

03063

3

24



INSTITUTO DE
INVESTIGACIONES
ELECTRONICAS

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

UNIDAD ACADEMICA DE LOS CICLOS PROFESIONAL
Y POSGRADO DEL COLEGIO DE CIENCIAS Y HUMANIDADES

INSTITUTO DE INVESTIGACIONES
EN MATEMATICAS APLICADAS Y EN SISTEMAS

RED LOCAL PARA LA COMUNICACION ENTRE PROCESOS EN TIEMPO REAL

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACION

P R E S E N T A

CESAR R. FREYRE MENDIBURU

MEXICO, D.F.

ABRIL 1987

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

I N D I C E

| | | |
|---------|---|----|
| | INDICE | 0 |
| 1 | INTRODUCCION | 1 |
| 2 | FUNDAMENTOS DE LAS REDES DE COMPUTADORAS | 5 |
| 2.1 | SISTEMAS DISTRIBUIDOS | 5 |
| 2.2 | REDES LOCALES | 9 |
| 2.3 | TOPOLOGIA | 9 |
| 2.4 | EL MODELO ISO | 13 |
| 3 | DISEÑO DE LA ARQUITECTURA DE LA RED CP/32 | 18 |
| 3.1 | TOPOLOGIA DE LA RED CP/32 | 18 |
| 3.2 | ARQUITECTURA DE LA RED CP/32 | 19 |
| 3.2.1 | LA CAPA FISICA | 21 |
| 3.2.1.1 | MAQUINA FISICA | 21 |
| 3.2.1.2 | MAQUINA VIRTUAL | 24 |
| 3.2.1.3 | PROTOCOLO DE ENLACE ENTRE CONTROLADORES | 31 |
| 3.2.1.4 | TERMINACION ANORMAL DE TRANSFERENCIA | 34 |
| 3.2.2 | LA CAPA DE COMUNICACION DE DATOS | 35 |
| 3.2.2.1 | ESTRUCTURAS DE COMUNICACION | 40 |
| 3.2.3 | LA CAPA DE TRANSPORTE | 43 |
| 3.2.4 | LA CAPA DE APLICACION | 51 |
| 3.2.5 | DETECCION DE FALLAS EN LA COMUNICACION | 58 |
| 4 | INTERFACES Y PROTOCOLOS | 60 |
| 4.1 | PROTOCOLOS | 61 |
| 4.2 | INTERFACES | 62 |
| 5 | CONCLUSIONES | 66 |
| | REFERENCIAS | 69 |
| | BIBLIOGRAFIA | 72 |

CAPITULO 1

INTRODUCCION

Es de actualidad el uso de sistemas de cómputo en muy diversas aplicaciones. Los requerimientos para cada aplicación son muy variados, y los sistemas de cómputo ofrecen diversas opciones para cumplir dichos requerimientos. Estos requerimientos funcionales son las restricciones que el sistema de cómputo debe satisfacer, es decir, es la definición del problema objeto de la aplicación [22].

En aplicaciones en tiempo real tales como el control de procesos (generación y distribución de energía eléctrica, plantas químicas, maquinaria para fabricación), procesamiento de señales en comunicaciones, transacciones comerciales en línea (autorizaciones de crédito), son restricciones el tiempo de respuesta, la disponibilidad y la confiabilidad del sistema [22].

Una solución factible para estas restricciones son los Sistemas Distribuidos basados en multiprocesadores. Son más complejos que los sistemas centralizados (mainframe), pero ofrecen mayor capacidad de procesamiento y de expansión a bajo

costo, además de ser más flexibles para satisfacer los requerimientos de las aplicaciones, que los grandes sistemas centralizados. Un factor de complejidad es la comunicación entre los procesadores y para el cual se han desarrollado diferentes tipos de tecnología [26].

En el Departamento de Simulación del Instituto de Investigaciones Eléctricas se ha desarrollado un Sistema de Adquisición de Datos (SAD) para centrales termoeléctricas, con base en el proyecto J-100 [7].

Entre los requerimientos de este sistema se especificó una disponibilidad de operación del 98.9 %. Para lograr este requerimiento, y después de un análisis del caso [16], se determinó que el SAD se diseñara con un sistema tolerante a fallas, implantado con una arquitectura reconfigurable. Un sistema reconfigurable implica el uso parcial o total de componentes duplicados en la arquitectura del sistema (redundancia en hardware y software) [26].

La arquitectura del sistema tolerante a fallas se definió con base en un sistema dual de minicomputadoras homogéneas (GOULD Modelo CONCEPT/32). Este tipo de minicomputadoras son aceptables para aplicaciones en tiempo real, por sus características de hardware y software - Sistema Operativo en tiempo real-, pero no ofrecen servicios para la comunicación entre ellas.

Por otro lado, los Sistemas Operativos (SO) para redes de computadoras pueden ser clasificados [20], primero como aquellos en los que cada nodo (computadora) tiene su propio SO distinto de los otros, y que no ofrece por sí mismo facilidades de comunicación entre nodos, sino que la red es implantada por programas independientes ejecutados en cada nodo (llamados network operating system), y segundo como aquellos en el que el SO está diseñado explícitamente con las facilidades de comunicación y que además forma un solo sistema en conjunto con todos los nodos que forman la red (llamados distributed operating system).

En este trabajo se presenta el diseño de una red local (Canal Paralelo de 32 bits -CP/32-) con las siguientes características:

- Es una extensión a un SO en tiempo real diseñado sin las facilidades de comunicación entre nodos.
- Implantado como un programa independiente que es ejecutado en cada nodo.
- Conformar un SO homogéneo para toda la red.

Con la característica de ser una extensión a un SO en tiempo real se quiere decir que la red está diseñada para extender la comunicación entre procesos, que son ejecutados en un ambiente de tiempo real, y cuando éstos se encuentran ubicados en distintas computadoras.

Aún cuando las especificaciones de la red no se ajustan con las normas establecidas por la Organización Internacional de Normalización (ISO), para su diseño seguimos la metodología para la definición de su Modelo de Referencia, el cual define la arquitectura de una red como una jerarquía de funciones de comunicación. Cada función esta ubicada en lo que llama capa o nivel de comunicación.

El diseño de la arquitectura de la red implica la definición y realización de:

- las capas de comunicación, y
- las relaciones entre las capas (interfaces y protocolos).

En el capítulo 2 se presentan los fundamentos de las redes de computadoras, así como el Modelo de Referencia de ISO. En el capítulo 3 se describe la arquitectura de la red CP/32 y en el capítulo 4 sus interfaces y protocolos. En el capítulo 5 se presentan las conclusiones sobre la implantación de la red.

CAPITULO 2

FUNDAMENTOS DE LAS REDES DE COMPUTADORAS

El objetivo de este capítulo es presentar de una manera simplificada un conjunto de conceptos relativos a las redes de computadoras, así como el Modelo de Referencia propuesto por la Organización Internacional de Normalización (ISO).

2.1 SISTEMAS DISTRIBUIDOS

Un Sistema Distribuido está constituido por un conjunto de elementos, como son tareas (programas) y recursos (periféricos, archivos) ubicados en computadoras distintas, entre los que se establecen determinadas relaciones con el fin de implantar conjuntamente una solución del problema objeto de la aplicación de dicho sistema. Las relaciones que se establecen entre los elementos del sistema pueden ser (Fig. 2.1, adaptada de la ref. 1) la comunicación entre tareas y la comunicación entre tareas y recursos [1].

ELEMENTOS:

Ⓣ TAREAS

Ⓜ RECURSOS

RELACIONES:

COMUNICACION ENTRE TAREAS



COMUNICACION ENTRE TAREAS Y RECURSOS



FIG. 2.1- COMPONENTES DE UN SISTEMA
CENTRALIZADO O DISTRIBUIDO (14)

En un sistema centralizado las relaciones son establecidas por el sistema operativo; en el caso de los sistemas distribuidos, para llevar a efecto dichas relaciones deberá existir un mecanismo que implante la comunicación entre los elementos que forman el sistema distribuido: una red de comunicación entre computadoras (Fig. 2.2, adaptada de la ref. 1).

Este mecanismo puede o no estar integrado en el SO; desde este punto de vista los SO son clasificados en sistemas operativos para redes (network operating system) y sistemas operativos distribuidos (distributed operating system) [6,20].

En los sistemas operativos para redes, cada nodo tiene su propio sistema operativo no orientado a redes, y las facilidades de comunicación entre nodos es controlada por programas independientes que son ejecutados en cada nodo.

En un sistema operativo distribuido, las facilidades de comunicación están integradas en él, formando un sistema operativo homogéneo para toda la red.

Por su aplicación, las redes de computadoras pueden ser de tres tipos: redes para compartir recursos, redes para procesamiento distribuido y redes para comunicación remota [25].

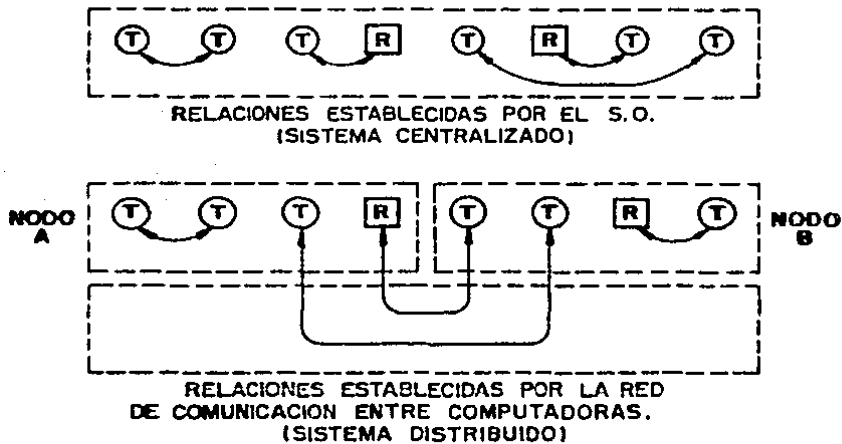


FIG. 2.2: RELACIONES ENTRE LOS COMPONENTES DE UN SISTEMA (14)

Las redes cuya aplicación es la de compartir recursos proveen la comunicación necesaria para que los recursos en una computadora puedan estar disponibles para otras computadoras, con la apariencia de estar localmente disponibles. Entre las actividades que requieren este tipo de redes se cuentan el acceso y transferencia de archivos remotos y el acceso a bases de datos distribuidas; la comunicación en este tipo de red se establece generalmente entre tareas en una computadora y recursos (de entrada/ salida por ejemplo) en otra computadora.

Las redes para procesamiento distribuido proveen la comunicación necesaria para el intercambio de información entre tareas o procesos ubicados en diferentes computadoras y que cooperan para la realización de una determinada función. Ejemplos de sistemas que utilizan este tipo de redes son los sistemas de control de procesos en tiempo real y sistemas de procesadores múltiples para bases de datos.

El objetivo de las redes para comunicación remota es realizar, a bajo costo, la comunicación entre un usuario y un sistema remoto a través de terminales remotas, con el propósito de tener acceso a los elementos del sistema remoto, por ejemplo, procesadores de palabras, compiladores, etc.

En cualquiera de los tres tipos de redes, la comunicación consiste algunas veces en el intercambio de mensajes cortos e independientes (transacciones ente tareas) y en otras en el flujo secuencial de datos (transferencia e impresión de

archivos).

2.2 REDES LOCALES

Stallings [17] define una red local como una red de comunicación que provee la interconexión de una variedad de dispositivos dentro de una área geográfica limitada.

Los dispositivos pueden ser computadoras, terminales, periféricos, sensores, etc; la distancia geográfica está en el rango de 0.1 a 25 Km.

