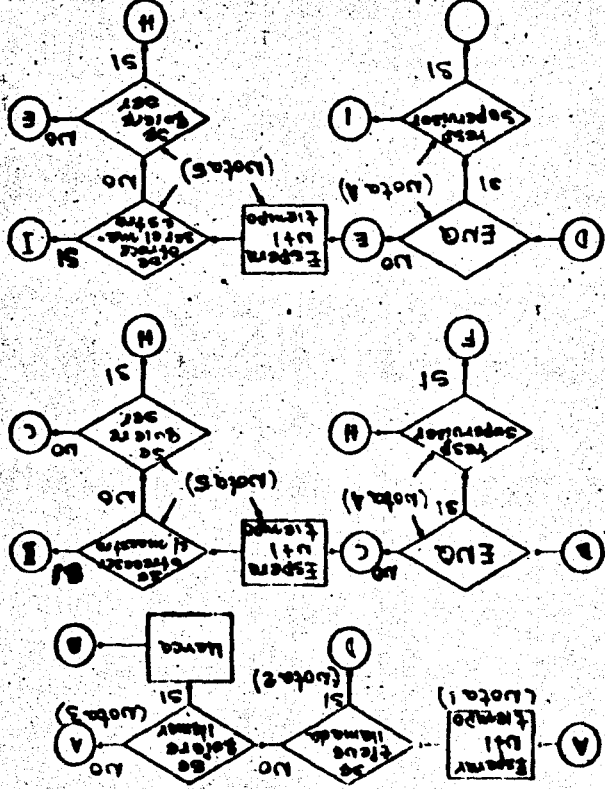
Si la nueva estación maestra tiene un mensaje para la nueva estación esclava, ésta puede proceder a iniciar la función de identificación descrita antes, sobre la determinación de prontitud de recibir mensajes de la nueva estación esclava; ésta puede proceder directamente al proceso de transferencia de mensajes. Si la nueva estación maestra no tiene datos para la nueva estación esclava o si ésta no desea ser la estación maestra, ésta iniciará la función de desconexión obligatoria (DEOT).

DIAGRAMA 3.8 DISCIPLINA POINT-TO-POINT SWITCHED CONTINION (1 DE 3)



Punto A

Рубто В



Maestra

Esclava

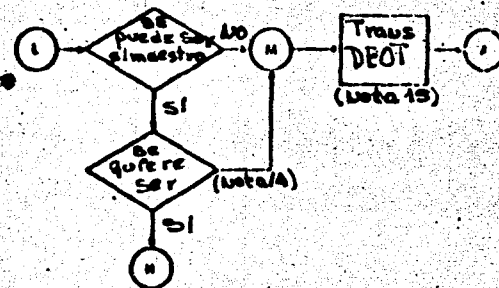
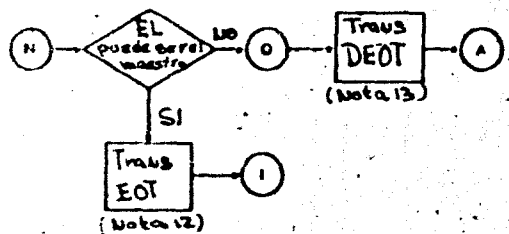
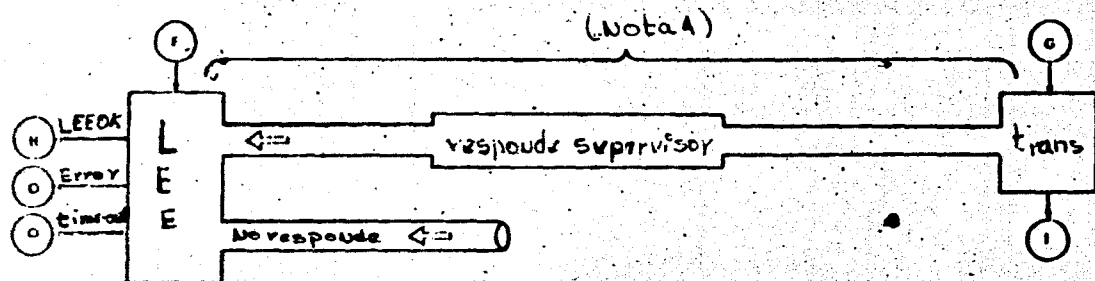


DIAGRAMA 3.8 DISCIPLINA POINT-TO-POINT SWITCHED CONTENTION (2 DE 3).



DIAGRAMA 3.8. DISCIPLINA POINT-TO-POINT SWITCHED CONTENTION (3 DE 3)

NOTAS DE LA DISCIPLINA DE LINEA

"POINT-TO-POINT SWITCHED CONTENTION"

- 1) El intercambio de tiempos de espera dentro del estado ocioso es un intento para preveer llamadas telefónicas simultáneas.
- 2) El reconocimiento de una llamada telefónica es generalmente realizada por la generación de una interrupción en la estación llamada.
- 3) El deseo de tener el estado maestro es, dado por el deseo de transmitir datos.
- 4) En algunos sistemas, por acuerdo previo, el establecimiento del circuito de comunicación es considerado una implicación de ENQ y la estación llamada automáticamente transmite una respuesta indicando que el circuito está hecho y el enlace de comunicación también.
- 5) Después de terminada una comunicación, las dos estaciones pueden pelear por el estado maestro.
- 6) Cuando una estación desea transmitir le enviará a la otra un ENQ y esperará un tiempo determinado.
- 7) En caso de una inválida respuesta o no respuesta a un inicial ENQ, la estación que quiere el estado maestro enviará de nuevo un ENQ.
- 8) Si la estación B no está lista para transmitir, enviará un prefijo opcional seguido por un NAK. La es

tación A, detecta el NAK, y puede tener de nuevo el estado maestro.

- 9) Si la estación B está lista para recibir, ésta puede enviar un prefijo seguido por un ACK (el prefijo puede incluir dirección, estado, etc). La estación A detectará la respuesta afirmativa, asumirá el estado maestro y procederá a la transmisión de mensajes.
- 10) Si la estación maestra recibe una inválida respuesta o no recibe respuesta a la transmisión de un bloque, ésta exige una respuesta a su ENQ.
- 11) Si la esclava recibe el mensaje con algún problema, transmitirá un NAK, para pedir de nuevo el mensaje.
- 12) Después de que se termine una transmisión, la esclava puede pelear por ser la que tenga el estado maestro.
- 13) Después de completar la operación de transferencia de mensajes, la maestra puede enviar un EOT y proceder a una desconexión.
- 14) Si la nueva estación maestra tiene un mensaje para la nueva estación esclava, ésta puede proceder a transferir mensajes.
- 15) Si la nueva estación maestra no tiene datos para la nueva esclava o si no desea seguir siendo la maestra iniciará una función de desconexión.

3.5.9 DISCIPLINA DE LINEA

POINT-TO-POINT MODO "BATCH" (ver diag 3.9)

Modo Batch es una extensión de comunicación punto a punto, con la diferencia que se estarán transmitiendo archivos muy grandes.

Por ejemplo, cuando se mete un programa fuente, en una estación remota (RJE), para ser compilado en la central y que el código objeto regrese a dicha estación.

En una conexión conmutada, la estación que llama es considerada la estación maestra por la inicial transacción y ésta enviará un ENQ siguiendo inmediatamente a entablar la conexión. En una conexión arrendada, ninguna estación puede obtener el estado maestro por enviar ENQ. Como ambas enviarían simultáneamente, diferentes valores de time out son asignados para evitar conflictos repetidos.

Cuando la estación que quiere el estado maestro transmite el ENQ y no recibe respuesta, ésta puede retransmitir n veces el caracter ENQ. En caso de líneas conmutadas, la conexión deberá ser desconectada y restablecida para hacer una nueva llamada. En caso de líneas arrendadas, otro proceso de recuperación será requerido.

Cuando la estación que quiere el estado maestro recibe n veces NAK de la estación esclava después de haber transmitido n veces el ENQ, ésta procederá a transmitir DEOT (DLEEOT), y en ese momento se desconectará el circuito.

Cuando se transmite el ENQ de parte de la estación que quiere el estado maestro a la estación esclava y ésta recibe otro carácter que no es el ENQ, la esclava regresará al estado de lectura. Para conexión de líneas conmutadas la que llama, o maestra, eventualmente colgará la línea.

Si la estación maestra no tiene más bloques para entonces enviará EOT; con esto ofrece el estado maestro a la otra estación.

Por otro lado cuando la estación maestra no quiere más comunicación con la estación esclava, (en líneas conmutadas) la estación maestra enviará DLEEOT para desconectar el circuito.

Podrá proceder al texto que envía la maestra un encabezado opcional indicando dirección de la estación a la que va el mensaje y número de mensaje. Si esta opción no es ejercitada, esta información puede ser incluida en el texto.

Cuando hay un fracaso de la estación maestra por no re

cibir ACK o NAK después de haber enviado una transmisión de mensajes, causará la transmisión de un ENQ. A esa petición, la esclava repetirá el último carácter de control enviado. Si después de n reintentos la condición no es correcta, otro proceso de recuperación será requerido.

Si la estación esclava quiere suspender la transmisión batch con la maestra, ésta enviará interrupción inversa (DLE < ACK) en lugar de un reconocimiento positivo. Esto, será interpretado por la estación maestra como un ACK y ésta puede ceder el estado maestro aún cuando queden bloques para ser transmitidos. Esto será acordado por previo arreglo.

Si en algún caso la estación maestra no puede ser capaz de pedir cambio de estado maestro y pueda pasar el próximo bloque, ésta cancelará la petición de interrupción inversa y será reiniciada por la esclava en el próximo punto lógico, en el proceso de transferencia de datos.

