

UNIVERSIDAD NACIONAL DE INGENIERIA
Facultad de Ingeniería Eléctrica y Electrónica



**Problemas de enrutamiento en
conmutadores ATM de grandes dimensiones**

TESIS

PARA OPTAR EL GRADO DE MAESTRO EN CIENCIAS

Mención **TELEMÁTICA**

Presentado por:

BARTOLOMÉ SÁENZ LOAYZA

Lima - Perú

2 005

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

Problemas de enrutamiento en conmutadores ATM de grandes dimensiones

TESIS

PARA OPTAR EL GRADO DE MAESTRO EN CIENCIAS

MENCION : TELEMÁTICA

PRESENTADO POR BARTOLOMÉ SÁENZ LOAYZA

LIMA - PERÚ

RESUMEN

La tesis está referida al problema general de diseño de un conmutador ATM $N \times N$ ($N > 1000$) de gran capacidad, gran ancho de banda y alta eficiencia. A pesar de los avances en el diseño de arquitecturas de conmutación, los conmutadores de dimensiones prácticos están limitados tecnológicamente y físicamente en cuanto al empaquetado del sistema. El presente trabajo se refiere a la escalabilidad del conmutador, es decir, se exponen las formas de construir un conmutador de gran dimensión, sin sacrificar el throughput, la eficiencia y el retardo.

Se exponen tres arquitecturas, de las cuales el Knockout Generalizado y el PINIUM están basadas en los siguientes principios:

- 1) El principio del Knockout aprovecha el comportamiento estadístico de los paquetes o celdas que llegan a la entrada y por tanto se reduce la complejidad de la interconexión,
- 2) los buffers de cola de espera ubicado en la salida tienen un mejor comportamiento en lo que respecta retardo, throughput y eficiencia, y

- 3) la inteligencia distribuida en el enrutamiento de los paquetes o celdas a través del sistema interconectado elimina los conflictos internos de encaminamiento.
- 4) Bloques básicos de implementación facilitan su escalabilidad.

Además, la arquitectura del Knockout Generalizado presenta otras características tales como garantizar la secuencia FIFO de los paquetes o celdas, capacidad intrínseca de uso broadcast y multicast y, es compatible con paquetes de longitud variable, evitando así su estandarización.

La arquitectura PINIUM propone la configuración tridimensional donde no existen interconexiones cruzadas entre los módulos. físicamente puede ser construido como si fuera una matriz tridimensional de procesadores paralelos. Esta arquitectura consiste de una sección de distribución y otra de concentración. Todos los planos de conmutación dentro de cada sección están totalmente particionados tal que entre ellos no existe interferencia alguna y cada plano de conmutación está subdividido en pequeños módulos. Esta arquitectura, emplea en la etapa de concentración el Knockout Generalizado para aprovechar el comportamiento estadístico de las celdas entrantes y la reducción en hardware. En el caso del tráfico Bursty, puede distribuir más de una celda hacia un puerto de salida en cada time slot.

En la arquitectura de conmutación rápida escalable propuesta en [3], la fábrica tiene una estructura duplicada bilateral y permite un crecimiento modular en su tamaño desde muy pocos puertos hasta gran cantidad de puertos. La fábrica es expandida conectando fábricas básicas adicionales a la estructura ya existente a través de algunos módulos de interconexión. La arquitectura también hace posible ajustar el throughput del sistema.

Finalmente, se hace un breve análisis de la probabilidad de pérdida de celdas en el desempeño de la arquitectura del Knockout Generalizado. Para ello se consideran dos fuentes de pérdidas de celdas: por Knockout y por Scheduling.

INDICE

	Págs.
Dedicatoria	
CAPÍTULO I	
Introducción	1
1.1. Formulación del problema	3
1.2. Objetivos de la tesis	3
1.3. Organización de la tesis	4
1.4. Metodología	4
CAPÍTULO II	
PRELIMINARES	5
Conmutación ATM	5
2.1. Conmutación	5
2.2. Concentración y multiplexado	11
2.3. Expansión y multiplexado	13
2.4. Definición de términos	13
2.4.1. Switching Fabric	13
2.4.2. Basic Switching Building Block	13
2.4.3. Sistema de conmutación	13
2.5. Requerimientos de conmutación	13
2.5.1. Tasas de información	14
2.5.2. Broadcast/Multicast	14
2.5.3. Desempeño	15
a) Bloqueo en conexión	15
b) Probabilidad de inserción y pérdida de celda	16
c) Retardo de conmutación	16
2.6. Bloques de construcción de un conmutador básico	17
2.6.1. Aplicación de la disciplina de espera en cola	18
a) Un buffer de cola de espera en la entrada del conmutador	19
b) Un buffer de cola de espera en la salida del conmutador	19
c) Un buffer de cola de espera en el centro del conmutador	20
2.6.2. Desempeño	21

2.6.2.1. Modelos analíticos	21
a) Para un buffer de cola de espera en la salida	22
b) Para un buffer de cola de espera en la entrada	24
c) Para un buffer de cola de espera central	27
2.6.3. Parámetros de implementación para la construcción de bloques básicos de un conmutador ATM	30
2.6.3.1. Buffer de cola de espera en la entrada	32
2.6.3.2. Buffer de cola de espera intermedia	33
2.7. Sistemas típicos de conmutación	34
2.7.1. Sistema de conmutación Knockout	35
2.8. Implementación de un conmutador genérico ATM	42
2.9. Enrutamiento	44
Resumen	46

CAPÍTULO III

ARQUITECTURAS DE CONMUTADORES DE GRANDES DIMENSIONES	47
3.1. Introducción	47
3.2. Arquitectura Knockout Generalizada	49
3.2.1. Generalización del principio Knockout	50
3.2.2. Disciplina de colas de salida	53
3.2.3. La fábrica interconectada	55
3.3. Arquitectura ATM plegable de un conmutador escalable	58
3.3.1. Arquitecturas plegables	58
3.3.2. Fábrica de tres estados y dos lados	60
3.3.3. Capacidad de crecimiento de la estructura conmutacional	63
a) Procedimiento de expansión.	63
b) Número de conmutadores elementales.	64
c) Puntos de ramificación.	68
3.4. Arquitectura de un conmutador ATM con memoria compartida	72
3.5. Arquitectura PINIUM	73
a.- Distribución	75
b.- Concentración	78
c.- Expansión y empaquetado de un conmutador PINIUM	80
Resumen	82

CAPÍTULO IV

PROBLEMAS DE ENRUTAMIENTO EN CONMUTADORES ATM

DE GRANDES DIMENSIONES	84
4.1. Problemas de enrutamiento	84
4.2. Algoritmo de enrutamiento distribuido	87
4.3. Análisis del desempeño	89
4.3.1. Rutas internas no definidas entre los módulos en sucesivos estados	90
4.3.2. Determinación de la ruta interna entre estados	99
4.4. Probabilidad de pérdidas de celdas: <i>Rangos más precisos para un modelo de tráfico uniforme</i>	102
4.4.1. Sin ruta principal entre los módulos en estados sucesivos	103
4.4.2. Ruta interna principal entre los estados	104
4.5. Resultados numéricos	105
4.6. Desempeño del conmutador PINIUM	107
4.6.1. Análisis de la pérdida en el desempeño del conmutador	107
4.6.2. Análisis del retardo en el desempeño del conmutador	111
4.6.3. Análisis, evaluación y resultados de un conmutador PINIUM	112
Resumen	118

CAPÍTULO V

CONCLUSIONES	119
ANEXO	123
BIBLIOGRAFIA	132

INDICE DE FIGURAS

CAPÍTULO II

Figura 2.1: Red Pública y privada con tecnología ATM con tráfico de voz, vídeo y datos.

Figura 2.2: Modelo general de un conmutador .

Figura 2.3. Conmutación espacial.

Figura 2.4. Conmutación de tiempo.

Figura 2.5: Ejemplo de interpretación de la tabla 1.1.

Figura 2.6: Estructura de la cabecera en ATM para una Interfase de la Red de Usuario (UNI).

Figura 2.7: Estructura de la cabecera en ATM para la Interfase de Nodo a Red (NNI).

Figura 2.8: Estructura de una celda ATM.

Figura 2.9: Estructura básica de un conmutador en el que se muestran los procesos de concentración, multiplexación, demultiplexación y expansión respectivamente.

Figura 2.10: Conmutador elemental, red de 4x4 y de 8x8 de Banyan.

Figura 2.11: Aplicación de cola de espera en la entrada

Figura 2.12: Conmutador con cola de espera en la salida.

Figura 2.13: Conmutador con buffer de cola de espera intermedia.

Figura 2.14: Probabilidad de pérdida de celdas para buffer central y buffer en la salida (modo M/D/1).

Figura 2.15: Reducción de la memoria debido al buffer central de cola de espera para la probabilidad de pérdida de celdas de 10^{-9} y $p = 0.8$.

Figura 2.16. Conmutación de un elemento por Knockout.

Figura 2.17: Interfase de bus Knockout.

Figura 2.18: Desempeño de la probabilidad de pérdidas de celdas versus Número De salidas del concentrador (L) para diferentes valores de p y para diferentes valores de la probabilidad de pérdidas de celdas P_r .

Figura 2.19: Ejemplo de un Conmutador 2x2.

Figura 2.20: Ilustración de la función de desplazamiento.

Figura 2.21: Interconexión de red multiestado.

Figura 2.22: Conmutador Knockout con módulos multiestado.

Figura 2.23: Conexión de los VC y VP en conmutadores ATM.

CAPÍTULO III

Figura 3.1: La Red de Clos (N, n, m).

Figura 3.2: Probabilidad de pérdidas de paquetes mediante el principio del Knockout Generalizado para valores de m desde 0 a 40.

Figura 3.3: Probabilidad de pérdidas de paquetes mediante el principio del Knockout Generalizado.

Figura 3.4: Lo mismo que la figura anterior, pero para valores m de 0 a 20.

Figura 3.5: Tasa de celdas simultáneas aceptadas para un grupo de tamaño n .

Figura 3.6: Arquitectura del conmutador que emplea el principio del Knockout Generalizado

Figura 3.7: Arquitectura de un conmutador escalable.

Figura 3.8: Arquitectura modificada de un conmutador escalable con ruta definida.

Figura 3.9: Escalabilidad de un "switching fabric" plegable.

Figura 3.10: Fábrica de dos lados y tres estados.

Figura 3.11: Una fábrica modificada de 3 estados.

Figura 3.12: Otra fábrica modificada de 3 estados.

Figura 3.13: Duplicando el tamaño de la fábrica básica.

Figura 3.14: Principio de expansión de la fábrica.

Figura 3.15: Una aplicación de la expansión de la fábrica de 64 a 128 líneas.

Figura 3.16: Número de conmutadores elementales de 8×8 en fábricas de varios tamaños.

Figura 3.17: Estructura de la fábrica $3N$.

Figura 3.18: Arquitectura de un conmutador con memoria compartida.

Figura 3.19: Arquitectura básica de un conmutador PINIUM.

Figura 3.20: Distribución de punto a multipunto. Método de aproximación de filtros de celda.

Figura 3.21: Distribución de punto a multipunto. Método de esquema VIA.

Figura 3.22: Selector concentrador priorizado de 16 a 4.

Figura 3.23: Expansión de una distribución.

CAPÍTULO IV

Figura No. 4.1: Un conmutador clásico TST.

Figura No. 4.2: Comparación de A_i y B_j .

- Figura No. 4.3:** Asignaciones paralelas en forma cíclica de mini-celda a mini-celda
- Figura No. 4.4:** Probabilidad de pérdida de paquetes en el principio del Knockout Generalizado.
- Figura No. 4.5:** Selección del camino en “competencia directa”.
- Figura No. 4.6:** Probabilidad de pérdida de celdas como una función del parámetro de expansión para diferentes valores de N , $p=0.9$ y $n=16$.
- Figura No. 4.7:** Probabilidad de pérdida de celdas como una función del parámetro de expansión para diferentes valores de carga de tráfico. $N=\infty$ y $n=16$.
- Figura No. 4.8:** Selección del camino en “competencia diagonal”.
- Figura No. 4.9:** Relación de expansión requerida como una función del tamaño de paquete n para diferentes probabilidad de pérdidas de celdas, con una carga de 90% y $N=\infty$.
- Figura No. 4.10:** Límites máximos y mínimos en la probabilidad de pérdida de celdas como una función del parámetro de expansión m para una carga de tráfico de 100 y 70%. Tamaño del conmutador $N=\infty$, y tamaño del paquete de salida $n=16$.
- Figura No. 4.11:** Carga entregada a la entrada con $\rho_{in}=1$ y para diferentes valores de C_{prom} .

CAPÍTULO I

INTRODUCCION

El modo de transferencia asíncrono (ATM) esta basado en la conmutación rápida de paquetes y ha sido adoptado por la Unión Internacional de Telecomunicaciones – Sector de Normalización de la Telecomunicación (ITU-T) como un método de transporte para redes digitales de servicio integrado de banda ancha (B-ISDN) desde 1993. El sistema ATM puede decirse que consiste de una técnica de paquetes tanto en la conmutación como en la multiplexación en el cual los paquetes están definidos de celdas de 53 bytes cada uno. Las celdas son entonces multiplexadas en uniones o enlaces para obtener la flexibilidad necesaria para soportar servicios de todos los propósitos.

ATM es un modo de transferencia orientado a paquetes y que usa una técnica de multiplexación por división de tiempo asíncrono y está orientada a conexión. Los identificadores de conexión son asignados a cada unión de la conexión cuando es requerido. Tal conexión se libera cuando no es necesaria por algún tiempo. El modo de orientado a conexión incluye al canal virtual que preserva la secuencia de las celdas.

El problema básico del sistema de conmutación ATM ha sido y es disponer de una central de conmutación de $N \times N$ provista de un alto desempeño en la interconexión de paquetes desde N puertos de entrada hasta N puertos de salida (para $N \geq 1000$). Tales conmutadores requerirán soportar rangos de throughputs desde decenas de gigabits por segundo hasta cientos y miles de gigabits por segundo, con una latencia medida en decenas de microsegundos y una probabilidad de pérdida de celdas muy pequeña (del orden de 10^{-10}). Los objetivos anteriores se pueden lograr con la implementación de conmutadores elementales paralelos que usen un procesamiento distribuido simple por celda entre muchos circuitos VLSI de alta velocidad.

En los últimos 12 años se vienen proponiendo diferentes tipos de arquitecturas de conmutadores, esto es, el de punto a punto y/o los conmutadores

multicast. En el diseño de conmutadores de punto a punto el enrutamiento de celdas y la contención en el puerto de salida son las mayores exigencias para la implementación de conmutadores elementales de paquetes. Es preferible desarrollar ambas funciones en una forma distribuida tal que se evite un procesamiento centralizado de celdas que pueda causar un cuello de botella y limitar el tamaño del conmutador. La contención del puerto de salida puede ser resuelto ubicando buffers ya sea en la entrada, en la salida o en el interior del conmutador. Para el caso del diseño de los conmutadores multicast, es necesario tener en cuenta tres consideraciones siguientes:

- a) Cómo replicar celdas.
- b) Cómo direccionar las celdas multicast, y
- c) Cómo reducir la complejidad de la tabla de interpretación.

Entre estas tres consideraciones, el direccionamiento multicast es un problema fundamental y un desafío desde que existen 2^N patrones multicast para N destinatarios, mientras que existen solamente N patrones unicast para la comunicación punto a punto. Para distinguir patrones multicast, al menos N ($=\log_2 2^N$) bits de información son necesarios, mientras que para la comunicación unicast el enrutamiento de información requerida tiene una longitud de $\log_2 N$. Existen tres posibles esquemas para el direccionamiento multicast. El primero es transportar la dirección en la cabecera de la celda. El segundo es transportar el patrón multicast que es un mapa de bits de todos los puertos de salida: cada bit indica si la celda será aceptada en el puerto de salida respectivo. El tercer esquema frecuentemente aplicado a una red no autoenrutada usa direccionamiento indirecto por medio de la identificación del multicast que es traducida en un mapa de bit de los puertos de salida usado por el multicast. Los tres esquemas propuestos se caracterizan por tener una cabecera de celda larga para conmutadores de gran tamaño por lo que requerirán muy alta velocidad de operación en el conmutador elemental.

En el presente trabajo se enfocan los conceptos básicos y algunas arquitecturas de conmutación para los sistemas multicast basados fundamentalmente en el sistema Knockout, toda vez que el objetivo básico de la presente tesis es examinar su desempeño en conmutadores modulares de grandes

dimensiones. El resultado y técnicas de análisis, sin embargo, son más generales y aplicables a otras arquitecturas con buffers en la salida. En la presente tesis, el análisis de los gráficos y la comparación de resultados va mas allá de lo descrito en las fuentes especializadas en el tema.

1.1. FORMULACIÓN DEL PROBLEMA

Se han desarrollado diferentes arquitecturas de conmutadores ATM para diferentes aplicaciones. Los factores que más han influido en el desarrollo e implementación de las arquitecturas de conmutadores ATM son la alta velocidad al cual debe de operar el conmutador y la característica estadística del flujo de información a través del conmutador ATM.

Así mismo, se han desarrollado técnicas de interconexión y algoritmos de enrutamiento. En lo que respecta al desarrollo de los algoritmos se puede expresar que lo más importante es que se tenga que su tiempo de respuesta sea rápida, porque los paquetes o celdas deben ser asignados a una ruta de salida, por tanto, la arquitectura debe ser posible de implementar en hardware. Un controlador central puede ser usado para obtener información del tráfico global para derivar un formato óptimo posible. Sin embargo, la obtención y la distribución de información requerida se convertiría en un cuello de botella severamente limitado al resultado de la velocidad de asignación.

Se han propuesto otras arquitecturas y técnicas de conmutación y sus respectivos algoritmos de enrutamiento, pero que no todos son posibles de implementar en forma directa, por lo que aún queda mucho que investigar en tal sentido, dado que aún se tienen problemas de enrutamiento.

1.2. OBJETIVOS DE LA TESIS

Realizar un análisis comparativo de los problemas de enrutamiento en una arquitectura modular de un conmutador ATM de grandes dimensiones, lo cual incluirá:

- El comportamiento probabilístico de las celdas y/o paquetes.
- La mejora en desempeño, rendimiento y retardos en las arquitecturas de conmutadores ATM.

- Un estudio del enrutamiento de las celdas a través de los “Switch Fabrics” (módulos de conmutadores elementales) que permitan minimizar los conflictos que se puedan presentar.

1.3. ORGANIZACIÓN DE LA TESIS

La presente tesis está organizada en 5 capítulos. En el primer capítulo se presenta la introducción, formulación del problema, objetivos y organización de la presente tesis. En el segundo capítulo se presenta una descripción del sistema conmutador ATM y los conceptos básicos necesarios para el desarrollo del presente trabajo. En el tercer capítulo se describen las arquitecturas de conmutadores ATM basadas en el sistema Knockout y cuyos resultados probabilísticos son comparados. En el cuarto capítulo se presenta un análisis general y detallado del comportamiento y desempeño de la arquitectura Knockout Generalizado. En el quinto capítulo se presentan las conclusiones correspondientes a la presente tesis. Del mismo modo, se incluyen en un anexo los programas de los gráficos más importantes.

1.4. METODOLOGÍA.

- Estudio de algunos tipos de arquitecturas de conmutadores de grandes dimensiones.
- Estudio de algoritmos de enrutamiento distribuidos de los conmutadores.
- Simulación de los algoritmos y estudio del desempeño.
- Análisis de los resultados.

CAPÍTULO II

PRELIMINARES

Conmutación ATM.

Es un sistema de conmutación usado en aplicaciones de voz, vídeo y datos bajo el principio de *orientado a conexión*, lo que permite optimizar en el tiempo la transmisión de datos a sus destinos correspondientes.

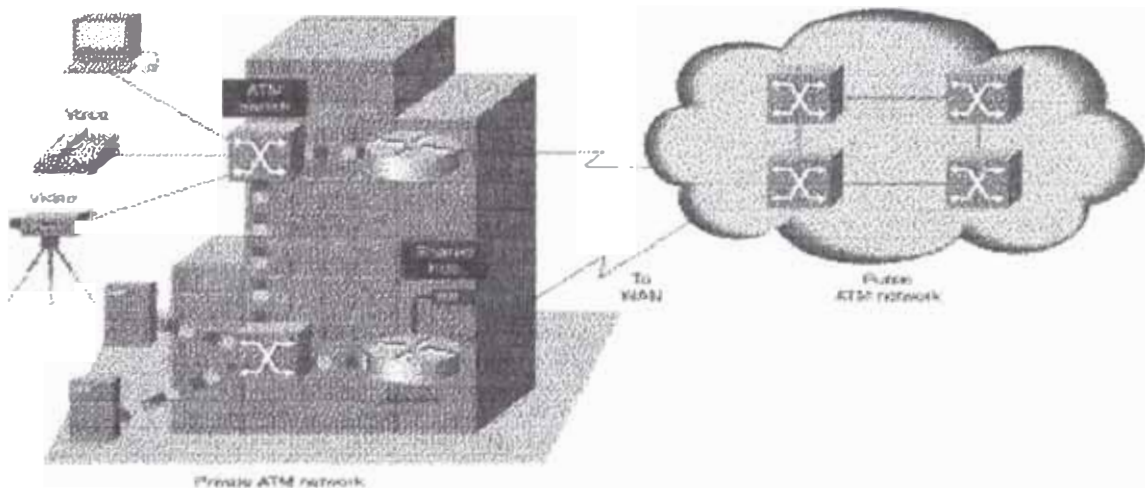


Figura 2.1: Red Pública y privada con tecnología ATM con tráfico de voz, vídeo y datos.

2.1. Conmutación.

Se refiere al transporte de información desde un canal de entrada lógica ATM hasta un canal de salida lógica ATM, para ser seleccionados del total de canales de salida lógicas. Este canal lógico ATM seleccionado debe cumplir las siguientes condiciones:

- Una entrada/salida física, debe disponer de un número de puerto físico.
- Un canal lógico en un puerto físico, debe estar caracterizado por disponer de un identificador de canal virtual (VCI) y/o el identificador de la ruta virtual (VPI).

En la figura 2.2 se puede notar que se dispone de un controlador de entradas y de un controlador de salidas. El controlador de entrada sincronizará las celdas entrantes mediante un reloj interno y los dos controladores serán unidos mediante la conexión de la red correspondiente. Para el proceso de conmutación: tanto la entrada y el identificador de canal o ruta de entrada tienen que estar relacionados con la salida y el identificador de canal o ruta virtual de salida respectivamente. Dos funciones deben ser implementadas en un sistema de conmutación ATM. Estas dos funciones pueden ser comparadas con las aplicadas a los sistemas clásicos de conmutación STM (Synchronous Transfer Mode).

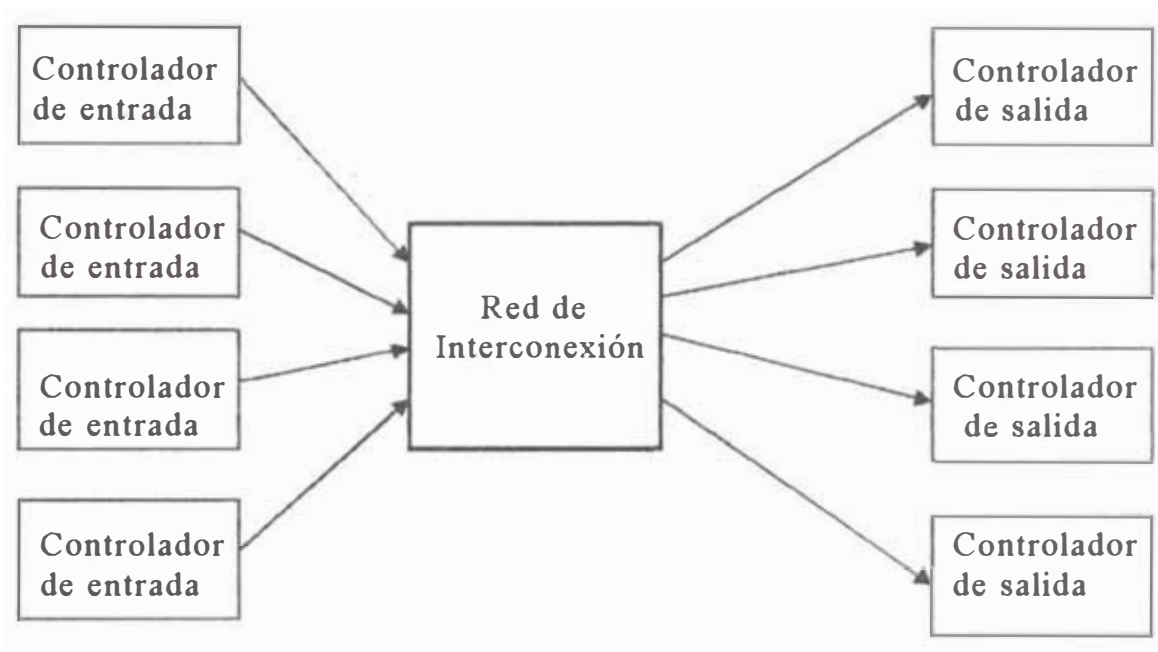


Figura 2.2: Modelo general de un conmutador .

La primera función es comparable a la función de conmutación espacial (S), en el cual el aspecto más importante es el enrutamiento, es decir, la información es encaminada internamente desde una entrada hacia una salida. Una aplicación se puede ver en la figura 2.3, donde la información de la entrada 1 es transportada hacia la salida 3, mientras que la información de la entrada 3 es transportada hacia la salida M.

La segunda función puede ser comparada con el intercambio del **intervalo de tiempo o celda (T)** en un tiempo de conmutación, una aplicación se muestra en la figura 2.4; es decir, el tiempo de celda **i** es conmutada al tiempo de celda **j**, mientras que la información de tiempo de celda **k** es transportada al tiempo de celda **l**.

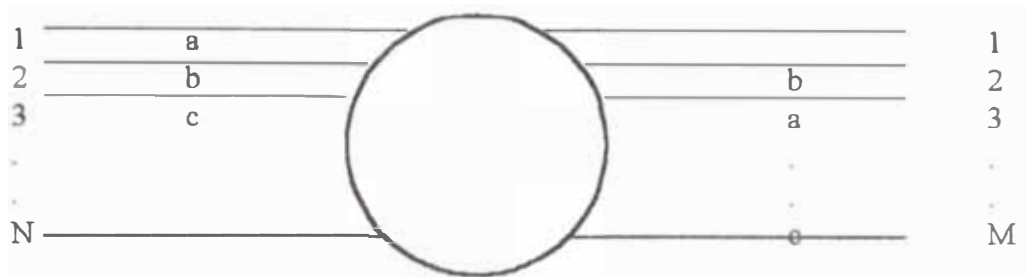


Figura 2.3. Conmutación espacial.

Los modernos conmutadores STM, con frecuencia están compuestos de una combinación de estos estados. Sin embargo, en el caso de un conmutador ATM, el tiempo de identificación en un trama fija es reemplazado por un canal lógico de identificación. En un sistema de conmutación ATM desaparece el concepto de tiempo pre-asignado a una celda o paquete de datos, pero se incrementa el problema de almacenamiento si existen dos o más canales lógicos que contienen datos para un mismo tiempo de celda; lo que queda resuelto por el uso de memorias temporales (buffers) de cola de espera para almacenar celdas ATM antes de ser enviadas hacia la salida. La función de cola de espera es la segunda en el sistema de conmutación ATM.

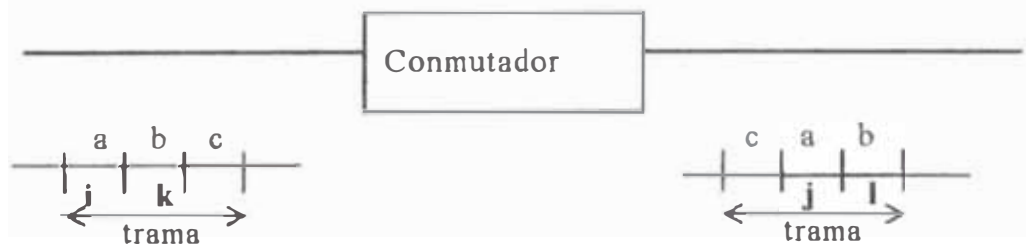


Figura 2.4. Conmutación de tiempo.

El principio básico de un conmutador ATM, visto desde el exterior, es equivalente a lo que se esquematiza en la figura 2.5, en el cual, se muestra que las celdas ATM entrantes son físicamente conmutadas desde la entrada E_j hacia una salida S_j , y a su vez el código de su cabecera es traducido desde un valor de entrada a a un valor de salida B . En cada conexión de entrada y salida individuales, los valores de las cabeceras son únicas, pero idénticas cabeceras pueden hallarse en diferentes conexiones (ejemplo: x en las conexiones E_1 y E_n).

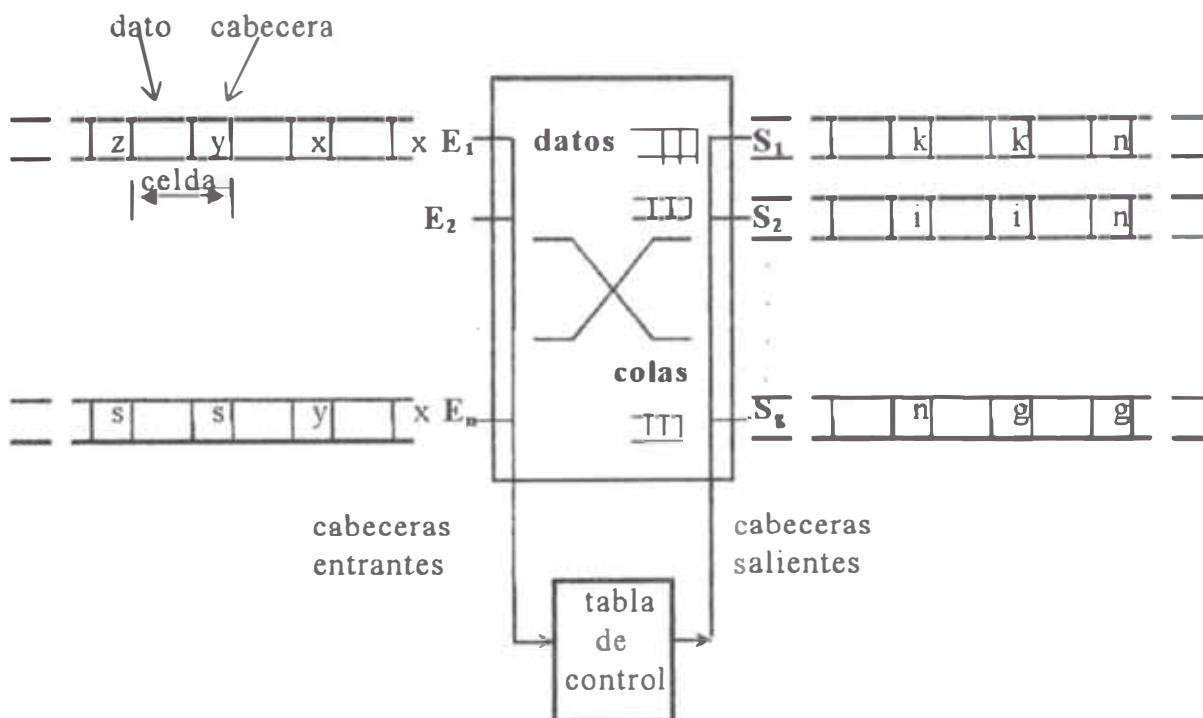


Figura 2.5: Ejemplo de interpretación de la tabla 1.1.

En la tabla 2.1, se muestra, como una aplicación práctica, una tabla de operación o traducción, en la que todas las celdas que tienen un código o valor de cabecera igual a x en el enlace de entrada E_1 son conmutadas hacia la salida S_1 siendo el valor de su cabecera traducida ("conmutada") a un valor k . Todas las celdas con una cabecera de valor x en el enlace E_n son también conmutadas a la salida S_1 , pero sus cabeceras toman el valor de n . En la figura 2.6 se muestra en bloques la estructura típica de una celda ATM.

Tabla de interpretación

Conexión de entrada	Tiempo de celda	Conexión de salida	Tiempo de celda
E_1	x y z	S_1 S_q S_2	k n i
· · ·	· · ·	· · ·	· · ·
E_n	x y s	S_1 S_2 S_q	n i g

Tabla No. 2.1. Principio del conmutador ATM.

Un sistema ATM está considerado como un modo específico de transferencia de paquetes orientados a conexión, basado en multiplexación por división de tiempo asíncrono que usa celdas de longitud fija.

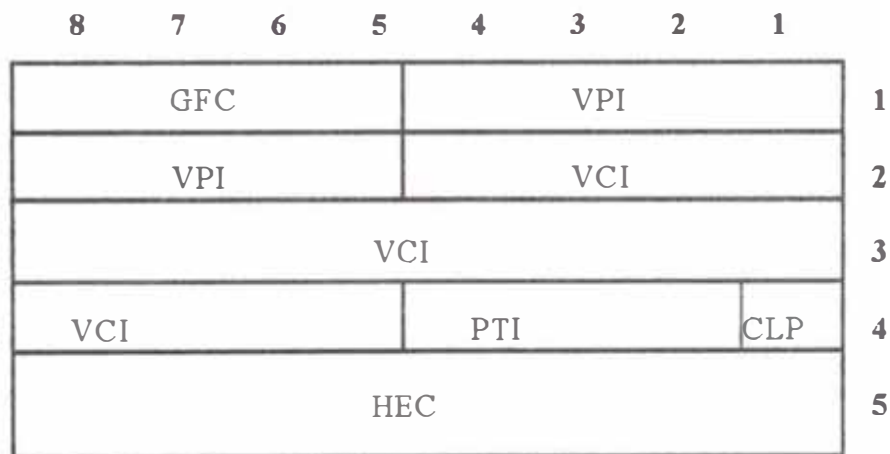


Figura 2.6: Estructura de la cabecera en ATM para una Interfase de la Red de Usuario (UNI),

Cada celda consiste de una cabecera y un campo de información. La cabecera es lo que se usa primero para identificar las celdas que pertenecen al mismo canal virtual dentro de un multiplexor asíncrono por división de tiempo y para de esa forma llevar a cabo un enrutamiento apropiado. La secuencia de las celdas es mantenida por el canal virtual.

En el proceso de enrutamiento, los valores de las cabeceras están asignadas a cada sección de una conexión para una duración completa de la conexión y traducida cuando se conmuta desde una sección hacia otra. Es decir, la cabecera permite verificar si los medios físicos de conexión están libres o no, para luego proceder con el proceso de conmutación, por eso se dice que el sistema es orientado a conexión. La señalización y la información útil son transportados en canales virtuales separados. En un sistema ATM se tienen dos tipos de conexiones: de canal virtual (VCC) y de camino virtual (VPC).

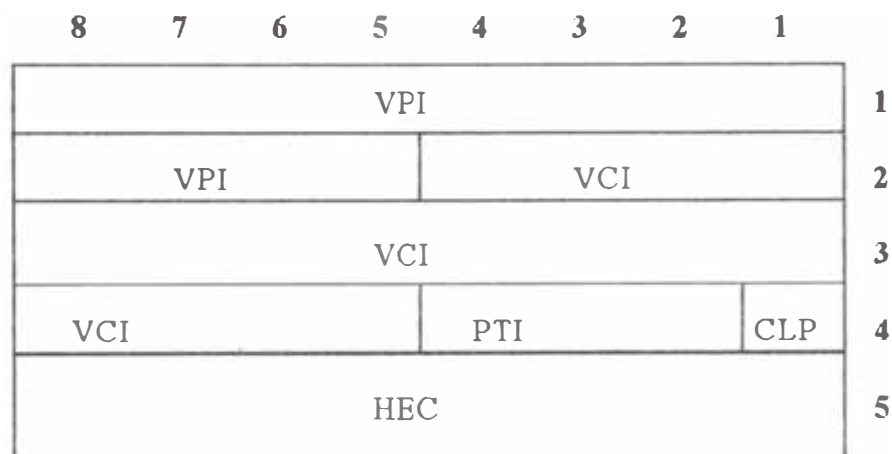


Figura 2.7: Estructura de la cabecera en ATM para la Interfase de Nodo a Red (NNI).

En las figuras 2.6 y 2.7 se tienen que:

- GFC : Control de flujo genérico.
- VCI : Identificador de canal virtual.
- VPI . Identificador de camino virtual.
- PTI : Identificador del tipo de carga.
- CLP . Prioridad de pérdida de celdas.
- HEC : Control del error de cabecera.

En la figura 2.8 se puede apreciar que la estructura de la celda ATM contiene 5 octetos para la cabecera y 48 octetos para el campo de información. Los octetos son enviados en un orden creciente, empezando por el primer octeto de la cabecera. Dentro de un octeto, los bits son enviados en un orden decreciente, empezando por el octavo bit. Para todos los campos de una celda ATM, el primer bit enviado es también el bit más significativo (MSB).

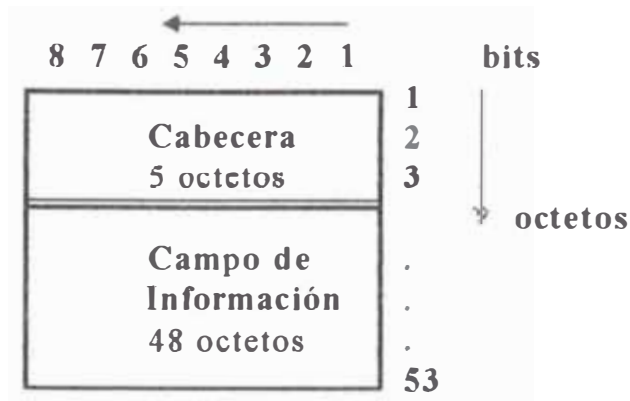


Figura 2.8: Estructura de una celda ATM.

Por lo indicado en los párrafos anteriores de la presente sección, queda demostrado que dos funciones básicas deben ser desarrollados en un conmutador ATM y estos son: “conmutación espacial” y “conmutación de cabeceras”, sin embargo, tal como se puede ver en la figura 2.5 es posible que dos celdas de diferentes entradas lleguen simultáneamente a un conmutador ATM y sean destinados a una misma salida.

Por lo tanto, ambos no pueden ser dirigidos a una misma salida simultáneamente. Por lo que, en algún lado del conmutador debe estar provisto un buffer de cola de espera para almacenar las celdas temporalmente. Esto es típico de un conmutador ATM, donde se multiplexan estadísticamente las celdas. En consecuencia, el sistema de cola de espera debe estar provisto en un conmutador ATM para asegurar que las celdas que son simultáneamente destinadas a la misma salida sean almacenadas y no perdidas o descartadas. Finalmente, podemos decir que, un conmutador ATM desarrolla tres funciones básicas: Enrutamiento (conmutación espacial), cola de espera e interpretación del código de cabecera. La forma cómo estas funciones están implementadas y su ubicación dentro del conmutador, generarán una distinción entre conmutadores ATM, en cuanto a la solución del o de los problemas de conmutación.

2.2. Concentración y multiplexado.

La información de N entradas es multiplexada en M salidas ($N > M$), siendo importante diferenciar los conceptos de multiplexaje y concentración. Cuando se emplea el concepto de multiplexación, el énfasis está puesto en el mezclado

estadístico de diferentes canales virtuales ATM (flujo de celdas) en un simple flujo ATM.

Cuando se emplea el concepto de concentración, lo que se desea es forzar la reducción del número de entradas a un menor número de salidas.

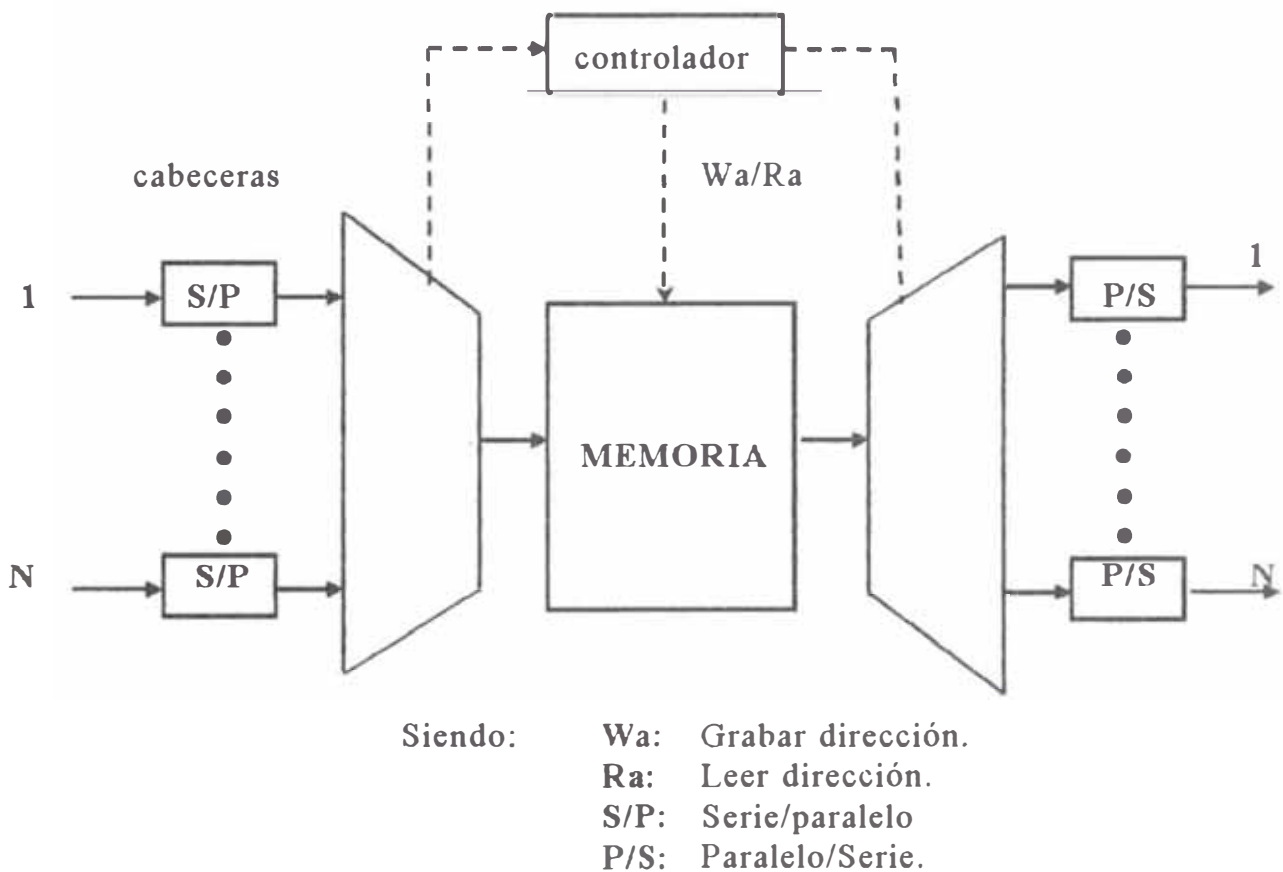


Figura 2.9: Estructura básica de un conmutador en el que se muestran los procesos de concentración, multiplexación, demultiplexación y expansión respectivamente.