Otras características de las redes locales son: velocidad de transmisión, en el rango de 0.1 a 100 Mbits/seg; baja relación de errores, de $1 \cdot 10^{-8}$ a $1 \cdot 10^{-11}$ y suele ser propiedad de una sola organización. Estas características distinguen las redes locales de los sistemas multiprocesadores y de las redes de largo alcance.

2.3 TOPOLOGIA

La naturaleza de una red de computadoras está caracterizada en términos de su topología, el medio de transmisión utilizado y el método de acceso al medio, que determinarán el tipo de información, la velocidad y eficiencia de la comunicación, y más aún la clase de aplicaciones que una

red puede soportar [18].

Los medios de transmisión más utilizados actualmente en las redes locales de computadoras son el conductor doble, el cable coaxial y la fibra óptica, que definen los tres tipos de topologías más comunes: bus/árbol y estrella con el método de acceso llamado CSMA/CD, y anillo, con el método llamado "token ring" [5,20].

TOPOLOGIA BUS/ARBOL. Esta topología está caracterizada por su configuración multipunto, es decir, que permite conectar más de dos dispositivos capaces de transmitir, uno a la vez, a través del medio físico (Fig. 2.3).

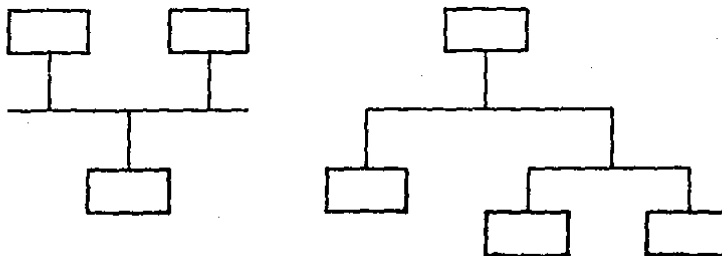


FIG.2.3:- TOPOLOGIAS TIPO BUS/ARBOL

El tipo de conexión es por paquetes de información, cada uno con tres tipos de campos: dirección fuente (transmisor), dirección destino (receptor) y el campo que contiene la información.

La transmisión de información empleando esta topología se realiza con dos técnicas: banda base y banda ancha.

La banda base es una técnica digital aplicada al cable coaxial y conductor doble; mientras que la banda ancha es una técnica analógica en el rango de la frecuencia de radio y es aplicada únicamente en cable coaxial.

TOPOLOGIA ESTRELLA. En la topología estrella se utiliza un elemento central de conmutación (en algunos casos una computadora) al cual está conectado punto a punto cada dispositivo de la red (Fig. 2.4).

El tipo de conexión utilizado en esta topología es el conocido como conmutación de circuitos y la técnica de transmisión puede ser en banda base o en banda ancha, a través de cable coaxial y conductor doble.

La topología estrella presenta el inconveniente de que, si el elemento central de la red falla, implica que la red completa fallará.

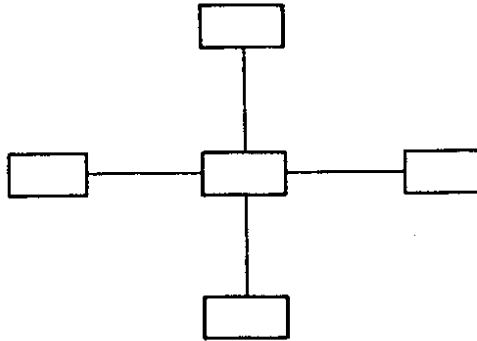


FIG. 2.4: TOPOLOGIA TIPO ESTRELLA

TOPOLOGIA ANILLO. La topología anillo consiste en un lazo cerrado en el que cada nodo está conectado a un elemento repetidor, formando una conexión punto a punto (Fig. 2.5).

La información es transmitida por paquetes, cada uno conteniendo un campo con la dirección destino. Conforme el paquete circula por los repetidores, el campo de la dirección es copiado por cada uno de ellos, y si un repetidor reconoce la dirección entonces copia el resto del paquete.

El enlace de un repetidor con otro es generalmente establecido con conductor doble o fibra óptica, y algunas veces con cable coaxial.

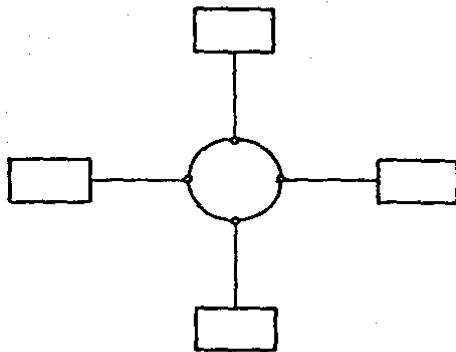


FIG.2.5.- TOPOLOGIA TIPO ANILLO

En la literatura especializada se ha difundido los trabajos relacionados con la clasificación de los diferentes tipos de topologías más comunes; entre ellos se pueden consultar las referencias [3],[17],[18],[23] y [26].

2.4 EL MODELO ISO

La Organización Internacional de Normalización (ISO) es una federación de organismos nacionales de normalización que se ocupa de la elaboración de recomendaciones internacionales para el estudio de temas específicos relacionados con las comunicaciones y la transmisión de información.

Para estudiar el tema de los sistemas distribuidos, se creó en 1977 el subcomité denominado Interconexión de Sistemas Abiertos (OSI). La consecuencia del estudio realizado por OSI fué la definición de un Modelo de Referencia para la comunicación de sistemas abiertos, es decir, su objetivo fué definir un conjunto de normas para que un sistema sea capaz de comunicarse con otros sistemas [1,27].

Con este objetivo, OSI presenta de una manera concreta lo que define como la arquitectura de la comunicación entre sistemas abiertos [1]. La metodología utilizada para el análisis de la arquitectura del Modelo de Referencia consiste en una estructuración según una jerarquía de niveles o capas; cada capa realiza determinadas funciones de comunicación, utilizando los servicios definidos por la capa inferior y proporsionando a su vez servicios a la capa inmediata superior. El resultado de lo anterior fue la elaboración de 7 capas que definen la estructura general del Modelo de Referencia (Fig. 2.6, adaptada de la ref. 14).

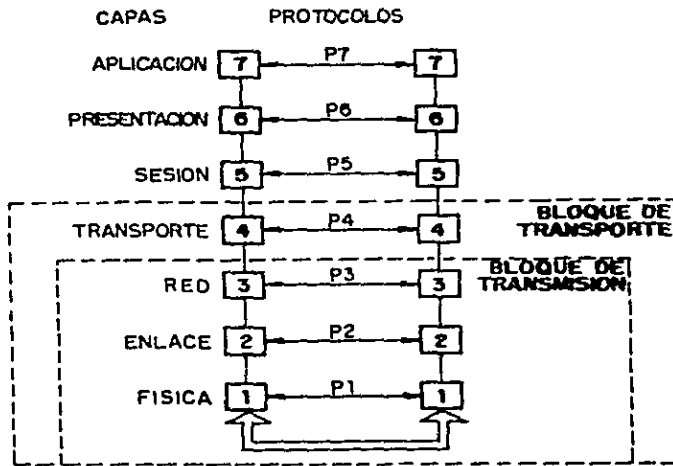


FIG. 2. 6.- ESTRUCTURA DEL MODELO ISO (14)

A continuación se describen brevemente las 7 capas del Modelo de Referencia.

La primera capa del modelo es llamada la Capa Física. Esta capa se encarga de activar, mantener y desactivar la conexión física entre los dispositivos que forman la red; define además los procedimientos y características funcionales de la interfaz al medio físico, considerando que las especificaciones mecánicas y eléctricas son parte del medio mismo.

La capa 2, la de enlace, implanta las funciones de sincronización y el control de errores de la información transmitida por el canal físico. En la capa 3, la Capa de Red, se establece la trayectoria de comunicación entre los dispositivos y recursos de la red; también implanta los algoritmos de corrección de errores.

El objetivo de la capa 4, la Capa de Transporte consiste en proporcionar un servicio de transporte de la información a través de los distintos sistemas, optimizando los recursos de comunicaciones para minimizar los costos en los intercambios de información (especialmente en los medios de transmisión públicos). Cabe hacer notar [1] que para el caso de la comunicación entre computadoras, el nivel de transporte se encuentra implantado en cada máquina, es decir, el objetivo de este nivel es realizar el transporte de la información entre las computadoras que forman la red, sin preocuparse de la forma en que la información es transportada y los medios de transmisión utilizados.

En la capa 5, la Capa de Sesión, se lleva a cabo el manejo y la sincronización de conversaciones entre dos procesos distintos; para cada conversación se establecerá una sesión entre los procesos correspondientes, y dejará de existir una vez que la conversación finalice. La capa 6, la Capa de Presentación, tiene como fin asegurar la interpretación del formato de la información que intercambian

los procesos de aplicación que dialogan; esta capa permite la heterogeneidad de la codificación de la información, por ejemplo, un sistema utiliza el código ASCII y el otro el EBCDIC, entonces uno de ellos deberá ser transformado al otro formato, asegurando el entendimiento entre los procesos.

Y por último, la capa 7, la de Aplicación, proporciona los servicios directamente al proceso de aplicación. Algunos de los servicios para las aplicaciones son: transferencia de archivos (acceso y manejo), transferencia de "trabajos", transmisión de mensajes, servicios de directorio, acceso a los sistemas que integran la red a través de terminales virtuales, etc.

Ahora bien, las normas que ISO especifica son los protocolos y servicios de cada capa del modelo; éstos se encuentran en varios estados de desarrollo: las normas de las capas 2,3,4 y 5 han sido completadas y son estándares internacionales, mientras que las normas de las capas 6 y 7 están bajo normalización [24].

Sobre este tema se pueden consultar las referencias [1],[17], [20],[24] y [27].

CAPITULO 3

DISEÑO DE LA ARQUITECTURA DE LA RED CP/32

3.1 TOPOLOGIA DE LA RED CP/32

La topología de la red CP/32 la definimos con base en el medio físico de transmisión que debía ser utilizado.

Como medio físico se disponía de un canal paralelo de 32 bits con conductor doble y de acceso compartido por cada dos nodos; el medio físico así especificado permite un enlace punto a punto con cada uno de los nodos que forman la red, lo cual nos llevó a definir una topología tipo anillo para la red CP/32. En la figura 3.1 se muestra esta topología en una red con cuatro nodos.

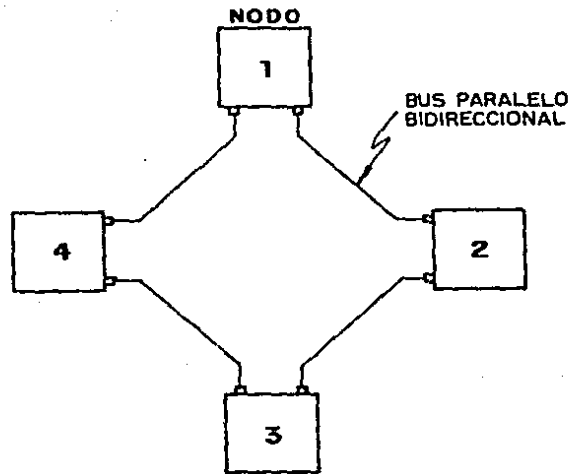


FIG. 3.1- TOPOLOGIA DE LA RED CP/32

3.2 ARQUITECTURA DE LA RED CP/32

El modelo conceptual de la red CP/32 lo diseñamos con base en la metodología empleada en la definición del Modelo de Referencia de la ISO. Esta metodología, la más aceptada en el diseño de sistemas de comunicación, consiste en conceptualizar la arquitectura de una red de comunicación a través de capas o niveles de abstracción que definen una jerarquía de funciones de comunicación [27].

