

03063

2

24

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

COLEGIO DE CIENCIAS Y HUMANIDADES

UNIDAD ACADÉMICA DE LOS CICLOS

PROFESIONALES Y DE POSGRADO

INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS

APLICADAS Y EN SISTEMAS

ESPECIALIZACIÓN, MAESTRÍA EN CIENCIAS DE LA

COMPUTACIÓN

Sistema de Control de Transmisión de
Circuitos Virtuales (SCT/CV) para la Red-IIMAS

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A

OSCAR LUIS CANTU MARTINEZ

TESIS CON
FALLA DE ORIGEN

1988



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.

CONTENIDO

	Página		Página
Lista de Figuras	VI	Redes de Banda Ancha	28
Prefacio	1	Medio de Comunicación	28
CAPITULO II Redes de Computadores	5	Redes Tipo Anillo	28
Introducción	5	Redes Tipo Bus	28
¿Qué es una Red	6	Redes Tipo Estrella	29
El Porque de las Redes	8	Redes Tipo Anillo	29
Mayor Confiabilidad	9	Combinaciones para LANs	29
Mayor Relación Precio/Comportamiento	9	Bus o Anillo con	
Mayor Eficiencia de Operación	9	Cable de Pares Trenzados y Banda Base	29
Crecimiento Incremental	10	Bus o Anillo con	
Compartir Recursos	10	Cable Coaxial y Banda Base	29
Medio de Comunicación Poderosa	10	Bus de Alta Velocidad con	
Util para Organizaciones Distribuidas	11	Cable Coaxial y Banda Base	29
Objetivo de las Redes	11	Bus con	
Clasificación de Redes	12	Cable Coaxial y Banda Ancha	30
Topologías de Redes	14	Estrella con	
Estrella	14	Cable de Pares Trenzados y Banda Base	30
Anillo	14	Bibliografía	31
Lazo	15	CAPITULO II: Arquitecturas y Protocolos de Redes	33
Completamente Interconectada	15	Introducción	33
Bus	15	Protocolos e Interfaces	33
Arbol	15	Comunicación Física y Comunicación Virtual	34
Malla	17	Arquitecturas y Procesos Pares	36
Satélite	17	Modelo ISO de Referencia	37
Radio	17	Capa Física	38
Redes Punto-a-punto y Difusión	17	Capa de Enlace de Datos	38
Técnica de Conmutación	18	Capa de la Red	38
Conmutación de Circuitos	18	Capa de Transporte	40
Conmutación de Mensajes	19	Capa de Sesión	40
Conmutación de Paquetes	21	Capa de Presentación	41
Cobertura de la Red	21	Capa de Aplicación	42
Redes de Area Local (LAN)	21	Arquitectura de Red-IMMS	42
Redes Locales de Alta Velocidad (HSLN)	22	Capa de Comunicación de Datos	43
Redes de Area Extensa (WAN)	22	Capa Inter-Red	45
Interconexión de Redes	22	Capa de Transporte	45
Clasificación de Redes de Area Local	23	Capa de Aplicaciones	45
Esquema de Control y Acceso	23	Bibliografía	47
Redes Tipo Anillo	23	CAPITULO III: Protocolos de Transporte	49
Anillo Dividido	24	Introducción	49
Inserción de Registro	24	Servicios de la Subred	51
Acceso por Estafeta	25	Datagramas	52
Redes Tipo Bus/Arbol	25	Circuitos Virtuales	52
CSMA/CD	25	Servicio de Transporte	53
Acceso por Estafeta	26	Protocolo de Transporte	57
Accesos Múltiples por Multinivel	26	Direccionamiento de Procesos	57
Redes Tipo Estrella	27	Transferencia Básica de Datos	60
Esquema de Modulación	27	Manejo de Conexiones	61
Redes de Banda Base	27	Confiabilidad y Recuperación de Errores	62
		Control de Flujo y Almacenamiento Intermedio	64
		Multiplexaje	65
		Sincronización	65

	Página
Seguridad y Prioridad	66
Primitivas de una Interfaz de Transporte	66
Manejo de Puertos	67
SOLICITAR_PUERTO	67
LIBERAR_PUERTO	68
Manejo de Conexiones	68
CONNECTAR	68
ESPERAR_CONEXION	69
ACEPTAR_RECHAZAR	69
DESCONNECTAR	70
DESTRUIR	70
REINICIAR	71
Transferencia de Datos	71
ENVIAR	71
RECIBIR	72
INTERRUPCION Y ATENDERINT	72
Verificación y Cancelación	73
ESTADO	73
CANCELAR	73
Servicio de Datagramas	74
ENVIAR_DATAGRAMA	74
RECIBIR_DATAGRAMA	74
Bibliografía	76
CAPITULO IV: Protocolo del SCT/CV	77
Introducción	77
Subred de Comunicaciones de Red-ILMMS	77
Servicios de la Capa de Comunicación de Datos	79
Primitivas de Comunicación de Datos	80
Extensión de los Servicios por la Capa Inter-Red	80
Primitivas de Inter-Red	81
Servicios del SCT/CV	82
Protocolo del SCT/CV	83
Direccionamiento de Procesos	83
Transferencia Básica de Datos	86
Manejo de Conexiones	89
Establecimiento	90
Reinicialización	92
Terminación	92
Confiabilidad y Recuperación de Errores	94
Control de Flujo y Almacenamiento Intermedio	97
Multiplexaje	98
Sincronización	99
Seguridad	99
Prioridad	101
Restricciones por parte de la PDP bajo RSX-11M	101
Sistema Operativo RSX-11M	102
Memoria	102
DR11-B	103
Lenguaje Ensamblador	104
Bibliografía	105

	Página
CAPITULO VI: Implantación del SCT/CV Bajo RSX-11M	106
Introducción	106
Estructura Operativa	107
Interfaz de Comunicación de Datos	107
Sistema de Circuitos Virtuales	107
Interfaz de Usuario	109
Selección de la Estructura de Implantación	109
Comunicación entre Procesos	111
ACP	112
Biblioteca de Funciones	112
Estructura de Implantación	113
Manejador de Comunicaciones de la Red	116
Estructuras de Datos	119
Tabla de Puertos y Solicitudes (TPS)	119
Cancelación de Puertos	122
Tiempos de Espera (TIME-OUTS)	122
Conversión de Direcciones	124
Funciones de Bajo Nivel del MC	125
Activar Puertos	125
Liberar Puertos	126
Enviar Mensajes	127
Recibir Mensajes	127
Revisión de Estructuras	128
Biblioteca de Funciones	129
Estructuras de Datos	130
Bloque Descriptor de Puerto (BDP)	131
Buffers para Mensajes	134
Buffers de Estado	136
Colas de Solicitud	137
Tiempos de Espera (TIME-OUTS)	139
Bibliografía	141
CAPITULO VII: Guía del Usuario	142
Introducción	142
Interfaz de Usuario	143
Funciones de Manejo de Puertos	144
ARNEPT	145
CRARPT	147
ACTREC	148
Funciones de Manejo de Buffers de Comunicación	149
DAMEBF	150
LIBARBF	151
Funciones de Manejo de Conexiones de Transporte	152
CONNECT	155
ESPCON	158
DESCON	158
REINT	159
Funciones de Manejo de Mensajes de Comunicación	160
ENVMSJ	161
ESPMSJ	163
RCBMSJ	165

	Página
Funciones de Información de Estado _____	168
EDOPT _____	169
Definiciones de Usuario _____	171
CAPITULO VIII: Conclusiones _____	173
Características Principales del SCT/CV _____	173
Resultados Adicionales _____	177
Recomendaciones _____	178
APENDICES:	
A: Descripción de las funciones de acceso a la Capa de Comunicación de Datos _____	179
B: Códigos de Respuesta de las Funciones de Bajo Nivel del SCT/CV _____	182
GLOSARIO _____	183

LISTA DE FIGURAS

Figura 1.1	Evolución de los Esquemas de Acceso a los Recursos de Cómputo _____	7
Figura 1.2	Red de Computadoras _____	10
Figura 1.3	Topologías de Redes de Computadoras _____	12
Figura 1.4	Técnicas de Conmutación _____	20
Figura 2.1	Arquitectura de Cuatro Capas _____	30
Figura 2.2	Modelo de Referencia de ISO _____	32
Figura 2.3	Arquitectura de Red-IHMAS _____	34
Figura 2.4	Topología de Red-IHMAS _____	44
Figura 3.1	Esquema de Ubicación de un Protocolo de Transporte _____	50
Figura 3.2	Subredes de Comunicaciones _____	54
Figura 3.3	PT en Función del SS, Cuando el ST Permanece Constante _____	56
Figura 3.4	PT en Función del ST, Cuando el SS Permanece Constante _____	57
Figura 3.5	Servicios de un Protocolo de Transporte _____	58
Figura 3.6	Dirección Jerárquica _____	58
Figura 4.1	Direccionamiento entre Procesos _____	60
Figura 4.2	Formato de un Mensaje del SCT/CV _____	65
Figura 4.3	Diagramas de Transición del Manejo de Conexiones _____	91
Figura 4.4	Diagrama de Estado del Manejo de Conexiones _____	93
Figura 4.5	Diagrama General del Bloque Descriptor de Puerto del SCT/CV _____	100

	Página
Figura 5.1 Estructura Operativa del SCT/CV _____	108
Figura 5.2 Estructuras de Implantación _____	113
Figura 5.3 Estructura de Implantación del SCT/CV _____	117
Figura 5.4 Tabla de Puertos y Solicitudes (TPS) _____	120
Figura 5.5 Fórmula del BDP _____	133
Figura 5.6 Estructura para Manejo de Buffers _____	133
Figura 5.7 Estructura para Buffers de Estado _____	138
Figura 5.8 Manejo de Colas de Mensajes del BDP _____	138

PREFACIO

El gran auge e importancia que adquirió a finales de los setentas el campo de las redes de computadoras, a nivel internacional, motivó a un grupo de investigadores del IIMAS (Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas) de la UNAM (Universidad Nacional Autónoma de México), a participar en este campo, trabajando en el diseño y desarrollo de redes locales de computadoras. Así fue como se dió origen al Proyecto Red-IIMAS. Este proyecto fue conocido inicialmente como proyecto REDLAC, pero será referido únicamente como Red-IIMAS de Alta Velocidad o simplemente Red-IIMAS.

Red-IIMAS evolucionó, desde una red local de computadoras experimental de baja velocidad, hasta llegar a consolidarse en la actualidad, como una red local de computadoras experimental de alta velocidad.

En la fase de red de baja velocidad, Red-IIMAS contó con una arquitectura de cuatro capas extraída como una simplificación operativa del modelo de referencia propuesto por ISO (Organización Internacional de Estándares) para la interconexión de sistemas abiertos. Esta arquitectura fue implantada en las computadoras PDP-11s del Instituto. En esta versión, la Capa de Comunicación de Datos, formada por líneas de comunicación punto-a-punto y un protocolo sencillo, permitía la transmisión de información a 9,600 bps. La Capa de Transporte soportada por el sistema "Hermes", operaba en base a un protocolo de datagramas confiables y era

accedida por los procesos de aplicación desde rutinas desarrolladas en lenguaje ensamblador de PDPs (MACRO-11). Finalmente, la Capa de Aplicaciones contaba con un "logger" y un listador de archivos, ambos bastante sencillos.

En la evolución de la red de baja velocidad a la red de alta velocidad se modificó la arquitectura de Red-IIMAS, y no tanto en cuanto al número de capas, sino más bien en el contenido de las mismas. En esta fase, la Capa de Comunicación de Datos debería quedar en un sistema externo a los anfitriones que permitiera su comunicación a través de un canal compartido (cable coaxial) de alta velocidad (10 Mbps), utilizando un protocolo tipo Ethernet (CSMA/CD). Así mismo, la Capa de Transporte debería quedar soportada por un sistema que ofreciera un servicio fin-a-fin de comunicación de circuitos virtuales, que fuera confiable y eficiente, para los procesos de la Capa de Aplicaciones. Este sistema debería tomar los servicios de la Capa de Comunicación de Datos, aprovechar sus características principales y desarrollar los protocolos que le permitieran cumplir con los objetivos planteados para su servicio. Además, debería permitir su acceso a través de una interfaz más amistosa que la de su predecesor. Finalmente, la meta de la Capa de Aplicaciones fue desarrollar un nuevo logger (Activador de Procesos "AP") acorde con el nuevo protocolo de la Capa de Transporte, que apoyara la creación de múltiples instancias de servidores y reemplazar el listador de archivos de la red, por sistemas de transferencias de archivos, manejo de correo electrónico y manejo de terminales virtuales, entre otros.

En la fase actual de Red-IIMAS de Alta Velocidad, el Sistema de Control de Transmisión de Circuitos Virtuales (SCT/CV) es quien cumple con los objetivos planteados para el desarrollo de la Capa de Transporte, y constituye el material de la presente tesis.

La presentación de la tesis se encuentra dividida en siete capítulos, dos apéndices y un glosario de términos técnicos. De los siete capítulos, los dos primeros contienen información introductorio, el tercero proporciona la base teórica que apoya el diseño del SCT/CV y los cuatro últimos constituyen la información relevante al desarrollo del SCT/CV. El contenido de cada capítulo es el siguiente:

En el primer capítulo se proporcionan algunos conceptos importantes del campo de las redes de computadoras.

En el segundo capítulo se revisan conceptos relacionados con arquitecturas y protocolos de redes, se da una descripción del modelo de referencia propuesta por ISO, y se termina con una revisión de la arquitectura de Red-IIMAS de Alta Velocidad.

El capítulo tres contiene información referente a protocolos de transporte en general, donde se proporciona un resumen de aspectos importantes que deben ser considerados cuando se trabaja en el desarrollo de este tipo de sistemas.

En el capítulo cuatro se describen los puntos de mayor relevancia del proceso de diseño del protocolo del SCT/CV y en el capítulo cinco se relatan los aspectos importantes de la implantación del SCT/CV en las FDP-11s, bajo el sistema operativo RSX-11M.

El capítulo seis se dedica al manual de usuario del SCT/CV, donde se describen todas y cada una de las funciones de acceso a los servicios proporcionados por el sistema, así como cada uno de sus parámetros. Además, para cada función se proporciona un ejemplo de su manejo, para facilitar su comprensión.

En el capítulo siete, dedicada a las conclusiones del desarrollo, se describen las características más importantes del SCT/CV, consideradas como parte de los logros obtenidos, se describen algunos logros adicionales y se dan algunas recomendaciones para modificaciones futuras.

Finalmente, se incluyen dos apéndices y un glosario de términos técnicos. En el apéndice A se describen las funciones de acceso a la Capa de Comunicación de Datos y en el apéndice B se proporcionan los códigos de respuesta de las funciones de bajo nivel del SCT/CV. El glosario tiene la finalidad de aclarar posibles dudas que llegaran a surgir sobre la terminología manejada en este escrito.

CAPITULO I

Redes de Computadoras

INTRODUCCION

Aunque algunas de las ideas y conceptos de las redes de computadoras datan desde antes de los sesentas, no fue sino hasta fines de esta década cuando comenzaron a verse los primeros desarrollos del campo. Los factores que apoyaron esto fueron el grado de madurez y avance que alcanzaron en ese momento las tecnologías de comunicaciones y computación, así como la accesibilidad que lograron sus costos. Conforme prosiguió la incursión en el campo, los costos continuaron reduciéndose y las tecnologías siguieron evolucionando, con lo cual las redes adquirieron cada vez más una mayor importancia que dió origen en un tiempo muy corto, a una gran variedad de sistemas avocados a proporcionar siempre las mejores soluciones en el manejo informático y el procesamiento computacional en general.

Este capítulo tiene como objetivo proporcionar una breve ilustración de algunos aspectos importantes de este campo, tales como: que son las redes de computadoras, que beneficios proporcionan, cuáles son sus objetivos, y cuáles son los criterios más utilizados en su clasificación. Con esto, se intenta proporcionar un panorama general del campo, donde el IIMAS ha tratado de participar, incursionando en el análisis y solución de problemas relacionados con el diseño y desarrollo de la Red local IIMAS de alta velocidad, entre cuyos

componentes se encuentra el Sistema de Control de Transmisión de Circuitos Virtuales (SCT/CV) como el protocolo de la capa de transporte de la arquitectura de la red.

QUE ES UNA RED

Los inicios de la computación estuvieron caracterizados por los centros de cómputo, constituidos por una gran computadora (figura 1.1a), a la cual los usuarios llevaban sus trabajos para su procesamiento. Sin embargo, la madurez y conjunción de las tecnologías de comunicaciones y computación, así como el interés de realizar el trabajo computacional desde las áreas de trabajo, generó una fuerte influencia que cambió este concepto de usuarios llevando sus trabajos a la computadora, por otro en el cual, la computadora debería ser llevada a los usuarios.

En la evolución de llevar la computadora a los usuarios, el primer paso consistió en la extensión de líneas de acceso remoto, que permitieron que terminales, lectoras de tarjetas, impresoras y otros periféricos mas, comenzaran a ser ubicados fuera del centro de cómputo como puede verse en la figura 1.1b. Esto dió origen a los primeros tipos de acceso desde las áreas de trabajo de los usuarios. En su tiempo este tipo de conexiones se conocieron como redes de computadores, pero en la actualidad se describen mas acertadamente como redes de periféricos, redes de teleproceso o simplemente redes de acceso.

Continuando la evolución, el siguiente paso, mostrado en la figura 1.1c, implicó la sustitución de los grandes equipos por varios equipos más pequeños e independientes localizados en las áreas de

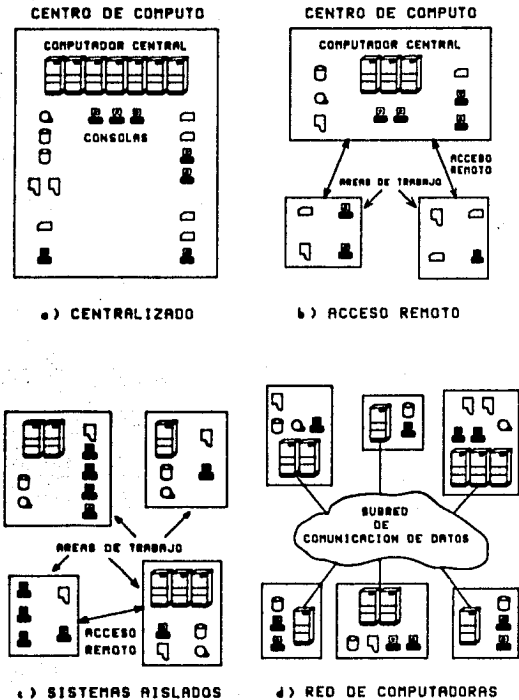


Figure 1.1
EVOLUCION DE LOS ESQUEMAS DE
ACCESO A LOS RECURSOS DE COMPUTO

trabajo. Aunque esta opción proporcionó mejoras en ciertos casos, no llegó a proporcionar la solución integral esperada. Para resolver las deficiencias de esta nueva configuración, se comenzaron a interconectar las máquinas dispersas en las áreas de trabajo, mediante lo que hoy en día se conoce como subredes de comunicación de datos. Ver figura 1.1d.

En la actualidad vamos como el modelo de una gran computadora y el de un conjunto de máquinas pequeñas aisladas, están siendo desplazados rápidamente por otro en el cual un conjunto de computadores independientes, pero interconectados, realizan el trabajo. Los sistemas conformados así son denominados redes de computadoras.

Varios han sido las definiciones planteadas para una red de computadoras, pero la más aceptada es la que lo define como un conjunto de computadoras autónomas interconectadas. Considerándose que dos o más computadoras se encuentran interconectadas si son capaces de intercambiar información. La interconexión puede darse a través de alambre de cobre, fibras ópticas, satélite, láser, etc. Para cumplir con el requerimiento de computadores autónomos, se excluyen de la definición a las configuraciones de máquinas con relaciones maestro esclavo, y los casos donde existan grandes computadoras con dispositivos remotos como terminales, impresoras, etc.

EL PORQUE DE LAS REDES

La estructuración de las capacidades de cómputo mediante varios computadoras autónomas interconectadas ha sido apoyado por el desarrollo y conjunción de las tecnologías de comunicaciones y

computación. Pero la razón para que los usuarios opten por esta nueva alternativa está respaldado por una serie de factores, de los cuales, los principales se describen a continuación:

Mayor Confiabilidad

En el caso de una máquina grande, si ésta falla se acaban todas las capacidades de cómputo, mientras que en el caso de una red de computadoras, si fallan algunas máquinas, normalmente siempre existirán recursos de cómputo disponibles.

Mejor Relación Precio/Comportamiento

El desarrollo formidable de los circuitos electrónicos VLSI, ha permitido que se puedan tener en espacios pequeños, computadoras que operan a muy buenas velocidades, con amplios recursos y a precios muy razonables. Esto ha permitido que varias máquinas pequeñas interconectadas proporcionen una mejor relación precio/comportamiento que el equivalente de una grande o varias pequeñas pero aisladas. Los costos de producción de muchas máquinas pequeñas son mas bajas que los de pocas máquinas grandes.

Mayor Eficiencia de Operación

En el caso de una máquina grande, ésta debe ser de propósito general, lo cual hace pensar que no puede ser realmente eficiente para todo tipo de procesamiento. En el caso de una red, es posible dedicar algunas máquinas a funciones específicas, como por ejemplo, una para el manejo de bases de datos, otra u otras para procesamiento numérico, otra para diseño ayudado por computadora, etc., obteniéndose una mayor eficiencia y un mejor comportamiento del sistema total.

Crecimiento Incremental

En el caso de una máquina grande, su expansión llega hasta un límite en el cual sólo queda cambiarla por otro modelo más grande, que eventualmente podría llegar al mismo caso. En las redes, esto es diferente ya que es factible iniciar con una configuración de pocas máquinas a un precio muy accesible e ir adicionando nuevas máquinas cuando sea necesario. En teoría, la expansión de una red es ilimitada.

Compartir Recursos

Las redes permiten compartir recursos de cómputo sin importar su ubicación. Esto proporciona facilidades alternativas que permiten un manejo mas adecuado de los mismos. Muchas han sido las organizaciones que en su crecimiento han adquirido varias máquinas para sus diferentes departamentos, las cuales, al encontrarse dispersos y aislados, no permiten compartir entre si sus recursos. La conexión de los computadores a través de una red puede permitir que compartan sus recursos, agilizando la operatividad global de la organización y reduciendo costos.

Medio de Comunicación Poderoso

Una red de computadoras proporciona una herramienta natural de apoyo para la comunicación. Lo cual permite que, por ejemplo, dos personas que se encuentren separadas físicamente por grandes distancias puedan participar en el desarrollo de trabajos conjuntos casi tan fácilmente como si estuvieran en el mismo sitio. También apoyan la comunicación local entre oficinas, empresas, etc., permitiendo el manejo de esquemas

de comunicación interactivos o de correo.

Util Para Organizaciones Distribuidas

Existen muchas compañías que son de naturaleza distribuida, donde su operación está dada por varios establecimientos ubicados en diferentes localizaciones geográficas, pero los cuales requieren ser regidos por la misma organización. En estos casos las redes de computadoras encuentran ubicación de manera natural y directa.

Estos son solo algunos de los porqués las redes de computadoras están adquiriendo gran importancia como una nueva alternativa, para la solución de problemas relacionados con la informática, las comunicaciones y la computación en general.

OBJETIVO DE LAS REDES

Las redes de computadoras nacieron en la búsqueda de una nueva estructura computacional que proporcionara beneficios adicionales a los de los esquemas tradicionales. No obstante, las redes han tenido que pasar por diversas etapas, y aun continúan madurando, y conforme vayan surgiendo mejores tecnologías y se vaya adquiriendo un mejor dominio del campo, las redes proporcionarán nuevos y mejores beneficios.

El objetivo del desarrollo de las redes de computadoras es llegar a obtener una buena integración de las máquinas y sus recursos, para así, poder llegar a proporcionar a los usuarios nuevas facilidades, así como eficientizar y mejorar otras que ya se dan en la actualidad. Además, también se espera que tanto las computadoras como la misma infraestructura de comunicaciones de la red, apoyen el desarrollo de

sistemas operativos distribuidos, los cuales, utilizando las capacidades de enlace y comunicación de la red, pueden realizar el manejo y administración de las computadoras y sus recursos, pero así poder llegar a ofrecerle a los usuarios ya no una computadora, sino un sistema de procesamiento (máquina virtual) de capacidades y recursos ilimitados.

CLASIFICACION DE REDES

La diversidad de redes de computadoras que se han desarrollado, ha propiciado el surgimiento de una variedad de criterios de clasificación, los cuales, seguramente seguirán madurando y enriqueciéndose, conforme los desarrollos tecnológicos vayan proporcionando nuevas técnicas para la construcción y manejo de redes. El objetivo principal de estos criterios es proporcionar a los usuarios una base de evaluación y comprensión de las redes.

Una red de computadoras puede ser subdividida en dos componentes principales que son los sistemas de procesamiento de datos o anfitriones y la subred de comunicaciones, como se muestra en la figura 1.2. Los anfitriones son los sistemas inteligentes que se enlazan y comunican a través de la red. Estos pueden ser computadoras de propósito general; computadoras de propósito especial como: servidores de archivos, terminales, impresoras y comunicaciones; estaciones de trabajo inteligentes; etc. Mientras que la subred de comunicaciones conformada por nodos y enlaces, proporciona la infraestructura de comunicación, a través de la cual se comunican los anfitriones.

de comunicación interactivos o de correo.

Util Para Organizaciones Distribuidas

Existen muchas compañías que son de naturaleza distribuida, donde su operación está dada por varios establecimientos ubicados en diferentes localizaciones geográficas, pero los cuales requieren ser regidos por la misma organización. En estos casos las redes de computadoras encuentran ubicación de manera natural y directa.

Estos son solo algunos de los parques las redes de computadoras están adquiriendo gran importancia como una nueva alternativa, para la solución de problemas relacionados con la informática, las comunicaciones y la computación en general.

OBJETIVO DE LAS REDES

Las redes de computadoras nacieron en la búsqueda de una nueva estructura computacional que proporcionara beneficios adicionales a los de las redes tradicionales. No obstante, las redes han tenido que pasar por diversas etapas, y aun continúan madurando, y conforme vayan surgiendo mejores tecnologías y se vaya adquiriendo un mejor dominio del campo, las redes proporcionarán nuevos y mejores beneficios.

El objetivo del desarrollo de las redes de computadoras es llegar a obtener una buena integración de las máquinas y sus recursos, para así, poder llegar a proporcionar a los usuarios nuevas facilidades, así como eficientizar y mejorar otras que ya se dan en la actualidad. Además, también se espera que tanto las computadoras como la misma infraestructura de comunicaciones de la red, apoyen el desarrollo de

sistemas operativos distribuidos, los cuales, utilizando las capacidades de enlace y comunicación de la red, pueden realizar el manejo y administración de las computadoras y sus recursos, para así poder llegar a ofrecerle a los usuarios ya no un computador, sino un sistema de procesamiento (máquina virtual) de capacidades y recursos ilimitados.

CLASIFICACION DE REDES

La diversidad de redes de computadoras que se han desarrollado, ha propiciado el surgimiento de una variedad de criterios de clasificación, los cuales, seguramente seguirán madurando y enriqueciéndose, conforme los desarrollos tecnológicos vayan proporcionando nuevas técnicas para la construcción y manejo de redes. El objetivo principal de estos criterios es proporcionar a los usuarios una base de evaluación y comprensión de las redes.

Una red de computadoras puede ser subdividida en dos componentes principales que son los sistemas de procesamiento de datos o anfitriones y la subred de comunicaciones, como se muestra en la figura 1.2. Los anfitriones son los sistemas inteligentes que se enlazan y comunican a través de la red. Estos pueden ser computadoras de propósito general; computadoras de propósito especial como: servidores de archivos, terminales, impresoras y comunicaciones; estaciones de trabajo inteligentes; etc. Mientras que la subred de comunicaciones conformada por nodos y enlaces, proporciona la infraestructura de comunicación, a través de la cual se comunican los anfitriones.

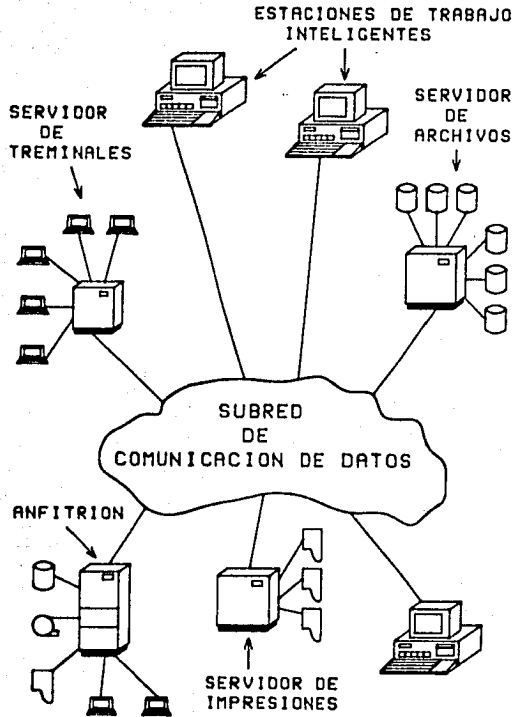


Figura 1.2
RED DE COMPUTADORAS

De las dos partes constitutivas de las redes de computadoras, la subred posee el mayor grado de desarrollo y estudio, y es básicamente en sus características y estructura donde se basan la mayoría de los criterios importantes de clasificación. Estos criterios toman en cuenta la conformación física de los nodos y enlaces (topología); la técnica de conmutación; el área de cobertura de la red; el esquema de control y acceso del canal de comunicación; el esquema de modulación y, el medio de comunicación. Los últimos tres se aplican principalmente a las LANs.

Topologías de Redes

La topología es un esquema de clasificación de redes que se basa en la asignación de un nombre que denota la organización o arreglo que guardan los nodos de la red y sus enlaces de interconexión como parte de la subred de comunicaciones. Las topologías básicas son:

ESTRELLA. En esta topología, los nodos de la red denominados nodos satélites, se enlazan a través de un nodo o computador central. El nodo central se encarga de manejar y coordinar el establecimiento de enlaces de comunicación entre los nodos satélites. Cuando un nodo satélite desea comunicarse con otro, se lo solicita al nodo central, quien, utilizando la técnica de conmutación de circuitos, establece el enlace correspondiente. Una vez establecido el enlace, los nodos satélites conectados se comunican como si existiera una conexión física dedicada (punto-a-punto) entre ellos. Ver figura 1.3a.

ANILLO. En esta topología, los nodos se conectan al canal a través de unos dispositivos denominados repetidores. Estos proporcionan las funciones de inserción, recepción y retransmisión de datos sobre el

Redes de Computadoras

canal. Los repetidores constituyen los elementos activos que, en conjunto con los segmentos de canal, conforman la trayectoria cerrada de transmisión unidireccional para la comunicación de los nodos. En la figura 1.3b se ilustra este tipo de red.

LAZO. En la topología de lazo, los nodos son interconectados utilizando enlaces dedicados (punto-a-punto) formando una trayectoria cerrada. En ésta no se comparte un mismo canal como en el caso de las redes tipo anillo. Ver figura 1.3c.