Punto "A"

Punto "B"

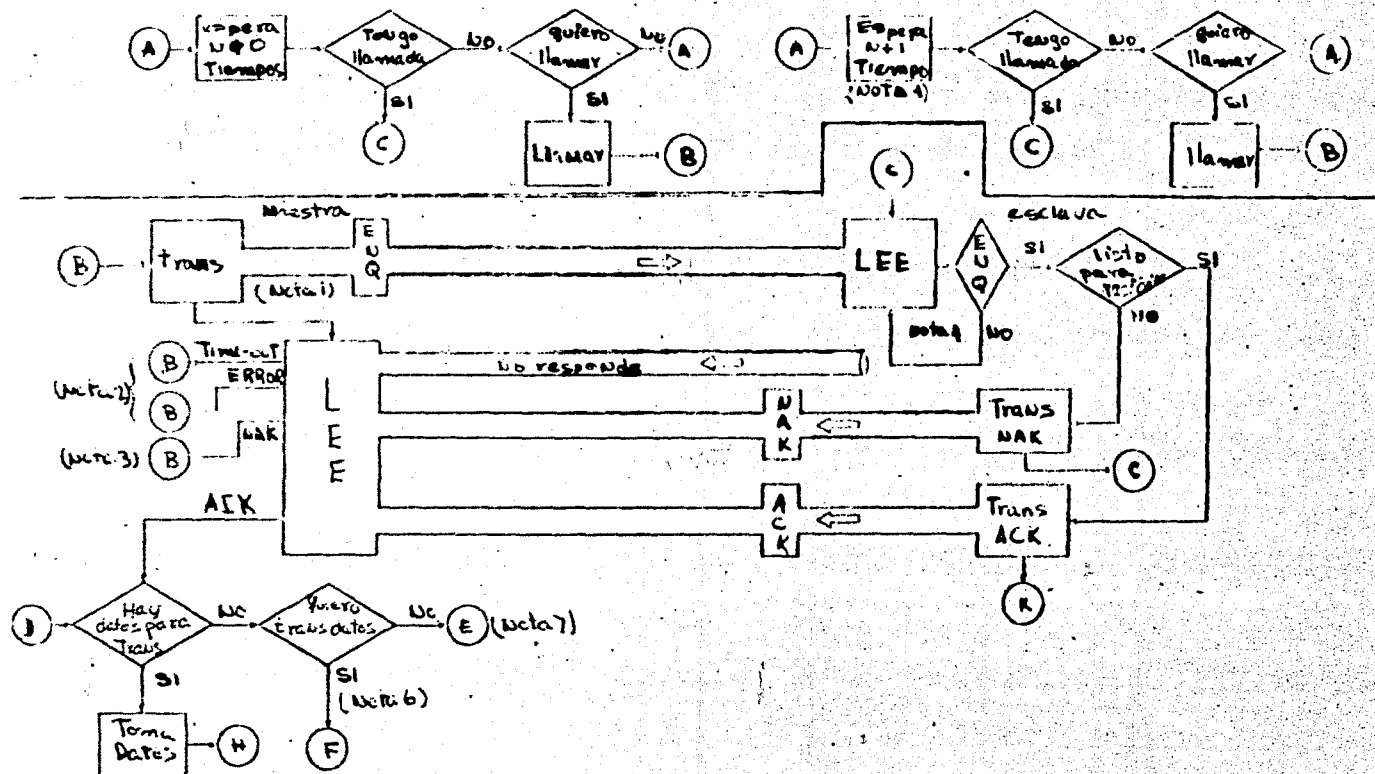


DIAGRAMA 3.9 DISCIPLINA MODO BATCH (1 DE 3)

Esclava

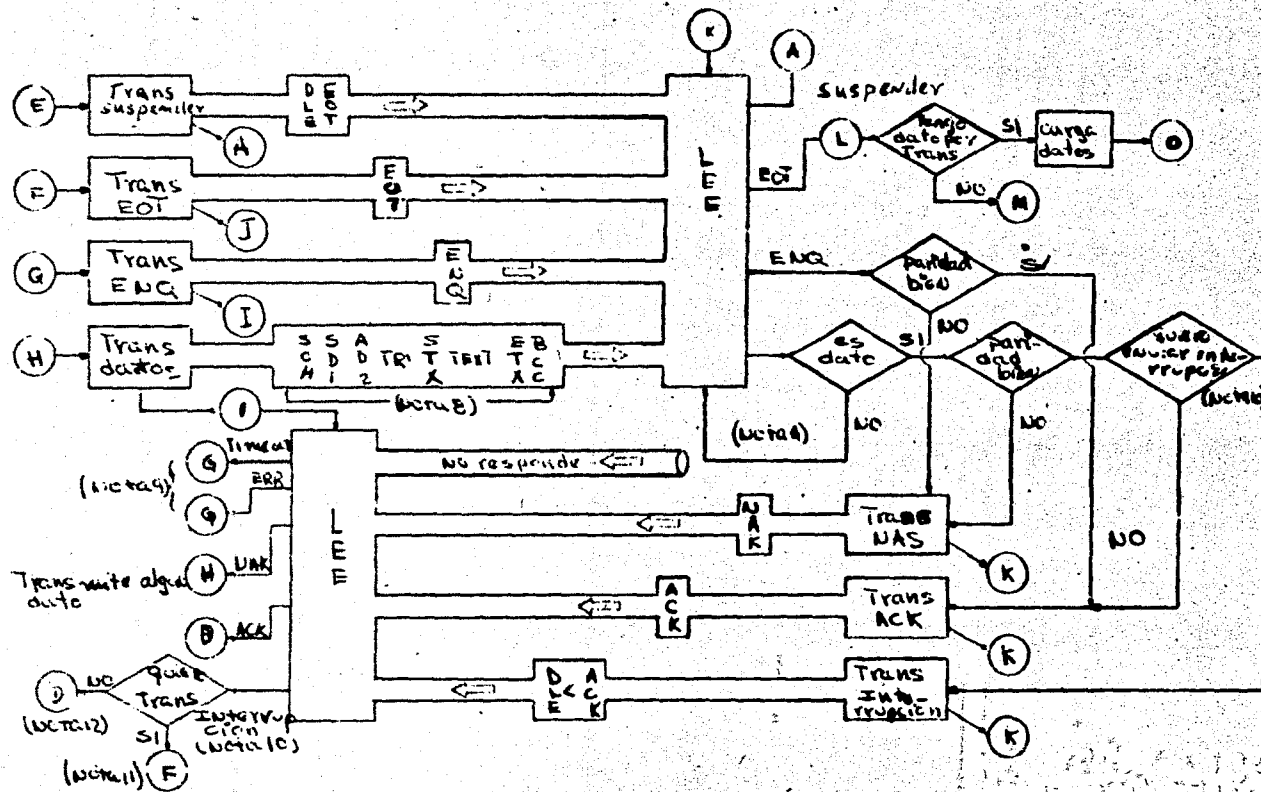


DIAGRAMA 3.9 DISCIPLINA MODO BATCH (2 DE 3)

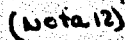


DIAGRAMA 3.9 DISCIPLINA MODO BATCH (3 DE 3)

NOTAS DISCIPLINA DE LINEA MODO "BATCH"

- 1) En conexiones conmutadas, la estación que llama es la que es considerada como la estación maestra y enviará un ENQ para establecer la conexión.
- 2) Si se recibe una respuesta inválida o no hay respuesta a un ENQ para obtener el estado maestro, la estación que envió el ENQ puede enviarlo n veces más.
- 3) Cuando un NAK es recibido n veces, por enviar ENQ una conexión conmutada, la estación que llama transmite "DLEEOT" desconectando el circuito.
- 4) Cuando se tenga condición de error o problema de transmisión, la esclava regresará al estado de read.
- 5) Esta condición es la misma que la nota 4 excepto que la esclava fué la maestra original y controlará la conexión conmutada.
- 6) Si la estación no tiene más bloques para enviar, transmitirá un EOT, ofreciendo el estado maestro.
- 7) Si no hay la conexión deseada, la maestra original enviará un "DLEEOT" para desconectar el circuito.
- 8) La maestra puede poner antes del texto un opcional encabezado, indicando la dirección de la estación a la cual el mensaje va a ser enviado, como también el número del mensaje.

- 9) Cuando llega a haber una falla en la maestra para recibir información, esta transmitirá a la esclava un ENQ, para que le sea repetido el mensaje.
- 10) Si la esclava quiere suspender la transmisión de la maestra, mandará un (DLE < ACK) en lugar de un ACK, la maestra retendrá el bloque de transmisión.
- 11) La maestra concede requerimiento para interrupción y deja el estado maestro a la otra estación.
- 12) En algunos casos la maestra no puede conceder requerimientos y pasa el próximo bloque. Ella (la maestra) cancela la interrupción.

3.5.10 DISCIPLINA DE LINEA

"SEQUENTIAL SELECT" (ver diag 3.10)

INTRODUCCION

Este proceso puede ser deseable para implementar selección secuencial de un número de dispositivos esclavos sobre una línea específica, en vez de selección de grupo. Un plan de direccionamiento enlazado es usado para cubrir esta posibilidad. La última esclava seleccionada reconoce la elección y el mensaje de datos.

DISCIPLINA

La estación maestra transmite un EOT, que puede ser la terminación de una previa transmisión de secuencia. Para minimizar el efecto de ruido, la selección secuencial ordenada puede seguir inmediatamente.

Si una esclava recibe y reconoce un EOT seguido por las direcciones (AD1, AD2) y SEL, ésta no responderá al mensaje de selección secuencial pero recibirá el mensaje con el SOH. La estación maestra puede programáticamente reelegir n veces (definido por cada sistema particular y n puede ser igual a cero). Después de n reintentos fallidos, será ejecutado el proceso de recuperación de error apropiado.

Si la esclava no está lista para recibir, transmitirá un NAK, la maestra normalizará y reintentará la selección de la esclava en una secuencia conveniente; sin embargo, para alguna instalación, esto puede desearse para repetir la selección ordenada inmediata.

Habría un error de condición si la estación maestra recibe otro carácter que no sea ACK o NAK, o responderá con un time out a un mensaje sequential select. La estación maestra puede programáticamente reelegir n veces (n

puede ser cero). Después de n reintentos fallidos se ejecutará un proceso de recuperación de error.

Una respuesta ACK a un mensaje sequential select indica que la esclava recibió satisfactoriamente y está en estado listo para recibir. La maestra puede ahora transmitir el mensaje de datos.

Si la estación maestra no recibe una respuesta (ACK o NAK) a un mensaje, ésta hace time out y programáticamente retransmitirá n veces el dato (donde n está definido por cada sistema particular y n puede ser cero).

Después de n reintentos fallidos, será ejecutado un proceso de recuperación apropiado. La maestra retendrá el mensaje para transmitir en la próxima selección secuencial a esas esclavas.

Si existe paridad de caracteres, chequeo de bloque o la identificación de esclava no es válida por la esclava responsable, ésta enviará NAK. La maestra puede retransmitir el dato n veces (donde n puede ser cero).

Después de n reintentos fallidos, se ejecutará un proceso de recuperación apropiado. La maestra reintentará la transmisión en la próxima selección secuencial a esas esclavas.

Cuando la esclava responda a un mensaje con un ACK, es to es una indicación que el mensaje fue recibido apropiadamente, y que no tuvo error de paridad y de bloque. La EOT (puede ser el principio de otro select o poll) para completar el ciclo sequential select. Después de haber terminado el ciclo, en este punto, la estación esclava puede checar para otro caracter de control tal como POLL, SEL, FSL o BSL, pues el formato de mensaje de esos caracteres es el mismo que los mensajes de selección secuencial.