La arquitectura de la red CP/32 la diseñamos con cuatro capas (Fig. 3.2), de las cuales tres definen las funciones de comunicación en la red - bloque de comunicación- y la cuarta está formada por los procesos de aplicación o usuarios de la red.

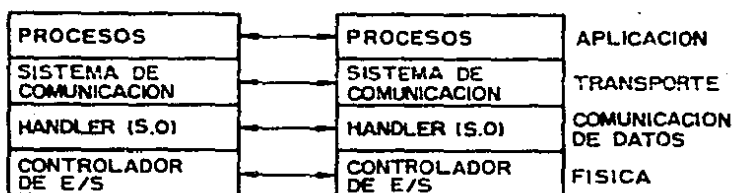


FIG. 3.2: MODELO DE LA RED CP/32

Las cuatro capas son:

- La capa física (máquina física y virtual),
- La capa de comunicación de datos, (manejador o "handler"),
- La capa de transporte (sistema de comunicación), y
- La capa de aplicación (procesos de aplicación).

Tomando como base la capa física (canal de comunicación ofrecido por el fabricante de las computadoras utilizadas), la capa de comunicación de datos implicó la adaptación de un manejador o "handler" (el cual forma parte del S0) a los

requerimientos de la red CP/32, y la realización de primitivas que llevan a cabo la función de esta capa de comunicación. En este trabajo se presentarán las modificaciones que se realizaron con respecto al handler.

La capa de transporte, con base en las capas física y de comunicación de datos la diseñamos como un proceso en tiempo real que realiza las funciones de comunicación entre procesos ubicados en distintas computadoras; además implicó la definición de la interfaz correspondiente a este nivel y que provee los servicios de comunicación a la capa de aplicación; en esta última capa ubicamos a los procesos que forman la aplicación y que son los usuarios de la red. En este capítulo se describe en forma independiente cada una de estas capas, es decir, cada capa se presenta como un nivel de abstracción con sus interrelaciones y posteriormente se hace la relación con sus capas adyacentes.

3.2.1 LA CAPA FISICA -

En la red CP/32, la capa física consiste en:

- máquina física (controlador de entrada/salida), y
- máquina virtual asociada.

3.2.1.1 MAQUINA FISICA -

La máquina física está formada por el controlador de entrada/salida. Las principales características de este dispositivo son [11]:

- Capacidad de direccionamiento de 16 Mbytes.
- Velocidad de transmisión de 3.2 Mbytes/seg. a una distancia máxima de 50 pies.
- Capacidad de detectar errores de transmisión/recepción.
- Capacidad de acceso directo a memoria (DMA).

La arquitectura del controlador es dependiente de la arquitectura de la computadora; físicamente está conectado por un lado al bus central de la computadora, y por el otro al controlador de la computadora adyacente a través del conductor doble. Un ejemplo del enlace físico de una red con cuatro nodos se muestra en la Fig. 3.3.

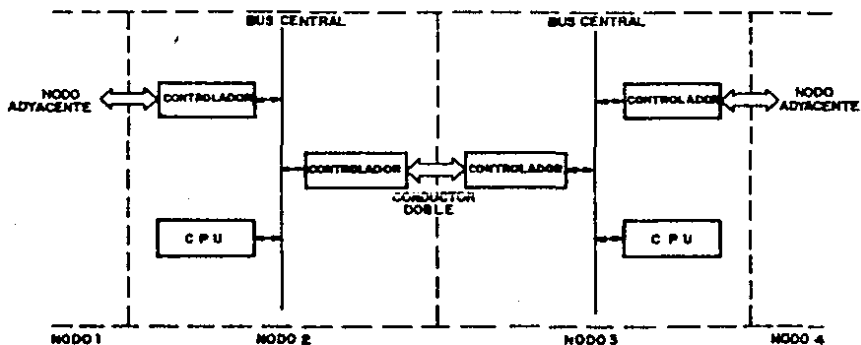


FIG. 3.3.- ENLACE FISICO DE LA RED CP/32

De acuerdo con la topología de la red CP/32 que se definió (Fig. 3.1), cada computadora está ligada con las adyacentes por medio de dos controladores; cada controlador es identificado por su dirección física, la cual es utilizada para definir la ubicación de cada computadora con respecto a las otras, como veremos más adelante.

Funcionalmente, el controlador esta dividido en 3 partes (Fig. 3.4 adaptada de la ref. 11):

- Interfaz al bus central de la computadora.
- Dispositivos de la lógica de control.
- Interfaz de 32 bits y 14 líneas de control al bus de datos (canal físico).

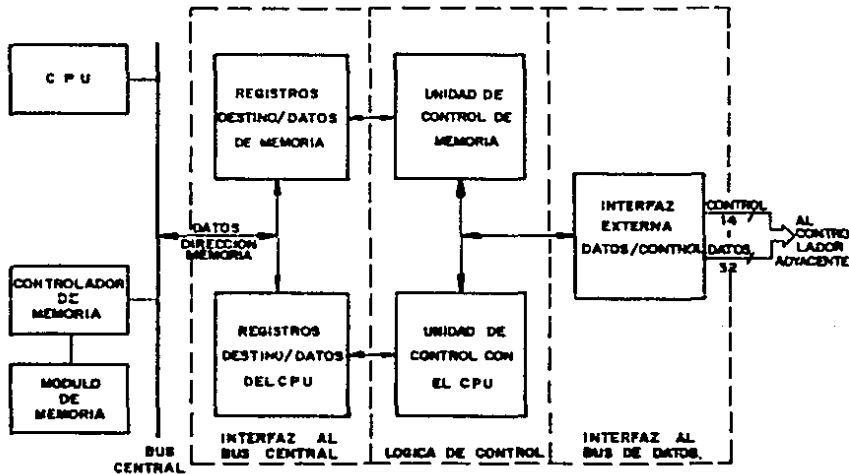


FIG. 3.4. DIAGRAMA FUNCIONAL DEL CONTROLADOR

La interfaz al bus central consiste de registros de destino/datos independientes para el CPU y la memoria. Estos registros son controlados por dos unidades lógicas que funcionan de modo simultáneo e independiente. Esta característica permite el acceso directo a memoria (DMA).

3.2.1.2 MAQUINA VIRTUAL -

La máquina virtual de la capa física consiste en la lógica de control (firmware) que implanta las funciones internas del controlador. En la Fig. 3.5 se muestra su interacción con el medio externo.

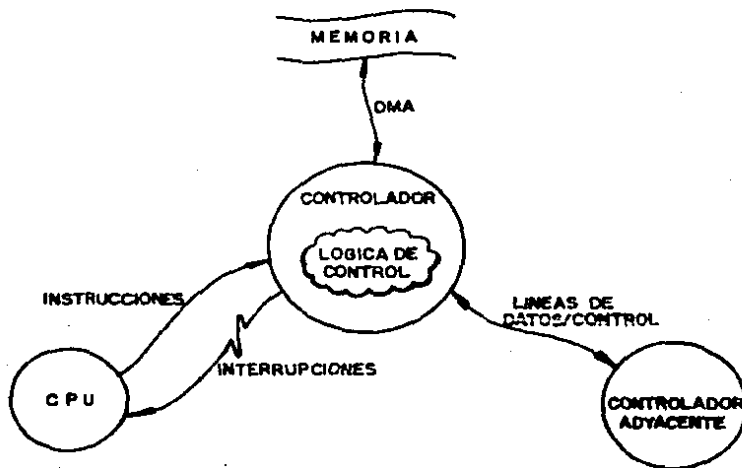


FIG. 3.5: INTERFAZ DE LA LOGICA DE CONTROL

La interacción de la lógica de control esta definida por:

1. Comunicación del CPU con el controlador, a través de instrucciones de E/S iniciadas por el CPU:
 - Arranque (START I/O)
 - Paro (HALT I/O)
 - Prueba (TEST DEVICE)
2. Comunicación del controlador con el CPU, a través del canal de acceso directo a memoria (DMA) y de interrupciones generadas por el controlador. Las causas principales de una interrupción son:
 - fin de operación (transmisión, recepción, paro).
 - petición de enlace.
3. Comunicación entre controladores, definida por los protocolos de enlace ("handshake") y de transmisión/recepción.

La lógica de control considera cuatro estados de operación [11]:

- No ocupado.
- Paro.
- Petición de enlace.
- Inicio de transferencia.

ESTADO NO OCUPADO. Este estado se puede definir como un estado inactivo en el que la lógica de control está en espera por una condición que lo llevará a un nuevo estado de operación. El diagrama de la Fig. 3.6 muestra los estados básicos de operación. En esta figura se puede observar un estado más, que es transitorio al salir del estado no ocupado;

en ese estado se establecen ciertas condiciones iniciales, como por ejemplo activar señales (banderas que indican el nuevo estado) y discriminar la condición por la que la lógica de control salió del estado no ocupado.

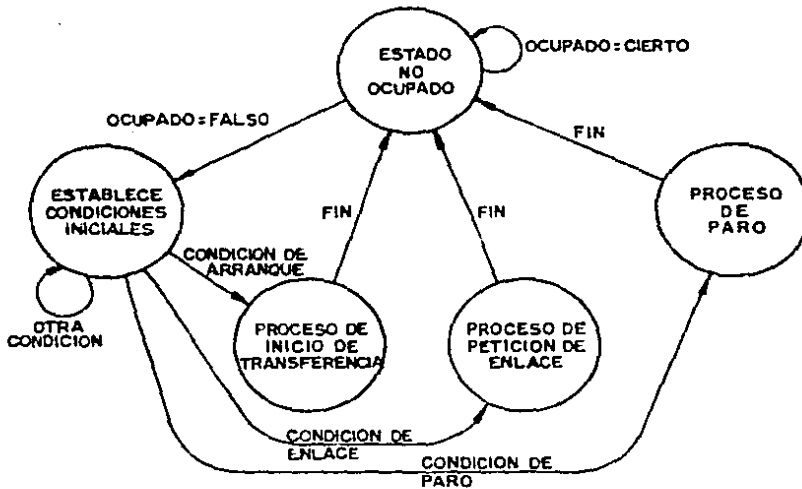


FIG. 3.6.- ESTADOS DE OPERACION DE LA LOGICA DE CONTROL

PROCESO DE PETICION DE ENLACE. A este estado se llega como consecuencia de una petición de transmisión de un controlador adyacente, a través de las señales de control. El objetivo de este proceso es generar una interrupción al CPU cuando un controlador adyacente desea transmitir a éste controlador, indicando que se ha establecido el enlace entre los controladores.

El siguiente es el proceso de petición de enlace [11]:

```
Procedimiento petición-enlace.  
Realiza protocolo de enlace con el controlador adyacente  
Prende bandera de enlace establecido  
SI está pendiente una interrupción de mayor prioridad  
  al CPU ENTONCES  
  espera (para enviar una interrupción al CPU)  
FIN SI  
envía interrupción al CPU  
Activa bandera de estado operativo a no ocupado  
Fin procedimiento petición de enlace.
```

La rutina "espera", que utilizaré por conveniencia en algunos procedimientos, es la que realiza la espera por el evento indicado como argumento, y su terminación indica la ocurrencia de dicho evento.

Al terminar el proceso, la lógica de control regresará al estado no ocupado, quedando nuevamente en espera por una condición.