COMPLETAMENTE INTERCONECTADA. Cuando todos los nodos de la red tienen una conexión directa y no compartida con cada una de los demás nodos, la topología se clasifica como completamente interconectada o con redundancia completa. En la figura 1.3d se ilustra el caso.

BUS. En esta topología, los nodos se enlazan a través de un canal compartido utilizando derivadores pasivos, los cuales permiten la transmisión o recepción de información. En el proceso de recepción los nodos copian o leen la información cuando transita por el canal. El procedimiento de copiado se realiza sin afectar el curso normal de la información, y solo se lleva a cabo cuando el nodo es el destinatario. El canal no conforma una trayectoria cerrada como en el caso de una red tipo anillo, pero permite la propagación bidireccional de la información a partir de su punto de aplicación. Ver figura 1.3e.

ÁRBOL. En la topología de árbol o jerárquica, los nodos son interconectados en niveles conformando una estructura jerárquica. Hay quienes ven a una red de árbol como un caso general de una red tipo bus donde el árbol queda formado por un bus central y varios buses

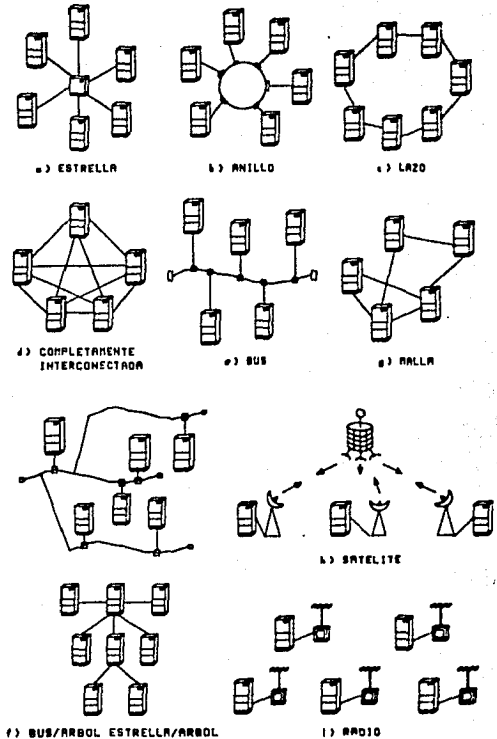


Figura 1.3
TOPOLOGIAS DE REDES DE COMPUTADORAS

ramificados de éste. Otras ven a esta topología como un caso generalizado de una red de estrella. En los casos que no es relevante la distinción entre la topología de árbol y las de bus o estrella, se puede utilizar la nomenclatura compuesta bus/árbol o estrella/árbol, según sea el caso. Ver figura 1.3f.

MALLA. Si los nodos son interconectados con enlaces directos de una manera compleja, y la topología no puede ser clasificada como alguna de las anteriores, se puede decir que la red posee una topología de malla o irregular. La figura 1.3g ilustra el caso.

SATELITE. Este tipo de red se conforma por un satélite geostacionario localizado en el espacio y los nodos de la red. Cada nodo posee una conexión a una estación terreno que le permite enviar y recibir información en las frecuencias de operación del satélite. El satélite funge como un repetidor que recibe la información enviada por los nodos desde los estaciones terrenas, en una frecuencia, y la retransmite (difunde) a todos los nodos utilizando otra frecuencia diferente. Ver figura 1.3h.

RADIO. En este tipo de red, los nodos poseen una conexión con un sistema de radio que les permite enviar y recibir información a través del aire utilizando ondas electromagnéticas. Cuando un nodo envía información, todos los demás la escuchan, pero solo lo toma el nodo que posee la dirección de la información transmitida. Ver figura 1.3i.

Redes Punto-A-Punto y Difusión

Las redes descritas por las topologías anteriores pueden ser agrupadas o clasificadas en dos grandes tipos, que son las redes

punto-a-punto y las redes de difusión.

Las redes punto-a-punto son las que conectan sus nodos utilizando enlaces directos entre ellos. En estas redes se utilizó lo técnico de transmisión de información guarda-reexpide (store and forward), en la cual, la información que envía un nodo, es recibida y almacenada por el siguiente, hasta tomarla completamente. En este momento, si la información no es para el nodo en cuestión, éste lo retransmite (reexpide), repitiéndose el mismo proceso las veces necesarias hasta alcanzar el nodo destino. En esta clasificación caen las topologías de estrella, lozo, completamente interconectada, estrella/árbol y malla.

Las redes de difusión se caracterizan por enlazar sus nodos compartiendo el mismo medio de comunicación. En este caso, la información no requiere ser almacenada y retransmitida ya que todos los nodos son contiguos. En este grupo caen las topologías de anillo, bus, bus/árbol, satélite y radio.

Técnica de Conmutación

Esta clasificación se basa en el esquema utilizado para compartir el ancho de banda (capacidad para transmitir información) del canal de comunicaciones entre todos los nodos de la red. Se tienen las siguientes técnicas:

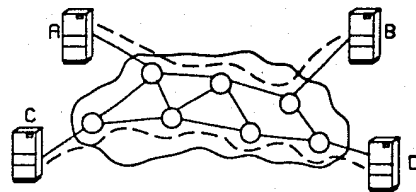
CONMUTACION DE CIRCUITOS. En este esquema, para cada conexión existe una fase de establecimiento, en la que se reservan los recursos necesarios para el enlace entre el par de nodos (a semejanza del sistema telefónico). Todos los recursos deben ser liberados al terminar la conexión y no pueden ser compartidos con otras conexiones.

Redes de Computadoras

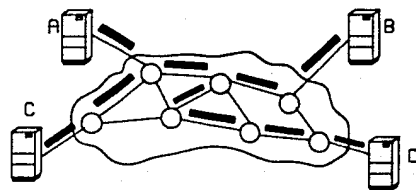
Este esquema es muy utilizado en las redes con topología de estrella y un caso típico son los conmutadores telefónicos, ya sean computarizados o no, como los CBX y PBX. En la figura 1.4a se ejemplifican dos enlaces establecidos a través de una subred de conmutación de circuitos.

La reservación de recursos implica la asignación de los nodos intermedios (o conexiones en conmutadores), canales de comunicación, y ancho de banda y memoria para buffers en los nodos intermedios, con lo cual se pueda establecer la trayectoria de enlace entre los nodos finales para así poder lograr su comunicación.

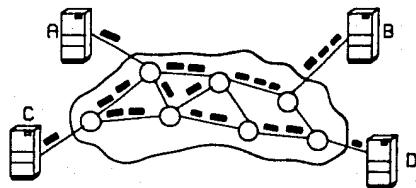
CONMUTACION DE MENSAJES. Esta técnica surgió junto con las primeras redes de computadoras, en un intento por mejorar las características proporcionadas por las redes de conmutación de circuitos. Aquí, la información es transmitida en bloques de cualquier tamaño (mensajes), sin realizar una asignación de recursos. En la transmisión de los mensajes se utiliza la dirección del destinatario como base para la selección de la ruta (como en el sistema de correos). Los problemas de esta técnica se originan al no existir un límite en el tamaño de los mensajes. Cuando se envía un mensaje muy grande se consume una gran cantidad de recursos (ancho de banda, memoria, canales, etc.) no programados (ineficiente), se requieren nodos de gran capacidad (costosos), el almacenamiento de los mensajes se realiza en disco en lugar de memoria (lento) y se limitan los tráficos interactivos por los de transferencia masiva (injusta). Esto hace que este esquema tenga un comportamiento peor que el de conmutación de circuitos, por lo que actualmente se le ve actualmente en uso. En la figura 1.4b se ejemplifica el caso de una red de conmutación de mensajes.



a) CONMUTACION DE CIRCUITOS



b) CONMUTACION DE MENSAJES



c) CONMUTACION DE PAQUETES

Figura 1.4
TECNICAS DE CONMUTACION

COMUTACION DE PAQUETES. La necesidad de corregir el esquema de conmutación de mensajes, condujo a limitar el tamaño máximo de los bloques de información, generándose la técnica de conmutación de paquetes. En esta técnica, al transmitir bloques de información más pequeños (paquetes), se proporciona un manejo más justo del ancho de banda del canal ya que los tráficos interactivos tienen las mismas posibilidades que los de grandes volúmenes. Además, pueden estar circulando por la subred varios paquetes que pertenecen al mismo o diferentes enlaces, y en general, se superan los problemas de la técnica de conmutación de mensajes. En la actualidad este es el esquema más utilizado en los diferentes tipos de topologías de redes. Ver figura 1.4C.

Cobertura de la Red

Otra clasificación de las redes de computadoras se basa en el alcance máximo entre nodos. Esta tiene implicaciones directas en aspectos técnicos como la velocidad de transmisión, el medio de comunicación, y el esquema de control y acceso al canal. En esta clasificación se encuentran las redes de área local (LAN), redes locales de alta velocidad (HSLN), redes de área extensa (WAN) e interconexión de redes, las cuales son descritas a continuación.

REDES DE AREA LOCAL (Local Area Networks). Las redes de área local (LAN) permiten la conexión de una gran cantidad de equipos en una área restringida, como un cuarto, un edificio, un conjunto de edificios, o un campo como el de una universidad, no llegando, en la actualidad, más allá de unas decenas de kilómetros (30 - 40 Km). Este tipo de

redes emplean interfaces de transmisión de precios razonables y altas velocidades que pueden llegar a los 20 Mbps.

REDES LOCALES DE ALTA VELOCIDAD (High Speed Local Networks). Otro tipo de redes locales son las de alta velocidad (HSLN), y se separan de las LANs debido a que persiguen objetivos diferentes. Mientras que una LAN es una red de propósito general que soporta una gran variedad de dispositivos y tipos de tráficos, las HSLN son un tipo especial de red diseñada para aplicaciones que requieren transferencias de datos de muy alta velocidad entre anfitriones o entre anfitriones y dispositivos de almacenamiento masivo, los cuales normalmente están limitados al cuarto del centro de cómputo. Sus características principales son [8]: velocidad de transmisión de 50 Mbps o más, distancia máxima entre nodos que va desde 0.5 a 1 Km, manejo de cables múltiples, manejo de un número muy reducido de estaciones y empleo de esquemas de acceso distribuido (tipo LAN). Este tipo de redes también son conocidas como "Back-End Networks".

REDES DE AREA EXTENSA (Wide Area Networks). Las redes de área extensa (WANs) conocidas también como redes de recorrido largo (Long Haul Networks "LHN") son redes de gran alcance, que permiten la conexión de una gran cantidad de computadoras. Su cobertura puede llegar a una ciudad o un país completo. Las velocidades de transmisión no sobrepasan las decenas de Kbps (64). Cuando estas redes se emplean como redes públicas se manejan como redes de transporte que permiten la conexión de computadoras y otras redes.

INTERCONEXION DE REDES. Este caso abarca la conexión de cualquier tipo de red, independientemente de su tecnología y características, siempre

y cuando se tengan los puentes (BRIDGES) o compuertas (GATEWAYS), necesarios para realizar las interconexiones, cambios de protocolos y acoplamiento de medios de comunicación. No hay que perder de vista que un conexionado de este tipo sólo proporciona la vía de comunicación, ya que los servicios son función del tipo de redes conectadas. Dependiendo de las redes interconectadas es posible llegar a cubrir áreas que alcancen extensiones a nivel continental o incluso al mismo planeta.

CLASIFICACION DE REDES DE AREA LOCAL

Para las LANs, los criterios de clasificación han sido mas específicos ya que, además de la topología, se consideran otras características adicionales. Estas características son el esquema de control y acceso al canal; el medio de comunicación; y el esquema de modulación, las cuales están relacionadas con la topología, y serán descritas en conjunto. Las topologías típicas de las LANs son las de bus, árbol, anillo y estrella.

Esquema de Control y Acceso

El esquema o protocolo de control y acceso define el procedimiento que, de forma distribuida o centralizada, se utiliza para compartir ordenadamente del canal de comunicaciones de una red, permitiendo que los nodos puedan comunicarse. Como la teoría de protocolos de control y acceso es bastante extensa, sólo veremos una breve descripción de los esquemas más utilizados.

REDES TIPO ANILLO. Las redes locales con esta topología han logrado un gran desarrollo debido al esfuerzo de compañías y universidades. Ejemplos de este tipo de redes son [3] el Anillo de Pierce, el Sistema

de Cómputo Distribuido (DCS) de la Universidad de California, el Anillo de MIT, el anillo de Cambridge, y mas recientemente, Token-ring de IBM. Los esquemas de control y acceso que se utilizan mas comúnmente son:

Anillo Dividido (Slotted-Ring). Este es un esquema de acceso aleatorio en el cual, el tiempo de recorrido del anillo es dividido en varios segmentos iguales, denominados ranuras. Las ranuras que dividen el tiempo del anillo circulan constantemente a su alrededor y proporcionan el espacio para la transmisión de los paquetes. Para transmitir, los nodos esperan el paso de una ranura libre, la cual es llenada con un paquete que contiene datos, direcciones e información de control. Después del llenado, el paquete comienza a circular en el anillo. Cuando un nodo ve su dirección en un paquete que circula por el anillo lo copia y enciende el bit de paquete recibido en el campo de control. Cuando el paquete da la vuelta, el nodo origen lo resume de la ranura, liberándola para la transmisión de otro paquete.

Inserción de Registro (Register Insertion). Este es un esquema de acceso aleatorio en el cual cada nodo posee dos registros de corrimiento. Uno de los registros es utilizado para cargar el paquete que se va a transmitir. Para la inserción de un paquete, los nodos deben esperar hasta detectar libre el anillo. En el momento en que un nodo detecta libre el anillo, vacía en éste el registro de corrimiento que posee el paquete. En este proceso es posible que el nodo comience a recibir un paquete que estaba circulando por el anillo. Para esto, se utiliza el segundo registro de corrimiento, en el cual se toma temporalmente el paquete, para ser devuelto inmediatamente después de terminar la inserción del primer registro. En este esquema los

registros de corrimiento son considerados como parte del anillo. Un paquete puede ser removido del anillo por el nodo que envía o el que recibe.

Acceso por Estafeta (Token-Access). Este es un esquema de acceso controlado en el cual se emplea un patrón de 'n' bits denominado la estafeta (token). La estafeta circula en el anillo pasando de nodo en nodo y es reconocida como el permiso para transmitir. Si un nodo no desea transmitir, solo deja pasar la estafeta, pero cuando sí quiere, al recibir la estafeta, modifica su patrón para indicar que ha ocupado el canal, e inmediatamente después de ésta inicia su transmisión. Cuando el nodo termina de transmitir regenera la estafeta indicando la liberación del anillo. El nodo origen es quien retira la información del anillo.

REDES TIPO BUS/ARBOL. Este tipo de topología fue la primera dentro del campo de LANs en tomar una madurez de estudio y desarrollo. En este tipo de redes, el canal proporciona un medio de transmisión de datos bidireccional que permite la comunicación de todos los nodos enlazados. Como ejemplos de este tipo de redes se tienen Ethernet de Xerox, Net/one de Ungermann-Boss, 3Com de 3Com Corporation, PC-Network de IRH, etc. Los esquemas de control y acceso utilizados son:

Esquema CSMA/CD (Múltiples Accesos por Detección de Portadora y con Detección de Colisiones). Este es un esquema de acceso aleatorio, en el cual para que un nodo pueda iniciar una transmisión es necesario que espere a detectar libre el canal de comunicaciones. Si el canal está ocupado, el nodo simplemente espera el tiempo necesario hasta que éste se libre. Cuando un nodo detecta libre el canal e inicia una

transmisión debe continuar monitoreando el canal por un período predefinido. Si en este período de monitoreo bajo transmisión no se detecta una colisión, se considera que la transmisión fue exitosa. En el caso de que varios nodos inicien una transmisión casi simultánea, por haber detectado libre el canal, incidirán en una colisión que será detectada en el período de revisión. Al detectar una colisión, los nodos primero deben detener su transmisión, luego obtener un número generado en forma aleatoria y finalmente, utilizar éste como el tiempo de espera que será utilizado antes de volver a reintentar la transmisión. Después de cada colisión los nodos duplican el tamaño de la ventana para la generación de tiempos aleatorios, con lo cual se obtiene una dispersión en el tiempo de la carga de transmisiones, y se reduce el número posible de colisiones.

Acceso por Estafeta (Token-Access). Este es un esquema de acceso controlado, utilizada también en redes con topología de bus. Su operación es similar al de la topología de anillo, con la diferencia de que aquí, la estafeta debe contener la dirección del siguiente nodo. La dirección proporciona el orden lógico de los nodos en la red, emulando un anillo. Cuando un nodo recibe la estafeta y no tiene algo que transmitir, la debe reenviar con la dirección del siguiente nodo en secuencia.

Accesos Múltiples por Multinivel (Multilevel Multiple Access). Este es un esquema de acceso controlado que proporciona un manejo ordenado del bus y evita las colisiones. También se conoce como MLMA, y su operación es como sigue. Un controlador central genera una bandera especial a intervalos regulares que indica el inicio de los marcos de

transmisión. Los morcos, son los segmentos de tiempo en los cuales todos los nodos tienen una oportunidad de transmitir, y se encuentran divididos en dos secciones. La primera sección, es la ranura de solicitud, en donde cada nodo posee un bit que le permite indicar si desea transmitir o no. Si un nodo desea transmitir lo anuncia encendiendo su bit en la ranura de solicitud. Al finalizar esta sección todos los nodos saben quienes transmitiran y en que orden, con lo cual, en la segunda sección del morco, transmiten los nodos que lo anunciaron, sin incurrir en colisiones.

REDES TIPO ESTRELLA. En este tipo de topología de red, la conexión de los nodos se realiza a través de un controlador o conmutador central, cuyo función consiste en proporcionar el control y multiplexaje de los puertos para permitir la conexión de los nodos de la red. El multiplexaje utilizado es por división de tiempo, en el cual se asigna una capacidad fija de ancho de banda para cada conexión. Esta topología es utilizada en PBX para la transmisión de datos y en CBX para transmisión de datos y voz.

Esquema de Modulación

En esta clasificación, las redes son organizadas tomando en cuenta el tipo de modulación utilizada para transferir la energía eléctrica a través del medio de comunicación. En base a esto, las redes pueden ser clasificadas en redes de banda base (base band networks) y redes de banda ancha (broad band networks).

REDES DE BANDA BASE. Este tipo de redes se caracterizan por utilizar un esquema de modulación digital para la transmisión de la información. En este esquema, cada transmisión ocupa completamente el

espectro de frecuencia del canal, por lo que para compartirlo, se utiliza el esquema de multiplexaje por división de tiempo (TDM). En este caso, la propagación de la señal sobre el canal es bidireccional, lo cual permite su aplicación en cualquier tipo de topología o medio de comunicación.

REDES DE BANDA ANCHA. Este tipo de redes utiliza un esquema de modulación analógica (radio frecuencia "RF"). En este caso, la señal solo ocupa una parte del espectro de frecuencia del canal, lo cual permite el empleo de los esquemas de multiplexaje por división de frecuencia (FDM) para compartirlo. El espectro de frecuencia del canal puede ser dividido en bandas o secciones, de tal forma que un solo canal físico puede soportar varios canales lógicos independientes que permitan la transmisión simultánea de datos, voz e imagen. Este esquema es aplicable principalmente en la topología de árbol, aunque también puede utilizarse en la de bus.

Medio de Comunicación

El medio de comunicación utilizado en las redes también guarda una relación directa con la topología de la misma, como se describe a continuación.

REDES TIPO ANILLO. Esta topología requiere enlaces punto-a-punto entre repetidores, los cuales pueden ser realizados utilizando cable de pares trenzados, cable coaxial o fibra óptica.

REDES TIPO BUS. Las redes con esta topología emplean cable de pares trenzados y cable coaxial. Las fibras ópticas en la actualidad presentan algunos problemas técnicos para su utilización.

REDES TIPO ESTRELLA. En esta topología se requieren enlaces punto-a-punto entre los nodos y el conmutador central. El cable de pares trenzados es adecuado ya que proporciona el ancho de banda necesario para manejar las velocidades de consultación a las que operan estos sistemas. Además es económica y fácil de instalar.

REDES TIPO ARBOL. Esta topología normalmente se implanta utilizando cable coaxial, y en algunos casos monores cables de pares trenzados.

Combinaciones para LANs

Como se ha visto, las redes locales pueden ser catalogadas en base a la topología, el medio de transmisión y el tipo de modulación. Las combinaciones más comunes proporcionadas en [1] son:

BUS O ANILLO CON CABLE DE PARES TRENZADOS Y BANDA BASE. Red de propósito general de bajo costo y de fácil instalación. Adecuada para una cantidad restringida de dispositivos de bajo costo y bajo rendimiento (low-throughput), en una área también restringida.

BUS O ANILLO CON CABLE COAXIAL Y BANDA BASE. Red de propósito general de costo moderado y fácil instalación. Soporta un número moderado de dispositivos en una área moderada. Adecuada para muchas de las aplicaciones de oficinas.

BUS DE ALTA VELOCIDAD CON CABLE COAXIAL Y BANDA BASE. Red de propósito especial de alto costo, que soporta un pequeño número de dispositivos de alto rendimiento (high-throughput), en una área restringida. Es adecuada para los requerimientos de una sala de cómputo.

BUS DE CABLE COAXIAL Y BANDA ANCHA. Red de propósito general flexible de costo moderado, que soporta un gran número de dispositivos sobre una área amplia. Puede manejar una variedad de tráfico utilizando multiplexaje por división en frecuencias. Adecuada para comunicación entre edificios y oficinas en una área extensa.

ESTRELLA CON CABLE DE PARES TRENZADOS. Este tipo de redes los conforman las arquitecturas de PBX y CBX. Es adecuada para soportar una gran cantidad de dispositivos de rendimiento limitado (limited throughput) y de costo moderado.

BIBLIOGRAFIA

- [1] Stallings, W.
"Local Networks", Computing Surveys, Vol.16, No. 1, marzo 1984, N.Y.
- [2] Geo, K.C.E.,
Introduction to Local Area Computer Networks, Macmillan Computer Science Series, 1983.
- [3] Strole Norman C.,
"A Local Communications Network Based on Interconnected Token-Access Rings: A Tutorial", IBM Journal of Research and Development, Vol.27, No. 5, sept. 1983, pp. 481-196 (reimpreso en Stallings, W., Local Network Technology, Second Edition, IEEE CS Press/North Holland, 1985, pags. 89-104).
- [4] Ahuja V.,
Design and Analysis of Computer Communication Networks, Mc Graw-Hill Computer Science Series, 1985.
- [5] Fritz J. S., Kaldenbach C. F., Pruger L. M.,
Local Area Network Selection Guidelines, Prentice-Hall, Englewood Cliffs, Nueva Jersey, 1985.
- [6] Tabagi F. A.,
"Multicast Protocols in Packet Communication Systems", IEEE Transactions on Communications, Vol. COM-28, No. 4, abril 1980, pags. 468-488 (reimpreso en Liebowitz B.H. y Carson J.H., Tutorial Distributed Processing, Third Edition, IEEE C.S. Press, 1980, pags. 574-594).

- [7] Freeman H.A., Thurber K.J.,
Local Network Equipment, IEEE C.S. Press, 1985.
- [8] Stallings W.,
Tutorial Local Network Technology, Second Ed., IEEE C.S. Press, 1985.
- [9] Freeman H. A., Thurber K. J.,
Tutorial Microcomputer Networks, IEEE C.S. Press, 1981.
- [10] Abrams M., Cotton I. W.,
Computer Network: A Tutorial, Fourth Edition, IEEE C.S. Press, 1984.

CAPITULO II

Arquitecturas y Protocolos de Redes

INTRODUCCION

Desde la aparición de las primeras redes experimentales de computadoras en los Estados Unidos, Canada, Europa y otros países, hace más de quince años, las redes han evolucionado de una manera sorprendente, siendo éstas, cada vez más versátiles, más poderosas, pero inevitablemente más complejas. Conforme esta evolución ha procedido, se han realizado grandes esfuerzos por sustituir los métodos iniciales de diseño de redes empleando técnicas caseras, por procedimientos sistemáticos que permitan organizarlas, comprenderlas y diseñarlas a través de metodologías más apropiadas y homogéneas.

En este capítulo se revisaran algunos de los conceptos principales del manejo de las redes y sus arquitecturas. Además, se verá una rápida descripción del modelo de referencia propuesto por ISO para la interconexión de sistemas abiertos y la arquitectura de Red-IIHMS.

PROTOSCOLOS E INTERFACES

Desde sus inicios, las redes han sido diseñadas estructurándolas jerárquicamente como una serie de capas. El número, nombres y funciones de las capas, han diferido para cada red. Sin embargo, todas coinciden en que el uso de la metodología de dividir el diseño de la red completa, en diseños para cada capa, lleva el propósito de cambiar

la solución de un problema grande e inmanejable por la solución de una serie de problemas más pequeños y manejables. En una red, las capas inferiores realizan las funciones básicas y ofrecen los servicios sobre los cuales se apoyan las capas superiores, aislándolos de los detalles de como son desarrollados sus servicios.

En una estructura jerárquica, para que dos capas del mismo nivel puedan entenderse, deben comunicarse utilizando el mismo lenguaje. Para ésto es necesario que se establezcan por mutuo acuerdo, las convenciones que definan que será comunicado, como será comunicado y cuando será comunicado. Este conjunto de convenciones es referido como el protocolo de comunicación. De esta forma, para que dos capas del mismo nivel jerárquico de una arquitectura puedan comunicarse, es necesario que ambas manejen el mismo protocolo, que defina el conjunto de reglas, formatos y significados que gobiernen el intercambio de datos entre ellos.

Entre cada par de capas adyacentes de una arquitectura existe una interfaz de comunicación. Una interfaz de comunicación define los primitivas de interacción y los servicios que la capa inferior le ofrece a la superior. En la figura 2.1 se ilustran los conceptos de protocolo e interfaz.

COMUNICACION FISICA Y COMUNICACION VIRTUAL

En una arquitectura de una red, sólo en la capa inferior existe comunicación físico (directa), la cual es lograda a través del medio físico de conexión (cable) entre los nodos. En los niveles superiores de la arquitectura, al no existir un enlace físico directo entre las capas del mismo nivel, sólo es posible manejar una comunicación

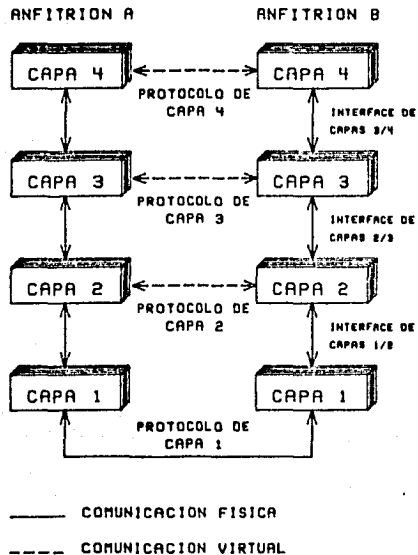


Figura 2.1
ARQUITECTURA DE CUATRO CAPAS

virtual entre ellos. En una comunicación virtual, cada capa pasa tanto datos como información de control a la capa inferior. Al alcanzar la capa de más bajo nivel, donde existe comunicación física, la información es enviada por el canal. En el lado opuesto de la arquitectura, una vez que la información es tomada por la capa de más bajo nivel, las capas la van subiendo en un procedimiento inverso al previo, hasta alcanzar la capa con el nivel jerárquico correspondiente. Como hemos visto, en una comunicación virtual, al no existir un enlace físico directo de comunicación, ésta se da a través de las capas inferiores de la arquitectura emulando una línea de comunicación virtual. En la figura 2.1 la capa 1 maneja comunicación física, mientras que las capas 2, 3 y 4 manejan comunicación virtual.

ARQUITECTURAS Y PROCESOS PARES

Una vez que un diseñador decide cuántas capas incluir en la red y cuáles deben ser las funciones que debe realizar cada una, la consideración más importante es establecer interfaces claras entre las capas. En términos generales, la arquitectura de una red se define como el conjunto de capas y protocolos que conforman a la red. Ni los detalles de implantación ni la especificación de las interfaces son parte de la arquitectura. Para implantar una arquitectura no es necesario que las interfaces en todas las máquinas de la red sean iguales.

Un último concepto que se explicará sobre redes, es el concerniente a procesos pares (peer process) y se refiere a los procesos que estando en el mismo nivel jerárquico en una arquitectura, utilizan el protocolo de la capa para su comunicación.

MODELO ISO DE REFERENCIA

Al principio los diseñadores de redes seleccionaban el número de capas y las funciones que deberían de realizar cada una de ellas, de acuerdo a su experiencia. Sin embargo, a finales de los setentas, los mismos diseñadores de redes apoyaron a la Organización Internacional de Estándares (ISO), para la formación de un subcomité que trabajara en el desarrollo de un estándar de arquitectura. El resultado obtenido por este subcomité fue la especificación de un modelo de referencia para la interconexión de sistemas abiertos (OSI), el cual fue concebido como un marco de trabajo para la definición y desarrollo de los estándares para los protocolos de las capas del modelo. El término "abierto" denota la habilidad de que sistemas heterogéneos puedan comunicarse siempre y cuando respeten el modelo de referencia y sus estándares asociados.

El modelo de referencia de ISO fue concebido como un modelo para la interconexión de redes de área extensa. No obstante, en la actualidad es exitosamente utilizado también en redes locales.

El modelo de referencia de ISO ha quedado definido como un conjunto de siete capas y las funciones que deben ser realizadas por cada una de ellas, para ofrecer un servicio determinado. Esta definición del modelo fue obtenida, generando una capa donde fuera necesario un diferente nivel de abstracción para realizar un conjunto de funciones bien definidas, seleccionando con cuidado los límites de cada una para minimizar el flujo de información a través de sus interfaces, y cuidando que el número de capas fuera lo suficientemente grande como

para separar funciones diferentes que no requirieran estar en la misma capa, pero que no fueran deseados como para que la sobrecarga impuesta por el manejo de la jerarquía se volviera impráctica o inmanejable. A continuación se da una breve descripción de las capas del modelo, comenzando por la de menor nivel jerárquico como se ilustra en la figura 2.2.

Capa Física

Esta capa de la arquitectura se encarga del manejo de los disciplinas de transmisión considerando para ello las características físicas, mecánicas y eléctricas del medio de comunicación y la descripción de su manejo y conexión. Entre estas se consideran el tipo de medio de comunicación (cable, aire, fibra óptica, etc.), la definición de conectores, niveles de señal, esquema de modulación, frecuencia de comunicación y toda lo relacionado con la forma de transmitir bits a través de un canal de comunicación.

Capa de Enlace de Datos

La función de esta capa es aprovechar el servicio de transferencia de bits de la capa física y proporcionar un servicio de comunicación de marcos, libre de errores entre nodos de la red. Para cumplir con esto, el protocolo de la capa debe manejar los esquemas de ruteo, control de flujo, y detección y recuperación de errores a nivel de nodos. En el caso de las redes locales, también es responsabilidad de la capa el manejo del esquema de acceso y control del canal de comunicación compartido.

