

# Conmutación de circuitos

Area de Ingeniería Telemática  
<http://www.tlm.unavarra.es>

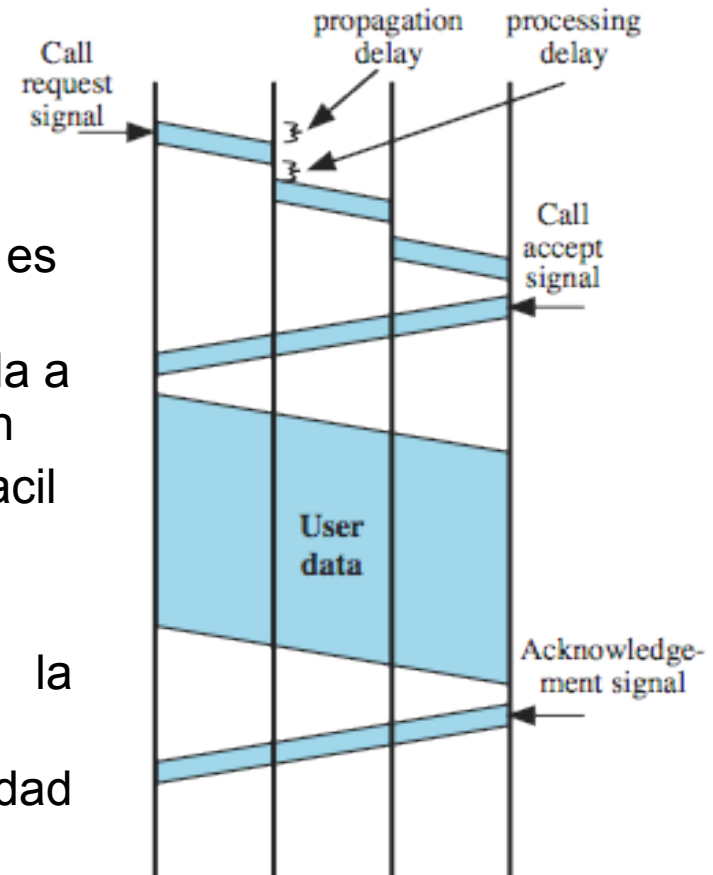
Arquitectura de Redes, Sistemas y Servicios  
3º Ingeniería de Telecomunicación

# C. Circuitos vs C. Paquetes

- Hemos visto redes de conmutación de paquetes
  - Los mensajes se dividen en paquetes
  - Los paquetes son transmitidos por un camino de origen a destino
  - Sin conexión (datagramas)
  - O circuitos virtuales
- Conmutación de circuitos
  - Se crea un circuito de la fuente al destino
  - El circuito físico real conectado queda dedicado
  - En c.c. virtuales los paquetes de diferentes circuitos comparten el servicio de transmisión

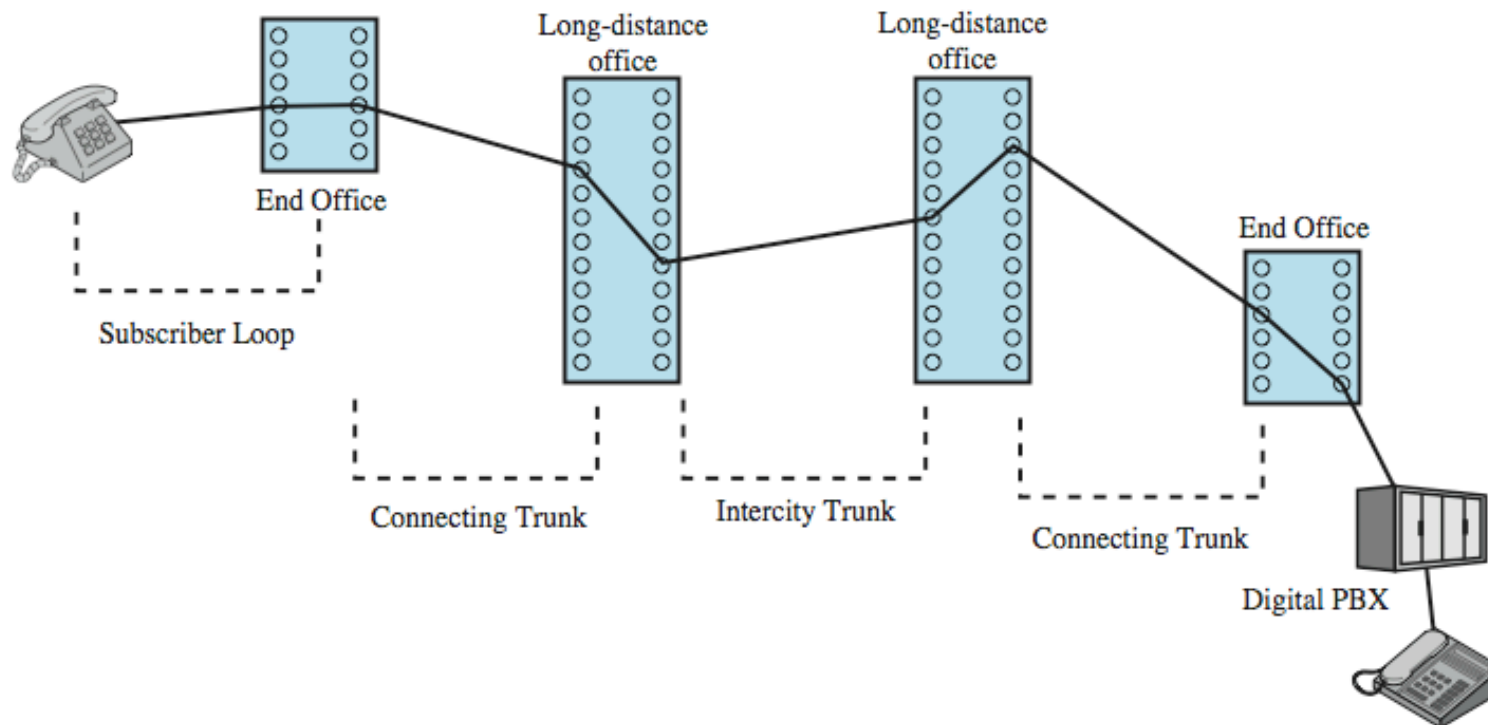
# Conmutación de circuitos

- Camino dedicado entre dos terminales
- Tres fases:
  - Establecimiento
  - Transferencia
  - Desconexión
- Ventajas
  - Una vez conectado, la transferencia es transparente
  - La capacidad del canal está asignada a la conexión durante toda su duración
  - Calidad de servicio conocida (más fácil que en conmutación de paquetes)
- Desventajas
  - Capacidad del canal asignada a la conexión durante toda su duración
  - Si no se envían datos: capacidad desperdiciada
  - Establecimiento añade retardo



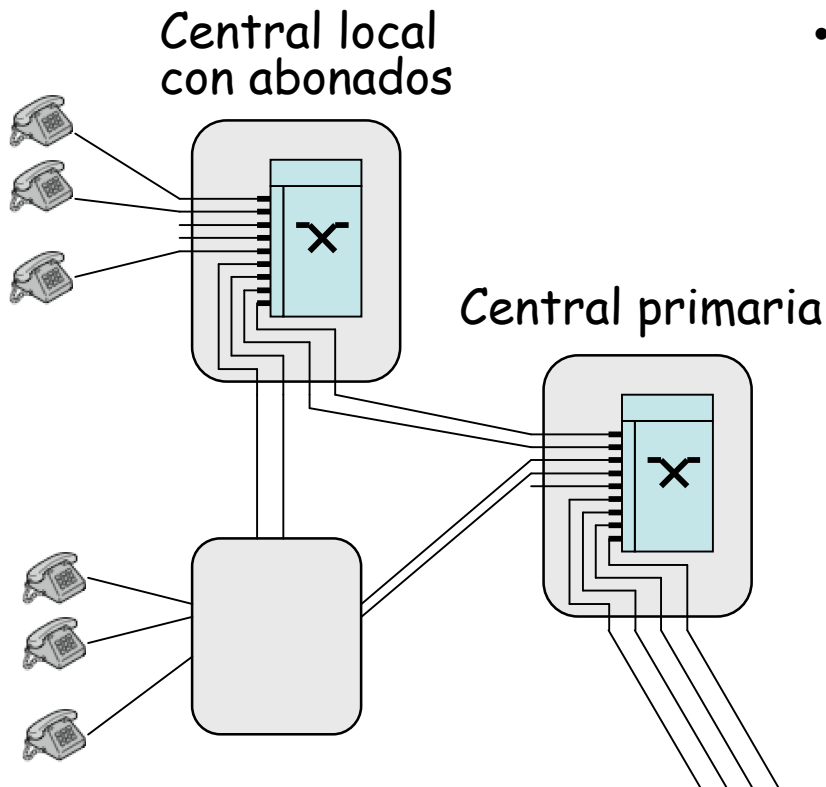
# Red pública telefónica conmutada

- Abonados (*subscribers*): teléfonos o modems
- Líneas de usuario (*subscriber line, local loop*): par trenzado
- Centrales de conmutación (*exchanges*)
  - Central local (*End-office*): tiene abonados (miles) de una zona localizada
- Enlaces (*trunks*):
  - En España más de 8.000 ayuntamientos: todas con todas → ¡ más de 32M enlaces !
  - Más de 700 ciudades (>10.000 habs): todas con todas → ¡ más de 200K enlaces !