PROCESO DE INICIO DE TRANSFERENCIA. El proceso de inicio de transferencia es iniciado por el CPU al ejecutar una instrucción de arranque (START I/O). Con este tipo de instrucción la lógica de control inicia una "operación de transferencia", ya sea de transmisión o de recepción.

Puede suceder, durante el inicio de la transferencia, que los dos controladores que forman el mismo canal inicien una operación de transmisión al mismo tiempo, ocasionando lo que llamamos una colisión en transmisión; esta colisión queda resuelta al definir una prioridad (por hardware) a cada uno de

los controladores. En el proceso de inicio de transferencia se detectará si el controlador tiene la mayor prioridad; si es así, entonces éste controlador tendrá la preferencia de realizar su transmisión.

El proceso de inicio de transferencia en el controlador con menor prioridad renunciará a su petición de transmisión, pero le indicará a su CPU (a través de una interrupción), que ha establecido el enlace para una operación de recepción.

El siguiente es el proceso de inicio de transferencia realizado por la lógica de control [11]:

```

Procedimiento inicia-transferencia.
Consigue dirección del bloque de comandos a ejecutar
SI bandera de enlace establecido ENTONCES
bandera de enlace establecido = FALSO
ejecuta comandos para operación de recepción
SINO
activa señal de petición de enlace
espera (por reconocimiento o petición de enlace de
controlador adyacente)
SI petición de enlace del controlador
adyacente ENTONCES /* colisión */
SI soy controlador con mayor prioridad ENTONCES
espera (por reconocimiento)
SINO
desactiva señal de petición de enlace
activa señal de reconocimiento, aceptando enlace
espera (por desactivación señal petición
enlace del controlador adyacente )
bandera de enlace establecido = CIERTO
SI esta pendiente una interrupción de mayor
prioridad al CPU ENTONCES
espera (por envío de interrupción al CPU)
FIN SI
envia interrupción al CPU
FIN SI
FIN SI
SI reconocimiento de enlace ENTONCES
ejecuta comandos para operación de transmisión
FIN SI
FIN SI
Bandera de estado operativo = NO OCUPADO
Fin procedimiento inicia-transferencia.

```

Una operación está definida por una lista de comandos llamada programa de canal, que a su vez consta de uno o más bloques de comandos, que son ejecutados por el controlador [11].

En la figura 3.7 (adaptada de la ref. 11) se muestra el formato de los bloques de comandos; estos bloques se contruyen en general para cualquier tipo de controlador utilizado en los sistemas GOULD. El código de operación indica la instrucción

que será ejecutada por el controlador, por ejemplo lectura o escritura del canal o su estado operativo.

La dirección del buffer de memoria especifica la localidad de memoria inicial de la información referenciada por el código de operación.

El contador de transferencia indica la longitud en palabras (32 bits) de la información referenciada por el código de operación. El estado de transferencia indica el estado de la operación que se realizó, es decir, la lógica de control indica en esta posición, al terminar el programa de canal, si ocurrió algún error y de que tipo.

En caso de error, el contador residual de transferencia indicará el número de palabras restante del buffer en el momento de ocurrir el error.

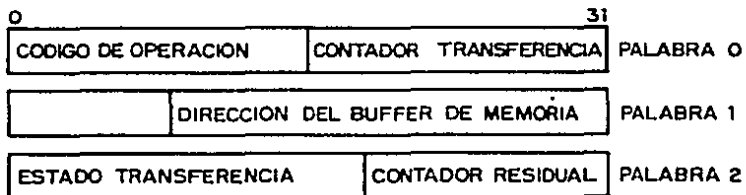


FIG.3.7: BLOQUE DE COMANDOS

PROCESO DE PARO. Estado en que el CPU ejecuta una instrucción de paro (HALT I/O). En este momento la lógica de control termina la ejecución del programa de canal, y coloca el estado de la transferencia en el último bloque de comandos en ejecución. En cualquier otro caso este proceso implica que la lógica de control regresa simplemente al estado no ocupado.

3.2.1.3 PROTOCOLO DE ENLACE ENTRE CONTROLADORES -

Las principales señales de control y la interconexión entre dos controladores se muestra en la Fig. 3.8 (adaptada de la ref. 11).

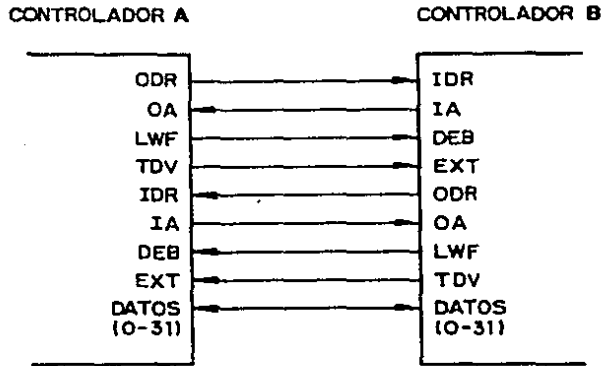


FIG. 3.8: DIAGRAMA DE SEÑALES

De acuerdo con la Fig. 3.8, el protocolo de enlace para la transferencia de una palabra se realiza del siguiente modo (Fig. 3.9):

Suponiendo que A transmite y B recibe, se tiene

1. El controlador A coloca una palabra en el bus de datos y notifica que desea enviar (ODR(A) activa).
2. El controlador B detecta la notificación de envío del controlador A (IDR(B) activa), entonces toma la palabra del bus y notifica el reconocimiento de recepción (IA(B) activa).
3. Listo para la transferencia de la siguiente palabra.

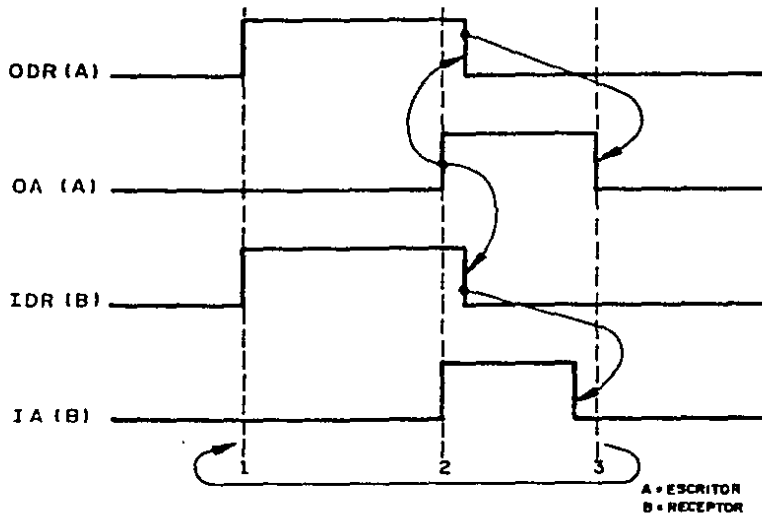


FIG. 3.9.- HANDSHAKE DE TRANSFERENCIA.

Para el caso de la terminación de transferencia o fin de programa de canal, el protocolo de enlace es como sigue (Fig. 3.10):

1. El controlador A notifica el envío de la última palabra de transferencia (ODR(A) y LWF(A) activas). El controlador A no transmitirá mas información hasta un nuevo enlace.
2. El controlador B detecta la notificación de envío de la última palabra (DEB(B) y IDR(B) activas), entonces toma la palabra del bus y notifica el reconocimiento de recepción (IA(B) activa).
3. El controlador B no esperará recibir mas información hasta un nuevo enlace.

Al terminar el protocolo de enlace ambos controladores enviarán una interrupción al CPU indicando el fin de la operación (A de transmisión y B de recepción).

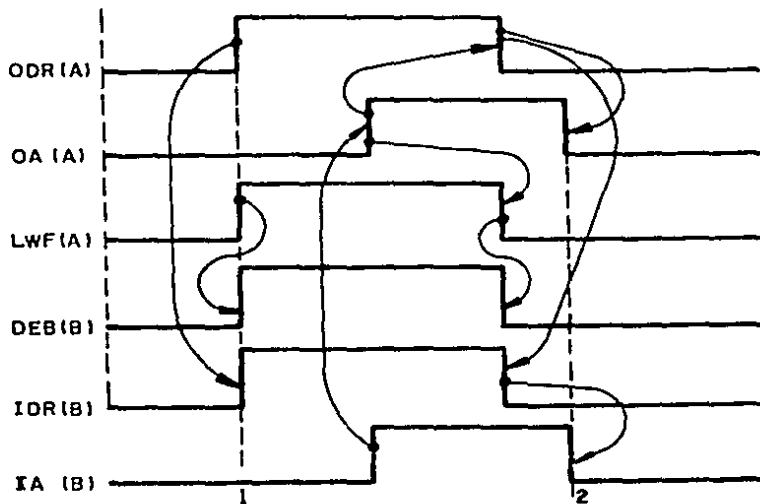


FIG. 3.10.- HANDSHAKE DE FIN DE TRANSFERENCIA.

3.2.1.4 TERMINACION ANORMAL DE TRANSFERENCIA -

Al terminar ambos controladores de ejecutar el programa de canal, la lógica de control pasa al estado no ocupado.

Si durante la ejecución de una transferencia el controlador transmisor o el receptor detectan una falla, o reciben una instrucción de paro de transferencia, esto es indicado al controlador opuesto activando la señal TDV (Fig. 3.8). La lógica de control de ambos controladores termina la transferencia, indica el estado de la operación en la palabra correspondiente del último bloque de comandos y envía una interrupción a sus respectivos CPU.

Las fallas que se pueden detectar son:

- Error de paridad en una operación de transferencia de información.
- Localidad de memoria no accesible durante una operación de transferencia de información.
- Detección de código de operación no válido para el controlador.
- Error de paridad o localidad de memoria no accesible durante la lectura de la dirección del programa de canal.
- Indicación de error al CPU cuando éste intenta ejecutar una instrucción de transferencia mientras el controlador esta procesando una instrucción anterior.
- Controlador opuesto fuera de línea o inactivo.

En este momento ya es necesario definir quien es el que "atiende" las interrupciones generadas por el controlador y el que realiza el inicio de las instrucciones de E/S.

3.2.2 LA CAPA DE COMUNICACION DE DATOS -

Las operaciones de transferencia a través del canal de comunicación son ejecutadas por el controlador, pero la función de "controlar" esas operaciones está asignada a la capa de comunicación de datos, la cual se puede ver como una interfaz entre las capas de transporte y física.

La capa de comunicación de datos de la red CP/32 la definimos con un manejador o "handler" especial para el manejo del canal, que debía cumplir con las siguientes funciones:

- Procesar las interrupciones generadas por el controlador.
- Ejecutar las instrucciones dedicadas de E/S.
- Contruir el programa de canal para las operaciones de transferencia.

- Notificar la recepción de información de un nodo adyacente.
- Definir (inicializar) el buzón de recepción.

En la Fig. 3.11 se muestra su interacción como un nivel de abstracción; por un lado, existe la interacción con el "usuario del canal", la cual se define a través de dos áreas de memoria: el buzón de recepción y el bloque de definición de operación (BDO), y de llamadas al supervisor (SVC) del SO. Por otro lado, la interacción con el controlador se establece a través de instrucciones de E/S (Arranque, Paro, Prueba) y de interrupciones generadas por el controlador.