Capa de la Red

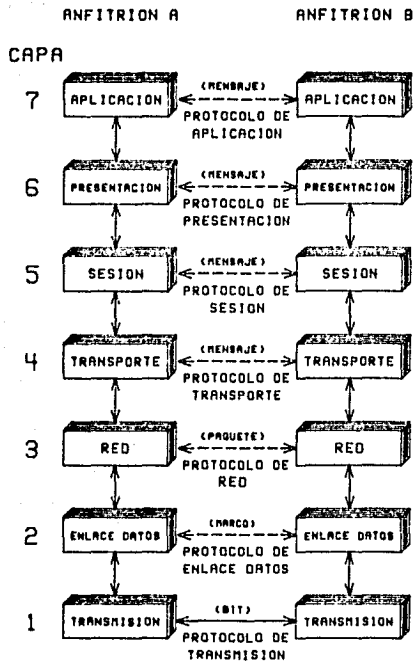


Figura 2.2
 MODELO DE REFERENCIA DE ISO

Esta capa se encuentra involucrada con la realización de las funciones que determinan el servicio final ofrecido por la subred para la comunicación de los anfitriones de la red. Para esto, la capa realiza las actividades de ruteo, control de flujo, manejo de errores, y control de congestión a nivel de la subred. Además se encarga de la contabilización del uso de los recursos para su cobro posterior y de la fragmentación y reensamble de mensajes en paquetes para su transferencia a través de la subred.

Capa de Transporte

Esta capa es responsable de proporcionar un servicio de comunicación fin-a-fin de proceso-a-proceso, que cumpla con los requerimientos de las capas superiores y procesos de aplicación. Este servicio debe ser consistente con los tres capas inferiores, de tal forma que se optimice su manejo, aprovechando sus características principales, y corrigiendo sus deficiencias. Las funciones básicas de la capa de transporte consisten en la asignación y manejo de direcciones para los procesos; establecimiento, manejo y terminación de conexiones; multiplexaje y demultiplexaje entre las direcciones de transporte y la subred; manejo de esquemas de sincronización, control de flujo y recuperación de errores fin-a-fin. Los servicios de esta capa pueden ser de datagramas si se trata del envío de mensajes independientes sin garantía de su entrega y orden, o de circuitos virtuales, si se garantiza la recuperación de mensajes perdidos, se eliminan duplicados y se entregan los mensajes en el orden enviado.

Capa de Sesión

Esta capa define la interfaz con el usuario y realiza las negociaciones necesarias para el establecimiento de sesiones con otros procesos, para que los usuarios puedan entrar en algún sistema remoto. El establecimiento de una sesión normalmente es una operación complicada. Su iniciación puede involucrar la validación de los extremos de la sesión, para verificar que poseen el derecho de enlace y poder asegurar el cobro de la cuenta correspondiente. Después de la validación del enlace, continúa la fase de negociación en la que ambos extremos acuerdan las opciones y parámetros que estarán en efecto para la sesión. Una vez establecida la sesión, también es responsabilidad de esta capa el manejo de la misma. Si la capa de transporte no ofrece un servicio confiable, o no garantiza el orden de entrega de los mensajes, la capa de sesión debe realizar estas funciones. La capa de sesión proporciona un servicio orientado a la aplicación. Esta capa también realiza la conversión de direcciones de sesión a direcciones de transporte.

En algunas redes, las capas de sesión y transporte son unidas para formar una sola capa, o simplemente la capa de sesión es eliminada completamente, cuando la capa de transporte satisface los requerimientos de comunicación del usuario.

Capa de Presentación

Esta capa de la arquitectura se encarga de realizar las conversiones de formato necesarias para que la información comunicada entre los procesos de aplicación pueda ser utilizada y comprendida, cuidando que en estas conversiones no se afecte el significado de la información. Además de esto, esta capa se encarga de realizar aquel tipo de

funciones comunes en la comunicación de procesos como pueden ser la compresión de texto, la encriptación y la conversión hacia y desde los estándares de red para terminales y archivos.

Capa de Aplicación

Esta es la capa de más alto nivel de la arquitectura. La definición de las funciones y protocolos de los procesos de aplicación están en manos de los usuarios, ya que éstos determinan los servicios que quieren obtener de la red. El tipo de protocolos que contempla esta capa son aquellos que se encuentran involucrados con el servicio final de la red, tales como: transferencia de archivos, terminales virtuales, bases de datos distribuidas, sistemas de reservaciones para líneas aéreas, entre otros. En esta capa existen problemas comunes para las diversas aplicaciones como son: lograr la transparencia de la red, la distribución del procesamiento, etc.

De las siete capas del modelo ISO, las tres superiores proporcionan las funciones que se encuentran involucradas con el soporte de los servicios proporcionados a los usuarios de la red, las tres capas inferiores se encuentran directamente involucradas con la transmisión de la información comunicada entre los anfitriones de la red, y la capa de transporte conforma el enlace entre ambos grupos, proporcionando la integridad de una comunicación fin-a-fin de proceso-a-proceso, y asegurando que el servicio acumulado por las tres capas inferiores, más ello mismo, cumplen con los requerimientos de las capas superiores.

Arquitecturas y Protocolos de Redes

Red-IIMAS ha sido planteada como una red experimental de computadoras de área local que permita la conexión de los equipos de cómputo del instituto para optimizar su manejo y compartir sus recursos. Su arquitectura fue diseñada como una simplificación operativa del módulo de referencia de ISO, contemplando que la simplificación permitiera manejar inicialmente los problemas de mayor importancia para una arquitectura y fundamentales para una configuración mínima de una red. Así fue como Red-IIMAS quedó definida en un conjunto de cuatro capas, cuidando de establecer los buses correctos para obtener un sistema con buenas características de funcionamiento, confiabilidad, servicio y rapidez de respuesta.

Las cuatro capas de Red-IIMAS son: Comunicación de Datos, Inter-Red, Transporte y Aplicaciones. En la figura 2.3 se ilustra la arquitectura de Red-IIMAS mientras que en la figura 2.4 se muestra su topología. Para mayores detalles sobre los antecedentes, objetivos y características globales de la red consultar [5] y [6]. A continuación se da la descripción de cada una de las capas de la arquitectura.

Capa de Comunicación de Datos

Esta es la capa de más bajo nivel jerárquico y su función consiste en el manejo de la disciplina de transmisión de datos a través de un canal de comunicación compartido (cable coaxial). El acceso al canal es arbitrado empleando un protocolo tipo Ethernet (CSMA/CT), con un esquema de codificación en banda base, a una velocidad de transmisión de 10 Mbps. El servicio proporcionado por esta capa consiste en la transmisión de mensajes de tamaño variable. Se utiliza un esquema de CRC-16 que permite que los mensajes entregados pasen una alta

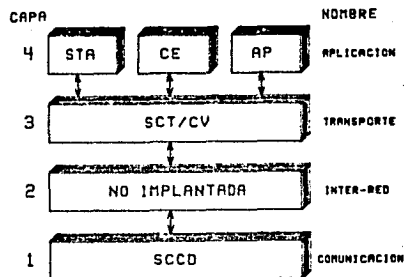


Figura 2.3
ARQUITECTURA DE Red-IIMAS

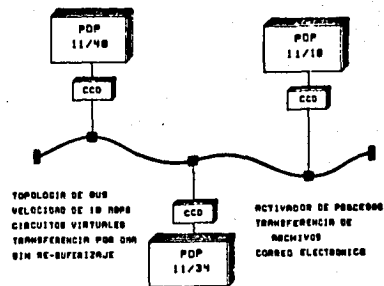


Figura 2.4
TOPOLOGIA DE Red-IIMAS

probabilidad (mayor al 99%) de ser correctos. Esta capa ha sido concebida como un controlador de comunicaciones de alta velocidad externo a los anfitriones y denominado Control de Comunicación de Datos (CCD).

Capa Inter-Red

Esta capa aún no ha sido implantada en la red, pero las funciones previstas para la misma son el ruteo, control de flujo, control de congestión, etc., involucrados en la comunicación de computadoras conectadas en diferentes redes. El servicio de esta capa es básicamente el mismo de la capa de comunicación de datos, pero extendido a nivel de redes interconectadas.

Capa de Transporte

Esta capa se encarga de controlar la transmisión de mensajes realizada por las capas inferiores (subred de comunicaciones), empleando para ello un protocolo de circuitos virtuales que le permite resolver los problemas de pérdida, duplicidad y ordenamiento de mensajes que se suscitan en la transmisión de los mismos a través de la subred de comunicaciones. Esta capa proporciona un servicio fin-a-fin de comunicación virtual libre de errores, de mensajes de tamaño variable, para los procesos de aplicación de la red. El sistema previsto para proporcionar este servicio se ha denominado Sistema de Control de Transmisión de Circuitos Virtuales (SCT/CV).

Capa de Aplicaciones

Esta es la capa superior de la arquitectura y se encuentra conformada

por los protocolos de alto nivel que proporcionan los servicios finales que la red provee a sus usuarios. Entre los servicios previstos para ser proporcionados por esta capa se tienen los de transferencia de archivos, constituida por el Sistema de Transferencia de Archivos (SIA), terminales virtuales (TV), correo electrónico (CE) y el activador de procesos o logger (AP).

BIBLIOGRAFIA

- [1] Green Paul E. JR., Miembro de IEEE,
 'An Introduction to Network Architectures and Protocols', IEEE
 Transactions on Communications, Vol. COM-28, No. 4, abril de 1980,
 pags. 413-424 (reimpreso en Abrams Marshall, Cotton Ira W., Computer
 Networks: A Tutorial, Fourth Edition, IEEE C.S. Press 1984, pags.
 163-174.
- [2] Tanenbaum Andrew S.,
 Computer Networks, Prentice Hall, 1981.
- [3] Folta Harold C., Omnicom, Inc.,
 'A Tutorial on the Open Systems Interconnection Reference Model',
 Open Systems Data Transfer, No. 1, Junio 1982, pags: 2-21 (reimpreso
 en Stallings Williams, Tutorial: Computer Communications:
 Architectures, Protocols and Standards, IEEE C.S. Press 1985, pags.
 7-26).
- [4] Tanenbaum Andrew S.,
 'Network Protocols', Computing Surveys, Vol. 13, No. 4, diciembre
 1981, pags. 453-489 (reimpreso en Stallings Williams, Tutorial:
 Computer Communications: Architectures, Protocols and Standards,
 IEEE C.S. Press 1985, pags. 27-63).
- [5] Cantó G., Hernández J., Márquez P., Nava F.,
 'Descripción de la Red Local IIMAS de alta velocidad',
 Comunicaciones Técnicas Serie Naranja Investigaciones, No 347,
 septiembre 1984, IIMAS, UNAM., México, D.F.
- [6] Cantó G., Márquez G., Nava F.,
 'RED-IIMAS', Memorias del Primera Conferencia Nacional de usuarios
 de equipo DIGITAL, noviembre 1986, México, D.F.
- [7] Kanangi V. y Dhas C. R.,
 'An Introduction to Network Architectures', IEEE Communications
 Magazine, Vol. 21, No. 7, octubre de 1983, pp. 44-50 (reimpreso
 en Stallings Williams, Tutorial: Computer Communications:
 Architectures, Protocols and Standards, IEEE C.S. Press 1985, pags.
 72-78).
- [8] Stallings Williams,
 'In Depth IEEE Project 802, Setting standards for local-area
 networks', COMPUTERWORLD, febrero 13 de 1984 (reimpreso en
 Stallings Williams, Tutorial: Computer Communications:
 Architectures, Protocols and Standards, IEEE C.S. Press 1985, pags.
 142-148).

CAPITULO III

Protocolos de Transporte

INTRODUCCION

Los protocolos de transporte, también conocidos como protocolos de control de transmisión, protocolos de control de red, entre otros, obtienen el primer nombre por referencia a la capa de su ubicación en la arquitectura de una red. Estos protocolos ocupan el cuarto nivel en la estructura jerárquica del modelo de referencia propuesto por ISO, y el tercero en la arquitectura de Red-THAS. En cualquiera de estos casos, las capas inferiores conforman la subred de comunicaciones, y la capa o capas superiores constituyen los usuarios del servicio de transporte, que serán referidos en conjunto únicamente como procesos aplicación, aunque pueden intermediar también los protocolos de las capas de sesión y presentación.

En una red, las capas de la subred de comunicaciones se responsabilizan del manejo de los mecanismos de transmisión, ruteo y control de la información, intercambiada entre nodos y anfitriones. De la misma manera, la responsabilidad de un protocolo de la capa de transporte consiste en tomar el servicio proporcionado por la subred, y realizar las funciones necesarias para ofrecer un servicio completo de comunicación fin-a-fin, que cumpla con los requerimientos de comunicación de los procesos de aplicación. En la figura 3.1 se ilustra la ubicación de la capa de transporte en un esquema de una arquitectura simplificada.

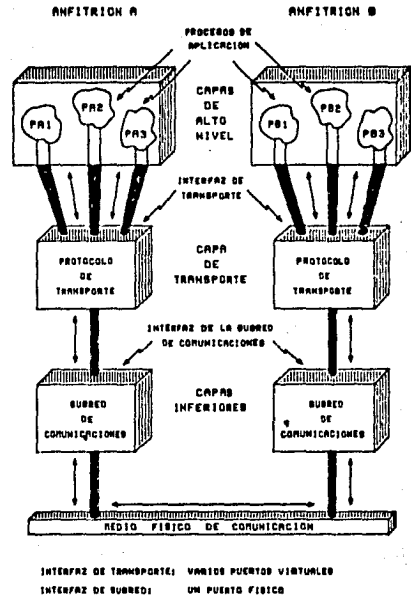


Figura 3.1
ESQUEMA DE UBICACION
DE UN PROTOCOLO DE TRANSPORTE

En el proceso de diseño de un sistema de transporte, es importante conocer primeramente cuales son los problemas que deben ser resueltos por su protocolo. Por lo tanto, antes de proceder con la fase de diseño de un protocolo de estos, es necesario saber con claridad cuales son los servicios que proporciona tanto la subred de comunicaciones, así como cuales son los servicios que el sistema de transporte debe proporcionar para la comunicación de los procesos de aplicación. En este capítulo se revisan los aspectos que deben ser considerados en el proceso de diseño de un protocolo de transporte, y se describen el tipo de funciones que deben ser consideradas para su interfaz.

SERVICIOS DE LA SUBRED

Uno de los puntos previos a la definición y diseño de un protocolo de transporte, consiste en la revisión de las características y servicios de la subred de comunicaciones, sobre las cuales se cimentará la construcción del protocolo, y que tienen una fuerte ingerencia en la definición de una gran parte de sus funciones y características.

La gran variedad de subredes de comunicación que se han desarrollado, ha propiciado que los procesos que requieran comunicarse a través de ellas, tengan que desarrollar procedimientos específicos para cada una. De la misma forma, los sistemas de transporte han tenido que evolucionar sobre esta gran diversidad de subredes, por lo cual en la actualidad se pueden encontrar grandes variaciones en el grado de servicios y sofisticación de estos [6]. Para facilitar el manejo de los subredes, éstos han sido clasificados, de acuerdo al tipo de servicio que ofrecen, en subredes de datagramas y subredes de

circuitos virtuales.

Datagramas

Una subred que ofrece un servicio de datagramas, acepta mensajes (o paquetes) de la capa de transporte, e intenta su entrega como unidades aisladas, sin manejar algún esquema de recuperación global de errores. Por lo tanto, los mensajes pueden arribar fuera de orden, pueden llegar repetidos y además pueden perderse. Cada mensaje transmitido siempre debe estar acompañada por las direcciones de origen y destino, que se utilizan para seleccionar la ruta por la que transitará. Este tipo de servicio se asemeja al servicio de correos o telégrafos.

Circuitos Virtuales

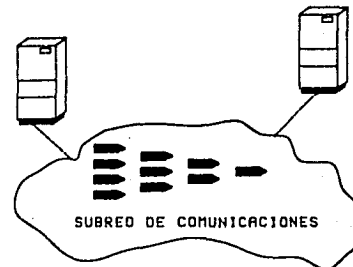
Cuando una subred proporciona un servicio de circuitos virtuales, ésta se responsabiliza de realizar una entrega íntegra y ordenada de los mensajes (o paquetes) que se envían a través de ella. Para esto, la subred debe realizar la recuperación de errores necesaria. En un esquema de circuitos virtuales siempre se requiere de una fase inicial de establecimiento de una conexión, en la cual se actualizan los tablos de ruteo en los nodos y anfitriones de la red. Las conexiones establecidas son utilizadas para realizar y controlar el intercambio ordenado de información. En este caso, los mensajes transmitidos, en lugar de llevar las direcciones de origen y destino, sólo requieren el identificador de la conexión, el cual es utilizado en conjunto con los tablos de ruteo de los nodos para la determinación del camino por el cual transitará el mensaje. Este tipo de servicio se asemeja al servicio proporcionado por el sistema telefónico.

Protocolos de Transporte

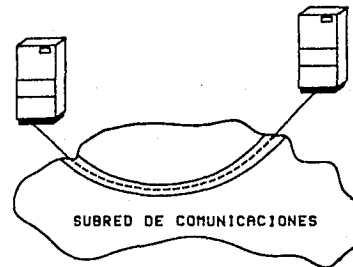
Además de poder caracterizar a una subred por el servicio que proporciona, también puede ser caracterizada de acuerdo a su estructura operativa interna. Por lo tanto, una subred que ofrece un servicio de circuitos virtuales, puede operar internamente ya sea como una subred de datagramas, o como una subred de circuitos virtuales. Mientras que en el caso de una subred que proporciona un servicio de datagramas, lo usual es que éste basada sobre una estructura operativa del mismo tipo. En la figura 3.2 se ilustran los conceptos de una subred de datagramas y una de circuitos virtuales.

Las funciones de un protocolo de transporte pueden ser triviales o muy complejas, dependiendo del servicio proporcionado por la subred de comunicaciones. Si la subred proporciona un servicio de circuitos virtuales, que garantiza una transferencia ordenada de mensajes, sin errores, sin pérdidas ni duplicaciones, entonces el desarrollo del protocolo de transporte resulta relativamente simple, ya que la subred realiza la mayor parte del trabajo. Pero, si por el contrario, el servicio proporcionado por la subred es de datagramas, el protocolo de transporte se vuelve mucho más complejo, ya que debe realizar funciones que le permitan preservar el orden de entrega de los mensajes, resolviendo los errores, pérdidas y duplicidades que se suscitan en la transmisión de los mensajes a través de la subred. En la figura 3.3 se ilustra el caso de un requerimiento de un servicio de transporte constante, con variaciones en el servicio de la subred de comunicaciones, donde, el protocolo de transporte queda como una función del servicio proporcionado por la subred.

SERVICIO DE TRANSPORTE



a) DATAGRAMAS



b) CIRCUITOS VIRTUALES

Figura 3.2
SUBREDES DE COMUNICACIONES

Protocolos de Transporte

Otra etapa previa al diseño de un protocolo de transporte consiste en obtener una definición clara de los servicios de comunicación que éste debe proporcionarlos a los procesos de aplicación. Una concepción del servicio de transporte, es la referente a la acumulación de facilidades hasta la interfaz superior del protocolo de transporte (interfaz de transporte), o sea, las facilidades de la subred más las facilidades del protocolo de transporte. En la práctica, se ve que el servicio de transporte dependen de cada red, pero en la mayoría de los casos, por lo menos, se ofrece un servicio de entrega de mensajes en secuencia y libre de errores como se especifica en la referencia [3].

En general, la idea de un servicio de transporte consiste en proporcionar una interfaz de comunicación universal, que otorgue las mismas facilidades a todos los procesos de aplicación, la cual debe ser conservada, independientemente del medio de comunicación y las técnicas de control de transmisión que se utilicen en la subred. Además, los procesos de aplicación deben poder acceder los servicios de transporte sin importar su ubicación, siempre y cuando se encuentren en alguna de las máquinas de la red, o redes interconectadas, como en el caso del TCP de ARPANET [5].

Si los servicios de la subred fueran constantes, el protocolo de transporte quedaría como una función de las facilidades requeridas para la comunicación de los procesos de aplicación, tal como se ilustra en la figura 3.4. Conjuntando los casos ejemplificados en las figuras 3.3 y 3.4, se ve que un protocolo de transporte puede definirse, en términos generales, como la diferencia entre los servicios requeridos por los procesos de aplicación y los

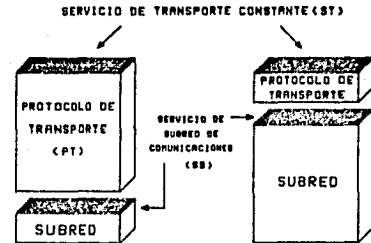


Figura 3.3
PT EN FUNCION DEL SS,
CUANDO EL ST PERMANECE CONSTANTE

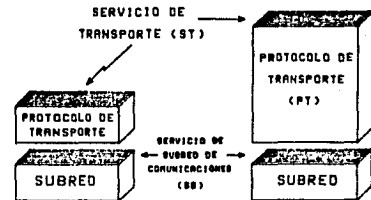


Figura 3.4
PT EN FUNCION DEL ST,
CUANDO EL SS PERMANECE CONSTANTE

Protocolos de Transporte

proporcionados por la subred [4]. En este contexto, si los servicios de la subred cambian, la capa de transporte también deberá modificarse, de tal forma que los servicios de transporte sean proporcionados empleando los nuevos servicios de la subred, con lo cual, todos los procesos que estén encima de la capa de transporte permanecerán intactos. En algunas redes se puede encontrar que los requerimientos de los procesos de aplicación pueden ser diversos, lo cual se puede solucionar, desarrollando varios protocolos de transporte. Por ejemplo, se puede desarrollar uno que proporcione un servicio de circuitos virtuales y otro que proporcione un servicio de datagramas, como se ilustra en la figura 3.5. Con ésto, los procesos de aplicación pueden elegir el más adecuado a su operación.

PROTOCOLO DE TRANSPORTE

El propósito fundamental de un sistema de transporte consiste en proporcionar un servicio de comunicación entre procesos de aplicación, que sea confiable y eficiente. Para que un sistema de transporte pueda proporcionar un servicio como éste, busándose en una subred de comunicaciones sea confiable, se requiere desarrollar un protocolo de comunicación, que forzosamente debe verse involucrado con la solución de diversos problemas. De los problemas que deben ser considerados en el diseño de un protocolo de transporte, los más importantes se encuentran relacionados con los siguientes tópicos:

- Direccionamiento de Procesos
- Transferencia Básica de Datos
- Manejo de Conexiones
- Confiabilidad y Recuperación de Errores

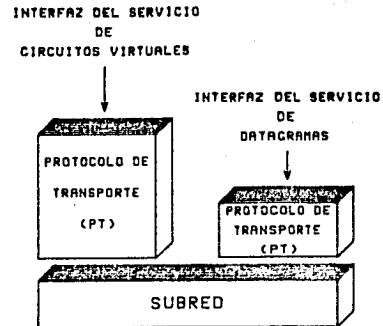


Figura 3.5
SERVICIOS DE UN
PROTOCOLO DE TRANSPORTE

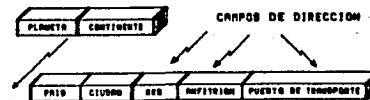


Figura 3.6
DIRECCION JERARQUICA

- Control de Flujo y Almacenamiento Intermedio
- Multiplexaje
- Sincronización
- Seguridad y Prioridad

Direccionamiento de Procesos

Para que en una red los procesos puedan comunicarse, es indispensable que exista un esquema de direccionamiento especial, el cual permita la identificación adecuada de cada uno de los procesos, proporcionándoles un identificador que lo distinga en forma única. Para esto, normalmente se utiliza un esquema de direccionamiento jerárquico, en el cual, la dirección de cada proceso se conforma como una cadena de campos que especifican de una manera general, las direcciones del país, la ciudad, la red, el anfitrión y finalmente la dirección de transporte. La dirección de transporte consiste del número de puerto que le asigna localmente a cada proceso el sistema de transporte. En la figura 3.6 se esquematiza el concepto de dirección jerárquica.

En el diseño de un protocolo de transporte se deben definir las particularidades del esquema de direccionamiento que se utilizará para la identificación de los procesos de aplicación. Para ésta, hay que especificar el tipo y número de puertos de transporte, a través de los cuales se identificarán y comunicarán los procesos; hay que seleccionar el esquema de asignación de puertos a procesos, el cual puede ser dinámico, estático o alguna combinación de ambos; y también hay que definir los mecanismos a través de los cuales los procesos pueden conocer la dirección de otros para comunicarse con ellos. La solución de este último punto puede realizarse definiendo un Activador

de Procesos (AP) o LOGGER, el cual debe residir en cada una de las máquinas de la red a nivel de la capa de aplicación. Los servicios del LOGGER deben ser solicitados en una dirección fija conocida, que debe ser respetada en todas las máquinas de la red. Cada LOGGER en coordinación con el sistema de transporte local, y a solicitud de los procesos remotos, realice la activación de procesos locales e informe el número de puerto que le fue asignado al proceso por el sistema de transporte. Con esto, los procesos pueden activar a sus servidores, y conocer su dirección dentro de la red, para así poder comunicarse con ellos y realizar las actividades que les encomienden sus usuarios.

Transferencia Básica de Datos

Para el intercambio de información entre los procesos, es necesario definir el esquema que debe manejar el sistema de transporte para tal fin. Entre los diferentes servicios de transferencia de datos que son proporcionados por los sistemas de transporte, se tienen los esquemas orientados a bytes y los orientados a registros o bloques. En los esquemas orientados a bytes, el sistema de transporte ofrece un servicio de transmisión de cadenas de bytes entre usuarios. Los bytes son empaquetados en bloques para transmitirlos a través de la subred. En este tipo de servicio, el protocolo de transporte es el responsable de decidir cuando debe terminar la construcción de un bloque y transmitirlo.

En el servicio de transferencia de datos orientado a bloques o registros, los procesos le pasan a la capa de transporte bloques de información de tamaño variable, denominados cortos o mensajes. Cuando un proceso le pasa a la capa de transporte un mensaje completo, éste

debe transmitirlo a la capa de transporte del proceso remoto, tan pronto como sea posible. Como cada mensaje completo que se transmite entre un par de procesos normalmente implica una acción específica para el proceso receptor, es importante que el servicio de transporte no imponga restricciones sobre el tamaño de los mensajes que se puedan intercambiar. Además, como los procesos receptores sólo actúan sobre la llegada de mensajes completos, el sistema de transporte no debe generar notificación alguna mientras no haya conformado un mensaje completo y correcto.

Manejo de Conexiones

Antes de que dos procesos puedan iniciar cualquier intercambio de mensajes, es necesario que primero cada uno de ellos esté consciente de la existencia del otro, y luego, cuando ya se requiera realizar un intercambio para cumplir alguna tarea, ambos procesos deben ejecutar en forma coordinada los pasos necesarios para establecer una conexión o asociación entre ellos.

Una conexión entre dos procesos consiste de un enlace lógico que les permite fijar una trayectoria de comunicación bidireccional, para el intercambio de información. Una conexión puede ser identificada en forma única relacionando el par de direcciones de los puertos de los procesos implicados, o mediante un número de enlace. Aunque entre un par de procesos pueden existir varias conexiones, una conexión sólo puede enlazar a un par de procesos, y un puerto sólo puede estar involucrado en una conexión en un instante determinado. Una conexión sólo debe perdurar el tiempo de comunicación entre los procesos involucrados en la relación.

El conjunto de la información de estado que inicializa y mantiene el sistema de transporte para cada flujo de información, tal como: la dirección del proceso local, la dirección del proceso remoto, los números de secuencia de los mensajes, los tamaños de los ventanas de secuencia, etc. es referida como una conexión [5].

Cuando dos procesos desean comunicarse, sus protocolos de transporte deben establecer una conexión (inicializar en forma coordinada la información de estado y estructuras de apoyo en cada lado). Cuando ha terminado la comunicación entre los procesos, éstos deben notificarlo al sistema de transporte, para que cancele la conexión, y los recursos sean liberados, y así puedan ser utilizados por otros usuarios.

En los casos en que las conexiones requieran ser establecidas entre anfitriones no confiables y sobre subredes de comunicación no confiables, se recomiendan mecanismos más sofisticados para el manejo de las fases de establecimiento y terminación de conexiones. En el caso del TCP de ARPANET se utilizan mecanismos con asignación de números de secuencia en base-a-relaj, para evitar inicializaciones erróneas de conexiones [5].

Confiability and Recuperación de Errores

Así como es necesario un esquema de identificación global de procesos, y mecanismos para el establecimiento de trayectorias entre ellos, para lograr la comunicación adecuada entre ellos, es igualmente importante que el intercambio de mensajes a través de las trayectorias sea logrado sin corrupción y sin pérdidas. En los eventos de falla de la subred de comunicaciones, el sistema de transporte debe proveer los

mecanismos de recuperación necesarios para su detección y corrección.

En el caso de un servicio de circuitos virtuales, el protocolo de la capa de transporte es responsable de la recuperación de los datos que se dañan, pierden, duplican o arriban a su destino fuera de orden, debido a circunstancias adversas que normalmente se presentan a lo largo de la trayectoria entre el transmisor y el receptor en la subred de comunicaciones.

El primer paso para lograr la recuperación de los errores está en su detección. Asignando números de secuencia a los mensajes, es posible detectar y corregir al momento de su recepción, las pérdidas de secuencia y duplicación. La pérdida completa de los mensajes es detectada utilizando esquemas de reconocimiento positivo sobre la recepción de mensajes correctos en la estación destino. Si el reconocimiento de arribo de un mensaje (ACK) no es recibido por la estación transmisora dentro de un intervalo de espera predeterminado (TIMEOUT), ésta debe retransmitirlo. La retransmisión de mensajes se realiza por un número máximo de intentos, después de los cuales, si no se recibe el ACK, se cancela la operación y se señala con el número de error correspondiente. Utilizando códigos para la detección de errores, es posible detectar daños en los mensajes transmitidos. Este código debe ser calculado y cargado en los mensajes antes de su transmisión, y luego recalculado en el receptor para su comparación. Si la comparación es correcta el mensaje es aceptado, pero si no, es desechado, en cuyo caso, eventualmente será retransmitido esperando que entonces llegue bien. El protocolo de recepción puede regresar un ACK después de recibir un mensaje correcto, o después de entregarlo al proceso de aplicación, dependiendo de los aspectos de diseño del

protocolo de transporte.

Control de Flujo y Almacenamiento Intermedio

Para realizar sus funciones, un sistema de transporte normalmente requiere cierta cantidad de espacio en almacenamiento intermedio (buffers) que le permita manejar los mensajes en tránsito de las diferentes conexiones establecidas por los procesos de aplicación. Aunque a la fecha los costos de memoria han bajado mucho, todavía resulta ser un recurso limitado en la mayoría de los computadores, por lo que el sistema de transporte posiblemente no pueda obtener toda la memoria necesaria para su operación. Para resolver este problema, los protocolos de transporte utilizan esquemas de control de flujo que permiten que las partes receptoras puedan administrar la cantidad y frecuencia de datos enviadas por las partes transmisoras del protocolo. En muchos de los casos, estos esquemas consisten en ventanas móviles de créditos, cuyos valores son regresados en cada ACK, e indican el margen que tienen los transmisores para seguir enviando datos, de acuerdo al espacio de memoria disponible y su prioridad. En protocolos como X.25, donde se manejan tamaños de ventana fijos, los ACKs y los permisos resultan ser sinónimos [1]. En los esquemas orientados a bytes, la ventana indica el número de bytes que pueden ser enviados por el transmisor antes del siguiente permiso. Mientras que en los esquemas orientados a registros, la ventana indica la cantidad de espacio en almacenamiento intermedio que el transmisor puede consumir, o el número de mensajes que puede enviar antes de otro nuevo permiso. En resumen, un esquema de control de flujo permite el control de un transmisor rápido para que no sobrepase las capacidades

de almacenamiento de un receptor lento.