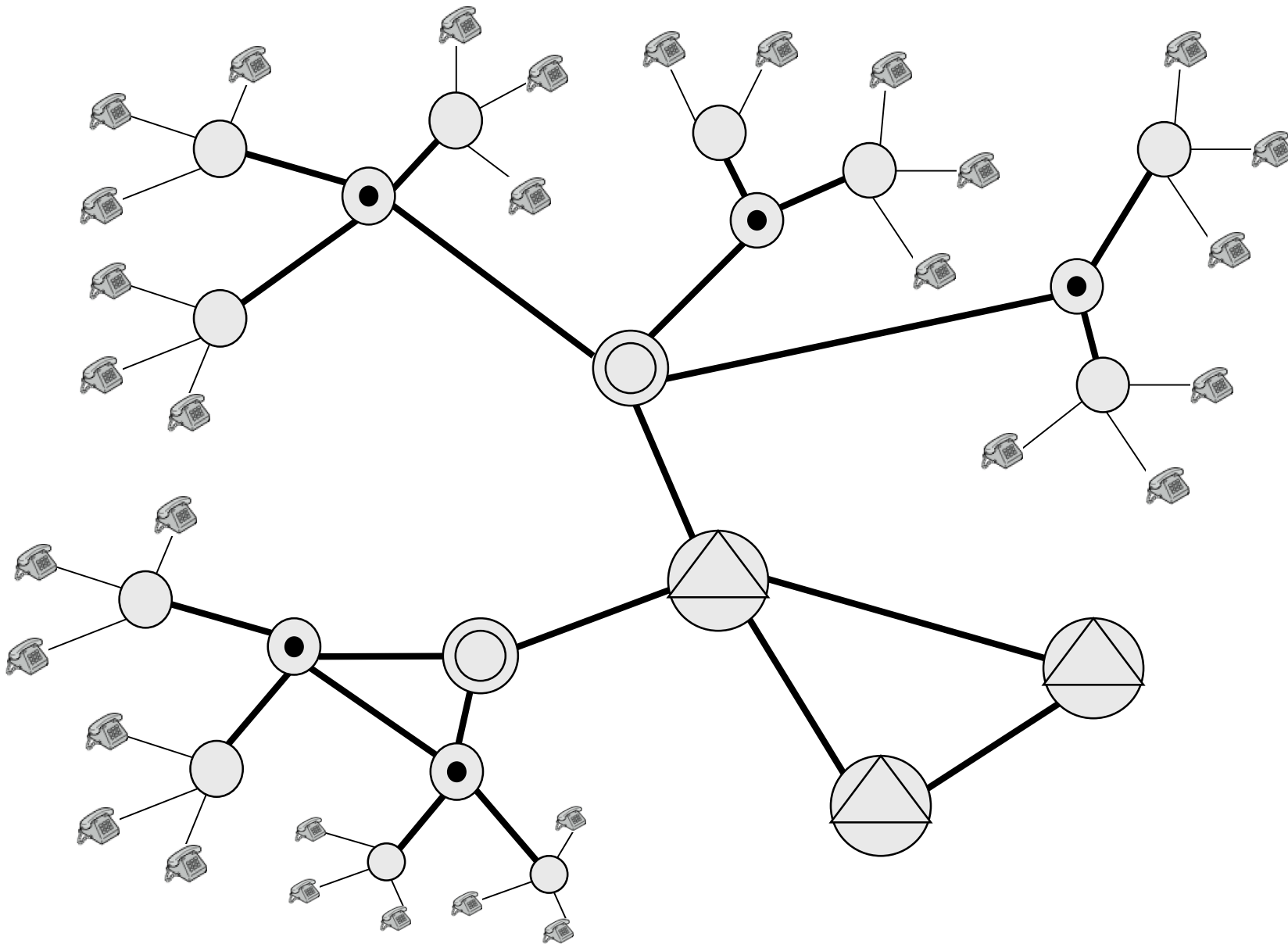


# Arquitectura

- Centrales locales:
  - Conectan a usuarios de esa central entre si
  - Conectan a usuarios a una de las líneas troncales
- Centrales primarias, secundarias, terciarias:
  - Conectan líneas entre centrales
- Los enlaces entre centrales son conjuntos de líneas que se pueden conectar por separado
- Bell System Hierarchy, Switch Class:
  - 1- Regional center
  - 2- Sectional center
  - 3- Primary center
  - 4- Toll center
  - 5- End office