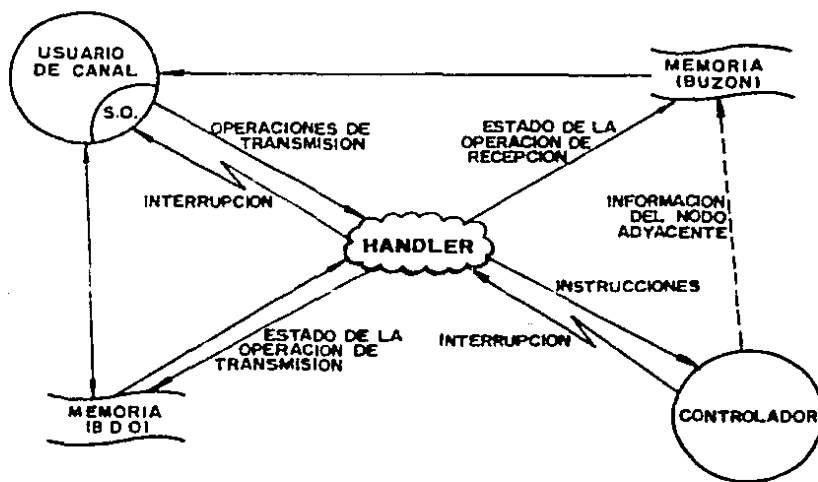


FIG. 3.11: INTERACCIONES DEL HANDLER

La modificación realizada al handler para el manejo del canal sigue la estructura definida para todos los handlers integrados al SO que manejan los dispositivos de E/S; por la forma en que están estructurados, permite que cada una de sus partes (punto de activación o "entry point") sea ejecutada independientemente de las otras; en el caso del handler para el manejo del canal, los puntos de activación que se modificaron para realizar las funciones principales son tres; siguiendo la Fig. 3.12, a continuación se presentan estos puntos de activación esquematizando una operación de transferencia.

1. Procesamiento de las interrupciones generadas por el controlador y de las operaciones de transmisión. Este es el procesador de las interrupciones generadas por el controlador, cuyos tipos habíamos definido como:
 - fin de operación (transmisión, recepción, paro), y
 - petición de enlace.

El funcionamiento de este punto de activación se presenta con el siguiente esquema. Las operaciones de transmisión son iniciadas por el usuario del canal en modo espera y entonces el proceso usuario del canal es "suspendido" por el sistema de control de E/S (SCES) del SO y éste a su vez activa al procesador de interrupciones; el tiempo suspendido termina hasta que su operación de transmisión haya sido completada por el controlador. Cuando el controlador termina la operación genera una

interrupción (por fin de transmisión) la cual es atendida por el procesador de interrupciones colocando el estado de la operación en el BDO y se indica el evento al SCES para que reactive nuevamente al proceso usuario suspendido. Cuando el proceso es reactivado puede entonces analizar el estado de su operación de transmisión.

En las operaciones de recepción, y una vez realizado el protocolo de enlace, el controlador que será receptor genera una interrupción por petición de enlace al handler. Este procesa la interrupción construyendo un programa de canal (con el código de lectura) e iniciando la operación con una instrucción de arranque (SIO).

Quando el controlador termina la recepción de información, genera una interrupción por fin de recepción. El procesador de interrupciones la procesa colocando el estado de la operación en el buzón de recepción, y a su vez genera una interrupción ("break") al proceso usuario para indicarle que tiene información en el buzón. En las interrupciones por fin de operación de paro, el handler coloca el estado de la operación, ya sea en el BDO ó en el buzón, dependiendo de la operación, con información relativa al controlador.

2. Control de las operaciones de transferencia. Este punto de activación es ejecutado por el SCES cuando detecta la pérdida de una operación ("time-out"). En este punto se

inicia una operación de paro (HIO) al controlador, y éste genera una interrupción por fin de operación de paro, que será procesada por el procesador de interrupciones.

3. Procesamiento del código de operación. En este punto de activación se realiza la decodificación de las operaciones que realizará el handler. Las operaciones que definimos para el manejo del canal son las de transmisión y la de inicialización del buzón de recepción. En esta última, el usuario del canal especifica el tamaño y el identificador del buzón (uno por cada controlador), así como el número máximo de bytes de la información que se recibirá; además en este punto se inicializarán los apuntadores de control del buzón. En la siguiente sección se detallará cada una de las estructuras de comunicación del handler, las cuales son el buzón de recepción y el BDO. Cuando se identifica una operación como de transmisión, entonces a través de este punto de activación se indica al SCES que ejecute la operación y a su vez ejecutará el punto de activación 1.

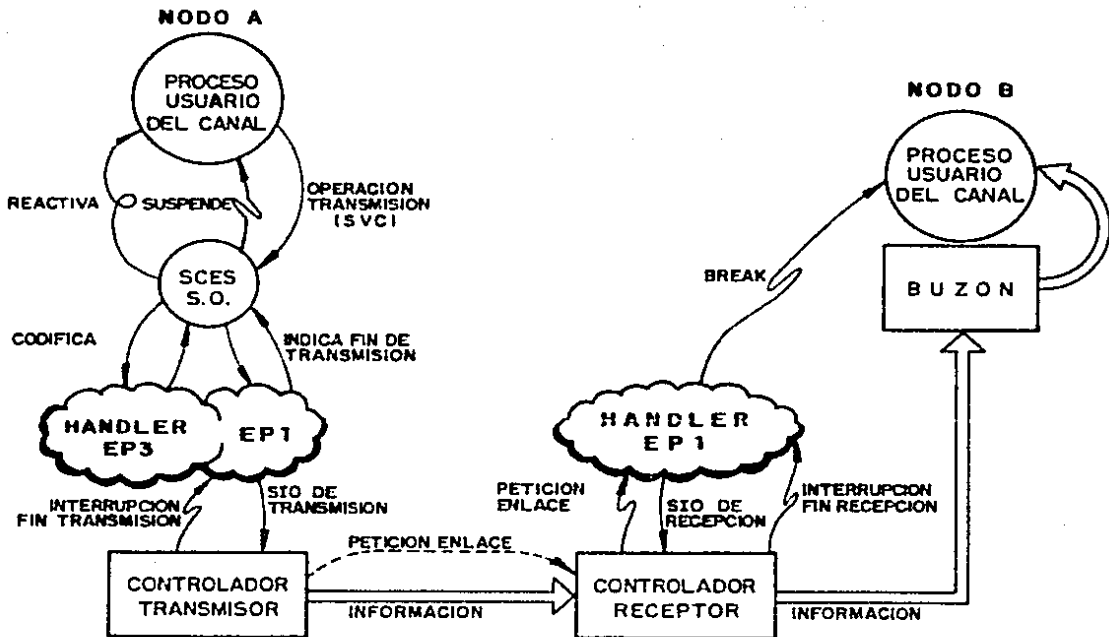


FIG. 3.12.- ESQUEMA DE UNA OPERACION DE TRANSFERENCIA

3.2.2.1 ESTRUCTURAS DE COMUNICACION -

La recepción de información en un nodo se realiza a través del buzón de recepción, que definimos por la necesidad de procesar las recepciones en modo asincrónico. Este modo nos permite mantener "aislado" al usuario del canal hasta la recepción completa de la información.

El buzón de recepción es una estructura (Fig. 3.13) alojada en una partición estática de memoria y está organizada como una cola tipo FIFO (First-In, First-Out); es utilizada por el handler, por el usuario del canal y por el controlador.

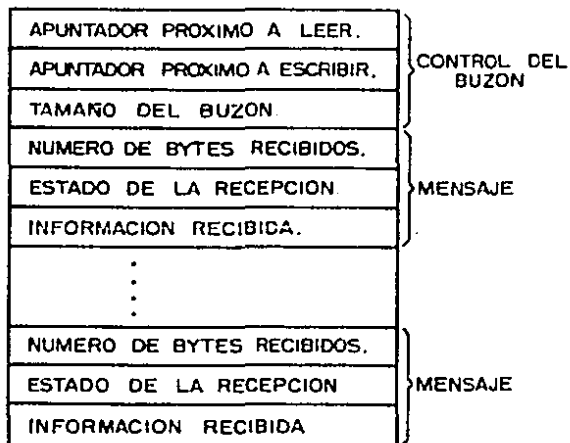


FIG. 3.13.- ESTRUCTURA DEL BUZON DE RECEPCION.

El buzón esta dividido en dos secciones principales. La primera sección es la de control del FIFO, y la segunda contiene los mensajes recibidos a través del canal de comunicación.

Cada mensaje esta formado por un encabezado, más el contenido del mensaje. El encabezado contiene el número de bytes de la información recibida y el estado de la recepción. En el contenido del mensaje está codificada la siguiente información:

- Tipo de mensaje.
- Longitud del mensaje.
- Nombre de los procesos transmisor y receptor.
- Dirección inicial del bloque de información en el caso de la transferencia de memoria compartida.
- Tabla de posición de nodos para el caso de la reconfiguración de la red.

Por cada nodo adyacente se tiene un buzón de recepción independiente y asociado a un handler. El usuario del canal se encargará de discriminar por cual de los buzones tomará el mensaje.

El bloque de definición de operación (BDO) es una estructura interna del S0 que maneja el SCES y que se utiliza para proporcionarle la información relativa a las operaciones solicitadas sobre los dispositivos de E/S que el S0 puede manejar, y además en él reporta el estado de las operaciones solicitadas.

En el caso de la red CP/32, el BDO es utilizado para definir las operaciones de transmisión a través del canal, y a su vez para reportar el estado de dichas operaciones. Se contruye con cada solicitud de transmisión de información en la interfaz que se ofrece al usuario del canal, como explicaremos más adelante.

La estructura general del BDO se muestra en la Fig. 3.14.

| | CODIGO DE OPERACION. | CODIGO CODIGO |
|---|----------------------|---------------|
| TIPO DE COMANDOS AL CONTROLADOR. | | |
| ESTADO DE LA OPERACION. | | |
| APUNTADOR A LA COLA DE E/S | | |
| DIRECCION DE DATOS. | | |
| PALABRAS ADICIONALES DE ESTADO DE LA OPERACION. | | |

FIG. 3.14.- ESTRUCTURA GENERAL DEL BDO.

3.2.3 LA CAPA DE TRANSPORTE -

Habiendo definido un nivel que pudiera interactuar con el medio físico (el handler), podemos entonces establecer el siguiente nivel de abstracción de la arquitectura de la red CP/32, la capa de transporte, último nivel del bloque de comunicación; la función de esta capa es de mantener el control de la información en la red, así como del estado operativo de los nodos que forman la red y realizar las funciones básicas de comunicación en tiempo real entre procesos ubicados en distintas computadoras.

Como mencionamos anteriormente, la capa de transporte la diseñamos como un proceso en tiempo real [2], el sistema de comunicación - usuario del canal-, que se encuentra ubicado en cada nodo. Las funciones básicas que realiza son las transferencias entre los nodos tanto de mensajes de los procesos, como del contenido de las áreas de memoria compartida.

En la Fig. 3.15 se muestra cómo interactúa el sistema de comunicación; los procesos solicitan al sistema sus transferencias de mensajes y memoria compartida, y éste las realiza por medio del BDO; cuando llegan mensajes al buzón procedentes de los nodos adyacentes, el sistema de comunicación los envía a los procesos correspondientes a través del mecanismo de mensajes del SO.

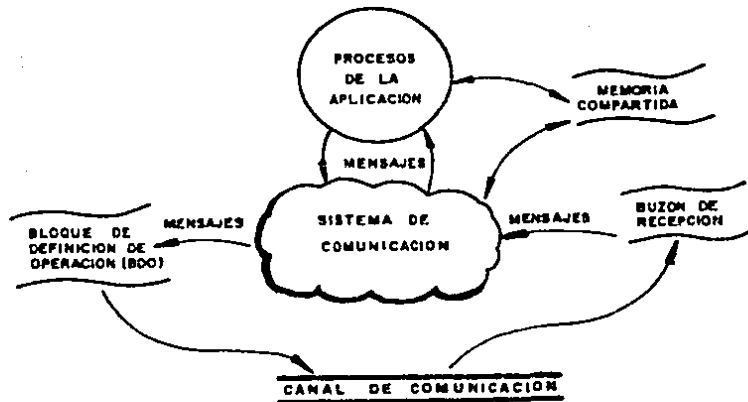


FIG. 3.15- SISTEMA DE COMUNICACION EN LA CAPA DE TRANSPORTE.