Multiplexaje

Para que varios procesos en un mismo anfitrión puedan compartir las facilidades de comunicación del sistema de transporte, éste proporciona un conjunto de direcciones lógicas o puertos de comunicación, que se utilizan como los nombres simbólicos o identificadores de los procesos. Como los servicios de la capa de transporte son proporcionados a través de sus puertos, y éstos deben ser multiplexados sobre un conjunto usualmente menor de canales físicos de la subred de comunicaciones, el sistema de transporte debe inicializar y mantener la información necesaria para realizar el multiplexaje y demultiplexaje de mensajes que se requiere. Para resolver esto, el sistema de transporte registra en sus tablas y estructuras las relaciones que guardan los puertos de transporte y direcciones de anfitriones que están manejando los procesos para su comunicación. Cuando un mensaje es transmitido, su encabezado debe contener los parámetros de direccionamiento necesarios para que, en combinación con las tablas y estructuras del sistema de transporte, se pueda realizar adecuadamente la entrega de los mensajes entre los procesos en comunicación.

Sincronización

Debido a que en la mayoría de las redes de conmutación de paquetes, los retardos de transmisión de paquetes pueden ser muy variables, e incluso los paquetes pueden llegar fuera de orden, los números de secuencia que utilizan los protocolos de transporte para la transmisión de los mensajes, permiten restablecer el orden de la

información, pero no así el tiempo de transmisión relativo entre ellos. Algunas aplicaciones con restricciones de tiempo real, tales como la transmisión de voz, gráficos, teleconferencias, o combinaciones de éstos en forma simultánea, requieren que los datos sean entregados con la misma periodicidad con que fueron transmitidos. Esta situación impone que los mensajes de transporte lleven en su encabezado información adicional, denominada figura de tiempo, la cual debe indicar el momento en el que éstos fueron producidos o emitidos. Los relojes en el transmisor y el receptor deben de correr a la misma velocidad aunque no es necesario que posean el mismo tiempo, ya que el objetivo es el poder reproducir el espaciamiento de tiempo correcto entre los mensajes. El retardo de entrega del primer mensaje determina el retardo de todos los siguientes. Si algún mensaje no llega antes del retardo límite, en la mayoría de los casos es desechado ya que ha perdido su validez dentro del contexto general de la información [1].

Seguridad y Prioridad

En algunos sistemas de transporte se permite que los usuarios puedan indicar el grado de seguridad y prioridad que desean en su comunicación. Dependiendo de la red estos servicios pueden estar apoyados por la subred de comunicaciones o no. Si estos valores no son manejados por alguna solicitud explícita del usuario, cuando no son necesarias, los protocolos de transporte deben tomar previsión para sus valores por omisión.

PRIMITIVAS DE UNA INTERFAZ DE TRANSPORTE

En un ambiente de red, los procesos utilizan protocolos para

coordinarse y realizar tareas para sus usuarios. Los procesos que desarrollan estos protocolos, acceden las facilidades de comunicación previstas por el servicio de transporte a través de su interfaz.

Otro factor importante del diseño de un sistema de transporte, consiste en la definición de la interfaz a través de la cual proporciona el servicio de transporte de datos a los procesos de aplicación. Para esto, es necesario considerar los requerimientos funcionales de la comunicación entre procesos, que permita definir el conjunto de primitivos o llamadas que, a semejanza de los previstos por un sistema operativo para la manipulación de archivos, permitan el acceso a los servicios del sistema de transporte. El conjunto de primitivos definidos así para un sistema de transporte, conforman la interfaz de transporte del protocolo.

Los primitivos de la interfaz de transporte pueden ser organizados de acuerdo al tipo de función que realizan, como por ejemplo, primitivos para el manejo de puertos, manejo de conexiones, transferencia de datos, verificación, cancelación y servicio de datagramas. A continuación se da una descripción general de estas primitivas.

Manejo de Puertos

En este conjunto de primitivos se encuentran aquellos que permiten la obtención y liberación de los puertos de transporte que son utilizados en la identificación y comunicación de los procesos de aplicación.

SOLICITAR_PUERTO. Todo sistema de transporte debe proveer algún tipo de primitivo que permita que los procesos de aplicación puedan obtener puertos, para realizar su comunicación con otros procesos. Dependiendo

de cada sistema, este primitivo debe contener el número de puerto local solicitado, o bien, la dirección de memoria donde el sistema de transporte registrará el número de puerto asignado. Otro parámetro posible es el tipo de puerto solicitado, si es que se manejan varios. En la ejecución de esta primitiva el sistema de transporte debe dar de alta el puerto asignado, y establecer e inicializar los estructuras de datos correspondientes para su manejo.

LIBERAR_PUERTO. Otra primitiva de este grupo es la que permite liberar un puerto de transporte. Esta primitiva es utilizada por los procesos de aplicación, cuando ya no requieren un puerto que tienen asignado. El parámetro que normalmente se especifica es el número del puerto local que se desea liberar.

Manejo de Conexiones

Cuando un sistema de transporte ofrece un servicio de transporte de datos basado en un protocolo de circuitos virtuales, es indispensable que los procesos de aplicación puedan obtener enlaces o conexiones para realizar su comunicación. Entre este tipo de primitivas se encuentran aquellas que permiten establecer, controlar y terminar conexiones lógicas, para que los procesos de aplicación puedan manejar trayectorias confiables en su intercambio de información.

CONECTAR. Esta primitiva la emplean los procesos de aplicación para solicitar el establecimiento de una conexión con otro proceso. Los parámetros que normalmente se especifican son: el puerto local asignado al proceso que emite la solicitud, la dirección del puerto

asignada al proceso remoto con quien se quiere establecer la conexión, y el tipo de servicio que se desea para el enlace. Con ésta, el sistema de transporte, intenta abrir una trayectoria confiable entre los dos puertos de transporte especificados. En la ejecución de este primitivo se establece una trayectoria local de enlace, entre el proceso y el sistema de transporte. Los procesos que se identifican ante el sistema de transporte mediante esta primitiva, deben de especificar un puerto remoto que este asignado a un proceso en alguna estación de la red, ya que de otra forma, fallará la operación de conexión.

ESPERAR_CONEXION. Utilizando esta primitiva, los procesos locales de una máquina le informan al sistema de transporte su disponibilidad de aceptar llamadas. En esta primitiva normalmente sólo se especifica como parámetro el número del puerto que designa el proceso para la espera del enlace. En la ejecución de la primitiva se crea una trayectoria local entre el proceso y el sistema de transporte a través del puerto especificado, y se establece el estado que indique que el proceso está listo para aceptar una conexión solicitada por algún proceso ya sea local o remoto, y que contenga como destino la dirección del puerto especificado como parámetro.

ACEPTAR, RECHAZAR. Cuando un proceso emite un **ESPERAR_CONEXION**, ha quedado disponible en la red a conexiones solicitadas por otros procesos. Cuando se recibe una solicitud de conexión, el sistema de transporte después de validarlo, le avisa al proceso que corresponde, proporcionándole los datos que definen de donde viene la llamada. Con ésta, el proceso puede tomar una decisión y responder con una

primitiva de **ACEPTAR**, si acepta la llamada, o de **RECHAZAR**, en caso de que no la acepta. En cualquiera de los casos, la respuesta debe ser regresada al proceso que emitió la solicitud de conexión, a través de su interfaz con el sistema de transporte. Las funciones realizadas por estas primitivas, también pueden ser indicadas en los parámetros de la primitiva **ESPERAR_CONEXION**, de tal forma de que cuando el sistema de transporte recibe una solicitud de conexión sabe si lo puede aceptar o no, notificando únicamente al proceso cuando ha recibido y aceptado una conexión.

Una conexión establecida entre un par de procesos, puede ser terminada ya sea porque ha concluido la tarea para la cual se abrió el enlace, o debido a algún problema no recuperable. Para tales efectos es posible utilizar una de las primitivas de la interfaz de transporte diseñadas para ésto, como son la de **DESCONECTAR** y la de **DESTRUIR**.

DESCONECTAR. Esta primitiva la emplean los procesos de aplicación para terminar una conexión de transporte de una manera controlada. Para su cumplimiento es necesario que la primitiva sea emitida en ambos extremos de la conexión. El parámetro requerido consiste en el identificador de la conexión, o el número de puerto local asignado al proceso que posee la conexión, según sea el caso. Antes de que el sistema de transporte confirme la operación, debe asegurarse de entregar todos los mensajes pendientes. Después que un proceso ha emitido esta primitiva, ya no puede realizar envíos adicionales.

DESTRUIR. Al contrario del caso anterior, esta primitiva permite que un proceso pueda terminar una conexión en forma unilateral, sin tener

la confirmación del extremo contrario de la conexión. Esta primitiva realiza el cierre de una conexión en forma no controlada, y normalmente se utiliza cuando surge algún problema irreparable. En este caso es factible que se pierdan mensajes como resultado de la ejecución de la operación.

REINICIAR. En un evento de pérdida de sincronía entre un par de procesos de aplicación enlazados, es posible que no puedan continuar con un enlace adecuado. Utilizando la primitiva de **REINICIAR**, se fuerza a que la conexión regrese a sus condiciones iniciales de operación, desechando cualquier mensaje que permaneciese en la asociación. Después de haber emitido esta primitiva, la asociación puede verse como ntra completamente nueva, con la ventaja de haberse evitado una nueva fase de establecimiento para la misma.

Transferencia de Datos

Una vez que se relacionan un **CONECTAR** con un **ESPERAR_CONEXION**, y la asociación se confirma con un **ACEPTAR**, los procesos enlazados pueden realizar el intercambio de mensajes necesario para cumplir con sus actividades. En este intercambio los procesos pueden hacer uso de las siguientes primitivas.

ENVIAR. Utilizando esta primitiva, los procesos de aplicación le indican al sistema de transporte su deseo de enviar información. Los parámetros manejados en este tipo de primitivas son el identificador de la conexión por la que se debe transmitir, la dirección del buffer donde se localiza el mensaje o el mensaje mismo, y el tamaño del

mensaje, si se trata de un esquema de registros. Si es un esquema orientado a bytes, en lugar de la dirección y tamaño del mensaje, se pasa el byte a enviar.

RECIBIR. Esta primitiva lo emplean los procesos para informarle al sistema de transporte que están listos para recibir mensajes (o cadenas de bytes). Los parámetros utilizados son el identificador de la conexión a través de la cual se quiere recibir, la dirección del buffer a ser utilizado en la recepción, y en el caso del modo registro, una variable para obtener el tamaño del mensaje recibido. Cuando un mensaje completo (o byte) es depositado en el buffer, el sistema de transporte se lo notifica al proceso, dando por terminada la operación.

INTERRUPCION y ATENDERINT. Para una gran mayoría de los procesos de aplicación, resulta necesario el envío y recepción de mensajes en forma urgente. Esto puede lograrse manejando un parámetro adicional en las primitivas de **ENVIAR** y **RECIBIR**, que indique la prioridad requerida, o bien, mediante el manejo de un par de primitivas adicionales, que se manejarían como interrupciones y serían las de **INTERRUPCION** y **ATENDERINT**. En este caso, la primitiva de **INTERRUPCION** debe operar igual que la de **ENVIAR**, con la diferencia de que la información se transmite en forma urgente, sin pasar por los mecanismos normales de control de flujo o secuenciamento. Para la recepción de los mensajes transmitidos en forma urgente, los procesos receptores deben asignar recursos empleando la primitiva de **ATENDERINT** que equivale a la primitiva de **RECIBIR**.

Verificación y Cancelación

Las primitivas utilizadas para verificar el estado de un puerto o una conexión, son de gran utilidad para los procesos en las fases de manejo de conexiones y transferencia de datos. Los primitivos de cancelación son importantes ya que permiten un control más directo de la operación de algunas de las primitivas anteriores, y evitan que un proceso pueda quedarse en una espera infinita, cuando exista un problema en la operación de la subred o el anfitrión remoto.

ESTADO. Este primitivo lo utilizan los procesos de aplicación para obtener información concerniente con el estado de un puerto o una conexión. Los parámetros que se manejan son el identificador de la conexión o puerto y la dirección de un buffer en el cual se regresa la información solicitada.

CANCELAR. Cuando un proceso ha emitido uno o varios primitivos de **ESPERAR_CONEXION**, y desea eliminar alguna de ellos, puede emitir la primitiva de **CANCELAR**.

En los casos de los primitivos de **ENVIAR** y **RECIBIR**, los procesos también pueden cancelarlas emitiendo la primitiva de **CANCELAR** correspondiente. En el caso de la primitiva para cancelar un **ENVIAR**, sólo se atenderá si no se ha transmitido el mensaje totalmente, un cuyo caso en lugar de terminar el envío del mensaje se envía una orden para que el sistema de transporte remoto tire la parte de mensaje recibido. En el caso de la cancelación de un **RECIBIR**, si la primitiva es emitida antes de que un mensaje completo sea pasado al proceso, siempre será atendida y resultará en la pérdida del mensaje.

En los casos de **INTERRUPCION** y **ATENDERINT** pueden existir primitivos de cancelación que operarán de manera similar a las primitivas anteriores. Los primitivos de cancelación permiten liberar los recursos asignados para el manejo de los primitivos del sistema de transporte, dejándolos disponibles para que otros usuarios los puedan utilizar.

Servicio de Datagramas

En un gran número de aplicaciones el concepto de conexión es manejado por los procesos que se comunican. Esto es valioso en los casos donde sólo se requiere manejar una cantidad pequeña de mensajes para realizar una tarea, por lo que resulta costoso en recursos el tener que establecer y destruir una conexión a través de la subred de comunicaciones. De acuerdo con el concepto de servicio datagramas para una interfaz de transporte, podemos tener los siguientes primitivos:

ENVIAR_DATAGRAMA. Con esta primitiva un proceso puede enviar un sólo mensaje. Los parámetros que se especifican en este tipo de primitivos son la dirección completa del proceso destino, la dirección completa del proceso local, la dirección del buffer que posee el mensaje a transmitir y el tamaño del mensaje. Este primitivo puede ser considerado en forma lógica como la secuencia de primitivos de **CONECTAR + ENVIAR + DESCONECTAR** del servicio de circuitos virtuales de la interfaz de transporte.

RECIBIR_DATAGRAMA. Con este primitivo un proceso informa al servicio

de transporte que se encuentre listo para aceptar mensajes a través de un puerto en especial. Los parámetros especificados aquí son la dirección del puerto local, la dirección del buffer donde se espera recibir el mensaje, y el tamaño del buffer de recepción. Esta primitiva equivale a un ESPERAR_CONEXION + ACEPTA + RECIBIR + DESCONECTAR del servicio de circuitos virtuales.

BIBLIOGRAFIA

- [1] Sunshine C. A.,
 'Transport Protocols for Computer Networks', Protocol and Techniques for Data Communications Networks, F.Kuo (Editor), 1981, págs. 35-77.
- [2] Stallings W.,
 'A Primer: Understanding Transport Protocols', Data Communications, noviembre 1984, págs. 201-215.
- [3] Tanenbaum Andrew S.,
 Computer Networks, Prentice Hall, 1981.
- [4] Sunshine C. A.,
 'Interprocess Communication Protocols for Computer Networks', Tesis de Ph.D., Universidad de Stanford, 1975.
- [5] Postel, J. (ed.),
 'DOD Standard Transmission Control Protocol', Defense Advanced Research Projects Agency, Information Processing Techniques Office, RFC 761, IEN 129, enero 1980.
- [6] Davies D., Barber D., Price W. y Solomonides C.,
 Computer Networks and Their Protocols, John Wiley & Sons, mayo 1981.

CAPITULO IV

Protocolo del SCT/CV

INTRODUCCION

El Sistema de Control de Transmisión de Circuitos Virtuales (SCT/CV) constituye el sistema que desarrolla el protocolo de la capa de transporte de la arquitectura de Red-IIMAS. El SCT/CV tiene el propósito de cumplir con el servicio de comunicación requerido por los procesos de la capa de aplicaciones.

Un protocolo adecuado para el SCT/CV, debe aprovechar al máximo las ventajas y características importantes de los capas de la subred de comunicaciones, solucionar problemas y conflictos no resueltos por los protocolos de las capas de la subred y proporcionar un servicio adecuado para la comunicación de los procesos de aplicación. De tal forma que el SCT/CV pueda proporcionar un servicio de comunicación fin-a-fin, confiable y eficiente que permita el enlace y comunicación de los procesos de aplicación, aislandolos de los detalles del manejo de las capas inferiores y de las modificaciones que en ellos lleguen a surgir.

En este capítulo se describen los aspectos más importantes del diseño del protocolo del SCT/CV, y las restricciones contempladas en su realización en las computadoras PDP 11/34, bajo el sistema operativo RSX-11M.

SUBRED DE COMUNICACIONES DE Red-IIMAS

Una red de computadoras se constituye de dos partes principales que son, la subred de comunicaciones y los anfitriones. Mientras la subred conformada por nodos y enlaces, se avoca al transporte de datos entre anfitriones, estos últimos constituyen los sistemas inteligentes que se comunican utilizando los servicios de la subred.

En Red-IIMAS la subred está constituida por los capas de comunicación de datos e inter-red, las cuales se encuentran ubicadas fuera de los anfitriones, en un sistema de comunicaciones de alta velocidad diseñado especialmente para la red y denominado Control de Comunicación de Datos (CCD). Las capas de transporte y aplicaciones, ubicadas dentro de los anfitriones, constituyen las capas superiores de la arquitectura de la red y se conforman por los programas que implantan los protocolos de comunicación confiable fin-a-fin y alto nivel. La capa de transporte es la primera dentro de los anfitriones, por lo que debe manejar el enlace entre la subred y el anfitrión, utilizando una interfaz de entrada/salida en la computadora.

Aunque la arquitectura de Red-IIMAS contemple dos capas para la subred de comunicaciones, su plan de desarrollo prevee inicialmente la solución de los problemas de comunicación concernientes a una sola red, y posteriormente, considerando su interconexión hacia otras redes. Por lo tanto, las capas de comunicación de datos, transporte y aplicaciones están planteadas para la misma fase, mientras que la capa inter-red lo está para una fase posterior.

Como el protocolo del SCT/CV está en función directa de los servicios proporcionados por la subred, y en la fase actual del proyecto, quedarán a cargo de la capa de comunicación de datos, entonces, sobre

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

Protocolo del SCT/CV

Éstos se basó el diseño del protocolo del SCT/CV. No obstante, se han tomado las previsiones necesarias para la inclusión a futura de la capa inter-red. En las siguientes secciones se revisan los servicios de la subred, provistos inicialmente por la capa de comunicación de datos, y sus expectativas con la inclusión de la capa inter-red.

Servicios de la Capa de Comunicación de Datos

La capa de comunicación de datos es la de más bajo nivel jerárquico de la arquitectura de Red-IMAS. Esta capa se encuentra avocada al manejo de la disciplina de transmisión de datos a través de un canal de comunicación compartido. El canal es proporcionado por un enlace físico de cable coaxial y es administrado utilizando un protocolo tipo ethernet (CSMA/CD), codificación en banda-base y una velocidad de transmisión de 10 Mbps. Esta capa se ha desarrollado en un dispositivo de alta velocidad externo a la computadora, utilizado para el enlace y comunicación de los anfitriones. Este dispositivo está conformado por el Control de Comunicación de Datos (CCD) y el Trans-Receptor (TR). El CCD maneja el protocolo de comunicación de la capa y define el manejo de la interfaz de enlace con la capa de transporte en el anfitrión. El TR se encarga de proporcionar el aislamiento eléctrico con el cable y el manejo de las señales eléctricas enviadas y recibidas por el mismo. Para mayor información sobre esta capa consultar las referencias [1], [2] y [3].

El servicio proporcionado por la capa de comunicación de datos consiste en el envío de mensajes de información de tamaño variable, entre anfitriones de la red. Este servicio es proporcionado a una alta velocidad por lo que, en condiciones de baja carga en el canal de

Protocolo del SCT/CV

comunicación, el retardo de transmisión de la información puede ser considerado como despreciable. Aunque el servicio de comunicación no es completamente confiable y pueden perderse algunos mensajes, el manejo de códigos de verificación redundante (CRC-16) permite que los mensajes entregados posean una muy alta probabilidad de ser correctos. Además, podemos considerar que la posibilidad de pérdida de mensajes, pérdida de secuencia, o almacenamiento prolongado de mensajes en algún nodo de la subred, son relativamente bajas debido al esquema y topología de la red.

Primitivas de Comunicación de Datos

El servicio proporcionado por la capa de comunicación de datos es obtenido a través de un conjunto de cuatro primitivas básicas que son:

ENVIA (Estación-Destino,Puerto-Destino,Dir-Msj,Tam-Msj)
RECIBE (Puerto-Local,Dir-Buffer,Tam-msj)
ELIMINA (Puerto-Local,Tam-msj)
INFORMA (Puerto-Local,Tam-msj)

Con éstas se permite que la capa superior pueda realizar las actividades necesarias para el enlace de sus protocolos de comunicación a través de la red de alta velocidad. En el apéndice A, se da una descripción de estas primitivas.

Extensión de los Servicios por la Capa Inter-Red

La capa inter-red se tiene considerado para una fase posterior de Red-IMAS. Su función como parte de la subred consiste en extender el servicio de la capa de comunicación de datos a nivel de redes

interconectadas, desarrollando los protocolos adecuados para resolver los conflictos que atañen a su interconexión. Los objetivos de esta capa prevén la interconexión inicial de redes locales con arquitectura de Red-IMAS, y posteriormente conexiones con otro tipo de redes tales como X.25, SNA, etc.

Esto contempla la posibilidad de interconexión de redes utilizando líneas de alta y baja velocidad, que permitan el enlace de redes cercanas o distantes. La capa inter-red debe responsabilizarse del manejo del esquema de direccionamiento, agregando un campo para la dirección de red, que le permita preservar los servicios de la capa de comunicación de datos y extenderlos a nivel de redes interconectadas. Para esto, tendrá que resolver los problemas de ruteo, control de flujo y conversión de formatos, necesarios para permitir que la capa de transporte pueda seguir utilizando los mismos primitivos de acceso de la capa de comunicación de datos, con sólo tener que agregar el manejo del parámetro adicional de dirección de red.

Se prevé que la capa inter-red, como parte de la subred de comunicaciones, deberá estar situada fuera de los nodos. Por esto, será desarrollado como un conjunto de programas residentes en los CCDs, y los circuitos electrónicos necesarios para proporcionar el enlace específico para la conexión entre redes. En el caso de la interconexión de redes a través de líneas de baja velocidad, incluirá posiblemente el manejo de las mudas correspondientes.

Primitivos de Inter-Red

Esta capa maneja los mismos primitivos que la capa de comunicación de datos sólo que adiciona el parámetro de red destino. Los cuatro nuevos

primitivos son:

ENVIAR (Red-Destino, Estación-Destino, Puerto-Destino, Dir-Msj, Tom-Msj)
 RECIBIR (Puerto-Local, Dir-Buffer, Tom-Msj)
 FINALIZAR (Puerto-Local, Tom-Msj)
 INFORMAR (Puerto-Local, Tom-Msj)

Con estos primitivos, la capa superior puede realizar las actividades necesarias para el enlace de sus protocolos de comunicación a través de las diferentes redes de alta velocidad que se encuentran interconectadas.

De lo anterior se concluye que el protocolo del SCT/CV puede basarse en los servicios de la capa de comunicación de datos sin problema alguno, ya que la inclusión de la capa inter-red sólo requerirá el manejo adicional del parámetro de dirección de red. No obstante, es importante que las estructuras y formatos establecidos en el diseño del SCT/CV contemplen los detalles necesarios para que las adaptaciones requeridas al adicionar la capa inter-red, sean lo más sencillas posibles.

SERVICIOS DEL SCT/CV

Una capa de transporte debe proporcionar un servicio fin-a-fin de comunicación para los procesos de aplicación de la red. En el caso del SCT/CV, se debe proporcionar un servicio fin-a-fin de comunicación virtual libre de errores, que sea confiable y eficiente, y que permita el manejo de mensajes de tamaño variable. Con esto, los procesos de aplicación quedarán liberados de la ardua tarea de manejo de la comunicación con sus procesos homólogos. Para que el SCT/CV pueda

cumplir con las características del servicio especificado, debe desarrollar los protocolos necesarios para ampliar los servicios de la subred, resolviendo los problemas que se susciten en la comunicación de datos y aislando a los procesos de aplicación de la ocurrencia de los errores. Además, esto debe ser realizado en una forma eficiente para poder aprovechar al máximo las cualidades y ventajas del diseño de las capas de la subred. El SCT/CV, también debe proporcionar un filtro para los cambios y modificaciones que se suscitan en la subred.

PROTOCOLO DEL SCT/CV

Para que el SCT/CV pueda cumplir con las características del servicio planteado para la comunicación de los procesos de aplicación, basándose en una subred constituida por la capa de comunicación de datos, debe manejar un protocolo que controle adecuadamente la transmisión de mensajes a través de la subred. Para esto se ha seleccionado un protocolo de circuitos virtuales que permita resolver los conflictos de pérdida, duplicidad y ordenamiento de mensajes. El protocolo para el SCT/CV se ha caracterizado en base a una solución particular de los problemas analizados en el capítulo anterior para el diseño de un protocolo de transporte, tomando como base el interés de proporcionar un servicio de comunicación confiable y eficiente, bajo un esquema de circuitos virtuales.

Direccionamiento de Procesos

Para la identificación de los procesos de la capa de aplicaciones, el SCT/CV proporciona un conjunto de 256 direcciones o puertos de transporte. De éstos, el puerto cero es de asignación fija o estática

Y se encuentra reservado para el Activador de Procesos (AP) o LOGGER. Este puerto se maneja como un domicilio conocido, donde los procesos de aplicación pueden solicitar los servicios del AP. Cada anfitrión de la red debe desarrollar su propio AP a nivel de la capa de aplicaciones. El AP proporciona el servicio de activación de procesos en la máquina en donde reside. Cuando el AP activa un proceso, le informa al proceso que solicitó su activación, el número de puerto que le asignó el SCT/CV al ser activado. Los puertos del 1 al 255 son de asignación dinámica y pueden otorgarse a cualquier proceso de aplicación. La asignación de éstos la realiza el SCT/CV en base a las solicitudes de los procesos. Ver figura 4.1.

La operación combinada de asignación de puertos dinámicos del SCT/CV y los servicios del AP, eliminan la necesidad de tener que manejar más puertos fijos o estáticos, obteniéndose un esquema bastante flexible. Al no predefinir servicios (p.ej., transferencia de archivos) o puertos fijos, los servicios proporcionados por los procesos de aplicación pueden ser accedidos en cualquier puerto, permitiendo que varios usuarios puedan trabajar con la misma aplicación en forma simultánea. Además, el manejo de puertos fijos puede ocasionar que existiendo puertos disponibles, un proceso no pueda ser activado porque no exista alguno que se le pueda asignar. En el esquema seleccionado para el SCT/CV, los procesos pueden obtener uno o más puertos que los identifiquen, pueden activar a sus homólogos o servidores en la misma u otra máquina, y además pueden conocer que puerto poseen los procesos que activan, para así poder comunicarse con ellos.

A diferencia de los puertos físicos de la capa de comunicación de datos, los puertos de transporte son identificadores simbólicos

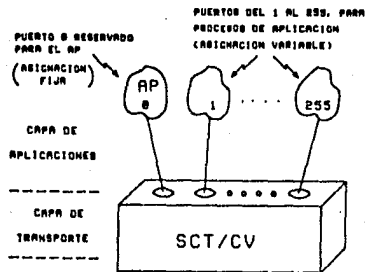


Figura 4.1

DIRECCIONAMIENTO ENTRE PROCESOS

PUERTO DESTINO	ESTACION DESTINO	M.ETDS
PUERTO ORIGEN	ESTACION ORIGEN	M.PTDS
RED ORIGEN	RED DESTINO	M.ETDR
NO. DE MENSAJE	TIPO MSG.	M.RODS
VENT. CONT. FLUJO	PRIORIDAD	M.RODR
TAMAÑO DEL TEXTO EN BYTES		M.ETRX
CCC DEL MENSAJE		M.BCCR
		M.ETRR
TEXTO DEL MENSAJE		

Figura 4.2

FORMATO DE UN MENSAJE DEL SCT/CV

(puertos virtuales) que se asignan a los procesos de aplicación. Para el manejo de los parámetros de cada puerto de transporte, es necesario definir una estructura que debe ser ligada al proceso al que se le asigna el puerto. Esta estructura se ha denominado Bloque Descriptor del Puerto (BDP) y se describe más adelante.

Transferencia Básica de Datos

El SCT/CV maneja un esquema de transferencia de datos orientado a bloques o registros de tamaño variable, que permite el intercambio de mensajes entre los procesos de aplicación de acuerdo a sus necesidades. Los mensajes de transporte se conforman de dos partes principales que son el encabezado y el texto. El encabezado de catorce bytes, está formado por varios campos de información y control necesarios para la operación del protocolo del SCT/CV. El texto, cuando no es un campo vacío, consiste básicamente de una cadena de "n" bytes que conforman un mensaje completo de un proceso de aplicación. En la figura 4.2 se muestra la estructura de un mensaje de transporte, y la descripción de cada uno de sus campos se da a continuación:

M.ETDS Campo de un byte, utilizado para el manejo del parámetro de dirección de la estación destino, hacia donde va dirigido el mensaje.

M.PTDS Campo de un byte, utilizada para el manejo del parámetro de dirección del puerto destino, hacia donde va dirigido el mensaje. Este parámetro, en conjunto con el anterior conforman la dirección de red del proceso destino de una conexión establecida.

M.ETDR Campo de un byte, utilizado para el manejo del parámetro de

dirección de la estación origen, de donde se emite el mensaje.

M.PTOR Campo de un byte, utilizado para el manejo del parámetro de dirección del puerto origen, de donde se emite el mensaje. Este parámetro, en conjunto con el anterior conforman la dirección de red del proceso origen de una conexión establecida.

M.RDDB Campo de un byte, reservado para el manejo de la dirección de red destino, hacia donde va dirigido el mensaje.

M.RDOR Campo de un byte, reservado para el manejo de la dirección de red origen, de donde se emite el mensaje.

M.TIPO Campo de un byte, utilizado para describir el tipo de mensaje de transporte. Los tipos que se pueden manejar son ilustrados en la tabla 4.1.

ABREVIATURA	VALOR (BIT)	DESCRIPCION
MSJ	128 (8)	Mensaje de datos
ACK	64 (7)	Mensaje de reconocimiento
RST	32 (6)	Mensaje de reinicialización
CON	16 (5)	Mensaje de conexión
RSP	8 (4)	Mensaje de respuesta
<hr/>		
R.ACK	1 (1)	Conexión o reinicio aceptado
R.NAK	2 (2)	Conexión o reinicio no aceptado
R.NEC	3 (2,1)	Puerta no espera conexión
R.OCP	4 (3)	Puerta ocupada

TABLA 4.1

Los mensajes de los tipos dados por los bits del 4 al 8 representan

operaciones mutuamente exclusivas y no pueden presentarse en forma simultánea. En el caso del tipo representado por el bit 4 (RSP), el mensaje representa una respuesta a una solicitud de establecimiento de conexión (CON) o de reinicialización de una conexión (RST). En este único caso, tienen validez los bits del 1 al 3, y describen el tipo de respuesta, de acuerdo a los valores proporcionados.

