



Universidad  
Politécnica  
de Cartagena

CONMUTACIÓN

PRÁCTICA 3

# CONMUTACIÓN ESPACIAL Y TEMPORAL

**Autores (en orden alfabético):**

Josemaría Malgosa Sanahuja

Pilar Manzanares López

Juan Pedro Muñoz Gea

## OBJETIVOS

- Entender qué es un conmutador espacial multietapa y qué ventajas e inconvenientes representa respecto al conmutador de matriz cuadrada (*crossbar*) de una sola etapa.
- Describir el modo de operación de los distintos algoritmos de selección de la matriz de la etapa intermedia.
- Comprender y aplicar la condición de Clos.
- Entender el funcionamiento de los conmutadores digitales basados en la multiplexación por división en el tiempo.
- Comprender el funcionamiento de las etapas de conmutación temporales (etapas T).
- Conocer los distintos sistemas de señalización.
- Comprender las funcionalidades de sincronismo y señalización.

**DURACIÓN DE LA PRÁCTICA:** 3 sesiones (6 horas)

**MATERIAL NECESARIO:** Simuladores MATSWIT.EXE y TIMSWIT.EXE (disponibles en el Aula Virtual)

## 1. CONMUTACIÓN ESPACIAL

El conmutador de matriz cuadrada o *crossbar* es la solución más simple al problema de interconectar  $N$  entradas con  $N$  salidas. Esta solución presenta accesibilidad total, ya que desde cualquier entrada se puede seleccionar cualquier salida, y no presenta bloqueo interno, ya que cada par entrada-salida dispone de un punto de cruce (ver Figura 1). Por otra parte, el número de puntos de cruce en un conmutador *crossbar* de  $N$  entradas y  $N$  salidas es de  $N \times N$ , es decir, el número de puntos de cruce ( $M$ ) crece en proporción al cuadrado del número de entradas (p.ej.  $N=1000$ ,  $M=10^6$  puntos de cruce), alcanzando para valores elevados de  $N$  soluciones demasiado costosas o no implementables.

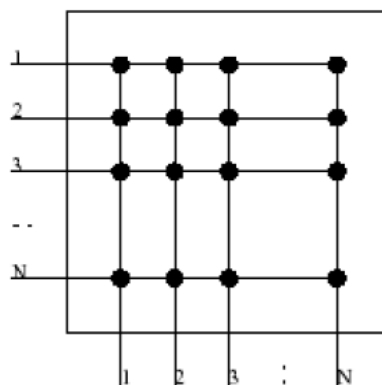


Figura 1. Conmutador de matriz cuadrada o *crossbar*

Aparece entonces la necesidad de nuevas arquitecturas que permitan mantener las características del *crossbar* disminuyendo el número de puntos de cruce. Un primer intento sería

usar 2 conmutadores de 1000 entradas y 100 salidas, disminuyendo el número de puntos de cruce a  $2 \times 10^5$  (ver Figura 2). El problema de esta solución es que, a pesar de presentar accesibilidad total, presenta bloqueo interno en sentido estricto (imposible establecer 101 enlaces simultáneos).

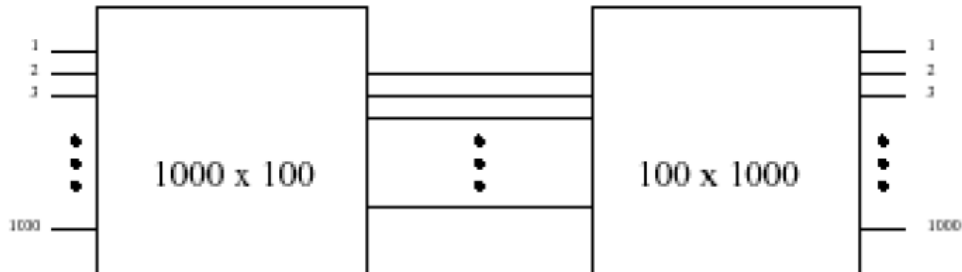


Figura 2. Dos conmutadores *crossbar* 100x100 y 100x1000 implementando un conmutador 1000x1000

Otro intento sería desdoblar el *crossbar* en varios, p.ej., 10 unidades de pares de *crossbars* de 100x10 (ver Figura 3). Ahora el número de puntos de cruce ha disminuido a  $2 \times (100 \times 10) \times 10 = 2 \times 10^4$ , pero aparece un nuevo problema: además de bloqueo interno, se ha perdido la accesibilidad total, ya que no existe interconexión posible entre distintas unidades.