Las funciones de comunicación que realiza el sistema de comunicación son:

- Transmisión/recepción de mensajes entre procesos ubicados en distintos nodos, con opción de reconocimiento por parte del nodo receptor.
- Copiar zonas de memoria compartida entre los procesos de un nodo a otro distinto por solicitud de un proceso. En este caso, el diseñador de los procesos se encarga de la organización de la memoria compartida en cada nodo.
- Establecer un mecanismo de vigilancia cruzada entre los nodos adyacentes.
- Actualizar el estado operativo de la red.
- Notificación de los procesos que lo soliciten, de las fallas en la red, así como el estado de la comunicación con los nodos adyacentes.

El procedimiento general del sistema de comunicación es el siguiente:

Procedimiento SC.

Inicialización del sistema (abrir/adjudicar canales, verificar estados de operación, inicializar buzones de recepción)

HACER siempre

Establecer enlace con nodos adyacentes

HACER mientras no falla la transmisión

Suspenderse de ejecución por un tiempo t

SI terminó el tiempo t ENTONCES

transmitir mensaje de vigilancia a

ambos nodos (procedimiento de transmisión)

verificar el estado de la transmisión

SINO

procesar solicitudes de transmisión

(procedimiento de transmisión)

verificar el estado de las transmisiones

FIN SI

SI mensajes en buzón recepción ENTONCES

procesar todos los mensajes del buzón

(procedimiento de recepción)

FIN SI

FIN HACER mientras

FIN HACER siempre

FIN procedimiento SC.

La conexión de la red CP/32 se realiza con enlaces bidireccionales; aprovechando este tipo de enlace se optimizó de un modo sencillo la transmisión de los mensajes entre los nodos, seleccionando la trayectoria más corta al nodo receptor.

Para lograr esta optimización se utilizó una tabla de posición de los nodos; cada entrada en la tabla representa la posición de un nodo j (columna) con respecto al nodo i (renglón). La posición relativa de un nodo con respecto a otro está dada por la dirección física del controlador, así,

por ejemplo si las direcciones de los dos controladores de cada nodo adyacente son 40 y 50, entonces un nodo cualquiera quedará localizado a la derecha (D = 40) o a la izquierda (I = 50) de ese nodo. La Fig. 3.16 muestra un ejemplo de la tabla de posición para una red con 5 nodos.

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | X | D | D | I | I |
| 2 | I | X | D | D | I |
| 3 | I | I | X | D | D |
| 4 | D | I | I | X | D |
| 5 | D | D | I | I | X |

FIG. 3.16- TABLA DE POSICION PARA UNA RED DE 5 NODOS

Esta tabla se requiere para la localización del nodo destino en el proceso de transmisión. Para la transmisión de un mensaje, el proceso que transmite deberá especificar la identificación del nodo destino, así como la opción de reconocimiento del nodo destino, es decir, el proceso especificará si desea esperar por el reconocimiento del mensaje que transmite.

El siguiente es el procedimiento del sistema de comunicación para la transmisión de un mensaje.

Procedimiento transmisión.

Conseguir la posición del nodo destino
(posición = D o I)

SI reconocimiento ENTONCES
suspender de ejecución al proceso transmisor
agregar identificador de mensaje a la tabla
de mensajes pendientes
establecer valor máximo de espera por
reconocimiento (temporizador)

FIN SI
agregar al mensaje el identificador del
nodo origen

transmitir el mensaje al nodo destino
(con la rutina ESCRIBECANAL)

analizar el estado de la transmisión
(con el procedimiento analizafalla)

Fin procedimiento transmisión.

El procedimiento que analiza el estado de la operación de transmisión es el siguiente:

Procedimiento analizafalla.

SI falla en transmisión ENTONCES
actualizar la tabla de posición de nodos
notificar a los nodos de la actualización
SI nodo destino = nodo fallado ENTONCES
notificar falla al proceso en nodo origen

FIN SI
transmitir el mensaje por el nodo opuesto
(con la rutina ESCRIBECANAL)

SI falla en segunda transmisión ENTONCES
regresa al estado de inicialización del
sistema y trata de establecer de nuevo el
enlace con los nodos adyacentes.

FIN SI

FIN SI

Fin procedimiento analizafalla.

La recepción de un mensaje es un proceso un poco más complejo, ya que es necesario llevar un control de los mensajes que requieren reconocimiento del nodo destino.

Existen varios algoritmos que logran el reconocimiento de mensajes, y para la red se adaptó el algoritmo de ventanas [21]. Este consiste en un arreglo (tabla de mensajes pendientes) de n entradas (Tanenbaum [20] prueba que el número óptimo es 8). Cada entrada contiene el identificador de los mensajes por los que no se ha recibido reconocimiento y tiene asociada un temporizador que indica el valor máximo de espera por el reconocimiento para el mensaje correspondiente, así como el identificador del proceso transmisor. Cuando se recibe el reconocimiento de un mensaje, entonces se libera la entrada que corresponde a ese mensaje. Si el tiempo asociado a un mensaje expira, entonces se hace continuar la ejecución del proceso suspendido, indicándole la pérdida de su mensaje.

La estructura de la tabla de mensajes pendientes se muestra en la Fig. 3.17.

| IDENTIFICADOR MENSAJE | IDENTIFICADOR PROCESO | TEMPORIZADOR |
|--|--|--|
| <div style="text-align: center;">⋮</div> | <div style="text-align: center;">⋮</div> | <div style="text-align: center;">⋮</div> |
| | | |

FIG. 3.17.- ESTRUCTURA DE LA TABLA DE MENSAJES PENDIENTES

Cuando el sistema de comunicación es notificado de la llegada de un mensaje (por el handler), lo toma del buzón de recepción y analiza su encabezado para determinar el tipo de mensaje y entonces procesarlo. El siguiente es el proceso de recepción de mensajes del sistema de comunicación.

Procedimiento recepción.

nodo n recibe mensaje k en el buzón

consigue el mensaje del buzón (con la rutina LEEBUZON)

SI mensaje k es para el nodo n ENTONCES

 SI mensaje k es de reconocimiento ENTONCES

 actualizar la tabla de mensajes pendientes

 activar la ejecución del proceso correspondiente

 SINO

 enviar mensaje al proceso indicado en el mensaje k

 SI proceso está activo ENTONCES

 SI opción con reconocimiento de k ENTONCES

 conseguir posición del nodo origen

 a partir de la tabla de posición

 transmitir mensaje K de reconocimiento

 al nodo origen indicando además la

 existencia del proceso

 (con la rutina ESCRIBECANAL)

 analiza el estado de la transmisión con el

 procedimiento analizafalla.

 FIN SI

 SINO

 SI opción con reconocimiento de k ENTONCES

 conseguir posición del nodo origen

 a partir de la tabla de posición

 transmitir mensaje K de reconocimiento

 al nodo origen

 (con la rutina ESCRIBECANAL)

 analiza el estado de la transmisión

 con el procedimiento analizafalla.

 FIN SI

 FIN SI

FIN SI

SINO

 determinar nodo a transmitir a partir del nodo

 que envió el mensaje

 transmitir el mensaje k al nodo determinado

 (con la rutina ESCRIBECANAL)

FIN SI

Fin procedimiento recepción.

3.2.4 LA CAPA DE APLICACION -

La red CP/32 se diseñó no sólo para cumplir los requerimientos particulares del SAD, sino para cumplir también los requerimientos generales de cualquier aplicación orientada al procesamiento distribuido.

Las capas descritas hasta el momento forman lo que llamamos el bloque de comunicación; este bloque define el alcance de la comunicación en la red, es decir, las funciones de comunicación y el control de la red estarán bien establecidos en este bloque.

Ahora bien, la siguiente capa en la arquitectura de la red la definimos como la capa de aplicación - usuario de la red-.

En la práctica, cualquier aplicación puede ser implantada en esta arquitectura, por ejemplo los sistemas para la automatización de oficinas, los simuladores para entrenamiento (salas de control) y bases de datos distribuidas entre otros.

La aplicación actual de la red CP/32 es en el sistema tolerante a fallas del Sistema de Adquisición de Datos (SAD) para centrales termoeléctricas; el SAD se especificó con una disponibilidad de operación del 98.9 % [7], la cual requirió que se diseñara con un sistema tolerante a fallas. Antes de presentar esta aplicación, definiremos el contexto del término "tolerancia a fallas".

Weitzman [26] establece que "tolerancia a fallas puede ser definido como la capacidad de un sistema de superar las fallas de hardware y/o los errores de software sin intervención humana".

Una de las técnicas para implantar esta capacidad de tolerancia a fallas es con sistemas reconfigurables, es decir, para una aplicación en particular y dependiendo del nivel de tolerancia a fallas que se requiera, éste se puede obtener duplicando componentes.

Los atributos básicos que cualquier sistema con capacidad de tolerancia a fallas debe tener, los define Russell [17] como:

- Detección rápida de las fallas.
- Habilidad para aislar la falla detectada, y de acuerdo a su evaluación y confinamiento conmutar al equipo alterno y continuar en operación.
- Permitir la reparación del equipo que falló mientras el sistema de aplicación continúa en operación.
- Habilidad para restablecer nuevamente en la configuración el equipo reparado.

El sistema que logra la capacidad de tolerancia a fallas del SAD es el Sistema Distribuido de Respaldo y Recuperación de Fallas (SDRF). Este sistema ocupa un primer nivel de aplicación de la red CP/32, que además se diseñó para manejar otras aplicaciones.

La arquitectura de hardware del SDRF se basa en un sistema dual de computadoras (red CP/32 con dos nodos) y cada una incluye, además de un CPU y módulos de memoria, una unidad de disco y su procesador, un procesador de E/S, un módulo de tiempo real y un acelerador de operaciones en punto flotante, además de una unidad de cinta y su procesador.

Partiendo de esta configuración, la capa física quedó integrada con dos controladores, uno por cada computadora. Además, con esta arquitectura de hardware, el diseño de la red CP/32 [10] que se llevó a cabo es un subconjunto del alcance de las funciones y algoritmos presentados para una red con más de dos nodos [8]; por ejemplo, para esta aplicación no fue necesario implantar en la red la optimización en la transmisión de los mensajes.

Presentaremos ahora el panorama general del SDRF, analizando el procedimiento de detección de fallas y evaluación del daño, así como el mecanismo de recuperación en que se basa.

El conjunto de fallas que considera el SDRF se clasificaron [14] según la fuente de la falla (hardware, software y falla de energía) y el tiempo de su ocurrencia (a priori y a posteriori), es decir, una falla a priori se detecta antes de que pueda causar un daño al sistema, mientras que en una falla a posteriori se considera que el daño causado es quien manifiesta la falla. De todas las posibles

combinaciones, en el SDRF se consideran las siguientes:

- fallas de hardware a priori y a posteriori
- fallas de software a posteriori, e
- interrupción del suministro de energía.