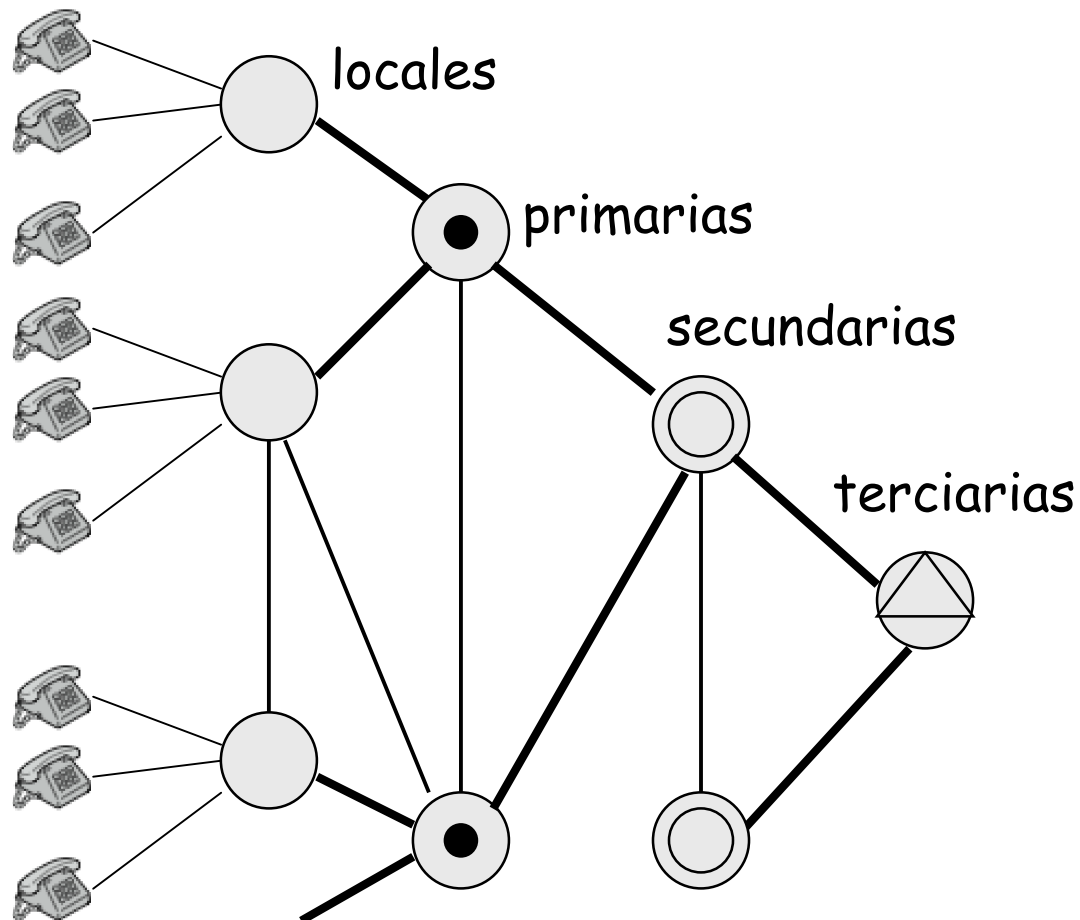


# Arquitectura

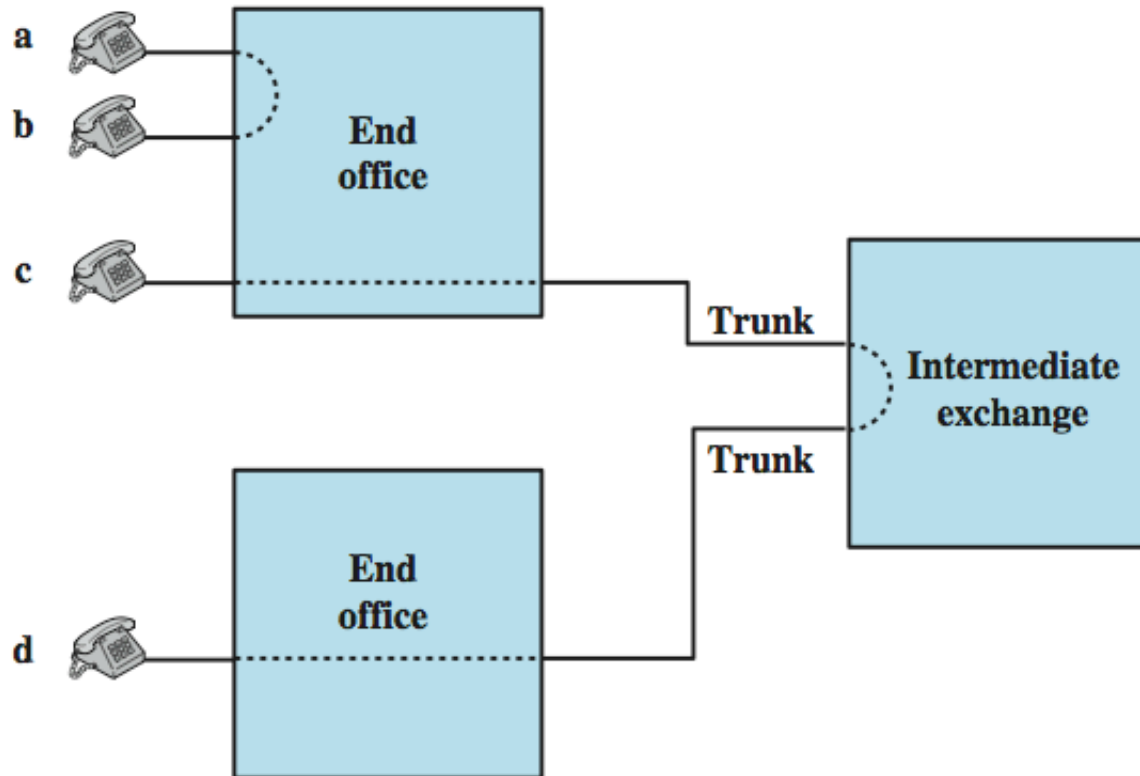


# Arquitectura

- Las centrales se organizan en red jerárquica por niveles (locales, primarias, secundarias, terciarias...)
- Facilita el encaminamiento: siempre hay un superior jerárquico



# Establecimiento de circuitos

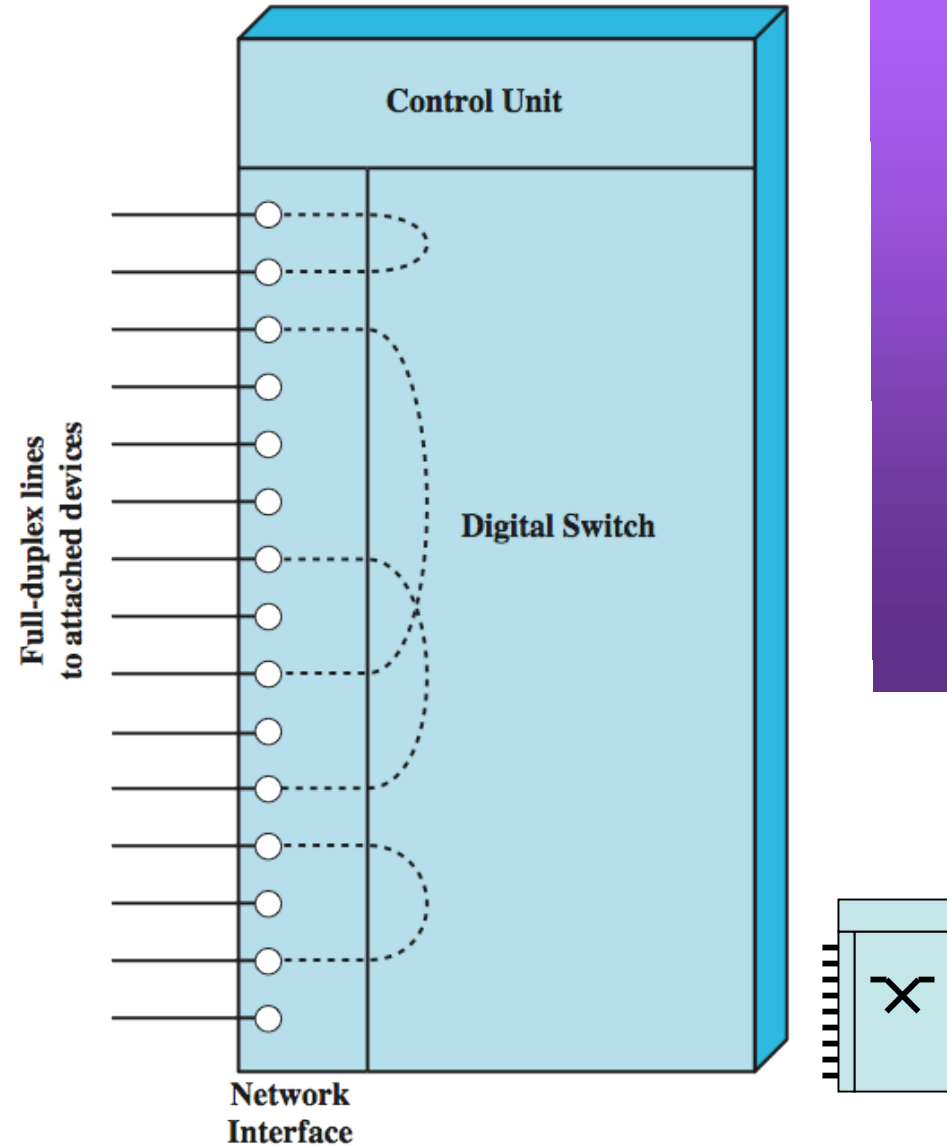
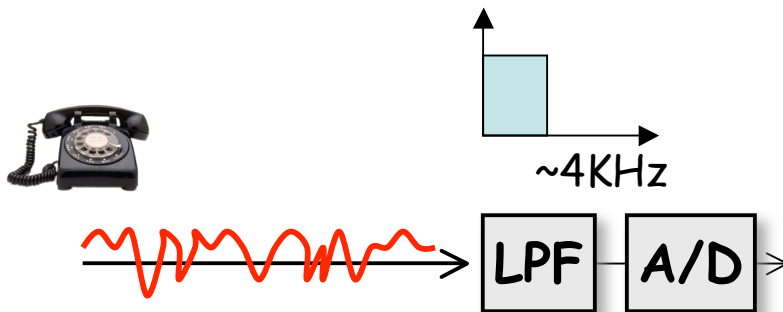


- Tráfico entre usuarios de la misma central no sale de la central
- Tráfico entre usuarios de diferentes centrales se cursa a través de enlaces troncales (*trunk*) y posibles centrales intermedias