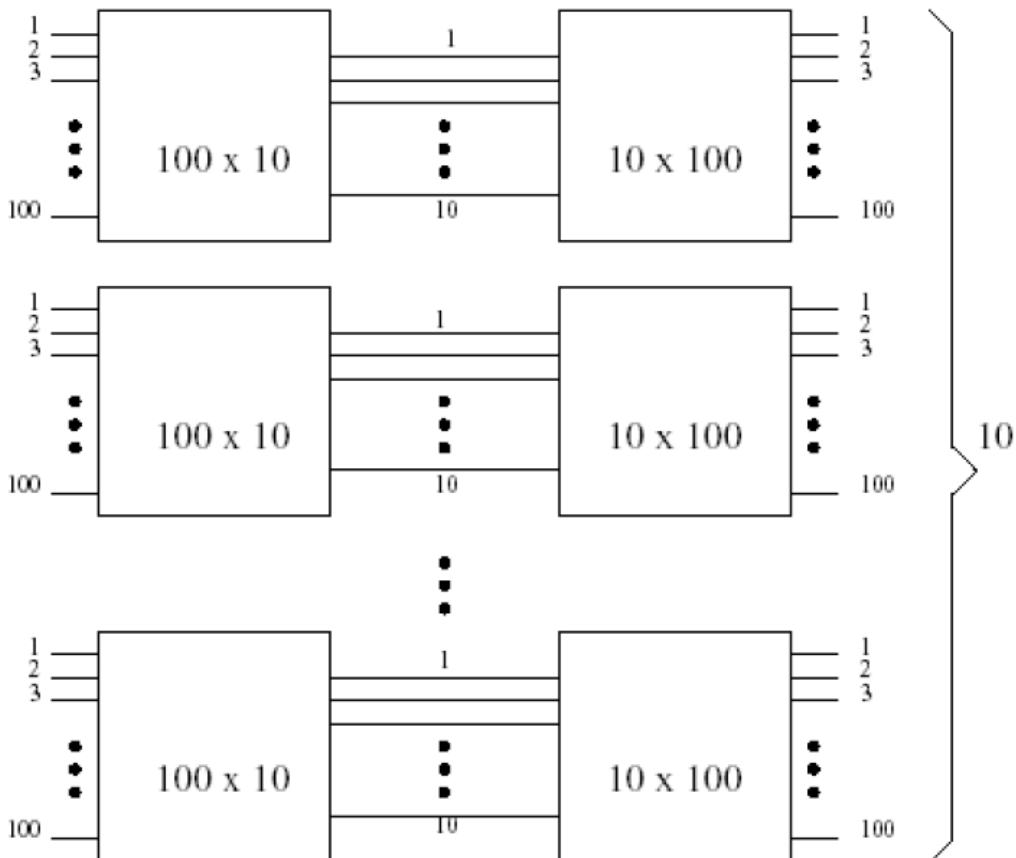


Figura 3. Etapas de 100x10 implementando un conmutador 1000x1000

Finalmente, la solución a utilizar es la red multietapa (ver Figura 4), que presenta accesibilidad completa. Para comparar el conmutador multietapa con el *crossbar* hay que estudiar el número de puntos de cruce que presenta. Habiendo un total de  $N/n$  matrices de tamaño  $n \times k$  en las etapas de entrada y de salida, y siendo  $K$  el número de matrices en la etapa intermedia de tamaño  $(N/n) \times (N/n)$ , el número de puntos de cruce viene dado por  $M=2 \times (N/n) \times n \times k + k \times (N/n)^2$ .