Maestra

Esclava (5)

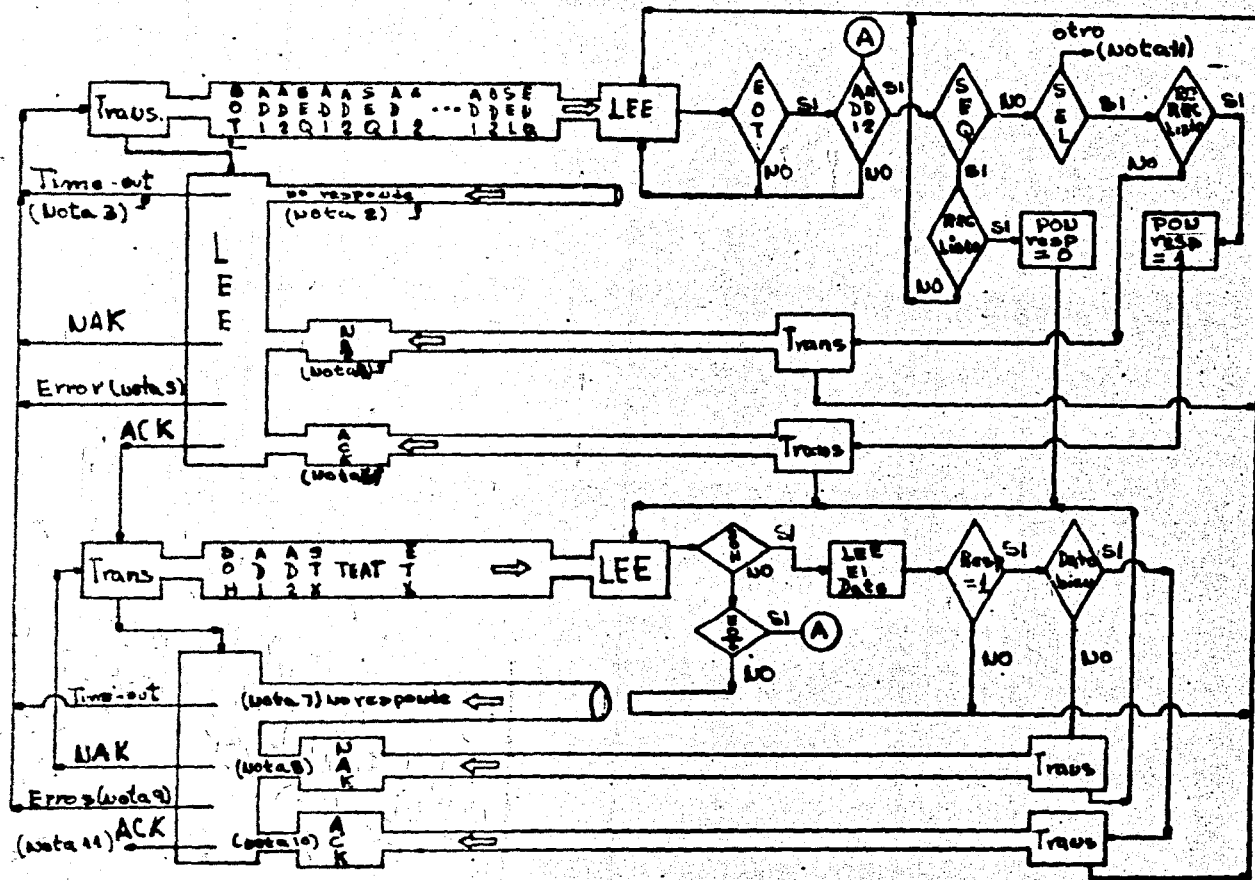


DIAGRAMA 3.10 DISCIPLINA SEQUENTIAL SELECT

NOTAS DE LA DISCIPLINA "SEQUENTIAL SELECT"

- 1) Este EOT pudo haber sido la terminación de una transmisión previa.
- 2) Si la esclava recibe y reconoce un EOT seguido por su dirección (AD1, AD2) y un SEQ, ella no responde al mensaje sequential select, pero recibirá el mensaje que principie con el SOH.
- 3) El time out en el maestro es causado por la esclava, por no responder a un caracter de selección secuencial con un NAK o ACK.
- 4) Si la esclava no está lista para recibir, como indica un caracter NAK, la maestra retransmitirá la selección de la esclava en la secuencia apropiada.
- 5) Existe una condición de error si la maestra recibe alguna otra cosa diferente a un ACK, NAK o time out de un mensaje de selección secuencial.
- 6) Una respuesta de ACK a un mensaje de selección secuencial, indica que la esclava está lista para recibir y que recibió bien.
- 7) Si la maestra no recibe o responde (ACK o NAK) a su mensaje, ella puede tener un tiempo fuera y programáticamente retransmitir el dato.
- 8) Si los caracteres de paridad, chequeo de bloque o la identificación de la esclava no son válidos para la esclava responsable, ella transmitirá un NAK.

La maestra puede retransmitir n veces.

- 9) Existe una condición de error si la maestra recibe alguna otra cosa diferente a un NAK, ACK o time out a un mensaje de datos.
- 10) Cuando la esclava responde a un mensaje de datos con un ACK, éste es un indicador que el mensaje fue recibido apropiadamente, la maestra puede ahora transmitir un EOT para completar el ciclo de la selección secuencial.
- 11) En este punto la esclava puede checar otro caracter de control tal como un POL, SEL, FSL o BSL como también el mismo mensaje de selección secuencial.

3.5.11 DISCIPLINA DE LINEA

"POINT-TO-POINT NO-CONTENTION"

(Línea privada, solo envía) (ver diag 3.11)

Este proceso se aplica cuando la estación maestra envía un mensaje a la estación esclava. Este procedimiento puede también aplicarse a líneas switcheadas, después de haber hecho la conexión y la relación maestra-esclava queda establecida.

Si la maestra tiene algo que transmitir entonces enviará el principio de cabeza, la dirección, el principio de texto, el texto, fin de texto y el BCC, y en caso contrario

estará siempre en espera de poder transmitir.

Si cuando transmitió el bloque, recibe una respuesta in válida o no hay respuesta, ésta puede enviar un carácter ENQ para solicitud de respuesta. Esta puede ser enviada n veces (n es definido por cada sistema particular y n puede ser igual a cero). Nótese que este uso de ENQ puede resultar una pérdida o duplicación de bloques de transmisión. Una forma de abolir estos problemas es numerando respuestas o numerando bloques o ambas. Si después de repetir el carácter ENQ n veces, no es recibido un válido reconocimiento, la estación maestra puede salirse a un proceso de recuperación de error.

Ahora, si la esclava recibe un mensaje pero con paridad de caracteres, paridad en bloque o falla en domicilio de terminal, transmitirá un carácter NAK para llamar a repetir la transmisión. Este puede ser repetido n veces. (n puede ser cero). Después de n intentos, la maestra puede salirse a un proceso de recuperación.

Cuando transmite un EOT significa el fin de un cambio de datos y regreso a control a un estado ocioso. Sin embar go, esto no da lugar a un medio de contention.

Esclava

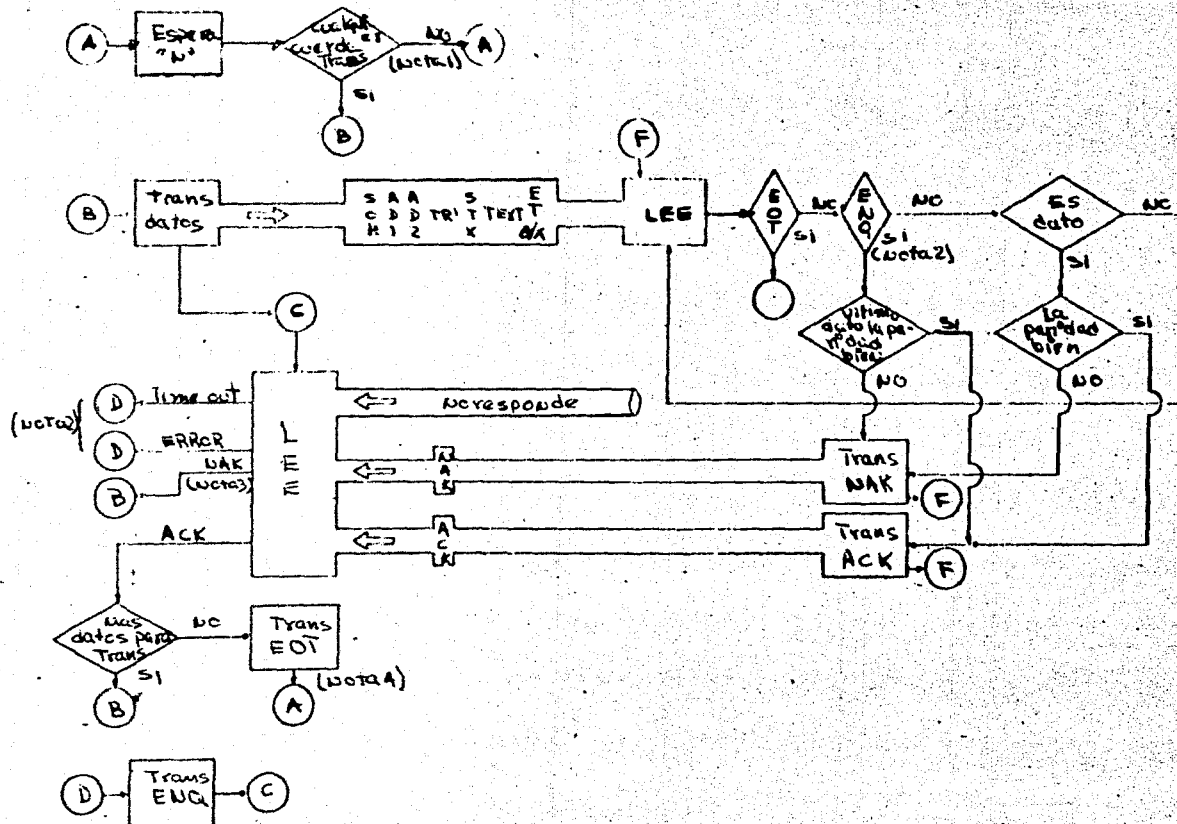


DIAGRAMA 3.11 DISCIPLINA POINT-TO-POINT NO-CONTENTION (LINEA PRIVADA, SOLO ENVIA)

NOTAS DE LA DISCIPLINA "POINT-TO-POINT NO-CONTENTION"
(LINEA PRIVADA, SOLO ENVIA)

- 1) El deseo de transmitir datos está usualmente basado en el término de algún programa o algún estímulo externo; por ejemplo, la disponibilidad de un archivo a ser transmitido.
- 2) Si la maestra no recibe respuesta a una válida transmisión de un bloque, esta puede pedir respuesta al caracter ENQ. Este caracter puede ser enviado n veces.
- 3) Si la esclava recibe un mensaje con paridad en caracteres, chequeo de bloque o fallas en la dirección de la terminal, ésta transmitirá un caracter NAK para que le sea transmitida la información una vez más.
- 4) La transmisión de un EOT significa el fin de un intercambio de datos y regresar al control a un estado disponible.

3.5.12 DISCIPLINA DE LINEA

"POINT-TO-POINT NO-CONTENTION"

(LINEA PRIVADA, SOLO RECIBE) (ver diag 3.12)

Este procedimiento se aplica cuando la estación de control solicita un mensaje de otra estación. El procedimiento puede también aplicarse a líneas conmutadas; una vez hecha la conexión, la relación maestra-esclava queda establecida.

La estación de control transmite un ENQ, que es para preguntar a la otra estación si tiene un mensaje para transmitir.

Ahora si la estación de control recibe una inválida respuesta o no responde al caracter ENQ, ésta puede retransmitir el caracter ENQ n veces (donde n puede ser cero). Después de n tentativas, la de control puede salirse a un procedimiento de recuperación de error.

Cuando no llega a haber respuesta a un caracter ENQ que le envió la estación de control a la esclava, se genera un time out. Si la otra estación tiene datos para enviar, ella asume el estado maestro y empezará a transmitir.

Cuando llega a responder con EOT la estación esclava a

un caracter ENQ, indica al control que la otra estación no tiene datos para transmitir, y no por eso pasará el estado maestro a la otra estación.

Si la estación maestra recibe una respuesta inválida o no respuesta a un bloque de transmisión, ésta puede enviar el caracter ENQ que indica al control que la otra estación no tiene datos para transmitir, y no por eso pasará el estado maestro a la otra estación.