Por ejemplo, en la figura 2.9 se muestra la estructura básica de un conmutador con memoria compartida, en el cual las celdas entrantes son convertidas de acceso serial a acceso paralelo y guardadas en forma secuencial en una memoria de acceso aleatorio. Un controlador de memoria, basándose en las cabeceras de las celdas y un arreglo interno de enrutamiento, decidirá el orden de lectura de las celdas. Las celdas salientes son demultiplexadas y convertidas de la forma paralela a la forma serial.

2.3. Expansión y demultiplexado.

La operación de Expansión/Demultiplexación viene a ser una inversa al de Concentración/Multiplexación.

2.4. Definición de términos.

En el desarrollo del sistema Conmutador ATM se han introducido diferentes términos, los que a su vez ya se emplean en forma normalizada a nivel mundial, los que definiremos a continuación en forma concisa.

2.4.1. “Switching Fabric”.

Expresa lo que sería un gran conmutador implementado por bloques de conmutación, interconectados en una topología específica. El Switching Fabric es definido cuando su topología está determinada y cuando son implementados sus bloques básicos de conmutación en forma bien definida, aunque existen algunos Switching Fabrics como aplicación de funciones especiales.

2.4.2. “Basic Switching Building Block”

Viene a ser el bloque de conmutación básico o elemental para la implementación genérica de un “ATM Switching Fabric”.

2.4.3. Sistema de conmutación.

Es el componente usado para encaminar celdas dentro de la interconexión de muchas celdas. Como se ha explicado en un acápite anterior, cuando un conmutador acepta una celda entrante desde un enlace físico, mediante el uso de su tabla de enrutamiento determina el enlace de salida al cual enviar. Si el enlace de salida está ocupado, el conmutador hará el uso de cola de espera para las celdas destinadas para dicho enlace físico de salida, no enviando nada hasta que la ranura de celda esté libre.

2.5. Requerimientos de conmutación.

Una red de banda ancha debe ser capaz de transportar todo tipo de información, es decir, desde telecontrol sobre una señal de voz hasta una señal de vídeo de alta calidad. Pero estos servicios, tienen diferentes requerimientos en términos de tasa de bits o velocidad binaria (desde unos cuantos Kbps hasta el

orden de cientos de Gbps), comportamiento en el tiempo (tasa de bits constante o variable), transparencia semántica (tasa de pérdida de celdas, tasa de error de bit) y tiempo de transmisión (sobre todos los retardos y fluctuaciones). Todos estos requerimientos deben ser satisfechos por conmutadores de banda ancha. Cabe aclarar que el usuario especifica la tasa de bits cuando ésta es constante, mientras que para el caso de tasa variable de bits el usuario define la tasa promedio de longitud de celda, el período en que se genera una ráfaga de bits, y la tasa del flujo de bits definida como principal. En un tráfico de baja prioridad, el usuario crea la tasa de bits disponibles (ABR) para su aplicación.

2.5.1. Tasas de información.

Desde que las tasas de bits para diferentes servicios son muy diversas, un gran número de éstos deben ser conmutados. Estas tasas varían desde algunos Kbps (por ejemplo telecontrol) hasta el orden de Gbps. Esto no significa que el conmutador ATM tenga que operar internamente en el orden de Gbps. Se sabe que la conmutación se realiza sobre unas barras paralelas, por lo que una menor velocidad es usada internamente, o los Gbps pueden ser multiplexados a un único enlazador, tal que, internamente se pueden implementar a muy altas velocidades.

2.5.2. Broadcast / multicast

En un conmutador clásico de paquetes, sólo las conexiones punto a punto estaban disponibles, porque la información tenía que ser conmutada desde una entrada lógica a otra de salida lógica. Sin embargo, en los conmutadores ATM actuales y del futuro los requerimientos son mayores, tanto que, algunos servicios tienen una naturaleza de “copia”, por lo que el conmutador debe ser capaz de proveer una funcionalidad multicomercial (Multicast) y comercial de banda ancha (broadcast), es decir, proveer información desde una fuente hacia muchos destinos y hacia todos los destinos. Estas funciones son típicamente requeridas para servicios tales como mail, librería de vídeo, distribución de señales de televisión. Es decir, la facilidad puede ser requerida desde muchos circuitos troncales hacia múltiples líneas de usuarios y desde éstas hacia otras múltiples líneas de usuarios.

2.5.3. Desempeño.

En el conmutador clásico STM, el desempeño estaba caracterizado fundamentalmente por: la inversa del tiempo mínimo de llegada entre dos eventos básicos. Esto es denotado como el intervalo de emisión pico de un conexasionado ATM, conocido como PCR (Peak Cell Rate) y el cual ha sido normalizado en la actualidad como **throughput** (cuya definición más precisa es: tasa total del tráfico de salida/tasa de tráfico de entrada), la probabilidad de bloqueo en conexión, tasa de bit de error y retardo de conmutación; pero en el conmutador ATM se han considerado dos parámetros adicionales muy importantes, éstos son la probabilidad de pérdida o inserción equivocada de celdas y la fluctuación en el retardo.

El PCR y la tasa de error de bit están principalmente determinados por la tecnología y la dimensión del sistema. Por ejemplo, usando tecnologías de alta velocidad como el CMOS, BICMOS o ECL, las tasas de bit del orden de Mbps o Gbps pueden fácilmente ser alcanzados, con una tasa de bit de error aceptable. Un gran **throughput** del conmutador puede ser obtenido mediante una determinada topología del conmutador.

Básicamente, tres de los parámetros antes mencionados son los que llaman nuestra especial atención para el caso de un conmutador ATM; éstos son, ***bloqueo en conexión, la probabilidad de pérdida o inserción equivocada de celdas y el retardo de conmutación.***

a) Bloqueo en conexión

ATM está definido como un sistema orientado a conexión. Esto significa que al conectarse debe hallarse necesariamente una conexión lógica entre la entrada y la salida. Debe tenerse en cuenta, que, aunque ATM es un sistema orientado a conexión no significa que la implementación del conmutador esté internamente orientado a conexión.

El bloqueo de conexión está determinado por la probabilidad de que no existan suficientes recursos entre la entrada y la salida que garanticen todas las conexiones existentes y la nueva conexión; esto es, si existieran suficientes

recursos de ancho de banda y valores o códigos de cabecera disponibles en la entrada y salida del conmutador, no ocurriría un bloqueo interno.

b) Probabilidad de inserción y pérdida de celda

Sucede que en un conmutador ATM, es posible que demasiadas celdas estén destinadas para un mismo enlazador que puede ser interno ó externo. La consecuencia de ello es que habrán mas celdas que la capacidad de un buffer de cola de espera, por lo que se perderán las celdas sobrantes. La probabilidad de pérdida de celdas debe ser mantenida dentro de los límites para asegurar una alta transparencia semántica. Los valores típicos de la probabilidad de pérdida de celda están entre 10^{-8} y 10^{-12} . Sin embargo, algunas arquitecturas de conmutadores ATM son diseñados para que no haya competencia de celdas para una misma fuente (uso adecuado de buffers de colas de espera), por lo que no habrá pérdida de celdas internamente.

También es posible que las celdas ATM sean internamente mal encaminadas, tal que ellas lleguen erróneamente a otras conexiones lógicas. La probabilidad de inserción de esta celda debe ser mantenida dentro de los límites considerados normales, y los valores de 1000 veces ó más son mejores que el rango de pérdidas de celdas típicamente mencionadas en la literatura [4], [9] y [27].

c) Retardo de conmutación

El tiempo para conmutar una celda ATM es también un factor importante. Los valores típicos mencionados para el retardo de conmutadores ATM están en el rango de 100 a 1000 μ s con una fluctuación de 80 μ s o menos. El valor de la fluctuación es en muchos casos determinado mediante la probabilidad de que el retardo del conmutador excederá en un cierto valor.

A tal valor se le denomina cuantil y típicamente se considera al valor comparable al rango aceptable de pérdida de celda. Por ejemplo, una fluctuación de 100 μ s a 10^{-10} cuantiles significa que la probabilidad de que el retardo en el conmutador sea mayor que 100 μ s es menor que 10^{-10} .

2.6. Bloques de construcción de un conmutador básico

La estructura de un conmutador ATM está compuesto de bloques de construcción, también llamado elementos de conmutación. La interconexión y la interacción de estos elementos de conmutación para conformar una estructura de

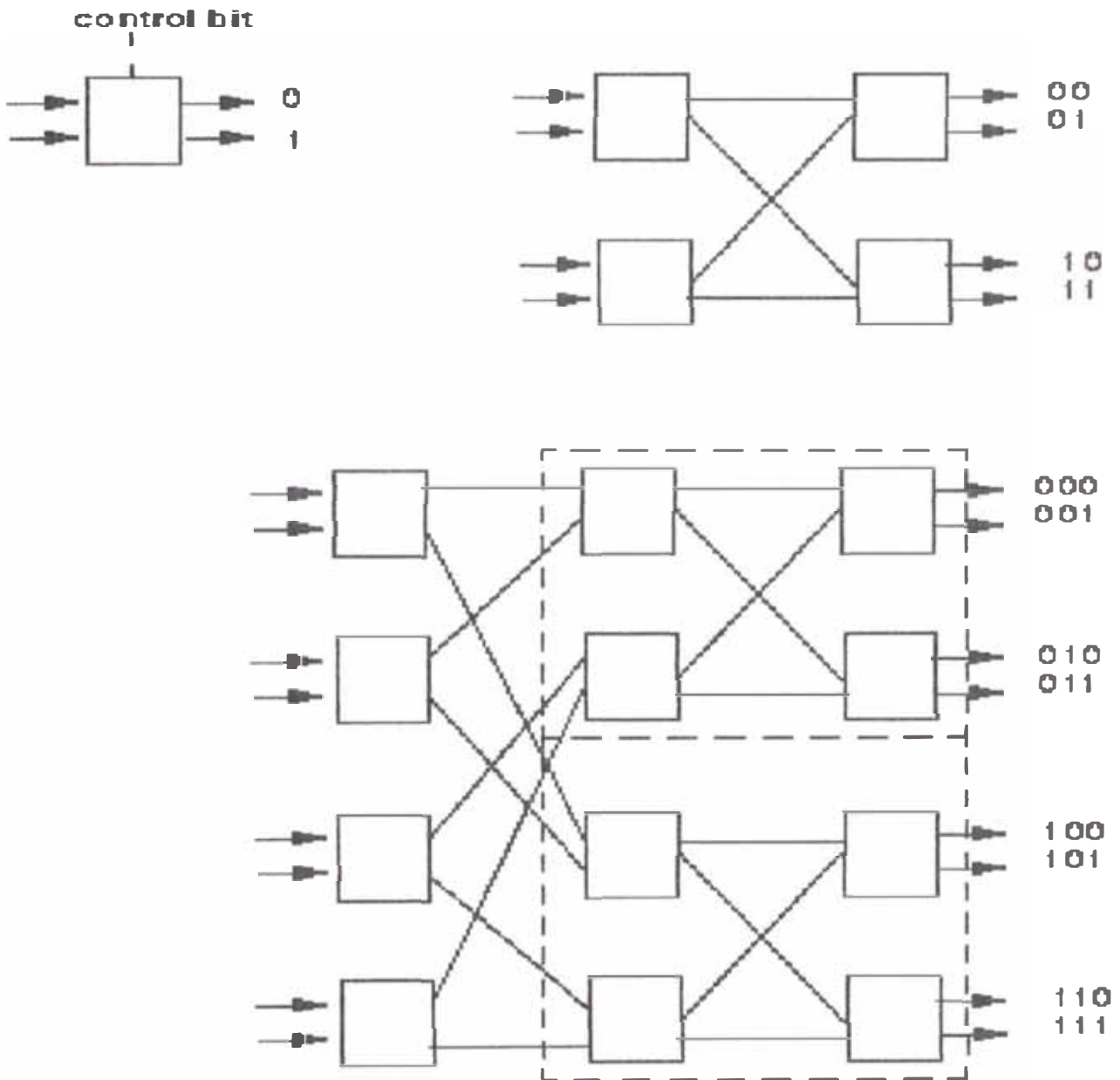


Figura 2.10: Conmutador elemental, red de 4x4 y de 8x8 de Banyan.

conmutación ATM, será detallada en la siguiente sección. Los elementos de conmutación ATM, típicamente son pequeños. Valores de 2 entradas y 2 salidas a 150 Mbps hasta 16 entradas y 16 salidas a 2.4 Gbps se han reportado en la

literatura. Sin embargo, los valores máximos no tienen límite. El tamaño (número de entradas / salidas) y la velocidad dependen de la tecnología de fabricación usada y el nivel que se desea lograr.

En la figura 2.10 se muestra un ejemplo de cómo se construye un conmutador de red a partir de un conmutador elemental de 2x2. La red de Banyan de 4x4 y de 8x8 es la que muestra y cuya forma de trabajo depende de un control que permite un adecuado enrutamiento de las celdas desde la entrada hacia una determinada salida.

Las funciones de conmutación de un elemento de conmutación están orientados principalmente a afrontar los problemas de colas de espera, porque, actualmente un elemento de conmutación ATM actúa como un multiplicador estático. Si dos celdas ATM llegan a dos entradas del elemento de conmutación para una misma salida durante la duración de una celda, uno de ellos tiene que estar en cola de espera para un tiempo de celda posterior. Dependiendo de la arquitectura particular de elementos de conmutación y la velocidad interna requerida, existe la posibilidad de almacenar temporalmente (cola de espera) esta celda ya sea en la entrada, en la salida o internamente en el elemento de conmutación. Sin embargo, debe notarse que algunas arquitecturas de conmutación están conformadas por elementos de conmutación que no tiene buffers internos. Estos buffers internos pueden sólo ser evitados si la estructura del conmutador está completamente sin bloqueo.

2.6.1. Aplicación de la disciplina de espera en cola

Un buffer de cola de espera, es una memoria temporal que permite bloquear una cantidad de celdas almacenándolas en secuencia y luego dejándolas salir en el mismo orden de llegada, normalmente del tipo FIFO (First In – First Out). La función básica de un elemento de conmutación es bloquear las celdas destinadas a una misma salida. En un elemento de conmutación, son posibles 3 estrategias de bloqueo. Estas son determinadas por la ubicación física de los buffers, esto es, en las entradas, en las salidas ó en la parte central del elemento de conmutación.

a) Un buffer de cola de espera en la entrada del conmutador

El objetivo de esta solución es resolver el problema de bloqueo de celdas en la entrada. Cada entrada dispone de un buffer dedicado, el cuál permite que se almacenen todas las celdas que lleguen hasta que una lógica arbitraria de control determine ó seleccione a uno de los buffers. El conmutador entonces transferirá las celdas ATM desde la cola de espera de entrada hasta la salida sin obstrucción interna alguna.

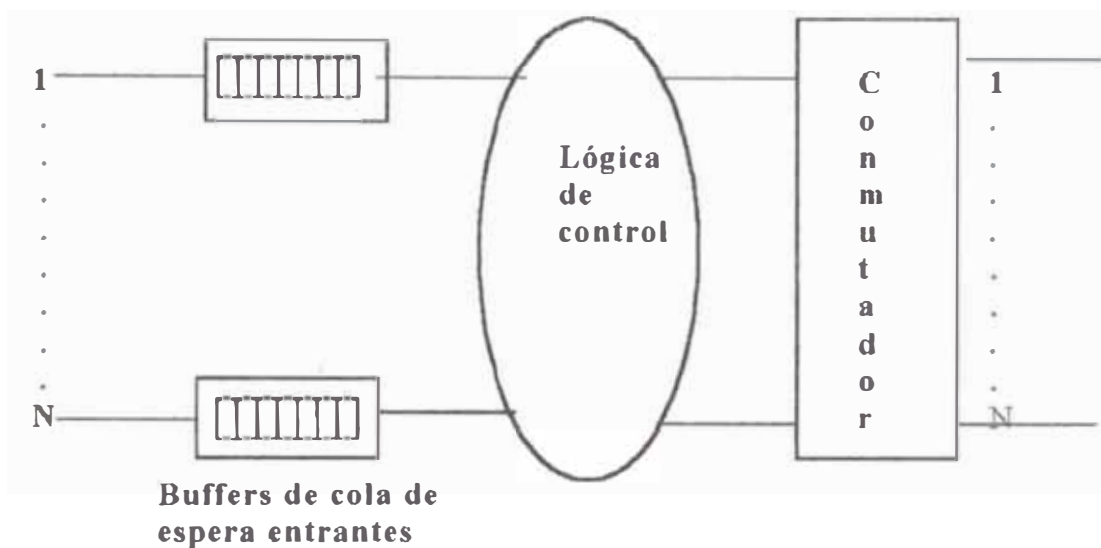


Figura 2.11: Aplicación de cola de espera en la entrada

Se toma en forma secuencial, desde la primera celda de una cola de espera; por ejemplo, si una celda de la entrada "i" es seleccionada para ser transferida hacia la salida p. Si la entrada "j" también tiene una celda destinada para la salida p, esta celda será bloqueada. La transferencia conmutada dura un tiempo denominado "tiempo de celda" o "tiempo de ranura o slot".

b) Un buffer de cola de espera en la salida del Conmutador

El objetivo de esta solución es transferir las celdas de diferentes entradas destinadas a una misma salida durante un "tiempo de celda". Sin embargo sólo una celda será transferida hacia una salida, y como consecuencia generará el bloqueo de las otras celdas destinadas a la misma salida. Esto se hace mediante la ubicación de buffer de cola de espera en cada una de las salidas del elemento de conmutación.

En principio, las celdas pueden llegar simultáneamente hacia todas las entradas destinadas para una misma salida. Para asegurar que ninguna celda se pierda en el proceso de transferencia o conmutación antes de llegar al buffer de cola de espera de la salida, la transferencia de celda debe ser llevada a cabo a una velocidad igual a N veces la velocidad de las entradas. El sistema debe permitir escribir N celdas en el buffer de cola de espera durante un “tiempo de celda”.

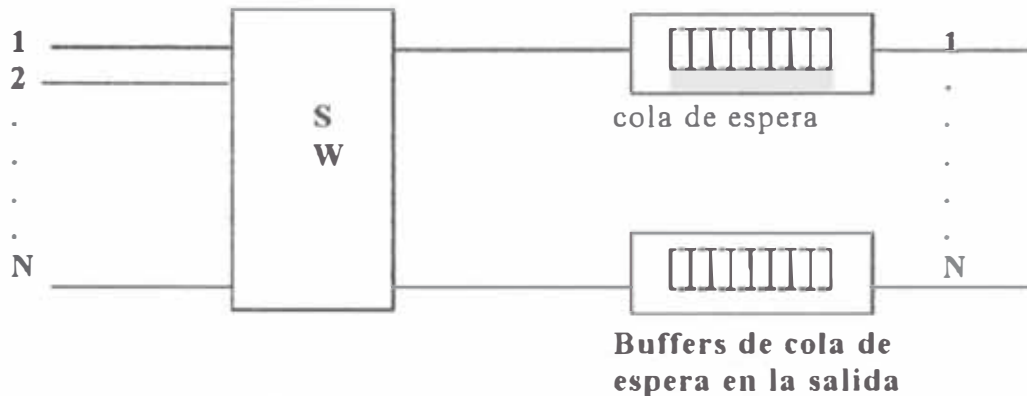


Figura 2.12: Conmutador con cola de espera en la salida.

c) Un buffer de cola de espera en el centro del conmutador

En este principio, los buffers de cola de espera no están dedicados a cada entrada y/o salida individualmente, sino que son compartidos entre todas las

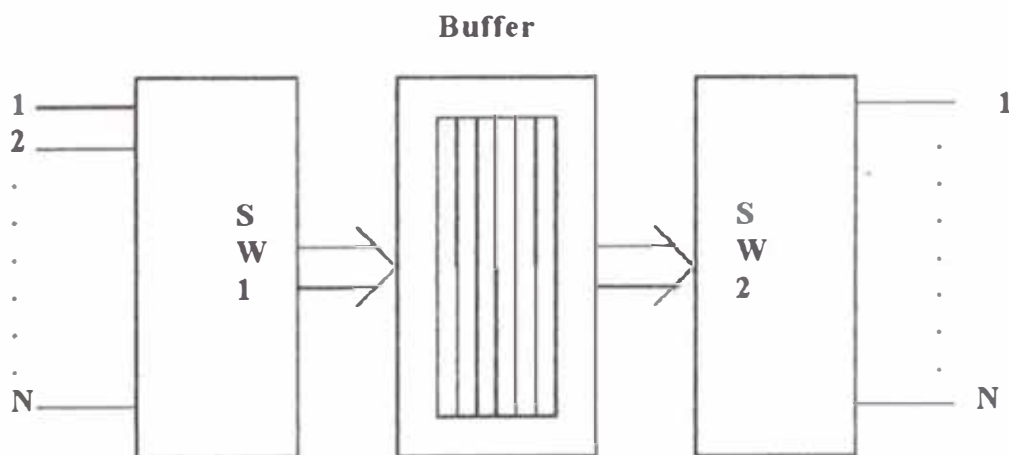


Figura 2.13. Conmutador con buffer de cola de espera intermedia.

entradas y salidas. En este caso cada celda que llega será directamente almacenada en el buffer central. Cada salida seleccionará las celdas que están destinadas a dicha salida mediante el principio FIFO desde el buffer central. El principio FIFO no es tan simple en este caso, toda vez que las celdas para diferentes destinos están mezcladas en el único buffer central. Esto significa que la memoria central debe estar direccionada en forma aleatoria. Una cola de espera lógica debe añadirse al principio FIFO. Por lo tanto, debido a que las celdas deben almacenarse y leerse en forma aleatoria, se requiere de un sistema de manejo de memoria más sofisticado.

2.6.2. Desempeño.

Las soluciones planteadas tienen una forma diferente de desempeño en términos de pérdidas de celdas, retardos requerimientos de buffers para celdas, en un cierto patrón de tráfico y carga en las entradas del conmutador. Se concluye que la mayor longitud de cola de espera y por ende mayor tiempo de espera es en el sistema de buffers en la entrada, seguida por el sistema con buffers en la salida y siendo el mejor con buffer central, en el cual se tendrá una sola cola pero de espera, aunque será necesario proveer una lógica inteligente que seleccione las celdas para salidas definidas.

2.6.2.1 Modelos Analíticos

Una fórmula analítica de menor complicación y que resulta en una buena aproximación a la realidad, es la planteada por Karol en 1986 [15], quien asume un modelo simplificado para celdas que llegan a los enlazadores (linkers) de entrada. Esto es, que el proceso de llegada de las celdas a cada entrada está basada en un proceso independiente de Bernoulli. Esto significa que en cualquier intervalo de tiempo (slot) dado, la probabilidad que una celda llegue hacia una entrada particular es p ($0 \leq p \leq 1$). En promedio, cada entrada es usada en un nivel p . Se asume también que el conmutador a ser modelado tiene N entradas y N salidas. Cada celda tiene la misma probabilidad $1/N$ de ser direccionada hacia alguna salida, tal que la probabilidad de que una celda llegue hacia una entrada y para una determinada salida es de p/N .

a) Para un buffer de cola de espera en la salida

La probabilidad X_i que i celdas lleguen hacia un cierto buffer de salida durante un tiempo de celda, puede calcularse en la forma siguiente.

X_i = probabilidad de que lleguen i celdas hacia un buffer de salida.

Para: $i = 0, 1, 2, \dots, N$.

$$X_i = C_N^i \left(\frac{p}{N}\right)^i \left(1 - \frac{p}{N}\right)^{N-i} \quad (1)$$

La función de generación de probabilidad $X(z)$ de la variable aleatoria X_i sera:

$$X_{(z)} = \sum_{i=0}^N z^i X_i = \sum_{i=0}^N z^i C_N^i \left(\frac{p}{N}\right)^i \left(1 - \frac{p}{N}\right)^{N-i}$$

Luego:

$$X_{(z)} = \left(1 - \frac{p}{N} + z \frac{p}{N}\right)^N \quad (2)$$

Esta función de generación de probabilidad puede ser usada para obtener la función de generación de probabilidad del número de celdas en el buffer $Q(z)$, tal como lo planteó Kleinrock en 1975 [16], en la forma siguiente:

$$Q_{(z)} = \left(\frac{(1-p)(1-z)}{X_{(z)} - z}\right) \quad (3)$$

El promedio del tamaño del buffer definido por Q puede ser calculado diferenciando (3) con respecto a z y tomando el límite para $z \rightarrow 1$.

Luego:

$$\bar{Q} = \frac{p^2(N-1)}{2N(1-p)} \quad (4)$$

El promedio del tamaño del buffer (Q_p) del bien conocido sistema de cola de espera M/D/1, con un proceso Poisson es conocido desde la teoría básica de colas de espera, también planteada por Kleinrock en 1975 [16], en la forma siguiente:

$$\bar{Q}_p = \frac{p^2}{2(1-p)} \quad (5)$$

Esto nos permite expresar el promedio del tamaño del buffer (Q) del sistema modelado como una función del tipo Poisson:

$$\bar{Q} = \frac{(N-1)}{N} \bar{Q}_p \quad (6)$$

tal que el tamaño promedio del buffer converge al sistema M/D/1 para $N \rightarrow \infty$. Esto no sólo es válido para el tamaño promedio del buffer, sino también para la distribución de la función de probabilidad del tamaño del buffer. Inclusive si $N \rightarrow \infty$, entonces (1) y (2) resultan:

$$\lim_{N \rightarrow \infty} [X_i] = \frac{p^i e^{-p}}{i} \quad (7)$$

y luego:

$$\lim_{N \rightarrow \infty} [X_{(z)}] = e^{-p(1-z)} \quad (8)$$

Esta fórmula muestra que el límite la característica de X_i es exactamente representado por el proceso de Poisson. Empleando(8) y (3) se obtiene:

$$\lim_{N \rightarrow \infty} [Q(z)] = \frac{(1-p)(1-z)}{e^{-p(1-z)} - z} \quad (9)$$

La cual es igual a la función de generación de probabilidad de un sistema M/D/1 de teoría de colas en la condición de estado estático. Empleando el teorema de Little [14], se calcula el tiempo promedio de espera del sistema como:

$$\bar{W} = Q\left(\frac{1}{p}\right)$$

$$\bar{W} = \frac{p(N-1)}{2N(1-p)},$$

luego:

$$\bar{W} = \frac{\bar{W}_p (N-1)}{N} \quad (10)$$

Se asume que \bar{W}_p representa el tiempo promedio de espera de un sistema M/D/1. En (10) se puede ver que para pequeños valores de carga (p), el tamaño promedio del buffer se hace muy pequeño (sólo con presencia de algunas celdas), pero para valores de carga mayores ó iguales a 0,8 el tiempo promedio de espera se incrementa exponencialmente. Una carga de 0,8 significa que el 80% de las celdas entrantes contienen datos útiles mientras que el 20% de las celdas entrantes están vacías. Estas celdas vacías serán descartados y no almacenados en el buffer de cola de espera. Por tanto, una conmutación basada en el principio de buffers en la salida tiene un comportamiento bien controlado para cargas superiores de 80 a 85%. También se puede ver que el impacto de N sobre el tiempo promedio de espera es también limitado.

b) Para un buffer de cola de espera en la entrada

Para modelar la característica de un elemento de conmutación con buffer de cola de espera en la entrada, se asume el mismo proceso de llegada

considerada para buffer de cola de espera en la salida, es decir, el principio del proceso de Bernoulli. Las celdas que llegan son almacenadas en el buffer de entrada y el cual actúa como un FIFO. El principio de servicio de los diferentes buffers es bastante simple. Esto es, si sólo una celda de posibles N primeras celdas de N entradas de buffers está destinada para una salida, obviamente será la seleccionada. Si j celdas están direccionadas hacia una salida particular, la selección de una celda es llevada a cabo en forma aleatoria, cada celda a ser seleccionada tendrá la probabilidad de $1/j$. Las otras celdas deberán esperar para un proceso de selección posterior en el siguiente tiempo de celda.

Para analizar las características de los buffers de cola de espera de entrada, se asume que todos ellos están saturados en cada entrada del buffer y que las celdas están en cola de espera.

Supongamos que B_n^i celdas están destinadas hacia la salida i pero bloqueadas en la cabecera de la cola durante el n ésimo tiempo de celda desde que ellas no fueron seleccionadas en el proceso de selección aleatoria. Durante el n ésimo tiempo de celda, un número de celdas serán usadas y transportados hacia sus respectivas salidas, habilitando un espacio para un número de nuevos i cabeceras de líneas o colas de celdas.

También se asume que A_n^i nuevas celdas aparecen en la cabecera de cola de espera para la salida i durante el n ésimo tiempo de celda. El hecho de que sólo una celda puede ser usada por la salida i durante un tiempo de celda, es debido a que en cada nuevo tiempo de celda el número de celdas bloqueadas en la cabecera de la cola de espera es igual al número de celdas bloqueadas durante el tiempo de celda previo B_{n-1}^i menos 1 (ya usada) más las nuevas celdas A_n^i que llegan, es decir:

$$B_n^i = \text{máx}(B_{n-1}^i - 1 + A_n^i, 0) \quad (11)$$

En cada buffer de entrada, nuevas celdas van llegando a la cabecera de la cola de espera; el total de celdas para todas las colas de espera en el tiempo de celda n ésimo será:

$$L_n = \sum_{i=1}^N A_n^i \quad (12)$$

Este valor representa el número total de celdas a ser servidas por el conmutador durante el tiempo de celda $n+1$. Si no ocurre bloqueo alguno, las N celdas serán conmutadas. Sin embargo, si B'_{n-1} celdas fueran bloqueadas, L_n también puede expresarse de la forma siguiente:

$$L_n = N - \sum_{i=1}^N B'_{n-1} \quad (13)$$

El promedio de L_n en la situación de estado estacionario, al que se le denota como L siendo igual a $N \cdot p$, donde p representa el throughput de conmutación por cada enlace de salida:

$$L = Np \quad (14)$$

Si $N \rightarrow \infty$, el número de celdas que se mueven hacia la cabecera de la cola de espera destinadas en estado estacionario para la salida i es calculada según la distribución de Poisson, con una variación p . Esto significa que la característica para $N \rightarrow \infty$ del valor principal en el estado estacionario de B^i es como el de un sistema $M/D/1$, luego:

$$B^i = \frac{p^2}{2(1-p)} \quad (15)$$

Si se considera el promedio de (13) y considerando (14) resultará:

$$\bar{L} = N - \sum B^i$$

o

$$\sum_{i=1}^N \bar{B}^i = N(1-p) \quad (16)$$

ó si $N \rightarrow \infty$, entonces:

$$\bar{B}^i = 1-p \quad (17)$$

Siendo entonces:

$$p^2 - 4p + 2 = 0 \Rightarrow p = 2 + \sqrt{2}$$

Si se combinan las ecuaciones (15) y (17), se obtendrá el valor de la probabilidad de carga p ya considerada, tal como puede verse que el conmutador saturado con $N \rightarrow \infty$, alcanza una carga p igual a:

$$p_{m\acute{a}x} = 2 - \sqrt{2} = 2 - 1,4142 = 0,5858 \quad (18)$$

Por lo que, un conmutador basado en un buffer de entrada tiene un rendimiento limitado. La máxima carga obtenible es de 58,6%, por lo que el tamaño promedio de la cola de espera es muy limitado.

c) Para un buffer de cola de espera central

Este planteamiento de solución presenta un comportamiento exacto al planteado con un buffer en la salida del conmutador. Esto significa que el principal tiempo de espera es exactamente igual al del sistema con buffer en la salida. Sin embargo, desde que múltiples colas de espera son combinadas en una sola memoria físicamente existente; su mayor ventaja de este sistema es reflejada en el número de celdas a ser almacenadas en la memoria central (buffer central). Un control lógico más complejo es requerido para asegurar que la memoria trabaje según la disciplina FIFO para todas las salidas. El tamaño de la memoria puede ser calculada como una convolución de N salidas individuales de colas de espera.

Además desde que la memoria Buffer será compartida, se hace necesario un uso más efectivo de la memoria. Se puede hacer una estimación de esta ganancia, por lo que se calculará la probabilidad de pérdida de celdas, truncando la cola de la distribución de probabilidad, el cual es obtenido con infinitos buffers de salida.

El número de celdas Y en el buffer de la cola de espera central será igual a la suma de celdas X en cada buffer de cola de espera de salida, tal que la distribución de probabilidad del número de celdas en el buffer compartido es la convolución de la distribución de cola de espera de cada puerto de salida.

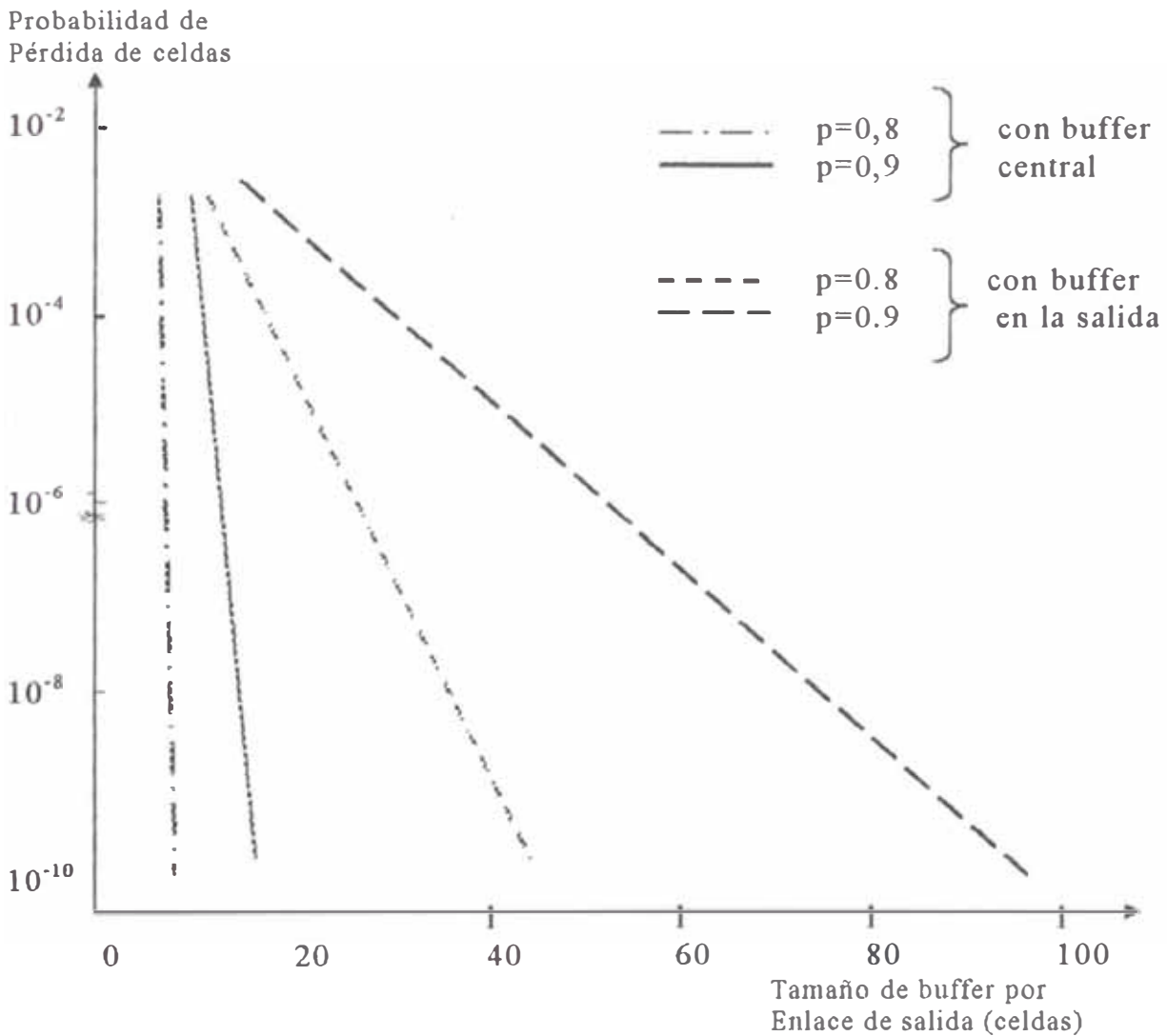


Figura 2.14: Probabilidad de pérdida de celdas para buffer central y buffer en la salida (modo M/D/1).

Se asume que el conmutador tiene N salidas, y que $M_X(v)$ es la función de generación de momentos de X . Si el proceso es descrito por el modelo $M/D/1$, entonces:

$$M_{X(v)} = \frac{(1-p)(1-e^{-v})}{(1-e^{-v})e^{(1-e^{-v})p}} \quad (19)$$

o:

$$m_{X(v)} = \log M_{X(v)} ;$$

luego:

$$m_{X(v)} = \log(1-p) + \log(1-e^{-v}) - \log(1-e^{-v} \cdot e^{(1-e^{-v})p}) \quad (20)$$

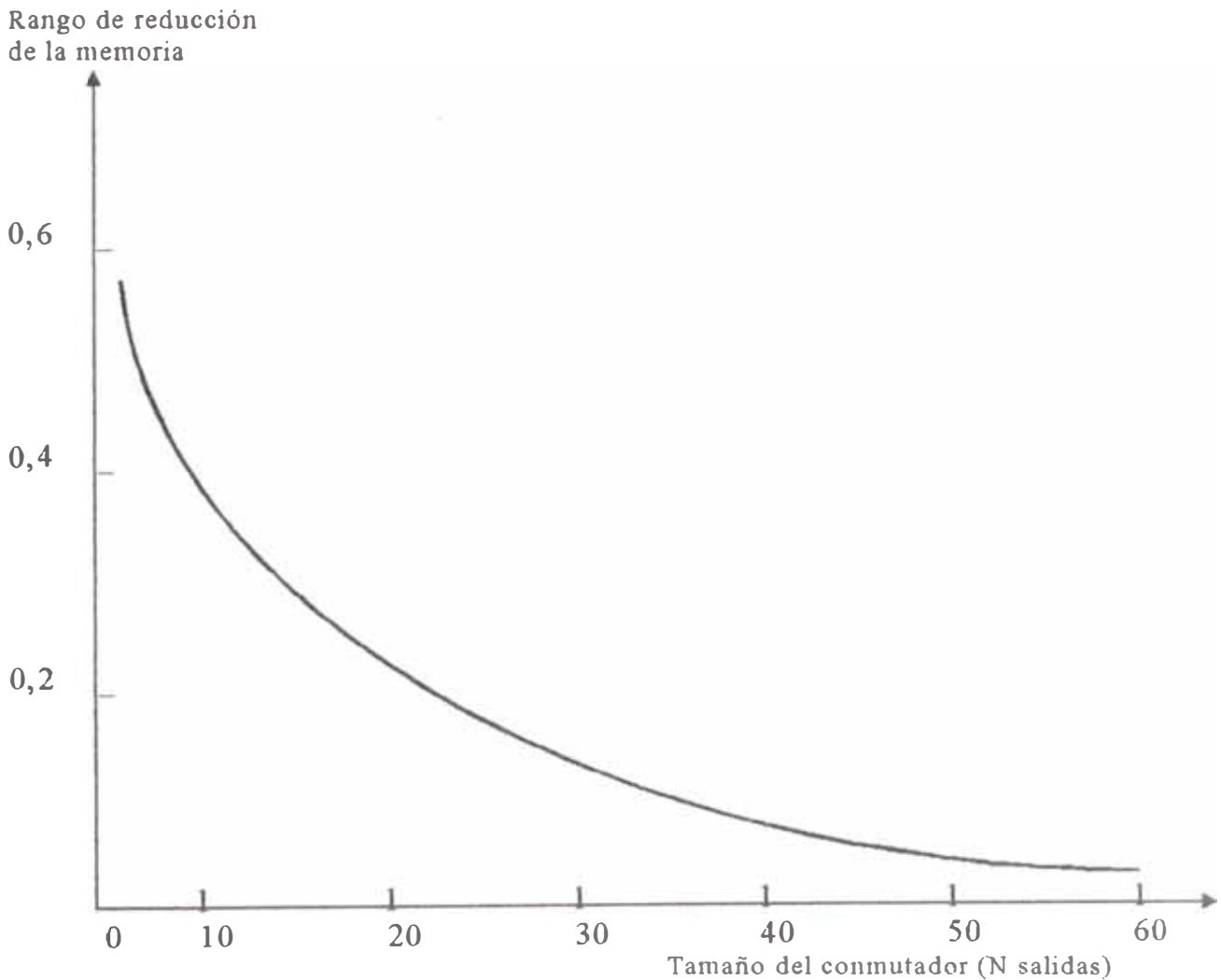


Figura 2.15: Reducción de la memoria debido al buffer central de cola de espera para la probabilidad de pérdida de celdas de 10^{-9} y $p = 0.8$.

Esta fórmula puede ser usada para calcular la cota de Chernoff para la solución de memoria compartida como:

$$P_{|Y \geq Nm_{1(v)}|} \leq e^{N(m_{1(v)} - v \cdot m_{1(v)})} \quad (21)$$

Empleando la fórmula anterior se puede calcular el tamaño del buffer para un cierto rango de pérdida de celdas. La relación entre el tamaño del buffer y el rango de pérdida de celda se puede graficar y hacer una comparación, tal como se muestra en la figura 2.14. De ella se puede concluir que este sistema garantiza el mismo rango de pérdida de celda de 10^{-8} , el tamaño de la cola de espera en el sistema con cola de espera en la salida es alrededor de 40 celdas; para el sistema de cola de espera intermedia, se ha reportado que se requiere menos de 10 celdas por cada enlace de salida.

La figura 2.15 muestra la ganancia en el tamaño de la memoria para el caso de un buffer de cola de espera central. Dicha ganancia depende del tamaño del conmutador el cual depende de N que es el número de celdas a conmutar.