M.NUMM Campo de un byte, utilizado para manejar el número de secuencia de los mensajes.

M.MODO Campo de un byte, utilizado para representar la prioridad de envío de un mensaje. Este campo sólo es válido cuando está encendido el bit 8 (MSJ) de M.TIPO, con lo cual se describe el envío de un mensaje de datos en modo normal (M.MODO = 1), o en modo urgente (M.MODO = 2).

M.CRED Campo de un byte, utilizado por el mecanismo de control de flujo del protocolo. Este campo es válido cuando M.TIPO tiene encendido el bit 8 (MSJ) o el bit 7 (ACK), e indica el número de buffers que están disponibles en el lado del receptor, o sea, el número máximo de mensajes que el transmisor puede enviar antes de un nuevo permiso.

M.TMTX Campo de una palabra (2 bytes), utilizado para registrar el tamaño del texto del mensaje en bytes, cuando se trata de un mensaje de datos (M.TIPO = 128 *MSG*). En los casos en que M.TIPO presente otro valor, este campo no es válido, pero por consistencia debe ser llenado con el valor de cero.

M.BCCM Campo de una palabra (2 bytes), utilizado para cargar el valor

del BCC del mensaje. El BCC es calculado sobre los parámetros del encabezado con los valores correctos, con el campo del BCC en ceros y con todos los datos del mensaje. El cálculo del BCC consiste del OR EXCLUSIVO de todos los bytes y es explicada más adelante.

M.TEXT Campo de 'n' bytes, que constituye un mensaje de la capa de aplicaciones y conforma el texto de un mensaje de transporte. Este campo sólo es válido cuando el campo M.TIPO tiene el valor 128, o sea que se trata de un mensaje de datos. El tamaño máximo de este campo depende de las características de las capas inferiores, de tal forma que, el texto máximo de un mensaje de transporte es igual al tamaño máximo del texto de un mensaje de comunicación de datos (o inter-red) menos el encabezado transporte, que de acuerdo a las condiciones actuales ha quedado limitado a un máximo de 1000 bytes.

La restricción anterior se basa en uno de los objetivos de Red-IIMAS, el cual plantea el establecimiento de una arquitectura que permita la conexión de máquinas heterogéneas, y como muchas máquinas no poseen suficiente memoria, el diseño del SCT/CV ha restringido el manejo de mensajes a un tamaño máximo dado por las características de las capas inferiores. Evitándose la necesidad de tener que partir y reensamblar mensajes, que requiere memoria adicional. En base a ésta, el SCT/CV proporciona un servicio de transferencia de mensajes de tamaño variable, con un máximo limitado a 1000 bytes.

Manejo de Conexiones

El SCT/CV proporciona su servicio de comunicación virtual libre de errores, el cual requiere del establecimiento de conexiones entre los procesos que se comunican. Una conexión entre dos procesos, debe ser

establecida antes de que éstos puedan realizar cualquier intercambio de información. Las conexiones establecidas a través del SCT/CV, requieren del manejo de información de estado, lo cual debe ser manejada en una estructura de datos denominado Bloque Descriptor de Puerto (BDP). Todo puerto activo debe poseer su propio BDP que lo identifique, y le permita registrar cualquier operación que implique algún cambio de estado para él. La identificación de las conexiones se realiza mediante la asociación de los direcciones de los procesos de aplicación involucrados, y debe ser registrada en los BDP de los puertos que proporcionen los dos extremos de la conexión. Una vez que es establecida una conexión y son inicializados todos los variables de control, los procesos pueden comunicarse en forma confiable y eficiente. El manejo de las conexiones por parte del SCT/CV opera en base a tres tipos de transacciones que son: establecimiento, reinicialización y terminación, las cuales, se describen a continuación.

ESTABLECIMIENTO. Para activar una conexión, los SCT/CVs involucrados deben realizar las validaciones y operaciones requeridos en base a los solicitudes emitidos por los procesos de aplicación. Para ésto, el SCT/CV debe proporcionar a los procesos dos primitivos básicos, los cuales en forma sincronizada permitirán el trámite y la espera de una conexión. Para concretar un enlace entre un par de procesos, primeramente uno de ellos debe emitir una primitivo de espera por una conexión, con lo cual el SCT/CV correspondiente establece los parámetros y estado que permiten recibir la conexión solicitada. Después de ésto, el otro proceso debe emitir una primitivo que le permita tramitar una conexión, especificando en ésta los parámetros

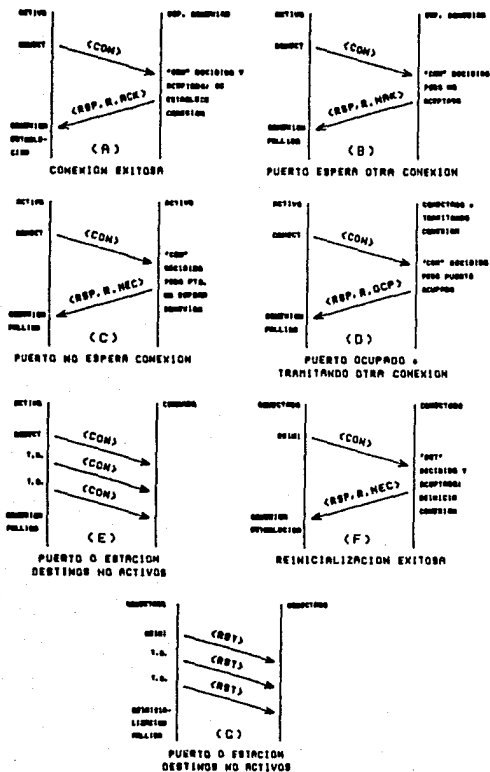


Figura 4.3

DIAGRAMAS DE TRANSICION DEL MANEJO DE CONEXIONES

del puerto involucrado en la espera. En la atención de la primitiva de tramitar una conexión, pueden existir varios casos de acuerdo al estado o condición del puerto que se desea enlazar. En la figura 4.3 se ilustran los diagramas de transición correspondientes a los cinco casos posibles de una operación de establecer una conexión. De los cinco casos, el (A) representa la única posibilidad de conexión exitosa. En los casos del (B) al (E), se representan los cuatro casos generales de las condiciones que conducen a una falla en el establecimiento de una conexión. En la figura 4.4 se muestra el diagrama de estado correspondiente.

REINICIALIZACION. Una conexión del SCT/CV normalmente permite una comunicación confiable y eficiente entre procesos de aplicación. En condiciones adversas del comportamiento de la subred podría llegar a ocurrir alguna circunstancia especial que provocara que el SCT/CV perdiera el control de alguna conexión, impidiendo que los procesos que la utilizan continuaran con un servicio adecuado. Para resolver este problema, el protocolo del SCT/CV provee mecanismos que permiten el restablecimiento de conexiones, conduciéndolas a un estado inicial, adecuado para un buen funcionamiento. En esta operación es posible que se pierdan algunos mensajes, que deben ser recuperados por los procesos de aplicación en base a los parámetros que el SCT/CV les regresa en sus operaciones. Los diagramas de transición y de estado de la operación de restablecimiento de una conexión, también son ilustrados en las figuras 4.3 (incisos F y G) y 4.4.

TERMINACION. Bajo condiciones normales de operación, los procesos de aplicación cuentan con un enlace adecuado y confiable de comunicación.

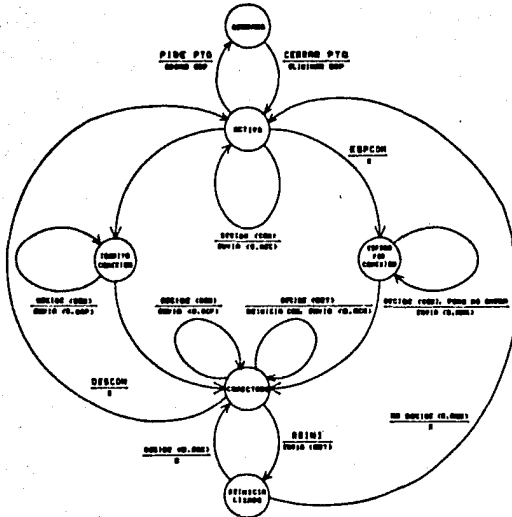


Figura 4.4
DIAGRAMA DE ESTADO DEL MANEJO DE CONEXIONES

proporcionada por el SCT/CV. Un enlace de este tipo, permite que los procesos puedan realizar la clausura de la conexión que los enlaza, en forma correcta. Por lo tanto, cuando los procesos de aplicación deciden terminar un enlace, lo único que tienen que hacer es acordarlo entre sí y entonces emitir cada uno su propia solicitud de clausura. Tomando como base este criterio, no se ha previsto un protocolo especial de cancelación de conexiones para el SCT/CV, por lo que su única responsabilidad cuando recibe una solicitud de cancelación, consiste en clausurar el lado correspondiente, presuponiendo que el extremo contrario hará lo propio. En este esquema se deja la responsabilidad a los procesos de aplicación. No obstante, es conveniente que en la implantación del SCT/CV se tomen las medidas de protección pertinentes, para evitar que los puertos se queden colgados cuando los procesos de aplicación terminen anormalmente o simplemente sin liberarlos.

Confiabilidad y Recuperación de Errores

El SCT/CV es responsable del manejo de los mecanismos que le permitan resolver los problemas suscitados en la comunicación de datos, para así poder proporcionar un servicio de comunicación virtual libre de errores. La base de corrección de los errores que ocurren en la subred se encuentra en su detección adecuado. Para esto, el diseño del SCT/CV ha tomado las siguientes consideraciones:

Para detectar la pérdida de mensajes en la subred de comunicaciones se requiere un esquema de reconocimiento, el cual está principalmente en función del tiempo de respuesta de la subred. Como el tiempo de retraso de la subred de Red-IINAS es muy bajo, se ha seleccionado un

esquema de reconocimiento positivo con ventana deslizante de un bit, conocido también como protocolo de paro-y-espera (stop-and-wait), con retransmisión por cumplimiento del tiempo de espera previsto para recibir el reconocimiento. El manejo de esquemas con retransmisión por cumplimiento de un tiempo de espera acarrea la generación de mensajes duplicados, lo cual debe ser considerado y resuelto.

Para aceptar un mensaje entregado por la subred, es necesario saber si no sufrió alguna alteración. Como la capa de comunicación de datos utiliza un código CRC-16, que le permite obtener una alta confiabilidad en la detección de errores, para el SCT/CV se ha seleccionado un código de verificación sencillo que permite detectar los errores que puedan ocurrir en las transferencias entre la subred y el anfitrión. Este esquema maneja un código de verificación de paridad vertical, el cual es obtenido mediante el cálculo de la función OR EXCLUSIVO de todos los bytes del mensaje. El código debe ser calculado en la transmisión de los mensajes, y recalculado en su recepción. El receptor compara ambos valores y sólo acepta los mensajes en los que concuerda la comparación. La operación del cálculo del parámetro empleado para la verificación de errores es lo siguiente:

(+)	Byte 1		
	Byte 2		
<hr/>			
(+)	Res 2	Byte i	Desde $i = 1$ hasta n , son los n bytes del mensaje.
	Byte 3		
<hr/>			
	Res 3	Res n	es el resultado del OR-EXCLUSIVO de los n bytes del mensaje.
	.		
	.		
	.		

Res n

El valor Res n debe ser cargado en el campo M.BCHM del encabezado de cada mensaje, para utilizarlo en el proceso de comparación en su recepción, contra el valor recalculado.

Para la detección de pérdidas de secuencia o duplicados de mensajes se utilizan números de secuencia, que son cargados y transferidos en los mensajes. Para el SCT/CV se ha reservado un campo de un byte en el encabezado de los mensajes, denominada M.NUMN. El valor de la ventana para los números de secuencia depende tanto de las características de almacenamiento, ancho de banda y tiempo de respuesta de la subred, así como del esquema de reconocimiento del sistema de transporte. En el caso de Red-TRMAS, el almacenamiento de la subred es nulo, el ancho de banda es grande y el tiempo de retraso de la subred es despreciable, por lo tanto, los característicos de la subred permiten el empleo de una ventana mínima de un bit, mientras que el esquema de reconocimiento del SCT/CV, también permite el mismo valor. Por lo tanto, los números de secuencia pueden estar en el rango de 0 a n , donde n debe ser un número mayor o igual a uno. Aun así, a mayor n se espera un mayor margen de seguridad. En el diseño del SCT/CV se debe permitir que el valor de n pueda ser sintonizado en el rango de 1 a 256, con un valor por omisión de 7, para un buen margen de seguridad. Al manejar valores de secuencia del 0 al 7, se obtiene un buen margen de seguridad que se espera que siga siendo válida cuando se incorpore la capa inter-red.

Para eficientizar el tiempo de respuesta del SCT/CV, los

reconocimientos generados por la recepción de un mensaje correcto, deben ser enviados al momento en que el mensaje es validado por el sistema de transporte y no cuando se le pasan al proceso de aplicación receptor.

Control de Flujo y Almacenamiento Intermedio

Normalmente un sistema de transporte requiere para su operación una cierta área de memoria en la máquina para almacenar temporalmente los mensajes en tránsito y poder realizar las validaciones, retransmisiones, etc., necesarias para cumplir con las características de su servicio. En el caso del SCT/CV, se debe manejar un conjunto de buffers de transmisión y recepción que permitan el almacenamiento temporal de los mensajes en tránsito. Esta función debe realizarse manejando eficientemente la memoria de la máquina, ya que por lo general es un recurso limitado y se quiere obtener una versión eficiente para el SCT/CV.

Aunque el manejo de "buffers" debe ser caracterizado para cada máquina, se debe utilizar un esquema de control de flujo que limite el envío de mensajes cuando no existe espacio para recibirlos. El esquema de control de flujo seleccionado para el SCT/CV consiste de un mecanismo de créditos, en el cual, el receptor le informa al transmisor la cantidad de "buffers" que posee para la recepción de mensajes. En este mecanismo, en la variable M.CRED de cada ACK se registra el número de "buffers" que posee el receptor para recibir mensajes. Una vez que en un ACK se ha regresado un valor de cero en M.CRED, informando que el receptor no posee espacio para recibir más mensajes, y el transmisor ha dejado de enviar, para reactivar el

envío se utiliza el mismo ACK, pero con el valor de M.CRED actualizado. Los mensajes de datos también permiten la reactivación de envíos y la actualización de la ventana de control de flujo, ya que estos también deben contener un valor de M.CRED actualizado.

Multiplexaje

Para poder identificar los procesos de la red, éstos deben solicitar y obtener uno o más puertos de transporte, los cuales deben ser multiplexados y demultiplexados por el SCT/CV a través de un solo canal de comunicación proporcionado por la subred. Para el manejo adecuado de sus puertos, el SCT/CV debe manejar una estructura de datos denominada Bloque Descriptor de Puerto (BDP). Un BDP consiste de un conjunto de parámetros importantes que permiten organizar el manejo de cada puerto, y conocer su estado de operación en cualquier instante que se requiera. En el SCT/CV debe existir un BDP por cada puerto asignado dentro de la red. Entre la información contenida en un BDP se encuentran los siguientes grupos de datos:

- a) Parámetros para la identificación del puerto, conexión y estado de operación.
- b) Parámetros que permitan el manejo de las ventanas de secuencia, los mecanismos de control de flujo, la cancelación de duplicados y el manejo de tiempos de espera para retransmisiones.
- c) Parámetros que permitan el manejo de los buffers para el envío y recepción de mensajes.

La definición concreta del BDP depende de cada implantación, y debe

Protocolo del SCT/CV

tomar en cuenta que además de cumplir con las características del protocolo, permita un buen desempeño dentro de la máquina. En la figura 4.5 se ilustra un esquema de la conformación de un BDP para el SCT/CV de Red-IIMAS.

Sincronización

En el caso de Red-IIMAS, a la fecha no se tiene previsto el manejo de aplicaciones en tiempo real que requieran de algún esquema de sincronización en el SCT/CV, que permita la reproducción de información exactamente con la misma periodicidad con que fue generada. Por tanto, en esta versión del SCT/CV no se contempla manejo alguno con respecto a este punto, aunque no se descarta para una versión futura donde los procesos de aplicación requieran esta facilidad.

Seguridad

En los casos en que la subred de comunicaciones proporcione algún mecanismo de seguridad, el sistema de transporte es responsable de ponerlo al alcance de los procesos de aplicación a través del manejo de parámetros en sus primitivos o mediante el empleo de primitivos especiales. En el caso de Red-IIMAS, la subred no maneja códigos para la seguridad en la transmisión de información, por lo tanto, el SCT/CV no realiza actividad alguna al respecto. Es importante aclarar que de acuerdo al modelo ISO, cuando la subred no provee algún mecanismo de seguridad, esto puede ser resuelto a nivel de la capa de presentación, pero no es responsabilidad de la capa de transporte. En el caso de Red-IIMAS se ha respetado este criterio.

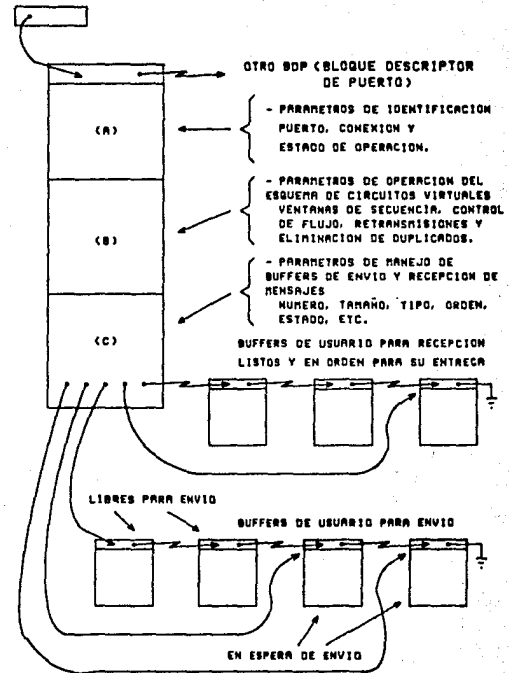


Figura 4.5
DIAGRAMA GENERAL DEL
BLOQUE DESCRIPTOR DE PUERTO DEL SCT/CV

Prioridad

Para el manejo de prioridades en la transferencia de mensajes, el protocolo del SCT/CV contempla dos modos de operación que son el normal y el urgente. En el modo normal los mensajes son transferidos ordenadamente siguiendo todos los mecanismos establecidos para el flujo de datos de los enlaces, tal como se explicó en los puntos previos. En el modo urgente los mensajes son enviados y entregados tan pronto como las actividades del SCT/CV y la subred de comunicaciones lo permitan, sin tener que pasar por las vías normales. La única diferencia de estos modos de operación radica en la urgencia de entrega. Aunque un mensaje urgente puede ser de cualquier tamaño como los mensajes normales, se recomienda que sean mensajes de atención pequeños.

Para el manejo de los modos de operación se ha reservado un campo en el encabezado de los mensajes de transporte denominado M.MODO. Este campo puede ser cargado con los valores de 1 y 2, indicando la prioridad normal o urgente que respectivamente pueden tener los mensajes en su manejo por el SCT/CV.

RESTRICCIONES POR PARTE DE LA PDP BAJO RSX-11M

En la sección anterior se vieron los detalles y características importantes del protocolo para el SCT/CV, los cuales deben ser respetados en todas las implantaciones de las máquinas que deseen pertenecer a la red, de tal forma que cumpliéndose con las especificaciones de la arquitectura de Red-IIMAS y el protocolo del SCT/CV, se pueden obtener los servicios proporcionados por Red-IIMAS.

Para la implantación del SCT/CV en cada máquina de la red, es necesario definir algunos detalles adicionales, que sin poner a la implantación del SCT/CV fuera del protocolo especificado, contemplen las características y restricciones importantes de la máquina y permitan una buena implantación del sistema.

Las observaciones de los detalles importantes para la implantación del SCT/CV en las computadoras PDP de la familia DIGITAL, operando bajo el sistema operativo RSX-11M, se especifican a continuación:

Sistema Operativo RSX-11M

Uno de los grandes atractivos de las máquinas PDP, consiste en su sistema operativo multiusuario RSX-11M, el cual proporciona un ambiente de trabajo agradable, ya que cuenta con buenas herramientas y documentación necesaria para realizar actividades íntimamente ligadas con la operación del mismo sistema. Esto es importante en el diseño de un sistema de transporte de una red, ya que si no se posee buena documentación del sistema operativo, el trabajo de implantación se complica bastante y difícilmente podría obtenerse una buena realización. En el caso de RSX-11M se poseen algunos listados del sistema operativo, que en adición a los manuales de la misma proporcionaron una buena ayuda en el desarrollo e implantación del SCT/CV en las PDP.

Memoria

Un problema serio de las PDP es que son máquinas de pocos recursos y en particular, la memoria no es la excepción. La PDP 11/34 posee una capacidad total de memoria de 256 Kbytes, los cuales son utilizados

para la ejecución del sistema operativo, así como todas las tareas de los usuarios. En particular, en el caso de cada tarea se tiene un máximo de 64 Kbytes, para el código, estructuras, etc. Esto hace ver la importancia de que en la implantación del SCT/CV en estas máquinas se optimice el uso de la memoria. Si esto no se realiza adecuadamente, se corre el riesgo de que el SCT/CV sea el único proceso que pueda correr en la máquina, limitando a los procesos de aplicación de sus servicios por no poder entrar a la máquina.

DR11-B

En la interfaz con la capa de comunicación de datos, el CCI utilize un esquema de DMA (Acceso Directo a Memoria), que debe ser caracterizado por cada anfitrión en particular, de acuerdo a sus facilidades de entrada/salida. En el caso de la PDP se ha seleccionado la interfaz DR11-B, que provee un mecanismo de comunicación con el exterior a través de un esquema de transferencias por DMA. Esta interfaz proporciona una manera rápida y eficiente de leer o escribir información directamente sobre la memoria física de la máquina. Utilizando el DR11-B se ha dado la definición de la interfaz con la capa de comunicación de datos, y es responsabilidad de la implantación del SCT/CV en la PDP, el atenderla y manejarla adecuadamente para lograr un buen funcionamiento de la red en la máquina.

Aunque el protocolo para el SCT/CV especifica el manejo de 256 puertos de transporte, una simplificación adoptada en el manejo del DR11-B, en esta fase del proyecto, limita únicamente a 8 puertos, donde el AP conserve el 0 y del 1 al 7 son para los procesos de aplicación. Esta simplificación, no es restrictiva ya que los PDPs son computadoras

pequeñas (minis) que se saturan con mucho menos de 8 procesos simultáneos, por lo tanto es un número adecuado para su capacidad, y más que una limitación se vuelve una protección operativa para el sistema.

Lenguaje Ensamblador

Las PDP son máquinas que permiten el manejo de diversos tipos de lenguajes de programación de alto nivel, pero debido al tipo de operaciones que debe realizar el SCT/CV en conjunto con el sistema operativo y a las restricciones de espacio y eficiencia planteadas para su implantación, el sistema debe ser desarrollado utilizando el lenguaje ensamblador de la máquina.

Protocolo del SCT/CU

BIBLIOGRAFIA

- [1] Contó D., Hernández J., Márquez P., Nova F.,
"Descripción de la Red Local IIMAS de Alta Velocidad",
Comunicaciones Técnicas, Serie Naranja Investigaciones. No. 367,
septiembre 1984., IIMAS - UNAM, México, D.F.
- [2] Márquez P., Contó D., Nova F.,
"Proyecto RED-IIMAS", Comunicaciones Técnicas, Serie Amatilla # 59,
Memorias del Primer Congreso Interno del IIMAS, marzo 1984, IIMAS -
UNAM, México, D.F.
- [3] Contó D., Márquez P., Nova F.,
"Red-IIMAS", Memorias de la Primera Conferencia Nacional de Usuarios
de Equipo Digital, DECUS-MEXICO, noviembre 1984, págs. 4-18, DGSCA-
UNAM, México, D.F.

CAPITULO V

Implantación del SCT/CV Bajo RSX-11M

INTRODUCCION

El protocolo del Sistema de Control de Transmisión de Circuitos Virtuales (SCT/CV), define el conjunto de formatos y procedimientos, que deben ser respetados por cada implantación de la capa de transporte de la arquitectura de Red-IIMAS. Mediante éste, los procesos de aplicación pueden obtener el servicio fin-a-fin de comunicación virtual libre de errores que se ofrece en su diseño.

En el capítulo anterior se revisaron las características del protocolo del SCT/CV, y algunas restricciones impuestas en su desarrollo por las computadoras PDP, las cuales definieron el ambiente de trabajo para la implantación del sistema en ellas.

En este capítulo se revisan los detalles importantes de la implantación del SCT/CV en las computadoras PDP de la familia DIGITAL, operando bajo el sistema operativo RSX-11M. Se ven las alternativas involucradas en el proceso de selección de la estructura de implantación del sistema, se proporciona la descripción de las estructuras de datos utilizadas, se explican los esquemas de tiempos de espera (time-outs) empleados, se describe el esquema de cancelación de puertos utilizado cuando las tareas terminan anormalmente y otros detalles más de lo puesto en marcha del sistema. La intención es proporcionar un panorama general del procedimiento seguida y los

aspectos importantes contemplados en la implantación del SCT/CV, por que sirva como base de nuevas implantaciones del SCT/CV en otras máquinas.

ESTRUCTURA OPERATIVA

El primer paso del proceso de implantación del SCT/CV consistió en determinar su estructura operativa, la cual se ilustra en la figura 5.1. La estructura operativa del SCT/CV proporciona la organización lógica modular del sistema, de acuerdo a las funciones que debe realizar. Los módulos estructurales son: la Interfaz de Comunicación de Datos, el Sistema de Circuitos Virtuales y la Interfaz de Usuario. Cada uno de estos módulos realiza un conjunto de funciones bien definidas y los tres en conjunto describen el comportamiento global del sistema.

Interfaz de Comunicación de Datos

Este es el módulo inferior de la estructura operativa y está constituido por la parte del SCT/CV que se encarga del manejo de los primitivos que definen la interfaz de la capa de comunicación de datos. Es decir, realiza la programación y manejo de la interfaz de DMA de la computadora (DR11-B), el multiplexaje de mensajes entre las capas de comunicación de datos y aplicaciones, y algunas funciones de control del CCD.

Sistema de Circuitos Virtuales

Este módulo realiza las funciones relacionadas con el manejo del protocolo de control de transmisión de circuitos virtuales que permite

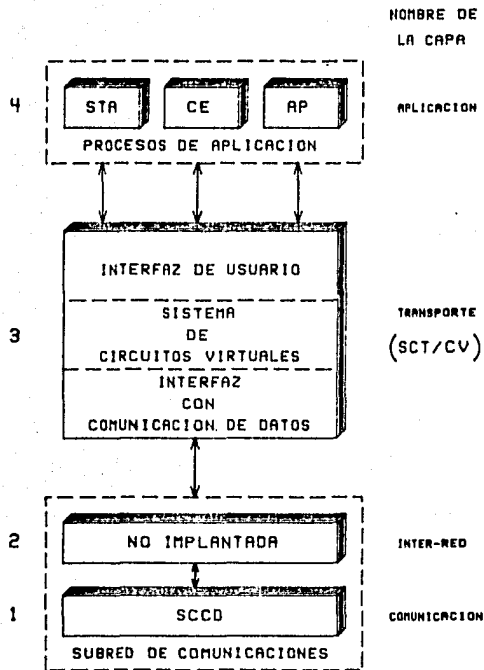


Figura 5.1
ESTRUCTURA OPERATIVA DEL SCT/CV

proporcionar el servicio fin-a-fin de comunicación virtual libre de errores. Entre las funciones involucradas en este módulo se encuentran: el manejo de las estructuras para los puertos de transporte y los buffers de comunicación; las solicitudes de envío y recepción de mensajes que generan los procesos de aplicación; los ventanas de secuencia para la transmisión y recepción de mensajes; y el esquema de control de flujo. Su diseño toma en cuenta las restricciones de la máquina en cuanto a manejo de memoria para estructuras de datos, buffers para el manejo de los mensajes y espacio para el mismo código del SCT/CV.

Interfaz de Usuario

Este es el módulo superior de la estructura operativa del SCT/CV y es el encargado de desarrollar el conjunto de funciones de la interfaz con la capa de aplicaciones. El manejo adecuado de estas funciones permite que los procesos de aplicación puedan tener acceso al servicio de comunicación y transporte de datos proporcionado por el SCT/CV. En este módulo se realiza la validación de las funciones y sus parámetros. Las funciones han sido diseñadas para permitir que los procesos de aplicación puedan ejercer un manejo de la red suficiente, flexible, clara y sencillo.

SELECCION DE LA ESTRUCTURA DE IMPLANTACION

La selección de la estructura de implantación para el SCT/CV consiste en determinar la conformación de la tarea que se utilizará para la implantación del sistema en una máquina. Esta selección es importante ya que de la estructura seleccionado se deriva la definición de una serie de aspectos de implantación importantes tales como: tipos y

formatos de estructuras de datos, interfaces entre submódulos, estructuración de las áreas de memoria para buffers, transferencia de datos con las copias adyacentes, sobrecarga en el manejo de los datos, y otros más. Además, el éxito en la obtención de una implantación eficiente depende en gran medida del tipo de estructura manejada.

Aunque un sistema de transporte puede ser desarrollado como una tarea de usuario normal, no es muy recomendable debido a la baja eficiencia y poco alcance que se puede esperar en el manejo de los recursos de la máquina, y por lo tanto, para la implantación del sistema. En el caso del SCT/CV, se mantiene el interés de obtener una implantación que sea tan eficiente como lo permita la máquina. Por esto, su implantación como un proceso de usuario normal no resulta atractiva.

Para poder lograr una implantación eficiente del SCT/CV, éste debe realizar un manejo adecuado de los recursos de la máquina, integrándose tan cerca del sistema operativo como sea posible. De esta forma, el SCT/CV podrá tener a su alcance los mecanismos para el manejo de los interfaces de entrada/salida, interrupciones, memoria, y en general, todos aquellos directivos del sistema operativo que le permitan apoyar las funciones de su protocolo de una manera eficiente.

En el proceso de selección de la estructura de implementación del SCT/CV para los PDP se analizaron varias posibilidades, entre las cuales, las tres principales son: comunicación entre procesos utilizando los propios mecanismos del sistema operativo de la máquina, manejo de un ACP (Ancillary Control Process) y manejo de un conjunto de funciones de biblioteca que puedan residir en el Área de memoria de cada proceso de aplicación. Ver figura 5.2.

En los tres casos anteriores, es necesario desarrollar un manejador (handler o driver) de comunicaciones, que permita realizar el control de la interfaz de DMA (DR11-B), que se seleccionó como vía de Entrada/Salida de la máquina para el acceso a la subred (capa de comunicación de datos). Aunque en cada caso el manejador puede tener requerimientos diferentes, su desarrollo es similar, por lo que no se consideró como factor de decisión para el proceso de selección. A continuación se describen los detalles de los tres casos.