Si la estación maestra recibe una respuesta inválida o no respuesta a un bloque de transmisión, ésta puede enviar el caracter ENQ requiriendo respuesta. Este puede ser enviado n veces (n puede ser cero). Nótese que, de este uso de ENQ, puede resultar la pérdida o duplicación de bloques de transmisión. Un medio de abolir estos problemas es el uso alternado de respuestas numeradas o bloques numerados o ambos. Si después de repetir el caracter ENQ n veces una válida respuesta no es recibida, la maestra cesará de transmitir ENQ.

Si se llega a presentar una condición de error, i.e., una transmisión mutilada o time out excesivo, la estación maestra regresará a el estado de lectura. Este loop da oportunidad para discontinuar la transmisión y/o entrar a algún tipo de recuperación de error.

Si la esclava recibe un mensaje con paridad en caracteres, chequeo de bloque o falla en la dirección de la terminal, podrá ser transmitido el caracter NAK, para llamar a repetir la transmisión. Esta puede ser transmitida n veces (n puede ser cero). Después de n intentos, la esclava puede salirse a un procedimiento de recuperación de error.

Esclava

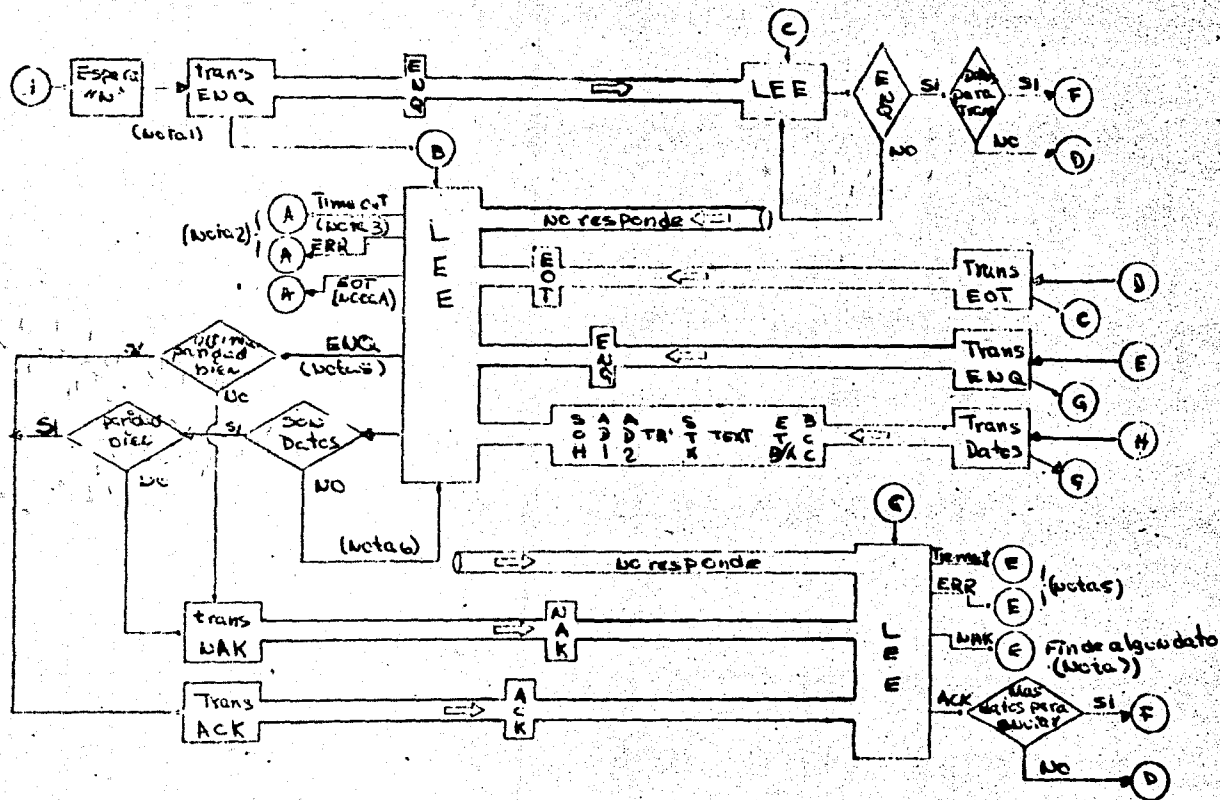


DIAGRAMA 3.12 DISCIPLINA POINT-TO-POINT NO CONTENTION (LINEA PRIVADA, SOLO RECIBE)

NOTAS DE DISCIPLINA "POINT-TO-POINT NO-CONTENTION"

(LINEA PRIVADA, SOLO RECIBE)

- 1) La transmisión de un ENQ, desde el control, es para pedir a la estación si tiene un mensaje para transmitir.
- 2) Si el control recibe una respuesta inválida o no hay respuesta a un caracter ENQ, ésta puede retransmitir el ENQ n veces.
- 3) Esta función de time out es sólo aplicable después de la transmisión del ENQ, desde el control a la esclava. Si la otra estación tiene datos a enviar, ésta asume el estado maestro y empieza a transmitir.
- 4) Una respuesta de EOT a un ENQ indica al control que la otra estación no tiene datos a transmitir y no desea asumir estado maestro.
- 5) Si la estación maestra recibe una respuesta inválida o no responde a una transmisión de bloque, ésta puede repetir la transmisión de ENQ para que se le de una respuesta. Si después de repetir el caracter ENQ n veces, un ACK no es recibido, la maestra cesará de transmitir ENQ.
- 6) Dentro de una condición de error, por pérdida de transmisión, o excesivos time out, la maestra regresará al estado de lectura. Este loop da al control la oportunidad para discontinuar la transmisión y/o

entrar a algún proceso de recuperación de error.

- 7) Si la esclava recibe un mensaje con paridad en caracteres, chequeo de bloque, o falla de dirección, ésta transmitirá un NAK para indicar que le sea repetida la transmisión. Después de n intentos fallidos, la esclava entrará a un proceso de recuperación de error.

4. PROGRAMA DE APLICACION

En este capítulo veremos un ejemplo de disciplina de línea y control para un dispositivo. Esta combinación en la B-6700 es llamada "Request".

El dispositivo que utilizaremos para el ejemplo es una pantalla de video que además es direccionable, por lo tanto utilizaremos disciplinas de línea que tomen en cuenta que en una línea puede haber más de una estación.

Como vimos en el capítulo anterior, necesitamos dos disciplinas y un control para el protocolo, una que recibe y otra que transmite. Para esto utilizaremos la disciplina de "Polling" para recibir, la de "Selecting" para transmi-

tir y el "control poll" para el control.

Este "Request" queda como sigue:



```

1000 *****
2000
3000
4000
5000
6000
7000
8000
9000
CONTROL POLL:
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
CONTROL DE ULTIMA ESTACION
LINE(TALLY(0)) = LINE(TALLY(1))+1.
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
VARIABLE DE UN BIT QUE NOS SIRVE PARA SABER CUANDO
UN MENSAJE A SIDO ENCOLADO POR ALGUNA ESTACION EN
LA LINEA. ESTA VARIABLE ES VERDADERA CUANDO UN MENSAJE
HA SIDO INSERTADO EN LA COLA DE ESTACION VACIA
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
LINE(QUEUED) = TRUE.
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
STATION ES UNA VARIABLE DE UN BYTE USADA PARA SELECCIONAR
UNA PARTICULAR ESTACION CONFORME AL VALOR ASOCIADO AL IN-
DICE DE LA ESTACION SOBRE LA LINEA
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
IF STATION > 0 THEN
  EGCTH
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
PAUSA QUE SIRVE PARA QUE EL DCP PUEDA VER OTRAS ESTA-
CIONES (LA DURACION DE LA PAUSA ES DE 3.4 MICROSEGUNDOS)
Y DESPUES PARA A LA SIGUIENTE INSTRUCCION
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
PAUSE
STATION = STATION-1.
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
QUE LA ESTACION SEA VALIDA ESTA EN FUNCION DEL VALOR
ASOCIADO AL INDICE DE LA ESTACION QUE FUE ALMACENADO
EN LA VARIABLE STATION AL MOMENTO DE INICIALIZAR EL DCP.
DESPUES SE PREGUNTA SI LA ESTACION ESTA LISTA (PUERLE NO ES-
TA EN CASO DE UNA TERMINACION DE ERROR EN LL REQUEST).
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
IF STATION(VALID) THEN
  IF STATION(READY) THEN
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
SE PREGUNTA SI HAY MENSAJES EN LA COLA DE LA ESTACION
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
IF STATION(QUEUED) THEN
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
SE MANDA INICIAR EL REQUEST DE TRANSMIT
INITIATE REQUEST.
0000
1000
2000
3000
4000
5000
6000
7000
8000
9000

```

```

00001000
00002000
00003000
00004000
00005000
00006000
00007000
00008000
00009000
00010000
00011000
00012000
00013000
00014000
00015000
00016000
00017000
00018000
00019000
00020000
00021000
00022000
00023000
00024000
00025000
00026000
00027000
00028000
00029000
00030000
00031000
00032000
00033000
00034000
00035000
00036000
00037000
00038000
00039000
00040000
00041000
00042000
00043000
00044000
00045000
00046000
00047000
00048000
00049000
00050000
00051000
00052000
00053000
00054000
00055000
00056000
00057000
00058000

```

1 SF PREGUNTA POR EL ESTADO DE LA ESTACION

IF STATION(ENABLED) THEN

2 SF PREGUNTA POR EL NUMERO DE RETRANSMISION DE LA
3 ESTACION

IF STATION(TALLY)>LINE(TALLY(1)) THEN
BEGIN

STATION(TALLY) = STATION(TALLY)-
LINE(TALLY(1)).
LINE(QUEUED) = TRUE.

END
ELSEF

4 SF PREGUNTA SI HAY ESPACIO, SI LO HAY INDICA QUE UN
MENSAJE HA SIDO ENCOLADO POR ALGUNA ESTACION.

IF NOSPACE THEN
BEGIN

STATION(TALLY)=0.
LINE(QUEUED)=TRUE.

END
ELSEF
BEGIN

5 SF INICIALIZA EL CONTADOR DE FRECUENCIA AL VALOR
6 MAXIMO Y SE TRANSFIERE EL CONTROL AL REQUEST DE RECEPCION

STATION(TALLY) = STATION(FREQUENCY).
INITIATE ENABLEINPUT.

END.

END.
GO TO 0.

7 SF ASIGNA A LA VARIABLE "STATION" EL VALOR ASOCIADO AL
8 INDICE DE LA ESTACION

STATION = MAXSTATIONS.

IF LINE(QUEUED) THEN

BEGIN

LINE(QUEUED) = FALSE.
IF LINE(TALLY(1))=0 THEN

BEGIN

LINE(TALLY(1)) = 1.

9 ES PUESTO FALSO EL CONTROL DE LINEA OCUPADA, PARA VER SI
10 EN LA COLA DE REQUEST HAY UN MENSAJE "DEWRITE". EN CASO
11 DE EXISTIR, SE TRANSFIERE EL CONTROL AL REQUEST
12 DE RECEPCION SI ES UN READ (TYPE=34) O AL DE TRANSMISION
13 SI ES UN WRITE (TYPE=33)

LINE(BUSY) = FALSE.
DELAY(1 SEC).
LINE(BUSY) = TRUE.