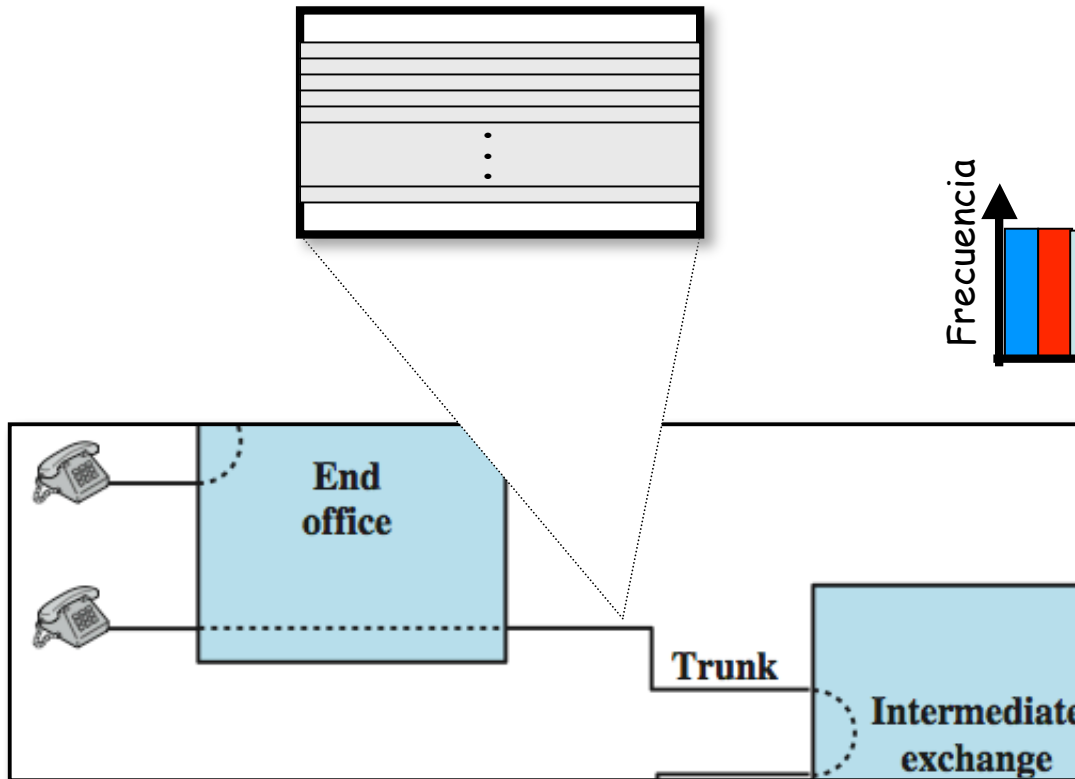
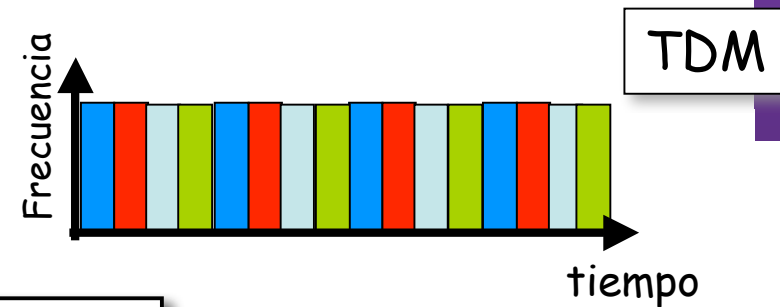
# Elemento de conmutación de circuitos

- Líneas de entrada
  - Full-duplex
- Unidad de control
  - Establece, mantiene y libera caminos en el switch
- Conmutador digital
  - Conecta entre si las líneas de entrada según le indica la unidad de control



# Trunks (troncales)

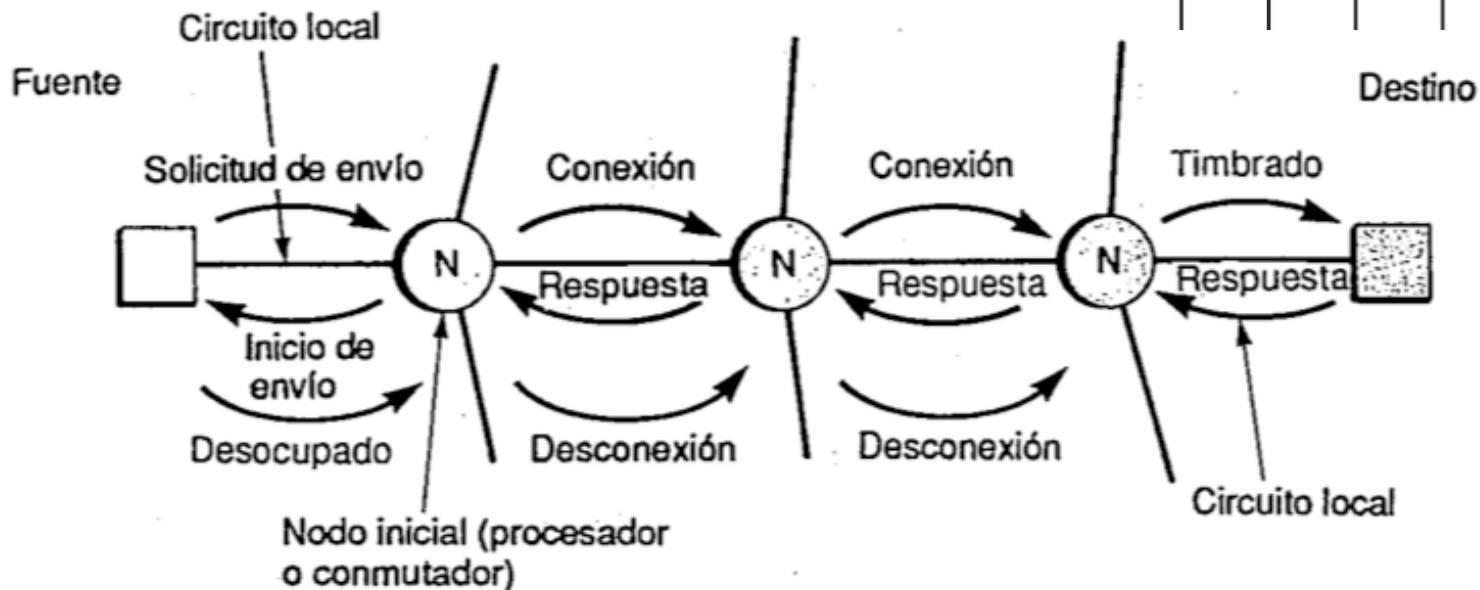
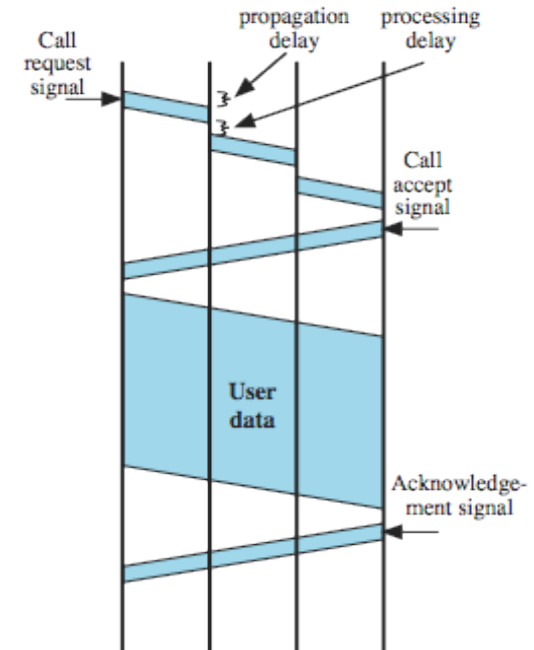
- Capacidad para múltiples circuitos simultáneos
  - Múltiples medios físicos (cables)
  - FDM
  - TDM



# Señalización

*“Intercambio de información de control entre los nodos de la red y entre terminales de abonado y la red”*

- Las unidades de control de las centrales se comunican entre si para
  - Establecimiento de llamadas
  - Liberación de llamadas