Comunicación entre Procesos

En su operación normal, el SCT/CV requiere de un conjunto de mecanismos de enlace con los procesos de aplicación locales, que le permitan definir la interfaz a través de la cual pueda proporcionar sus servicios. Una forma de resolver esto, consiste en utilizar los mecanismos que proporcionan la mayoría de las máquinas para la comunicación entre sus procesos. En este esquema, el SCT/CV puede ser implantado como una tarea de usuario (tal vez operando en algún modo de alta prioridad), el cual, apoyándose en las funciones que a ese nivel le proporciona el sistema operativo para el enlace con los procesos de aplicación, podrá ofrecer sus servicios. Los problemas de este esquema en los PDPs bajo RSX-11M son que los mecanismos de comunicación entre procesos, no están pensados para fuertes volúmenes de intercambio de información, además de que no proporcionan un esquema realmente eficiente. Por otro lado, el implantar el SCT/CV de esta forma restringe sus posibilidades en el manejo de los recursos de la máquina, complica en cierta forma el manejo de la interfaz de usuario y no permite un manejo eficiente del área de memoria

Implantación del SCT/CV Bajo RSX-11M

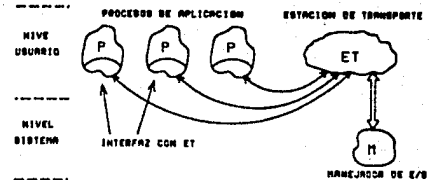
requerida, ya que tanto los procesos de aplicación como el SCT/CV deben poseer buffers separados. Ver figura 5.2a.

ACP

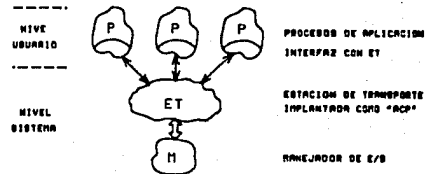
El predecesor del SCT/CV en la arquitectura de Red-11M3 fue un protocolo de comunicación que proporcionaba un servicio de datagramas confiables denominado HERMES. Este sistema fue desarrollado como el primer protocolo de transporte para la arquitectura de la red y se implantó en base a una metodología característica de las PDP, conocida como "Ancillary Control Process" (ACP), y sus manejadores correspondientes.

Una implantación en ACP permite manejar funciones relacionadas con los recursos de la máquina, las directivas del sistema operativo y los manejadores, de una manera eficiente. Una implantación así para el SCT/CV, proporciona una estructura que permite un manejo eficiente de los recursos de la máquina, pero tiene varios problemas fuertes. Estos son: es un sistema muy particular de estas máquinas, y no se cuenta con la documentación adecuada por parte de DIGITAL; no existe infraestructura de apoyo para el desarrollo y prueba de sistemas bajo este esquema, lo cual complica la implantación; los sistemas desarrollados son tan dependientes del sistema operativo, que cualquier cambio de versión del mismo los afecta seriamente; no facilita un manejo eficiente de la memoria, ya que tanto el usuario como el SCT/CV deben poseer sus propios buffers. En la figura 5.2b se ilustra la estructura utilizando un ACP.

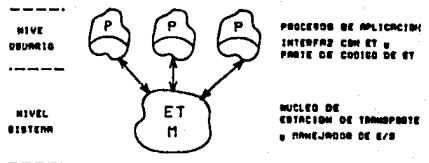
Biblioteca de Funciones



A) COMUNICACION ENTRE PROCESOS



B) ACP



C) BIBLIOTECA DE FUNCIONES

Figura 5.2
ESTRUCTURAS DE IMPLANTACION

Esta opción, ilustrado en la figura 5.2c, se basa en el desarrollo de parte del SCT/CV como un conjunto de funciones, que implantando la interfaz de usuario y parte del protocolo de circuitos virtuales, pueden ser accedidas de la librería de la red. En este esquema se tiene la ventaja de que el código de las funciones puede ser ligado con los procesos de aplicación, residiendo, por lo tanto, en su espacio de direccionamiento. Esto permite que se pueda lograr una comunicación rápida y eficiente entre los procesos de aplicación y el SCT/CV, además de que permite compartir los mismos buffers entre los procesos de aplicación y el SCT/CV. Para que este esquema este completo requiere, al igual que los anteriores, de un manejador de comunicaciones. El manejador debe proporcionar la comunicación con el exterior y el sistema operativo, además de constituir el núcleo de la capa de transporte. Esta parte del SCT/CV, tiene la ventaja de estar cerca del sistema operativo y pueda realizar las actividades en forma eficiente.

Aunque cualquier desarrollo que se involucre directamente con el sistema operativo acarrea riesgos de dependencia, esta opción es menos dependiente que el caso del ACP, y además permite que los buffers de comunicación puedan ser compartidos. Con esto se puede obtener una seguridad razonable en el desarrollo del sistema, se permite un manejo eficiente de la memoria de la máquina y se reduce el tiempo de tránsito de información entre las capas adyacentes, aprovechándose el esquema de DMA. Además, en este caso no se requiere de algún tipo de información especial, como en el caso del ACP.

En resumen, de los tres casos anteriores, la opción de comunicación

entre procesos empleando los esquemas disponibles, no proporciona una alternativa eficiente y adecuado, y aunque tal vez en otros máquinas podría llegar a ser una buena opción (o la única), en el caso de esta implantación del SCT/CV no es una solución aceptable. La opción del ACP, es buena pero muy vulnerable a cambios de versión del sistema operativo, además de que, si no se cuenta con información, el desarrollo se complica y no hay muchas esperanzas de obtener una buena implantación. Finalmente, la opción de la librería de funciones, resulta igualmente interesante como la del ACP, pero sin sus desventajas principales. Este esquema facilita el manejo de la interfaz de usuario, permite un manejo eficiente de la memoria para buffers, permite el acceso a las funciones del sistema operativo, permite reducir el tiempo de transferencia y procesamiento de información, entre otros. Por lo tanto, es claro que la mejor opción para el desarrollo del SCT/CV en las computadoras PDP bajo el sistema operativo RSX-11M, es la de utilizar una biblioteca de funciones y un manejador de comunicaciones.

ESTRUCTURA DE IMPLANTACION

La estructura de implantación del SCT/CV fue evaluada cuidadosamente para cumplir con los requisitos planteados en el desarrollo de su protocolo (como se vid en la sección anterior). Esta estructura, está constituida por dos módulos contenidos totalmente en el anfitrión, y han sido desarrollados en lenguaje ensamblador. Estos módulos son una Biblioteca de Funciones (BF) y un Manejador de Comunicaciones de la Red (MC). La BF es asequible para los procesos de aplicación desde un lenguaje de alto nivel, a través de las funciones de la interfaz de usuario. El MC realiza el manejo de la interfaz con el exterior de la

Implantación del SCT/CV Bajo RBX-11M

computadora, definido por la capa de comunicación de datos, y algunas otras funciones. Estos dos módulos se comunican entre sí a través de la interfaz que los separa y que se ha denominado interfaz de bajo nivel. En la figura 5.3 se ilustra la estructura de implantación del SCT/CV, y la descripción completa de sus módulos se da a continuación.

Manejador de Comunicaciones de la Red

La necesidad de una interfaz de Entrada/Salida en el anfitrión para el enlace con la capa de comunicación de datos, requiere del desarrollo de un manejador como parte del SCT/CV, el cual se ha denominado Manejador de Comunicaciones de la Red (MC). El MC no sólo permite el manejo y control de la Interfaz de DMA (**IDMA** que es el DR11-B) para lograr la comunicación con la capa inferior, sino que además permite realizar las funciones necesarias para la interacción del SCT/CV con el sistema operativo.

Es importante destacar que en la PDP, como en las demás máquinas, los manejadores son parte del sistema operativo por lo que su código debe estar bien estructurado, ser eficiente, y respetar los reglas de manejo de las estructuras de datos del sistema y sus interrupciones. En la PDP, un manejador posee varios puntos de acceso, de los cuales, los importantes para el desarrollo del SCT/CV fueron los del inicio de operaciones, terminación de Time-Outs, cancelación de tareas y recepción de interrupciones. Estos puntos fueron aprovechados para optimizar la implantación del SCT/CV.

De acuerdo a la figura 5.3, las funciones del SCT/CV que se han implantado en el MC, tienen que ver con la parte de control global del

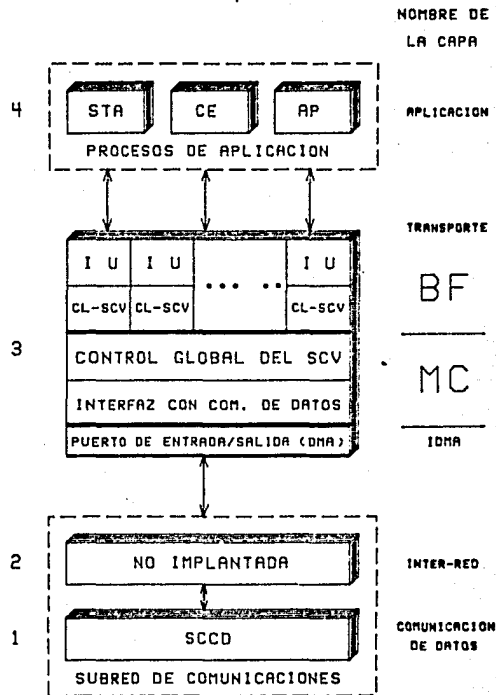


Figura 5.3
ESTRUCTURA DE IMPLANTACION DEL SCT/CV

Sistema de Circuitos Virtuales y el manejo de la interfaz con la capa de comunicación de datos. La parte de control global involucra el manejo de las funciones de coordinación de todos los procesos de aplicación a través de sus puertos correspondientes de control local (CL-SCV) de la BF. Estas funciones tienen que ver con la asignación y cancelación de puertos de transporte, la validación de todo tipo de solicitud, el redireccionamiento de solicitudes de envío y recepción de mensajes, la conversión de direcciones virtuales a físicas y la definición de la interfaz de comunicación de bajo nivel que existe entre el MC y la BF. En el manejo de la interfaz con la capa de comunicación de datos se realizan funciones tales como la programación y control de la interfaz de DMA de la computadora, el multiplexaje y demultiplexaje de mensajes entre las capas de comunicación de datos y aplicaciones, y la inicialización y coordinación del CCD.

Para la implantación de las funciones de control global del sistema de circuitos virtuales y de manejo de la interfaz de comunicación de datos en el MC, fue necesario definir una estructura de datos denominada Tabla de Puertos y Solicitudes (TPS), determinar los mecanismos para la liberación de los puertos que dejan colgados los procesos de aplicación, definir el manejo de tiempos de espera para proteger al MC de problemas en la interacción con la capa de comunicación de datos, manejar la conversión de direcciones que permitiera la programación adecuada de la interfaz de DMA y la transferencia directa entre los buffers de las capas adyacentes y, finalmente, definir la interfaz de bajo nivel a través de la cual se comunicarían el MC y la BF. Estos aspectos importantes del proceso de implantación del MC se describen a continuación.

ESTRUCTURAS DE DATOS. Además de las estructuras normales de operación que utiliza un manejador, como son el Bloque de Control del Dispositivo (DCB), el Bloque de Control de la Unidad (UCB), y el Bloque de Control de Estado (SCR), en el MC fue necesario definir una estructura denominada Tabla de Puertos y Solicitudes (TPS) que permitiera el manejo adecuado de los puertos de transporte y las solicitudes de envío y recepción de mensajes emitidas por los procesos de aplicación a través de la BF.

Tabla de Puertos y Solicitudes (TPS). En la POP, las solicitudes que emiten los procesos a los manejadores son realizadas a través de pequeñas estructuras de datos, denominadas "I/O Packets". Estas estructuras contienen la información concerniente al tipo de operación solicitada por los procesos, y los parámetros necesarios para su realización. Cuando una solicitud de éstas no puede ser atendida inmediatamente por un manejador, se debe guardar en alguna estructura, en espera de la oportunidad para su ejecución.

La TPS es una estructura del Manejador de Comunicaciones de la Red, que se utiliza para apoyar el esquema de coordinación de los puertos de transporte entre todos los procesos de aplicación, y para realizar el manejo ordenado de las solicitudes de envío y recepción de mensajes que son emitidas por la BF. En la figura 5.4 se muestra esta estructura, y la descripción de sus campos es la siguiente.

PS.EDO Campo de una palabra (2 bytes) utilizado para manejar el estado de los puertos del SCT/CV. El estado de cada puerto es representado por el bit correspondiente dentro de esta palabra, con los valores de,

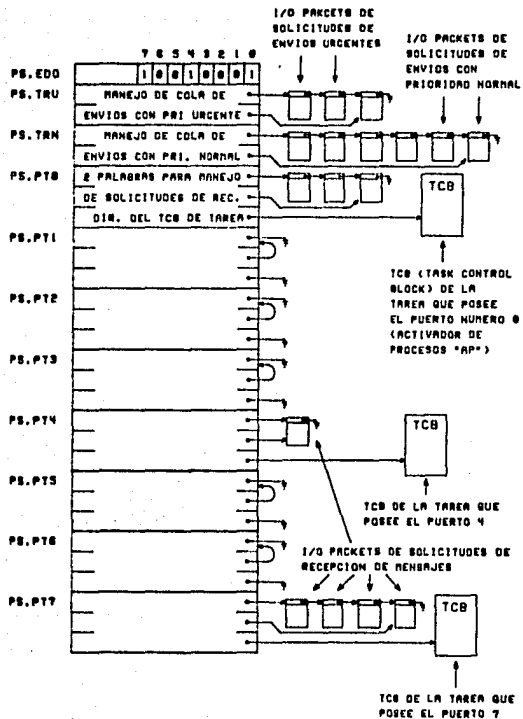


Figura 5.4

TABLA DE PUERTOS Y SOLICITUDES (TPS)

cero para disponible y uno para asignado. Por ejemplo, si el bit 5 de la palabra PS.EDD tiene el valor de uno, entonces el puerto cinco se encuentra asignado, pero si posee el valor de cero el puerto está disponible.

La estructura actual está conformada para manejar únicamente ocho puertos (del 0 al 7), ya que éste fue el valor que se determinó para el manejo de la interfaz con la capa de comunicación de datos. No obstante, no es una limitante ya que es muy sencillo expandir este valor, aunque difícilmente la PDP podría soportar más de ocho procesos simultáneos.

PS.TRU Campo de dos palabras (4 bytes) utilizado para el manejo de la cola de solicitudes de envío de mensajes con prioridad de urgentes. Esta cola se organiza como un FIFO de las solicitudes emitidas por los procesos de aplicación a través de todos los puertos activos.

PS.TAN Campo de dos palabras (4 bytes) utilizado para el manejo de la cola de solicitudes de envío de mensajes con prioridad normal. La cola es organizada como un FIFO de las solicitudes emitidas por los procesos a través de todos los puertos de transporte activos.

PS.PTB Campo de tres palabras (6 bytes) utilizado para el manejo de la cola de solicitudes de recepción de mensajes, y la dirección del Bloque de Control de la Tarea (TCB), de la tarea que posee el puerto. Las dos primeras palabras se emplean para el manejo de la cola de solicitudes de recepción con una organización FIFO. La tercera palabra es utilizada para almacenar la dirección del TCB de la tarea dueña, la cual es registrada al momento de asignar el puerto. En este caso la

"n" representa un valor entero, entre cero y siete, que corresponde al valor del puerto a que se hace referencia. En la estructura de la TPS existen ocho conjuntos de tres palabras para PS.PTn, consecutivos, dados por PS.PT0, PS.PT1, ..., PS.PT7.

CANCELACION DE PUERTOS. Un problema de implantación encontrado normalmente en el desarrollo de un sistema de transporte multiusuario, consiste en determinar la manera de cancelar los puertos asignados, cuando las tareas que los poseen terminan de forma anormal, y por lo tanto, no tienen la oportunidad de hacerlo a través de los mecanismos dispuestos para ello. La responsabilidad del SCT/CV al respecto, consiste en evitar que se vayan quedando puertos colgados, y que no puedan ser reutilizados. Para resolver este problema, se ha aprovechado una característica del sistema operativo de la PDP. Cuando el sistema operativo detecta que una tarea termina pero posee solicitudes activas con un manejador, antes de permitirle que pueda concluir, le pasa el control al manejador involucrado. Cuando el sistema operativo le transfiere el control al MC en el punto de exceso de cancelación de tareas, le proporciona la dirección del TCB de la tarea que está terminando. Utilizando este valor, el MC revisa la TPS, comparando el valor de la tercer palabra de todos los campos PS.PTn. Si en ese momento se encuentra que alguna de los puertos activos pertenece a la tarea que termina, se liberan todas las solicitudes que posee, ya sean de envío o recepción y se libera el puerto dejándolo en estado de no asignado. Con ésto, el puerto queda disponible para otras tareas, y la tarea puede terminar sin dejar puertos colgados.

TIEMPOS DE ESPERA (TIME-OUTS). El enlace del MC con la capa de

comunicación de datos implantada en el CCD, se realiza a través de una interfaz de DMA (DR11-B), la cual debe ser programada por cada operación de transferencia requerida. Después que el MC programa una operación en el DR11-B, se coordina con el CCD, para que este haga lo correspondiente, y así pueda iniciarse la transferencia de datos. Una vez activada una transferencia, el MC debe esperar, primero por la respuesta del DR11-B y luego por la del CCD. En la respuesta del CCD, éste le informa al MC el resultado que obtuvo en la operación.

El hecho de que el MC esté supeditado a esperar la respuesta del CCD en cada transferencia, puede presentar serios problemas de dependencia sobre el buen funcionamiento del CCD, si el MC no tomara las medidas de protección adecuadas. Para ésto, se ha aprovechado la facilidad de manejo de tiempos de espera que poseen los manejadores en la PDP, y el punto de acceso de "time-outs". Al iniciar una operación, el MC activa el tiempo de espera correspondiente. Si una operación no termina antes que el tiempo de espera programado, se cancela la operación y se toman las medidas pertinentes de acuerdo al caso. Si la respuesta del CCD ocurre antes de que concluya el tiempo de espera, éste es apagado y se da fin a la operación en base al resultado proporcionado por el CCD.

En los casos en que el tiempo de espera termine antes que la operación de transferencia, el MC reintenta la operación un máximo de tres veces, si después de ésto no tiene éxito, envía una señal de reestablecimiento (RESET) al CCD, para inicializarlo. Después de ésto, el MC vuelve a intentar la operación de transferencia por otras tres veces más. Si después de ésto el MC continúa sin éxito, termina la operación de transferencia estableciendo el código de error

correspondiente.

CONVERSION DE DIRECCIONES. El SCT/CV se ha desarrollado para operar en diferentes espacios de direccionamiento. Por un lado, el código de las funciones de la BF reside en los espacios de direcciones de los diferentes procesos de aplicación que interactúan en la red. Por otro lado, el código del MC se encuentra en el espacio de direcciones reservado para los manejadores de E/S de la máquina. Esta división reduce la cantidad de memoria requerida para buffers, y facilita la comunicación tanto entre los procesos de aplicación y el código del SCT/CV de la BF, como entre el código del SCT/CV del MC y el sistema operativo. Sin embargo, requiere de la conversión de direcciones virtuales de los buffers que comparten las BFs y los procesos de aplicación, a las direcciones físicas de los mismos. Esto es requisito para que puedan coordinarse los partes correspondientes de las BFs y el MC, y el MC pueda programar la interfaz de DMA con las direcciones físicas de los buffers de los procesos de aplicación. Con esto se permite la transferencia directa de mensajes entre los buffers de los procesos de aplicación y los de la capa de comunicación de datos, sin tener que almacenarlos en algún lugar intermedio. Esto permite la reducción, tanto del tiempo de procesamiento de la capa, como de la cantidad de memoria necesaria para su operación, obteniendo un mejor rendimiento en la operación de la red.

Para resolver el problema de conversión de direcciones, se utilizaron algunas rutinas del sistema operativo, cuyo manejo fue extraído de los listados fuentes del mismo. Las rutinas empleadas fueron %RELOC, %MPPHY, del archivo IDSUB.MAC. Además, fue necesario manejar otros

parámetros de las estructuras de datos como son el registro de páginas KISARÁ y el bit UC.QUE del UCB, para lograr la interacción necesaria con el ambiente de estado de los parámetros que refieren a los procesos. Para una información detallada consultar la referencia [2].

FUNCIONES DE BAJO NIVEL DEL MC. La interacción entre el Manejador de Comunicaciones de la Red y la Biblioteca de Funciones se realiza a través de una pequeña interfaz, definida de tal forma que estas dos módulos podrían verse como dos subcopias del SCT/CV. El MC proporciona un servicio de comunicación básico, a través de un conjunto de funciones de bajo nivel sobre el cual se construyen las funciones de la BF (funciones de alto nivel), la que a su vez proporciona el servicio de comunicación fin-a-fin de circuitos virtuales. La definición de la interfaz del MC, así como los formatos y significados de cada una de sus funciones de acceso se proporcionan a continuación.

Activar Puertos. Para la activación y asignación de un puerto de transporte, la BF debe solicitar a su vez al MC la asignación de un puerto disponible, el cual, utilizando la TPS sabe si existen o no dentro del contexto general de toda la máquina, y cuales son. Para realizar la solicitud de puertos, la BF utiliza la siguiente directiva del MC:

`D10W%S #ID.APT,,,#IOSB,,<#NUMPTO>`

Donde:	D10W%S	Es la directiva del sistema utilizada para las operaciones de Entrada/Salida.
	#ID.APT	Es la función del MC utilizada para solicitar un puerto de transporte.
	#NUMPTO	Dirección de la variable donde el MC regresa el

Implantación del SCT/CV Bajo RBX-11M

valor del puerto asignado. La excepción, es cuando se solicita el puerto del AP, con el valor de cero, en cuyo caso se da únicamente la respuesta a la operación.

#IOSB Apuntador a un bloque de dos palabras, donde el MC regresa el resultado de la operación. En el Apéndice B se proporcionan estos códigos.

Los parámetros no incluidos en la directiva se manejan de acuerdo a las restricciones y características de la máquina y pueden ser consultados en [1].

Liberar Puertos. Para la liberación de puertos de transporte, la BF debe solicitarse al MC, el cual, primero cancela todas las solicitudes de envío y recepción que estén pendientes en las estructuras y después de esto procede con la cancelación correspondiente. El formato de la directiva de liberación es el siguiente:

Q10W#S #ID.DPT,,,#IOSB,,<#NUMPTD>

Donde: **Q10W#S** Es la directiva del sistema utilizada para las operaciones de Entrada/Salida.
#ID.DPT Es la función del MC utilizada para solicitar la cancelación o liberación de un puerto.
#NUMPTD Dirección de la variable donde se le comunica al MC el puerto implicado en la operación.
#IOSB Apuntador a un bloque de dos palabras, donde el MC regresa el resultado de la operación. En el Apéndice B se proporcionan estos códigos.

Los parámetros no incluidos en la directiva se manejan de acuerdo a las restricciones y características de la máquina y pueden ser consultados en [1].

Implantación del SCT/CV Bajo RSX-11M

Enviar Mensajes. Para el envío de mensajes, el MC proporciona dos directivas diferentes, que se utilizan para envíos a estaciones remotas y a la estación local, que son:

Q10#S #IO.WVB,,,#IOSB,,<#NUMPTD,#INIMSJ,#TAMMSJ,#PRIMSJ>

Q10#B #ID.WLC,,,#IOSB,,<#NUMPTD,#INIMSJ,#TAMMSJ,#PRIMSJ>

Donde: **Q10#S** Es la directiva del sistema utilizada para las operaciones de Entrada/Salida.
#IO.WVB Es la función del MC utilizada para solicitar el envío de un mensaje a una estación remota.
#ID.WLC Es la función del MC utilizada para solicitar el envío de un mensaje a la misma estación.
#NUMPTD Dirección de la variable donde se le comunica al MC el puerto por el cual se desea realizar el envío del mensaje.
#INIMSJ Dirección de la variable que posee la dirección virtual del inicio donde se encuentra localizado el mensaje en memoria.
#TAMMSJ Dirección de la variable que contiene el valor del tamaño del mensaje a transmitir.
#PRIMSJ Dirección de la variable que contiene el valor de la prioridad de envío del mensaje. Donde URGENTE = 1 y NORMAL = 0.
#IOSB Apuntador a un bloque de dos palabras, donde el MC regresa el resultado de la operación. En el Apéndice B se proporcionan estos códigos.

Los parámetros no incluidos en la directiva se manejan de acuerdo a las restricciones y características de la máquina y pueden ser consultados en [1].

Recibir Mensajes. Para el manejo de recepciones de mensajes el MC provee una directiva que permite colocar solicitudes de recepción en la cola del puerto dentro de la TPS del MC. Una vez recibidos estos

solicitudes, quedan en espera de recibir algún mensaje de la línea. Cuando se recibe un mensaje, el MC lo pasa a la DF, de acuerdo a los parámetros de la directiva emitida. La directiva de recepción es:

`QIOSB #IO.RVB,,,#IOSB, <#NUMPTD,#INITEX,#TANBUF,#TAMTEX>`

Dónde:

<code>QIOSB</code>	Es la directiva del sistema utilizada para las operaciones de Entrada/Salida.
<code>#IO.RVB</code>	Es la función del MC utilizada para solicitar la recepción de un mensaje.
<code>#NUMPTD</code>	Dirección de la variable donde se lo comunica al MC, el número de puerto para recibir.
<code>#INITEX</code>	Dirección de la variable que posee la dirección virtual del inicio donde se encuentra localizado el buffer para la recepción del mensaje.
<code>#TANBUF</code>	Dirección de la variable que contiene el valor del tamaño del buffer para recibir el mensaje.
<code>#TAMTEX</code>	Dirección de la variable donde el MC regresara el valor del tamaño del mensaje recibido.
<code>#IOSB</code>	Apuntador a un bloque de dos palabras, donde el MC regresa el resultado de la operación. En el Apéndice B se proporcionan estos códigos.

Los parámetros no incluidos en la directiva se manejan de acuerdo a las restricciones y características de la máquina y pueden ser consultados en [1].

Revisión de Estructuras. Existen dos directivos más del MC que pueden ser utilizados para revisar o monitorear el estado de la estructura de la TPS que, aunque son muy simples, fueron de una gran ayuda en el proceso de implantación y depuración del MC. Estas directivas son:

`QIOWS #IO.ATT,,,#IOSB`

`QIOWS #IO.DET,,,#IOSB`

Dónde:

<code>QIOWS</code>	Es la directiva del sistema utilizada para las operaciones de Entrada/Salida.
<code>#IO.ATT</code>	Es la función del MC utilizada para inicializar un apuntador a la estructura de la TPS.
<code>#IO.DET</code>	Es la función del MC utilizada para revisar el valor de la palabra apuntada, después de leer un valor con esta directiva, se incrementa el apuntador, seleccionando la siguiente palabra de la estructura de la TPS. Al leer la última palabra, el apuntador se reinicializa a la primera palabra de la estructura.
<code>#IOSB</code>	Apuntador a un bloque de dos palabras, donde el MC regresa el resultado de la operación, y en el caso del <code>IO.DET</code> , en la segunda palabra también el contenido de la dirección leída. En el Apéndice B se proporcionan los códigos.

Los parámetros no incluidos en la directiva se manejan de acuerdo a las restricciones y características de la máquina y pueden ser consultados en [1].

Biblioteca de Funciones

La idea de manejar parte del SCT/CV como una Biblioteca de Funciones (BF), surgió de la ideología utilizada por el lenguaje 'C', donde, por cada una de las funciones invocada se le incorpora a la aplicación el código que contiene tanto el manejo de alto nivel de la función, así como las llamadas al núcleo de la red, que en este caso es proporcionada por el MC. El manejo de la BF permite que el SCT/CV pueda compartir el mismo espacio de direcciones de los procesos de aplicación, con lo cual ambos pueden acceder los mismos buffers. Al no requerirse buffers separados para los procesos y el SCT/CV, se optimiza el manejo de memoria. Al no tener que estar recopilando información entre los procesos de aplicación y el SCT/CV se ahorra tiempo de procesador y se eficientiza la transferencia de datos y, por

tanto, el esquema de comunicaciones puede tener un mejor comportamiento.

Un detalle importante que debe ser cuidado en este esquema es que, como el código de las funciones de la BF se liga a cada tarea, es necesario que sea compacto y eficiente, pero sin dejar de ser completo y sencillo, para que así, los procesos de aplicación puedan obtener un servicio de enlace y comunicación adecuado o un costo razonable.

La BF está constituida por la parte de código que realiza el manejo de las funciones de la Interfaz de Usuario (IU), y por el conjunto de funciones que realizan la parte de Control Local del Sistema de Circuitos Virtuales (CL-SCV), como se ilustra en la figura 5.3. La IU, es la parte de código del SCT/CV a través de la cual, los procesos de aplicación acceden el servicio proporcionado por el SCT/CV. El código de control local (CL-SCV), es la parte del SCT/CV que se encarga de las funciones de manejo del esquema fin-a-fin de comunicación de circuitos virtuales, tales como: manejo de enlaces, puertos, buffers, mensajes, reconocimientos, ventanas de comunicación y esquema de control de flujo.

Para cumplir con las funciones requeridas en la BF, fue necesario definir estructuras de datos, esquemas para el manejo de tiempos de espera (time-outs) y colas de solicitudes, que son descritos a continuación.

ESTRUCTURAS DE DATOS. Las estructuras definidas para la operación de la BF, son el Bloque Descriptor de Puerto (BDP), los buffers para el envío y recepción de mensajes, y los buffers para solicitud de

información de estado.

Bloque Descriptor de Puerto (BDP). Esta estructura consiste de un conjunto de datos que permiten el manejo adecuado de los parámetros importantes de los puertos, así como de las conexiones en que puedan involucrarse. Cada puerto activo debe tener asignado un BDP que lo identifica y describe. En la figura 5.5 se ilustra el formato del BDP y la descripción de cada uno de sus campos es la siguiente:

P.NPT Campo de un byte, utilizado para registrar el número de puerto al que hace referencia el bloque descriptor.

P.EDO Campo de un byte, utilizado para registrar el estado de operación en el que se encuentra el puerto descrito. Los estados posibles son los siguientes:

ABREVIATURA	VALOR	DESCRIPCION
EP.INA	-1	Inactivo o cerrado
EP.ACT	1	Activo
EP.EFN	2	En espera por conexión
EP.TCN	4	Tramitando conexión
EP.CON	8	Conectado

P.NBR Campo de un byte, utilizado para registrar el número de buffers que el puerto tiene disponibles para la recepción de mensajes.

P.NRT Campo de un byte, utilizado para registrar el número de buffers que el puerto tiene disponibles para el envío de mensajes.

P.TBR Campo de una palabra (2 bytes), utilizado para el registro del

Implementación del SCT/CV Bajo RSX-11M

tamaño de los buffers de recepción.

P.TBT Campo de una palabra (2 bytes), utilizada para el registro del tamaño de los buffers de transmisión.

P.VER Campo de un byte, utilizado para el manejo de la ventana de recepción, requerido por el esquema de secuenciamiento de los mensajes.

P.VET Campo de un byte, utilizado para el manejo de la ventana de transmisión, requerido por el esquema de secuenciamiento de los mensajes.

P.CND Campo de un byte, utilizado para el manejo de los códigos de condición del estado del puerto. El valor de este campo se inicializa con el valor de éxito y sólo cambia por el valor correspondiente a un código de error, cuando ocurre un problema que es irreparable para el SCT/CV. El proceso de reinicialización de una conexión puede cambiarlo nuevamente a éxito si se corrige el problema.