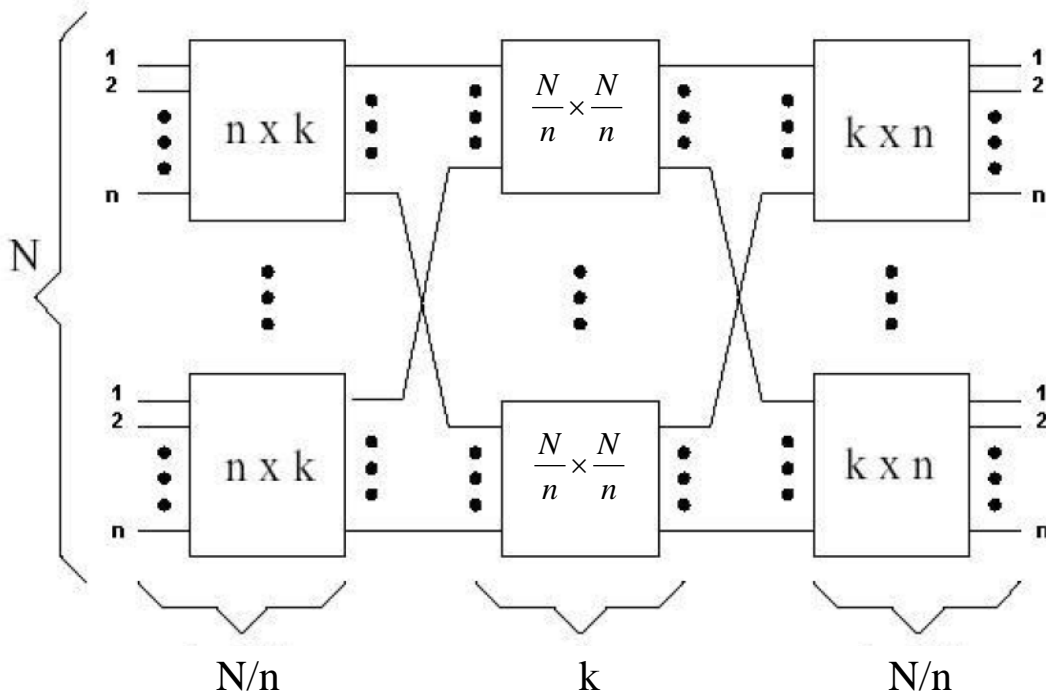


Figura 4. Conmutador multietapa

Para que una red multietapa no presente bloqueo interno en sentido estricto debe de cumplir la condición de Clos. Vamos a imaginar una situación en la que en la matriz 1 de la etapa de entrada de la red, de tamaño  $n \times k$ , se han establecido  $n-1$  conexiones hacia distintas matrices de salida diferentes de la matriz de salida 1. Análogamente, la matriz 1 de la etapa de salida de la red, de tamaño  $k \times n$ , tiene  $n-1$  conexiones establecidas procedentes de distintas matrices de entrada distintas de la matriz de entrada 1. En el peor caso, se estarán ocupando  $2 \times (n-1)$  matrices de la etapa intermedia (etapa de distribución). Si se desea iniciar una conexión desde la matriz de entrada 1 hacia la matriz de salida 1, para que no haya posibilidad de bloqueo, el número de etapas de distribución debería ser mayor o igual a  $2 \times (n-1) + 1 = 2n-1$ .

## 2. ALGORITMOS DE SELECCIÓN DE CAMINOS

Se utilizará el simulador MATSWIT.EXE, opción A (*Blocking in a multi-stage switch*).

2.1. Para la red de conexión que se presenta responder a las siguientes preguntas: ¿Tiene accesibilidad completa?, ¿Por qué?, ¿Cuántos puntos de cruce requiere y qué porcentaje de ahorro presenta respecto a una matriz cuadrada  $16 \times 16$ .

2.2. Mediante una ejecución paso a paso (opción F1-*Step*) describir el modo de operación de los distintos algoritmos de selección de la matriz de la etapa intermedia (opción F5-*New algorithm*). Etiquetar cada uno de dichos algoritmos mediante una de las siguientes palabras: aleatorio, secuencial o rotante.

2.3. Seleccionar uno de los algoritmos, por ejemplo el a, y explicar mediante un ejemplo por qué se producen situaciones de bloqueo interno. Para cada algoritmo realizar 10 simulaciones rápidas (opción F3-*Fast run*) y observar en cuántas se establecen 16 conexiones. Compara los resultados obtenidos. ¿Qué algoritmo prevé que es mejor desde el punto de vista del bloqueo interno?

Seleccionar la opción B (*Switch performance statistics*) del simulador MATSWIT.EXE.

2.4. Para cada algoritmo realizar tres simulaciones rápidas (opción F3-*Run*) de 1000 secuencias, en cada una de las cuales se intentarán establecer 16 llamadas. Obtener en cada simulación el número medio de conexiones realizadas y el porcentaje de secuencias con éxito. Hallar los valores medios de dichos parámetros y comparar el comportamiento de los diferentes algoritmos.

2.5. ¿Cuál es el número mínimo de conexiones que puede realizar cada algoritmo sin tener bloqueo interno?

### 3. COMPORTAMIENTO EN RÉGIMEN CONTÍNUO

Se utilizará el simulador MATSWIT.EXE, opción C (*Continuous switch operation*).

3.1. Mediante una ejecución paso a paso (opción F1-*Step*) explicar cómo se representan las conexiones en el diagrama mostrado.

3.2. Mediante una ejecución (opción F2-*Run*) obtener una situación de bloqueo. A la vista de las matrices de distribución conectadas a las correspondientes matrices de entrada y de salida de la conexión bloqueada, ¿cómo se refleja dicha situación de bloqueo en el diagrama de conexiones?