La detección de las fallas de hardware a priori se realiza a través de un proceso de diagnóstico en línea - que forma parte del SDRF-, cuyo principio consiste en realizar operaciones sobre los dispositivos críticos de cada computadora (todos los mencionados anteriormente exceptuando la unidad de cinta y su procesador). En la detección de fallas de hardware a posteriori se utiliza el mecanismo de vectorización de excepciones del SO, mientras que las fallas de software a posteriori se detectan utilizando un procedimiento de vigilancia basado en contadores asociados a cada proceso de la aplicación (en este caso el SAD). Cabe mencionar que el bloque de comunicación de la red realiza otro nivel de detección de fallas, a través de su mecanismo de vigilancia cruzada entre las computadoras.

Para la evaluación del daño se establece el siguiente criterio. A cada dispositivo que conforma la computadora y a cada proceso de la aplicación se le asocia un nivel de severidad, siendo el nivel uno a los dispositivos y procesos no críticos, es decir, el nivel de severidad uno implica únicamente el reporte de la ocurrencia de la falla; el nivel de severidad dos es asociado con los dispositivos y tareas críticos y la falla de uno de ellos implica una acción de

recuperación de la falla.

Antes de resumir el procedimiento de respaldo y recuperación, tomaremos en cuenta las siguientes consideraciones con respecto al SDRF:

1. La configuración operativa del SDRF es una computadora activa - primaria-, que mantiene la ejecución de los procesos de aplicación y la otra computadora de reserva - respaldo-, en la que se mantiene una copia de la información manejada por la primaria.
2. El SDRF es un proceso ejecutado en ambas computadoras y es quien vigila la operación de las mismas.
3. La computadora de respaldo, además de asumir su función específica, permite la ejecución de otras aplicaciones (simulador del SAD para entrenamiento de operadores) y el desarrollo y mantenimiento de programas.

El procedimiento de respaldo es un proceso distribuido, realizado por el SDRF en ambas computadoras y se resume en lo siguiente. El área de memoria compartida de cada computadora es dividida lógicamente en secciones correspondientes y los procesos de aplicación son agrupados de acuerdo a la sección que "ven". Además, el grupo de procesos tiene asociados ciertos archivos comunes; entonces, el estado de un grupo en función del tiempo se definió como el área de memoria y los archivos correspondientes al grupo en un instante dado.

Durante su ejecución, los procesos (en sincronía todos los que pertenecen a un grupo) solicitan el respaldo de su estado - definido como punto de respaldo-, entonces el SDRF, a través de la red, copia en la computadora de respaldo el estado del grupo en sincronía. Esta actualización en la

computadora de respaldo se realiza mientras los procesos en la computadora primaria continúan su ejecución (ya han terminado su punto de respaldo) [9].

Cuando ocurre una falla en la computadora primaria, el SDRF transfiere el control de los procesos de aplicación en forma inmediata y automática a la computadora de respaldo, iniciándose el proceso de recuperación. Los procesos reiniciarán su ejecución a partir de su último estado existente en la computadora de respaldo.

Cabe hacer notar que el proceso de respaldo y recuperación es totalmente "transparente" para los procesos de la aplicación, y los servicios para llevarlo a cabo se implantaron a través de una interfaz para el usuario [15].

El segundo nivel de aplicación lo forma el SAD. El SAD para centrales termoeléctricas proporciona beneficios que permiten la toma de decisiones en la operación, supervisión y planeación de la planta de una forma sencilla y oportuna.

Las funciones principales del SAD son:

- Adquisición y monitoreo en tiempo real de la información relacionada con el estado de la planta.
- Auxiliar a los operadores mediante la presentación adecuada de la información adquirida.
- Monitoreo y registro de secuencia de eventos, tanto de disparo de equipo como de procedimientos de operación.
- Despliegue de guías especializadas de operación en caso de alarma.
- Generación de reportes con información histórica y estadística para evaluar y planear políticas de operación de la planta.
- Monitoreo y registro de tiempos de operación de equipos y generación de reportes para la programación de actividades de mantenimiento.

3.2.5 DETECCION DE FALLAS EN LA COMUNICACION -

Como en todos los sistemas de transmisión de información, es importante considerar las fallas ocurridas en los medios utilizados para la transmisión.

Ya hemos visto las fallas que se pueden detectar en el canal de comunicación (por la capa física y a su vez por la de transporte); además de estas fallas, se detectan la falla de uno ó más nodos de la red (a través del mecanismo de vigilancia) y los errores por distorsión en la información; éstos últimos son detectados con el algoritmo de CRC (cyclic redundant check) [4], y se considera que son causados por falla en los controladores y no por ruido, por tanto es una falla constante y en este caso también se declarará fallado el canal.

Cuando ocurre una falla en transmisión, la computadora que lo detecta retransmitirá el mensaje por el nodo opuesto e indicará la falla del nodo correspondiente. Los procedimientos para la recuperación realizados al ocurrir una falla están implantados en los procedimientos de transmisión y recepción utilizados por el sistema de comunicación.

En un estudio realizado con base en datos proporcionados por el fabricante de los equipos utilizados en la implantación de la red [16], se muestra que tanto el canal de comunicación como la computadora tienen una disponibilidad [26] del 99.88 % y 98.76 % respectivamente. La alta disponibilidad mostrada, nos asegura la operación de la red en un ambiente confiable y con seguridad.

CAPITULO 4

INTERFACES Y PROTOCOLOS

En el capítulo 3 se presentó la arquitectura de la red CP/32 desde el punto de vista funcional. La metodología utilizada - capas o niveles de abstracción- en el diseño de la arquitectura es la técnica de ingeniería de software de mayor aceptación para el diseño de los sistemas de comunicación [27], y tiene como objeto conceptualizar independientemente cada una de las funciones que se requieren para la transmisión de información a través de los medios de comunicación.

Cada capa realiza un subconjunto de dichas funciones (la máquina virtual) para ofrecer servicios específicos a la capa superior, y cuando es necesario, utiliza los servicios ofrecidos por la capa inferior. Los servicios que una capa ofrece a otra superior se implantan a través de las llamadas rutinas de interfaz, mientras que la comunicación entre cada capa se define por medio de reglas o convenciones llamadas protocolos [18].

En este capítulo trataremos de los protocolos de comunicación y las interfaces para los servicios de cada una de las capas de la red CP/32. La Fig 4.1 muestra las

interfaces y protocolos de comunicación en la red CP/32 que se describen a continuación.

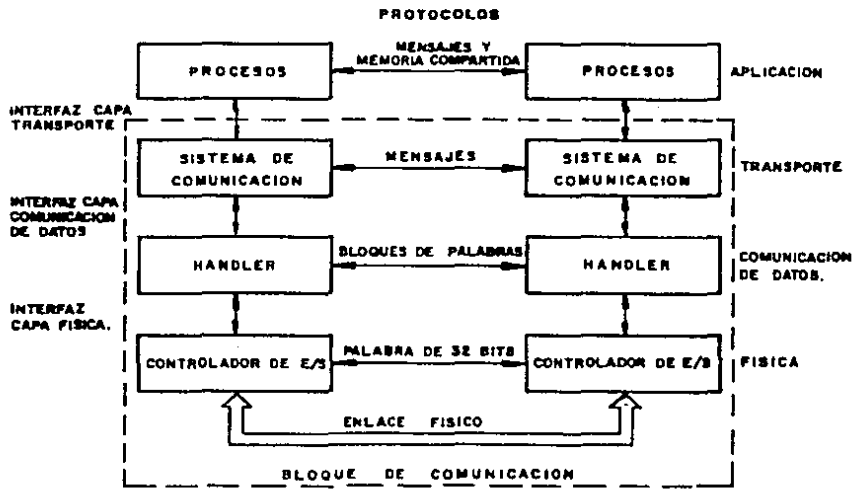


FIG. 4.1.- INTERFACES Y PROTOCOLOS DE LA RED CP/32.

4.1 PROTOCOLOS

CAPA FISICA. El protocolo de la capa física consiste en palabras de 32 bits, transmitidas y recibidas de acuerdo al protocolo de enlace definido en la capa física.

CAPA COMUNICACION DE DATOS. La comunicación establecida entre el handler de cada nodo es por medio de bloques de palabras. Cada programa de canal define los bloques de palabras transferidas de un handler a otro.

CAPA TRANSPORTE. En este nivel, la comunicación es por medio de mensajes, en donde cada uno de ellos tiene un formato específico. Los tipos de mensaje son:

- De proceso a proceso.
- Notificación de transferencia de memoria compartida.
- Actualización de la tabla de posición.
- Reconocimiento de un mensaje.
- Vigilancia entre nodos.

CAPA APLICACION. La comunicación entre procesos es por medio de mensajes y memoria compartida. En este nivel, el diseñador de los procesos de aplicación define el formato de los mensajes, así como la estructura de la memoria compartida en cada nodo.

4.2 INTERFACES

A continuación se describen las interfaces entre las distintas capas que forman la red.

INTERFAZ CAPA FISICA. Hemos visto en el capítulo 3 que la comunicación entre el handler y el controlador se establece por medio de las instrucciones de E/S y de las interrupciones generadas por el controlador. Esta es en sí, la interfaz de la capa física y la de más bajo nivel en la que se puede observar la interacción entre las capas de comunicación de datos y la física.

INTERFAZ CAPA COMUNICACION DE DATOS. Los servicios que ofrece el handler al sistema de comunicación se realizan con las siguientes rutinas:

LEEBUZON. Lee un mensaje del buzón de recepción.

LEESTADO. Lee del buzón de recepción el numero de bytes y el estado de una operación de recepción.

INICIABUZON. Asocia el buzón de recepción al programa que forma el sistema de comunicación, además de inicializarlo.

LIBERABUZON. Libera el buzón de recepción del programa que forma el sistema de comunicación.

ESCRIBECANAL. Transmite un mensaje a través del canal de comunicación.

ABRECANAL, CIERRACANAL. Adjudica/libera lógicamente el canal de comunicación al programa que forma el sistema de comunicación.

Para la aplicación en la que se utilizó la red, estas rutinas fueron escritas en lenguaje ensamblador de la máquina GOULD; el procedimiento general que realiza cada una de estas rutinas se describe a continuación.

Procedimiento leebuzón.

Consigue los apuntadores de control del buzón

Consigue las direcciones del buzón y del buffer destino

Tranfiere la información del buzón al buffer destino

Ajusta los apuntadores de control del buzón

Verifica el estado de la operación

Fin procedimiento leebuzón.

Procedimiento leestado.

Consigue la dirección del buzón que se desea analizar (se especifica como argumento el buzón)

Consigue los apuntadores de control de buzón

Consigue del buzón el número de bytes de información recibida y el estado de la recepción

Asigna los argumentos de salida

Fin procedimiento leestado.

Procedimiento inicia/libera buzón.
Consigue la dirección del buzón
Inicializa apuntadores de control del buzón
o libera buzón
Llama al EP3 del handler, a través del SVC
correspondiente
Fin procedimiento inicia/libera buzón.

Procedimiento escribecanal.
Construye segunda parte del BDO con la dirección del
buffer a transmitir
Realiza validación del buffer que se transmitirá
Realiza la escritura a través de un SVC
Verifica el estado de la operación
Fin procedimiento escribecanal.

Procedimiento abre/cierra canal.
Construye primera parte del BDO a partir de los argumentos
Adjudica y abre canal ó lo cierra a través de un SVC
Verifica la realización de la operación
Fin procedimiento abre/cierra canal.

INTERFAZ CAPA TRANSPORTE. En este nivel se ofrecen los servicios de comunicación a los procesos de la capa de aplicación y se realizan con las siguientes rutinas:

TRANSMITEMENSAJE. Este servicio transmite el mensaje de un proceso a otro proceso en el nodo especificado.

TRANSMITEMEMORIA. Este servicio se utiliza para transmitir una área de memoria compartida, especificando el número de palabras a transmitir, la dirección inicial de memoria y el nodo destino.