2.6.3. Parámetros de Implementación para la Construcción de Bloques Básicos de un Conmutador ATM.

Los requerimientos de realización física de los 3 sistemas planteados son muy diferentes, no sólo debido a las diferencias en el tamaño de los buffers de colas de espera, sino también debido a los requerimientos de velocidad y control lógico adicional para la memoria correspondiente. En la actualidad, se tienen definidos los 3 parámetros que influyen en la complejidad de los diferentes sistemas de buffers de colas de espera antes planteados:

- a) **Tamaño de la cola de espera.**- El tamaño depende de los requerimientos del desempeño del sistema (como rango de pérdida de celda, carga retardo) y el principio de buffer de cola de espera seleccionado. Está reflejado en el número de buffers de celdas a ser soportados por el conmutador.
- b) **Velocidad de memoria.**- El tiempo de acceso de la memoria de la cola de espera del conmutador depende del principio de la cola de espera, pero también del tamaño N y la velocidad de enlaces de entrada y salida en el conmutador.

c) *Control de memoria.*- Para controlar las colas de espera de un conmutador, es requerido un control lógico adicional. La complejidad de dicho control lógico depende del principio de la cola de espera. Por ejemplo, para el principio FIFO se requiere un control lógico simple, mientras que en un conmutador con buffer intermedio se requiere una administración de una memoria dinámica.

Los 3 parámetros antes mencionados hacen que la selección de los 3 sistemas de conmutadores sean dependientes de la disponibilidad tecnológica de circuitos integrados (CMOS, ECL, etc.) y caracteriza el tamaño (tal como el ancho del circuito integrado) tanto como la velocidad al cuál el sistema tiene que operar (por ejemplo, 100, 250, 600 Mbps) y en el tamaño N de los bloques de construcción del conmutador básico (2x2 hasta 32x32). Para comparar las 3 alternativas, se asume que N describe el número de entradas y salidas, F la velocidad de enlace de cada una de las entradas y salidas del bloque de fabricación básica y W el ancho de la memoria de cola de espera sea accionada en paralelo.

El buffer de cola de espera en la salida puede recibir una celda desde todas las N entradas simultáneamente (N operaciones de escritura), donde una celda puede ser enviada hacia la salida (una operación de lectura). Todas estas N+1 operaciones tienen que ser realizadas durante un tiempo de celda.

Como el tiempo de acceso a la memoria del sistema de la cola de espera de la salida es $W/[(N+1).F]$. En los casos en que, en una memoria es posible realizar la escritura y lectura simultáneas, sólo N operaciones de escritura deben llevarse a cabo durante un tiempo de ranura o celda. Entonces, el tiempo de acceso de memoria es de $W/(N.F)$. Esta última opción es sólo posible para una moderada reducción del tiempo de acceso a la memoria. Es decir, una memoria de doble puerto requerirá típicamente 1,5 veces la superficie del circuito integrado que el de una memoria de un sólo puerto. Por lo que, una memoria de dos puertos no es recomendable para un sistema con buffer de cola de espera en la salida.

Tal como puede verse en la tabla No. 2.2, el tiempo de acceso a la memoria en el ejemplo es un poco menor a 6,3 ns. La ganancia obtenida por una memoria

de doble puerto en tiempo de acceso no es el deseado, considerando su complejidad incrementada.

El control lógico de un buffer de cola de espera en la salida es considerablemente simple. Esta lógica está basada exclusivamente en el principio FIFO, por lo que sólo requiere un puntero de lectura y escritura para controlar la selección de la dirección de la memoria respectiva durante dichas operaciones.

En la implementación, se conectan todas las entradas con todas las salidas mediante un bus TDM. Si ninguna celda se perdiera en el bus TDM, opera a una velocidad de $N.F$, para de esa forma asegurar que todas las N celdas son transferidas durante un tiempo de celda. Debido al requerimiento de altas velocidades, más el hecho de que un bus no es ideal para tales velocidades, debido a las reflexiones en los terminales de los buses, hace que el bus TDM no sea una selección ideal como medio de transferencia de un sistema con buffer de cola de espera en la salida, especialmente si el conmutador elemental tiene que ser implementada mediante muchos circuitos integrados.

Sin embargo, si se tiene una longitud muy pequeña de un bus (por ejemplo en un circuito integrado), los problemas de reflexión prácticamente desaparecen, haciendo que un bus TDM sea una solución ideal como un medio de transferencia debido a que no es complejo; cabe aclarar, que la solución con bus TDM fue planteado en el sistema de conmutación ROXANNE [27].

Sin embargo, se han planteado la implementación de buses diferentes al TDM para mejorar la velocidad, pero requieren mayor control lógico, pudiendo ser sin bloqueo alguno de celdas [6] ó con mínima probabilidad de pérdida de celdas [9]. Sin embargo, dado que en la actualidad el medio de transmisión más usado es la fibra óptica, es posible idealizar la solución a este propósito.

2.6.3.1. Buffer de cola de espera en la entrada

En este sistema, la memoria sólo puede ser almacenada simultáneamente por una entrada y una salida. El tiempo de acceso a la memoria es de sólo $W/(2F)$ en caso de una memoria de un sólo puerto y de W/F en caso de una memoria de dos puertos.

Tal como puede verse en la tabla No. 2.2, el tiempo de acceso a la memoria no representa un mayor problema, toda vez que, un tiempo de acceso de 50 a 100ns puede ser manejable por las memorias existentes comercialmente.

En este sistema, (ver figura 2.11) la ganancia que se logra con una memoria de dos puertos, es considerada como una ventaja. La lógica de control de este buffer de cola de espera emplea la disciplina FIFO al igual que en el sistema con buffer en la salida. El medio de transferencia puede operar a una velocidad menor a $N.F$, desde que puede conmutar en el buffer de entrada para evitar una posible situación de contención, mediante el bloqueo del exceso de celdas en dichos buffers de entradas.

Sin embargo, desde que no se tiene buffer de cola de espera en la salida del sistema, las capacidades de distribución comerciales del medio de transferencia sólo pueden ser usadas mediante una lógica de control adicional para así evitar posibles contenciones en la salida.

2.6.3.2. Buffer de cola de espera intermedia

En este sistema se emplea el tiempo de acceso igual a $W/(2NF)$, desde que cada entrada y salida puede almacenar simultáneamente a la memoria. Para memorias de dos puertos el tiempo de acceso es reducido a $W/(NF)$. Para lograr altas velocidades, la memoria tiene que ser combinada con el control de la cola de espera en un sólo circuito integrado. La ventaja de una memoria de dos puertos, en este caso, respecto al tiempo de acceso tiene que ser evaluado respecto a la superficie mayor del circuito integrado.

El promedio de transferencia a la entrada y salida de la memoria central es de NF . El control de la memoria central es más complicado. Inclusive, celdas destinadas para una salida no serán almacenadas en forma contigua con otras celdas destinadas para una misma salida. Por lo que, se hace necesario el uso de memorias dinámicas. Sin embargo, ya que estos mecanismos deben ser realizados en tiempo real a altas velocidades, la selección de una de estas técnicas de ubicación de memorias es determinada por la implementabilidad del hardware.

Es decir, un conmutador ATM está compuesta, como ya se dijo, de bloques de construcción básica que son los elementos de conmutación. Los elementos de conmutación ATM reportados hasta el momento son de pocas entradas y pocas salidas (por ejemplo 16 entradas y 16 salidas a 2.4 Gbps). Sin embargo, los valores máximos no están limitados. El tamaño (número de entradas y salidas) y la velocidad dependen de la tecnología de fabricación empleada y el nivel de integración deseada.

En los elementos de conmutación, el problema principal es la cola de espera (queuing), por que en un conmutador ATM los elementos actúan como multiplexores estadísticos. Dependiendo de una arquitectura particular de un elemento de conmutación y de la velocidad interna requerida, la posibilidad de espera existe para una celda ya sea en la entrada, salida o en el interior del sistema conmutador. Sin embargo, debe notarse, que se han propuesto varias soluciones al respecto, tanto que inclusive se han construido conmutadores haciendo uso de ciertas arquitecturas como los que usan un buffer interno (sistema Tri State planteado por Martin Colliere – 1997 [17]) o con cola de espera en la entrada (analizada una estrategia de aplicación por Lillycutty Jacob y Anurag Kumar – 1996 [28]), entre otros.

2.7. Sistemas típicos de conmutación ATM.

En la conmutación ATM, se han planteado varios sistemas de conmutación, siendo los primeros el Knockout [9], Coprin [29] y el Athena [6], propuestos en 1987. El principio Knockout lo describiré en detalle en la siguiente sección. En base al principio Knockout, existen nuevos planteamientos de solución de conmutación para mayores dimensiones.

El sistema Coprin [29] fue diseñado originalmente para un sistema experimental de conexión de data, voz y video. En el prototipo, el elemento de conmutación operó con enlaces o conexiones de a 280 Mbps, con celdas de 15 bytes de información y un byte de encabezado y consistente en una matriz cuadrada de 16x16.

En enrutamiento del sistema Coprin [29] fue hecho basándose en el encabezado de la celda. Este encabezado contiene un número referencial, el cual es re-inicializado para cada conexión. Este número referencial es usado para determinar la salida física, al mismo tiempo debe ser traducido a un nuevo número referencial.

El sistema Athena [6], descrito en 1987, está basado en un buffer de cola de espera ubicado en la salida del sistema. La descripción de su implementación fue dada en 1990 y realizada en una tarjeta de 16 entradas por 16 salidas, cada uno operando a 600 Mbps. Está basado en un sistema orientado a conexión con un encabezado traducible en cada estado.

El sistema Roxanne [27], basado en un buffer de cola de espera central, fue presentado en 1990. Fue descrito como un sistema 32x32 y que operaba a 150 Mbps. El sistema para el enrutamiento de la celda, debe primero, interpretar las diferentes partes del conexionado. Además, algunas celdas requerirán una diferente interpretación de la información del enrutamiento, dependiendo del modo de enrutamiento de la celda.

2.7.1. Sistema de conmutación Knockout [9].

Su principio está basado en un buffer de cola de espera en la salida, sin embargo, algunas características del sistema con buffer central son válidas.

Tal como puede verse en la Tabla, el tiempo de acceso a la memoria en un conmutador con buffer en la salida será muy pequeño. Para reducir la velocidad de operación (reducir complejidad), el medio de transferencia Knockout emplea un sistema concentrador, para ello se introduce un criterio probabilístico de cierta pérdida de celdas. Por tanto el sistema de Conmutación Knockout no solo requiere de dimensionar los buffers de colas de espera, sino también, la concentración de este medio de transferencia.

El sistema de conmutación Knockout tiene N entradas y N salidas, cada uno operando a la misma velocidad. Las celdas de longitud fija (paquetes) llegan a cada una de las entradas en un trama de tiempo regular, definido como celdas ATM. El medio de transferencia está compuesto de N buses de distribución

comercial, uno para cada entrada. En la figura 2.16, se muestra que cada una de las N entradas envían sus celdas en buses comerciales separados, sobre el cual cada salida tiene un acceso a través de un bus interfase con N entradas. Esto significa que el medio de transferencia no tiene bloqueo y que en la entrada de un bus interfase no se han perdido celda alguna. Las celdas serán bloqueadas por una simple salida en este bus interfase, el cual es típico de este sistema de conmutación.

Descripción	Buffer en entrada	Buffer en salida	Buffer central
Memoria de un puerto	$W/(2F)$	$W/[(N+1).F]$	$W/(2NF)$
Ejemplo de aplicación	53,3 ns	6,3 ns	3,8 ns
Memoria de 2 puertos	W/F	$W/(NF)$	$W/(NF)$
Ejemplo de aplicación	106,6 ns	6,7 ns	6,7 ns

Tabla 2.2: Tiempo de acceso a la memoria para las tres disciplinas de ubicación de cola de espera.[33]

La estructura del bus tiene su mayor ventaja en que cada bus es controlado solamente por una entrada, permitiendo una implementación simple y una alta tasa de transmisión comparada con un bus que es compartido por múltiples entradas. Inclusive, si el bus tiene que ser compartido entre las entradas, un período de tiempo adecuado será aceptado entre todas las entradas incluidas. Sin embargo, se debe tener cuidado con las reflexiones de la señal si la velocidad del bus interfase se vuelve demasiado rápido.

A un bus interfase llegarán muchas celdas simultáneamente y todas destinadas para una misma salida. En el peor caso, N celdas están destinadas hacia una misma salida, en consecuencia, el bus interfase requiere de buffers de celdas en algún lado.

Para garantizar que ninguna celda se pierda en la fase de transferencia al buffer de celdas, la memoria debe grabar a N veces la velocidad de cada entrada. El sistema Knockout reduce dicha velocidad de grabación, mediante el uso de un bus interfase inteligente, el cual actúa como un concentrador con una probabilidad de pérdida de celdas igual a cero.

Este bus interfase, asociado con cada salida, se muestra en la figura 2.17, en el cual se muestran N filtros de celdas conectadas cada una de ellas a los N buses comerciales. Estos filtros examinan la dirección de cada celda que llega. Si está destinada hacia una salida específica, la celda es pasada al concentrador, en caso contrario la celda es descartada.

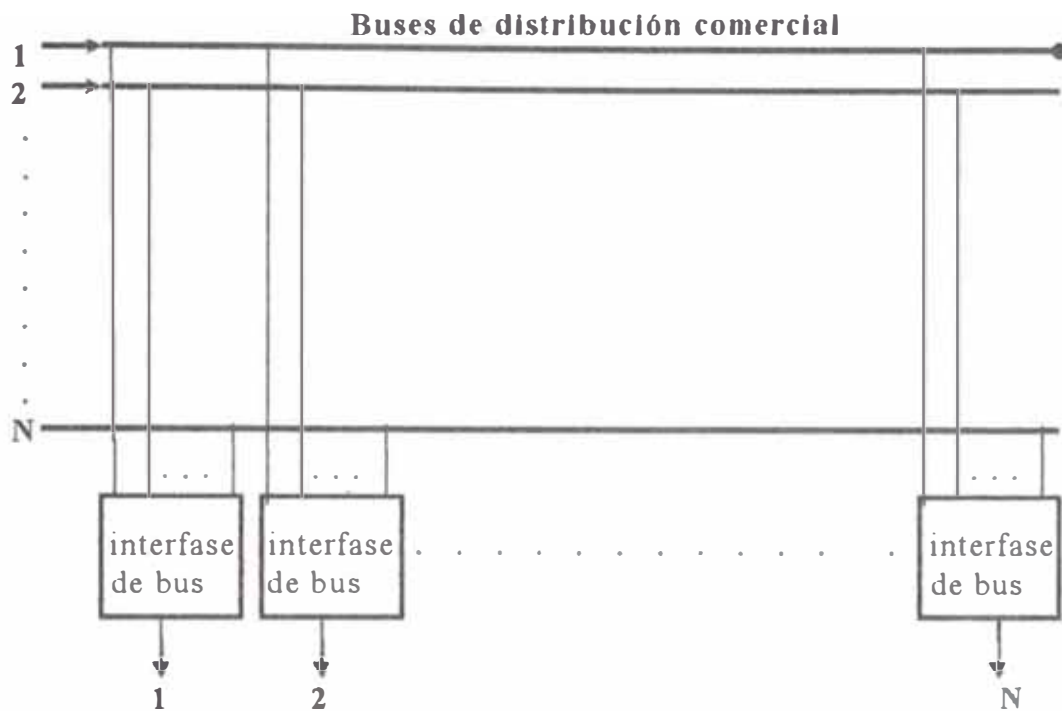


Figura 2.16. Conmutación de un elemento por Knockout [9].

La siguiente parte del bus interfase es el concentrador de N entradas a L salidas (intermedias) ($L < N$). Si K celdas llegan simultáneamente para una misma salida, entonces, éstas K celdas, siendo $K \leq L$, llegarán hacia una de las salidas del concentrador (1 a K). Si $K > L$, entonces todas las L salidas del concentrador tendrán celdas y $K - L$ celdas se perderán en el concentrador.

La probabilidad de pérdida de celdas en el concentrador no debe ser mayor que la probabilidad de pérdida de celdas en el sistema de conmutación Knockout. El principio de cálculo está basado en lo siguiente. Si se asume que todas las celdas que llegan hacia una entrada son independientes, con igual probabilidad de carga p e igualmente destinadas hacia cada salida, entonces la probabilidad P_k

de que K celdas lleguen simultáneamente al concentrador es una distribución binomial, es decir:

$$P_k = C_k^N \left(\frac{p}{N}\right)^k \left(1 - \frac{p}{N}\right)^{N-k} \quad k=0,1,\dots,N. \quad (22)$$

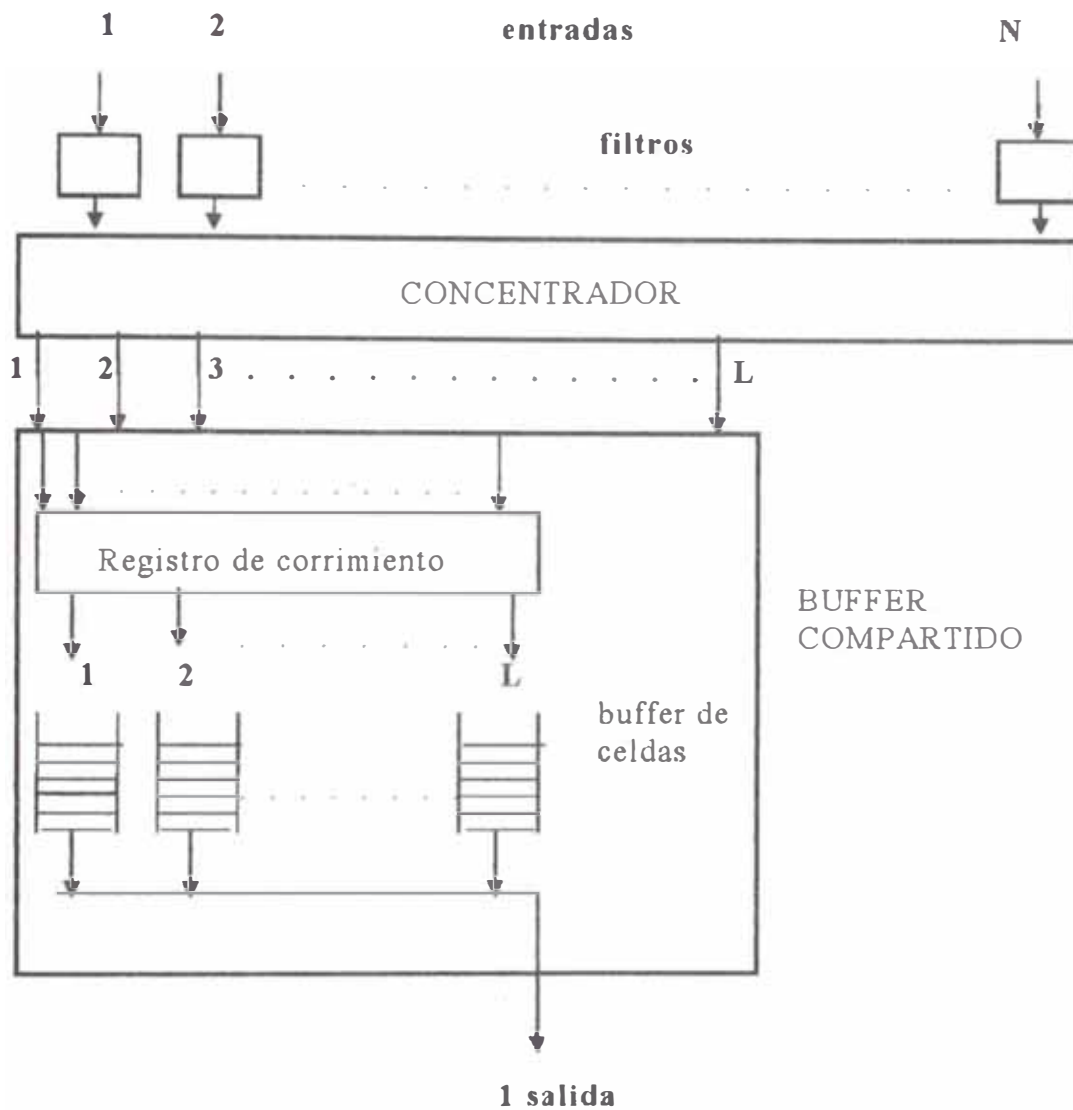


Figura 2.17: Interfase de bus Knockout. [9]

Si solamente L celdas pueden pasar a través del concentrador, entonces la probabilidad de pérdida de celdas estará dada por:

$$P_r = \frac{1}{p} \sum_{k=L+1}^N (k-L) C_k^N \left(\frac{p}{N}\right)^k \left(1 - \frac{p}{N}\right)^{N-k} \quad (23)$$

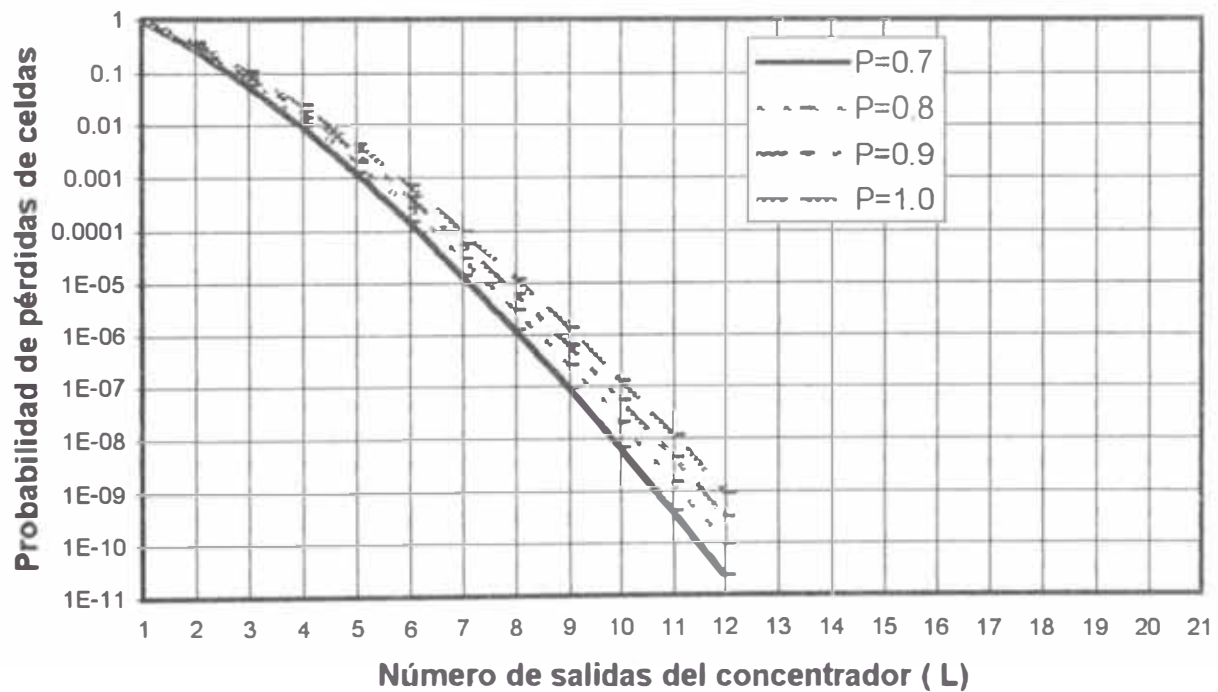
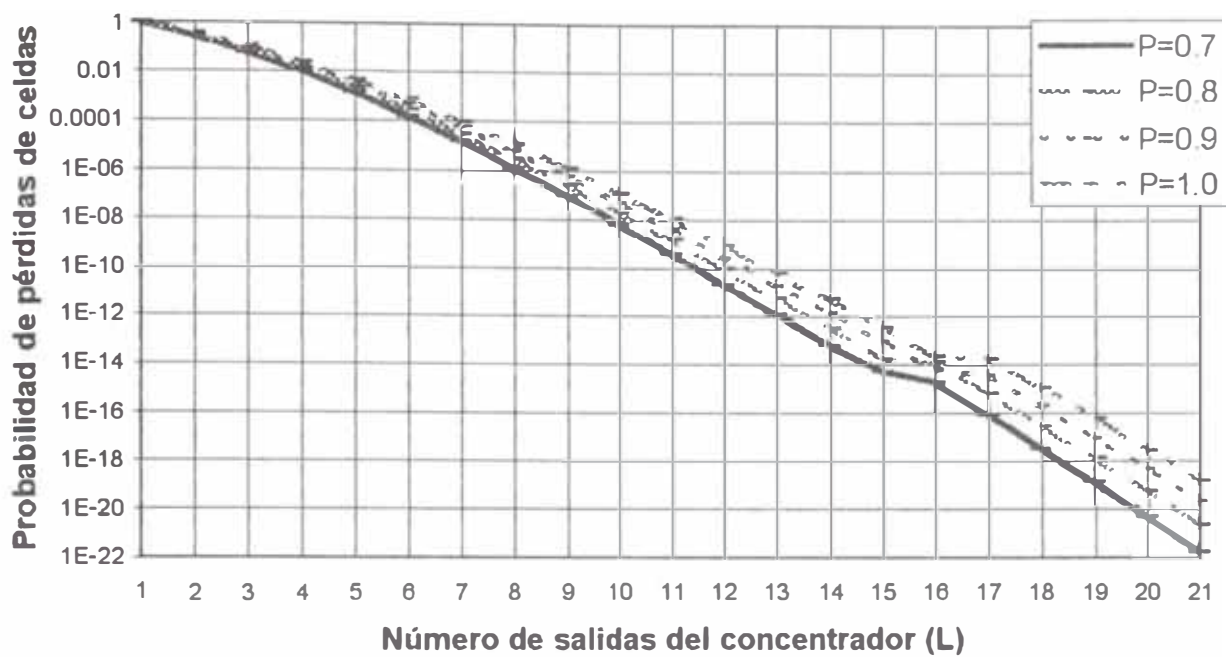


Figura 2.18: Desempeño de la probabilidad de pérdidas de celdas versus Número de salidas del concentrador (L) para diferentes valores de p y para diferentes valores de la probabilidad de pérdidas de celdas P_c .

Esta fórmula nos permite calcular la probabilidad de pérdida de celdas en función de L para diferentes valores de N y $p = 0,9$; y además, para diferentes valores de p y $N \rightarrow \infty$ se obtiene:

$$P_r = \left[1 - \frac{L}{p} \left[1 - \sum_{K=0}^L \frac{p^K e^{-p}}{K!} \right] \right] + \frac{p^L e^{-p}}{L!} \quad (24)$$

Al graficar (23) y (24) Yeh [9], se concluye que L no es tan sensitivo para p y N, pero es principalmente influenciado por el requerimiento de la probabilidad de pérdidas de celdas. Por ejemplo, si $L=12$ se obtiene una probabilidad de 10^{-10} para cualquier valor de p y N. Sin embargo, debemos ser muy cautelosos con las gráficas obtenidas por las ecuaciones (23) y (24), ya que éstas nos muestran solamente el desempeño del estado de concentración. La probabilidad de pérdida de celdas, debe ser considerada en comparación con los valores obtenidos en el principio de conmutadores con buffers de cola de espera en la salida, para el caso de colas de espera.

Mediante el uso de la fórmula (24), para el presente trabajo, se tabulan y obtienen valores para cada valor de la carga y al hacer la simulación respectiva se obtiene la gráfica que se muestra en la figura 2.18.

El concentrador en sí puede ser construido a partir de bloques simples de construcción. Estos bloques son básicamente un conmutador en el cual dos entradas disputan una salida y sólo uno de ellos será el ganador. Si solamente una celda ATM está presente en la entrada, entonces dicha celda será lógicamente seleccionada como ganadora; si dos celdas están presentes en la entrada, una de ellas será la ganadora y la otra perdedora.

Para construir un concentrador más grande, se organiza una forma de competencia en el cual el ganador o los ganadores son seleccionados automáticamente para una siguiente competición y torneo y bajo el principio de un conmutador de 2×2 . Para un concentrador con N entradas y L salidas, habrán L secciones de competencia para cada salida, tal como se pueden ver en las figuras 2.19 y 2.20 respectivamente. Después del concentrador, en el bus

interfase se halla el buffer de celdas que es donde se almacenan las celdas que salen desde el concentrador, pero que no salen directamente desde el bus. Desde que cada buffer está ubicado en la salida de cada bus interfase, la opción tomada es del principio de cola de espera, es decir, el número de acceso a memoria durante un tiempo de celda será igual a $N+1$ para una memoria de una sola entrada. Pero, en el sistema de conmutación Knockout este número es reducido a $L+1$, desde que el concentrador tiene reducido sus números de entrada de N a L . La implementación del buffer en la salida del conmutador es realizada por L buffers separados tipo FIFO y que requiere sólo dos accesos a memoria (uno para lectura y otro para escritura) por cada tiempo de celda, en vez de $L+1$. Sin embargo, para distribuir la carga sobre todos los buffers y obtener el mismo desempeño en pérdida de celdas, los L buffers deben ser compartidos y operar virtualmente como un simple buffer.

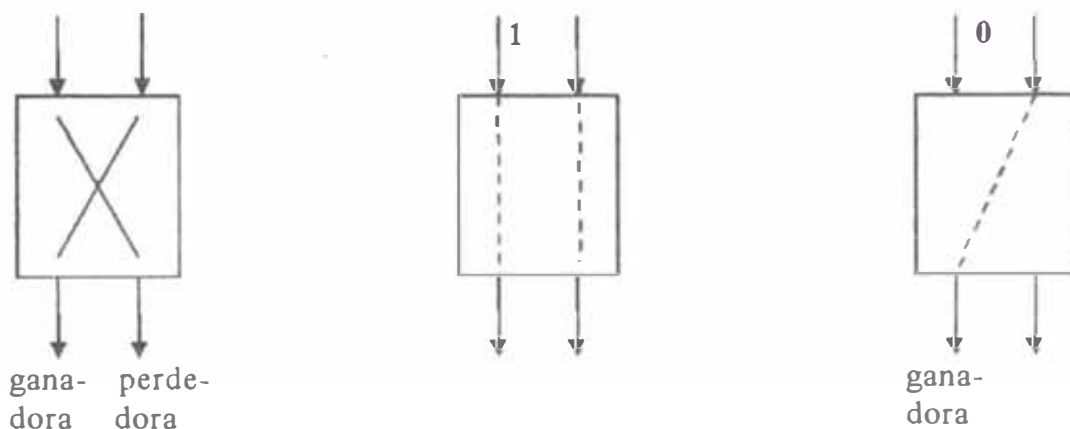


Figura 2.19: Ejemplo de un Conmutador 2x2.

Para lograr esto, se provee de un registro de desplazamiento, el cual garantiza que todos los L buffers estén igualmente cargados y óptimamente usados, a su vez, se garantiza la secuencia de celdas. El registro es necesario, por que las p celdas ganadoras llegan hacia el buffer de la izquierda (si $p < L$).

Este registro de desplazamiento circular de L entradas y L salidas, procede con el traslado, tal que los L buffers estén llenos, en forma cíclica. Esta operación se muestra en un esquema anterior, en el cual en el primer tiempo de celda o slot, llegan 5 celdas llenas hacia las 5 primeras entradas del registro. Durante este tiempo, las 5 celdas son directamente trasladadas a través de las 5

primeras salidas. Durante el siguiente tiempo de celda, llegan 4 celdas llenas, otra vez hacia las 4 primeras entradas del registro. Esto es, porque el concentrador siempre concentra hacia sus primeras salidas. Durante este tiempo de celda, la primera celda será trasladada hacia la salida 6, la segunda a la 7 y así sucesivamente. Durante el siguiente tiempo de celda, la primera celda será rotada hacia el segundo buffer. De tal modo, que todos los buffers sean llamados uniformemente. También, la secuencia de celdas puede ser garantizada por este registro para una adecuada implementación. En forma mas genérica, se puede describir el registro en la forma siguiente:

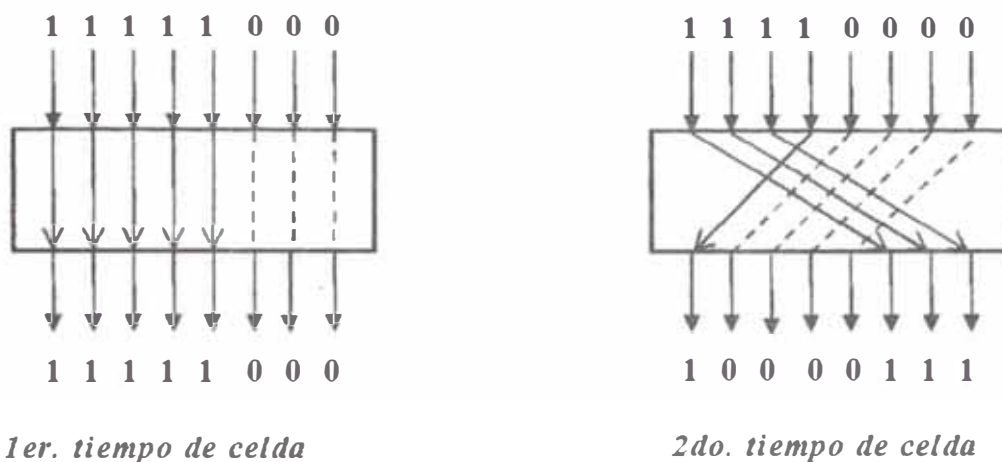


Figura 2.20: Ilustración de la función de desplazamiento.

Si S_i describe el número de posiciones, el registro tendrá que rotar hacia la derecha durante el tiempo de celda y, entonces S_{i+1} será igual a:

$$S_{i+1} = (S_i + K_i) \text{ mod } L \quad (25)$$

Si K_i representa el número de celdas que llegan durante dicho tiempo, se puede tomar $S_1=0$. un registro puede ser construido con $(L/2)\log_2 L$ 2x2 conmutadores elementales, controlado por una máquina que obedece a la ecuación (25).

2.8. Implementación de un conmutador genérico ATM.

Los bloques de conmutación son interconectados entre sí en una red y obtenemos un conmutador genérico enlazado por bloques de conmutadores básicos. Estos conmutadores tienen un gran número de entradas y salidas (cientos

y/o miles). Con frecuencia, dicho conmutador está compuesto de un gran número de bloques de conmutación idénticos, tal como se ha descrito en la sección anterior. Estos conmutadores están referidos como redes de interconexión multiestado (MIN).

En la figura No. 2.21, se representa un conmutador de 2x2, organizado en tres estados y cada estado de 4 conmutadores elementales. la interconexión de los estados es tal que las 8 entradas al sistema deben alcanzar la salida. Existen patrones de interconexión que plantean características interesantes de los MIN, pero, dado que el tema del presente trabajo no es un sistema MIN, sólo se señala como referencia conceptual, que se pueden tener conmutadores de 32x32 y otros más grandes.

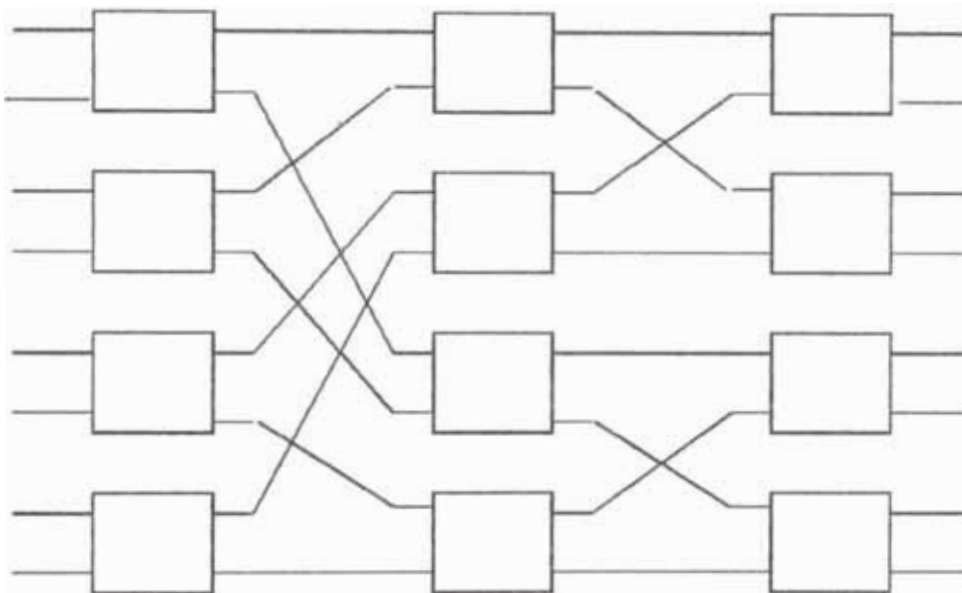


Figura 2.21: Interconexión de red multiestado.

En caso del sistema Knockout, que no es un MIN, se tiene una capacidad de expansión mediante el incremento de L entradas adicionales hacia cada concentrador, tal que se convierte en $(N+L)$ hacia L concentradores. En el esquema posterior, se muestra un conmutador Knockout de $2N$ entradas y $2N$ salidas, construido con $N \times N$ bloques de conmutación básicos del Knockout. Se requieren sólo $2N$ buffers y $2N$ Buses, también se requieren $2 \times 2N$ filtros y concentradores. Todos los concentradores tienen $N+L$ entradas utilizables y L salidas. Sólo $2N$ de estos concentradores usan estos $N+L$ entradas efectivamente,

los otros $2N$ concentradores usan sólo N de estas entradas. Este principio de expansión puede aplicarse para valores muy grandes, pero teniendo cuidado que el número de buses y el número de entradas por concentrador no sea muy grande.

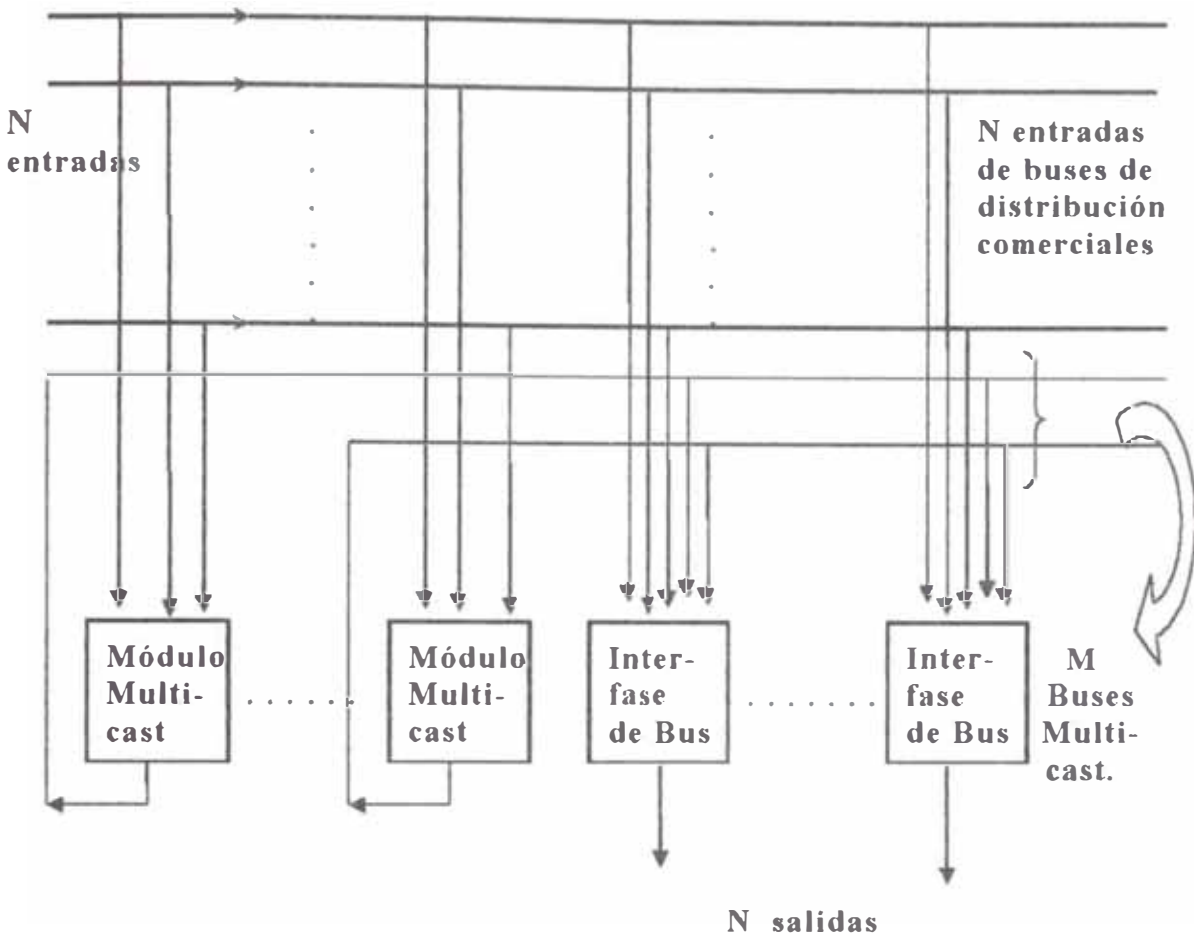


Figura 2.22: Conmutador Knockout con módulos multiestado.

2.9. Enrutamiento.

Como el sistema ATM es orientado a conexión, los valores o códigos de cabecera son asignados a cada sección de una conexión para una duración completa de la conexión y traducida cuando se conmuta de una sección a otra. La señalización y la información del usuario son transportadas en canales virtuales separados.

Dos formas de conexiones son posibles, es decir, las conexiones de canales virtuales (VCC) y las conexiones de rutas virtuales (VPC). Un VPC puede ser considerado como un agregado de los VCCs. Cuando se conmutan o multiplexan celdas, se producen dichas conexiones. Primeramente se realizan los que están basados en el VPC, luego después los del VCC. En la figura 2.23

se puede observar un bloque que solamente realiza la conmutación VP y otro que realiza la conmutación de ambos, es decir del VP y VC. Sin embargo, la conmutación de la parte VP puede no ser considerada o descartada, resultando en un conmutador puramente del VC.

El proceso de conexión se establece temporalmente o mientras dure todo el proceso de llamada o transferencia de información. Dicho establecimiento incluye la ubicación de los identificadores de los VCI y/o VPI. Del mismo modo, la ubicación de los recursos requeridos en el acceso del usuario y dentro de la red. Estos requerimientos están expresados en términos de transferencia (tasa de bits) y calidad de servicio.