P.NRE Campo de un byte, utilizado para el manejo del contador de retransmisiones. El valor de este campo se aplica al envío activo, y es reinicializado en cada nueva transmisión.

P.PTR Campo de dos palabras (4 bytes), utilizado para el manejo de la cola de buffers para la recepción de mensajes. La cola es organizada como un FIFO de los mensajes recibidos.

P.PTT Campo de dos palabras (4 bytes), utilizado para el manejo de la cola de buffers disponibles para el envío de mensajes.

P.PTA Campo de una palabra (2 bytes), utilizado como apuntador al

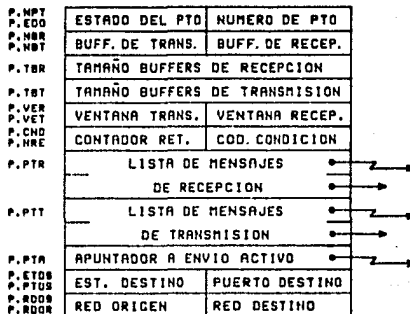


Figura 5.5
FORMATO DEL BOP

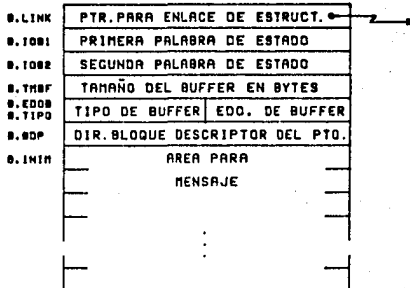


Figura 5.6
ESTRUCTURA PARA MANEJO DE BUFFERS

buffer que posee el mensaje que se transmite (mensaje en tránsito) en ese momento. Esta palabra en coordinación con la segunda de P.PTT se utilizan para manejar una lista FIFO, para el envío ordenado de mensajes. En esta cola los mensajes urgentes son ligados al inicio y no al final como en el caso de los mensajes con prioridad normal.

P.ETDS Campo de un byte, utilizado para registrar la dirección de la estación destino. Este campo sólo es válido cuando el campo P.EDO posee el valor de 2 (EP.ECN) o el de 4 (EP.TCN) o el de 8 (EP.CON).

P.PTDS Campo de un byte, utilizado para registrar la dirección del puerto destino. Este campo sólo es válido cuando el campo P.EDO posee el valor de 2 (EP.ECN) o el de 4 (EP.TCN) o el de 8 (EP.CON).

P.RDDS Campo de un byte, reservado para el manejo de la dirección de red destino, cuando el puerto participa en una conexión.

P.RDOR Campo de un byte, reservado para el manejo de la dirección de red origen, cuando el puerto participa en una conexión.

Buffers para Mensajes. La estructura utilizada para el manejo de los buffers que contienen los mensajes a ser enviados o recibidos se ilustra en la figura 5.4, y a continuación se proporciona la descripción de cada uno de sus campos.

B.LINK Campo de una palabra (2 bytes), utilizada para enlazar los buffers en la cola correspondiente, de acuerdo a su tipo y condición.

B.IDB1 Campo de una palabra (2 bytes), utilizada como primera palabra del bloque de IOSB requerido en el manejo de los macros QIO del

sistema que se emplea en la interacción con el Manejador de Comunicaciones de la Red.

B.IDB2 Campo de una palabra (2 bytes), utilizada como segunda palabra del bloque de IOSB requerido en el manejo de los macros QIO del sistema que son empleados en la interacción con el Manejador de Comunicaciones de la Red.

B.TMBF Campo de una palabra (2 bytes), utilizado para registrar el tamaño en bytes del buffer.

B.EDOB Campo de un byte, utilizado para registrar el estado en el que se encuentra el buffer. Los valores que se manejan son:

ABREVIATURA	VALOR	DESCRIPCION
EB.LIB	0	Estado inicial (LIBRE)
EB.USR	1	Buffer en posesión del Usuario
EB.OCF	2	Buffer Ocupado con información. Si es de: -Rec: Mensaje recibido, esperando que el proceso de aplicación lo pida. -Tr: Esperando para ser enviado.
EB.TRN	4	Buffer en Tránsito. Si es de: -Rec: Esperando un mensaje, en cola del Manejador de Comunicaciones. -Tr: En envío. Posiblemente esperando reconocimiento de su arriba.

B.TIPO Campo de un byte, utilizada para manejar el tipo de buffer de que se trata. Los tipos de buffers son:

ABREVIATURA	VALOR	DESCRIPCION
TB.REC	0	Buffer de recepción.
TB.TRN	1	Buffer de transmisión.

B.BDP Campo de una palabra (2 bytes), utilizada para registrar la

dirección del BDP, del puerto al que pertenece el buffer.

B.INIH Campo de n bytes, utilizada para indicar el inicio del mensaje almacenado en el buffer. Este campo puede ser válido o no, dependiendo del estado del buffer, dado por el campo B.EDOB.

Buffers de Estado. Para que los procesos puedan obtener información de estado correspondiente a sus puertos, se ha diseñado una estructura que utiliza el SCT/CV. Esta estructura está conformada por un bloque de 18 bytes representada en la figura 5.7. En esta estructura todos los campos son de un byte y su descripción es la siguiente.

E.ETL Dirección correspondiente a la estación (anfitrión) local.

E.MPT Número de puerto del que se solicita la información.

E.EDO Estado de operación del puerto.

E.CND Código de condición del estado del puerto.

E.NBR Número total de buffers de recepción.

E.MBT Número total de buffers de transmisión.

E.RLIB Número de buffers de recepción que están libres.

E.TLIB Número de buffers de transmisión que están libres.

E.RUSR Número de buffers de recepción que posee el usuario.

E.TUSR Número de buffers de transmisión que posee el usuario.

E.RDCP Número de buffers de recepción que están ocupados.

E.TOCP Número de buffers de transmisión que están ocupados.

E.RTRN Número de buffers de recepción en tránsito.

E.TTRN Número de buffers de transmisión en tránsito.

E.VER Valor de la ventana de recepción.

E.VET Valor de la ventana de transmisión.

E.ETRM Dirección de la estación remota.

E.PTRM Dirección de puerto remoto.

E.RDRM Reservado para dirección de Red Remota.

E.RDL Reservado para dirección de Red Local.

COLAS DE SOLICITUD. Para el envío y recepción de mensajes es necesario manejar colas que permitan preservar el orden adecuado. Para esto se han reservado cinco palabras en el BDP, las cuales se organizan de la siguiente forma. Un conjunto de dos palabras direccionadas por P.PTR, para la lista ordenada de los mensajes que han arribado de la línea y esperan ser tomados por el proceso de aplicación correspondiente. Un conjunto de dos palabras direccionadas por P.PTI para el manejo de la lista de buffers para transmisión, y una tercer palabra direccionada por P.PTA, que se utiliza en conjunto con el valor de la segunda palabra de P.PTI para la lista ordenada de mensajes en espera de ser transmitidos. Este esquema permite un manejo sencillo y adecuado para la manipulación ordenada de los mensajes que se envían y reciben, cuyo operación se basa en el esquema de ventanas de secuencia para corregir

E.ETL	MUN. DE PUERTO	ESTACION LOCAL
E.MPT		
E.EDO	CONDICION	ESTADO DEL PTO.
E.CNO		
E.NBR	BUFFERS TRANSM.	BUFFERS DE REC.
E.NBT		
E.MLIB	TRANS. LIBRES	RECEP. LIBRES
E.TLIB		
E.NUSR	TRANS. USUARIO	RECEP. USUARIO
E.TUSR		
E.ROCP	TRANS. OCUPADOS	RECEP. OCUPADOS
E.TOCP		
E.RTRAN	TRANS. TRANSITO	RECEP. TRANSITO
E.TTRAN		
E.VER	VENTANA TRANSM.	VENTANA RECEP.
E.VET		
E.ETRM	PUERTO REMOTO	ESTACION REMOTA
E.PTRM		
E.NDRM	RED LOCAL	RED REMOTA
E.NDL		

Figura 5.7

ESTRUCTURA PARA BUFFERS DE ESTADO

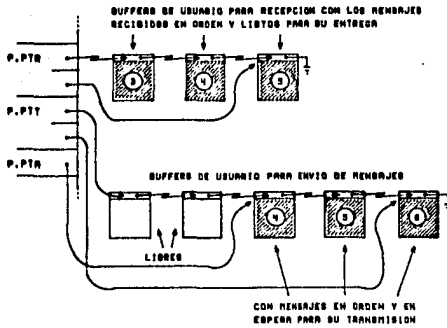


Figura 5.8

MANEJO DE COLAS DE MENSAJES DEL BDP

desordenes y duplicados. En la figura 5.8 se ilustran los colos de envío y recepción mencionadas.

TIEMPOS DE ESPERA (TIME-OUTS). La retransmisión de mensajes, realizada cuando se cumple el tiempo de espera estipulado y no se ha recibido el reconocimiento correspondiente a un mensaje transmitido, requiere de un esquema de tiempos de espera. En el caso de la implantación del SCT/CV, se han reservado los campos de P.NRE, P.PTA y P.EDO de cada BDP de los puertos activos, para el manejo de los tiempos de espera y la retransmisión de mensajes. Este esquema opera de acuerdo a los siguientes algoritmos.

Funcion_ENVIO_DE_MENSAJES

```

-----
-----
TRANSMITE_MJSJ_LINEA (ENVIO_ACTUAL, ...)
IF (TIME_OUTS <> ACTIVOS)
  FI ACTIVA (rutina_TIME-OUTS)
-----
-----

```

END

Rutina_TIME-OUTS

```

PTR <== DIR_PRIMER_BDP
LOOP: IF (ESTADO_DEL_BDP = ENVIO)
      IF (RETRANSMISIONES > 0)
        TRANSMITE_MJSJ_LINEA (ENVIO_ACTUAL, ...)
        RETRANSMISIONES = RETRANSMISIONES - 1
      ELSE
        ESTABLECE_CODIGO_ERROR
      FI
    FI
    IF (MAS_PUERTOS_ACTIVOS)
      PTR <== SIGUIENTE_DIR_BDP
      GOTO LOOP
    FI

```


END.

BIBLIOGRAFIA

- [1] RSX-11M Version 3.2, Executive Reference Manual, Digital Equipment Corporation, 1979.
- [2] RSX-11M Version 3.2, Guide to Writing on I/O Drives, Digital Equipment Corporation, 1979.
- [3] RSX-11M Version 3.2, I/O Drivers Reference Manual, Digital Equipment Corporation, 1979.

CAPITULO VI

Guía del Usuario

INTRODUCCION

El Sistema de Control de Transmisión de Circuitos Virtuales (SCT/CV) ha sido desarrollado con una interfaz de usuario de fácil manejo y accesible a los procesos de aplicación a través de un lenguaje de programación de alto nivel, como "C". La selección de "C" fue tomada por ser un buen lenguaje para el desarrollo de sistemas como son los procesos de aplicación. El diseño del SCT/CV no sólo contempla una interfaz de manejo sencilla, sino que además es poderosa y versátil.

El SCT/CV proporciona los medios para establecer mantener y terminar enlaces de comunicación confiables y eficientes, pero la comunicación fin-a-fin entre los procesos de aplicación. El SCT/CV trata de aprovechar al máximo las características importantes del diseño de la capa de comunicación de datos, y desarrolla los protocolos necesarios para proporcionar un servicio más completo.

En este capítulo se proporciona la información necesaria para el manejo adecuado de las funciones que permiten el acceso a los servicios del SCT/CV, en los computadores PDP bajo el sistema operativo RBX-11M. Se describe el manejo de las funciones de la interfaz de usuario del sistema, así como el de sus parámetros y los códigos regresados como resultado de su ejecución. Además, al final de cada explicación, se proporciona un ejemplo de su manejo.

INTERFAZ DE USUARIO

La interfaz de usuario del SCT/CV define el conjunto de primitivas a través de las cuales se proporciona el servicio de comunicación de la capa de transporte. Esta interfaz ha sido desarrollada cuidando que los procesos de aplicación puedan interactuar con el sistema de manera sencilla y directa, sin que por ello el sistema pierda eficiencia o completéz en su operación.

La interfaz de usuario del SCT/CV está constituida por cinco grupos de funciones que permiten el acceso a sus servicios. Cada grupo proporciona el manejo de las actividades relacionadas con algún tipo de operación específica del sistema, tales como con el manejo de puertos, manejo de buffers de comunicación, manejo de conexiones de transporte, manejo de mensajes de comunicación, e información de estado. Estos grupos de funciones son descritos ampliamente en las siguientes secciones.

FUNCIONES DE MANEJO DE PUERTOS

Este grupo de funciones del SCT/CV esta constituido por tres primitivos que permiten el manejo requerido para la asignación y cancelación de puertos de transporte, y la activación de recepciones.

El interés de los procesos por obtener un puerto de transporte, radica en la necesidad de poseer una dirección dentro de la red que les permita ser identificados, para poder comunicarse con otros. Las direcciones proporcionadas por los puertos de transporte en conjunto con las direcciones de las unidades (estaciones), constituyen la base de operación del esquema de comunicación y transporte de datos proporcionado por el SCT/CV. Los valores para los puertos de transporte manejados por el SCT/CV van del 0 al 7, donde el 0 se reserva para el AP, y del 1 al 7 son para los procesos de aplicación.

En las computadoras PDP con RSX-11M, cuando un proceso emite algún tipo de solicitud hacia a un manejador, éste no puede adquirir memoria para buffers en forma dinámica, si se requiere relocalizar la tarea. Este conflicto restringe a los procesos, de abrir un puerto más cuando necesen por lo menos un puerto con solicitudes de recepción de mensajes activadas. Para resolver este problema, el SCT/CV proporciona la función de activación de recepciones, que es parte de este grupo.

ABRE PUERTO

La función de abre puerto (ABREPT) es utilizada por los procesos de aplicación para solicitar la apertura y asignación de un puerto local de transporte. Una característica importante de esta función es que el SCT/CV permite que el usuario pueda definir el número y tamaño de los buffers que utilizará en su comunicación con los demás procesos. Estos valores pueden ser aplicados directamente sobre los parámetros de la función de acuerdo a los requerimientos del proceso, o bien, emplear los valores prestablecidos por el SCT/CV, cuando resulten adecuados. Los valores prestablecidos permiten el manejo de cuatro buffers de 512 bytes de espacio libre para cada mensaje, de los cuales dos buffers son de transmisión y dos de recepción. Si el parámetro NUBFRC de la función presenta el valor de cero, entonces el SCT/CV maneja los valores prestablecidos para NUBFRC, TMBURC, NUBFTR y TMBFTR. El formato de la función es el siguiente:

RES = ABREPT (NUMPTO,NUBFRC,TMBFRC,NUBFTR,TMBFTR)

Dónde:	NUMPTO	Si este parámetro es una dirección, el SCT/CV lo toma como la dirección donde regresará el valor del puerto asignado. Pero si es cero lo toma como la solicitud del puerto 0 del AP.
	NUBFRC	Es el número de buffers de recepción que se solicitan. Si su valor es cero se toma como condición para manejar los valores prestablecidos y no se hace caso de los siguientes parámetros.
	TMBFRC	Tamaño de los buffers de recepción.
	NUBFTR	Número de buffers de transmisión solicitados.
	TMBFTR	Tamaño de los buffers de transmisión.

- RES = + N EXITO, puerto asignado correctamente y el valor de N indica el número de puerto.
- 1 ERROR, al momento de emitir una directiva del sistema o manejador del DR11-B.
 - 2 ERROR, no existen puertos disponibles.
 - 3 ERROR, puerto asignado a otro proceso.
 - 4 ERROR, no hay DDP disponibles.
 - 7 ERROR, no hay suficiente memoria para buffers.

Ejemplo:

```
int res, nupt, nubr, tabr, nubt, tabt;
```

```
-----
nubr = 4;
tabr = 256;
nubt = 2;
tabt = 256;
-----
```

```
res = abrept (&nupt, nubr, tabr, nubt, tabt);
```

Con esta llamada se le solicita al SCT/CV un puerto de transporte, para el cual se requieren 4 buffers de recepción de 256 bytes y 2 buffers de transmisión de 256 bytes. Al terminar la función si todo es correcto, tanto en res como en nupt se regresa el valor del puerto asignado.

```
nupt = 0;
nubr = 0;
```

```
-----
res = abrept (nupt, nubr);
```

Con esta llamada se solicita el puerto cero (puerto del Activador de Procesos), utilizando los valores prestablecidos.

CRRAPT

CIERRA PUERTO

La función de cierre puerto (CRRAPT) es empleada por los procesos de aplicación para solicitarle al SCT/CV que cancele y libere alguno de los puertos de transporte que posee. En la ejecución de esta función el SCT/CV primero cancela todas las solicitudes de envío y recepción que pudiesen estar en espera de ser atendidas en las estructuras del sistema, luego libera la memoria empleada para los buffers y, finalmente, desactiva el puerto dejándolo disponible para cualquier otro proceso. Esta función es invocada con el siguiente formato:

RES = CRRAPT (NUMPTO)

Donde: NUMPTO Es el número de puerto implicado en la operación de cancelación.

- RES = + 1 EXITO, puerto cancelado correctamente.
- 1 ERROR, al momento de emitir una directiva del sistema o manejador del DR11-B.
 - 4 ERROR, operación inválida.
 - 8 ERROR, puerto fuera de rango.

Ejemplo:

```
int res, nupt;
```

```
-----
nupt = 5;
```

```
res = crrapt (nupt);
```

Con esta llamada se le solicita al SCT/CV que cancele el puerto de transporte número 5, el cual debió haber sido adquirido con la función abrept.

ACTREC

ACTIVA RECEPCIONES

La función ACTREC lo utilizan los procesos de aplicación para pedirle al SCT/CV que active la recepción de mensajes por sus puertos. El manejo correcto de los puertos debe seguir la siguiente regla: Utilizar primero la función ABREPT para activar todos los puertos que se requieran durante la operación del proceso, reservando de esta forma los recursos necesarios para su ejecución. Después de esto, y seguros de no requerir más memoria dinámica, utilizar la función ACTREC para activar la recepción de mensajes de los puertos adquiridos. El formato de manejo de esta función es el siguiente:

```
RES = ACTREC (NUMPTO)
```

donde:	NUMPTO	Es el número de puerto implicado en la operación de activación.
	RES = + 1	EXITO, en realización de la operación.
	- 1	ERROR, al momento de emitir una directiva del sistema o manejador del DR11-B.
	- 4	ERROR, operación inválida.
	- 8	ERROR, puerto fuera de rango.

Ejemplo:

```
int res, nupt;
-----
nupt = 5;
res = actrec (nupt);
```

Con esta llamada se le solicita al SCT/CV que active la recepción de mensajes del puerto de transporte 5, el cual debía haber sido obtenido con la función abrept.

FUNCIONES DE MANEJO DE BUFFERS DE COMUNICACION

Este grupo de funciones del SCT/CV permiten un manejo eficiente del espacio de memoria requerida para los buffers de comunicación que se utilizan en la transferencia de mensajes. Estas funciones permiten que los buffers puedan ser compartidos entre el SCT/CV y los procesos de aplicación, además de que proporcionan un manejo sencillo que apoya el trabajo de los diseñadores de procesos de aplicación.

En el manejo de los buffers se emplea un apuntador que identifica su inicio, o sea, desde donde el proceso de aplicación puede comenzar a escribir o a leer un mensaje. Este apuntador es la base tanto de manejo como de identificación del buffer, así que es importante que su valor no sea alterado cuando esté en posesión de los procesos. Si un proceso de aplicación requiere modificar el valor del apuntador para realizar sus actividades, en lugar de hacerlo directamente sobre éste, deberá copiar su valor en una variable auxiliar y con ésta realizar las modificaciones o el procesamiento pertinente.

DAMEBF

DAME BUFFER

Esta función lo utilizan los procesos de aplicación para solicitarle al SCT/CV un buffer para transmitir un mensaje a través de la red. Como resultado de ejecución de esta función el SCT/CV le regresa al proceso un apuntador al inicio del Área de memoria donde se encuentra el buffer. El apuntador es empleado por los procesos para la operación de llenado del buffer con el mensaje que desea transmitir, pero es importante que su valor no se modifique o pierda. El tamaño de los buffers es el estipulado en la apertura del puerto, así que no es mandado en esta función. El formato de esta función es el siguiente:

```
PTRBUF = DAMEBF (NUMPTO)
```

Donde: NUMPTO Es el número de puerto implicado en la operación de solicitud.

PTRBUF Variable utilizada para recibir el valor de apuntador al buffer obtenido, o el valor cero si ya no hay más buffers libres.

Ejemplo:

```
int nupt;
char #pbuf;
-----
nupt = 5;
pbuf = damebf (nupt);
```

Con esta llamada se le solicita al SCT/CV un buffer del puerto de transporte 5, para el envío de un mensaje. Si hay buffers disponibles, al terminar la operación pbuf tiene la dirección del buffer adquirido.

LIBRABF

LIBERA BUFFER

Esta función es empleado por los procesos de aplicación para liberar un buffer de comunicación. Si el buffer es de transmisión, el SCT/CV lo liberará y regresará a la cola de buffers libres pero, si es un buffer de recepción, se cambia su estado a buffer en tránsito y se omite la solicitud correspondiente al MC pero que el buffer sea colocado en la cola correspondiente y considerado en la recepción de nuevos mensajes que lleguen de la línea. El formato de esta función es el siguiente:

```
LDRABF (PTRBUF)
```

Donde: PTRBUF Apuntador al buffer que se libera.

Ejemplo:

```
char #pbuf;
-----
librabf (pbuf);
```

Con esta llamada se le solicita al SCT/CV la liberación de un buffer apuntado por pbuf, el cual puede ser de transmisión o recepción. Si el buffer es de transmisión, pbuf debió de haberse obtenido con la función DAMEBF.

FUNCIONES DE MANEJO DE CONEXIONES DE TRANSPORTE

Para el manejo del servicio de comunicación fin-a-fin, de circuitos virtuales del SCT/CV se proporcionan cuatro funciones que permiten establecer, mantener y terminar trayectorias confiables de comunicación entre cualquier par de procesos de aplicación. Este mecanismo de enlace se basa en los identificadores obtenidos por los procesos a través de los puertos de transporte que se les asignan.

El mecanismo de conexión del SCT/CV para las computadoras FDP permiten utilizar un puerto activo en diferentes conexiones. O sea, un proceso puede activar un puerto, conectarse con un proceso a través de él, terminar la conexión, y volver a conectarse con otro proceso utilizando el mismo puerto, todo sin ningún conflicto.

Para el establecimiento de una conexión entre un par de procesos se utilizan dos funciones del SCT/CV que deben ser manejadas en forma coordinada por los procesos de aplicación, las cuales son la función de traspasar conexión (CONNECT) y la de espera por conexión (ESPCON).

Cuando dos procesos establecen un enlace de comunicación, el SCT/CV les asegura un servicio de comunicación confiable y eficiente entre ellos. Bajo ciertas circunstancias, tales como un mal funcionamiento del CCD, desconexión de algún cable, infiltración destino fuera de operación, o en general alguna falla transitoria o permanente cuya corrección esté fuera del alcance del SCT/CV, éste no podrá resolverlo y por lo tanto le avisará al proceso de aplicación, para que tome las medidas pertinentes. Después de la corrección del problema o, simplemente a criterio del proceso de aplicación, se puede tratar de reestablecer la conexión utilizando la función REINI.

CONNECT

ESTABLECER CONEXION

Con la función de establecer una conexión (CONNECT), los procesos de aplicación le solicitan al SCT/CV la activación de un enlace con el proceso de aplicación descrito en los parámetros de la función. Cuando el SCT/CV recibe esta solicitud establece el estado adecuado para el puerto local e inicia la transacción de establecimiento, de acuerdo al protocolo especificado para el sistema. Al terminar la transacción se le informa al proceso del resultado de la misma. Para que esta función pueda tener éxito, es indispensable que el destinatario identificado por ESTDST y PTODST pertenezca a un proceso activo que haya emitido una función de espera por conexión (ESPCON), ya que, si el proceso destino no está activo o no espera por la conexión que se solicita, el proceso de establecimiento fallará. El formato de esta función es el siguiente:

RES = CONNECT (NUMPTO,ESTDST,PTODST)

Donde:	NUMPTO	Es el número de puerto implicado en la operación de conexión.
	ESTDST	Dirección de la estación destino que se desea enlazar.
	PTODST	Dirección del puerto de transporte del proceso destino a ser enlazado.
	RES = + 1	EXITO, en realización de la operación.
	- 1	ERROR, al momento de emitir una directiva del sistema o manejador del DR11-B.
	- 4	ERROR, operación inválida.
	- 8	ERROR, puerto fuera de rango.

-10xx ERROR, conexión no pudo ser establecida, y el valor de xx proporciona lo causa.

- xx = 01 Puerto destino rechaza la solicitud.
- 02 Puerto destino no espera conexión.
- 03 Puerto destino conectado o tramitando otra conexión.
- 04 Estación destino no activa.

Ejemplo:

```
int res, nupt, estds, ptods;
-----
nupt = 5;
estds = 14;
ptods = 6;
-----
res = conect (nupt,estds,ptods);
```

Con este llamado se le solicita al SCT/CV que establezca un enlace del puerto local de transporte 5 con el puerto 6 de la estación remota 14. El puerto local 5 debió haberse adquirido utilizando la función de apertura de puerto abrept.

ESPERAR POR CONEXION

Con la función de espera por conexión (ESPCON), los procesos le solicitan al SCT/CV el establecimiento de un enlace con algún proceso remoto. Cuando el SCT/CV recibe esta solicitud establece el estado adecuado para el procedimiento de conexión, pero en lugar de iniciar el trámite de conexión como en el caso anterior, se espera a recibir una solicitud de conexión. Al recibir la solicitud, el SCT/CV le valida de acuerdo a los parámetros de la función ESPCON, y sólo acepta el enlace si concuerda el chequeo de estaciones y procesos remotos contra locales. Cuando un proceso desea esperar por una solicitud de conexión que no tiene identificada, como podría ser el caso del Activador de Proceso, se pone la combinación de valores de cero para ESTDST y PTODST. Con esto el SCT/CV interpretará una recepción abierta, aceptando cualquier solicitud de conexión que se reciba para el puerto indicado sin importar de quien venga. En esta función es posible prefiar el tiempo máximo que se desea esperar por la llegada de una solicitud de conexión con el empleo del parámetro THPESP, después del cual si no se recibe alguna solicitud la operación termina y se le regresa el control al proceso. Al terminar la ejecución de esta función, el SCT/CV le informa al proceso de aplicación el resultado obtenido. El formato de esta función es el siguiente:

RFS = ESPCON (NUMPTO,ESTDST,PTODST,THPESP)

Donde: NUMPTO Es el número de puerto implicado en la operación de conexión.
ESTDST Dirección de la estación destino que se desea enlazar.

PTODS1 Dirección del puerto de transporte del proceso destino a ser enlazado.

IMPESP Tiempo máximo que será empleado para esperar la recepción de la solicitud de conexión. Si este valor es cero no hay límite de espera. El rango de valores que puede manejar esta variable es de 0 a 32,767 segundos.

RES = + 1 EXITO, en realización de la operación.

- 1 ERROR, al momento de emitir una directiva del sistema o manejador del DR11-B.

- 4 ERROR, operación inválida.

- 8 ERROR, puerto fuera de rango.

-10 ERROR, conexión no pudo ser establecida, se terminó el tiempo de espera.

Ejemplo:

```
int res, nupt, estds, ptods, tesp;
-----
nupt = 6;
estds = 13;
ptods = 5;
tesp = 0;
-----
res = espcon (nupt,estds,ptods,tesp);
```

Con esta llamada se le solicita al SCT/CV la espera de una conexión dirigida al puerto local 6, que venga del puerto 5 de la estación remota 13. El valor `tesp = 0`, desactiva el tiempo de espera, así que la espera de la solicitud de conexión es hasta que llegue, tarde lo que tarde. Al llegar una solicitud que cumpla con los parámetros especificados, el SCT/CV le avisa al proceso a través de `res`, y le regresa el control.

```
-----
nupt = 0;
estds = 0;
ptods = 0;
tesp = 60;
-----
```

```
res = espcon (nupt,estds,ptods,tesp);
```

En este caso, la solicitud de espera por conexión del puerto local cero, probablemente es emitida por el Activador de Procesos. La combinación `estds = ptods = 0`, le pide al SCT/CV que acepte la primer solicitud de conexión que se recibe para el puerto 0. Si la solicitud no llega en un plazo de 60 segundos, el SCT/CV termina la operación, registra su resultado en `res` y le regresa el control al proceso de aplicación.

DESCON

TERMINAR CONEXION

Cuando los procesos concluyen las actividades relacionadas con un enlace, deben liberar su conexión, para lo cual el SCT/CV provee la función de desconexión o terminar conexión (DESCON). Con esta función, los procesos pueden liberar a un puerto de una conexión, para luego, si es necesario, utilizarlo en una nueva. La cancelación de un enlace debe realizarse en ambos extremos de la conexión, por lo que la función debe ser emitida sobre los dos puertos enlazados. El formato de esta función es el siguiente:

RES = DESCON (NUMPTO)

Donde: NUMPTO Es el número de puerto implicado en la operación.

VAR = + 1 EXITO, en realización de la operación.
 - 4 ERROR, operación inválida.
 - 8 ERROR, puerto fuera de rango.

Ejemplo:

```
int res, nupt;
-----
nupt = 5;
-----
res = descon (nupt);
```

Con esta llamada se le solicita al SCT/CV que cierre la conexión en que participa el puerto local número 5.

REINI

REINICIALIZAR CONEXION

La función de reinicialización de una conexión (REINI) es utilizada por los procesos de aplicación para solicitarle al SCT/CV que reinicie un enlace, de acuerdo a las características de su conexión inicial. Esta función es emitida cuando el SCT/CV no puede corregir un problema por estar fuera de su alcance, y por lo tanto entra en un estado de error. La función puede ser invocada de cualquier lado del enlace o de ambos lados. Si el problema que dio origen a la falla no es corregido antes de emitir la función de REINI, lo más seguro es que no se logre la reinicialización. El formato de esta función es el siguiente:

RES = REINI (NUMPTO)

Donde: NUMPTO Es el número de puerto implicado en la operación.

VAR = + 1 EXITO, en realización de la operación.
 - 4 ERROR, operación inválida.
 - 8 ERROR, puerto fuera de rango.
 - 9 ERROR, Operación no concretada exitosamente por problemas operativos en la subred.

Ejemplo:

```
int res, nupt;

nupt = 5;
res = reini (nupt);
```

Con esta llamada se le solicita al SCT/CV que reinicialice la conexión en que participa el puerto local 5.

FUNCIONES DE MANEJO DE MENSAJES DE COMUNICACION

Este grupo de funciones contiene a los primitivos que se utilizan para el intercambio de mensajes entre los procesos a través de las conexiones establecidas por el SCT/CV. Estas primitivas permiten enviar y recibir mensajes en una forma ordenada, secuencial y sin pérdidas.