3.3. Mediante una ejecución rápida (opción F3-*Fast run*) de 100000 eventos obtener el tráfico ofrecido a partir del tráfico cursado (TC) y el tráfico rechazado (TR), y establecer las relaciones entre ellos a partir del número de conexiones cursadas (#C) y del número de conexiones rechazadas (#R).

#### 4. REDES SIN BLOQUEO

Se utilizará el programa MATSWIT.EXE, opción C (*Continuous switch operation*). En primer lugar es necesario configurar un tráfico ofrecido de 15 Erlangs, utilizando la opción F5-*Reset data*.

4.1. ¿Cuál es el número mínimo de matrices de distribución para no tener bloqueo interno en sentido estricto? ¿Depende dicho resultado del algoritmo seleccionado? Comprobar ambas respuestas mediante simulación.

4.2. ¿Cuántos puntos de cruce se requieren para garantizar la condición de no bloqueo en sentido estricto? Comparar este valor con el número de puntos de cruce que precisa una *crossbar* 16x16. A la vista del resultado anterior, ¿en qué condiciones es ventajoso diseñar una red de Clos?

#### 5. CONMUTACIÓN TEMPORAL

El hecho de disponer de un canal con un cierto ancho de banda y una serie de comunicaciones a llevar a cabo obliga a la repartición del recurso canal. Existen diferentes técnicas para la repartición de dicho recurso, entre ellas la multiplexación por división en el tiempo (TDM, *Time Division Multiplexing*). Ésta consiste en dividir el tiempo en distintas ranuras (*slots*) de tiempo, que pueden ser utilizadas por las distintas comunicaciones. Si además se habla de comunicación síncrona, dichas ranuras estarán confinadas dentro de intervalos regulares de tiempo en una trama que se repite periódicamente.

En este entorno aparece la conmutación temporal (etapas de conmutación T), consistente en una reordenación temporal de las ranuras. Así, las tramas (agrupaciones de ranuras) a la salida de un conmutador temporal aparecen con sus ranuras reordenadas en el tiempo, quedando la conmutación realizada.

Finalmente, para que un conmutador síncrono funcione correctamente, éste debe saber en cada momento hacia dónde conmutar una ranura. Es decir, aparecen las necesidades de sincronización y de señalización, conceptos que se trasladan a ranuras temporales.

#### 6. PRINCIPIOS DE LA MULTIPLEXACIÓN POR DIVISIÓN EN EL TIEMPO (TDM)

Se utilizará el simulador TIMSWIT.EXE, opción A (*Time-division multiplexing (tdm)*).

6.1. Dibujar y describir los diferentes módulos del esquema TDM que se muestra en pantalla (*source, outlet buffer, link, inlet buffer, receiver*). ¿Por qué se realizan conversiones paralelo-serie y serie-paralelo en la fuente y en el receptor, respectivamente?

6.2. Mediante una ejecución ciclo a ciclo (opción F1-*Cycle*), slot a slot (opción F2-*Timeslot*) o trama a trama (opción F3-*Frame*), según convenga, responder a las siguientes cuestiones:

- a) ¿Cuántos ciclos hay en cada slot y qué operaciones se hacen en cada ciclo? ¿Por qué es necesario el ciclo de espera *wait*?
- b) ¿Cuántos slots hay en cada trama? ¿Por qué?
- c) ¿Cuál es el desfase en slots entre fuente y receptor? ¿A qué se debe dicho desfase?
- d) ¿Cuántas tramas se requieren para enviar una palabra de las usadas en el simulador entre fuente y receptor? ¿Por qué?

## 7. ETAPA DE CONMUTACIÓN T

Se utilizará el simulador TIMSWIT.EXE, opción B (*Time switching with tdm*), y se seleccionará la opción F3-*Time switch demonstration*.

7.1. Dibujar la etapa T y describir todos los elementos que la integran (*source --Ts, outlet buffer--*, *receiver --Ts, inlet buffer--*, *switch --inlet buffer, outlet buffer, Fm, Ts, Cy--*), indicando la función que realizan, así como el contenido de la memoria de control. ¿Qué relación existe entre el origen y el destino de una conexión y la posición y el contenido de la memoria de control? ¿Cuál es el equivalente analógico de esta etapa T?