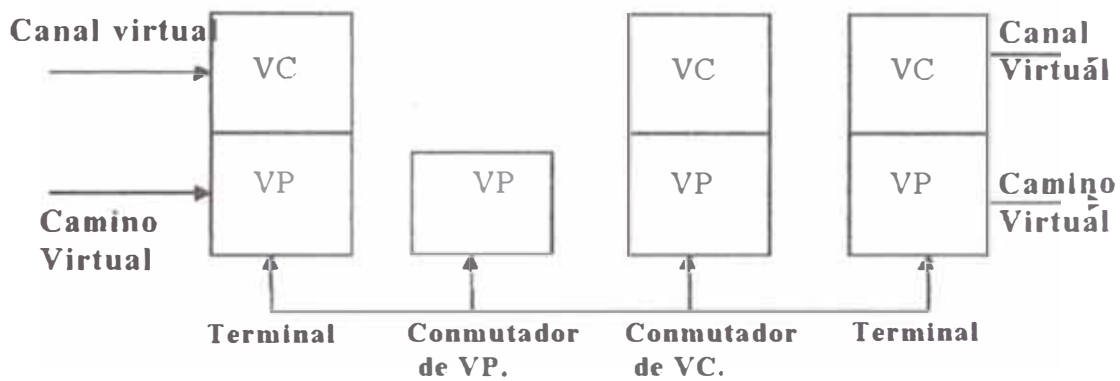


Figura 2.23: Conexión de los VC y VP en conmutadores ATM.

RESUMEN

A pesar de los avances en el diseño de conmutadores de paquetes ATM de $N \times N$, su tamaño ha quedado limitado tanto por las restricciones tecnológicas y físicas. En lo físico fundamentalmente es difícil implementar los circuitos impresos o fabricar circuitos integrados que tengan la capacidad de integrar la cantidad de conmutadores elementales. Por tanto surge un desafío en investigar, desarrollar y entender cómo construir un conmutador de grandes dimensiones pero de tamaño moderado y sin sacrificar su nivel de retardo, el throughput y su desempeño. Por lo que, en el presente capítulo se han desarrollado los conceptos básicos considerados necesarios para el mejor entendimiento del tema, como son el concepto de la conmutación ATM, su estructura básica basada en los procesos de concentración, multiplexación, demultiplexación y expansión respectivamente. Del mismo modo se describen los conceptos de la probabilidad de inserción y pérdidas de celdas, retardos de conmutación y criterios de colas de espera.

Se hace un análisis básico para evaluar las ventajas de tener buffers de cola de espera en una determinada ubicación del conmutador, así puede estar en la entrada, en posición intermedia o a la salida del conmutador; resultando más ventajosos los buffers ubicados en la salida del conmutador. Del mismo modo, se definen los parámetros necesarios para la implementación de un conmutador ATM modular y escalable.

Estando el tema de la tesis basada en un conmutador Knockout, se describe en forma breve su principio de operación como tal y su arquitectura para conmutación de paquetes.

CAPÍTULO III

ARQUITECTURAS DE CONMUTADORES ATM DE GRANDES DIMENSIONES

3.1. Introducción

Como se ha visto en el segundo capítulo, los sistemas de conmutación vienen siendo estudiados desde hace muchos años, pero el interés en el sistema de conmutación ATM es prácticamente recién desde la década pasada. En contraste a los circuitos de conmutación con patrones de tráfico definidos, un conmutador de paquetes debe ser diseñado para manipular más de un paquete en la entrada destinada para una misma salida, creándose un conflicto para la salida. Este escenario presentó, desde 1992, una oportunidad de desafío para la investigación, tanto en el arreglo de los buffers como en la estructura de conmutación interconectada.

Muchos conmutadores de grandes dimensiones, se han diseñado “en el papel”, pero se han presentando inconvenientes de tipo tecnológico e implementación física con el tamaño. Por ejemplo se han diseñado e implementado en un circuito integrado VLSI con pines de entrada y salida restringida a valores menores a 1000×1000 ; pero, Kai Y, Eng, Mark J. Karol y Yu-Shuan Yeh [1], propusieron un sistema de conmutación de $2M \times 2M$, basado en el principio Knockout [9], el cual generaliza dicho principio Knockout, logrando reducir los requerimientos de interconexión de paquetes y mejorando el retardo, eficiencia de salida y el desempeño.

Del mismo modo, se han propuesto muchas otras arquitecturas, pero existen una cierta gama de proposiciones basadas en la red de CLOS (inclusive el Knockout está basada en la red de 3 estados), entre otros, por ejemplo, Martín Collier [17], propone un sistema de tres estados, basado en investigaciones previas y en donde el desempeño en cualquier punto e instante del sistema de conmutación ATM (en bits/segundo) depende significativamente del proceso

usado en la fabricación. También, Bianchini y Kim [18] describieron un sistema prototipo con 155 Mbps de velocidad en transferencia de datos en enlazamiento y un throughput de 2,48 Gbps, construido empleando circuitos integrados “off-the-shelf”, pero cuya base estaba en la arquitecturas de redes de Banyan mostrado en la Figura 2.10 del capítulo II.

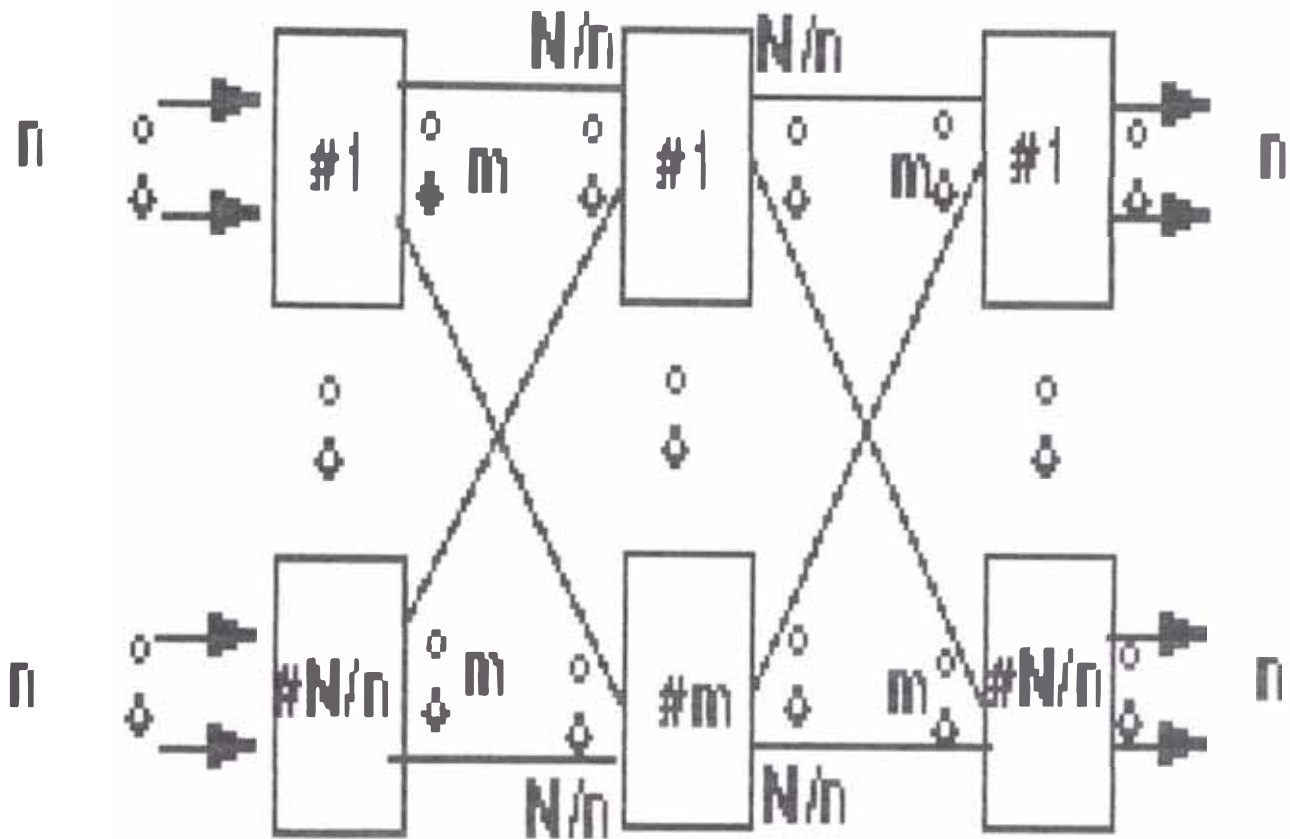


Figura 3.1: La Red de Clos (N, n, m).

Existen muchas formas de interconectar los módulos de conmutadores. El más común de todos ellos es la interconexión **multiestado** de módulos de red. En la Figura 3.1 se ilustra la red de 3 estados de Clos [3][20][31][39], que es un ejemplo típico de dicha modalidad de interconexión.

En la figura 3.1 existen N/n módulos de conmutación en el primer estado y cada uno de tamaño $n \times m$. Por tanto el segundo estado contiene m módulos y cada uno de tamaño $N/n \times N/n$ y el último estado tiene otra vez N/n módulos de tamaño $n \times m$. El tráfico puede balancearse toda vez que esta configuración provee m rutas distintas entre algún par de entradas y salidas. Debido a que cada celda puede tomar una ruta independiente, la secuencia de celdas puede ser recuperado en las

salidas. Usualmente se selecciona el camino menos congestionado durante el proceso de conexión. Un nuevo requerimiento no puede ser aceptado si la red internamente está congestionada. Las redes de Clos son estrictamente **sin bloqueo** si es que siempre existe un camino disponible entre una entrada libre y un par de salidas. Dado que en un sistema ATM el ancho de banda usado para la conexión varía para diferentes tiempos, considerar la condición de **no bloqueo** es una acción de mucha importancia.

De los estudios e investigaciones anteriores, un conmutador ATM debe estar restringido a un sólo circuito integrado y una sola tarjeta, además, estaría mejor si es restringido a un throughput global del orden de 40 Gbps a 200 Gbps, lo que implica usar tecnologías de integración caras y arquitecturas modulares, para que de esa forma se pueda distribuir a través de tarjetas o gabinetes. En el presente capítulo, se describen un par de arquitecturas propuestas en base al Knockout [9] y el mismo Knockout pero generalizado, dado que en el primer capítulo ya se ha detallado la arquitectura Knockout. Algunas de las arquitecturas que toman como base a la arquitectura Knockout son tema del presente trabajo, así se tiene, por ejemplo, la arquitectura doblada o duplicada y la arquitectura PINIUM [2], las que se describen en el presente capítulo.

3.2. Arquitectura Knockout Generalizada.

El principio Knockout generalizado está basado en tres principios básicos:

- a) El principio Knockout generalizado a grupos de salidas para reducir la complejidad de interconexión de paquetes simultáneos,
- b) Que el sistema con buffer en la salida tenga un efecto de mejoramiento de la característica retardo/throughput y el desempeño, y
- c) Un sistema de enrutamiento adecuado que no permita una congestión interna y a su vez reduzca la complejidad de interconexión de los módulos.

Mientras el adecuado enrutamiento de paquetes limita la complejidad de la interconexión de los conmutadores modulares, éste aspecto puede a su vez convertirse en un cuello de botella, dado que la asignación de enrutamiento tiene que ser hecho paquete por paquete. El tiempo de asignación requerido para

manejar todos los paquetes que entran es proporcional al tamaño del conmutador y esta duración también representa el tamaño mínimo del paquete permitido en el conmutador. Afortunadamente, en la actualidad existen algoritmos muy rápidos y que pueden ser implementados; por ejemplo, para una velocidad de línea de 150 Mbps se tiene la tecnología VLSI para un conmutador de 1000 x 1000 en períodos menores a 300 bits [1].

Sin embargo en la práctica, todos los paquetes no llegan a la entrada del conmutador sincrónicamente, existen interfaces modulares de entrada que proveen sincronización tal que los paquetes de longitud fija ingresen al conmutador en forma sincronizada en tiempos de celda [9]; y los paquetes de longitud variable ingresen al conmutador en forma sincronizada como si fueran mini-celdas [30]. La compatibilidad del conmutador con celdas de longitud variable elimina la necesidad de normalizar el tamaño de las celdas y hace que el conmutador tenga capacidad universal de transporte.

3.2.1. Generalización del principio Knockout.

Tal como se explicó en la sección anterior, en el principio Knockout se tiene un $L=8$ que nos permite tener una probabilidad de pérdida de paquetes por debajo de 10^{-6} , para cualquier conmutador de tamaño N con 90% de carga. En esta sección se explica la generalización del sistema Knockout a un grupo de salidas. Se considera entonces un grupo de n salidas y se les trata como si fueran una sola. Por tanto, de acuerdo a las consideraciones ya explicadas, la probabilidad de que un paquete está destinado a este grupo de salidas será simplemente n/N . Si se considera que sólo se permite que más de m paquetes pasen hacia los grupos de salida, entonces se tendrá que la probabilidad de pérdida de paquetes (P_r) esté dada por:

$$P_r = \frac{1}{n\rho} \sum_{k=m+1}^N (k-m) C_k^N \left(\frac{n\rho}{N} \right)^k \left(1 - \frac{n\rho}{N} \right)^{N-k} \quad (26)$$

y cuando $N \rightarrow \infty$ se tendrá:

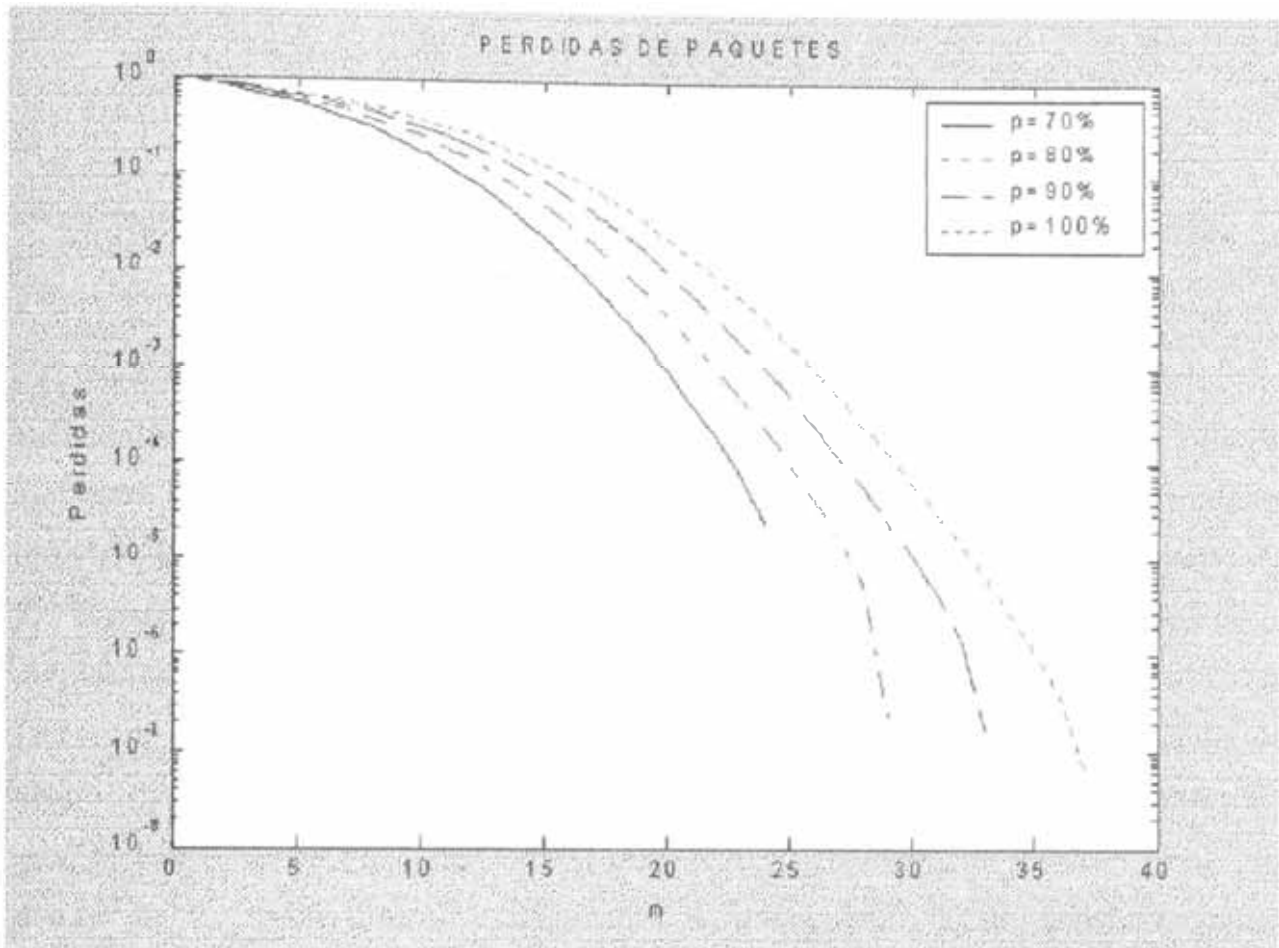


Figura 3.2: Probabilidad de pérdidas de paquetes mediante el principio del *Knockout Generalizado* para valores de m desde 0 a 40.

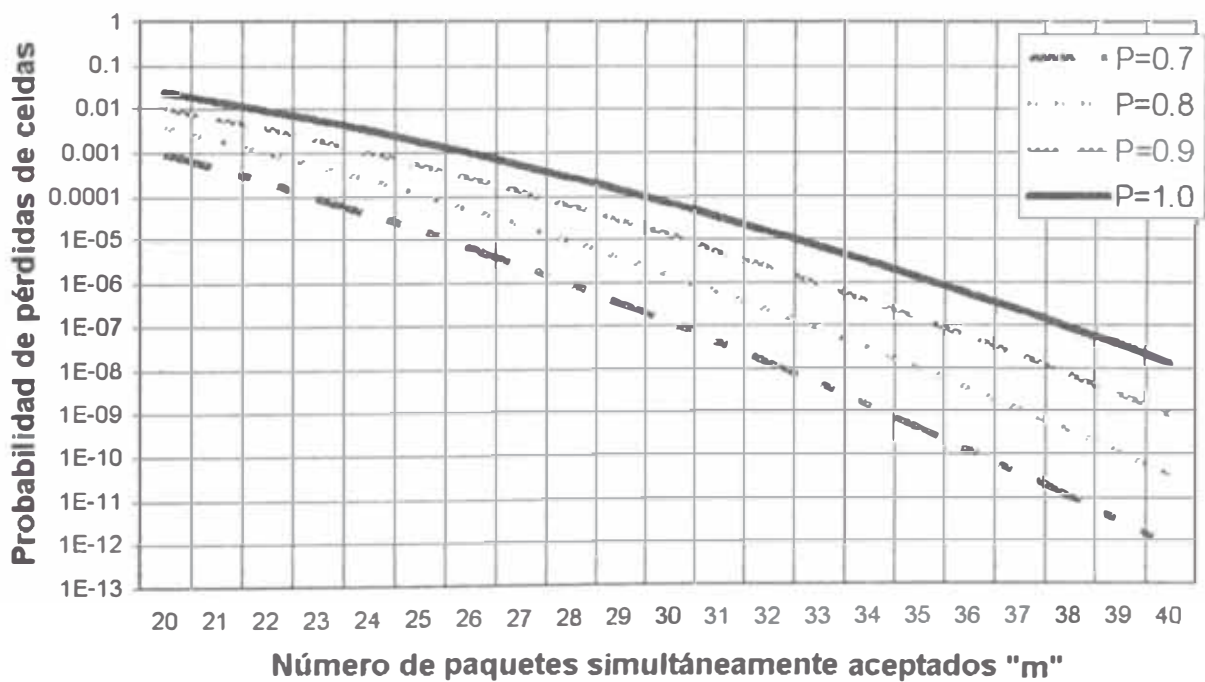


Figura 3.3: Probabilidad de pérdidas de paquetes mediante el principio del *Knockout Generalizado*.

$$P_r = \left[1 - \frac{m}{n\rho} \right] \left[1 - \sum_{k=0}^m \frac{(n\rho)^k e^{-n\rho}}{k!} \right] + \frac{(n\rho)^k e^{-n\rho}}{m!} \quad (27)$$

En las figuras 3.2, 3.3 y 3.4 se grafican la probabilidad de pérdidas de paquetes en función de m para diferentes valores de la carga con $n=16$ y $N \rightarrow \infty$; en cualquiera de dichas gráficas se pueden notar por ejemplo que $m=30$ es suficientemente grande para mantener una probabilidad de pérdida por debajo de 10^{-5} para una carga del 90%. En el caso de que los grupos de salida son tratados individualmente, el total de paquetes requeridos sería 128 que es el resultado de 8×16 para la misma probabilidad de pérdida. En la figura 3.5 se ejemplifica la mejora que resulta del agrupamiento de salidas, en dicha figura se grafica el factor de expansión m/n versus un rango práctico de n , que es el tamaño de grupo, para diferentes valores de pérdidas. En tal figura se puede notar que para una probabilidad de pérdida de 10^{-6} , m/n decrece rápidamente desde 8 hasta por debajo de 2,5 para tamaños de grupos n mayor que 16; un comportamiento similar es evidente para otras probabilidades de pérdidas de celdas. Este principio del Knockout generalizado forma la parte básica de la teoría del conmutador de grandes dimensiones.

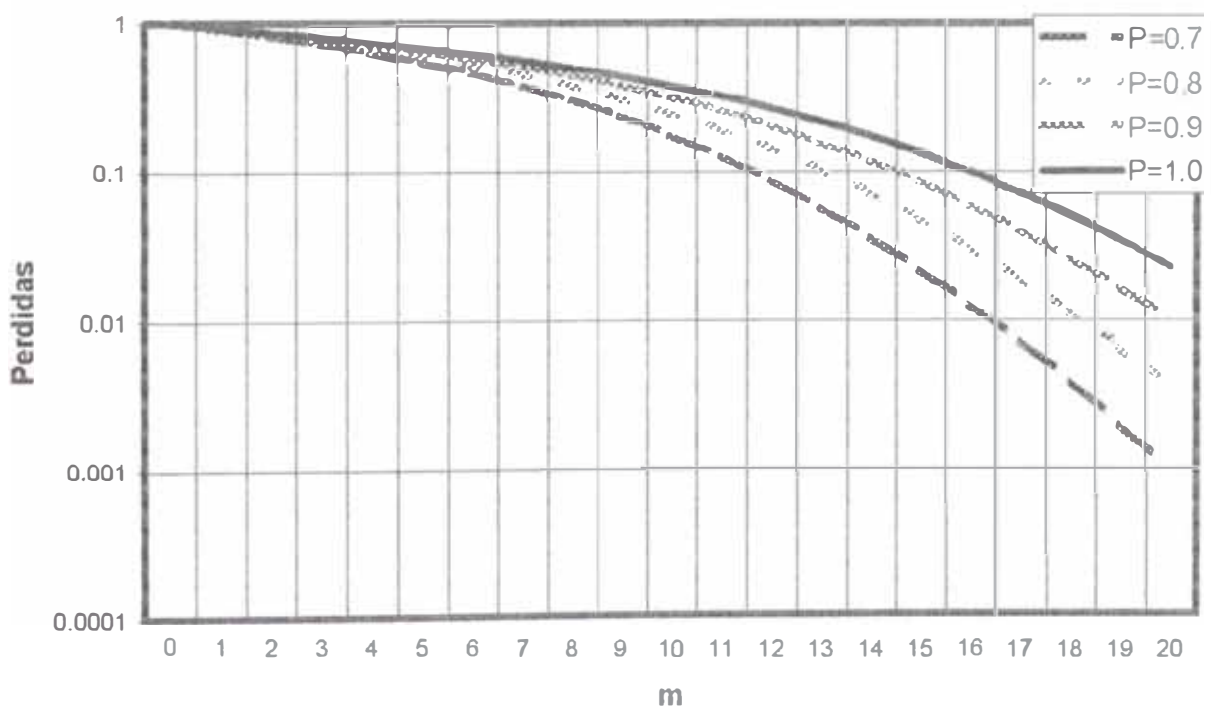


Figura 3.4: Lo mismo que la figura anterior, pero para valores m de 0 a 20.

3.2.2. Disciplina de colas de salida.

Cuando muchos paquetes o celdas llegan simultáneamente para una misma salida en un conmutador $N \times N$, deben ser bloqueados y almacenados ordenadamente en un buffer de salida, dado que sólo un paquete debe salir por vez (tiempo de celda). Todos los paquetes son conducidos a través de los módulos de conmutación hacia sus respectivos destinos, de modo tal que todos ellos sean almacenados en sus respectivos buffers de colas de espera en la salida, evitando así que sean retardadas por paquetes destinados a otras salidas.

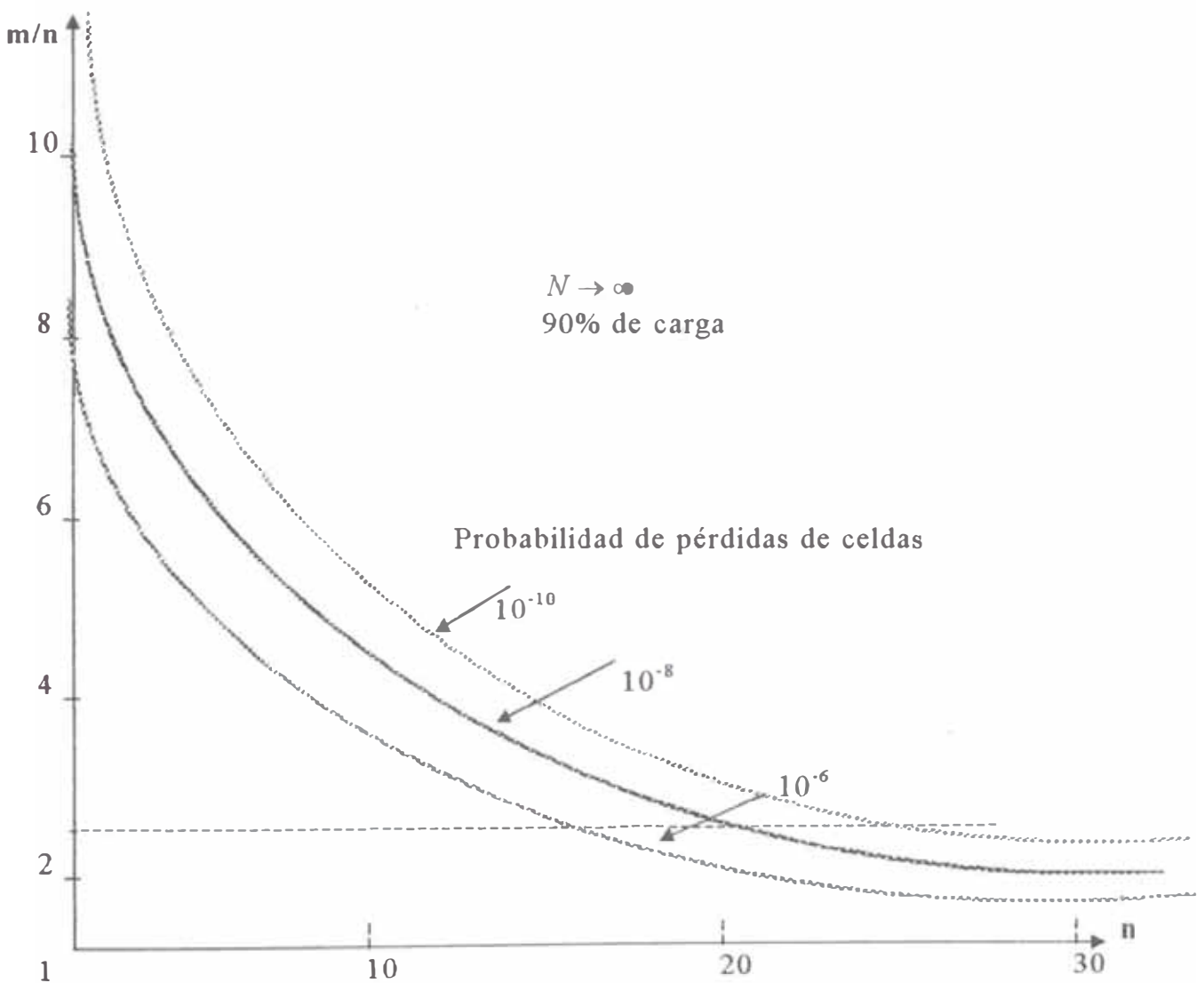


Figura 3.5: Tasa de celdas simultáneas aceptadas para un grupo de tamaño n .

Empleando el sistema Knockout generalizado y considerando que se selecciona una aceptable probabilidad de pérdida de celdas y aplicando las relaciones matemáticas (26) y (27), se determina un valor apropiado de m para un grupo de salidas dadas y cuyo tamaño es n . Luego, proveyendo más de m

paquetes puede enviarse a través de todo el sistema interconectado hacia cada salida de grupo, donde un módulo de conmutación de paquetes ideal (de m entradas y n salidas) con los buffers de cola de espera en la salida, están disponibles para la recepción; y todo el sistema conmutador habrá logrado en la salida un mejor desempeño y mínimo retardo y mejor throughput. Esto es ilustrado en la figura 3.6 donde el módulo de conmutación de paquetes para cada grupo de salida es denominado **módulo de paquetes de salida**.

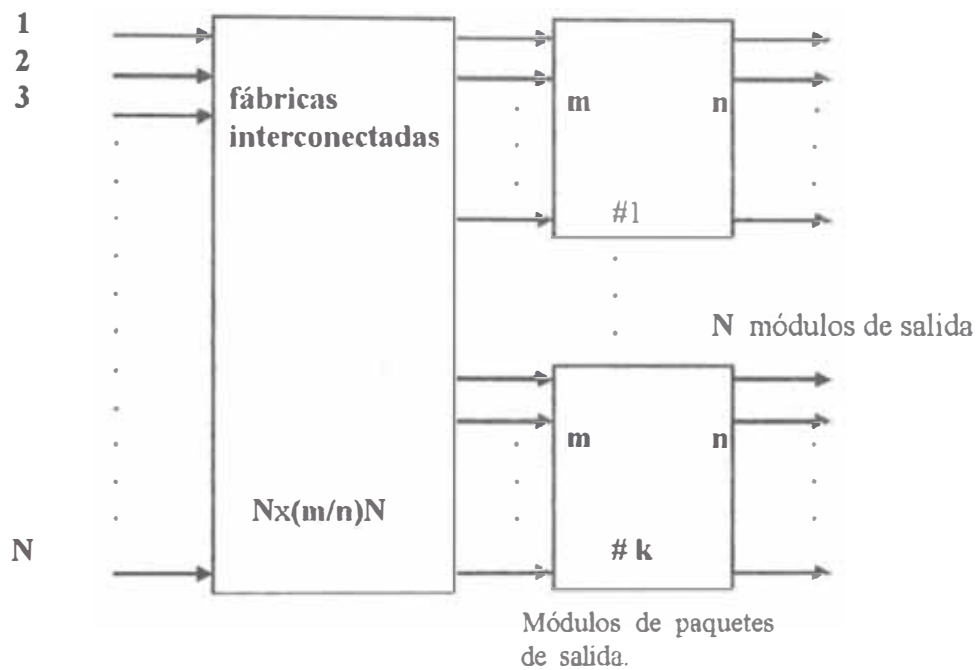


Figura 3.6: Arquitectura del conmutador que emplea el principio del Knockout Generalizado.

En la arquitectura mostrada en la figura 3.6, es necesario enfatizar en las siguientes consideraciones:

- El parámetro m en cada módulo de conmutación de paquetes de salida (de m entradas y n salidas) es independiente del tamaño N del conmutador. Consecuentemente, la sección de salida puede crecer indefinidamente mediante el empleo de más y más módulos de tamaño fijo e idénticos.
- No hay buffers en el interior del sistema de conmutación, por tanto su función es solamente transportar más de m paquetes hacia cada módulo de salida.

- Todo el almacenamiento temporal (en el buffer) de todos los paquetes es llevado a cabo en los módulos de conmutación de paquetes de salida (de m entradas y n salidas), por lo cual cualquier diseño de un buffer de cola de espera de salida debe asegurar un óptimo desempeño y mínimo retardo a través de todo el conmutador.

3.2.3. La Fábrica interconectada

A pesar de que podrían haber más de N paquetes entrantes, todos ellos destinados para un grupo de n salidas, cada uno de los módulos de conmutación de salida (de m entradas y n salidas) pueden aceptar sólo más de m paquetes ($m \ll N$). En otras palabras, usando el principio del Knockout generalizado[1], los módulos interconectados sólo tienen que proveer conexiones entre N entradas y mN/n salidas. En diseños prácticos con $m \approx 2,5n$, el número de salidas es entonces aproximadamente $2,5N$. Existen al menos dos diseños posibles. Primero, podemos diseñar un sistema modular interconectado con auto-encaminamiento (de N entradas y mN/n salidas), en el cual la selección de m posibles paquetes para cada uno de los grupos de salida es hecho automáticamente. Por ejemplo, usando buses comerciales más concentradores de N entradas y m salidas [9] u otros sistemas. Alternativamente, podemos usar un conmutador auto-enrutado simple pero rápido sin memoria temporal (buffer), un controlador de asignación de ruta para proveer un balance inteligente de carga. Este último es beneficioso debido a su escalabilidad, fácil implementación y flexibilidad.

Más adelante se expondrá cómo se hace la selección de paquetes y enrutamiento de las mismas. Por ahora, se asume que la mayoría de los m paquetes que llegan están destinados para un grupo específico de salida en un tiempo de celda. Un ejemplo que satisface las conexiones requeridas de $N:mN/n$ es la que se muestra en la figura 3.7. La estructura del conmutador se parece a la expuesta en [31], excepto en que el estado de la salida consiste de módulos del conmutador de paquetes.

Debido a la asignación inteligente de rutas, tal como se describe más adelante, no hay conflicto de vías o caminos mientras los paquetes son rotados a través del sistema conmutador interconectado (representado por los dos primeros

estados). La existencia de la asignación de dichas rutas está garantizada por la propiedad de no bloqueo de una red de tres estados para $m \geq n$. Cada uno de los conmutadores en el sistema interconectado son conmutadores elementales, auto-encaminados y dispositivos interconectados con poca memoria. Existen varias formas de implementar estos dispositivos. La problemática está en cómo hacerlos crecer de acuerdo a las necesidades y exigencias de un mercado cada vez más grande. Afortunadamente, el problema de crecimiento es idéntico al de los circuitos de conmutadores convencionales.

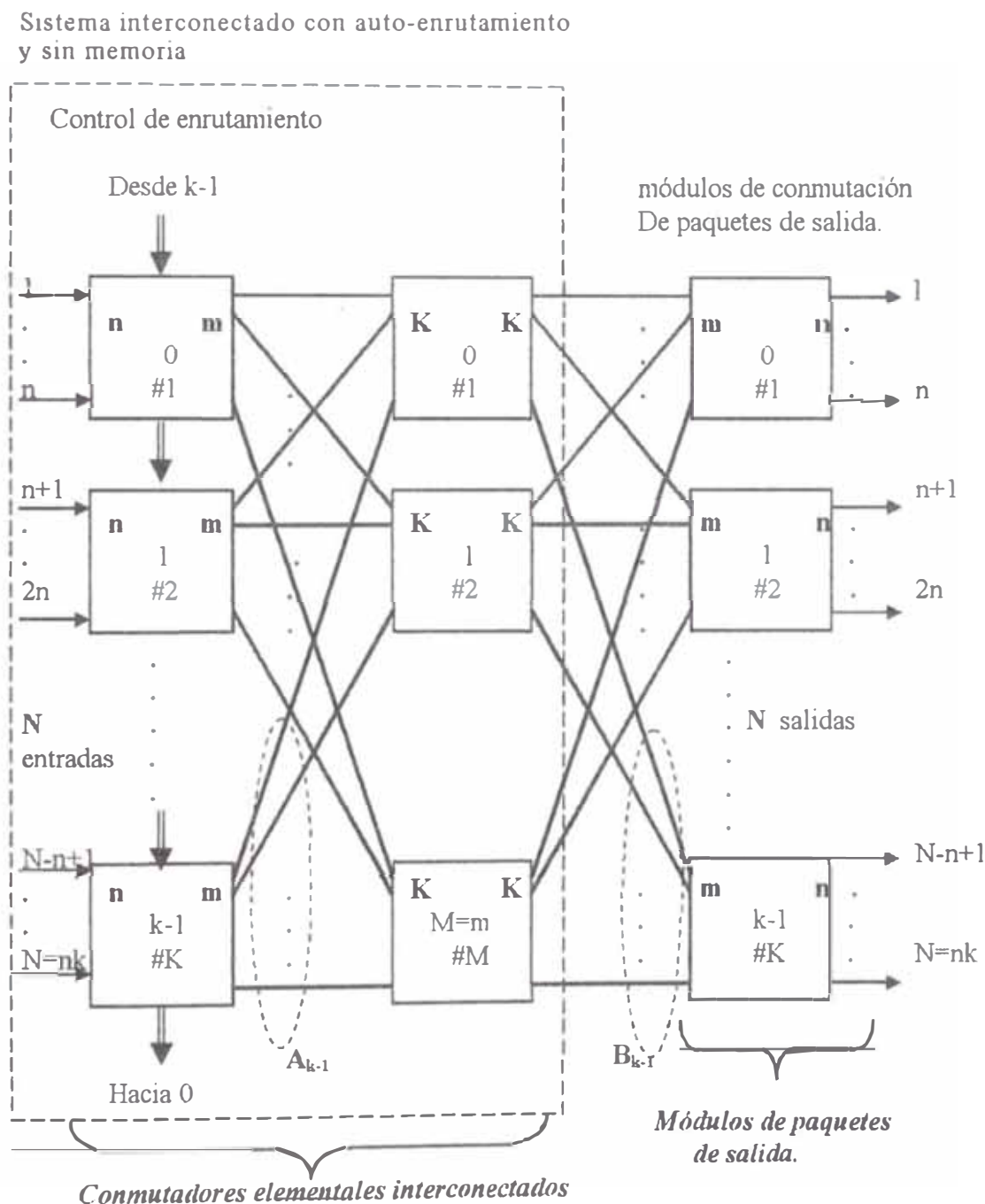


Figura 3.7: Arquitectura de un conmutador escalable.

La figura 3.7 es un ejemplo práctico para mostrar enrutamiento de distintos paquetes a través de conmutadores elementales interconectados. Su implementación actual toma diversas formas tecnológicas. Por ejemplo, en vez de tener $mK \times K$ módulos intermedios, se pueden usar $m/2$ (para $t=2$) de $(2K \times 2K)/2$ módulos intermedios con dos líneas dirigidas a cada módulo de salida (o entrada), tal como se muestra en la figura 3.8. Esta variación [1] y [4], denominada estado intermedio principal, tiene mejor determinación. Sin embargo, su mejora no es fina, pero sí genera mayor costo en hardware.

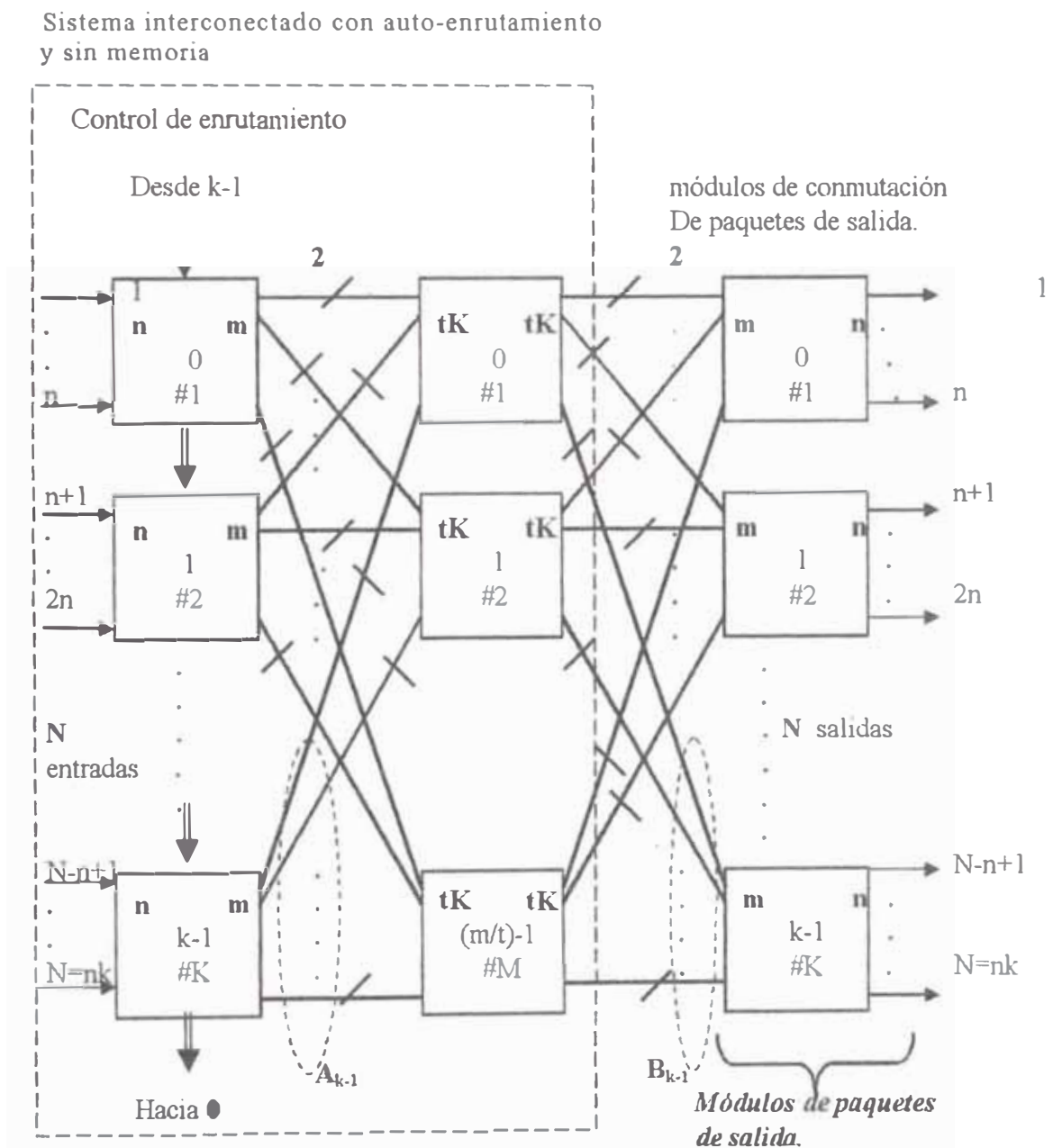


Figura 3.8: Arquitectura modificada de un conmutador escalable con ruta definida.