00059000.

00060000
00061000
00062000
00063000
00064000
00065000
00066000
00067000
00068000
00069000
00070000
00071000
00072000
00073000
00074000
00075000
00076000
00077000
00078000
00079000
00080000
00081000
00082000
00083000
00084000
00085000
00086000
00087000
00088000
00089000
00090000
00091000
00092000
00093000
00094000
00095000
00096000
00097000
00098000
00099000
00100000
00101000
00102000
00103000
00104000
00105000
00106000
00107000
00108000
00109000
00110000
00111000
00112000
00113000
00114000
00115000
00116000
00117000
00118000
00119000



```

END
ELSE
BEGIN
    LINE(TALLY[1]) = LINE(TALLY[0]).
    LINE(TALLY[0]) = 0.
END
GO TO 1.
END.

```

FI IDLE CAUSA A LA LINEA PONERSE EN ESTADO DE DESOCUPADA;
LA VARIABLE DE CONTROL "LINE(USY)" ES PULSTA FALSA.

indig.

DISCIPLINA DE TRANSMISION PARA DISPOSITIVO CON DIRECCION FISICA
(SELECTING)

FLOWEST SLIGHTSPAS.

2. CONDICIONES DE ERROR ETIQUETADAS

```
ERROR[0] = TIMOUT:100,  
LOSSOF CARRIER:100,  
STOPBIT:99,  
PARITY:110,  
NOFDFE:99.
```

EN CASO DE RECIBIR UN BREAK HABRA UNA TERMINACION DE FIRPOP; DICHA TERMINACION HANDBRA UN MENSAJE AL MCS Y AL MCC, HACIENDO QUE ESTA ESTACION SALGA DEL REQUEST Y SE PONGA NO-READY. EL MCS, HAS ADELANTE PONDRA DE NUEVO A LA ESTACION READY.

```
IF TOG(3) THEN
  BEGIN
    BREAK(TRANSMIT) = TRUE.
    TOG(3) = FALSE.
    TERMINATE ERROR.
  END.
```

2 ENVIA UNA PARTE DE LA DISCIPLINA "SELECTING"

INITIATE TRANSMIT.
TRANSMIT EOT.
TRANSMIT ADDRESS (RECEIVE).
TRANSMIT SFL.
TRANSMIT ENO.
FINISH TRANSMIT.

SE HANDEA AL ADAPTER CLUSTER INICIAR UN
RETAPDO; DESPUES DE FSTE, EL HARDWARE
ESTA LISTO PARA RECIBIR INFORMACION

INITIATE RECEIVE.

SE DECIDE UN CARACTER.
LA DECLACION PUEDE SER DE ALGUNA DE
LAS CONDICIONES QUE ESTAN INDICADAS

[illegible]

31000 Z ABAJO. DE SER ASI, SE IRA A LA ETI-
32000 Z QUETA CORRESPONDIENTE

33000 RECEIVE (1 SEC)

34000 [TIMEOUT:150,
35000 LOSSOFFCARDIER:100,
36000 STOPUIT:99,
37000 PARITY:110,
38000 UNFOVFL:99].

39000 Z DESPUES DE HABER ENVIADO LA PRIMERA PARTE DE
40000 LA DISCIPLINA, SE ESPERARA LA CONTESTACION DE
41000 UN "ACK" Y EN CASO DE RECIBIRSE SE TRANSMITI-
42000 RA: LOS CONTROLES Y EL TEXTO

43000 IF CHARACTER = ACK THEN
44000 BEGIN

45000 INITIATE TRANSMIT.
46000 TRANSMIT SOME
47000 INITIALIZE RCC
48000 TRANSMIT ADDRESS (RECEIVE).
49000 TRANSMIT STX.

50000 Z AL MANDAR ESTA INSTRUCCION, EL APUNTADOR DE
51000 MENSAJES DE TEXTO SE INICIALIZA EN CERO. CON
52000 ESTO, EL APUNTADOR ESTARA SEÑALANDO AL PRI-
53000 MER CARACTER DE DICHO TEXTO
54000 INITIALIZE TEXT.

55000 INITIALIZE TGC(0).

56000 Z SE PREGUNTA SI SE ESTA EN MODO SECUENCIA

57000 IF TGC(0) THEN
58000 BEGIN
59000 TGC(4) = TRUE. ZZ

60000 Z SE MANDA LIMPIAR LA PANTALLA Y SE LE DA UN
61000 Z RETARDO PARA QUE EL HARDWARE ESTE LISTO

62000 TRANSMITFFFFSPB.
63000 DELAY (500 HILLI).
64000 TALLY(2) = 1.

65000 Z SE TRANSMITE UN CARACTER DE PRINCIPIA
66000 Z CAMPO PROTEGIDO PARA SECUENCIA

67000 TRANSMIT 4"5A"

68000 11:
69000 FFTCHTENDUFRHFFER:15].
70000 IF CHAR LEO 4"F0" THEN
71000 IF CHAR LEO 4"F9" THEN
72000 BEGIN

73000 Z SE MANDA LA SECUENCIA

74000 TRANSMIT.

75000 Z CONTADOR PARA CAMPO PROTEGIDO

00181000
00182000
00183000
00184000
00185000
00186000
00187000
00188000
00189000
00190000
00191000
00192000
00193000
00194000
00195000
00196000
00197000
00198000
00199000
00200000
00201000
00202000
00203000
00204000
00205000
00206000
00207000
00208000
00209000
00210000
00211000
00212000
00213000
00214000
00215000
00216000
00217000
00218000
00219000
00220000
00221000
00222000
00223000
00224000
00225000
00226000
00227000
00228000
00229000
00230000
00231000
00232000
00233000
00234000
00235000
00236000
00237000
00238000
00239000
00240000
00241000



42000
43000
44000
45000
46000
47000
48000
49000
50000
51000
52000
53000
54000
55000
56000
57000
58000
59000
60000
61000
62000
63000
64000
65000
66000
67000
68000
69000
70000
71000
72000
73000
74000
75000
76000
77000
78000
79000
80000
81000
82000
83000
84000
85000
86000
87000
88000
89000
90000
91000
92000
93000
94000
95000
96000
97000
98000
99000
00000
01000
02000

15:

TALLY(2) = TALLY(2) + 1.
GO TO 11.

END.

IF TALLY(2) LSS 7 THEN
BEGIN

TALLY(2)=TALLY(2)+1.

TRANSMIT SP.

GO TO 15.

END.

* SE TRANSMITE EL CARACTER DE FIN DE
* CAMPO PROTEGIDO

TRANSMIT 4*4A*

TALLY(2)=TALLY(2)+1.

221:

SE PREGUNTA SI EL CARACTER EN LA LINEA
ES EL ULTIMO. DE NO SERLO, SE TRANSMITE
BLANCOS HASTA LLENARLA

IF TALLY(2) LS 80 THEN

BEGIN

TRANSMIT SP.

TALLY(2) = TALLY(2)+1.

GO TO 221.

END.

* CONTADORES TEMPORALES INICIALIZADOS

TALLY(2)=0.

TALLY(1) = 1.

18:

TALLY(1) = TALLY(1)+1.

DELAY (100 MILLI).

* SE TRANSMITE ESTOS CARACTERES PARA HACER EL
* MARGEN DE CAMPOS PROTEGIDOS Y RESERVAR LOS
* ESPACIOS PARA LOS CAMPOS NO PROTEGIDOS QUE
* SE UTILIZARAN

TRANSMIT 4*4A* SP SP SP SP SP SP SP 4*4A*.

TALLY(2)=TALLY(2) + 2.

IF TALLY(2) LS 80 THEN

BEGIN

TRANSMIT SP.

TALLY(2) = TALLY(2)+1.

GO TO 331.

END

TALLY(2)=0.

* CONTADOR QUE LLEVA EL CONTROL DE CUANTOS
* RENGLONES SE TENDRAN PARA HANDAR UN BLO-
* QUE DE INFORMACION DENTRO DEL MODO FORMATO

IF TALLY(1) LSS 12 THEN

GO TO 1A.

* ENTRA A MODO FORMATO LA TERMINAL Y MANDA
* A "HOME" EL CURSOR

00242000
00243000
00244000
00245000
00246000
00247000
00248000
00249000
00250000
00251000
00252000
00253000
00254000
00255000
00256000
00257000
00258000
00259000
00260000
00261000
00262000
00263000
00264000
00265000
00266000
00267000
00268000
00269000
00270000
00271000
00272000
00273000
00274000
00275000
00276000
00277000
00278000
00279000
00280000
00281000
00282000
00283000
00284000
00285000
00286000
00287000
00288000
00289000
00290000
00291000
00292000
00293000
00294000
00295000
00296000
00297000
00298000
00299000
00300000
00301000
00302000





```

13000 TRANSMIT ESC W.
14000 DELAY(20 MILLI).
15000 TRANSMIT DC4.
16000 DELAY(20 MILLI).
17000 GO TO 50.
18000 END.
19000
20000 * SI ESTA EN MODU SECUENCIA, LIMPIA LA PANTALLA
21000 IF TOG(4) THEN
22000 BEGIN
23000 TOG(4) = FALSE.
24000 GO TO 19.
25000 END.
26000
27000 * SI ES NUEVA PAGINA, LIMPIA LA PANTALLA
28000 IF PAGE THEN
29000 BEGIN
30000 GO TO 19.
31000 END.
32000
33000 *** SI ESTA APAGADA LA BANDERA DE PRIMER MENSAJE DE
34000 SALIDA, ESTA ES PRENDIDA Y SE VA A LIMPIAR LA
35000 PANTALLA
36000 IF NOT TOG(17) THEN
37000 BEGIN
38000 TOG(17) = TRUE.
39000 GO TO 19.
40000 END.
41000
42000 *** SI LA VARIABLE DE CONTROL DE CARRO ES VERDADERA,
43000 ENTONCES VERA EL "SKIP" PARA LIMPIAR LA PANTALLA
44000 IF PAPERMOTION THEN
45000 BEGIN
46000 IF SKIP THEN
47000 BEGIN
48000 TRANSMIT EFFSPB.
49000 DELAY (40 MILLI).
50000 TALLY(2) = 0.
51000 GO TO 22.
52000 END
53000 ELSE
54000 IF SPACE THEN
55000 BEGIN
56000
57000 *** EL SKIPCONTROL ES UTILIZADO EN CONJUNTO CON
58000 PAPERMOTION, SKIP Y SPACE PARA EL CONTROL DE
59000 CARRO. EN ESTE CASO ES UTILIZADO PARA INFOR-
60000 MAR CUANTOS SALTOS DE LINEA HAGA
61000
62000 TALLY(1) = SKIPCONTROL.
63000
64000 TALLY(1) = TALLY(1) + 1.
65000 IF TALLY(1) GT C THEN
66000 BEGIN
67000 TRANSMIT CL.
68000

```

```

00303000
00304000
00305000
00306000
00307000
00308000
00309000
00310000
00311000
00312000
00313000
00314000
00315000
00316000
00317000
00318000
00319000
00320000
00321000
00322000
00323000
00324000
00325000
00326000
00327000
00328000
00329000
00330000
00331000
00332000
00333000
00334000
00335000
00336000
00337000
00338000
00339000
00340000
00341000
00342000
00343000
00344000
00345000
00346000
00347000
00348000
00349000
00350000
00351000
00352000
00353000
00354000
00355000
00356000
00357000
00358000
00359000
00360000
00361000
00362000
00363000

```


210



GO TO 50.