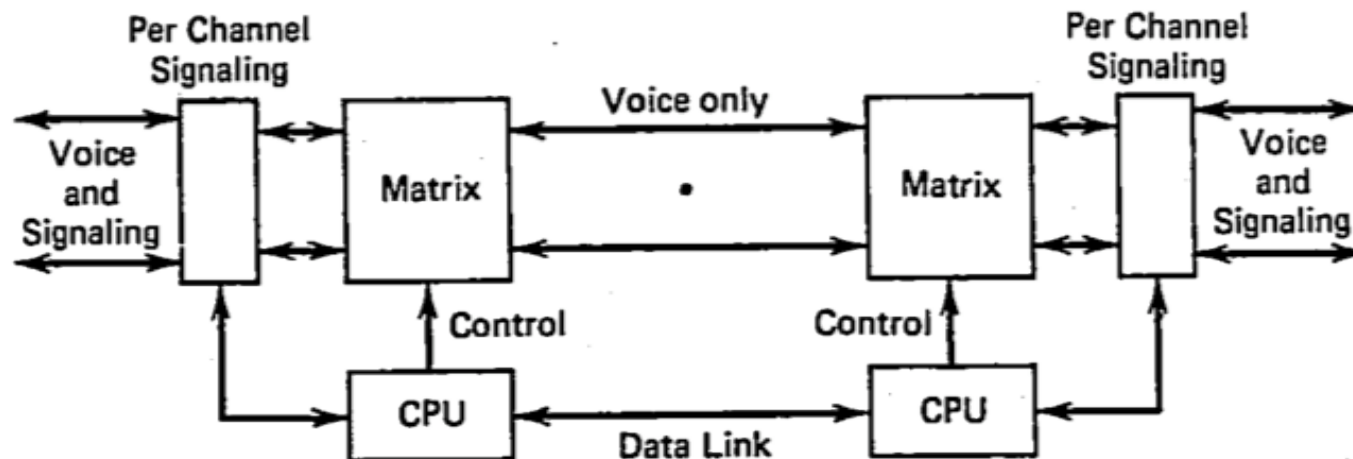
# Señalización

## Señalización en canal

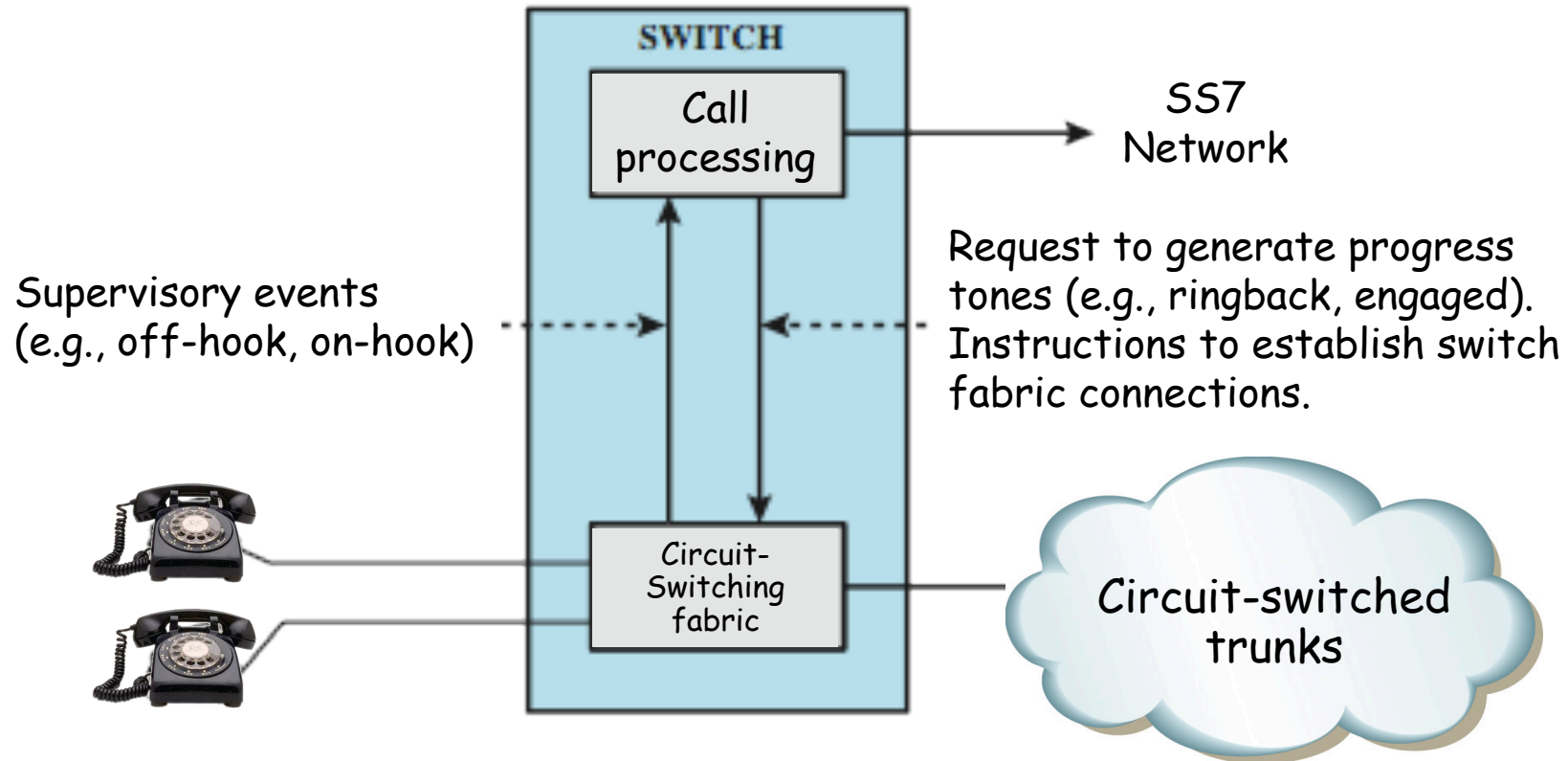
- Usa los mismos recursos de transmisión para la voz y para la señalización
- Puede ser “en banda” o “fuera de banda” (banda de frecuencias vocal)
- Ej. en banda: tonos en el marcado. Ej. fuera de banda: continua (DC) en el bucle de abonado para detectar el descuelgo

## Señalización por canal común (CCS = Common Channel Signaling)

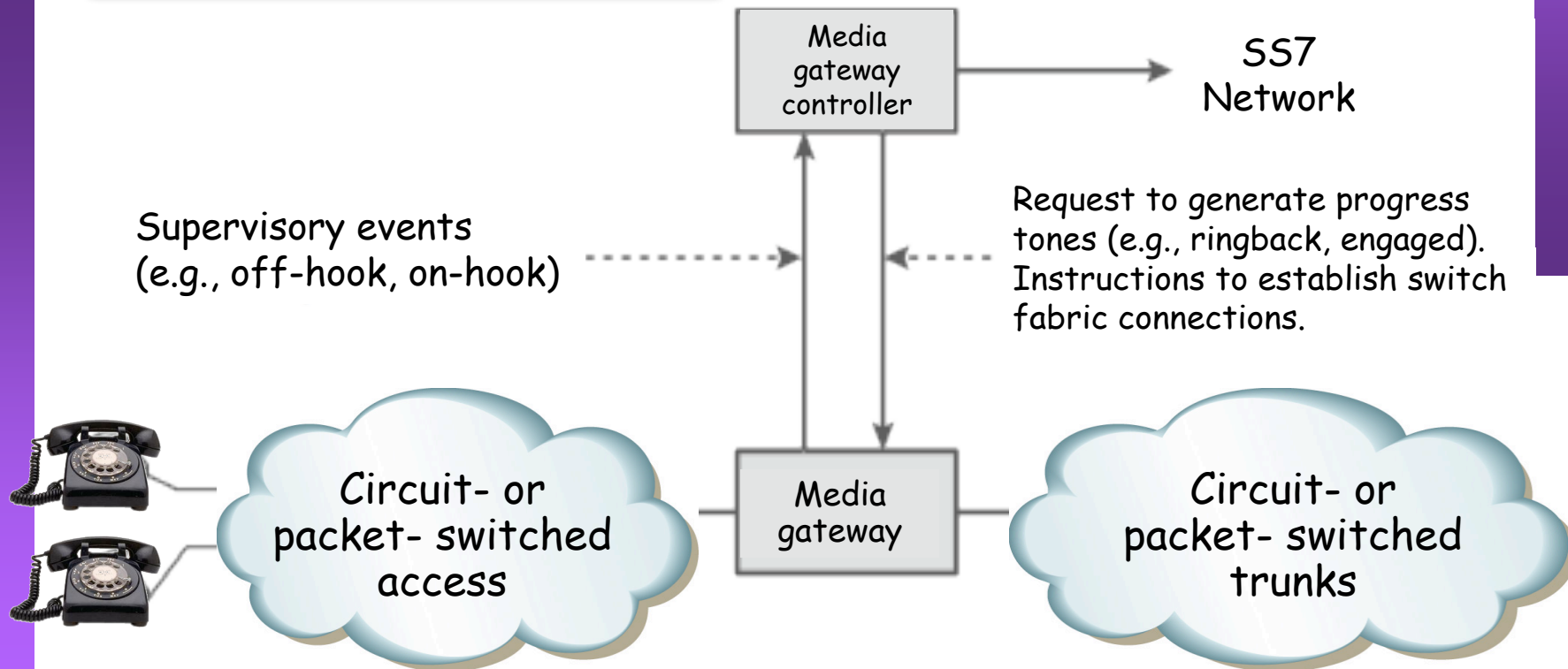
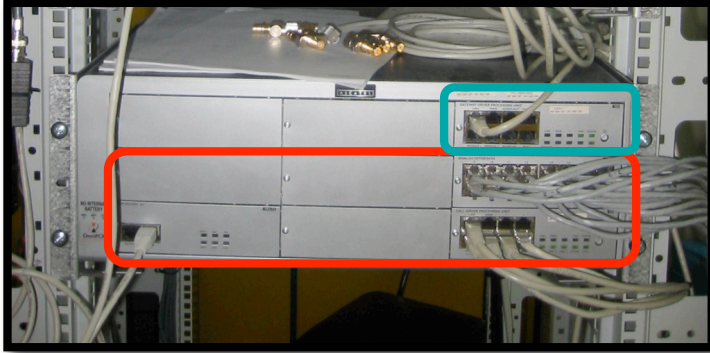
- Emplea un canal dedicado entre las CPUs de los conmutadores
- Puede ser CCS por “canal asociado”
- Los mensajes pasan entre los nodos de conmutación (*store-and-forward*)
- Los mensajes pueden emplear caminos diferentes a los de la voz
- Se crea así una red de conmutación de paquetes para la señalización
- El protocolo empleado hoy en día es el CCITT Signaling System No. 7 (SS7)