En el siguiente capítulo, se hace un análisis más detallado de las relaciones matemáticas por que tienen que ver con el problema de enrutamiento para el sistema de conmutador propuesto, lo que nos permitirá una mejor visión comparativa con otras arquitecturas propuestas y que tengan que estar relacionado casi en forma directa con el sistema Knockout propuesto en [9] en 1987 y cuya mejora se está enfocando en la presente tesis, de acuerdo a las investigaciones que nos muestran los autores [1] y [4] para una posible implementación de dicho conmutador.

3.3. Arquitectura ATM plegable de un conmutador escalable.

Uno de los requerimientos actuales para cualquier conmutador ATM es su capacidad de crecimiento. Es decir, la arquitectura de construcción debe permitir un crecimiento modular de su tamaño, a partir de un pequeño número de puertos hasta tamaños de conmutadores muy grandes. La arquitectura propuesta por Andrzej Jajszczyk y Wojciech Kabacinski [3], es aquella en la que es posible expandir el tamaño de un conmutador de dos bloques de entradas y sin modificar la instalación de cables existentes.

3.3.1. Arquitecturas Plegables [3]

La escalabilidad es una característica común de las arquitecturas agrupadas que emplean una memoria de conmutación. En estas arquitecturas, algunas partes del sistemas transportan información en una dirección y las mismas transportan información en sentido opuesto[32]. El concepto de expansión en cuanto a su tamaño se ilustra en la figura 3.9.

En dicha figura se asume que el conmutador elemental tiene la capacidad de 16 enlaces bidireccionales. Si sólo algunos de los puertos son necesarios, la fábrica consiste de un simple conmutador elemental (denotado por A). Para lograr una de alta capacidad mayor que 8 puertos, el segundo estado debe ser adicionado y el número del primer estado de conmutadores elementales tiene que ser incrementado. Por tanto, se obtiene el bloque B, conteniendo más de 64 puertos. El rendimiento requerido es logrado adicionando un número apropiado de conmutadores elementales al segundo estado. La fábrica de máxima capacidad, denotado por C, es obtenido análogamente. Si se asume que la última fábrica

consiste de más de 3 estados, puede emplearse la capacidad total (16 puertos) de los elementos del tercer estado para conectar a los módulos de dos estados, sin dejar algunos puertos para la conexión del siguiente estado. Debe notarse que el presente método de expansión ya no genera disturbios en enlaces existentes.

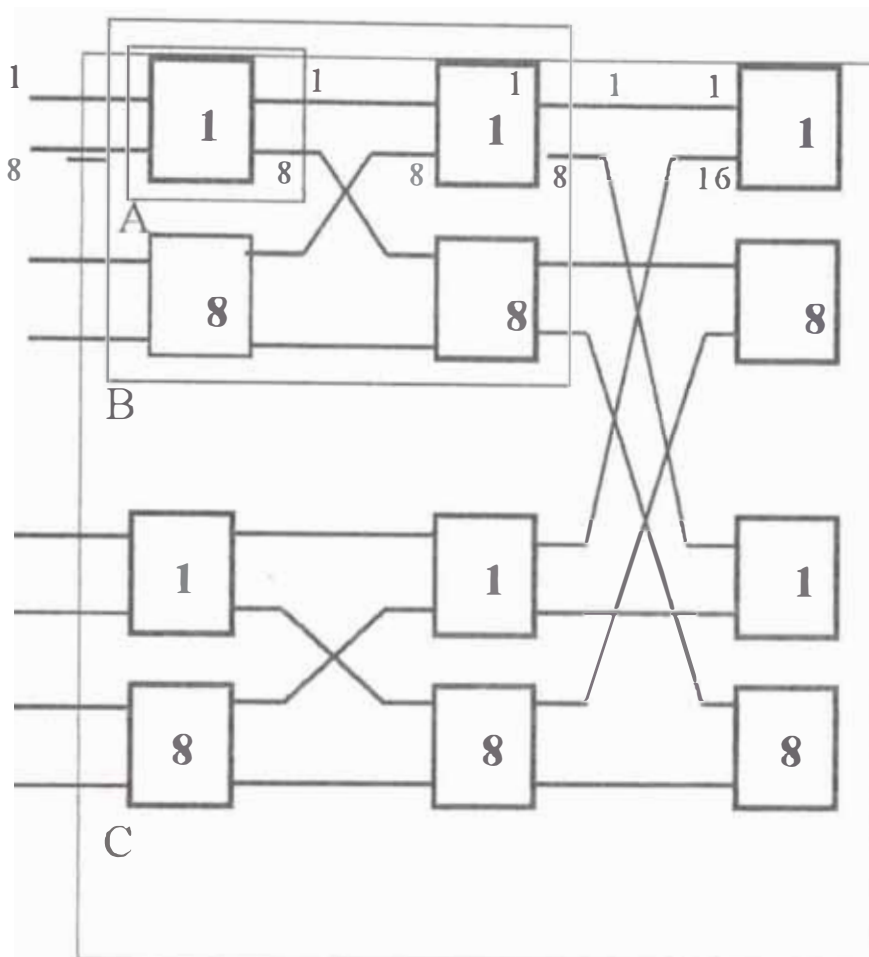


Figura 3.9: Escalabilidad de un "switching fabric" plegable.

En arquitecturas plegadas, las celdas son encaminadas a través de los diferentes conmutadores elementales, y dependen del direccionamiento de los puertos de entrada y salida. Por ejemplo, si una celda va a ser enviada entre dos puertos pertenecientes a un mismo conmutador elemental, sólo este conmutador elemental es responsable del proceso de conmutación. Una comunicación entre puertos del mismo módulo de dos estados, requiere de diferentes conmutadores elementales para encaminar las celdas hacia el segundo estado, desde donde las celdas son retornadas (reflejadas) hacia el primer estado. La comunicación entre los módulos hace que el punto de reflexión (retorno) se mueva hacia el tercer

estado. Por ejemplo una celda puede atravesar 5 conmutadores elementales desde el puerto de entrada para llegar hacia un puerto de salida.

El concepto descrito en el párrafo anterior es empleado en los diversos conmutadores elementales ATM o en arquitecturas de conmutadores rápidos, aunque algunos de ellos emplean patrones de interconexión diferentes entre estados. El concepto ha sido usado en el sistema Athena, Roxanne [6] [27] [32][33] y otros. La construcción de un conmutador está compuesto de tarjetas modulares de conmutación normalizadas de 128 entradas por 128 salidas de 150 Mbps cada uno. Físicamente, 4 enlaces de 150 Mbps son multiplexados a 600 Mbps, resultando en una tarjeta de conmutación modular de 32x32. Cada tarjeta estará compuesta de un número de conmutadores elementales integrados con buffers compartido de 32x32. Su desarrollo tecnológico al presente, resultó en una solución interina basada en un conmutador elemental integrada de 16x16 y tarjetas de 64x64. [3][27] [35].

3.3.2. Fábrica de 3 estados y 2 lados.

Muchos sistemas experimentales en la construcción de conmutadores ATM emplean una estructura de fábrica de 3 estados y dos lados [11][27], que tiene algunas ventajas respecto a la arquitectura plegable, incluyendo igual número de conmutadores elementales atravesados por cada celda (en el sistema plegable, este número varía, dependiendo de la selección del par de entradas y salidas) como si fuera un simple proceso de enrutamiento y mantenimiento.

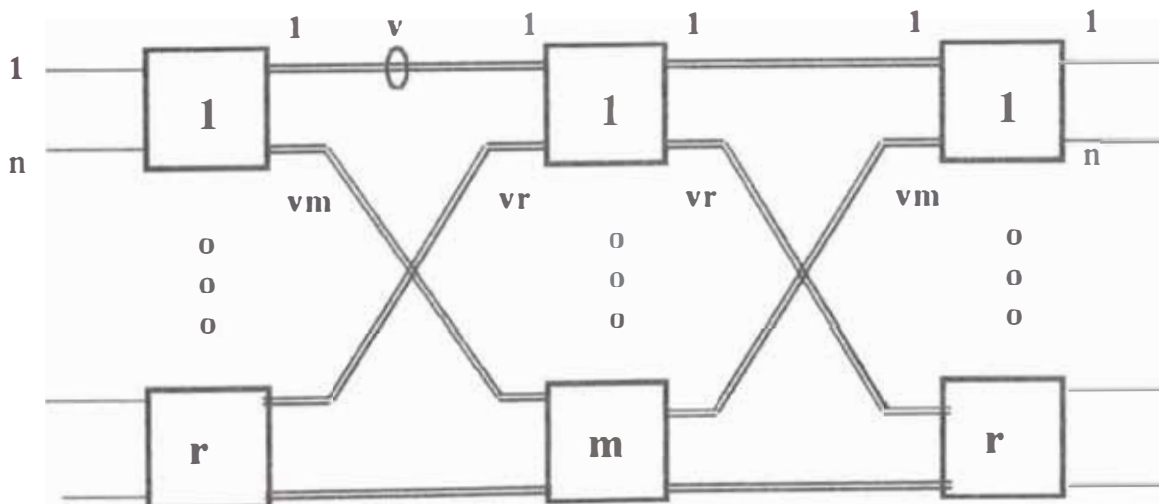


Figura 3.10: Fábrica de dos lados y tres estados.

Una transmisión unidireccional dentro de cada conmutador ATM elemental, reduce también los problemas de “crosstalk” que sí se presenta en los conmutadores bidireccionales usados en la arquitectura plegable. Sin embargo, su mayor problema es su escalabilidad. Es conocido por la teoría de la conmutación de redes, que el tamaño de la red de Clos puede expandirse reemplazando los conmutadores de estado intermedio por redes completas de 3 estados, como consecuencia obteniendo 5, 7, etc. estructuras de estados. Sin embargo, en la aplicación práctica en el sistema ATM, implicaría volver a cablear en forma masiva, generando una interrupción en el servicio.

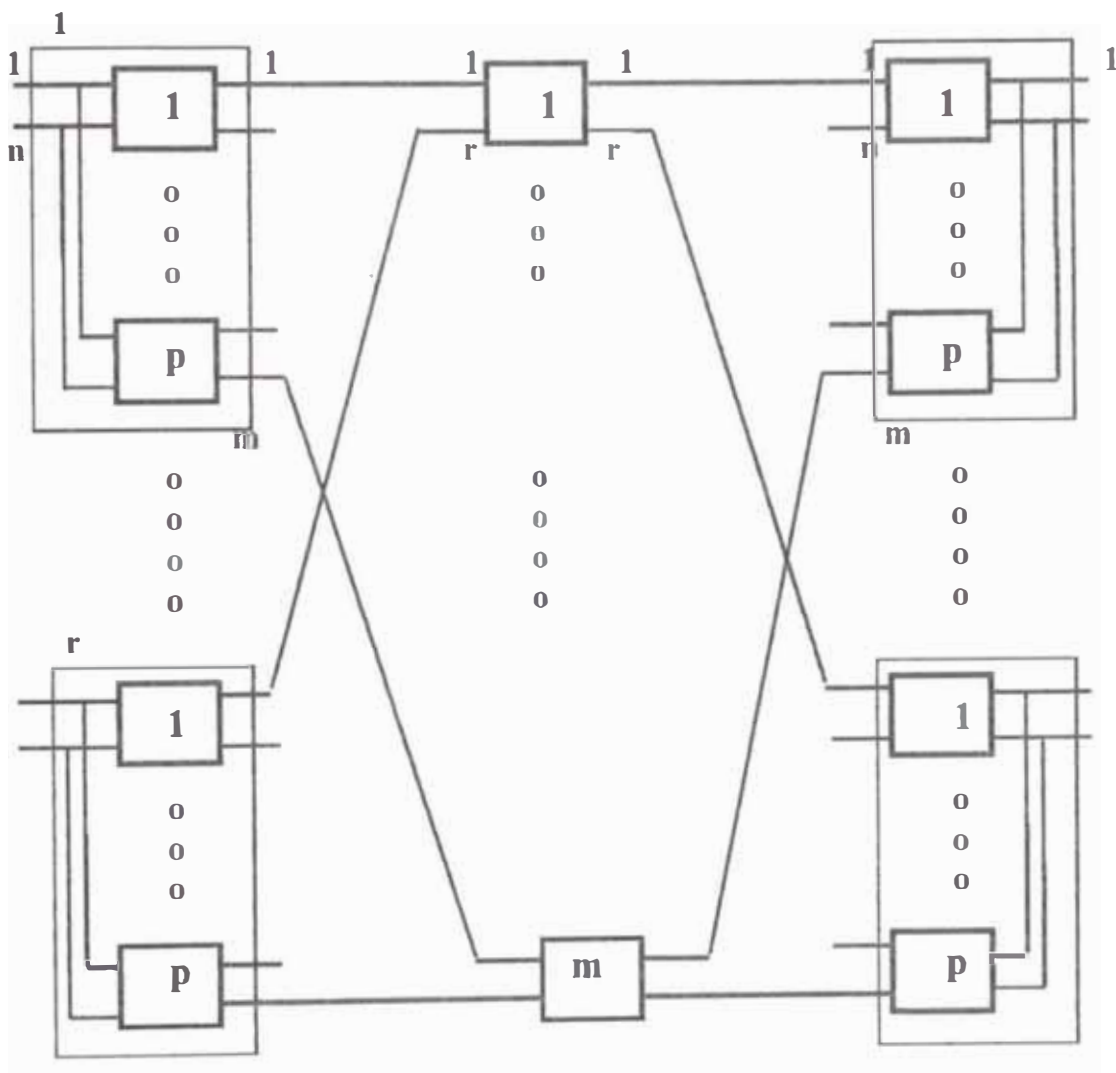


Figura 3.11: Una fábrica modificada de 3 estados.

El esquema básico del conmutador ATM (fábrica) de tres estados y dos lados es el que se muestra en la figura 3.10 [37]. La estructura es usada para

aplicaciones en circuitos de conmutación, aunque su forma de operación es diferente. Un “throughput” requerido puede ser logrado mediante una selección apropiada de los valores de los parámetros n , m , v y r tanto como la velocidad de enlace entre estados. El control de flujo y la rotación de un slot de entrada puede también influenciar en el “throughput” de la fábrica [40].

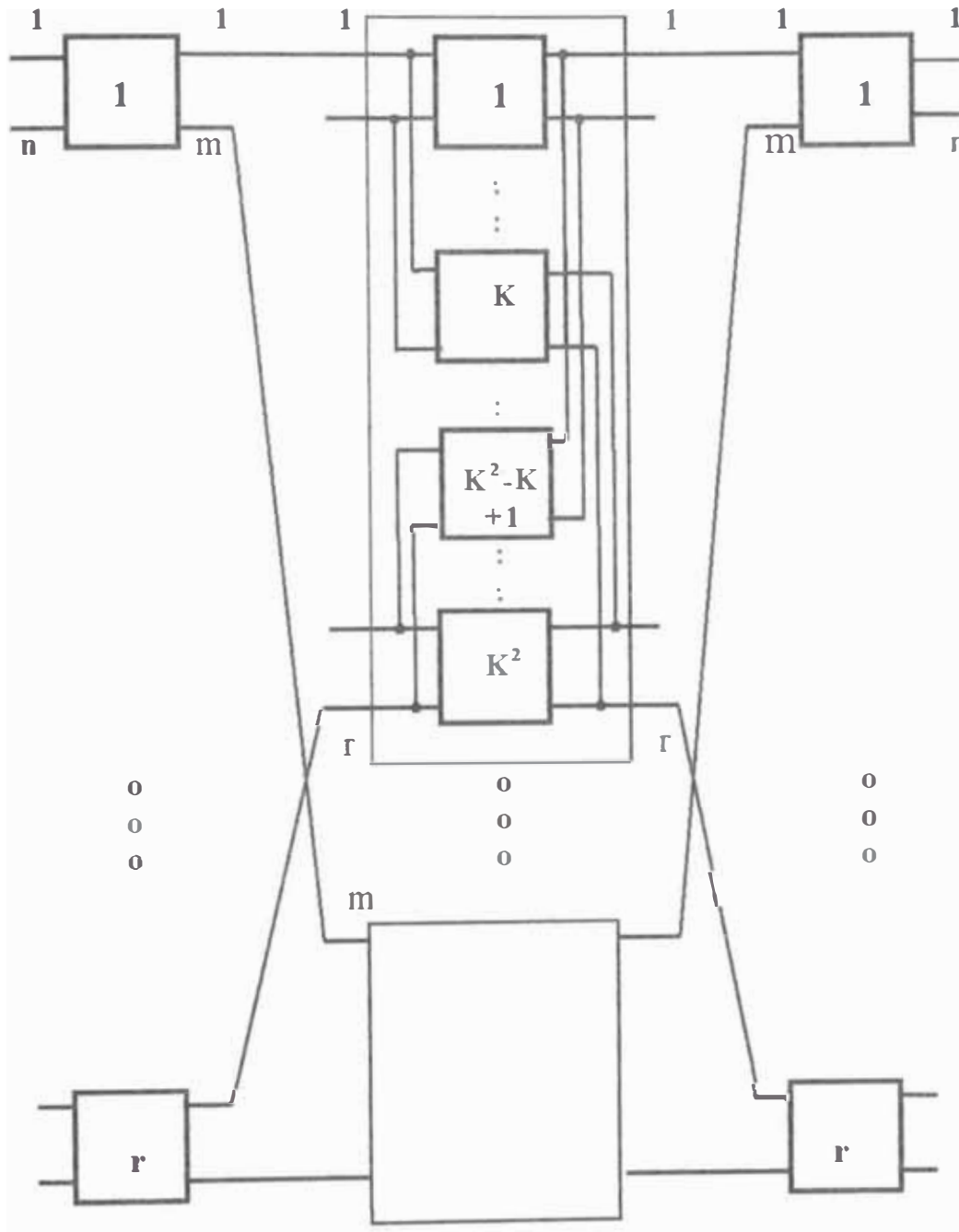


Figura 3.12: Otra fábrica modificada de 3 estados.

Dado que los conmutadores elementales son usualmente construidos usando la tecnología integrada, su número de puertos es fijo. Para obtener las

dimensiones requeridas de los conmutadores elementales, se puede usar una técnica basada en combinar un número de conmutadores elementales, tal como puede verse en las figuras 3.11 y 3.12 [38].

Por simplicidad de presentación, se asume que la multiplicidad de los enlaces interestados (v) sea igual a 1. Las técnicas arquitectónicas de las figuras 3.11 y 3.12 pueden ser aplicadas simultáneamente en la construcción de un conmutador, inclusive si éstas tienen más de 3 estados. Se puede notar que usando módulos multielementos, se puede reducir el número requerido de estados tanto como el número de conmutadores elementales, esto es, comparado con el diseño multiestado puro usado en un tipo de red como el de Clos [3][20] que estamos tratando en la presente tesis.

3.3.3. Capacidad de crecimiento de la estructura conmutacional

Se pueden considerar los siguientes casos:

a) Procedimiento de expansión.

La estructura básica que debe ser expandida está formada por estados óptimos que tienen N líneas en cada lado. Lo óptimo se refiere al desempeño, retardo, número de conmutadores elementales, entre otros. Esta fábrica puede tener la estructura mostrada en las figuras del 3.10 al 3.12 o la combinación de dichas estructuras o arquitecturas, considerando inclusive los que requieran de más de tres estados.

Para expandir el número de líneas 2 veces, añadimos una segunda fábrica idéntica a la estructura ya existente y los conectamos mediante módulos adicionales, tal como puede verse en la figura 3.13. Estos módulos tienen la misma estructura como los módulos de estado S_2 de la fábrica básica. Para incrementar el tamaño de la fábrica adicionando N líneas, se debe de añadir una fábrica de $N \times N$ y lo conectamos a la estructura ya existente usando los módulos S_2 . En general, para expandir el tamaño de la fábrica K veces, obtenemos una estructura mostrada en la figura 3.14, el cual tiene K fábricas básicas idénticas y el número de módulos que interconectan estas fábricas básicas lo gobierna la siguiente relación matemática:

$$2C_2^k = k(k-1) \quad (28)$$

Para incrementar el tamaño de la estructura de conmutación desde $(k-1)N$ a kxN líneas ATM es necesario añadir una fábrica básica de NxN y $2(k-1)$ módulos intermedios. En la figura 3.14 cada línea representa a un grupo de líneas entrantes o salientes desde los módulos de conmutación. En la figura 3.15 se ilustra el crecimiento o expansión de una estructura de 3 estados desde 64 hasta 128 líneas, esto como una aplicación práctica de la arquitectura desarrollada en este rubro.

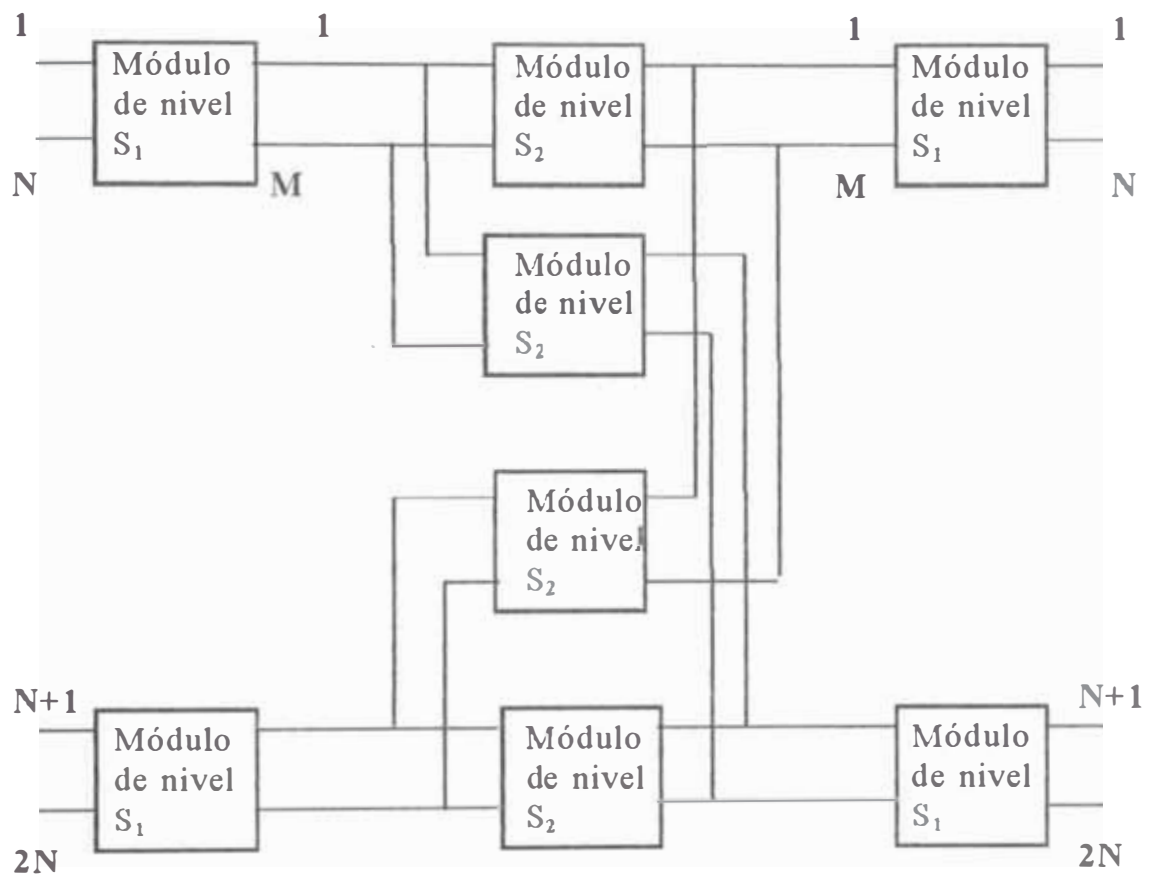


Figura 3.13: Duplicando el tamaño de la fábrica básica.

Se nota que una celda cruza el mismo número de conmutadores elementales independientemente del tamaño de la estructura de conmutación. El incremento de conmutadores elementales mejora el desempeño, pero incrementa su costo.

b) Número de conmutadores elementales.

El costo total del hardware del sistema de conmutación propuesto puede expresarse principalmente en función del costo de los conmutadores elementales y

el costo de los puntos de ramificación. En este acápite me ocuparé del primer punto. El número de conmutadores elementales para este sistema propuesto, según se señala en [3] está gobernado por:

$$C_{KN} = KC_N + K(K-1)C_{N,m} \quad (29)$$

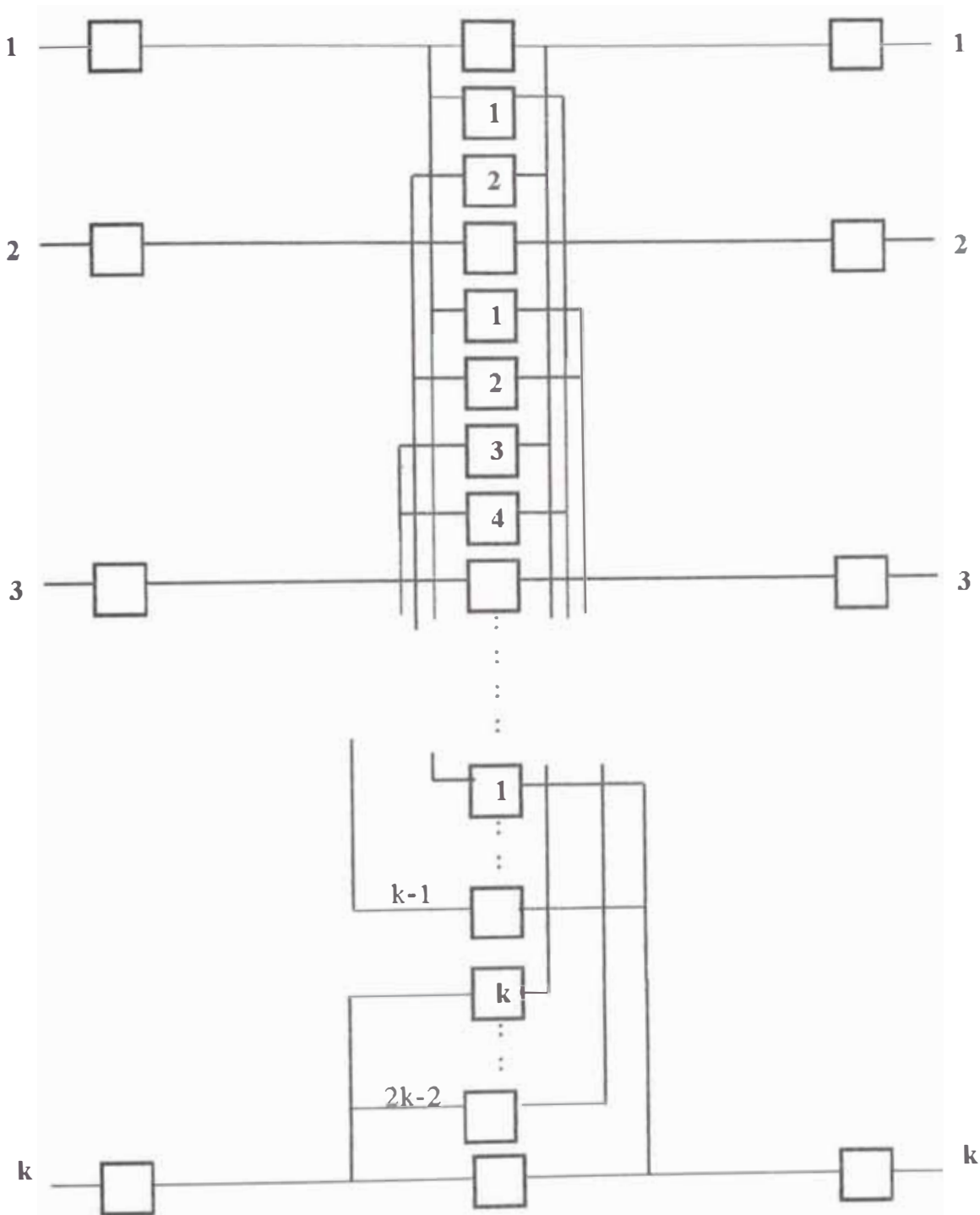


Figura 3.14: Principio de expansión de la fábrica.

Desde que el número de módulos intermedios crecen tan rápido como $k(k-1)$, tiene una influencia muy significativa en cuanto al costo total del sistema básico de construcción. Por tanto, es deseable usar módulos intermedios simples, que contengan mínima cantidad de conmutadores elementales en lo posible.

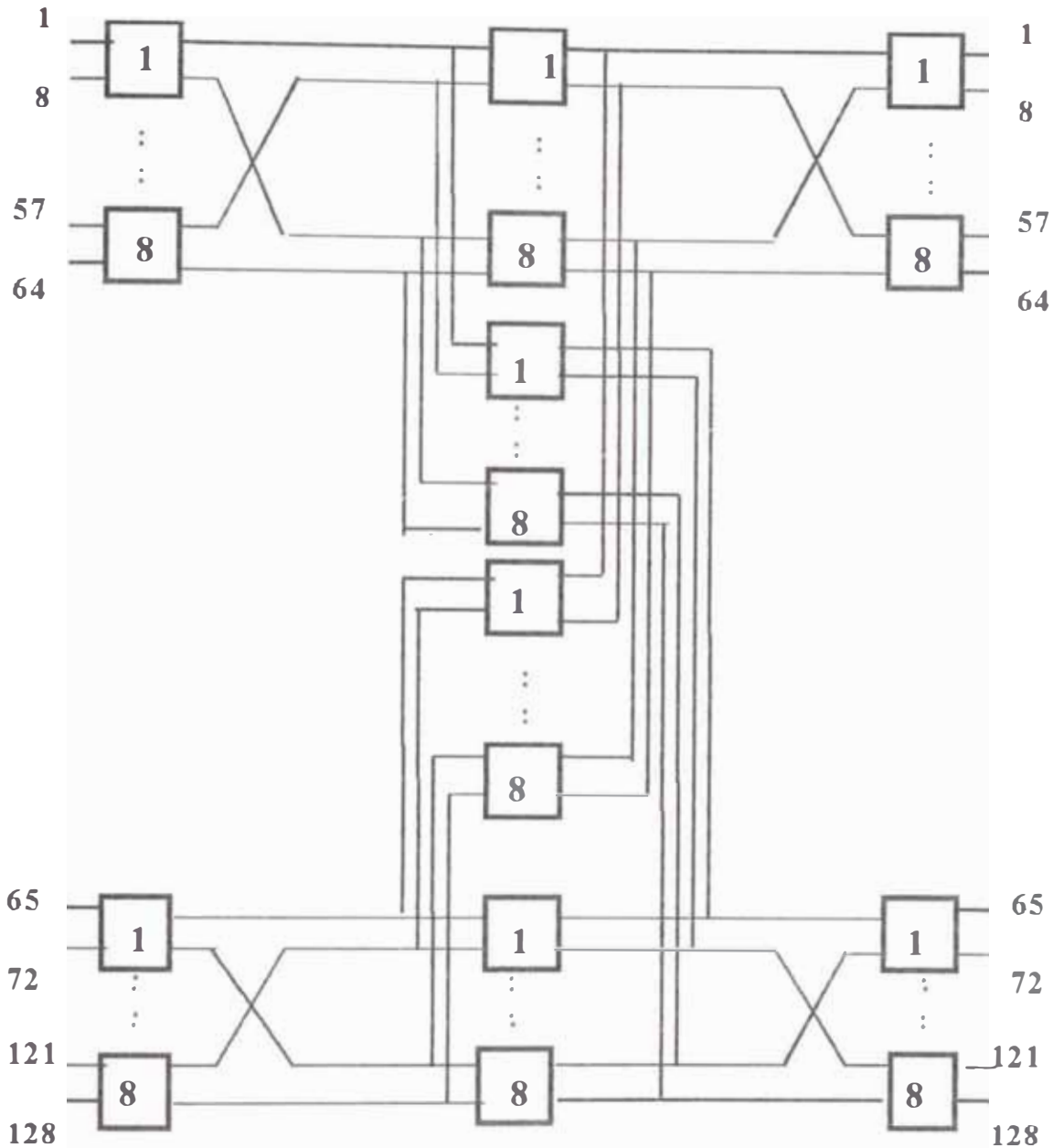


Figura 3.15: Una aplicación de la expansión de la fábrica de 64 a 128 líneas.

Donde:

C_{KN} : es el número de conmutadores elementales en un sistema básico de construcción que tienen KN líneas en cada lado.

C_N : Es el número de conmutadores elementales en un sistema básico de construcción.

$C_{N,m}$: Es el número de conmutadores elementales en la parte media del sistema básico de construcción.

En la figura 3.16 se ilustra la dependencia entre el costo del conmutador elemental y el tamaño del sistema conmutador básico a construir, en este caso para conmutadores elementales de 8x8 requerido en dos sistemas básicos modulares de varios tamaños. El primero tiene una estructura básica referida al mostrado en la figura 3.10 con $m=n=r=8$ y $v=1$ y la estructura básica del segundo está basada en las figuras 3.11 y 3.12 con $m=15, n=8, p=2$ y $r=8$. En un circuito de conmutación, estas dos fábricas pueden ser reconfigurables y libres de bloqueo en el sentido más estricto posible [20] [37].

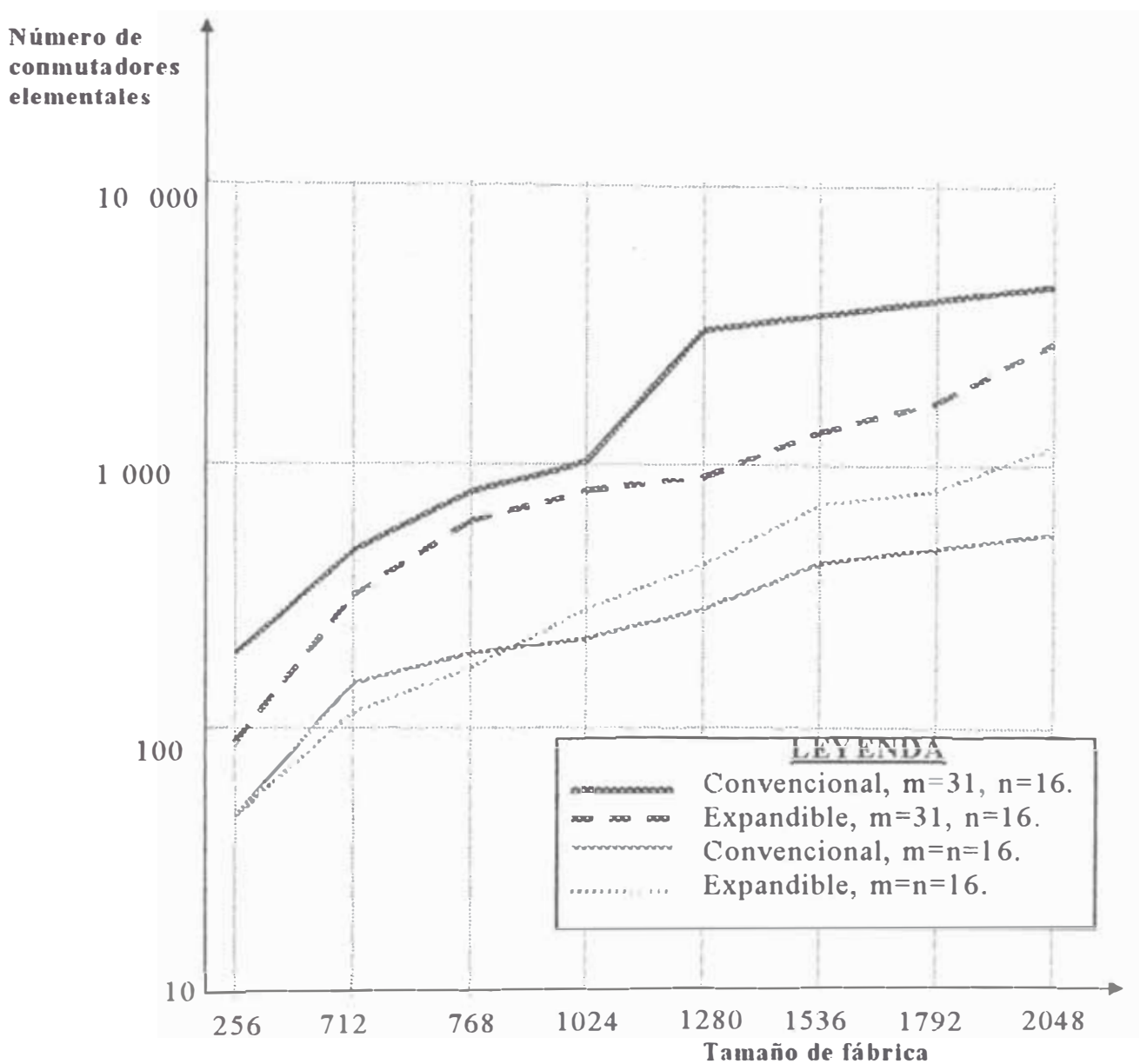


Figura 3.16: Número de conmutadores elementales de 8x8 en fábricas de varios tamaños.

Se pueden notar, sin embargo, que, en el caso de un conmutador ATM o un conmutador rápido de paquetes, siempre ocurrirá el problema del bloqueo en ambos sistemas de construcción indicados[41]. Por comparación, el número de conmutadores elementales que se muestran son los considerados en CLOS [3] [31] [39] como “reconfigurables” y “sin bloqueo” (obtenido de acuerdo a [37]). Todas las fábricas que pertenecen a una de las cuatro clases presentadas son óptimas en el sentido del mínimo número de conmutadores elementales requeridos y el costo de cada uno de los elementos se pueden calcular mediante el método de programación dinámica [20].

En el presente trabajo no se presentan mayores detalles al respecto, debido a que, el objetivo es presentar la existencia de diferentes planteamientos de arquitecturas de fabricación de conmutadores ATM de grandes dimensiones, siendo la presente una propuesta de fábrica y arquitectura, que es interesante considerar sobre todo, teniendo en cuenta el principio en el que se basa, que es el mismo del principio Knockout, pero también, este planteamiento está basado en otros criterios algorítmicos y análisis matemáticos previamente analizados [11] [20] [31] [37] [41], que salen del objetivo de la presente tesis.

c) Puntos de ramificación.

Una preocupación importante asociada a la escalabilidad de la fábrica planteada es respecto a su implementación, sobre todo en lo que se refiere a los puntos de continuación y crecimiento (indicados con puntos suspensivos en las figuras 3.11, 3.12 y 3.15). Tal objetivo, se puede lograr superando las diferentes dificultades tecnológicas como las reflexiones eléctricas que están presentes a lo largo de complicados cableados y debido a las señales de muy alta velocidad que manejan los conmutadores ATM. En algunos sistemas, una ramificación pasiva de puntos de división dan resultados satisfactorios[42]. En el caso de las ramificaciones pasivas, los conmutadores elementales deben tener la capacidad de aceptar sólo llamadas con valores apropiados de las cabeceras de auto enrutamiento. Otra opción es aplicar demultiplexores y multiplexores en puntos de división y combinación de los mismos, respectivamente. Los demultiplexores pueden ser controlados mediante bits de control en la cabecera de la celda.

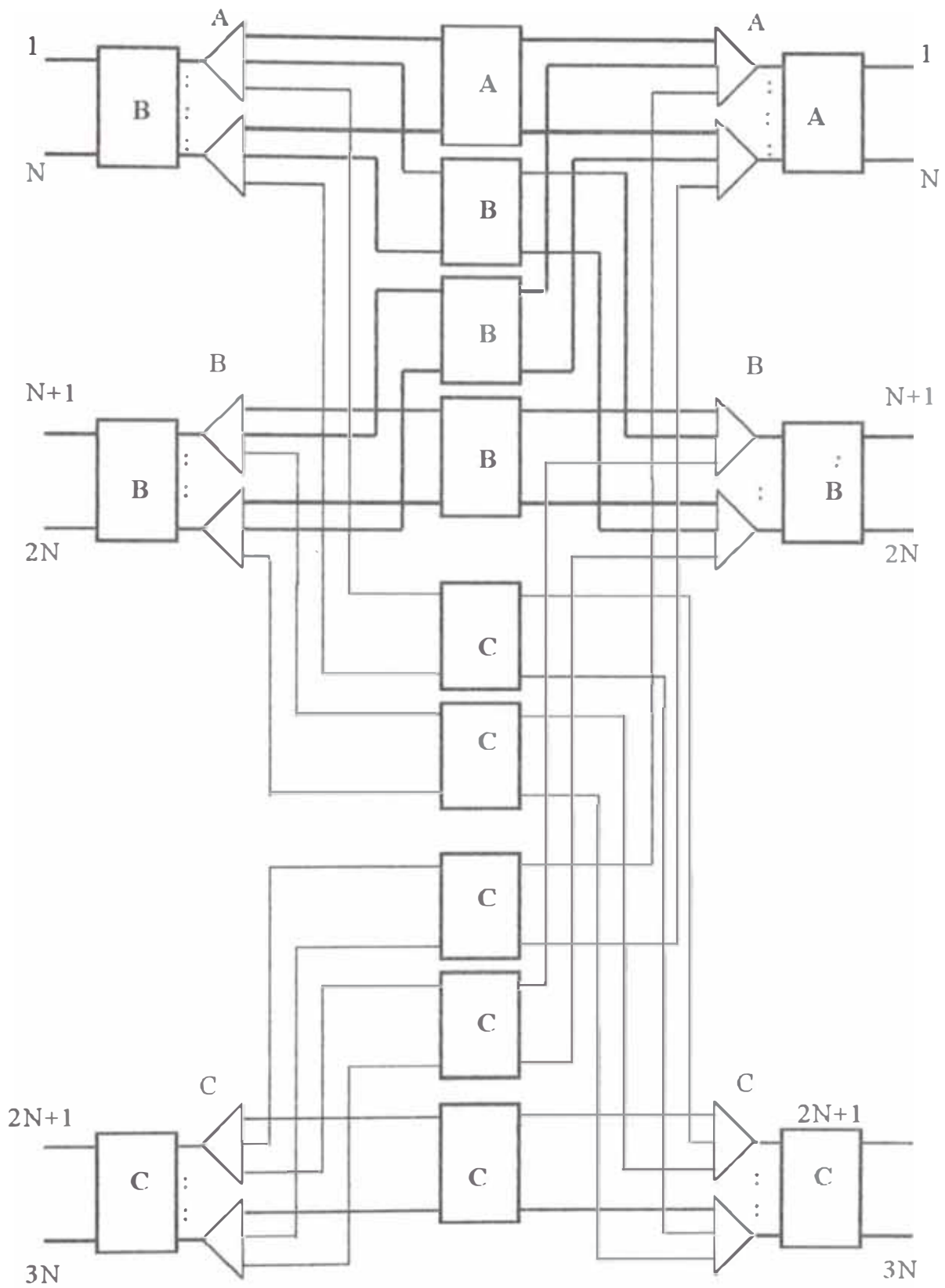


Figura 3.17: Estructura de la fábrica 3N.