IF CHAR = NUL THEN

GO TO 30.

IF CHAR = LF THEN

GO TO 30.

% EN CASO DE SER UN CR, SE LIMPIARA EL RESTO
% DE LA LINEA

IF CHAR = CR THEN

BEGIN

35:

IF TALLY(2) LS 80 THEN

BEGIN

TRANSMIT SP.

TALLY(2) = TALLY(2) + 1.

GO TO 35.

END

ELSE

TALLY(2) = 0.

END

ELSE

BEGIN

TRANSMIT.

TALLY(2) = TALLY(2) + 1.

IF TALLY(2) GT 80 THEN

TALLY(2) = TALLY(2) - 80.

END

GO TO 30.

40:

% SE LLEVA A CABO EL CONTROL DEL NUMERO DE
% CARACTERES TRANSMITIDOS EN UNA LINEA

IF TALLY(2) LS 80 THEN

BEGIN

% SE MITE BLANCOS EN EL RESTO DE LA LINEA

TRANSMIT SP.

TALLY(2) = TALLY(2) + 1.

GO TO 40.

END

ELSE

TALLY(2) = 0.

45:

% SE MANDA A LA TERMINAL SEGUIR EN ESTADO DE RECEPCION

TRANSMIT DC1.

50:

% SE TRANSMITE EL FIN DE TEXTO Y EL CHECADOR
% DE BLOQUE DE CARACTERES (BCC); SE TERMINA
% LA TRANSMISION

TRANSMIT LTY.

TRANSMIT BCC.

FINISH TRANSMIT.

% SE PREPARA PARA RECIBIR CONTESTACION A LO QUE
% TRANSMITIO. EN CASO DE SER UN "ACK", SE LIM-

00425000
00426000
00427000
00428000
00429000
00430000
00431000
00432000
00433000
00434000
00435000
00436000
00437000
00438000
00439000
00440000
00441000
00442000
00443000
00444000
00445000
00446000
00447000
00448000
00449000
00450000
00451000
00452000
00453000
00454000
00455000
00456000
00457000
00458000
00459000
00460000
00461000
00462000
00463000
00464000
00465000
00466000
00467000
00468000
00469000
00470000
00471000
00472000
00473000
00474000
00475000
00476000
00477000
00478000
00479000
00480000
00481000
00482000
00483000
00484000
00485000

1000 X PIA LA BANDERA DE LA ULTIMA ENTRADA Y LA DE
1000 RETRANSMISION. SI HACE UNA TERMINACION NORMAL
1000 X Y SE TRANSFIERE EL CONTROL

1000 INITIATE RECEIVE.
1000 RECEIVE (1 SEC) (ERROR(0)).
1000 IF CHARACTER = ACK THEN
1000 BEGIN
1000 TUG(2) = FALSE.
1000 TUG(1) = FALSE.
1000 TERMINATE NORMAL.

1000 END.

1000 X EN CASO DE RECIBIR UN CARACTER "NAK", SE PRENDE
1000 LA BANDERA DE "NAKFLAG" Y SE HARA UNA RETRANSMI-
1000 STION, SI SE ESTA DENTRO DE LOS LIMITES

1000 IF CHARACTER = NAK THEN
1000 BEGIN
1000 NAKFLAG = TRUE.
1000 TUG(2) = FALSE.
1000 GO TO 100.

1000 END.

1000 X SI EL CARACTER RECIBIDO NO FUE NAK O ACK, EXISTE
1000 UN ERROR EN LA TRANSMISION Y SE IRA A UN BASURERO
1000 PARA RECUPERARSE POR "TIMEOUT"

1000 FORMATERR = TRUE.
1000 GO TO 99.

1000 END.

1000 X EN CASO DE NO SER UN ACK LA PRIMERA TRANSMISION
1000 QUE INTENTOS, SE PREGUNTA POR UN NAK. DE SERLO, SE
1000 PASA A HACER UNA TERMINACION ("ENABLEINPUT") PARA
1000 ID AL REQUEST DE RECEPCION
1000 SI LA BANDERA "STATION (ENABLE)" ES FALSA, PASARA
1000 A LA SIGUIENTE INSTRUCCION. ESTA TERMINACION CAU-
1000 SARA UN SALTO DE CONTROL

1000 IF CHAR = NAK THEN
1000 BEGIN
1000 TERMINATE ENABLEINPUT.
1000 TERMINATE NOINPUT.

1000 END.
1000 FORMATERR = TRUE.

99:

1000 X BASURERO UTILIZADO PARA CUANDO HAY SOBREFLUJO EN
1000 EL BUFFER O CUANDO ESTA PRENDIDO EL BIT DE ERROR (STOPIIT)

1000 RECEIVE (50 HILLT) (ERROR(0)).
1000 GO TO 99.

110:

1000 X CUANDO HAY ERROR DE PARIDAD EL ADAPTED TRANSMITE
1000 "4*7F" PARA INDICARLO. ESTO ES UTILIZADO CON TER-
1000 MINALES SINCRONAS

1000 IF CHARACTER NE ALLOHES THEN
1000 GO TO 99.

100:

1000 X PREGUNTA SI LA VARIABLE DE RETRANSMISION NO SE HA

00486000
00487000
00488000
00489000
00490000
00491000
00492000
00493000
00494000
00495000
00496000
00497000
00498000
00499000
00500000
00501000
00502000
00503000
00504000
00505000
00506000
00507000
00508000
00509000
00510000
00511000
00512000
00513000
00514000
00515000
00516000
00517000
00518000
00519000
00520000
00521000
00522000
00523000
00524000
00525000
00526000
00527000
00528000
00529000
00530000
00531000
00532000
00533000
00534000
00535000
00536000
00537000
00538000
00539000
00540000
00541000
00542000
00543000
00544000
00545000
00546000





* EXCEDIDO DE LO INDICADO, DE SER ASI, HARA UNA TRANS-
* FERENCIA DE CONTROL, Y REINICIARA ESTE REQUEST

IF PETRY GT 0 THEN
BEGIN

120:

PETRY = PETRY - 1;
TERMINATE NOINPUT;

END.

* EN CASO DE QUE LA RETRANSMISION YA SE HAYA EXCEDIDO,
* HABRA UNA TERMINACION DE ERROR, ENVIANDOSE UN MEN-
* SAJE AL MCS Y AL DCC Y PONIENDOSE LA ESTACION CONC-
* NO-READY; EL MCS MAS TARDE PONDRÁ LA ESTACION READY

TERMINATE ERROR.

150:

IF PETRY GT 0 THEN
GO TO 120.

* INICIALIZACION DE LA VARIABLE DE RETRANSMISION
* EN 25 Y REINICIALIZACION DE ESTE REQUEST

INITIALIZE PETRY.
STATION(TALLY) = 25.
TERMINATE NOINPUT.

DISCIPLINA DE RECEPCION PARA DISPOSITIVO CON DIRECCION FISICA
(POLLING)

REQUEST POLLSPR:

* CONDICIONES DE ERROR ETIQUETADAS

ERROR(1) = TIMEOUT:110,
LUSOFCARRIER:110,
STOPBIT:99,
PARITY:110,
BUFOVEL:99.

* VARIABLE DE UN BIT QUE ES UTILIZADA PARA INDICAR CUANDO
* UNA ESTACION ESTA RECIBIENDO CARACTERES DE CONTROL

CONTROLFLAG = FALSE.

* SE PREGUNTA SI SE RECIBIO UNA INTERFERENCIA DE BRK. EN CASO DE SER CIERTO
* HABRA UNA TERMINACION QUE ENVIARA UN MENSAJE DE ERROR AL MCS Y AL
* DCC, FORZANDO A LA ESTACION SALIR DEL REQUEST Y DEJANDO LA ESTACION
* NO-READY; EL MCS MAS ADELANTE LA PONDRÁ READY

IF TOR(3) THEN
BEGIN

DEFK(TRANSMIT) = TRUE.
TOR(3) = FALSE.

00547000
00548000
00549000
00550000
00551000
00552000
00553000
00554000
00555000
00556000
00557000
00558000
00559000
00560000
00561000
00562000
00563000
00564000
00565000
00566000
00567000
00568000
00569000
00570000
00571000
00572000
00573000
00574000
00575000
00576000
00577000
00578000
00579000
00580000
00581000
00582000
00583000
00584000
00585000
00586000
00587000
00588000
00589000
00590000
00591000
00592000
00593000
00594000
00595000
00596000
00597000
00598000
00599000
00600000
00601000
00602000
00603000
00604000
00605000
00606000
00607000

TEPHILATE ERROR.

END.

AL MANDAR UN INICIO DE TRANSMISION CAUSA QUE EL ADAPTER SEA
PUESTO EN ESTADO DE TRANSMISION. DESPUES DE ESTO SE TRANSMITE
LA PRIMERA PARTE DE LA DISCIPLINA POLL

INITIATE TRANSMIT.
TRANSMIT FGT.
TRANSMIT ADDRESS (TRANSMIT).
TRANSMIT POL.
TRANSMIT END.
FINISH TRANSMIT.

10:

* AL MANDAR UN INICIO DE RECEPCION SE PREPARA AL ADAPTER A
RECIBIR DESPUES DE UN CIERTO RETARDO
EN CASO DE RECIBIR BLOQUE A CONDICION DE ERROR
QUE ANI APARECEN, IBA A LA ETIQUETA SERALADA.

INITIATE RECEIVE.
RECEIVE (1 SEC)
(TIMEOUT:150,
LUSSOFCAPIER:110,
STUFFIT:99,
PARITY:110,
BUFOVFL:99).

* SE PREPARA PARA RECIBIR UN SON COMO PRIMER CARACTER
IF CHARACTER = SON THEN

* SI FUE UN SON SE INICIALIZA EL CHECADOR DE BLOQUE DE CARAC-
* TRES

BEGIN
INITIALIZE HCC.

* SE RECIBE DIRECCION DE LA ESTACION
RECEIVE ADDRESS(TRANSMIT) (ERROR(0),ADDRESS:99).

* SE RECIBE EL INDICADOR DE DONDE SE INICIA EL TEXTO
* Y SE INICIALIZA EL TEXTO

RECEIVE STX (ERROR(0),PLIMATER:99).

INITIALIZE TEXT
* SE PREPARA ESPACIO PARA LA ENTRADA DEL MENSAJE

GETSPACE(99).
TGC(2) = FALSF.

* SE PREGUNTA SI SE PIDIO SECUENCIA

IF TGC(0) THEN
NEXT.

* SE INICIALIZA EL CONTADOR DE ESPACIOS

00600000
00609000
00610000
00611000
00612000
00613000
00614000
00615000
00616000
00617000
00618000
00619000
00620000
00621000
00622000
00623000
00624000
00625000
00626000
00627000
00628000
00629000
00630000
00631000
00632000
00633000
00634000
00635000
00636000
00637000
00638000
00639000
00640000
00641000
00642000
00643000
00644000
00645000
00646000
00647000
00648000
00649000
00650000
00651000
00652000
00653000
00654000
00655000
00656000
00657000
00658000
00659000
00660000
00661000
00662000
00663000
00664000
00665000
00666000
00667000
00668000
00669000



21: TALLY(0) = 0.