# Nodo tradicional

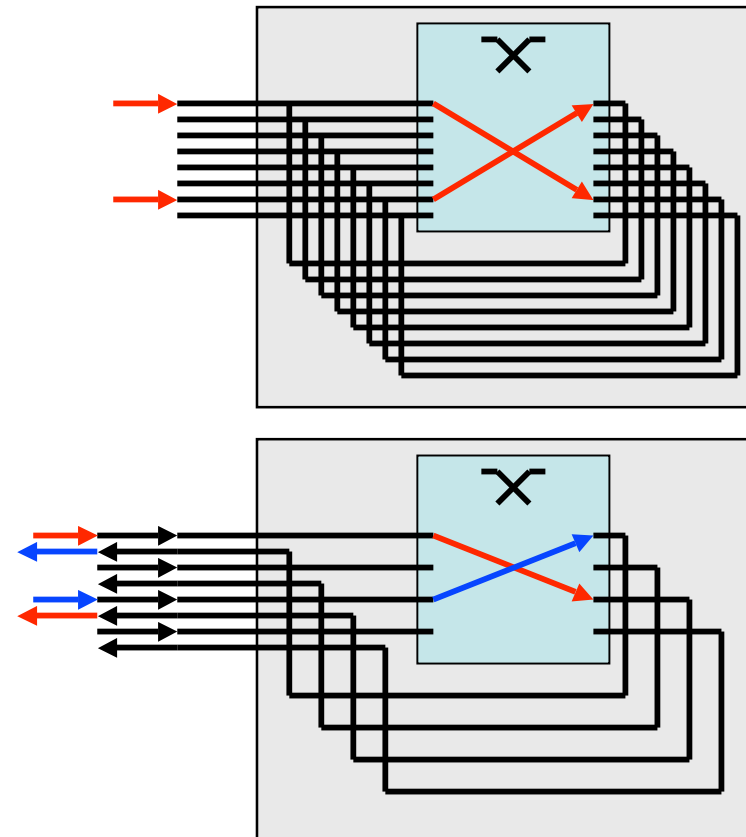
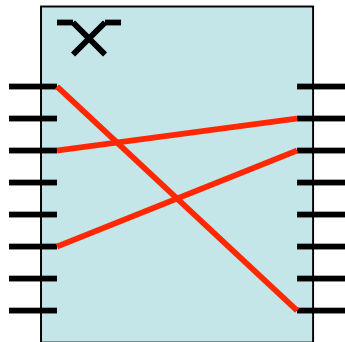


# Softswitch



# Conmutadores

- Permite conectar líneas de entrada a líneas de salida
- Se puede usar para construir un conmutador que interconecte líneas full duplex entre si



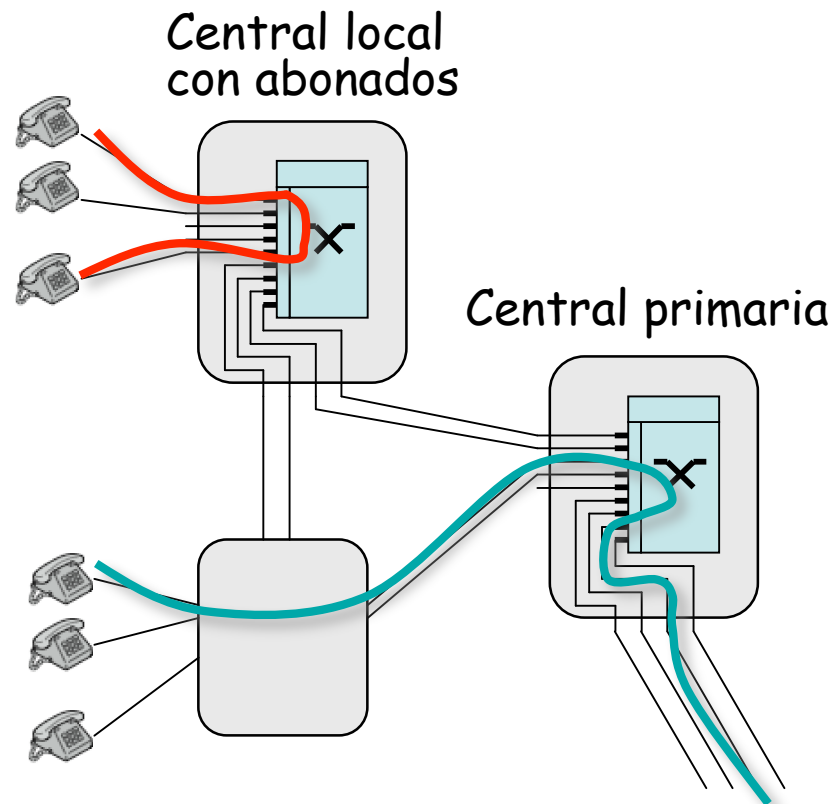
# Conmutadores

## **Line switch**

- Central local
- Inteconecta líneas individuales
- Debe conectar una entrada específica con una salida específica

## **Transit switch**

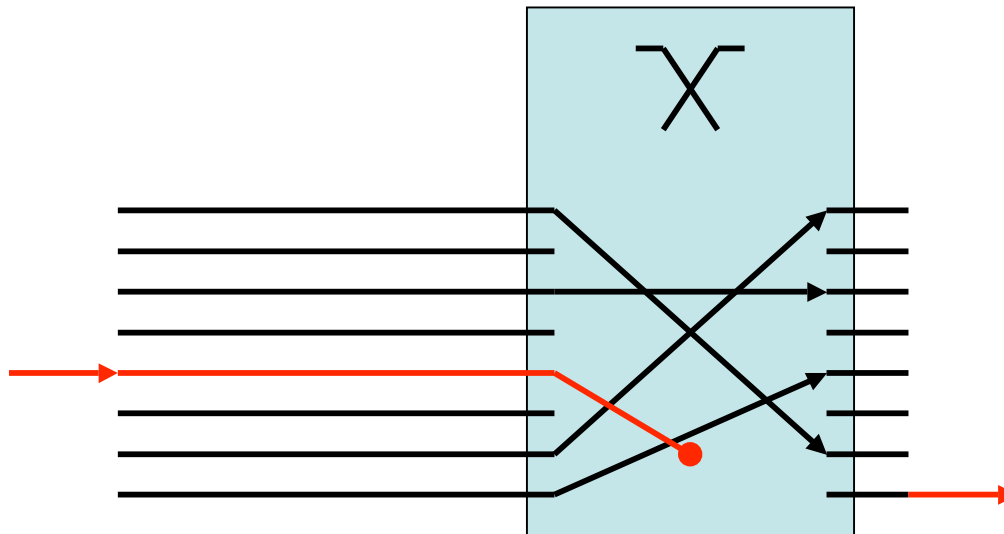
- Interconecta troncales
- Debe conectar una entrada específica con una salida *cualquiera en la dirección correcta*