7.2. Mediante una ejecución ciclo a ciclo (opción F1-*Cycle*), slot a slot (opción F2-*Timeslot*) o trama a trama (opción F3-*Frame*), según convenga, responder a las siguientes cuestiones:

- a) Si el conmutador está en el slot 4, en qué slot se encuentra cada una de las partes del sistema: la fuente, el primer enlace, el segundo enlace y el receptor? ¿A qué se deben tales desfases?
- b) En un ciclo de escritura, ¿cómo se determina la dirección de la memoria de voz donde se debe escribir? ¿Es la escritura secuencial o controlada?
- c) En un ciclo de lectura, ¿cómo se determina la dirección de la memoria de voz de donde se debe leer? ¿Es la lectura secuencial o controlada?
- d) En esta demostración, ¿cuál es el retardo que introduce el conmutador en cada canal de emisor? Justificar dicho retardo en relación al contenido de la memoria de control.
- e) Tomando todos los casos posibles, ¿cuáles son el máximo y el mínimo retardo que puede introducir el conmutador en un canal? Poner un ejemplo donde se alcancen ambas situaciones.

## 8. TIPOS DE SISTEMAS DE SEÑALIZACIÓN

Se llama señalización al proceso de intercambio de información entre el usuario que llama, el llamado y la red con el objetivo de establecer, mantener y liberar llamadas. En las redes de conmutación de circuitos las señales de control son el medio para gestionar la red y para establecer, mantener y finalizar las llamadas, intercambiando información entre el abonado y los conmutadores, entre los conmutadores entre sí y entre los conmutadores y el centro de gestión de red. Las funciones más importantes de la señalización son:

- Comunicación audible con el abonado, que incluye el tono de marcar, el tono de llamada, la señal de ocupado etc.
- Transmisión del número marcado a las centrales de conmutación que intentarán establecer la conexión.
- Transmisión de información entre conmutadores indicando que una llamada dada no se puede establecer.
- Transmisión de información entre conmutadores indicando que una llamada ha finalizado y que la ruta puede desconectarse.
- Generación de la señal que hace que el teléfono suene.
- Transmisión de información con fines de tarificación.
- Transmisión de información indicando el estado de los equipos, las líneas para emplear en el encaminamiento, mantenimiento y diagnóstico de fallos.

La señalización entre el abonado y la central local se basa fundamentalmente en el envío de señales analógicas (tono a invitación a marcar, tonos multifrecuencia asociado a cada dígito, señal de *ring*, señal de ocupado, etc.) que son posteriormente procesadas y digitalizadas para iniciar la señalización entre centrales. Por lo general, todas estas señales se transmiten en los mismos 4KHz del canal de voz y precisamente por ello son audibles. La señalización entre centrales maneja ya información digitalizada y puede ser por canal asociado (ACS) o por canal común (CCS). En el primer caso, la información de señalización viaja por la misma ruta que las muestras de la señal de voz, aunque eso sí, empleando un canal diferente. Toda la familia de tramas PDH (E1, E2, E3 y E4) utiliza ACS. En cambio, en la señalización por canal común, la información de señalización no utiliza la red de conmutación de circuitos sino que viaja por una red de conmutación de paquetes independiente y especializada. Este sistema es el más utilizado hoy en día.

## 9. SINCRONISMO Y SEÑALIZACIÓN EN UN SISTEMA TDM

Se utilizará el simulador TIMSWIT.EXE, opción B (*Time switching with tdm*), y se seleccionará la opción F5-*Synch & signalling demonstration*.



9.1. Mediante una ejecución ciclo a ciclo (opción F1-*Cycle*), slot a slot (opción F2-*Timeslot*) o trama a trama (opción F3-*Frame*), según convenga, responder a las siguientes cuestiones:

- a) ¿Qué tipo de señalización se utiliza, señalización asociada al canal o señalización por canal común? ¿Por qué? ¿En qué canal va ubicada? ¿Cuál es el canal de sincronismo?
- b) ¿Qué función tiene el módulo de sincronismo y señalización?
- c) Con el sistema de sincronizado, ¿qué operaciones realiza el módulo de sincronismo y señalización y el receptor al recibir el sincronismo de trama (representado en el simulador por el carácter "!") y el sincronismo de multitrama (representado en el simulador por el carácter "!!")? Tras recibir el receptor el sincronismo de multitrama, ¿cuál es el desfase entre fuente, conmutador y receptor? ¿Coincide con el indicado en el apartado 7.2?
- d) Para cargar la memoria de control se utilizan dos registros: el registro *Fm* y el registro *Ts4*. ¿Qué contiene cada uno de ellos? ¿Cuál de ellos determina la dirección de la memoria de control, el contenido de la memoria de control, la dirección de la fuente y la dirección del receptor? ¿Por qué?
- e) ¿Cuántas tramas hay en una multitrama? ¿Por qué?
- f) Describir la estructura de la trama y multitrama del sistema MIC 30+2 y compararla con la que se presenta en la práctica. ¿Qué analogías y diferencias existen entre ambas?