NOTIFICAFALLA. Servicio utilizado por un proceso para que le sea informado eventualmente la ocurrencia de una falla en el canal.

ESTADOCANAL. Este servicio es utilizado para solicitar el estado de la comunicación con los nodos adyacentes.

ACTUALIZATABLA. Con este servicio se le indica al sistema de de comunicación que actualize la tabla de posición de los nodos.

El objetivo de la interfaz de la capa de transporte es facilitar el uso de los servicios de comunicación; en general, el procedimiento de las rutinas que forman esta interfaz es construir el mensaje correspondiente al servicio solicitado y enviarlo (a través del mecanismo del SO) al sistema de comunicación. Este procedimiento se describe a continuación.

Procedimiento interfaz.

Construye el buffer que contiene el mensaje, indicando el código del mensaje y argumentos adicionales.

Construye el buffer que define el mensaje (manejado por el SO), en el que se especifica el nombre del proceso receptor, el número de bytes y la dirección del buffer que contiene el mensaje.

Valida la existencia del sistema de comunicación y, dependiendo de ésta, envía el mensaje a través de un SVC.

Verifica el estado de la operación.

Fin procedimiento interfaz.

CAPITULO 5

CONCLUSIONES

En este trabajo se presentó el diseño de la arquitectura de una red local para la comunicación entre procesos en tiempo real. La red fue diseñada para sistemas de aplicación en tiempo real.

Un sistema en tiempo real es aquel cuyas funciones son realizadas en un ambiente con restricciones tales como el tiempo de respuesta, la disponibilidad y confiabilidad [2].

La red CP/32 es una extensión al mecanismo de comunicación entre procesos de un SO en tiempo real y que realiza sus funciones permitiendo al sistema de aplicación cumplir con sus restricciones. Las funciones básicas de comunicación que realiza son la transferencia de mensajes entre procesos y la transferencia del contenido de áreas de memoria compartida, además de mantener el estado operativo de la red.

La aplicación en la que se utilizó la red es en el Sistema Distribuido de Respaldo y Recuperación de Fallas del Sistema de Adquisición de Datos para centrales termoeléctricas, que se basó en una arquitectura dual de computadoras. Con esta arquitectura se comprobó que el promedio estimado de tiempo para la ejecución de los servicios

de comunicación es de 0.0026 seg.

En general, para determinar el número de nodos que una arquitectura basada en la red CP/32 puede manejar se deberá tomar en cuenta las restricciones de tiempo de la aplicación; así si t_r es el tiempo requerido por la aplicación y t_c es el tiempo promedio de realización de un servicio de comunicación, entonces el número n de nodos para la arquitectura deberá cumplir la relación

$$2t_c(n-1) < t_r$$

en donde el tiempo de transmisión del nodo 1 al nodo n se considera igual al tiempo de transmisión del nodo n al nodo 1 (reconocimiento de mensajes).

El diseño de la red CP/32 se realizó con una de las técnicas de ingeniería de software de mayor aceptación en el diseño de sistemas de comunicación: la definición de capas o niveles de abstracción, que definen cada una de las funciones que se requieren para la transmisión de información a través de los medios de comunicación. Esta metodología es la utilizada por la Organización Internacional de Normalización (ISO) para la definición de su Modelo de Referencia.

El diseño de la red implicó la implantación de:

- las capas de comunicación, y
- las interfaces y protocolos correspondientes.

REFERENCIAS

- [1] Alabau, A., Ed.,
TELEINFORMATICA Y REDES DE COMPUTADORES,
Marcombo, 1983.
- [2] Allworth, S.T.,
INTRODUCTION TO REAL-TIME SOFTWARE DESIGN,
Springer-Verlag, N.Y., 1981.
- [3] Anderson, George A., Jensen, D.,
COMPUTER INTERCONNECTION STRUCTURES:
TAXONOMY, CHARACTERISTICS AND EXAMPLES,
Computing Surveys, Vol. 7-4,
Diciembre 1975, p.197-213.
- [4] Aram Perez, Wismer and Becker,
BYTE-WISE CRC CALCULATIONS,
IEEE Micro, Junio 1983.
- [5] Chou, W., Ed.,
COMPUTER COMMUNICATIONS, VOLUME I -PRINCIPLES,
Prentice-Hall, 1983.
- [6] Deitel, Harvey M.,
AN INTRODUCTION TO OPERATING SYSTEMS,
Adisson-Wesley, 1983.
- [7] Especificación para la instrumentación control y
automatización de la central termoeléctrica de
Manzanillo II, J-100,
Doc. C.F.E.
- [8] Fernández B., Roberto, César Freyre M.,
Carlos Ramirez V.,
MANUAL DE USUARIO DEL MECANISMO DE COMUNICACION
ENTRE COMPUTADORAS,
Doc. Depto. Simulación, I.I.E.,
Mayo 1986.
- [9] Fernández B., Roberto,
DOCUMENTO DE DISEÑO PARA EL PROGRAMA EJECUTIVO
DEL SISTEMA DE RESPALDO Y RECUPERACION DE
FALLAS DEL SADRE,
Doc. Depto. Simulación, I.I.E.,
Agosto 1985.
- [10] Freyre M., César,
DISEÑO DEL PROGRAMA MECANISMO DE COMUNICACION ENTRE
COMPUTADORAS DEL SISTEMA DE RESPALDO Y RECUPERACION
DE FALLAS DEL SADRE,

Doc. Depto. Simulación, I.I.E.,
Septiembre 1986.

- [11] Gould Inc.,
HIGH-SPEED INTER-BUS LINK, TECHNICAL MANUAL,
Marzo 1983.
- [12] Gould Inc.,
MPX-32 REFERENCE MANUAL, VOL. I,
Marzo 1983
- [13] Madnick, S.E., John J. Donovan,
OPERATING SYSTEMS,
McGraw-Hill, 1978.
- [14] Ramírez V., Carlos, Roberto Fernández B.,
César Freyre M.,
ESPECIFICACIONES PARA EL SISTEMA DE RESPALDO Y
RECUPERACION DE FALLAS DEL SADRE PARA LA CTE
MANZANILLO II,
Doc. Depto. Simulación, I.I.E.,
Abril 1986.
- [15] Ramírez V., Carlos, Ramón Castillo O.,
MANUAL DE USUARIO DE LOS SERVICIOS DEL SRS,
Doc. Depto. Simulación, I.I.E.,
Abril 1986.
- [16] Ramírez V., Carlos,
DISPONIBILIDAD DEL SADRE PARA LA CTE MANZANILLO II,
Doc. Depto. Simulación, I.I.E.,
Junio 1985.
- [17] Russell, P.J.,
THE DEVELOPMENT OF FAULT TOLERANT COMPUTER SYSTEMS
USING DUAL PROCESSING TECHNIQUES,
Software & Microsystems, Vol.4-3,
Junio 1985, p.71-74.
- [18] Stallings, William,
LOCAL NETWORKS,
Computing Surveys, Vol.16-1,
Marzo 1984, p.3-41.
- [19] Stallings, William,
LOCAL NETWORKS AN INTRODUCTION,
MacMillan, 1984.
- [20] Tanenbaum, Andrew S.,
COMPUTER NETWORKS,
Prentice-Hall, 1981.

- [213] Tanenbaum, Andrew S.,
NETWORK PROTOCOLS,
 Computing Surveys, Vol.13-4,
 Diciembre 1981.
- [223] Thurber, Kennet J.,
TECHNIQUES FOR REQUIREMENTS-ORIENTED DESIGN,
 Proceedings of the National Computer Conference,
 1977
- [233] Thurber, K. J., Masson, G. M.,
DISTRIBUTED-PROCESSOR COMMUNICATION SYSTEMS,
 Lexington Books, 1979.
- [243] Voelcker, John,
HELPING COMPUTERS COMMUNICATE,
 IEEE Spectrum, Marzo 1986, p.61-70.
- [253] Wecker, Stuart,
COMPUTER NETWORK ARQUITECTURES,
 IEEE Computer, Vol.12-9,
 Septiembre 1979, p.58-72.
- [263] Weitzman, Cay,
DISTRIBUTED MICRO-MINICOMPUTER SYSTEMS,
 Prentice-Hall, 1980.
- [273] Zimmerman, H.,
OSI REFERENCE MODEL -THE ISO MODEL OF ARQUITECTURE
 FOR OPEN SYSTEMS INTERCONNECTION,
 IEEE Transactions on Communication, Vol.COM-28,
 Abril 1980, p.425-432.

BIBLIOGRAFIA

- [28] Anderson, T.,
CAN DESIGN FAULTS BE TOLERATED?,
Software and Microsystems, Vol.4-3, Junio 1985, p.71-74.
- [29] Brow, Robert L., Peter J. Denning, Walter F. Tichy,
ADVANCED OPERATING SYSTEMS,
IEEE Computer, Vol.17-10, Octubre 1984.
- [30] Comer, Douglas,
OPERATING SYSTEM DESIGN: THE XINU APPROACH,
Prentice-Hall, 1984.
- [31] Dillon, T. S.,
APPLICATION OF DISTRIBUTED AND DEPENDABLE COMPUTER
SYSTEMS IN POWER SYSTEMS,
IEEE Electrical Power and Energy Systems, Vol.7-2,
Abril 1985.
- [32] Fortier, Paul J.,
DESIGN AND ANALYSIS OF DISTRIBUTED REAL-TIME SYSTEMS,
Mc.Graw Hill, 1985.
- [33] Graube, Maris, Michael C. Mulder,
LOCAL AREA NETWORKS,
IEEE Computer, Vol.17-10, Octubre 1984.
- [34] Harrison, Thomas J., Ed.,
MINICOMPUTERS IN INDUSTRIAL CONTROL. AN INTRODUCTION,
Instrument Society of America, 1978.
- [35] IEEE Computer, Vol.12-9, Septiembre 1979,
Ed. especial NETWORK PROTOCOLS.
- [36] IEEE Computer, Vol.11-1, Enero 1978,
Ed. especial DISTRIBUTED PROCESSING.
- [37] IEEE Communications Magazine, Vol.22-8, Agosto 1984,
Ed. especial ARQUITECTURES OF LOCAL AREA NETWORKS.
- [38] IEEE Transactions on Communication,
Vol.COM-28-4, Abril 1980,
Ed. especial COMPUTER NETWORK ARQUITECTURES AND PROTOCOLS.
- [39] Kleinrock, Leonard,
DISTRIBUTED SYSTEMS,
IEEE Computer, Vol.18-11, Noviembre 1985.
- [40] Reif, John H., Paul G. Spirakis,

REAL-TIME SYNCHRONIZATION OF INTERPROCESS COMMUNICATIONS,
ACM Transactions on Programming Languages and Systems,
Vol.16-2, Abril 1984, p.215-238.

- [413] Tebbs, David, Garfield Collins,
REAL-TIME SYSTEMS. MANAGEMENT AND DESIGN,
Mc.Graw Hill, 1977.
- [423] Wensley, J.H.,
FAULT TOLERANT TECHNIQUES FOR POWER PLANT COMPUTERS,
IEEE Transactions on Power Apparatus and Systems,
Vol.PAS-101-1, Enero 1982.
- [433] Wirth, Niklaus,
TOWARD A DISCIPLINE OF REAL-TIME PROGRAMMING,
Communications of the ACM, Vol.20-8, Agosto 1977.
- [443] Zorpette, Glenn, Ed.,
COMPUTERS THAT ARE NEVER DOWN,
IEEE Spectrum, Vol.22-4, Abril 1985, p.46-54.