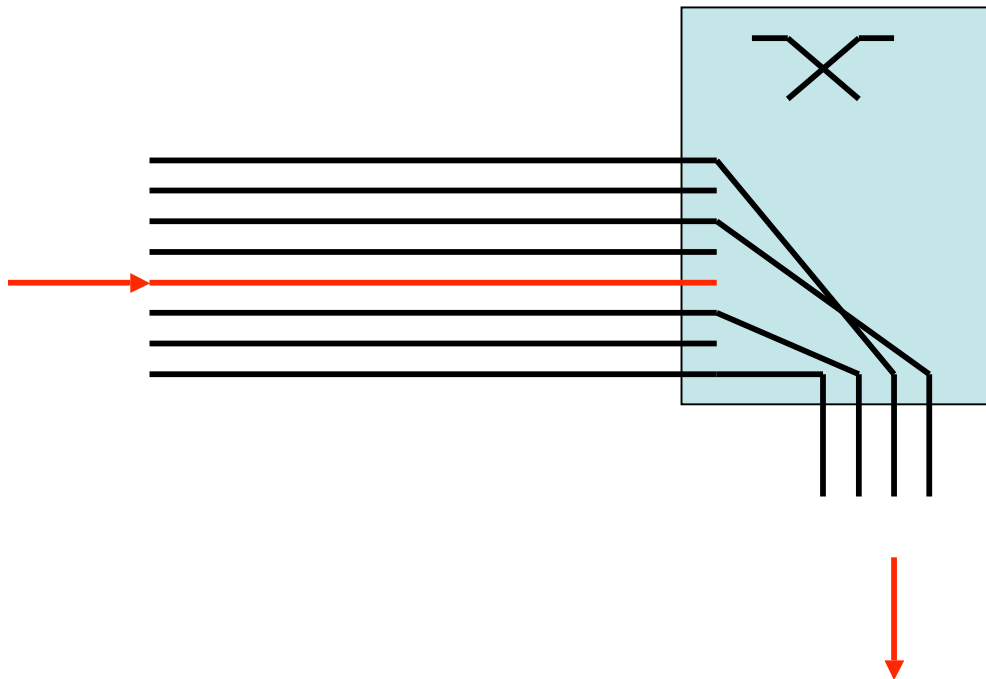
# Bloqueo

- Cuando no se puede interconectar dos estaciones aunque estén libres
- Bloqueo interno
  - El conmutador no tiene recursos para hacer llegar un circuito de la entrada a la salida



# Bloqueo

- Cuando no se puede interconectar dos estaciones aunque estén libres
- Bloqueo externo
  - El conmutador no tiene suficientes recursos de salida para cursar una nueva llamada



# Bloqueo

- Cuando no se puede interconectar dos estaciones aunque estén libres
- Red de conmutación con bloqueo
  - En sistemas de voz se suele utilizar
  - Llamadas de voz suelen ser de corta duración
  - Se dimensiona para que suceda infrecuentemente
- Red de conmutación sin bloqueo
  - Permite a todas las estaciones conectarse a la vez
  - La única causa por la que una conexión puede ser rechazada es porque la estación destino esté ocupada
  - Se utiliza más en redes de conmutación para datos

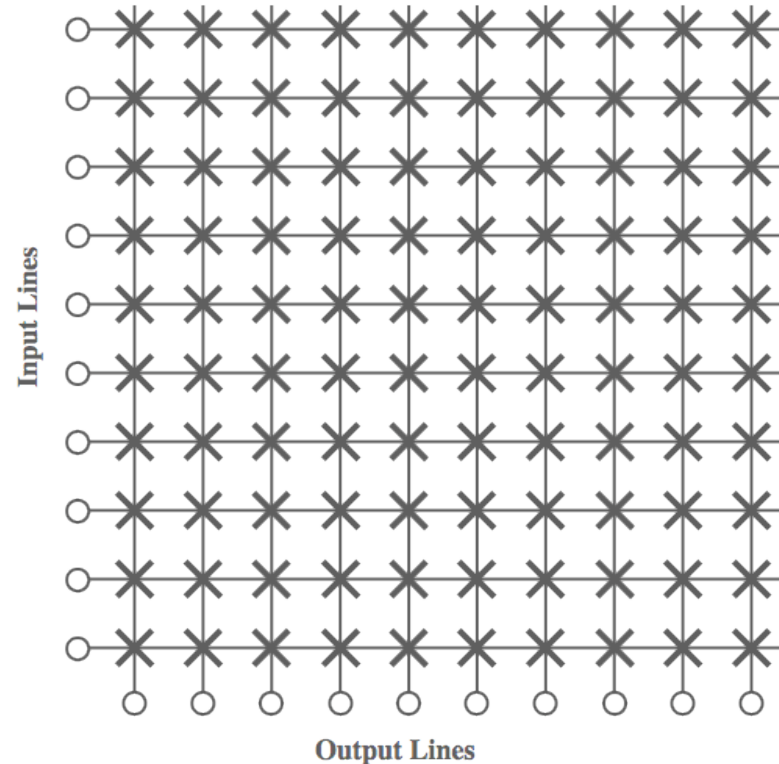
# Tipos básicos de conmutadores

- Conmutador espacial (S)
- Conmutador temporal (T)
- Conmutadores por fases (TST, STS...)

# Conmutador espacial

- Permite conectar las líneas de entrada con las líneas de salida elegidas (caminos espaciales)
- Tecnología **Crossbar**
  - Un bus por cada línea de entrada
  - Un bus por cada línea de salida
  - *Crosspoints* permiten conectar cada bus a cualquier otro
- La complejidad y coste depende del número de puntos de cruce

## *Space division switch*

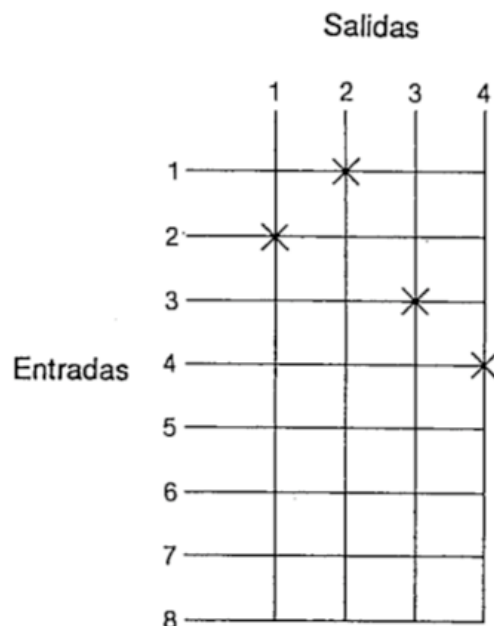


$N \times N$  crossbar matrix ( $N=10$ )

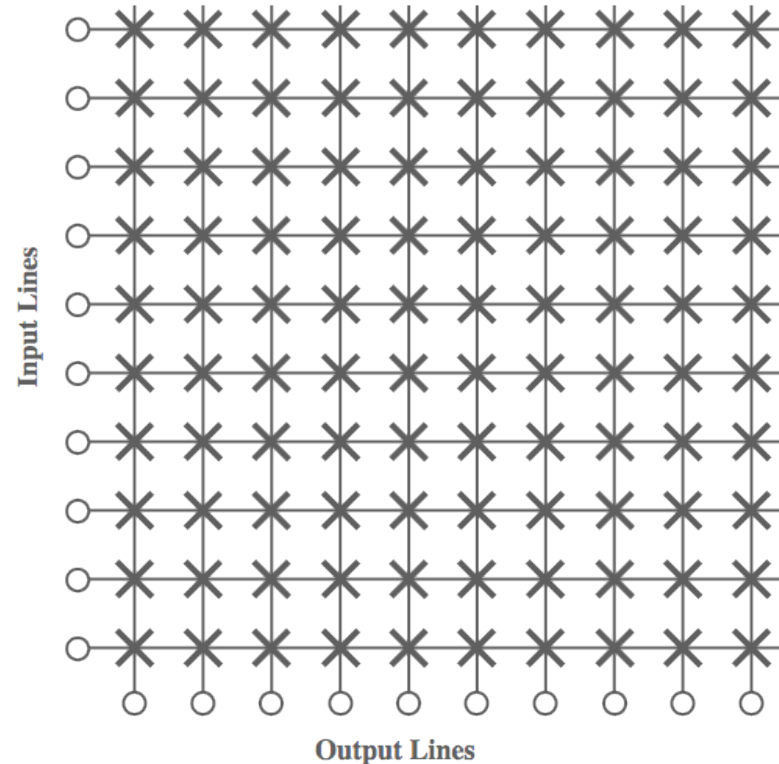
# Conmutador espacial

- Conmutador  $N \times K$
- Si  $K \geq N$  : sin bloqueo
- $N \times N$ :
  - El número de *crosspoints* crece con  $N^2$
  - Uso de *crosspoints* ineficiente
  - Máx  $N/2$  circuitos simultáneos
  - $N^2 - N/2$  *crosspoints* sin utilizar

- ¿Se puede hacer con menos *crosspoints*?