Otra opción más optimizada es el uso de fibras ópticas como medio físico de transmisión y que es tecnología actual en manejo de datos a nivel mundial.

Funciones más complejas tienen que ser desarrollados por los multiplexores. Estas funciones pueden incluir la prevención de congestiones internas cuando dos o más celdas compiten el acceso hacia la misma salida del multiplexor. Esto se puede superar mediante la ubicación de un buffer apropiado. Los multiplexores pueden mantenerse sólo si se le provee un mecanismo adecuado entre los conmutadores elementales en estados consecutivos tal como lo propone Kosaki [43] y la arquitectura del conmutador ATM con buffer compartido es descrito con precisión por Onvural [39] y es la que se describirá más adelante de una manera precisa y resumida.

Otra solución es la planteada por Itoh [44], en el cual la función de puntos de ramificación de la entrada y la salida (mostrada en la figura 3.11) es resuelta mediante los traductores en la cabecera de la celda.

Se puede notar que el costo de los puntos de ramificación puede incrementar considerablemente todo el costo de la implementación del sistema conmutador. El número de los módulos de conmutación, que en este caso sirven como multiplexores está gobernada por la siguiente fórmula, que se muestra como información adicional.

$$C_{M,K} = K \left[\frac{M}{\left[\frac{M}{K} \right]} \right], \quad K=2,3,\dots \quad (30)$$

Donde:

$C_{M,K}$: es el número de módulos de conmutación de estados S_2 .

M : Es el número de líneas ATM entre los módulos S_2 y S_1 .

$\lfloor X \rfloor$: Es el máximo entero menor o igual que X .

$\lceil X \rceil$: Es el mínimo entero mayor o igual que X .

Como ejemplo, se puede ver que si M es divisible por K el número de módulos de estados S_2 es duplicada (K^2 módulos en el centro de la fábrica y K^2 módulos sirviendo como multiplexores). Por tanto, en la figura 3.15, se tiene que $C_{M,K} = 4$ módulos de estado S_2 , cada uno conteniendo 8 conmutadores ATM de 8×8 . Debido a lo considerable del costo de los puntos de ramificación, todo el arreglo de una fábrica de estos conmutadores se hace más modular y flexible que los sistemas convencionales propuestos.

En cuanto al diseño físico de cualquier sistema, la base está en el sistema de partición [45]. Al inicio se debe tomar una decisión respecto al tamaño y la estructura del sistema a implementar. Hay una diferencia entre el costo inicial, la máxima capacidad de la tarjeta y su capacidad de crecimiento. También, la unificación de tipos de tarjetas es de gran importancia en el diseño. Los tipos de tarjetas dependen grandemente de la disponibilidad tecnológica así como de la estructura básica del sistema en diseño para su implementación. Un ejemplo que visualiza un posible sistema modular con muchas tarjetas separadas es la que se muestra en la figura 3.17. En el ejemplo, se emplean tarjetas separadas para los módulos de estados S_2 y S_1 (ver figura 3.13) y dos tipos de puntos de ramificación. Las letras A, B y C indican los módulos del sistema usados durante la ampliación o engrandecimiento desde N líneas hasta $3N$ líneas respectivamente. Se puede notar que los módulos de conmutación contienen múltiples conmutadores elementales (que puede verse en la figura 3.15).

Como se puede ver, el número de los multiplexores y demultiplexores crece linealmente tanto como crece el tamaño del sistema conmutador, por supuesto que también crecerá el tamaño de los mismos. En la práctica, como sugieren los investigadores y especialistas en la materia, para permitir un crecimiento moderado, los puntos de ramificación deben estar en una tarjeta y de acuerdo al tamaño, por lo que pueden permitir controlar la complejidad en cuanto al hardware. En tales casos, cada expansión adicionará solo nuevos cables (líneas gruesas en el esquema) sin generar más disturbios a los ya existentes. En la fórmula mostrada, puede fácilmente modificarse dando valores a K en el numerador (valores corrientes) y dando valores a K en el denominador (K representa a las tarjetas).

3.4. Arquitectura de un conmutador ATM con Memoria Compartida [39][43].

La característica que distingue a un conmutador con memoria compartida de $N \times N$ es su alta velocidad en el bus interno, con una tasa de bit N veces más grande que la tasa en cada línea de entrada y salida. Para un tiempo de celda de longitud F , el bus interno es capaz de transferir una celda en un minislot de longitud F/N . Todas las celdas recibidas durante un tiempo de celda son transferidas a la memoria compartida y con muy poco retardo. Se requiere convertir de la forma serial a la forma paralela para mantener una tasa de reloj aceptable; la tasa de reloj de bus solo necesita ser N/W veces más alta que la tasa de bit que ingresa, siendo W el ancho del bus y de la memoria.

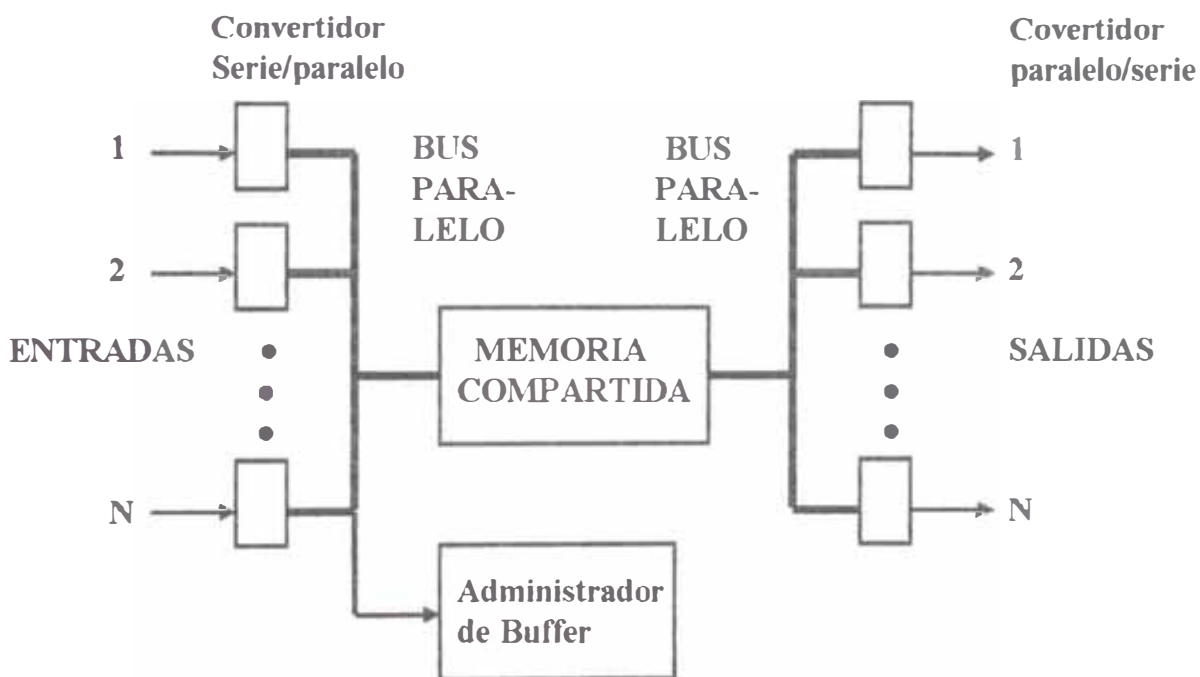


Figura 3.18: Arquitectura de un conmutador con memoria compartida.

En la memoria compartida, las celdas que se dirigen hacia una determinada salida son almacenadas en particiones diferentes. Durante un ciclo de salida, las celdas son descargadas hacia sus salidas. En el bus de salida se requiere solo un minislot de longitud F/N para descargar una celda. En consecuencia en un tiempo de celda ésta es leída desde la memoria compartida para cada salida. Si la

partición de una memoria para una salida está vacía, su correspondiente minislots queda también vacío.

Para su implementación, es necesario saber si las particiones de la memoria compartida son de tamaño fijo o variable. Si el tamaño de las particiones son variables, obviamente requiere un hardware más sofisticado, pero el comportamiento de la tasa de pérdida de celda es alcanzada, porque la partición de la memoria no sufre de "overflow" hasta que la memoria no quede totalmente vacía. La lentitud de las salidas a un tiempo dado pueden asignar algo de memoria a otras salidas y si eso sucede es para ser usado en el momento.

Su simplicidad de diseño de la arquitectura de conmutador ATM con memoria compartida lo hace popular para conmutadores de tamaño pequeño pero rápidos o veloces, sobre todo para interconectar un pequeño número de redes LAN. Sin embargo, para conmutadores de tamaño moderadamente grandes, la frecuencia del reloj necesario para el bus interno se hace intolerable. Por ejemplo, para una tasa de celdas que llegan a 155 Mbps con $N=32$ y $W=16$, el bus interno tiene que operar con una frecuencia de reloj de 310 Mhz.

3.5. Arquitectura PINIUM [2]

Esta arquitectura es ampliable y auto-enrutada y está basada en varios principios conceptuales, tales como:

- Principio Knockout que explota la característica estadística de las celdas que llegan y por lo que reduce la complejidad de interconexión,
- El uso de los buffers en la salida del conmutador, permiten reducir los retardos y lograr un mejor desempeño,
- Control distribuido en enrutamiento (comercial) de celdas a través de la fábrica interconectada sin conflictos internos de las rutas y,
- Bloques simples de construcción o implementación que facilitan su escalabilidad.

Otras de las características de la arquitectura propuesta son:

- Su capacidad intrínseca para un servicio multicast y broadcast,

- Construido o implementado con prioridad a su clasificación funcional o de acuerdo a su funcionalidad.
- Su capacidad de garantizar la secuencia de celdas que el primero que entra es el primero que sale (FIFO).

La arquitectura PINIUM consiste de la sección de distribución y otra sección de concentración. La sección de distribución está hecho de un bloque de un árbol multicast de base r . Este árbol de redes provee al conmutador las funciones de enrutamiento y servicio multicast. La sección de concentración está conformada de una fila de N a L concentradores selectores con prioridad. Estos concentradores selectores con prioridad proveen la selección del hardware prioritario y las funciones de concentración. Se emplea el principio Knockout en la sección de concentración para aprovechar su característica estadística de control de celdas que llegan y por tanto reducen la complejidad de interconexión. Con este diseño en el conmutador, se asume que las celdas que llegan a la entrada son como máximo L celdas simultáneas para un mismo puerto de salida. En la primera parte del presente capítulo se ha analizado el principio Knockout generalizado que también es aplicable aquí para un grupo de puertos de salida. Existe una reducción considerable en el hardware con el principio Knockout generalizado, y el conmutador puede operar bastante bien mientras no se sature su comportamiento de acuerdo al control de tráfico mediante el principio de Bernoulli. Sin embargo, en el caso de un tráfico por ráfagas, el tamaño del buffer se incrementa drásticamente en los bloques de memoria de salida compartida. Es posible determinar muchos patrones de tráfico tal que los conmutadores operen mediante el principio Knockout, es decir, no distribuyan más de una celda hacia cada puerto de salida en un time slot siempre que exista un pequeño número de celdas destinadas a un mismo puerto de salida. Pero, mediante el principio PINIUM no se tendrá este problema.

En la figura 3.19 se puede ver que cada puerto de entrada tiene un sistema de derivaciones multicast y cada puerto de salida tiene N a L selectores de concentrador prioritario. Por tanto, los módulos de conmutación de entrada y salida están completamente distribuidos. Esta fábrica de conmutador distribuido

proporciona una arquitectura distribuida flexible que es la clave para simplificar la operación y mantenimiento de todo el sistema de conmutación. La forma modular implica menos necesidad de sincronización y hace posible una implementación de alta velocidad. Este conmutador brinda servicios con prioridad y es un conmutador multicast que internamente no tiene bloqueo y tampoco tiene buffer.

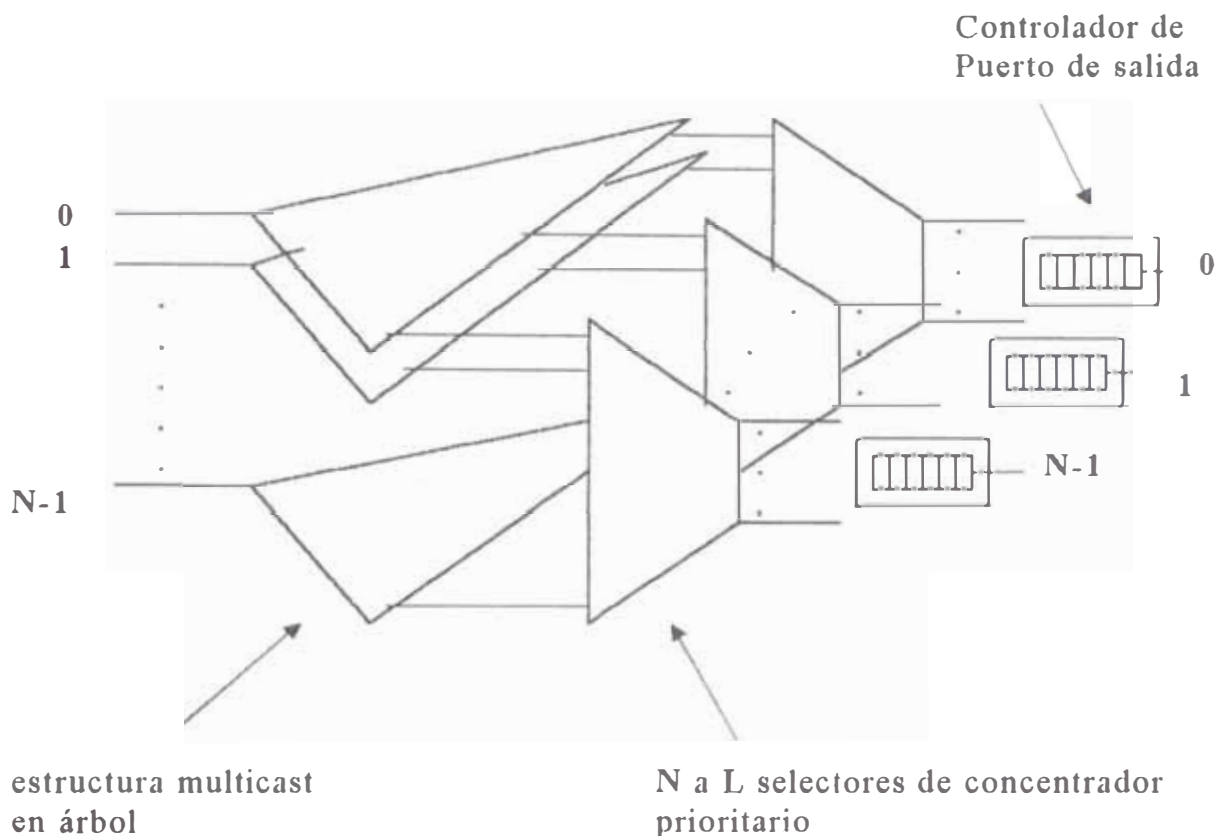


Figura 3.19: Arquitectura básica de un conmutador PINIUM.

a.- La Distribución.

La base r de la estructura en árbol es responsable de transportar hacia afuera las funciones de enrutamiento y multicasting. Las copias del multicast son auto generadas y auto encaminadas hacia sus destinos correspondientes de acuerdo a: $\log_r N$. El valor de r está determinado por la capacidad de la corriente que maneja el Fan In y Fan Out respectivamente. A diferencia de otros diseños de redes con copia sin bloqueo, la estructura del árbol en base r no sufrirá problemas de overflow, por tanto proveerá un servicio rápido y al instante a todas las celdas multicast que llegan.

La funcionalidad multicasting es una acción natural en una estructura en árbol de las redes. Cada puerto de entrada tiene un árbol de plano separado; por lo tanto, no hay conflicto entre la generación de copias múltiples con la llegada de otras celdas multicast. Se requiere una estructura en árbol de los planos para desarrollar muchas funciones básicas, los cuales rotan de punto a punto, de punto a un multipunto multicast, grupo de distribución y broadcasting. Existen muchas técnicas que permiten lograr servicios de un punto a un multipunto en base a una estructura en árbol y las redes Banyan en un paso simple. En el presente trabajo, se hace una breve referencia a dos técnicas que se adaptan al uso en redes con arreglo en una estructura en árbol en base r . Dichas técnicas son el filtrado de celdas y el esquema de direccionamiento de vértice aislado (VIA)[27].

La técnica de filtro de celda, es un método directo de envío de celdas broadcast hacia todos los puertos de salida. En cada una de las salidas de la estructura en árbol multicast, un filtro de celda es necesario para determinar si una celda está destinada hacia un puerto de salida en particular. La configuración correspondiente se muestra en la figura 3.20. Esta técnica requiere de una modificación en la sección de distribución del diseño arquitectural del conmutador PINIUM, empleando una estructura en árbol para el sistema broadcast en vez de la estructura en árbol para multicast. Desde que sólo es necesario un bit para representar una dirección del puerto de salida, quedan N extra bits en la cabecera de la celda. Este método tiene un mínimo exceso en el enrutamiento de celdas multicast en un paso simple. Sin embargo, son necesarios N^2 extra filtros de celdas, en la versión de filtro de celda para el conmutador PINIUM.

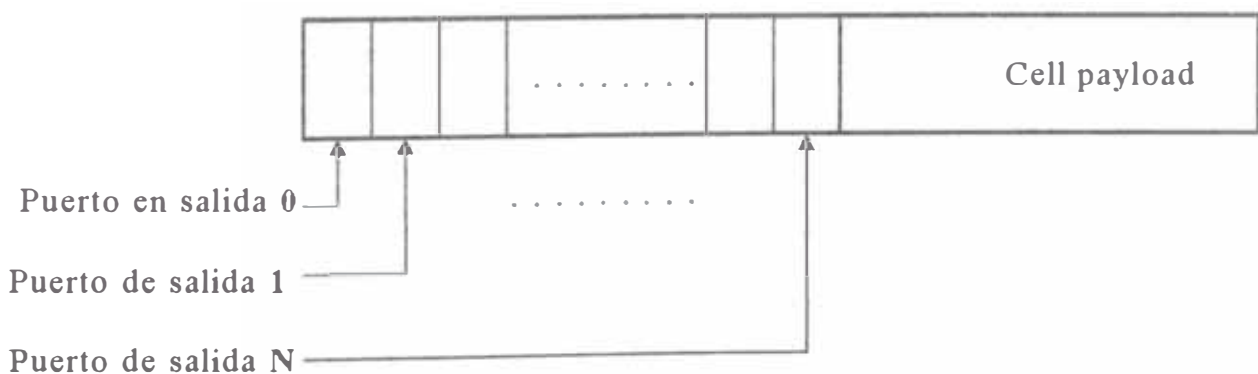


Figura 3.20: Distribución de punto a multipunto. Método de aproximación de filtros de celda.

El otro método es el esquema VIA. El formato de celda se muestra en la figura 3.21; cada bit representa el proceso de enlace de un nodo en una estructura en árbol multicast. En consecuencia, el número de bits en la cabecera de la celda VIA es igual al número de enlaces dentro de una estructura en árbol. La celda Via retiene la propiedad de autoenrutamiento, que provee los servicios multicast y no requiere un número extra de filtros de celdas. El exceso es ligeramente más alto que el del método de filtro de celda. En una estructura en árbol de base r el elemento multicast es uno de los 1 a r unidades multicast. El número de los extra bits requeridos está dado por: $r(N-1)/(r-1)$. Este exceso es asintóticamente cercano al método de filtro de celda donde r tiende a N . Por ejemplo, cuando $N=256$, $r=4$; en este caso se requiere un factor de 1,8 que incremente la velocidad en una estructura en árbol VIA multicast comparada con el factor de 1,6 de incremento de velocidad requerida en la técnica de filtrado de celda.

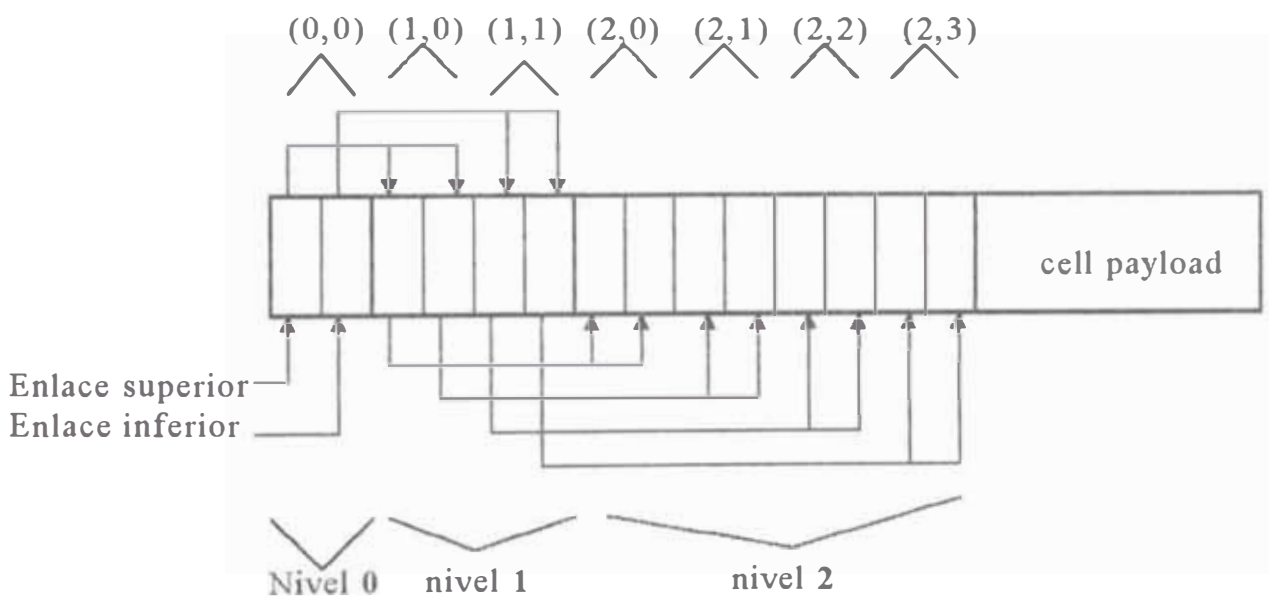


Figura 3.21: Distribución de punto a multipunto. Método de esquema VIA.

Los objetivos principales de la implementación de la estructura del árbol son mejorar el problema de la sincronización y proveer una técnica adecuada en la realización de un VLSI de alta velocidad. Inclusive, el hardware puede ser sustancialmente simplificado para poder implementar mediante fibras ópticas la técnica de filtrado de celda.

b.- La Concentración.

En la sección de concentración del conmutador PINIUM, cada puerto de salida tiene un plano de conmutación separada. Cada uno de estos planos de conmutación desarrollan dos funciones: Concentración y selección priorizada. No tiene importancia usar N clasificadores para cada puerto de salida. Una secuencia de celdas clasificadas es obtenida de acuerdo a la prioridad de las celdas. Por tanto, se pueden escoger L celdas más altas priorizadas para enrutar hacia un controlador de puerto de salida. Si se emplea un clasificador bitónico (conmutador elemental de dos entradas y dos salidas, tal como lo definió Banyan) [19][13][39], el número de estados y el número de elementos de conmutación en un clasificador están en el orden de $O((\log_2 N)^2)$ y $O(N(\log_2 N)^2)$, respectivamente. En consecuencia, se nota que es crucial el diseño de un clasificador concentrado con prioridad en la determinación de la complejidad del hardware. Además, es sabido que el clasificador denominado bitónico [13] [19] [39] limita cuánto un conmutador puede ser expandido. En las siguientes líneas describo la técnica para construir grandes clasificadores concentradores mediante el uso del Knockout generalizado.

El número N de entradas al conmutador es mucho mayor que los parámetros L del Knockout ($N \gg L$). Por tanto, no es razonable construir muchos selectores de N para cada puerto de salida. El principio del Knockout generalizado también implica que los conmutadores elementales implementados con todos los selectores de salida ya no se emplean en la actualidad. En este caso, podemos simplemente descomponer los selectores priorizados basados en el parámetro L del Knockout. Se puede distribuir el selector de las $2L$ entradas en dos componentes básicos: dos selectores de L y un mezclador de $2L$. Entonces podemos denominar a la estructura resultante como un mezclador concentrado con prioridad de L a $2L$.

En un selector concentrador con prioridad de N a L , la primera columna de la operación es hecha por los clasificadores de L para obtener N/L secuencias clasificadas. Un mezclador concentrador priorizado de $2L$ a L se usa entonces para mezclar cada dos secuencias seleccionadas dentro de una secuencia

seleccionada basada en la prioridad de las celdas. Una celda que llega hacia un controlador de un puerto de salida pasará a través de $\log_2(N/L)$ estados de mezcladores concentradores priorizados de $2L$ a L . En consecuencia, solo requerimos dos componentes básicos para construir selectores concentradores con prioridad de N a L y mezcladores concentradores con prioridad de $2L$ a L .

Selector de 4 Entradas mezclador concentrador priorizado de 8 a 4.

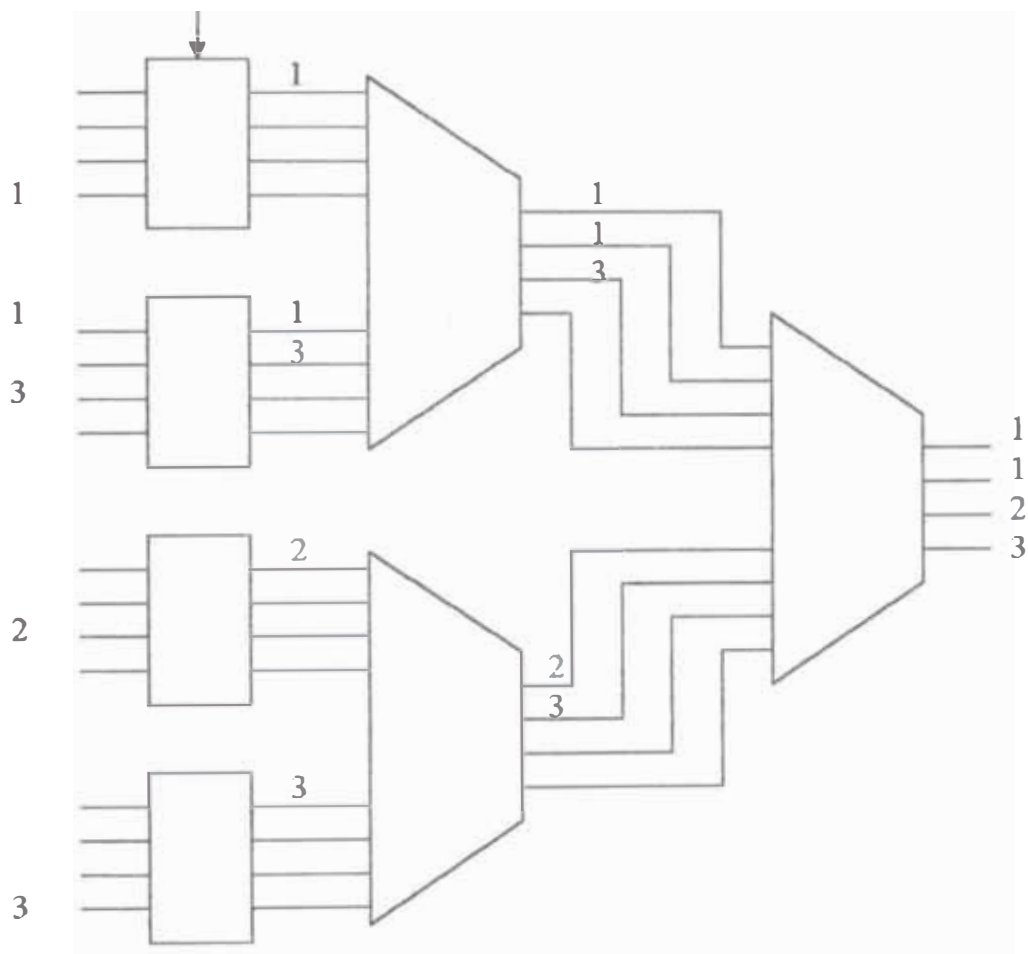


Figura 3.22: Selector concentrador priorizado de 16 a 4.

En este método de descomposición, un selector concentrador priorizado de N a L preserva todo lo mínimo de una información deseada y las L celdas de mayor prioridad a través de todo el plano de conmutación. Debido a que los dos componentes básicos de construcción son pequeños y se pueden construir mediante tecnologías conocidas, se puede hacer uso del concepto de modularidad

y escalabilidad. En la figura 3.22 se muestra un ejemplo de construcción de un selector concentrador priorizado con $N=16$, $L=4$, y celdas con prioridades 1, 2 y 3. Es necesario hacer notar que no interesa cómo se arreglan las 5 celdas en las entradas de los concentradores, sólo la celda con la menor prioridad se perderá.

c.- Expansión y empaquetado de un conmutador PINIUM.

Los planos de conmutación de un conmutador PINIUM puede incrementarse a dimensiones deseadas. El máximo valor de r de un árbol de base r es determinado por el Fan In y el Fan Out de corriente. Se asume que el

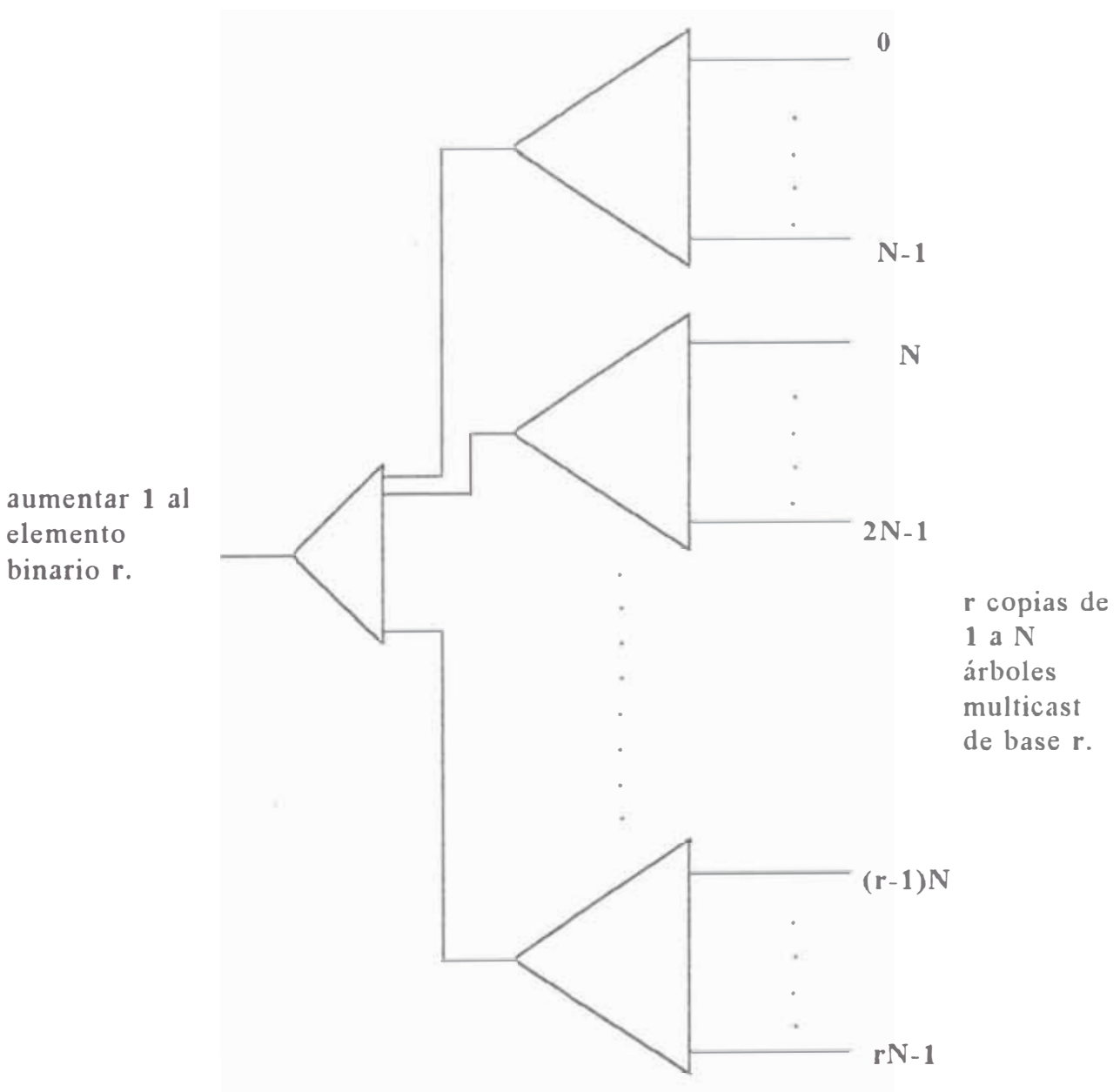


Figura 3.23: Expansión de una distribución.

mayor posible es el usado. Tal como se muestra en la figura 3.23 solo es posible aumentar 1 (uno) a la unidad multicast r al frente de r las N salidas del árbol para formar rxN salidas del árbol de planos.

La expansión incremental del clasificador concentrador con prioridad es también simple. Solo es necesario aumentar un bloque mezclador concentrador de $2L$ a L a dos selectores de N a L para obtener un selector concentrador de $2N$ a L .

En la arquitectura de un conmutador PINIUM es preferible usar una técnica de empaquetamiento tridimensional en el cual los planos de distribución son ortogonales a los planos de concentración. Cada puerto de entrada tiene un árbol de planos multicast y cada puerto de salida también tiene un selector concentrador priorizado. Lo más importante, es que los selectores de L y los mezcladores concentradores de $2L$ a L son pequeños y fácilmente escalables. Esta estructura provee una arquitectura muy promisoriosa y realizable en escalas grandes. En la configuración tridimensional, no existen cruces de cables entre módulos. Todas las conexiones de cables son arreglados en paralelo entre los módulos básicos de construcción. Por tanto, no existe discrepancia en el retardo en cuanto a la transmisión de una señal entre bloques de construcción, lo que permite suavizar los problemas de sincronización. Estas ventajas permitirán al usuario aumentar el tamaño del conmutador PINIUM fácilmente.

RESUMEN

En el presente capítulo se han descrito algunas arquitecturas para conmutadores ATM de grandes dimensiones, siendo los descritos: Knockout Generalizado, Plegables de tres estados y el PINIUM.

La arquitectura Knockout se basa en tres principios fundamentales:

- a) El principio del Knockout generalizado, en el cual la probabilidad de que más de m paquetes lleguen simultáneamente hacia un grupo pequeño de n salidas, excede en forma mínima a n , para cualquier conmutador de tamaño N , con $m \ll N$. Por tanto, solo valores mayores a m paquetes por tiempo de celda requieren ser transportados a través del sistema (fábrica) hacia un módulo de salida del conmutador de paquetes que tenga n salidas. Por ejemplo, para lograr una probabilidad de pérdida de paquetes de 10^{-6} se tiene que $m=33$ para $n=16$ y $N= \infty$ (con una carga del 90%);
- b) Disponer de buffers de cola de espera en la salida, tal como se ha detallado en el tercer capítulo de la presente tesis, genera mejor desempeño en el retardo y throughput, dado que sólo valores mayores a m paquetes son distribuidos de manera interrumpida a través del sistema interconectado hacia el módulo de paquetes de salida $m:n$.
- c) Se emplea inteligencia en el enrutamiento de paquetes a través del sistema interconectado y en forma rápida. El algoritmo de conexión distribuido con un desempeño casi óptimo es lo que se ha descrito en el tercer capítulo del presente trabajo.

La arquitectura con capacidad de crecimiento expuesta mediante las arquitecturas plegables, bilateral y tres estados presenta las siguientes ventajas:

- a) Escalable,
- b) Moderada complejidad en hardware,
- c) Estados de conmutación fijos y en pequeña cantidad y
- d) Alta dependencia.

La arquitectura PINIUM es una arquitectura de conmutador ATM escalable a grandes dimensiones que consiste de un sistema de árbol de raíz r y un sistema de red de Batcher modificada. Los bloques de construcción básicos son pequeños y son usados para construir conmutadores ATM de grandes dimensiones. Físicamente puede ser realizado como una matriz tridimensional de procesadores paralelos. Los módulos de conmutación de entrada y salida están completamente compartidos; y un conmutador elemental compartido provee una arquitectura distribuida flexible, lo cual es la clave que simplifica la operación y mantenimiento de todo el sistema de conmutación.

CAPÍTULO IV

PROBLEMAS DE ENRUTAMIENTO EN CONMUTADORES ATM DE GRANDES DIMENSIONES

4.1. Problema de enrutamiento.

Para conmutadores $N \times N$, tanto las entradas como las salidas están divididas en K módulos con n líneas cada uno. Las dimensiones de los módulos de entrada y salida son de n a m y de m a n respectivamente, y hay m módulos en el estado intermedio, cada uno de una dimensión $K \times K$. Los paquetes llegan a la entrada sincrónicamente, celda por celda, con direcciones de destino distribuidas uniformemente. Desde que cada módulo de paquetes de salida es un conmutador de paquetes, es suficiente rotar un paquete hacia una de las m entradas del módulo de salida, ya que dicho módulo contiene sus rutas o caminos de salida. Por lo tanto, el problema de enrutamiento consiste en dirigir los paquetes de entrada a través de la estructura interconectada hacia los módulos de salida sin conflictos de encaminamiento. Para cada tiempo de celda, el tráfico de paquetes puede ser escrito como:

$$T = \begin{bmatrix} t_{11} & t_{12} & \dots & \dots & \dots & t_{1k-1} & t_{1k} \\ t_{21} & t_{22} & \dots & \dots & \dots & t_{2k-1} & t_{2k} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & t_{ij} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ t_{k1} & t_{k2} & \dots & \dots & \dots & t_{kk-1} & t_{kk} \end{bmatrix} \quad (31)$$

La fórmula anterior describe convenientemente los requerimientos de tráfico, adaptándolo a nuestro requerimiento [23].

En la matriz se tiene que t_{ij} representa el número de paquetes que llegan al i -ésimo módulo de entrada destinado para el módulo de paquete de salida j -ésimo. La suma de filas (R_i) es el número total de paquetes que llegan a cada módulo de entrada, mientras que la suma de columnas (S_j) es el número de paquetes destinados para cada módulo de salida, los que están dados por:

$$R_i = \sum_{j=1}^k t_{i,j} \leq n < m ; \text{ para : } i = 1, 2, \dots, k. \quad (32)$$

$$S_j = \sum_{i=1}^k t_{i,j} \leq N ; \text{ para: } j = 1, 2, \dots, k. \quad (33)$$

Desde que en el principio Knockout generalizado, sólo se consideran paquetes mayores a m para cada grupo de salidas, se asume que una selección aleatoria sobre los m ganadores es hecha sobre los paquetes que llegan, y los perdedores son inmediatamente descartados de las entradas. La suma de las columnas entonces se puede re-escribir así:

$$S_j \leq m; \quad j=1,2,\dots,k. \quad (34)$$

Se asigna a cada paquete una ruta específica a través de los primeros dos estados, tal que, desde el módulo de entrada no salgan dos paquetes a la vez para una misma salida del módulo de entrada y que no lleguen a una misma entrada del módulo de salida dos paquetes destinados para una misma salida. Tal consideración es garantizada por la propiedad de no bloqueo de una red de tres estados de la red de Clos para $m > n$. Sin embargo, para explorar la asignación de algoritmos, es necesario revisar el teorema de existencia para un análisis posterior de la transformación espacio-tiempo.

Etiquetemos el conjunto de salidas desde cada módulo de entrada mediante A_i ($i=1$ hasta k) y el conjunto de entradas dentro de cada módulo de paquetes de salida mediante B_j ($j=1$ hasta k). Cada A_i y B_j contienen exactamente m elementos que denotan m líneas físicas. Si estos elementos son vistos como

tiempos de celda en vez de líneas físicas, entonces los dos primeros estados (bloques), denominado en sí el sistema interconectado, puede ser transformado en un conmutador tiempo-espacio-tiempo de $K \times K$, donde cada entrada a la trama A_i y salida de la trama B_j tiene m celdas, tal como se muestra en la figura No. 4.1.

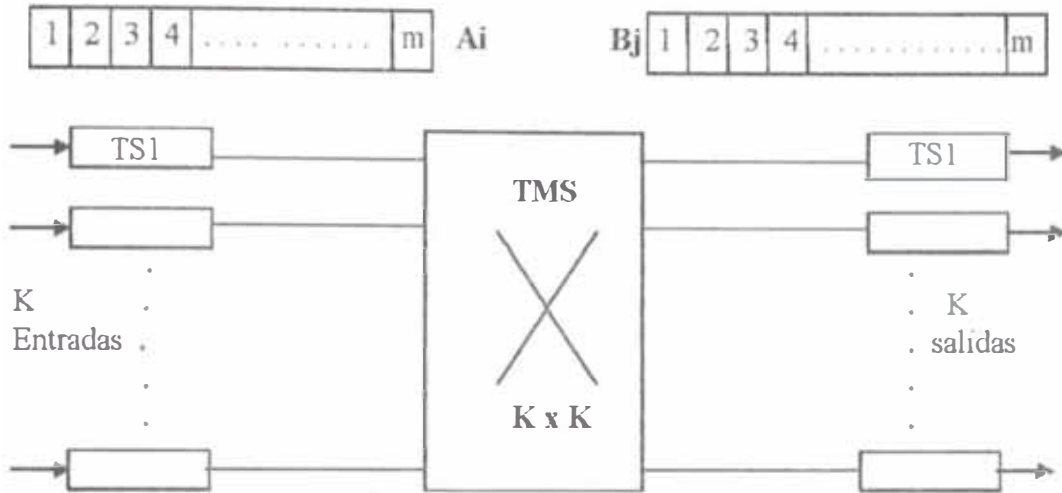


Figura No. 4.1: Un conmutador clásico TST.