IF SE RECIBE UN CARACTER Y SI EXISTE UNA CONDICION
DE ERROR, PASARA A LA ETIQUETA QUE CORRESPONDA,
EN CASO DE FIN DE TEXTO IRA A LA ETIQUETA 30.

RECEIVE(1 SEC) (ERROR(0),
ETX(30)).

96:

IF PREGUNTAMOS SI ES UN CARACTER DE FIN DE CAMPO PROTEGIDO O
UN CARACTER DE LF PARA PODER IR A RECIBIR

IF CHAR="10" THEN GO TO 21.
IF CHAR="25" THEN GO TO 21.

IF CONTADOR PARA CARACTERES EN LA LINEA
TALLY(2) = TALLY(2) + 1.
IF TALLY(2) LE 72 THEN
BEGIN

IF PREGUNTAMOS SI ES UN CARACTER BLANCO

IF CHAR = SP THEN

BEGIN

TALLY(0) = TALLY(0) + 1.
GO TO 21.

END

ELSE

BEGIN

TOG(2) = TRUE.

IF ALMACENAMOS TEMPORALMENTE UN CARACTER

TALLY(1) = CHAR.

24:

IF SE PREGUNTA POR CARACTERES BLANCOS

IF TALLY(0) GT 0 THEN
BEGIN

IF SE HACE UNA PAUSA PARA QUE EL DCP PUEDA ATENDER OTRAS ES-
TACIONES

PAUSE.

IF SE ALMACENA UN BLANCO Y EN CASO DE SER UN FIN DE BUFFER
SE IRA A LA ETIQUETA 99

STORE SP(ENDOFBUFFER:99).
TALLY(0) = TALLY(0) - 1.
GO TO 24.

END.

00619000
00670000
00671000
00672000
00673000
00674000
00675000
00676000
00677000
00678000
00679000
00680000
00681000
00682000
00683000
00684000
00685000
00686000
00687000
00688000
00689000
00690000
00691000
00692000
00693000
00694000
00695000
00696000
00697000
00698000
00699000
00700000
00701000
00702000
00703000
00704000
00705000
00706000
00707000
00708000
00709000
00710000
00711000
00712000
00713000
00714000
00715000
00716000
00717000
00718000
00719000
00720000
00721000
00722000
00723000
00724000
00725000
00726000
00727000
00728000
00729000



1000
 1000
 2000
 3000
 4000
 5000
 6000
 7000
 8000
 9000
 10000
 11000
 12000
 13000
 14000
 15000
 16000
 17000
 18000
 19000
 20000
 21000
 22000
 23000
 24000
 25000
 26000
 27000
 28000
 29000
 30000
 31000
 32000
 33000
 34000
 35000
 36000
 37000
 38000
 39000
 40000
 41000
 42000
 43000
 44000
 45000
 46000
 47000
 48000
 49000
 50000
 51000
 52000
 53000
 54000
 55000
 56000
 57000
 58000
 59000
 60000
 61000
 62000
 63000
 64000
 65000
 66000
 67000
 68000
 69000
 70000
 71000
 72000
 73000
 74000
 75000
 76000
 77000
 78000
 79000
 80000
 81000
 82000
 83000
 84000
 85000
 86000
 87000
 88000
 89000
 90000
 91000
 92000
 93000
 94000
 95000
 96000
 97000
 98000
 99000
 100000

CHAR = TALLY(1)
 STORE CHAR (ENDOFBUFFER:99).
 GO TO 21.

END.
 ELSE

2 EN CASO DE QUE SEA MAYOR A 72 CARACTERES EN LA LINEA,
 2 CANDE PARA EL CR PARA PASAR A LA LINEA SIGUIENTE

BEGIN
 TALLY(1) = CHAR.
 IF NOT TCG(2) THEN
 STORE SP (ENDOFBUFFER:99).
 TCG(2) = FALSE.
 STORE CR (ENDOFBUFFER:99).
 TALLY(1) = 0.
 TALLY(2) = 0.
 CHAR = TALLY(1).
 GO TO 96.

END.
 ELSE

2 EN CASO DE NO ESTAR EN MODO SECUENCIA.

BEGIN

2 RECIBIMOS ALGUN COMANDO O INFORMACION

RECEIVE (FPROR(0),
 CONTROL:200,
 ETX:30).

STORE (ENDOFBUFFER:99).

RECEIVE TEXT (FPROR(0),
 ETX
 ENDOFBUFFER:99).

END.

29:

30:

IF NOT TCG(2) THEN
 IF TALLY(1) GT 1 THEN
 STORE SP (ENDOFBUFFER:99).

2 RECIBIMOS EL CARACTER PARA CHECAR EL BLOQUE DE CARAC
 2 TERES.SI HAY UN ERROR EN EL CHEQUEO , IRA A LA ETIQU
 2 TA 99 Y SI HAY UN ERROR DE LOS YA MENCIONADOS ,
 2 IRA A LA ETIQUETA CORRESPONDIENTE

RECEIVE BCC (FPROR(0),
 RECERR:99).

2 SI NO RECIBIMOS UN CARACTER DE BREAK
 2 HAREMOS UNA TEMPORALIZACION TEMPORAL , YA QUE ESTA

00730000
 00731000
 00732000
 00733000
 00734000
 00735000
 00736000
 00737000
 00738000
 00739000
 00740000
 00741000
 00742000
 00743000
 00744000
 00745000
 00746000
 00747000
 00748000
 00749000
 00750000
 00751000
 00752000
 00753000
 00754000
 00755000
 00756000
 00757000
 00758000
 00759000
 00760000
 00761000
 00762000
 00763000
 00764000
 00765000
 00766000
 00767000
 00768000
 00769000
 00770000
 00771000
 00772000
 00773000
 00774000
 00775000
 00776000
 00777000
 00778000
 00779000
 00780000
 00781000
 00782000
 00783000
 00784000
 00785000
 00786000
 00787000
 00788000
 00789000
 00790000
 00790000



217

06-18

TOG(3) = TRUE.
GO TO 29.

[illegible]

00913000
00914000
00915000
00916000

5. CONCLUSIONES

Al emprender este trabajo no se pretendió contribuir al acervo de conocimientos en la especialidad de redes de cómputo, pero sí dar una contribución efectiva a la experiencia en México, desde el punto de vista de redes de terminales, ya que en redes de computadoras no existe.

La contribución que se hace, es que en la universidad se han llevado a cabo conexiones de muy variados dispositivos por medio de las disciplinas de línea que son de utilización comercial. Un ejemplo de esto se encuentra en el capítulo 4, donde utilizamos las disciplinas poll y selection y las adecuamos conforme a las características del dispositivo y a las necesidades del usuario. Este request descriptivo

to en el capítulo anterior, está siendo utilizado actualmente en las dos redes de la Universidad con algunos cambios, para poder ser utilizado por otro dispositivo con características físicas y lógicas diferentes.

Por otro lado, se deja ver que hay una gran necesidad de crear redes de computadoras, para aprovechar todas las facilidades que ellas brindan, es decir, compartir Software, paquetes estadísticos o de utilería (que cuesta mucho su renta), bases de datos y sistemas de información desarrollados en otros centros de cómputo, además de poder utilizar la computadora idónea para nuestras necesidades.

VOCABULARIO

- BACKUP - Archivo temporal formado por el sistema para una salida a impresión.
- BUFFER - Dispositivo almacenador de información.
- CANAL VIRTUAL - Vía lógica de comunicación.
- CIRCUITO - Enlace de comunicación entre dos dispositivos de la red.
- CIRCUITO VIRTUAL - Enlace lógico de comunicación entre dos dispositivos.
- COLA DE REQUEST - Lista o arreglo donde llegan los mensajes del sistema para el DCP.
- COMPARTIENDO RECURSOS - El uso de un recurso por un número diverso de usuarios.
- COMPILADOR - Programa verificador de semántica y sintaxis y traductor de lenguaje fuente a lenguaje objeto.
- COMPUTADORA ANFITRIONA - Computadora de una red accionando servicio a un nodo o Front-End.
- COMPUTADORA DE COMUNICACION - Computadora que actúa como interface entre otra computadora o una terminal y un enlace de comunicación.

- COMPUTADORA DE SWITCHEO - Procesador que ejecuta funciones de almacenador y despachador.
- CONCENTRADOR - Procesador programable con capacidad de almacenamiento y funciones de multiplexor.
- CONTROL DE LINEA - Protocolo primario de la red que administra los recursos del medio transmisión y en algunos casos detecta y corrige errores automáticamente.
- DEBUGGING - Detección de errores.
- DISPOSITIVO ESCLAVO - Dispositivo gobernado por un dispositivo asignado como maestro.
- DISPOSITIVO MAESTRO - Dispositivo que gobierna en la comunicación.
- DUPLEX - Término utilizado para conexiones a 4 hilos telefónicos.
- ENLACE - Ruta de comunicación entre dos nodos.
- ENTRADA REMOTA DE TRABAJOS - Es el modo de operación que permite la entrada de un conjunto de trabajos, por una lectora de tarjetas en un sitio remoto y la salida del conjunto de trabajos a una impresora o perforadora de tarjetas en dicho sitio.
- FRONT-END - Procesador que esta orientado para servir como interface de comunicación entre computadoras o una(s) terminal(es).
- HALF-WORD - Media palabra del DCP que puede ser derecha o izquierda; consiste de 3 bytes.

HARDWARE - Componentes físicos de una computadora (por ejemplo el procesador central).

INTERFACE - Puente lógico o físico entre dos estados.

LOOP - Ciclo de acción que puede ser infinito.

MULTIPLEXOR - Computadora que controla la entrada y/o salida de una computadora. Dispositivo no programable.

MULTIPUNTO - Línea en donde hay más de una estación.

NODO - Punto de convergencia, de rutas de comunicación en una red.

NODO SERVIDOR - Nodo que provee recursos de la red.

OVERHEAD - Sobrecargar al sistema.

PARIDAD - Término usado para indicar cuando hay una mala transmisión de datos.

PROCESO - Una secuencia sistemática de operaciones para producir un resultado específico.

RECURSO - Cualquier capacidad o servicio permitido al usuario, tal como anfitrión(s), programas, archivos de datos, capacidad para almacenar o una combinación de estas.

RED CENTRALIZADA - Red en la que todos los nodos son conectados a un solo nodo.

RED DISTRIBUIDA - Red en la cual, todos los nodos tienen múltiples conexiones a otros nodos.

RED DE ESTRELLA - Red de tipo centralizada.

- REQUEST - Unión de control y disciplinas de línea para la administración de una línea y control lógico de la estación (Nota: no confundir con cola de request).
- SIMPLEX - Término usado para conexiones a dos hilos telefónicos.
- SISTEMA OPERATIVO - Programa controlador maestro del sistema.
- SOFTWARE - Componentes lógicos de una computadora (por ejemplo el sistema operativo).
- STORE-AND-FORWARD - Función de almacenar y despachar información de algunos concentradores y Front-End.
- SUB-NETWORK - Red que ella misma es componente de una red muy grande.
- SWITCHEO DE MENSAJES - Función de direccionar mensajes.
- TALLYS - Registro de 8 bits que sirve para almacenar caracteres o usado como contador.
- TERMINAL - Un dispositivo conectado en la red de comunicación, por el cual se puede meter o sacar información del sistema.
- TIMEOUT - Tiempos fuera o de retardo, para establecer una comunicación.
- TOGGLE - Bandera que debe asumir siempre, uno de dos estados estables.