El envío de mensajes proporcionado por la función ENVMSJ, se realiza de acuerdo a los mecanismos de control de flujo, secuenciamento y corrección de errores definidos por el protocolo del SCT/CV. En el caso particular de esta implantación se han adicionado los modos de operación de los envíos, que son el síncrono y el asíncrono, con los cuales se le proporciona una mayor flexibilidad al usuario. Además se ha adicionado el manejo de la función de ESPENV, que permite coordinar el manejo de envíos asíncronos.

Cuando un proceso activa la recepción de mensajes de un puerto, todos los buffers de recepción de éste son involucrados en solicitudes de recepción, colocándose en la cola del puerto correspondiente en el Manejador de Comunicaciones de la Red (MC), en espera de recibir un mensaje. Cuando se recibe un mensaje correcto, el SCT/CV lo relaciona con una de las solicitudes del puerto correspondiente, y lo enlaza en la cola de mensajes recibidos del DMP correspondiente. Cada nueva recepción siempre es instalada en la cola con una organización FIFO en base a los números de secuencia que contienen los mensajes. En esta cola, los mensajes esperan en orden a ser tomados por el proceso de aplicación, para lo cual se utiliza la función de recepción de mensajes (RCBMSJ).

ENVMSJ

ENVIA MENSAJE

La función de envío de mensajes (ENVMSJ) es utilizada por los procesos de aplicación para transmitir mensajes a través de las conexiones establecidas. Esta primitiva puede operar en modo síncrono o asíncrono. En el modo síncrono cuando el proceso envía lo llorando, queda suspendido hasta que se obtiene el reconocimiento del mensaje, aunque esto implique varias retransmisiones, o simplemente hasta obtener el resultado de la operación, el cual es registrado en RES, antes de regresarle el control al proceso. En el caso asíncrono, el SCT/CV sólo toma la solicitud y valida los parámetros. Si todo está bien le regresa el control al proceso para que pueda continuar procesando, mientras que él se encarga de transmitir el mensaje, o los mensajes que estén en cola de envío del puerto. Con el parámetro PRIENV, se informa al SCT/CV la prioridad de transmisión del mensaje. Si la prioridad es normal, el mensaje se transmite de acuerdo al servicio de circuitos virtuales proporcionado por el protocolo. Pero, si la prioridad es urgente, el mensaje se transmite sin pasar por los mecanismos de secuencia, tratándose de enviar y entregar lo más pronto posible. El formato de esta función es el siguiente:

RES = ENVMSJ (NUMPTD, PTRBUF, NUMBYT, MODO, PRIENV)

Donde:	NUMPTD	Es el número de puerto implicado en la operación.
	PTRBUF	Apuntador a dirección del buffer que contiene el mensaje a ser enviado.
	NUMBYT	Número de bytes del mensaje a transmitir.
	MODO	Tipo de envío que se solicita, que puede ser

SINCRONO = 1 o ASINCRONO = 0.

PRIENV Prioridad de manejo en el envío del mensaje, que pueda ser URGENTE = 1, o NORMAL = 0.

RES = + 1 EXITO, un realización de la operación.
 - 1 ERROR, al momento de emitir una directivo del sistema o manejador del DR11-B.
 - 4 ERROR, operación inválida.
 - 8 ERROR, puerta fuera de rango.
 -11 ERROR, número de retransmisiones agotado.

Ejemplo:

```
int res, nupt, nuby, modo, penv;
char #pbuf;
```

```
-----
nupt = 5;
nuby = 25;
modo = 1;
penv = 0;
```

```
-----
res = envmsj (nupt,pbuf,nuby,modo,penv);
```

Con esta llamada se le solicita al SCT/CV la realización del envío de un mensaje de 25 bytes a través de la conexión manejada por el puerto local de transporte número 5. pbuf es el apuntador a un buffer que debe haber sido obtenida a través de la función dasebuf y llenada con el mensaje a transmitir antes de invocar esta función. El envío se realiza en modo síncrono y con prioridad normal.

ESPENV**ESPERA POR ENVIO**

En un envío síncrono, cuando el SCT/CV le regrese el control al proceso, si todo estuvo bien, la información del buffer ya ha sido transmitida y, por lo tanto, ya se puede recuperar el buffer para un nuevo mensaje. En el caso del envío asíncrono, el proceso tiene la posibilidad de estar conformando nuevos mensajes mientras los previos son transmitidos por el SCT/CV. El problema de esto consiste en como saber cuando ya ha sido transmitido el mensaje de determinada buffer, y por lo tanto, éste ya puede ser reutilizado. Para resolver este problema el SCT/CV ha dispuesto la función de espera por envío (ESPENV), con la cual el proceso obtiene tanta información de la condición de transmisión de los mensajes, así como de los buffers que puede reutilizar.

Para el manejo adecuado de las funciones de envío asíncrono y el espera por envío, es importante que el proceso lleve el control del orden en el cual emite sus envíos, y de los buffers que se relacionan con cada uno de ellos. En la emisión de envíos asíncronos, cuando al proceso se le agotan todos los buffers disponibles y lo juzgo conveniente, debe emitir un espera por envío, el cual opera de la siguiente forma. Si al regresarle el control al proceso después de ejecutar la función, el valor de RES es positivo, éste valor indica el número de envíos que se pueden realizar, considerando que los buffer se van liberando en el mismo orden en que se emitieron sus solicitudes de envío. Si el valor de RES es negativo, éste valor es el código de error de un problema de operación que se suscito, y PTRBUF es el

apuntador al primer buffer que ya no pudo ser enviado, así que después de la recuperación del problema, lo que debe hacer el proceso es reenviar los mensajes contenidos en los buffers dados a partir del indicado por PTRBUF. El formato de esta función es el siguiente:

RES = ESPENV (NUMPTO, PTRBUF)

Donde:

NUMPTO	Es el número de puerto implicado en la operación.
PTRBUF	Apuntador a dirección del buffer que contiene el mensaje a ser reenviado, en caso de error.
RES = + N	EXITO, N implica el número de envíos que se pueden realizar al terminar la función
- 1	ERROR, al momento de emitir una directiva del sistema o manejador del DR11-B.
- 4	ERROR, operación inválida.
- 8	ERROR, puerto fuera de rango.
-11	ERROR, número de retransmisiones agotado.

Ejemplo:

```
int res, nupt;
char *pbuf;
-----
nupt = 3;
-----
res = espenv (nupt, pbuf);
```

Al terminar la función si res es positivo, indica el número de buffers disponibles y por lo tanto el número de envíos que se pueden realizar. El control sobre los buffers lo debe llevar el proceso para saber cuales son los que puede utilizar. La atención de las solicitudes es secuencial.

RECIBE MENSAJE

Para que un proceso pueda obtener los mensajes recibidos o esperar por alguno, utiliza la función de recepción de mensajes (RCBHSJ). Esta función puede operar en modo síncrono o asíncrono. En el modo síncrono cuando el proceso emite la función, queda suspendido hasta que el SCT/CV tiene algo para él. Si al momento de emitir la función existe por lo menos un mensaje en cola, el SCT/CV se lo pasa inmediatamente al proceso y éste continúa procesando. En el caso asíncrono, si al momento de emitir la función de recepción no hay mensajes recibidos, se le informa al proceso y se le regresa el control. Estos dos modos de operación permiten una gran flexibilidad para el manejo de recepciones. Cuando se emite una solicitud de recepción síncrona, se puede indicar en la variable IMPESP el tiempo máximo que el proceso desea esperar por la recepción del mensaje. Si el tiempo de espera termina antes que la recepción del mensaje, se le regresa el control al proceso y se le informa lo sucedido. En esta función también se puede especificar el manejo de una rutina de acción (RUTACC), la cual es invocada por el SCT/CV cuando recibe un mensaje. Una vez que termina la rutina de acción y el SCT/CV obtiene nuevamente el control, se lo regresa al proceso en la instrucción siguiente a la llamada de RCBHSJ. Una rutina de acción es un código que toma acciones sobre la recepción de mensajes, el cual debe ser dispuesto por cada proceso de aplicación de acuerdo a sus requerimientos. Cuando el SCT/CV le pasa el control a la rutina de acción ya son válidos los valores de los parámetros de la llamada RCBHSJ. El formato de esta función del SCT/CV es el siguiente:

RES = RCBMSJ (NUMPTD, IPTRBUF, INUMBYT, IRUTACC, MODO, TNPESP)

Donde:

NUMPTD	Es el número de puerto implicado en la operación.
IPTRBUF	Dirección donde el SCT/CV regresará la dirección de inicio del buffer que contiene el mensaje recibido.
INUMBYT	Dirección donde el SCT/CV regresará el número de bytes del mensaje recibido.
IRUTACC	Dirección de la rutina de acción que debe ser llamada al recibir el mensaje. Si el proceso no desea manejarlo llena este campo con el valor de cero.
MODO	Tipo de recepción solicitada, que puede ser SINCRONO = 1 o ASINCRONO = 0.
TNPESP	Si MODO = SINCRONO, este campo es válido y representa el tiempo máximo de espera antes de abortar la recepción. Si TNPESP = 0, no se activa el tiempo de espera, por lo tanto la espera es hasta que llegue el mensaje. El máximo valor permisible es 32,767 seg.

VAR = + 1 EXITO, en realización de la operación.
 - 4 ERROR, operación inválida.
 - 3 ERROR, mensaje no recibido.
 - 8 ERROR, puerto fuera de rango.

Ejemplo:

```
int res, nupt, nuby, modo, tresp;
char *pbuf;

-----

nupt = 3;
modo = 1;
tresp = 60;

-----

res = rcbmsj (nupt, pbuf, nuby, 0, modo, tresp);
```

Con esta llamada se solicita al SCT/CV la recepción de un mensaje por la conexión del puerto de transporte número 5, en modo síncrono y con un tiempo de espera de 60 segundos. Al terminar la llamada si se recibió un mensaje, pbuf es el apuntador al inicio del buffer que lo contiene y nuby es el número de bytes del mensaje. En esta llamada no se solicita el manejo de una rutina de acción.

FUNCIONES DE INFORMACION DE ESTADO

El último grupo de funciones de la interfaz del SCT/CV está constituido por una sola función que se utiliza para obtener información sobre el estado de operación de los puertos.

EDOPT

ESTADO DE PUERTO

Con la función de estado de puerto (EDOPT), los procesos de aplicación le piden al SCT/CV que les proporcione la información de estado concerniente a algún puerto en particular. Para la operación de esta función, el usuario debe disponer un buffer de 18 bytes, denominado Buffer de Estado, en el cual el SCT/CV le regrese la información solicitada. En el capítulo anterior se proporcionó la información completa sobre la descripción del buffer de estado, y el formato de la función es el siguiente:

RES = EDOPT (NUMPTO, PTRBUF)

Donde:

NUMPTO	Es el número de puerto implicado en la operación.
PTRBUF	Es el apuntador o dirección de un buffer de 18 bytes donde el SCT/CV pose la información de estado del puerto.
RES = + 1	EXITO, en realización de la operación.
- 4	ERROR, operación inválida.
- 8	ERROR, puerto fuera de rango.

Ejemplo:

```
struct stat {
```

```
char  etio,
      pti0,
      nubi,
      nubi,
      tlib,
      rlib,
      tsar,
      rsar,
      tcap,
      rcap,
      ttrn,
```

```

ptrn;
votr;
verc;
strm;
ptrm;
edpt;
cnpt;
) bufedo;
-----
main ()
<
-----
int res, nupt;
-----
nupt = 3;
-----
res = edapt (nupt, &bufedo, &lla);

```

Con esta llamada se le solicita al SCT/CV que proporcione la información de estado correspondiente al puerto local de transporte número 5. Después de la llamada, si todo es correcto en bufedo.xxxx, se pueden obtener los valores de los parámetros manejados en el buffer de estado. En este caso xxxx corresponde a cualquiera de los campus definidos en la estructura (etlo, ptlo,, edpt, cnpt).

DEFINICIONES DE USUARIO

La operación adecuada del SCT/CV requiere que el usuario incluya en su código las definiciones de operación del sistema. Estas definiciones se encuentran en el archivo de la red denominado defred.sct y deben ser ligadas con los procesos de aplicación y la BF con la utilidad IKB. Con la inclusión del archivo defred.sct, el SCT/CV opera de acuerdo a lo establecido en su diseño, pero ésta no es rígida ya que el usuario puede generar su propio archivo (pej. defusr.sct), de acuerdo a sus intereses muy particulares y ligarlo con su proceso y la BF.

El archivo que posee la BF se llama redlib.obl y se encuentra al igual que el archivo de definiciones defred.sct, en lo cuento [1,54] del disco del sistema. Las definiciones contenidas en el archivo defred.sct son:

ST.MRE == 3.	Número máximo de retransmisiones de envío.
ST.TOE == 2.	Time-Out en seg. para reenvío de Mensajes.
ST.MRC == 3.	Número máximo de retransmisiones para demás funciones.
ST.TOC == 2.	Time-Out en seg. para reenvío de demás funciones.
ST.ETL == 8.	Dirección Física (CCD) de la estación local.
ST.PTI == 0.	Puerto inicial.
ST.PTF == 7.	Puerto final.
ST.PTH == 3.	Número máximo de puertos por proceso.
ST.THF == 22.	Tamaño de encabezado de mensajes de transporte.
ST.NBR == 2.	Número de buffers para recepción.
ST.TBR == 512 + ST.THT	Tam. de buffers para recepción.
ST.NBT == 2.	Número de buffers para envío.
ST.TBT == 512 + ST.THT	Tam. de buffers de envío.

Gua del Usuario

ST.VCO == 7 | Valor máximo de la ventana de secuencia.

Para un mayor detalle de la forma de ligar y construir términos bajo
RSX-11M, consultar el manual del IKB del sistema.

CAPITULO VII

Conclusiones

El protocolo del SCT/CV ha sido diseñado para cumplir con los objetivos planteados para la Capa de Transporte de la Arquitectura de Red-IIMAS de Alta Velocidad. Su implantación se ha realizado bajo el sistema operativo RSX-11M de Digital para PDP-11s. Bajo estas condiciones, el SCT/CV ha obtenido una serie de características relevantes que lo distinguen, y que forman parte de los resultados obtenidos en su desarrollo.

En este capítulo se proporcionan las características principales del SCT/CV, se describen algunos resultados adicionales y se termina con la enumeración de algunas recomendaciones de modificación, para ser consideradas en alguna fase posterior de su desarrollo.

CARACTERISTICAS PRINCIPALES DEL SCT/CV

PROTOCOLO DE CIRCUITOS VIRTUALES. El SCT/CV es un sistema de transporte que se basa en un protocolo que le permite ofrecer un servicio fin-a-fin de comunicación virtual libre de errores, de mensajes de tamaño variable, confiable y eficiente, que libera a los procesos de la Capa de Aplicaciones del manejo de aspectos de comunicación, confinándolos únicamente al manejo de sus propios protocolos.

El protocolo del SCT/CV permite aprovechar las características importantes del diseño de las capas inferiores y se avoca a la corrección de problemas de niveles jerárquicos superiores.

ESQUEMA DE ASIGNACION DINAMICA DE PUERTOS. El SCT/CV proporciona un esquema de manejo de direcciones sencillo, práctico y preciso, que responde a las necesidades de los procesos de la Capa de Aplicaciones y proporciona las siguientes ventajas:

o **Múltiples Instancias por Servicio.** Al no manejarse un esquema en el cual los servicios se ofrecen sobre direcciones fijas, permite la existencia de múltiples instancias de cualquier proceso de la Capa de Aplicaciones.

o **Reducción de Sobrecarga para Anfitriones.** Cualquier implantación de una red de computadoras requiere de tiempo de procesamiento de los anfitriones. En el caso del SCT/CV, al eliminar la necesidad de existencia de servidores de red siempre activos en espera de solicitudes de servicio, se reduce la sobrecarga impuesta a los anfitriones.

o **Manejo Justo de Recursos.** En el SCT/CV, fuera del puerto 0 que se encuentra reservado permanentemente para el AP (Activador de Procesos), ningún otro proceso debe mantener recursos de transporte reservados cuando no estén en uso. De esta forma, al existir puertos (recursos) disponibles, éstos pueden ser utilizados por quien los necesite. Esto permite un manejo justo de los recursos de comunicación y memoria.

Conclusiones

ESTRUCTURA DE IMPLANTACION. El hablar de transportabilidad para este tipo de sistemas es muy relativo, debido a su alta dependencia tanto de la computadora como de su sistema operativo. En el caso del SCT/CV no podría hablarse de transportabilidad, pero si de generalidad si nos remitimos a su estructura de implantación. En este sentido, se puede decir que la estructura de implantación del SCT/CV, proporciona un buen marco a seguir para la implantación de sistemas de transporte.

MANEJO DE BUFFERS DE USUARIO. El SCT/CV proporciona un conjunto de funciones que le permiten realizar la asignación y manejo de los buffers para envío y recepción de mensajes. Al administrar el SCT/CV los buffers de los procesos, les facilita a los diseñadores de aplicaciones su manejo, se elimina la necesidad de que el SCT/CV requiera buffers propios y se permite la programación de transferencias de información directas entre las Capas de Comunicación de Datos y Aplicaciones, aprovechándose el esquema de DMA nativo. Gracias a esto, se ha logrado una reducción sustancial del tiempo de respuesta del sistema, se ha economizado una buena cantidad de semario y tiempo de procesamiento en los anfitriones y, en cierta medida se ha reducido la complejidad del SCT/CV.

CANCELACION AUTOMATICA DE PUERTOS. El SCT/CV ha aprovechado su estructura de implantación y los característicos de RSX-11M, para poder cancelar en forma sencilla y directa, los puertos que se quedan abiertos cuando los procesos terminan sin liberarlos. El esquema utilizado es bastante sencilla y libera a los anfitriones

Conclusiones

de una sobrecarga en procesamiento.

COMPACTO. El SCT/CV es un sistema compacto que requiere 4 kw de espacio por cada tarea de aplicación y 3 Kw por cada anfitrión. Al ser el SCT/CV un sistema compacto permite que en cada anfitrión se disponga de más espacio para los procesos de aplicación, así como para otras tareas.

INTERFAZ AMISTOSA. El SCT/CV proporciona una interfaz consistente en un conjunto de funciones de fácil manejo y comprensión. Esta interfaz es accesible a los procesos de la Capa de Aplicaciones a través de código desarrollado en el lenguaje 'C'. Una interfaz de este tipo facilita y agiliza la comunicación con la Capa de Aplicaciones, y se debe gracias a que:

o Posee un conjunto amplio y suficiente de funciones, que permiten ejercer un manejo de red suficiente y flexible.

o Posee funciones para el manejo de buffers de envío y recepción de mensajes.

o Permite el manejo de modos de operación síncrono y asíncrono.

o Permite el manejo de prioridades de envío de mensajes normal y urgente.

o Permite la activación de rutinas de acción.

o Permite el manejo de tiempos de espera programables.

Conclusiones

RESULTADOS ADICIONALES

Además de haber logrado el sistema requerido para la Capa de Transporte de la Arquitectura de Red-IIMAS, se han obtenido los siguientes:

- o Generación de infraestructura tecnológica. El SCT/CV es un sistema abierto que proporciona una infraestructura para el desarrollo y experimentación sobre nuevos y mejores protocolos de aplicación.
- o Gracias al conjunto de funciones de bajo nivel del SCT/CV implantadas en su MC (núcleo de transporte) se posee una infraestructura muy valiosa que permite la experimentación, de cambios en el protocolo del mismo SCT/CV, o de nuevos protocolos de transporte.

Conclusiones

RECOMENDACIONES

Aunque el SCT/CV posee características importantes, se han visto algunas posibles modificaciones, que pueden ser consideradas para una siguiente versión, las cuales pretenden robustecerlo y darle una mayor complejidad. Las modificaciones propuestas serían:

- o Agregar funciones de monitoreo y control de red, las cuales deberán ser accesibles sólo desde un puerto en particular que se designe para el sistema que lo implante.
- o Agregar funciones estructuradas de envío y recepción de mensajes, que les faciliten a los procesos de aplicación el llenado y vaciado de los buffers de mensajes.
- o Adicionarle al protocolo del SCT/CV características especiales que le permitan una mayor confiabilidad en un ambiente de redes interconectadas, donde no se posean condiciones de comunicación tan controladas como en Red-IIMAS.
- o Diseñar un sistema configurador de estaciones de transporte, el cual le permita al operador de la red realizar una fácil y rápida configuración del SCT/CV. Para esto, el SCT/CV debe ser capaz de tomar su configuración operativa de un archivo que genere el sistema configurador.
- o Pasar actividades realizadas actualmente en el módulo de la Biblioteca de Funciones al Manejador de Comunicaciones de Transporte. Esto tiene el propósito de reducir los requerimientos de memoria en el espacio destinada a los procesos de aplicación.

APENDICE A

PRIMITIVAS DE ACCESO DE LA CAPA DE COMUNICACION DE DATOS

La interfaz de acceso a la capa de comunicación de datos se ha definido a través del manejo de cuatro primitivos que permiten el envío y recepción de mensajes de tamaño variable, con un tamaño máximo de 1 Kbyte. Para el enlace y comunicación de la PDP con el CCD donde reside la capa de comunicación de datos, se ha utilizado una tarjeta interfaz conocida como DR11-B. El DR11-B es una interfaz de DMA (Acceso Directo a Memoria) que opera en base a cuatro registros de 16 bits y proporciona un buen esquema de entrada/salida para el manejo de fuertes volúmenes de información. Utilizando el DR11-B se realizó la conexión requerida con el CCD y se permitió la definición interna de manejo de la interfaz con la capa de comunicación de datos a través de las cuatro primitivas que se describen a continuación:

ENVIA (Estación-Destino,Puerto-Destino,Dir-Mensaje,Tam-Mensaje)

Esta es una primitiva controlada por la capa de transporte, a través de la cual, el SCT/CV le solicita a la capa de comunicación de datos el envío del mensaje dado por Dir-Mensaje y Tam-Mensaje, con el destinatario Estación-Destino Puerto-Destino. Los parámetros de dirección y tamaño del mensaje son utilizados para la programación del DR11-B para la transferencia entre el anfitrión y el CCD, mientras que los de destinatario son utilizados por la subred y el sistema de

transporte de la estación remota, para la realización de las funciones de ruteo, multiplexaje y entrega, correspondientes.

RECIBE (Puerto-Local,Dir-Buffer,Tam-Mensaje)

Esta es una primitiva controlada por la capa de transporte, en la cual, el SCT/CV le solicita a la capa de comunicación de datos que le transfiera el mensaje que se encuentre al inicio de su cola de espera, y que posee el valor de Puerto-Local como destinatario. Los parámetros de Dir-Buffer y Tam-Mensaje se utilizan para programar el DR11-B para la transferencia del mensaje desde el CCD al anfitrión, y poder cargarlo directamente en uno de los buffers del proceso de aplicación involucrado. El tamaño del mensaje lo obtiene el SCT/CV a través de la primitiva de informe que se describe más adelante.

ELIMINA (Puerto-Local,Tam-Mensaje)

Esta es una primitiva controlada por la capa de transporte, con la cual, el SCT/CV le instruye a la capa de comunicación de datos que elimine el mensaje que se encuentra al inicio de su cola de espera, debido a que no lo puede recibir por no existir alguna solicitud que lo requiera, ya sea porque el Puerto-Local especificado en el mensaje no está activo, o simplemente porque el proceso de aplicación que posee el Puerto-Local no ha dispuesto buffers para su recepción. Después de que la capa de comunicación de datos realiza la eliminación solicitada, debe de actualizar la cola, ya sea que existan o no más mensajes en espera.

INFORMA (Puerto-Local,Tar-Mensaje)

Esta es una primitiva controlada por la capa de comunicación de datos, con la cual, esta capa le informa al SCT/CV cual es el mensaje que se encuentra al inicio de la cola de recepciones que maneja. El propósito de esta primitiva, como su nombre lo refiere, es la de informarle al SCT/CV con Puerto-Local y Tar-Mensaje, de quien es el mensaje que posee y cual es su tamaño, con lo cual, el SCT/CV pueda determinar si pide su recepción o eliminación. En el caso de que el SCT/CV pida la eliminación del mensaje, la capa de comunicación debe realizar la correspondiente y luego proceder a revisar si existen más mensajes en la cola. Si existen más mensajes, utilizando nuevamente la primitiva de INFORMA, se le notifica al SCT/CV sobre el mensaje que ahora se encuentra al inicio de la cola. En el caso de ya no existir más mensajes, la capa de comunicación de datos se espera hasta recibir uno nuevo.

APENDICE B

CODIGOS DE RESPUESTA DE LAS FUNCIONES DE BAJO NIVEL DEL SCT/CV

CODIGO	REPRESENTACION	DESCRIPCION
1	IS.SUC	Exito en Operación
- 2	IE.IFC	Función Ilegal
- 3	IE.DNR	Error en Tarjeta del DR11-B
- 4	IE.VER	Error en Transferencia con CCD
- 6	IE.SPC	Direccionamiento Ilegal de Memoria
- 7	IE.DNA	Puerto No Activo
- 8	IE.DAA	Puerto Asignado a Otra Tarea
- 15	IE.ABO	Operación Abortada
- 65	IE.OFL	CCD Fuera de Línea
- 74	IE.TMO	TMO Transcurrido
- 84	IE.ALC	No Hay Puertos Disponibles

GLOSARIO

ANCHO DE BANDA. Capacidad de una canal de comunicaciones para transmitir información.

AMFITRION. Computadora enlazada a una red, la cual proporciona un medio de entrada a la misma o bien provee a la red con servicios como manejo de bases de datos, ejecución de programas o lenguajes especiales, manejo de periféricos, procesamiento de datos, etc.

BANDA ANCHA. Técnica de transmisión, en la cual se aplica la información al cable como una señal analógica. Es decir, la señal se modula en radio frecuencia (RF) para poderla transmitir.

BANDA BASE. Técnica de transmisión de señal sin modulación. La señal digital (unos y ceros) es aplicada directamente al cable como pulsos de voltaje o corriente.

BUFFER. Almacenamiento temporal utilizado para compensar las diferencias de velocidad, o tiempo de ocurrencia de eventos, cuando se transfieren datos de un lugar a otro. Este almacenamiento también es utilizado para fines de revisión de datos, validación, corrección, transformación, etc.

CIRCUITO VIRTUAL. Trayectoria de comunicación de existencia conceptual o aparente establecida y manejada mediante tablas de conexión. Un circuito virtual no realiza un uso exclusivo de los recursos existentes y solo existe mientras se está utilizando para la

transferencia de datos.

COLIBION. Evento que ocurre cuando dos o mas paquetes son transmitidos al mismo tiempo sobre un medio compartido, y del cual resulta la incapacidad de reconocer los paquetes.

COMPUERTA. Nodo común a dos o mas redes, el cual tiene la función de realizar la transformación de protocolos entre las redes conectadas. La compuerta además, también puede realizar funciones de control de flujo, reformato de datos y control de errores entre las redes.

CONEXION. Enlace físico o virtual entre un par de nodos, que es establecido a través de un procedimiento especial, con la finalidad de proporcionar una vía de comunicación adecuada para los nodos.

CONMUTACION DE CIRCUITOS. Técnica de conmutación de una red, en la cual, mediante un procedimiento de enlace se establecen trayectorias físicas dedicadas entre los nodos que se comunican.

CONTENCION. Condición que ocurre cuando mas de una estación intentan apoderarse del mismo canal al mismo tiempo.

DATABRAMA. Paquete autocontenido e independiente de otros paquetes, el cual no requiere reconocimiento alguno, y posee la información suficiente para su ruteo desde el nodo origen al nodo destino.

DERIVADOR. Dispositivo que permite enviar y recibir información hacia y desde el canal de comunicación de una red. Estos son especialmente utilizados en las redes tipo bus.

DIFFUSION. Proceso de transmitir información sobre un medio de

comunicación compartido, de tal forma que todas las estaciones lo escuchan. En este caso aunque la información queda al alcance de todas las estaciones, sólo es tomada por la estación destino.

ESTAFETA. Conjunto de "n" bits que se utiliza en las redes con topologías de bus y anillo, para el control de acceso al medio compartido. Este patrón es reconocido como el permiso para transmitir.

ENCRIPCION. Técnica de conversión de datos fuentes en una forma no entendible, a partir de la cual, aplicando el proceso inverso, es posible recuperar los datos fuentes.

ESTACION. Punto final lógico de una red que puede ser una terminal, un procesador (anfitrión), o un programa de aplicación.

GUARDA-REEXPIDE. Método de transmisión de información a través de una serie de nodos, donde cada uno de éstos recibe la información, la almacena y posteriormente la retransmite hacia el siguiente nodo.

INTERFAZ. Límite que existe entre dos capas adyacentes de una red, que define los primitivos de interacción y los servicios que la capa inferior le ofrece a la superior.

MARCO. Unidad de transferencia empleado por la capa de enlace de datos, de acuerdo a la definición proporcionada por el modelo ISO de referencia.

MEDIO DE TRANSMISION. Trayectoria física entre transmisores y receptores en una red de comunicaciones.

MENSAJE. Conjunto de bytes que se maneja como unidad de transmisión entre las capas de alto nivel de la arquitectura de una red de

computadoras. Normalmente un mensaje de una capa está formado por un encabezado y una parte de datos, donde, la parte de datos a su vez puede consistir de un mensaje de la capa superior.

MODEM. Dispositivo que modula y demodula las señales que son transmitidas a través de ciertos tipos de facilidades de comunicación de datos.

PAQUETE. Conjunto de bits que incluyen datos mas información de control y es la unidad básica de transferencia de los redes de conmutación de paquetes.

PUNTO-A-PUNTO. Esquema de conexión en el cual, por cada par de estaciones adyacentes se comparte un enlace de transmisión.

PROTOCOLO. Conjunto de reglas, formatos y significados que gobiernan la operación de unidades funcionales (capas del mismo nivel jerárquico) para lograr una comunicación.

PUNTE. Dispositivo que enlaza dos o mas redes locales de conmutación de paquetes que sean homogéneas. Este dispositivo realiza la función de rutear paquetes emitidos en una red con direcciones de destinatarios localizados en otra. Para esto, el puente toma los paquetes de una red, los almacena y finalmente, de acuerdo a la dirección destino, los retransmite hacia la nueva red.

RED DE COMPUTADORAS. Conjunto de computadoras autónomas que se encuentran interconectadas y son capaces de intercambiar información.

REPETIDOR. Dispositivo que recibe datos de una línea de comunicación y los transmite, bit a bit, sobre otra línea tan rápido como los recibe,

Glosario

sin almacenarlos.

RUTEO. Operación de seleccionar la trayectoria apropiada para la transferencia de datos dentro de una red, asegurando que todos los datos son dirigidos al destino apropiado.

SERVIDOR. Computadora dedicada que ofrece un servicio a los usuarios de la red. Los servicios ofrecidos por estos sistemas pueden ser: manejo de archivos, manejo de terminales, manejo de impresiones, etc.

TERMINACION O TERMINADOR. Resistencia eléctrica localizada en los extremos de un cable que se utiliza para absorber la señal cuando esta llega al final, evitando la reflexión o rebote de la señal.

TOPOLOGIA. Designación de un nombre distintivo que se da a una red, de acuerdo a la configuración que guardan sus nodos y enlaces.

VENTANA DESLIZANTE. Método de control de flujo en el cual, las estaciones manejan los números de los paquetes en base a una ventana de números. La ventana de números se desliza desde cero, hasta un máximo, después del cual regresa a cero para volver a incrementarse.