El mecanismo de asignación, ahora se puede ver como un conmutador TST (tiempo-espacio-tiempo) clásico [23], condición necesaria y suficiente para garantizar un perfecto y óptimo resultado satisfecho por Yu Shuan Yeh, Michael O. Hluchyj and Anthony S. Acampora [9] y K. Y. Eng and A. S. Acampora [23]. Para la asignación de un paquete arbitrario en un módulo de entrada hacia su módulo de salida, el estado de A_i y su correspondiente B_j puede ser evaluado según la figura No. 4.2 mostrada, donde cualquier par vertical de celdas que representan a las líneas físicas pueden ser escogidas como una ruta válida de interconexión.

Un programa o plan total será perfecto si todos los paquetes están satisfactoriamente asignados y sin encaminamientos en conflicto. Sin embargo, un plan óptimo requiere de información global de todo el tráfico T , y su implementación tiende a ser muy compleja. Afortunadamente, es bien conocido que el conocimiento parcial de T es suficiente para proyectar soluciones no tan

óptimas que son fáciles de implementar y adecuar, esto es con una eficiencia del 80%, tal como puede verse en [24].

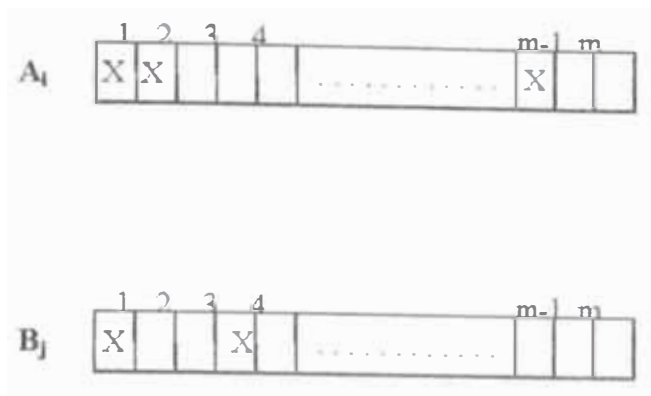


Figura No. 4.2: Comparación de A_i y B_j .

(X= ocupado, blanco= libre)

4.2. Algoritmo de enrutamiento distribuido.

Un requerimiento de suma importancia en un enrutamiento o establecimiento de un algoritmo es que debe de ser sumamente rápido en el procesamiento, por que los paquetes tienen que ser asignados a rutas en bases de celda por celda. Consecuentemente, el algoritmo debe ser implementable en hardware. Un controlador central debe ser usado para almacenar la información de todo el tráfico y luego derivarlo a un programa o proceso óptimo. Sin embargo, el lograrlo y distribuir la información requerida se convertirá en un cuello de botella severamente limitante para la velocidad de asignación resultante. Por tanto, en el presente trabajo se propone un algoritmo de enrutamiento distribuido sub-óptimo que es muy rápido y fácilmente implementable en hardware.

Es conveniente incluir la asignación de hardware como parte de cada módulo de entrada. Consideremos el primer módulo de entrada en particular. Deben haber un total de n paquetes llegando en un mismo instante, y el hardware debe de asignar estos n paquetes dentro de alguna duración definida como Δ y que sea menor que un tiempo de celda. Nosotros, además dividimos el Δ en K mini-celdas de longitud Δ/K , y luego concentramos nuestro estudio en la operación dentro de una mini-celda. Durante cada mini-celda, el B_j de un módulo específico de salida estará disponible para la asignación a un módulo de entrada

A_1 , y así sucesivamente, comparando A_i versus B_j . Nótese que hay más de n comparaciones requeridas en cada mini-celda tanto como hay n paquetes de entrada. Mientras A_i es comparado con B_j , simultáneamente $A_{i+1(\text{mod } K)}$ es comparado con el $B_{j+1(\text{mod } K)}$, y así sucesivamente, de tal forma que permite una asignación paralela. En el siguiente minislots, cada B_j es movido hacia el siguiente A_i (lo mismo sucede con los $A_{i+1(\text{mod } K)}$) en una forma cíclica, tal como se muestra en la figura No. 4.3.

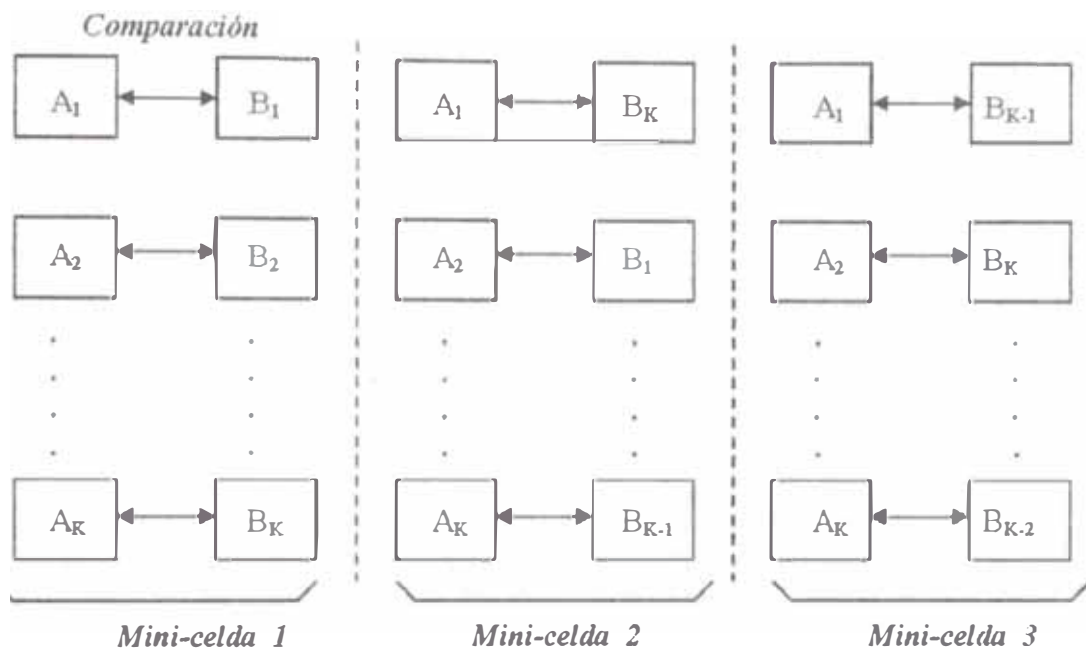


Figura No. 4.3: Asignaciones paralelas en forma cíclica de mini-celda a mini-celda

Desde que existen K módulos de salida, un total de K mini-celdas son necesarias en cada celda. La operación de comparar A_i con B_j puede ser implementada mediante la tecnología VLSI para altas velocidades.

En hardware, todos los B_{j_s} ($j=1, \dots, k$) están representados por un bus del bit m que está interconectado a través de los módulos de entrada, permitiendo el acceso de los bit m de los B_{j_s} hacia cada módulo de entrada durante cada mini-celda. En esta forma, los B_{j_s} están distribuidos sobre todas las entradas para asignaciones simultáneas.

Desde que las asignaciones de las rutas tienen que ser por paquetes, la velocidad de la asignación del hardware determina la longitud mínima del paquete

tanto como el tamaño final del conmutador. Mediante la tecnología actual de los VLSI se pueden superar fácilmente conmutadores de 1000 x 1000 con variación en línea superior a 150 Mb/s estableciéndose la longitud mínima del paquete del orden de 300 bits, lográndose además reducir el tiempo de trabajo predeterminado.

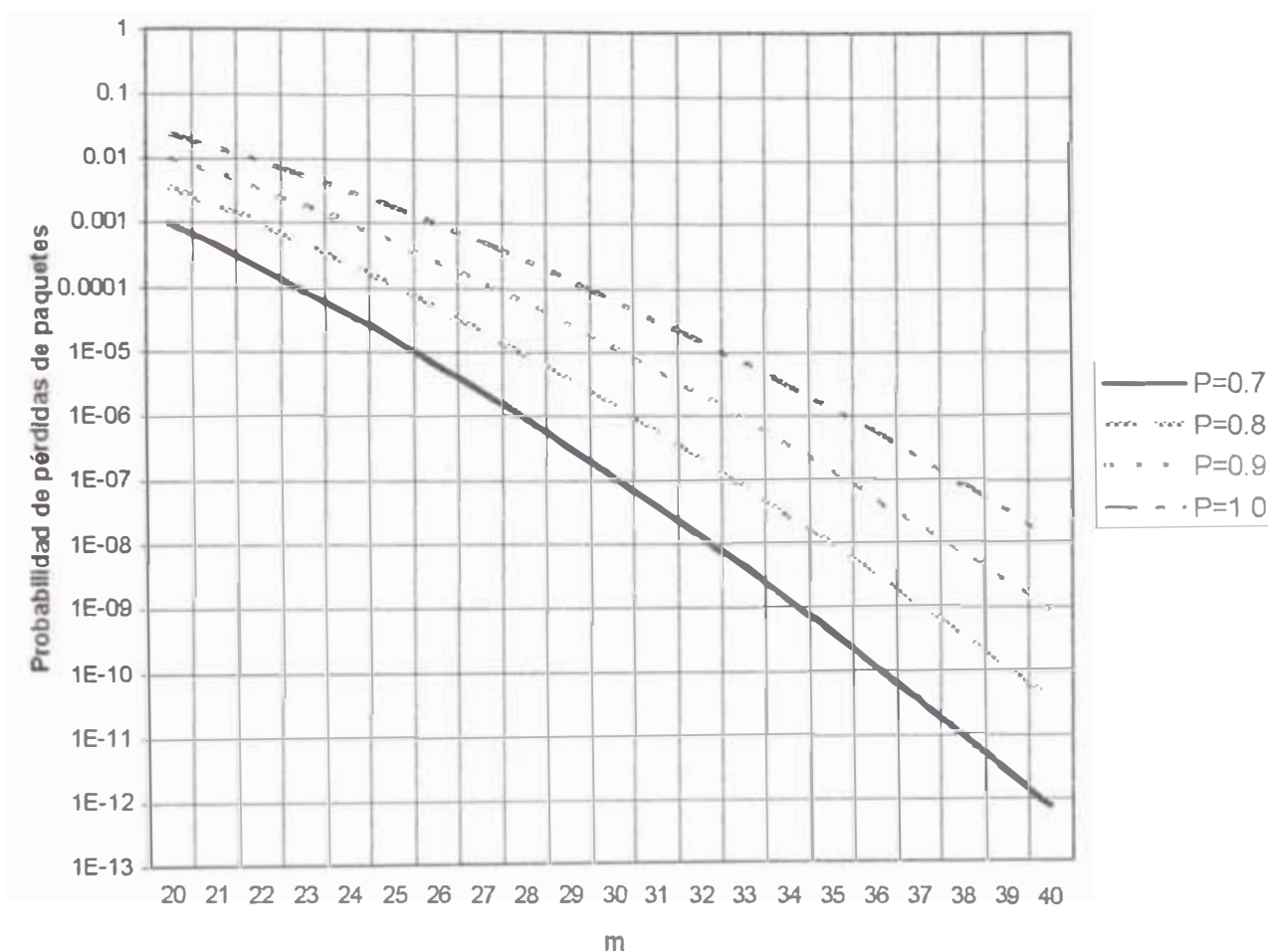


Figura No. 4.4: Probabilidad de pérdida de paquetes en el principio del Knockout generalizado.

4.3. Análisis del desempeño.[4]

Debido a que la arquitectura del conmutador usa un buffer de cola de espera en la salida, presenta mejor desempeño, throughput y menor retardo, por lo que sólo se requiere calcular la probabilidad de pérdida de celdas o paquetes. Existen dos causas que generan pérdidas de celdas en un conmutador ATM de arquitectura knockout: Pérdida por Knockout (si llegan demasiadas celdas a los

puertos de entrada destinadas para un mismo grupo de puertos de salida) y pérdida debido al procesamiento subóptimo (como solo un controlador de asignación de rutas distribuida es usado para la velocidad y eficiencia, se pierden las celdas cuando el controlador pierde control sobre una ruta a través del conmutador). Para el presente análisis se desprecia cualquier tipo de pérdida de celdas que ocurran en el módulo de salida del conmutador de paquetes debido a la sobresaturación del buffer, ya calculados para diversos tamaños de buffers, cualquier módulo de conmutador de paquetes y cualquier módulo de tráfico.

En el presente trabajo se hace un análisis de una condición límite superior de la probabilidad de pérdida de celdas para patrones de llegada arbitrarios de celdas independientes. La condición es para el peor caso y últimos puertos de entrada y salida determinados o definidos. Se asume que las celdas que llegan a diferentes puertos de entrada son independientes.

Sea $P_{I,j}$ la probabilidad que una celda llegue al puerto de entrada I ($0 \leq I \leq N-1$) destinado al módulo de salida j ($0 < j < k-1$). El subíndice I en $P_{I,j}$ indica el puerto de entrada, mientras que i indicará el módulo de entrada. Desde que la determinación está hecha en una base de slot por slot, $P_{I,j}$ cambiará de un tiempo de celda al siguiente.

En primer término se analizará el caso ilustrado en la figura No. 3.7, en el cual las rutas internas entre los módulos no están definidas. Posteriormente se analizará la situación con rutas internas definidas entre módulos en sucesivos estados, representados en la figura No. 3.8.

4.3.1. Rutas internas no definidas entre los módulos en sucesivos estados

Una celda puede ser llevada desde el módulo de entrada i hasta el módulo de salida j , a lo largo del camino o ruta a través del módulo del segundo estado, solamente si los elementos que se corresponden en A_i y B_j tiene valor 0, tal como se muestra en la figura No. 4.5. El proceso esquematizado en la figura No. 4.5 es conocido como "competencia directa". Si existen muchos caminos disponibles (más de uno en competencia), uno es seleccionado en forma aleatoria con igual

probabilidad. El desempeño del algoritmo será diferente que el indicado aquí, si la selección del camino de enrutamiento se hace en forma no aleatoria cuando existen muchas celdas en competencia.

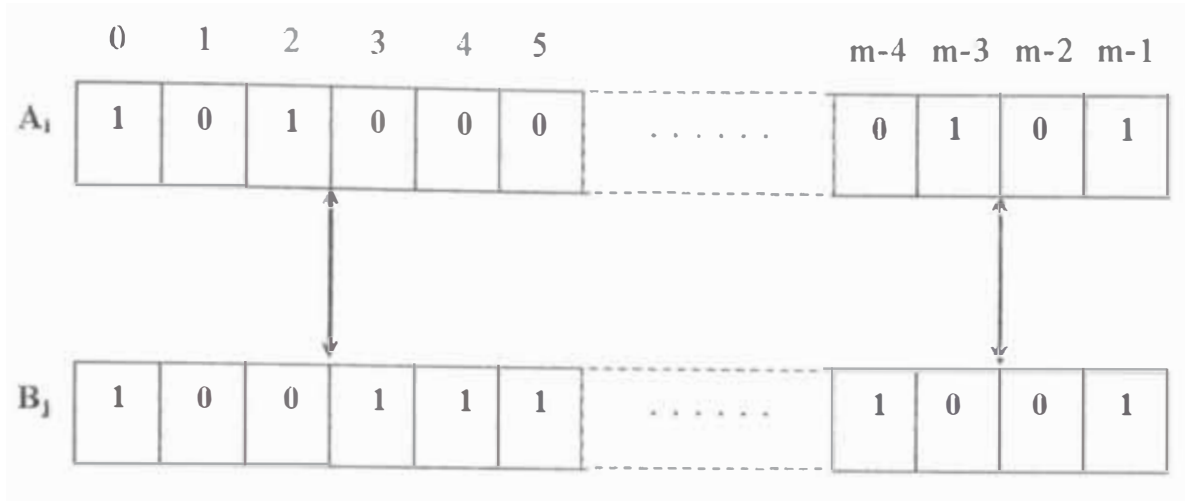


Figura No. 4.5: Selección del camino en "competencia directa".

Donde: 0 representa un camino disponible.
 1 representa un camino ocupado.

Supongamos $(m-r)$ elementos de A_i y $(m-s)$ elementos de B_j tienen el valor 0. Esto corresponde a r que ocupa el camino que va hacia la salida desde el módulo de entrada i y s ocupa el camino entrante que lleva hacia el módulo de salida j . Se asume que estos "0" elementos en A_i y B_j son localizados en forma aleatoria con distribuciones uniformes e independientes. El criterio asumido es apropiado desde que no existe un criterio global normalizado en el esquema de enrutamiento y desde que el camino es seleccionado en forma aleatoria cuando existen muchas competencias. En el intento de asignar un camino para una celda desde el módulo de entrada i hacia el módulo de salida j , las ubicaciones de los elementos "0" en A_i y B_j puede no ser independiente si es que otras celdas ya han sido definidos en el uso de A_i y B_j . Estas dependencias, entonces, aumentan la probabilidad de hallar *competencias directas* forzando algún o algunos **unos (1s)** en A_i alinearse con los **unos (1s)** en B_j . En el cálculo de la condición límite superior en la probabilidad de pérdida de celdas, creo conveniente ignorar las ventajas de tales dependencias.

Existen c competencias directas (representando c caminos disponibles desde A_i hasta B_j si la distribución de los *ceros* ($0s$) en B_j es tal que exactamente c de ellos están alineados con los *ceros* de A_i y los restantes $(m-s-c)$ *ceros* de B_j estén alineados con los *unos* de A_i . Este escenario de que c competencias directas puedan ocurrir en $\binom{m-r}{c} \binom{r}{m-s-c}$ formas de un total de $\binom{m}{m-s}$ distribuciones posibles de los $(m-s)$ 0 's en B_j . Por tanto, dados r y s , la probabilidad $P_{C|RS}(c/r,s)$ de que se tengan c competencias directas está distribuida en forma hiper-geométrica:

$$P_{C|RS}(c/r,s) = \frac{\binom{m-r}{c} \binom{r}{m-s-c}}{\binom{m}{m-s}} \quad (35)$$

con: $\text{máx.}\{0, (m-r-s)\} \leq c \leq \text{mín}\{(m-r), (m-s)\}$.

Cuando $(r+s)$ es menor que m , habrán al menos $(m-r-s)$ competencias directas y obviamente el número de competencias directas no debe exceder a $(m-r)$ ni $(m-s)$. El rango de c en la ecuación (35) está definida por dichas restricciones.

A partir de la ecuación (35), la probabilidad de que no existan competencias directas para r y s dados, tal que $r+s \geq m$, es:

$$P_{C|RS}(0/r,s) = \frac{\binom{m-r}{0} \binom{r}{m-s}}{\binom{m}{m-s}} = \frac{r!s!}{(r+s-m)m!} \quad (36)$$

Para: $r < m, \quad s < m, \quad (s+r) > m.$

Supongamos que una celda destinada para el j -ésimo módulo de salida llega al i -ésimo módulo de entrada. La celda se perderá en la entrada durante el

proceso de asignación del camino si es que no existen *competencias directas* de 0 's cuando el vector de caminos disponibles del módulo de entrada A_i es comparado con B_j , que es el vector de caminos disponibles del módulo de entrada. Esta probabilidad de pérdida de celdas está dada por la ecuación (36), asumiendo que existen r caminos ocupados del módulo de entrada y s caminos ocupados del módulo de salida.

Desde que (36) es una función no decreciente en r , se obtiene la condición límite en la probabilidad de pérdida de celdas $P_{pérdida}$, para todos los patrones posibles que lleguen, considerando que: $r=(n-1)$ en (36). Es decir, se asume que hacia todos los otros puertos de entrada del *i-ésimo* módulo de entrada llegan celdas y estas otras $(n-1)$ celdas también son definidas satisfactoriamente. Si el tráfico incluye algunas conexiones *multicast*, entonces celdas adicionales pueden haber sido asignadas a las rutas y es necesario usar un valor mayor que $(n-1)$ para r en la ecuación (36); en caso contrario el siguiente análisis también es válido para el caso multicast. No se considera en el presente análisis conexiones multicast.

Sea S el número de celdas destinadas al j -ésimo módulo de salida entre todas las otras celdas que llegan a las $(N-1)= (nk-1)$ líneas de entrada. Entonces siendo la ecuación (36) una función no decreciente en s , y ya que como máximo m celdas pueden ser dirigidas hacia el módulo de salida, luego la probabilidad condicional de pérdidas de celdas $P_{pérdida|S}(s)$ podemos definir en la forma siguiente:

$$P_{pérdida|S}(s) \leq \begin{cases} \frac{(n-1)s!}{(n-1+s-m)!m!} & ; \quad m-n+1 \leq s \leq m \\ 1 & ; \quad m+1 \leq s \leq N-1 \end{cases} \quad (37)$$

Multiplicando (37) por $P_S(s)$ y sumando respecto a s se obtiene:

$$P_{pérdida} \leq \sum_{s=m-n+1}^m P_S(s) \frac{(n-1)s!}{(n-1+s-m)!m!} + \sum_{s=m+1}^{N-1} P_S(s) \quad (38)$$

Desde que $\frac{(n-1)s!}{(n-1+s-m)!m!}$ es una función no decreciente de s de acuerdo al patrón de llegada de las celdas al módulo de entrada se puede definir así:

$$W_s \equiv \left\{ \begin{array}{ll} \frac{(n-1)s!}{(n-1+s-m)!m!} & ; \quad s = m-n+1 \\ \frac{(n-1)s!}{(n-1+s-m)!m!} - \frac{(n-1)(s-1)!}{(n-1+(s-1)-m)!m!} & ; \quad m-n+1 < s \leq m \end{array} \right\} \quad (39)$$

Luego la parte derecha de la ecuación (38) se puede escribir como:

$$\sum_{s=m-n+1}^m P_s(s) \frac{(n-1)s!}{(n-1+s-m)!m!} + \sum_{s=m+1}^{N-1} P_s(s) = \quad (40)$$

$$W_{m-n+1} \sum_{s=m-n+1}^{N-1} P_s(s) + W_{m-n+2} \sum_{s=m-n+2}^{N-1} P_s(s) + \dots + W_m \sum_{s=m}^{N-1} P_s(s)$$

Donde cada W_s es positiva. También se puede escribir la ecuación anterior en la forma siguiente:

$$\sum_{s=m-n+1}^M P_s(s) \frac{(n-1)s!}{(n-1+s-m)!m!} + \sum_{s=m+1}^{N-1} P_s(s) = W_{m-n+1} + W_{m-n+2} + \dots + W_m -$$

$$P_r(0 \leq S \leq m-n)W_{m-n+1} - P_r(0 \leq S \leq m-n+1)W_{m-n+2} - \dots - P_r(0 \leq S \leq m-1)W_m \quad (41)$$

Para obtener el límite superior en la sumatoria en (41) debemos minimizar el término $P_r(0 \leq S \leq z)$, para $z=m-n, m-n+1, \dots, m-1$. Como el proceso es una distribución del número de sucesos en muestras independientes se tendrá que las probabilidades $P_r(0 \leq S \leq z)$ son minimizadas cuando $P_{i,j} = \frac{E\{S\}}{(N-1)}$ para $0 \leq i < (N-1)$, siendo $z \geq E\{S\}$. Debido a que el módulo de salida del conmutador de paquetes sólo tiene n puertos de salida, el patrón del tráfico debe ser tal que $E\{S\} < n$. Luego si $(m-n) \geq n$ o equivalentemente $m > 2n$, entonces la distribución uniforme de $P_{i,j}$ minimizará todos los términos de en la ecuación (41). En otras

palabras, si $m > 2n$ entonces las probabilidades $P_r(0 < S < z)$ son minimizadas cuando S es distribuida en forma binomial:

$$P_s(s) = \binom{N-1}{s} \left(\frac{np}{N-1}\right)^s \left(1 - \frac{np}{N-1}\right)^{N-1-s}, \text{ para } 0 \leq s \leq N-1 \quad (42)$$

Donde $p = \frac{E\{S\}}{n}$ es la carga normalizada entregada al j -ésimo módulo de salida. Sustituyendo la ecuación (42) para $P_s(s)$ en la ecuación (38) se obtiene el límite superior en la probabilidad de pérdidas de celdas para todos los posibles patrones de tráfico que llegan:

$$P_{\text{pérdida}} \leq \sum_{s=m-n+1}^{m-1} \binom{N-1}{s} \left(\frac{np}{N-1}\right)^s \left(1 - \frac{np}{N-1}\right)^{N-1-s} \cdot \frac{(n-1)s!}{(n-1+s-m)m!} + \sum_{s=m}^{N-1} \binom{N-1}{s} \left(\frac{np}{N-1}\right)^s \left(1 - \frac{np}{N-1}\right)^{N-1-s} \quad (43)$$

En la ecuación (43) la segunda sumatoria representa al componente de la probabilidad de pérdida debido al Knockout del total de la probabilidad de pérdidas de celdas $P_{\text{pérdida}}$. La primera sumatoria en la misma ecuación (43) es la pérdida adicional debido a la determinación sub-óptima.

Para valores de n y m fijos (tal que el conmutador $N \times N$ esté compuesto de pequeños módulos de conmutación de paquetes), $N \rightarrow \infty$ cuando $k \rightarrow \infty$. Tomando límite en la ecuación (43) cuando $k \rightarrow \infty$, se obtiene un límite superior en la probabilidad de pérdidas de celdas para un tamaño arbitrario de conmutadores y para todos los posibles patrones de tráfico que llegan será:

$$P_{\text{pérdida}} \leq \sum_{s=m-n+1}^{m-1} e^{-np} \frac{(np)^s}{s!} \cdot \frac{(n-1)s!}{(n-1+s-m)m!} + \sum_{s=m}^{\infty} e^{-np} \frac{(np)^s}{s!}$$

$$P_{\text{pérdida}} \leq 1 + \sum_{s=m-n+1}^{m-1} e^{-np} \frac{(np)^s (n-1)}{(n-1+s-m)m!} - \sum_{s=0}^{m-1} e^{-np} \frac{(np)^s}{s!} \quad (44)$$

Considerando $n=1$ en la ecuación (44), la pérdida debido al Knockout $1 - \sum_{s=0}^{m-1} \frac{(e^{-p})^s p^s}{s!}$ coincide con la fórmula dada por (39) que es la pérdida de celdas en la entrada debido al Knockout para el peor caso[9].

Para comparar la pérdida debido al *Knockout* con las pérdidas debidas al *Knockout + scheduling*, se debe notar en la ecuación (44) que $\frac{(np)^s}{(n-1+s-m)!}$ es maximizada cuando $s = \lceil np + m - n \rceil$ donde $\lceil v \rceil$ es el entero más pequeño y mayor o igual que v . Por tanto, se puede definir la probabilidad de pérdida por la siguiente ecuación:

$$P_{p\acute{e}rdida} \leq 1 + e^{-np} \frac{n!}{m!} \cdot \frac{(np)^{\lceil np+m-n \rceil}}{(\lceil np-1 \rceil)!} - \sum_{s=0}^m e^{-np} \frac{(np)^s}{s!} \quad (45)$$

En la ecuación (45) dado que: $e^{-np} \frac{(np)^{\lceil np+m-n \rceil}}{(\lceil np-1 \rceil)!}$ es una función no decreciente de p y considerando $p=1$ (100% de carga) se tiene:

$$P_{p\acute{e}rdida} \leq 1 + \frac{n^{m+1}}{m!} e^{-n} - \sum_{s=0}^m e^{-n} \frac{e^s}{s!} = \sum_{s=m}^{\infty} e^{-n} \frac{n^s}{s!} + (n-1) \frac{n^m}{m!} e^{-n}. \quad (46)$$

Finalmente, usando la ecuación (46) y la condición del límite inferior de la probabilidad de pérdida para la pérdida por Knockout se tendrá:

$$P_r(m \leq S) \leq P_{p\acute{e}rdida} \leq P_r(m \leq S) + (n-1)P_r(S = m). \quad (47)$$

En la ecuación (47), $P_r(m \leq S)$ es la pérdida por Knockout y $(n-1)P_r(S = m)$ es la pérdida debido a la determinación.

Se debe notar que la transformación *espacio a tiempo* hace la determinación aleatoria del conmutador *tiempo-espacio-tiempo* (TST) que es similar al *proceso de competencia* usado en el algoritmo de determinación subóptima, tal como se puede ver en la figura No. 4.5; pero obviamente no son idénticos. La competencia de los *tiempos de celda* en un conmutador TST es simétrica cuando el máximo número de *slots* en las entradas y en las salidas es la misma. Pero la competencia de las rutas no es simétrica, porque el número máximo de rutas disponibles desde el módulo de entrada es n y el máximo número de rutas disponibles para el módulo de salida es m . Además, la asimetría y el desempeño de un circuito de conmutación TST se evalúa mediante el análisis *estado-estático*, donde sea requerido otra técnica de análisis para un conmutador de paquetes ATM, porque este conmutador encamina las celdas en forma independiente, aleatoria y slot por slot. Las celdas no determinadas durante el **tiempo de celda** en la llegada son eliminadas antes de que se inicie el siguiente **tiempo de celda** y el estado de todas las rutas son limpiadas para el acceso de nuevas celdas al inicio de cada **tiempo de celda**.

4.3.2. Determinación de la ruta interna entre estados.

Dado que se ha asumido la necesidad de una competencia directa entre A_i y B_j para establecer una ruta a través de los conmutadores elementales desde el i -ésimo módulo de entrada hasta el j -ésimo módulo de salida; estos requerimientos pueden ser dejados de lado al establecerse las principales rutas internas. En la figura No. 4.8 existen t líneas entre los módulos del primer y el segundo estado (igualmente entre los módulos del segundo y el tercer estado). Existen $M=m/t$ troncales asociadas con cada módulo de entrada (o salida), uno a cada uno de los M módulos del segundo estado. El análisis en la sección 4.3.1 fue para $t=1$ y $M=m$.

La probabilidad de pérdidas de celdas será minimizada estableciendo rutas internas principales entre estados. Desde que una celda puede usar algunas de las t rutas entre los módulos, *competencias diagonales* (mostrada en la figura No. 3.8 para $t=2$) además de las *competencias directas*, pueden ser usados para establecer una ruta desde el módulo de entrada hasta el módulo de salida. Tal

como aumenta t el número de rutas adicionales se incrementan y la probabilidad de pérdidas de las celdas se aproxima a la pérdida por Knockout y el cual ocurre exactamente cuando $t=m$.

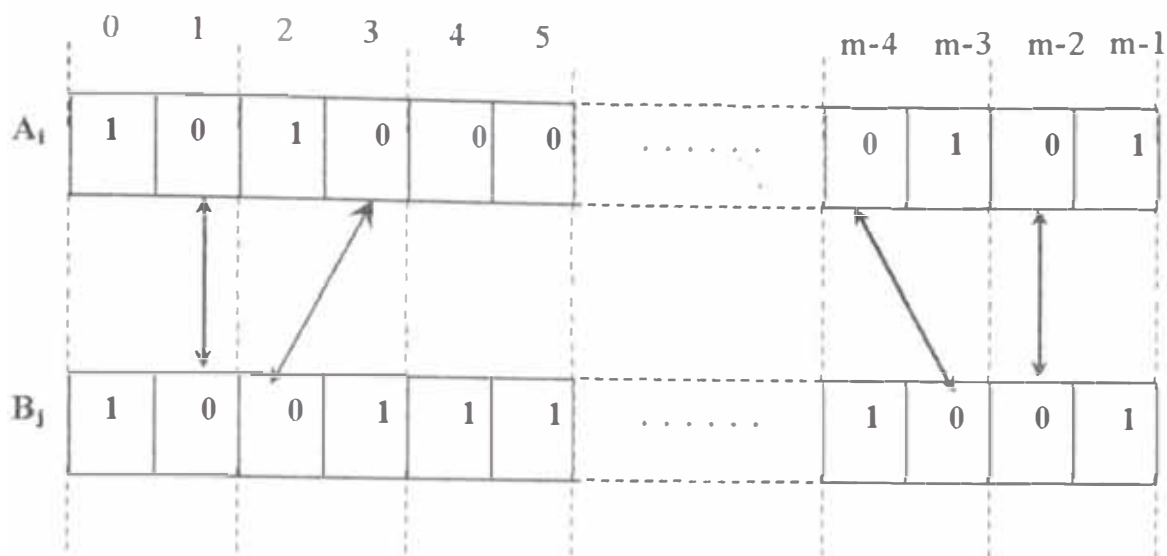


Figura No. 4.8: Selección del camino en "competencia diagonal".

Donde: 0 representa un camino disponible.
 1 representa un camino ocupado.

La complejidad de la fábrica conmutacional se incrementa por que el tamaño de los módulos del segundo estado se incrementan desde $K \times K$ hasta $tK \times tk$. En el caso extremo en el que $t=m$ y $M=1$, el segundo estado del sistema se convierte en un solo módulo completamente conectado de $mK \times mk$ (es decir igual a: $N\binom{m}{n} \times N\binom{m}{n}$), y no existe pérdidas debido a la determinación sub-óptima. En esta parte se establece un límite superior en la probabilidad de pérdidas de celdas para varios valores en el tamaño de t para un patrón arbitrario de celdas independiente que llegan a la entrada del sistema, tal que esta decisión justifique el incremento de la complejidad para la mejora del desempeño. Tal como se ha expresado líneas arriba, se asume que existe competencia aleatoria cuando existen muchas posibilidades.

Para obtener un límite superior en la probabilidad de pérdidas de celdas para todos los patrones de llegada posibles, se sigue un procedimiento similar al usado en el párrafo anterior. La única diferencia es que se debe calcular la

probabilidad de no tener competencias directa ni diagonal, cuando una celda destinada para el j -ésimo módulo de salida llegue hacia el i -ésimo módulo de entrada.

Sea R_o el número de todas las troncales ocupadas que salen del i -ésimo módulo de entrada. Dados que existen R caminos de salida ocupados desde el i -ésimo módulo de entrada, la distribución condicional estará dada por la siguiente relación matemática:

$$P_{R_o}(r_o / r) = \frac{\sum_{h=0}^{\lfloor \frac{r-r_o}{t} \rfloor} (-1)^h \binom{r_o+h}{h} \binom{M}{r_o+h} \binom{m-t(r_o+h)}{r-t(r_o+h)}}{\binom{m}{r}} \quad (48)$$

$$\max\{0, r - (m - M)\} \leq r_o \leq \lfloor r/t \rfloor$$

El denominador de la ecuación (48) es el número de formas de distribuir los r caminos ocupados entre los m caminos que salen desde el i -ésimo módulo de entrada. El numerador de la misma ecuación representa al número de formas de obtener exactamente los r_o caminos troncales completamente ocupados de los M caminos troncales salientes.

No pueden llevarse a cabo las *competencias directa* ni la *diagonal* si y solo si todas las líneas de entrada disponibles dirigidas hacia el j -ésimo módulo de salida viene desde los módulos del segundo estado que corresponden a estos r_o caminos troncales de entrada completamente ocupados. En otras palabras, no existen competencias si los $(M-r_o)$ caminos troncales hacia el j -ésimo módulo de salida están completamente ocupados. Cuando $s \geq t(M-r_o)$ existen $\binom{tr_o}{s-t(M-r_o)}$ formas para que ocurra la presente propuesta; por tanto, en vez de la ecuación (36), la probabilidad de no tener competencia directa o diagonal para un r y s dados es:

$$P_{D/R_s}(0/r,s) = \sum_{r_0} P_{R_0/R(r_0/r)} \frac{\binom{tr_0}{t-t(M-r_0)}}{\binom{m}{s}} \quad (49)$$

Haciendo $r=(n-1)$ en las ecuaciones (48) y (49) y sustituyendo $P_{D/R,S}(0/n-1,s)$ por $P_{C/R,S}(0/n-1,s) = \frac{(n-1)s!}{(n-1+s-m)m!}$ en las ecuaciones (43) y (44) se determinan los siguientes límites en la probabilidad de pérdidas de celdas para todos los patrones de tráfico de llegada de celdas:

$$P_{p\acute{e}rdida}(t) \leq \sum_{s=m-n+1}^{m-1} \binom{N-1}{s} \left(\frac{np}{N-1}\right)^s \left(1 - \frac{np}{N-1}\right)^{N-1-s} P_{D/R,S}(0/n-1,s) + \sum_{s=m}^{N-1} \binom{N-1}{s} \left(\frac{np}{N-1}\right)^s \left(1 - \frac{np}{N-1}\right)^{N-1-s} \quad (50)$$

Luego cuando $N \rightarrow \infty$:

$$P_{p\acute{e}rdida}(t) \leq 1 + \sum_{s=m-n+1}^{m-1} e^{-np} \frac{(np)^s}{s!} P_{D/R,S}(0/n-1,s) - \sum_{s=0}^{m-1} e^{-np} \frac{(np)^s}{s!} \quad (51)$$

4.4. Probabilidad de pérdidas de celdas: Rangos más precisos para un modelo de tráfico uniforme.

En esta sección se hace el cálculo de rangos más precisos en el cálculo de la probabilidad de pérdidas de celdas para un modelo de tráfico uniforme e independiente. La distribución de celdas que llegan tienen una distribución idéntica en el conmutador. En particular, sea ρ que representa a la probabilidad de celdas que llegan al puerto de entrada en un *tiempo de celda*. Se asume que las celdas que llegan a los diferentes puertos de entrada son independientes y las celdas llegan con una distribución uniforme para las diferentes salidas, tal que una celda está destinada hacia uno de los k módulos de salida con una probabilidad de $1/k$. Para calcular la pérdida de celdas se requiere especificar solamente el módulo destino de salida y no así el puerto destino de salida.

Sea una celda destinada para el *j-ésimo* módulo de salida que llega al *i-ésimo* módulo de entrada. Para esta consideración, el número R de celdas que llegan al *i-ésimo* módulo de entrada en los (n-1) puertos está distribuido en forma binomial con los parámetros (n-1) y ρ en la forma siguiente:

$$P_R(r) = \binom{n-1}{r} \rho^r (1-\rho)^{n-1-r}; \text{ para: } 0 \leq r \leq (n-1) \quad (52)$$

De estas R celdas, sean X celdas que también están destinadas al *j-ésimo* módulo de salida estará distribuida binomialmente en la siguiente forma:

$$P_{X/R}(x/r) = \binom{r}{x} \left(\frac{1}{k}\right)^x \left(1 - \frac{1}{k}\right)^{r-x}; \quad \text{Para: } 0 \leq x \leq r. \quad (53)$$

Del mismo modo, el número Y de celdas de los otros módulos de entrada (ya no el *i-ésimo*) que están destinadas hacia el *j-ésimo* módulo de salida está distribuido binomialmente en la forma siguiente:

$$P_Y(y) = \binom{n(k-1)}{y} \left(\frac{\rho}{k}\right)^y \left(1 - \frac{\rho}{k}\right)^{n(k-1)-y}, \text{ para: } 0 \leq y \leq n(k-1). \quad (54)$$

Por tanto, el número total S de celdas destinadas hacia el *j-ésimo* módulo de salida entre los que llegan hacia los (N-1) = (nk-1) líneas de entrada es:

$$S = X + Y \quad (55)$$

4.4.1. Sin ruta principal entre los módulos en estados sucesivos.

Se limita el $P_{pérdida}$ asumiendo que todas las R celdas y todas las S celdas mayor al máximo valor de m, ya han sido satisfactoriamente programados o concluidos. Notar que X 1's en A_i se conectan con los X 1's en B_j , y combinando las ecuaciones (36), (52), (53) y (54) se obtiene el siguiente rango más alto de

$P_{pérdida}$:

$$P_{pérdida} \leq \sum_{r=0}^{n-1} P_R(r) \sum_{x=0}^r P_{X/R}(x/r) \left[\sum_{y=m-r}^{m-x-1} P_Y(y) \frac{(r-x)y!}{(r+y-m)(m-x)} + \sum_{y=m-x}^{n(k-1)} P_Y(y) \right] \quad (56)$$

Tomando el límite de la ecuación (56) cuando $k \rightarrow \infty$ (corresponde a $N \rightarrow \infty$ con n y m definidos), se obtiene:

$$\begin{aligned}
 P_{\text{pérdida}} &\leq \sum_{r=0}^{n-1} P_R(r) \left[\sum_{y=m-r}^{m-1} e^{-np} \frac{(np)^y}{y!} \frac{r!y!}{(r+y-m)m!} + \sum_{y=m}^{\infty} e^{-np} \frac{(np)^y}{y!} \right] \\
 &= 1 + \sum_{r=1}^{n-1} P_R(r) \sum_{y=m-r}^{m-1} e^{-np} \frac{(np)^y r!}{(r+y-m)m!} - \sum_{y=0}^{m-1} e^{-np} \frac{(np)^y}{y!}
 \end{aligned} \tag{57}$$

Con una carga del 100% la ecuación (57) es equivalente con la ecuación (44).

4.4.2. Ruta interna principal entre los estados.

Sustituyendo $P_{D/R,S}(0/r, x+y)$ en la ecuación (56) y en la ecuación (57) se obtienen los siguientes rangos en la probabilidad de pérdida de celda:

$$P_{\text{pérdida}} \leq \sum_{r=0}^{n-1} P_R(r) \sum_{x=0}^r P_{X/R}(x/r) \left[\sum_{y=\max\{0, (m-x-r)\}}^{m-x-1} P_Y(y) P_{D/R,S}(0/r, x+y) + \sum_{y=m-x}^{n(k-1)} P_Y(y) \right] \tag{58}$$

y cuando $N \rightarrow \infty$ se tiene:

$$\begin{aligned}
 P_{\text{pérdida}}(t) &\leq \sum_{r=0}^{n-1} P_R(r) \left[\sum_{y=m-r}^{m-1} e^{-np} \frac{(np)^y}{y!} P_{D/R,S}(0/r, y) + \sum_{y=m}^{\infty} e^{-np} \frac{(np)^y}{y!} \right] \\
 &= 1 + \sum_{r=1}^{n-1} P_R(r) \sum_{y=m-r}^{m-1} e^{-np} \frac{(np)^y}{y!} P_{D/R,S}(0/r, y) - \sum_{y=0}^{m-1} e^{-np} \frac{(np)^y}{y!}
 \end{aligned} \tag{59}$$

La sumatoria sobre y en la ecuación (58) se inicia en $\max\{0, (m-x-r)\}$ en vez de iniciar en $(m-r)$, como en la ecuación (56). Con una carga del 100% la ecuación (59) coincide con la ecuación (51).