I N D I C E D E F I G U R A S

FIG 1.1	VISTA DEL USUARIO, EN UNA RED DE COMUNICACION DE COMPUTADORAS	4
FIG 1.2	VISTA DEL USUARIO EN UNA RED DE COMPUTADORAS	6
FIG 1.3	SISTEMA DISTRIBUIDO DE COMPUTADORAS	10
FIG 1.4	PLAN FUTURO DEL SISTEMA DISTRIBUIDO DE COMPUTADORAS	14
FIG 1.5	ESQUEMA DE LA RED DE LA UNIVERSIDAD DE MEXICO	16
FIG 1.6	RED CENTRALIZADA	18
FIG 1.7	RED DISTRIBUIDA	19
FIG 1.8	RED DISTRIBUIDA TIPO ANILLO	19
FIG 2.1	CONFIGURACION DEL SISTEMA B-6700 INCLUYENDO COMUNICACION DE DATOS	51
FIG 2.2	DIAGRAMA DE BLOQUE DEL DCP	60
FIG 2.3	GENERACION DEL AMBIENTE DE OPERACION DEL DCP	67
FIG 2.4	REFERENCIAS DE INTERSECCIONES DEL PROGRAMA DE NDL	68
FIG 2.5	SUBSISTEMA DE COMUNICACION DE DATOS EN LA B-6700	71
FIG 2.6	SISTEMA DE DATACOM INICIALIZADO	74
FIG 2.7	ENTRADA DE UN MENSAJE DE LA ESTACION A	78
FIG 2.8	SALIDA DE UN MENSAJE PARA LA ESTACION A	80

FIG 3.1	ESTRUCTURA DE UN ESTRATO TRIPLE Y PROTOCOLO JERARQUICO DE DOS LADOS	85
FIG 3.2	NOTACION USADA PARA REFERENCIAR LOS COMPONENTES DE PROTOCOLOS JERARQUICOS	86
FIG 3.3	LA INTERFACE ENTRE SUCESIVOS PROTOCOLOS DE EXTREMO	89
FIG 3.4	LA ESTRUCTURA PROPUESTA PARA UN PROTOCOLO DE EXTREMO $E_{1,i}$ MODULAR	93
FIG 3.5	ESTRUCTURA PROPUESTA PARA UN PROTOCOLO DE EXTREMO $E_{r,i}$ MODULAR	94
FIG 3.6	LA ESTRUCTURA DE $E_{1,i}$ DESPUES DE LA MODIFICACION PARA ACEPTAR EL TIPO DE MENSAJE MC'	98
FIG 3.7	PROTOCOLO DE LA DISCIPLINA ENVIADORA	109
FIG 3.8	PROTOCOLO DE LA DISCIPLINA RECEPTORA	111

INDICE DE CUADROS

CUADRO 3.1	CODIGO ASCII	114
CUADRO 3.2	CODIGO EBCDIC	115

INDICE DE DIAGRAMAS

DIAGRAMA 3.1	DISCIPLINA POLL	129
DIAGRAMA 3.2	DISCIPLINA SELECTION	136
DIAGRAMA 3.3	DISCIPLINA FAST SELECT	141
DIAGRAMA 3.4	DISCIPLINA BROADCAST SELECT	146
DIAGRAMA 3.5	DISCIPLINA GROUP SELECTION	152
DIAGRAMA 3.6	DISCIPLINA GROUP POLL	160
DIAGRAMA 3.7	DISCIPLINA POINT-TO-POINT CONTENTION	166
DIAGRAMA 3.8	DISCIPLINA POINT-TO-POINT SWITCHED CONTENTION	174
DIAGRAMA 3.9	DISCIPLINA MODO BATCH	182
DIAGRAMA 3.10	DISCIPLINA SEQUENTIAL SELECT	190
DIAGRAMA 3.11	DISCIPLINA POINT-TO-POINT NO-CONTENTION (LINEA PRIVADA, SOLO ENVIA)	194
DIAGRAMA 3.12	DISCIPLINA POINT-TO-POINT NO-CONTENTION (LINEA PRIVADA, SOLO RECIBE)	199

REFERENCIAS

- { 1} *BDLC-a link-control method that can handle up to 127 unacknowledged messages*, Michael J. Bedford, Burroughs corp., Detroit, Mich.
- { 2} *Interprocess communication protocols for computer networks*, by Carl Allan Sunshine, Digital Systems Laboratory Dept. of electrical Engineering and Dept. of Computers, Stanford University.
- { 3} *Data link control*, J. Ray Kersey IBM-SCD, Research triangle Park, N.C.
- { 4} *Survey of Network Control Programs in the ARPA computer Network*, by Jonathan B. Postel Mitre Technical report MTR-6722 Mclean; Virginia 22104.
- { 5} *Multiple-Access communications for Computer Nets*, Jay W. Schwartz, Naval Research Laboratory, Michael Muntner Defense Communications Agency-system engineering facility.
- { 6} *Economic Considerations in Computer-Communication Systems* D. A. Dunn, Engineering-Economic System Dept., Stanford University, A. J. Lipinski, Institute for The Future, Menlo Park, California.

- { 7} *Toward Modular Hierarchical Structures for protocols in Computer Networks*, Mohamed G. Gouda and Eric G. Manning, Computer Communications Networks Group, University of Waterloo, Waterloo, Ontario, Canada.

- { 8} *Analytical Techniques for computers Networks Analysis and Design*, L. Fratta, Politécnico di Milano and Centro Telecomunicazioni, V. Montanari, Istituto per l'Elaborazione dell' Informazione IEI-CRN, PISA

- { 9} *Planning computer-Communication Networks*, H. Frank and I. T., Network analysis Corporation.

- {10} *System Design for Computer Networks*, Peter G. Neumann, Stanford Research Institute, Menlo Park, California.

- {11} *Survey of Analytic Methods for Computer Network Design* Leonard Kleinrock, University of California, Los Angeles.

- {12} *The terminal IMP for the ARPA Computer Network*, by S.M. Ornstein, F. E. Heart, W. R. Crowther, H. K. Rising, S. B. Russell and A. Michel Belt Beranek and Newman Inc., Cambridge, Massachusetts.

- {13} *Function-oriented protocols for the ARPA Computer Network*, S. D. Crocker, Advanced Research Projects Agency, J. F. Heafner, the Rand Corporation, R. M. Metcalfe, Massachusetts Institute of Technology, and Jonathan B. Postel, University of California.

- {14} *Flow Control in the ARPA Network*, Leff Herrman, Aiken Computation Laboratory, Harvard University, Cambridge, Mass., 02138, U.S.A.

- {15} *The stability problem of Broadcast Packet Switching Computer Networks*, G. Fayolle, E. Gelenbe, J. Labetoulle, D. Bastin, IRIA-Laboratoria, Donaine de volucean, France

- {16} *Flow Control In a Resource-Sharing Computer Network*, Robert E. Kahn, member, IEEE, and William R. Crowther.

- {17} *Computer Communication Network Design-Experience with Theory and Practice*, by Howard Frank, Network Analysis Corporation, Glen Cove, New York, Robert E. Kahn, Bolt Beranckand Newman Inc., Cambridge, Massachusetts, and Leonard Kleinrock, University of Massachusetts, Los Angeles, California.

- {18} *Topological Optimization of Computer Networks*, by Howard Frank and Wushow Chou.

- {19} *The Interface message processor for ARPA Computer Network*, by F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther and D. C. Waiden, Bolt Beranach and Newman Inc. Cambridge, Massachusetts.

- {20} *Terminal-Oriented Computer-Communication Networks*, by Mischa Schinartz, Feilow, IEEE, Robert R. Boorstyn, member, IEEE, and Raymond L. Pickholtz, member, IEEE.

- {21} *The Canadian Universities Computer Network Topological Considerations*, by John de Mercado, Rene Guindon; John Da Silva, and Michel Kadoch, Ministry of Communications, Ottawa, Canada.

- {22} *Resource-Sharing Computer Communications Networks*, by Robert E. Kahn, member, IEEE.

- {23} *Analytic and simulation methods in computer Network design*, by Leonard Kleinrock, University of California, Los Angeles, California.

- {24} *Computer Networks*, Text and references for a tutorial Dr. Marshall Abrams, Mr. Robert P. Blanc, Mr. Ira, W. Cotton, Sponsored by the IEEE Computer Society.

- {25} *The Principles of a data Communication Network for Computers and Remote Peripherals*, by D. W. Davies, National Physical Laboratory, Teddington, Middlesex, UK
- {26} *Computer Network Development to achieve resource sharing*, by Lawrence G. Roberts, and Barry D. Wessler, advanced research projects agency Washington, D. C.
- {27} *The Design of a switching system to allow remote access to computer, services by other computers and terminal devices*, by R. A. Scantlebury, and P. T. Wilkinson, National Physical Laboratory, Teddington, Middlesex, U. K.
- {28} *Algunas ventajas de una red de tele-información en México*, Enrique Grapa M., Héctor Pinzón y Raymundo Segovias, IIMAS, UNAM, México, D.F.
- {29} *Consideraciones Generales sobre el diseño de redes de tele-informática*, Renato Barrera R., IIMAS, UNAM, México, D. F.
- {30} *Redes de Computadoras- Una Introducción*, Gursharan S. Sidhn y Raymundo Segovia, IIMAS, UNAM, México, D.F.

- {31} *The datapac Network and the SL-10 Data Network Processor*,
by David A. Twyver, Bell-Northern Research, Canada.
- {32} *A Maintenance System Architecture for Packet Switching
Data Networks*, by B. A. Parsons, Transcanada telephone
Co., W. A. MCCRUM, Bell Northern Research, Canada.
- {33} *Some Criteria for standards in computer/Communications*,
by V. C. Macdonald, Gobierno de Canada, Ottawa, Canada.
- {34} *Protocol for a Computer Network*, by D. B. McKay and
D. P. Karp.
- {35} *Introduction to data Communication*, by Donald E. Murphy
with Stephen A. Kallis, Jr.
- {36} *Data Transmission*, by M. D. Bacon, and G. M. Bull,
Department of Computer Science the Hatfield Polytechnic
- {37} *Transport Protocol. Standard host-host Protocol for
heterogeneous computer networks*, by H. Zimmermann, M.
Elie, Reseau Cyclades, SCH 519.1
- {38} *Data-Entry Vertical Terminal Mapping Study for EURONET*
by N. Naffah, IRIA. Domaine de Voluveau-BP. 105 78150
Rocquencourt, France.

- {39} *Data Communications Processor Reference Manual (1054384)*
Burroughs Corporation.
- {40} *Data Communications Funtional Description (5000060)*,
Burroughs Corporation.
- {41} *Data Communications Processor Technical Manual (1041639)*
Burroughs Corporation.
- {42} *Data Communications Funtional Description, Information*
Manual. Burroughs Corporation.
- {43} *DC ALGOL, Reference Manual (5000052)*, Burroughs Corpo-
ration.
- {44} *Training course data communications, Student Material*
(EP 4048). Burroughs Corporation.
- {45} *NDL, Reference Manual (5001522)*, Burroughs Corporation.