Luego para este caso, la probabilidad de pérdida de paquetes de entrada para el peor caso estará expresado por la siguiente relación matemática:

$$P_r = 1 - \sum_{K=0}^m \frac{(np)^K e^{-np}}{K!} \tag{60}$$

** Todas las ecuaciones desde la (35) hasta la (60) se han tomado de [4].

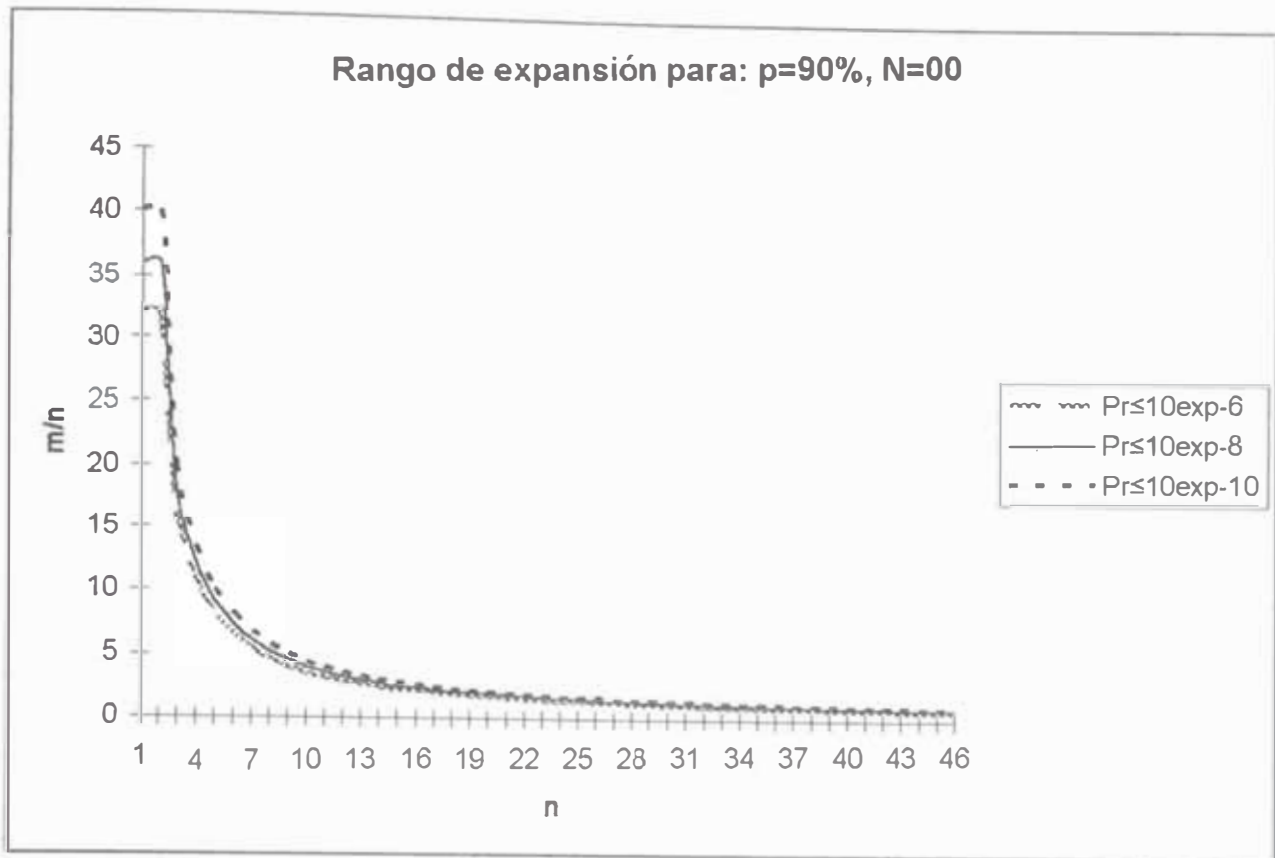


Figura No. 4.9: Relación de expansión requerida como una función del tamaño de paquete n para diferentes probabilidad de pérdidas de celdas, con una carga de 90% y $N=\infty$.

La figura No. 4.10 muestra las gráficas que son el ejemplo del comportamiento típico del modelo probabilístico para el peor caso, o sea para la ecuación (60).

4.5. Resultados Numéricos.

Para un grupo de salida de tamaño $n=16$, en la figura No. 4.6 se muestra el rango más alto en la probabilidad de pérdida de celda para un *patrón arbitrario de celdas independientes que llegan*. La probabilidad de pérdida de celda se muestra para una carga del 90% como una función del parámetro de expansión m y para varios valores de N que expresa los diferentes tamaños de los conmutadores y para la consideración de que no existe un camino troncal interno ($t=1$). Las ecuaciones que se han graficado son el (43) y (44) respectivamente. Se puede notar que es mayor la probabilidad de pérdidas de paquetes cuando $N=\infty$ que cuando $N<\infty$ y como el objetivo fundamental del presente trabajo así como la arquitectura analizada es aumentar el tamaño de los conmutadores a grandes

Empleando el mismo valor del tamaño del grupo de salida, es decir con $n=16$, en la figura 4.7 se muestra un límite superior en la pérdida de paquetes para diferentes valores de cargas de tráfico en el módulo de salida. Es decir, aquí se puede ver que por ejemplo para los valores de $m=40, 38, 36$ y 34 se garantizarán que la probabilidad de pérdidas de celdas está por debajo de 10^{-6} para los diferentes patrones de llegada de las celdas a dicho módulo. En dicha figura se muestran los gráficos para cargas de tráfico 1.0, 0.9, 0.8 y 0.7 respectivamente. Del mismo modo, también se pueden ver que para garantizar una probabilidad de pérdidas de paquetes menor a 10^{-9} serán necesarios disponer de módulos de salida de conmutación de paquetes cuyas dimensiones sean de 47×16 , 45×16 , 42×16 y 39×16 para las cargas de tráfico indicados respectivamente.

En la figura 4.9 se muestra el rango de expansión requerida para garantizar que la probabilidad de pérdida de paquetes está por debajo de lo considerado, a su vez ilustra la ventaja que se tendrá al considerar las salidas agrupadas. Se considera $m \geq 33$ y un $n=16$, para el análisis de las simulaciones correspondientes.

En la figura 4.10 se comparan el límite inferior (pérdidas sólo debido al Knockout) y el del límite superior (pérdidas por Knockout + scheduling) en la probabilidad de pérdidas de paquetes.

4.6. Desempeño del conmutador PINIUM.

4.6.1. Análisis de la pérdida en el desempeño del conmutador.

En esta sección, se desarrolla el desempeño general de la arquitectura PINIUM para un sistema de tráfico *unicast* y *multicast*, en forma de tráfico aleatorio y a ráfagas. Se considera que se ignora el hecho de que las celdas tienen diferentes prioridades. Aquí se define la carga de entrada como el número promedio de celdas que llegan hacia el puerto de entrada en un *time slot*. Sea ρ_{in} la carga de entrada a considerar para cada puerto de entrada y que cumpla la condición de que $\rho_{in} < 1$. Si a_i es la probabilidad de que i es el número de celdas destinadas para un mismo puerto de salida, y que P_{loss} es la probabilidad de pérdida promedio de celdas, entonces se tendrá la siguiente expresión general:

$$P_{loss} = \sum_{i=L+1}^N \left(\frac{i-L}{\rho_{in}} \right) a_i \quad (61)$$

Tráfico Aleatorio. - Se asume que el tráfico de entrada sigue la distribución de Bernoulli, y que hay infinitos ambientes de espera en los controladores del puerto de salida. En el ambiente de un tráfico unicast, se tendrá la siguiente relación matemática:

$$a_{loss} = \binom{N}{i} \left(\frac{\rho_{in}}{N} \right)^i \left(1 - \frac{\rho_{in}}{N} \right)^{N-i} \quad (62)$$

En un ambiente del peor desempeño, se tendrá que $N \rightarrow \infty$, entonces:

$$a_i = \left(\frac{\rho_{in}^i e^{-\rho_{in}}}{i!} \right) \quad (63)$$

Entonces, se tiene la fórmula ya conocida, que es la siguiente:

$$P_{loss} = \left(1 - \frac{L}{\rho_{in}} \right) \left(1 - \sum_{i=0}^L \frac{\rho_{in}^i e^{-\rho_{in}}}{i!} \right) + \frac{\rho_{in}^L e^{-\rho_{in}}}{L!}, \quad (64)$$

Para el caso de una carga de máxima entrada en un ambiente unicast de: $\rho_{in}=1$, y $L=8$ (ó $L=16$), se tendrá que P_{loss} aproximadamente de 10^{-6} (pero, en sí P_{loss} es de casi 10^{-14}).

En un ambiente de tráfico multicast, se asume que todas las celdas destinadas para un mismo "rótulo" de puerto de salida que vaya hacia una cola de espera imaginaria en un mismo *time slot* antes de ir a un proceso Knockout. Sea ρ_{out} la carga efectiva ofrecida al puerto de salida adherido. Con una estructura de árbol multicast del conmutador PINIUM, no hay un overflow mientras todas las copias deseadas son generadas.

Supongamos que cada puerto de entrada tiene un requerimiento de celdas de C_i copias multicast, donde: $0 \leq i < N$. Si los C_i son independientes e idénticamente distribuidos, y todos los puertos de salida están uniformemente cargados, entonces es fácil demostrarlo, empleando funciones de generación, tal

que el número de celdas que llegan hacia el rótulo de puertos de salida, siga la distribución binomial. Además, la función es solamente dependiente del valor de C_{prom} , que es el número promedio de copias requeridas por cada puerto de entrada. Por tanto para una cola de espera considerada imaginaria, se tendrá:

$$\rho_{out} = C_{prom} \rho_{in} \quad (65)$$

Por tanto, cuando la carga de entrada es alta, el valor de ρ_{out} puede ser tan grande que uno obtenido después del servicio multicast. Para el desempeño del sistema, se puede reemplazar el ρ_{in} por ρ_{out} en (62) y entonces se obtiene para (61):

$$P_{loss} = \sum_{i=L+1}^N \binom{i-L}{\rho_{out}} \binom{N}{i} \left(\frac{C_{prom}}{N} \rho_{in} \right)^i \left(1 - \frac{C_{prom}}{N} \rho_{in} \right)^{N-i} \quad (66)$$

Cuando $N \rightarrow \infty$, se obtiene una ecuación que es similar a la ecuación (64) y es el siguiente:

$$P_{loss} = \left(1 - \frac{L}{\rho_{out}} \right) \left(1 - \sum_{i=0}^L \frac{\rho_{out}^i e^{-\rho_{out}}}{i!} \right) + \frac{\rho_{out}^L e^{-\rho_{out}}}{L!} \quad (67)$$

Como un ejemplo, si asumimos que cada celda de entrada requiere un número de copias que sigue una distribución de geometría truncada con parámetro g , ($0 < g < 1$), y P_i es la probabilidad de requerir i números de copias por celda, entonces tendremos:

$$P_i = \frac{(1-g)g^{i-1}}{1-g^N}, \quad \text{para: } 1 \leq i \leq N. \quad (68)$$

Por tanto, el número promedio de copias requeridas en este caso será:

$$C_{prom} = \sum_{i=1}^N i \frac{(1-g)g^{i-1}}{1-g^N} = \frac{1}{1-g} - \frac{Ng^N}{1-g^N}, \quad (69)$$

Como N tiende a infinito, para el peor caso, tendremos:

$$C_{prom} = \frac{1}{(1-g)} \quad (70)$$

Empleando este resultado juntamente con (65) y (67) se estará listo para evaluar el desempeño de un conmutador PINIUM para un patrón de tráfico multicast.

Tráfico a Ráfagas.- Este tipo de tráfico no es tema del presente trabajo, pero considero importante hacer una breve mención que corrobore con la presente tesis en cuanto al uso, diseño de arquitectura y su análisis de desempeño del conmutador. El diseño del conmutador PINIUM sin buffers internos es un factor importante que provee al sistema una propiedad de no usar memoria. No hay un impacto en el desempeño del conmutador bajo el tráfico a ráfagas. Para investigar el desempeño del conmutador bajo la influencia de un tráfico a ráfagas, se requiere hallar la carga principal ofrecida tanto como las copias multicast requeridas. Por ejemplo, supongamos que el proceso de llegada a los puertos de entrada es un Markov idéntico de dos estados modulado con el proceso Bernoulli. La fuente es caracterizada por una matriz de transición siguiente:

$$\begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$$

y una variación de la matriz:

$$\begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix}$$

Donde α y β son probabilidades de transición entre dos estados, y γ_i , en el cual $i \in \{1,2\}$ es la probabilidad en el que la fuente en el estado i tiene una celda que llega al puerto de entrada en un time slot. La carga principal de entrada ofrecida es el siguiente:

$$\rho_m = \frac{\beta\gamma_1 + \alpha\gamma_2}{\beta + \alpha} \tag{71}$$

Entonces, este parámetro se puede usar en (65) y (67) para calcular la probabilidad de pérdida de celdas. Esta conclusión se verifica mediante el uso de la técnica de análisis matricial [25].

4.6.2. Análisis del retardo en el desempeño del conmutador.

Después de ir a través de la sección de concentración del conmutador PINIUM, el proceso de llegada de las celdas al controlador del puerto de salida sigue una distribución binomial truncada. Sea t_i que denote el número de celdas destinada hacia un rótulo del controlador del puerto de salida, entonces se tendrá:

$$t_i = \left\{ \begin{array}{ll} a_i & \text{para: } 0 \leq i \leq L. \\ \sum_{j=L}^N a_j = 1 - \sum_{j=0}^{L-1} a_j & , \text{ para: } i = L. \end{array} \right\}, \quad (72)$$

En este modelo de tiempo discreto, se asume que los que llegan y los que parten ocurren en el mismo time slot simultáneamente. Cada celda requiere un time slot para su procesamiento; en consecuencia, una celda que llega hacia un buffer de cola de espera vacío no puede ser distribuido instantáneamente. En el caso de un sistema de infinitos buffers de colas de espera, si X_i , para $i > 0$, denota la probabilidad de estado estático de la longitud del buffer de cola de espera, se tendrán las siguientes ecuaciones:

$$x_i = \left\{ \begin{array}{ll} (1-t_0)x_0 / t_0, & \text{para: } i = 1. \\ \left(x_{i-1} - x_0 t_{i-1} - \sum_{j=1}^{i-1} x_i - j t_j \right) / t_0, & \text{para: } 2 \leq i \leq L+1. \\ \left(x_{i-1} - \sum_{j=1}^L x_i - j t_j \right) / t_0, & \text{para: } L+2 \leq i. \end{array} \right\}, \quad (73)$$

y con:

$\sum_{i=0}^{\infty} x_i = 1$, se puede obtener el retardo promedio de una celda en el controlador del puerto de salida mediante la siguiente fórmula de Little [15]:

$$\text{Retardo esperado} = \frac{\text{longitud de buffer esperado}}{\text{carga ofrecida}} = \frac{\sum_{i=1}^{\infty} i x_i}{1 - x_0} \quad (74)$$

En la siguiente figura se muestra el retardo de celda en el controlador de puerto de salida con respecto a las diferentes cargas ofrecidas, para: $N \rightarrow \infty$ y $L=16$.

4.6.3. Breve análisis, evaluación y resultados de la arquitectura PINIUM.

Este conmutador tiene dos diferentes secciones, un conjunto de planos multicast y una fila de concentradores clasificadores priorizados. Para calcular el tamaño de estados, un arreglo en árbol de base r con N puertos de salida requiere $\log_r N$ estados. Un clasificador de concentradores priorizados consta de dos partes: Un clasificador L y de $2L$ a L concentradores mezcladores priorizados. Asumimos que ambos están hechos a partir del método Bitónico de Batcher [19], entonces, un clasificador L tendrá $(\frac{1}{2})\log_2 L(\log_2 L + 1)$ estados, y un concentrador de $2L$ a L tendrá $\log_2 2L$ estados. En un concentrador clasificador priorizado de N a L , se puede obtener N/L secuencias monotónicas clasificadas, usando una columna de L clasificadores. Entonces, otros $\log_2(N/L)$ estados de $2L$ a L mezcladores concentradores son usados para obtener las últimas L celdas. Para el caso del conmutador PINIUM, Ka y León García [2], asumen que $S_r(N, L)$ es el número total de estados con N entradas y un parámetro L del Knockout, entonces se tendrá:

$$S_r(N, L) = \log_2 N + (\log_2 L + 1)\log_2 N - \frac{\log_2 L(\log_2 L + 1)}{2}, \quad (75)$$

Para un conmutador Knockout, el número de estados varía, tal como se ha visto en [knockout], para una entrada de 8 hay 4 salidas concentradas y requiere 8 estados, para el caso de 8 entradas y dos salidas concentradas se requieren 5 estados. Por tanto, el conmutador PINIUM puede desempeñarse tan bien como un conmutador Knockout en términos de latencia. Además, L es pequeño cuando se

compara con N ; el número de estados del conmutador PINIUM está en el orden de $O(\log_2 N)$. En consecuencia, el conmutador PINIUM tiene una latencia mucho más pequeña que aquellos clasificadores de redes de Batcher.

Para calcular el número de conmutadores elementales se asume que cada conmutador elemental multicast tiene la misma complejidad como el de 2×2 en una sección de concentración. Habrán $(N-r)/(r-1)$ elementos multicast de base r en un arreglo de planos en árbol. En una concentración de planos, se tendrán $S_p(L)$ conmutadores elementales en un mezclador concentrador priorizado de $2L$ a L , el cual puede ser expresado así:

$$S_p(L) = \frac{L(\log_2 L + 2)}{2}, \quad (76)$$

Y cada L clasificadores tiene $(1/4)\log_2 L(\log_2 L + 1)$ conmutadores elementales. Si $S_e(N, L)$ es el número total de conmutadores elementales en un conmutador PINIUM con N entradas y un parámetro L del Knockout, se tendrá:

$$S_e(N, L) = N \left[\frac{N-1}{r-1} + N \frac{(\log_2 L)^2 + 3\log_2 L + 4}{4} - \frac{L(\log_2 L + 2)}{2} \right], \quad (77)$$

A partir de esta expresión, se puede afirmar que el número de conmutadores elementales está en el orden de $O((N\log_2 L)^2)$, en el caso de un conmutador Knockout, si se obvian los N^2 filtros de celdas y elementos retardados, habrán sólo $O(N^2 L)$ conmutadores elementales. Por tanto, el conmutador PINIUM incurre sólo en una pequeña latencia para todas las celdas ATM que llegan. Además, provee otras características tales como la función de clasificación priorizada y servicios multicast. Pero, existe una desventaja en cuanto al número de conmutadores elementales, pero debido a la existencia de la tecnología VLSI, el peso de dicho factor puede reducirse. Entonces, con la descomposición de N a L concentradores clasificadores priorizados, este conmutador presenta la ventaja de menor número de estados, menor número de conmutadores elementales y una muy fácil expansión.

El desempeño del conmutador PINIUM en la condición de tráfico aleatorio es esquematizado de acuerdo al modelo matemático (67) cuando $\rho_{in}=1$. Al analizar el desempeño del conmutador cuando N tiende a infinito, el tráfico entrante se expande después de dejar los servicios multicast, y en la tabla No. 4.1 se tabula la carga efectiva entregada a un puerto de salida cuando una celda de entrada es destinada para diferentes direcciones de C_{prom} dados para algún valor de ρ_{in} . Bajo la influencia de esta expansión de tráfico, se esquematiza la figura No. 4.11 para $\rho_{in}=1$.

ρ_{out}	$C_{prom}=1$	2	3	4	5	6	7	8	9	10
$\rho_{in}=1$	1,00	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0
0,90	0,90	1,8	2,7	3,6	4,5	5,4	6,3	7,2	8,1	9,0
0,80	0,80	1,6	2,4	3,2	4,0	4,8	5,6	6,4	7,2	8,0
0,70	0,70	1,4	2,1	2,8	3,5	4,2	4,9	5,6	6,3	7,0
0,60	0,60	1,2	1,8	2,4	3,0	3,6	4,2	4,8	5,4	6,0
0,50	0,50	1,0	1,5	2,0	2,5	3,0	3,5	4,0	4,5	5,0

Tabla No. 4.1: *Carga entregada al Puerto de salida con la celda destinada para C_{prom} destinos.*

La figura 4.11 mostrada presenta la gráfica de la probabilidad de pérdidas de celdas versus el valor del parámetro L del Knockout para diferentes valores de C_{prom} . Para obtener bajas probabilidades de pérdidas de celdas tales como 10^{-6} o 10^{-10} , se debe hacer una selección apropiada del tamaño de salidas de los concentradores clasificadores priorizados. En la siguiente tabla, como ejemplo, se muestran los valores mínimos de L para tener un promedio de probabilidad de pérdidas de celdas de 10^{-6} .

Se puede observar que si una carga efectiva en la salida es ofrecida será consistentemente mayor que uno, entonces la disciplina de cola de espera en la salida, en comparación con el principio Knockout, es una excelente aproximación para manipular con facilidad el tráfico de las celdas entrantes. Por ejemplo, en la figura No. 4.11 con $\rho_{in}=1$ y $L=21$, el conmutador puede sustentar una probabilidad de pérdida de celdas de 10^{-10} inclusive si se tiene una expansión de tráfico

$C_{prom}=4$. La gráfica indica la superioridad del conmutador PINIUM en manipular movimientos muy fuertes del tráfico multicast entrante. Esta observación es importante debido a que el repentino y fuerte movimiento del tráfico multicast puede obviamente deteriorar el desempeño de otros tipos de diseños de conmutadores. En la tabla No. 4.3 se desea hallar la carga máxima aceptable ofrecida a un puerto de salida y al cual lo llamamos $\rho_{out,max}$ en el conmutador PINIUM, para algunos valores de L y algún criterio de probabilidad de pérdida de celdas. El valor de $\rho_{out,max}$ es calculado usando el método de bisección con una tolerancia ϵ , el cual se define de la siguiente manera:

$$\epsilon \geq \left| \frac{P_{p\acute{e}rdida(\rho_{out,max})} - P_{p\acute{e}rdida,req}}{P_{p\acute{e}rdida,req}} \right| \quad (78)$$

L	$C_{prom}=1$	2	3	4	5	6	7	8	9	10
$\rho_{in}=1$	8	11	13	15	17	19	20	22	24	26
0,90	8	10	12	14	16	18	20	21	22	24
0,80	8*	10	12	14	15	16	18	20	21	23
0,70	8*	9	11	13	14	15	17	18	19	21
0,60	8*	8	10	12	13	15	16	17	18	19
0,50	8*	8	10	11	12	13	14	15	16	17

Tabla 4.2: Valores del parámetro L del knockout en concordancia del número de copias requeridas por cada celda de entrada para que se tenga una probabilidad de pérdida de celdas de 10^{-6} , cuando N tiende a infinito y *L=8 es tomado como el tamaño mínimo.

Donde $P_{p\acute{e}rdida(\rho_{out,max})}$ es la probabilidad de pérdida de celdas debido a la carga máxima aceptable en el puerto de salida y $P_{p\acute{e}rdida,req}$ es la probabilidad de pérdida de celdas deseada. Por tanto, si $L=16$, a partir de la tabla 4.3; la presencia transitoria de la carga en el puerto de salida puede ser tan alto como 2 con la probabilidad de pérdida de celdas aproximada de 10^{-10}

Se ha obviado el impacto de la expansión del tráfico en los buffers de control en los puertos de salida. Para lograr la condición de estado estático y acomodar el tráfico multicast, se debe mantener el promedio de la carga en la entrada de que no exceda $1/C_{prom}$. En este caso, la carga en el puerto de salida puede ser mantenida por debajo de uno, y diferentes valores de L pueden ser escogidos para combatir el problema de presencia transitoria excesiva de tráfico. Similarmente, el efecto de ráfagas de flujos de tráfico de entrada no tendrán esencialmente un impacto sobre el conmutador debido al diseño del buffer interno de capacidad infinita. Sin embargo, tiene un efecto substancial en el tamaño del buffer en cada controlador del puerto de salida y se debe entonces, calcular el tamaño del buffer muy cuidadosamente.

En [2] se considera el tamaño del buffer de 1024 celdas para simular el caso de un buffer de tamaño infinito que espera en el controlador del puerto de salida. Además, asume $N=64$, $L=8$, y que el tráfico aleatorio de entrada de 3 diferentes clases de prioridad sigue la distribución geométrica truncada con un parámetro h . Sea μ_i , la probabilidad de que una celda que entra tiene la prioridad i , donde $i=1,2,3$. Por lo que en [2] se define de la siguiente forma:

$$\mu_i = \frac{(1-h)h^{3-i}}{1-h^3}, \quad \text{para: } i \in (1,2,3). \quad (79)$$

Considerando una ambiente de tráfico unicast; la carga de entrada tiene que ser menor que uno para lograr un sistema estable.

$\rho_{out,max}$	$P_{pérdida,req} = 10^{-6}$	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-13}
$L=8$	0,968750	0,709375	0,521875	0,385937	0,286816	0,213672	0,159326	0,118874
16	4,342188	3,615625	3,031250	2,559375	2,168750	1,846875	1,575000	1,346924
32	13,834375	12,329688	11,071875	9,989062	9,042969	8,217188	7,484375	6,839062

Tabla No. 4.3: Carga efectiva máxima en la salida para diferentes probabilidades de pérdida y diferentes valores de L con $\varepsilon = 0,01$.

** Todas las ecuaciones desde la (61) hasta la (79) se han tomado de [2].

RESUMEN

En el presente capítulo se ha hecho un análisis detallado, determinándose los límites máximo y mínimo de la probabilidad de pérdidas de celdas (pérdidas de paquetes), cuya consideración es importante para que la arquitectura de Knockout Generalizado tenga la capacidad de crecimiento. Los resultados de dicho análisis indican que las pérdidas por Knockout y debido al criterio subóptimo pueden ser muy pequeños.

La mínima diferencia existente entre los límites máximo y mínimo permiten mejorar mediante el uso del algoritmo definido como óptimo (pero lento), por lo que, uno o dos puertos adicionales de entrada deben añadirse a cada módulo conmutador de paquetes, o rutas internas deben ser convertidas en rutas principales para así compensar la consideración del criterio subóptimo. Según el gráfico de la figura No. 4.9, para garantizar una probabilidad de pérdida de celdas de 10^{-9} , la arquitectura puede requerir de módulos de conmutación de paquetes de dimensiones de 47 x 16, 45 x 16, 42 x 16 y 39 x 16 para cargas de tráfico de 100, 90, 80 y 70% respectivamente.

CAPÍTULO V

CONCLUSIONES

- 1.- En el presente trabajo se han expuesto las principales arquitecturas de conmutadores ATM de grandes dimensiones que he considerado importantes y cuyos conceptos están basados en el sistema Knockout. Existen arquitecturas propuestas en forma combinada entre el Knockout y el sistema Batcher entre otras, pero para el presente estudio sólo se ha considerado los que están basados en el sistema Knockout.
- 2.- Para el desarrollo de esta tesis se han considerado conceptos previamente desarrollados y encontrados en la bibliografía, así como el análisis matemático respectivo del tema. Así, se tiene que el conmutador ATM es un conmutador multiplexado en el tiempo y cuyo sistema de tráfico está determinado por el criterio estrictamente probabilístico. En lo posible se han definido criterios y algoritmos considerando una condición óptima la cual ha sido analizada según un modelo probabilística y no un análisis aleatorio.
- 3.- La arquitectura del sistema Knockout generalizado presenta características de tener mejor desempeño en retardo y throughput, garantiza la secuencia FIFO de paquetes, capacidad de broadcast y multicast, sobre todo su compatibilidad con longitud variable de los paquetes lo que evita la necesidad de normalizar el tamaño de éstos. Por ejemplo, según consideraciones y recomendaciones dadas por los investigadores especializados en el tema [1][2][3], para un $N > 1000$, un conmutador de paquetes de 4096×4096 (con línea de entrada de 150 Mbps) se puede construir usando módulos de conmutación de paquetes de 42×16 y módulos de interconexión de 256×256 ; siendo ambos, factibles de ser implementados físicamente. Teóricamente, para $N = 1000$ entradas y que cada uno opere a 150 Mbits/s, entonces se pueden tener conmutadores del orden de 150 Gbits/s que incluyan interconexiones a multiprocesadores y MAN's y WAN's de muy alta velocidad.

- 4.- En las arquitecturas plegables, bilateral y tres estados, el tamaño de la fábrica puede crecer lentamente hasta una dimensión bastante grande. La arquitectura permite ajustar el throughput dependiendo del criterio que se defina en el diseño inicial. Las soluciones exactas de los puntos de ramificación y su influencia en el costo total requiere un mayor análisis y estudio. La conclusión más importante para este caso es que la nueva fábrica no requerirá del incremento del número de estados de conmutación para su crecimiento o escalamiento, y el cual siempre será igual al número de estados fijados en la fábrica básica o elemental.
- 5.- La modularidad de la arquitectura PINIUM, implica menor exigencia en sincronización y hace posible su implementación para muy altas velocidades. En [2] se estima que puede construirse un conmutador con una capacidad del orden de los terabits empleando la tecnología VLSI.
- 6.- En los gráficos 2.18 y 3.3 se ve que para la misma probabilidad de pérdida de paquetes, el número de paquetes aceptados es bastante variado. Por ejemplo para una carga del 70%, para la probabilidad de 10^{-8} se puede ver que en la figura 2.18 el número de paquetes aceptados (tamaño del concentrador) es del orden de 10 (Knockout simple) y en la figura 3.3 para la misma probabilidad el número de paquetes aceptados es del orden de 32 (Knockout generalizado).
- 7.- En el caso de la arquitectura PINIUM, para un criterio algorítmico diferente, pero basado en el Knockout Generalizado, para la probabilidad mínima de pérdidas típica del orden de 10^{-6} a 10^{-10} el número de paquetes está entre 18 y 21, para C_{prom} de 4, tal como puede observarse en la figura 4.11. Los resultados numéricos en detalle pueden observarse en las tablas 4.1, 4.2 y 4.3.
- 8.- El tamaño del sistema interconectado de conmutadores elementales depende de m/n (figura 3.5), siendo para todos los casos graficados $n=16$ y m variable según lo indicado en los acápites (6) y (7); pero en la figura 3.16 se muestra un esquema para el caso de la arquitectura plegable y diferentes

valores de m , que permiten evaluar el tamaño modular que puede tener un conmutador ATM en dicha arquitectura.

- 9.- La arquitectura del Knockout Generalizado es atractivo en cuanto a su escalabilidad y capacidades broadcast y multicast; y sobre todo su adaptabilidad a los paquetes de longitud variable. La arquitectura PINIUM presenta un mejor comportamiento en sistemas broadcast y multicast. Las arquitecturas Plegable y de Memoria Compartida son más adecuados para conmutadores ATM de menor tamaño y mínima cantidad de LAN's.
- 10.- En la figura 4.7 se presenta la variación del tráfico (cantidad de paquetes aceptados) para una misma probabilidad de pérdidas debido sólo al *Knockout + Scheduling* y para diferentes valores de carga de tráfico. En la figura 4.10 se puede observar el comportamiento comparativo de las curvas, para la probabilidad de pérdidas de paquetes debidos al *Knockout* y el *Knockout+Scheduling* en función del tráfico y para dos cargas de tráfico, notándose que la cantidad de paquetes aceptados (tráfico) varía en la práctica aproximadamente en dos.
- 11.- Entre las consideraciones más importantes, como resultado de las últimas investigaciones (1995 – 2003) para el diseño y análisis de un conmutador multicast de gran escala, considero que deben emplearse:
 - a) El principio Knockout multicast, ampliado al principio del Knockout generalizado para incorporar la capacidad multicast, ésto, debido a su menor complejidad que el PINIUM,
 - b) Buffers de cola de espera en la salida con capacidad de duplicar la celda para lograr el mejor desempeño en retardo y throughput,
 - c) Procesamiento distribuido para la réplica de celdas, filtros de celdas y contención de celdas, y
 - d) Estados múltiples para reducir la necesidad adicional de direcciones de enrutamiento.
- 12.- Tal como expresa Martin Collier [17], El tema sigue siendo de investigación y un reto constante no solo en el desarrollo tecnológico sino también, en lo

que respecta a la implementación práctica y el circuito de enrutamiento y control.

BIBLIOGRAFÍA

- [1] Kai Y. Eng, Mark J. Karol and Yu-Shuan Yeh; "A Growable Packet (ATM) Switch Architecture: Design Principles and Applications". *IEEE Transactions on Communications*. Vol. 5, No. 8. Febrero de 1992.
- [2] Ka Lun Eddie Law, Alberto Leon-Garcia. "A Large Scalable ATM Multicast Switch". *IEEE Journal of Selected Areas in Communications*. Vol. 15. No. 5. Junio de 1997.
- [3] Andrzej Jajszczyk and Wojciech Kabacinski. "A Growable (ATM) Switching Fabric Architecture". *IEEE Transactions on Communications*". Vol. 43. No. 2. Abril de 1995.
- [4] Mark J. Karol and Chih-Lin I. "Performance Analysis of a Growable Architecture for Broad-Band Packet (ATM) Switching". *IEEE Transactions on Communications*. Vol. 40. No. 2. Febrero de 1992.
- [5] H. Jonathan Chao and Byeong-Seog Choe. "Design and Analysis of a Large-Scale Multicast Output Buffered ATM Switch". *IEEE/ACM Transactions on Networking*. Vol. 3. No. 2. Abril de 1995.
- [6] Martin de Prycker. "Asynchronous Transfer Mode. Solution for Broadband ISDN". *Alcatel Bell. Antwerp Belgium. Edit. Ellis Horwood*. 1993.
- [7] Knochen Wang and Ming-Howe Cheng. "Design and Analysis of a Growable Multicast ATM Switch". *IEEE Transactions on Communications*. Vol. 48. No. 7. Julio del 2 000.
- [8] Soung Y. Liew and Tony T. Lee. "Bandwidth Assignment with QoS Guarantee in a Class of Scalable ATM Switches". *IEEE Transactions on Communications*. Vol. 48. No. 3. Marzo del 2 000.
- [9] Yu Shuan Yeh, Michael O. Hluchyj and Anthony S. Acampora. "The Knockout Switch: A Simple, Modular Architecture for High Performance Packet Switching". *IEEE Journal on Selected Areas in Communications*. Vol. 5. No. 8. Octubre de 1987.
- [10] D. X. Chen and J. W. Mark. "SCOQ: A Fast Packet Switch with Shared Concentration and Output Queueing". *IEEE/ACM Transactions on Networking*. Vol. 1. Febrero de 1993.
- [11] Y. Kato et al. "A VLSIC for the ATM switching system". In Proc. XIII Int. Switching Symp., Stockholm, Sweden. Vol. III, pp. 27 - 32. June 1990.

- [12] Alexander Gersht and Alexander Shulman. "Architecture for Restorable Call Allocation and Fast VP Restoration in Mesh ATM Networks". *IEEE Transactions on Communications*. Vol. 47. No. 3. Marzo de 1999.
- [13] H. S. Kim and A. Leon-Garcia. "Non-blocking Property of Reverse Banyan Network". *IEEE Transactions on Communications*. Vol. 40. Marzo de 1992.
- [14] J. D. C. Little. "A Proof for the Queueing Formula: $L=\lambda W$ ". *Oper. Res.* Vol. 9. 1961.
- [15] Mark J. Karol, Hluchyj M. G. and Morgan S. P. "Input versus Output Queueing in a Space-Division Packet Switch". *IEEE Transactions on Communications*. Vol. 35. No. 12. Diciembre de 1987.
- [16] Kleinrock L. "Queueing Systems". *John Wiley & Sons Publication*. New York. 1975.
- [17] Martín Collier. "A Three-Stage ATM Switch with Cell-Level Path Allocation". *IEEE Transactions on Communications*. Vol. 45. No. 6. Junio de 1997
- [18] Ronald P. Bianchini and Hyong S. Kim. "The Tera Project: A Hybrid Queueing ATN Switch Architecture for LAN". *IEEE Journal on Selected Areas in Communications*. Vol. 13. No. 4. Mayo de 1995.
- [19] K. Batcher. "Sorting Networks and their Applications". In Proc. AFIPS Spring Joint Comput. Conf. Vol. 32. 1968.
- [20] A. Jajszczyk. "A dynamic Programming Approach to Optimization of Switching Networks Composed of Digital Switching Matrices". *IEEE Transactions on Communications*. Vol. COM-35. Diciembre de 1987.
- [21] T. Sudo. "Present and Future Direction for Multichip Module Technologies". *IEEE Solid-State Circuits*. Vol. 30. Abril de 1995.
- [22] N. H. E. Weste and K. Eshraghian. "Principles of CMOS VLSI Design. A Systems Perspective. 2nd. Edition. Reading, MA: Addison-Wesley. 1993.
- [23] K. Y. Eng and A. S. Acampora. "Fundamental Conditions Governing TDM Switching Assignments in Terrestrial and Satellite Networks". *IEEE Transactions on Communications*. Vol. COM-35. Julio de 1987.
- [24] C. Rose and M. G. Hluchyj. "The Performance of Random and Optimal Schedule in a Time-multiplex Switch". *IEEE Transactions on Communications*. Vol. COM-35. Agosto de 1987.
- [25] K. L. E. Law. "Large Scale ATM Switching Systems". *Phd. Disertation, Dep. Elec. Comput. Eng. Univ. Toronto. Toronto, Ont. Canada. 1995.*

- [26] **Ka Lun Eddie Law, Alberto Leon-Garcia.** "Multicast and self-routing in ATM radix trees and banyan networks". *In Proc. IEEE INFOCOM'95*, pp. 951 - 959.
- [27] **Henrion M., Schrodi K., Boettle D., De Somer M., Dieudonne M.** "Switching network architecture for ATM based broadband communications". *ISS'90. Stockolm, June 1990.*
- [28] **Lillykutty Jacob and Anurag Kumar.** "Delay performance of some Scheduling Strategies in an Input Queuing ATM Switch with Multiclass Bursty Traffic". *IEEE/ACM Transactions on networking. Vol. 4, No. 2, Abril 1996.*
- [29] **Coudreuse J. P., Serval M.** "Prelude: an asynchronous time division switched network". *ICC'87. Seattle. Junio 1987.*
- [30] **K. Y. Eng, M. G. Hiuchyj.** "A Knockout switch for variable-length packets". *IEEE Journal on Selected Areas in Communications. Vol. SAC-5*, pp. 1426-1435. *Diciembre 1987.*
- [31] **V. E. Benes.** "On rearrangeable three-stage connecting networks". *Bell System Technology Journal*, pp. 1481-1492. *Septiembre 1962.*
- [32] **De Prycker M., De Somer M.** "Performance of a service independent switching network with distributed control". *IEEE Journal on Selected Areas in Communications. Vol. 5, No. 8. October 1987.*
- [33] **M. de Prycker.** "Asynchronous Transfer Mode: Solution for Broadband ISDN". *Chickester, England: Ellis Horwood, 1991.*
- [34] **M. de Prycker et al.** "An ATM switching architecture with intrinsic multicast capabilities for the Belgian broadband experiment". *In Proc. XIII Int. Switching Symp., Stokholm, Sweden. Vol. V, pp. 111 - 118. Mayo/Junio 1990.*
- [35] **M. A. Henrion et al.** "Switching network architecture with intrinsic multicast capabilities communications". *In Proc. XIII Int. Switching Symp., Stokholm, Sweden. Vol. V, pp. 1 - 8. June 1990.*
- [36] **T. R. Banniza, G. J. Eilenberger, B. Pauwels, Y. Therasse.** "Design an technology aspects of VLSI's for ATM switches". *IEEE Journal on Selected Areas in Communications. Vol. 9, pp. 1255 - 1264. October 1991.*
- [37] **A. Jajszczyk.** "On nonblocking switch networks composed of digital symmetrical matrices". *IEEE Transactions on Communications. Vol. COM-31. Enero de 1983.*
- [38] **A. Jajszczyk.** "Novel architecture for a digital switching network". *Electron. Lett., Vol. 20, pp. 683. Agosto de 1984.*

- [39] **Onvural, Raif O.** "Asynchronous transfer mode networks: performance issues", *Chapter 7. Boston: Artech House, 1994.*
- [40] **Y. Shobatake et.al.** "A one chip scalable 8x8 ATM switch LSI employing shared buffer architecture". *IEEE Journal on Selected Areas in Communications. Vol. 9, pp. 1280-1288. Octubre de 1991.*
- [41] **Y. Sakurai, N. Ido, S. Gohara and N. Endo.** "Large scale ATM multistage switching network with shared buffer memory switches". *In Proc. XIII Int. Awitching Symp. Stockholm Sweeden. Mayo/junio de 1990. Vol. IV, pp. 121 – 126.*
- [42] **W. Fischer, O. Fundneider, E. H. Goeldner and K. A. Lutz.** "A scalable ATM switching system architecture". *IEEE Journal on Selected Areas in Communications. Vol. 9, pp. 1299 - 1307. Octubre de 1991.*
- [43] **T. Kozaki et. al.** "32x32 shared buffer type ATM switch VLSI's for B-ISDN's". *IEEE Journal on Selected Areas in Communications. Vol. 9, pp. 1239 - 1247. Octubre de 1991.*
- [44] **A. Itoh et. al.** "Practical implementation and packaging technologies for large-scale ATM switching system". *IEEE Journal on Selected Areas in Communications. Vol. 9, pp. 1280 - 1288. Octubre de 1991.*
- [45] **T. C. Banwell et. al.** "Physical design issues for very large ATM switching systems". *IEEE Journal on Selected Areas in Communications. Vol. 9, pp. 1227 - 1238. Octubre de 1991.*
- [46] **Joseph Y. Hui and Edward Arthurs.** "A broadband packet switch for integrated transport". *IEEE Journal on Selected Areas in comunicatios. Vol. SAC-5. No. 8. pp. 1264 – 1273. Octubre 1987.*
- [47] **Rajgopal Kannan, Radim Bartos, K. Y. Lee and Harry F. Jordan.** "Sxmin: A Self-Routing High-Performance ATM Packet Switch Based on Group-Knockout Principle". *IEEE Transactions on Communications. Vol. 45. No. 6. Junio de 1997.*
- [48] **Tony T. Lee y Cheuk H. Lam:** "Path Switching – A Quasi-Static Routing Scheme for Large-Scale ATM Packet Switches". *IEEE Journal on Selected Areas in comunicatios. Vol. 15. No. 5. Junio de 1997.*
- [49] **Soung Y. Liew y Tony T. Lee:** "Bandwidth Assignment with QoS Guarantee in a Class of Scalable ATM Switches". *IEEE Transactions on Communications. Vol. 48. No. 3. Marzo del 2000.*
- [50] **Kuo Chen Wang y Ming-Howe Cheng:** "Design Analysis of a Growable Multicast ATM Switch". *IEEE Transactions on Communications. Vol. 48. No. 7. Julio del 2000.*