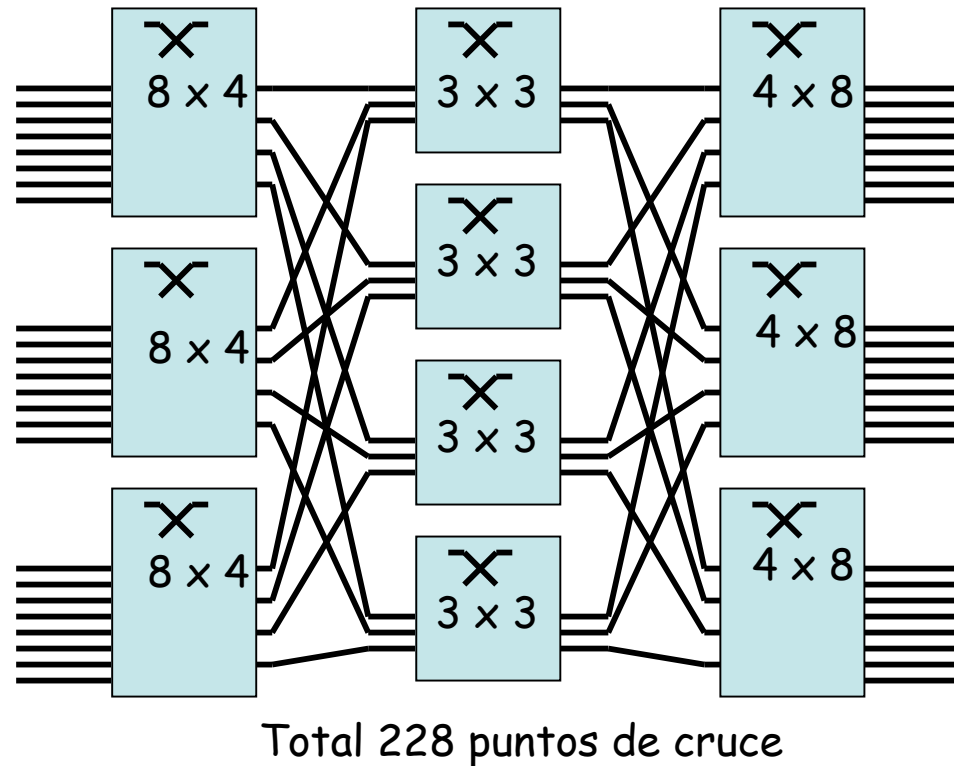
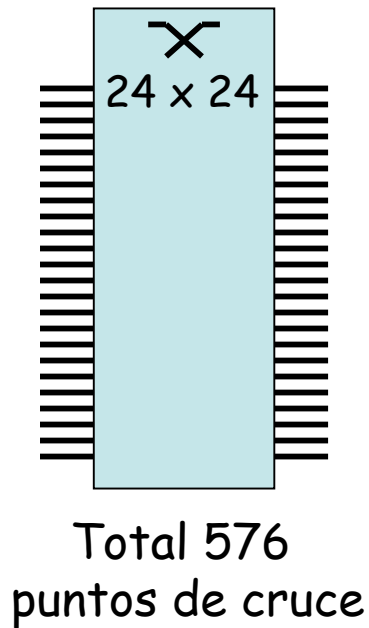
## Space division switch



$N \times N$  crossbar matrix ( $N=10$ )

# Conmutador con etapas (*stages*)

- Diferentes etapas
  - Seleccionamos líneas y las mandamos a conmutadores intermedios
  - Los conmutadores intermedios envían las líneas al bloque de salida deseado
  - Conmutadores internos más sencillos
  - Más de un camino interno posible

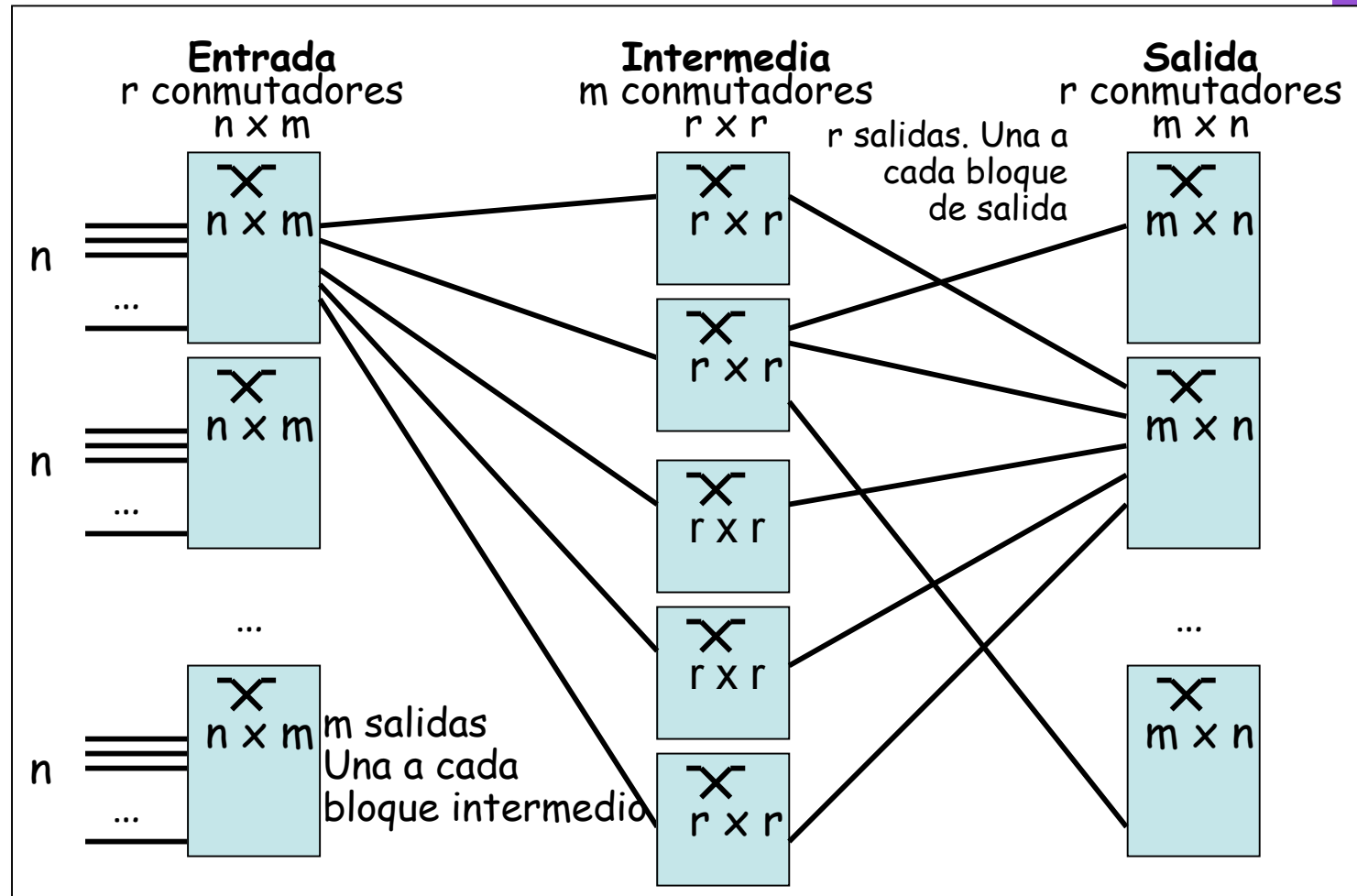
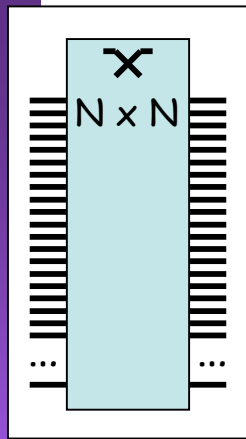


# Crosspoints

$C_1 = N^2$  crosspoints

$C_2 =$

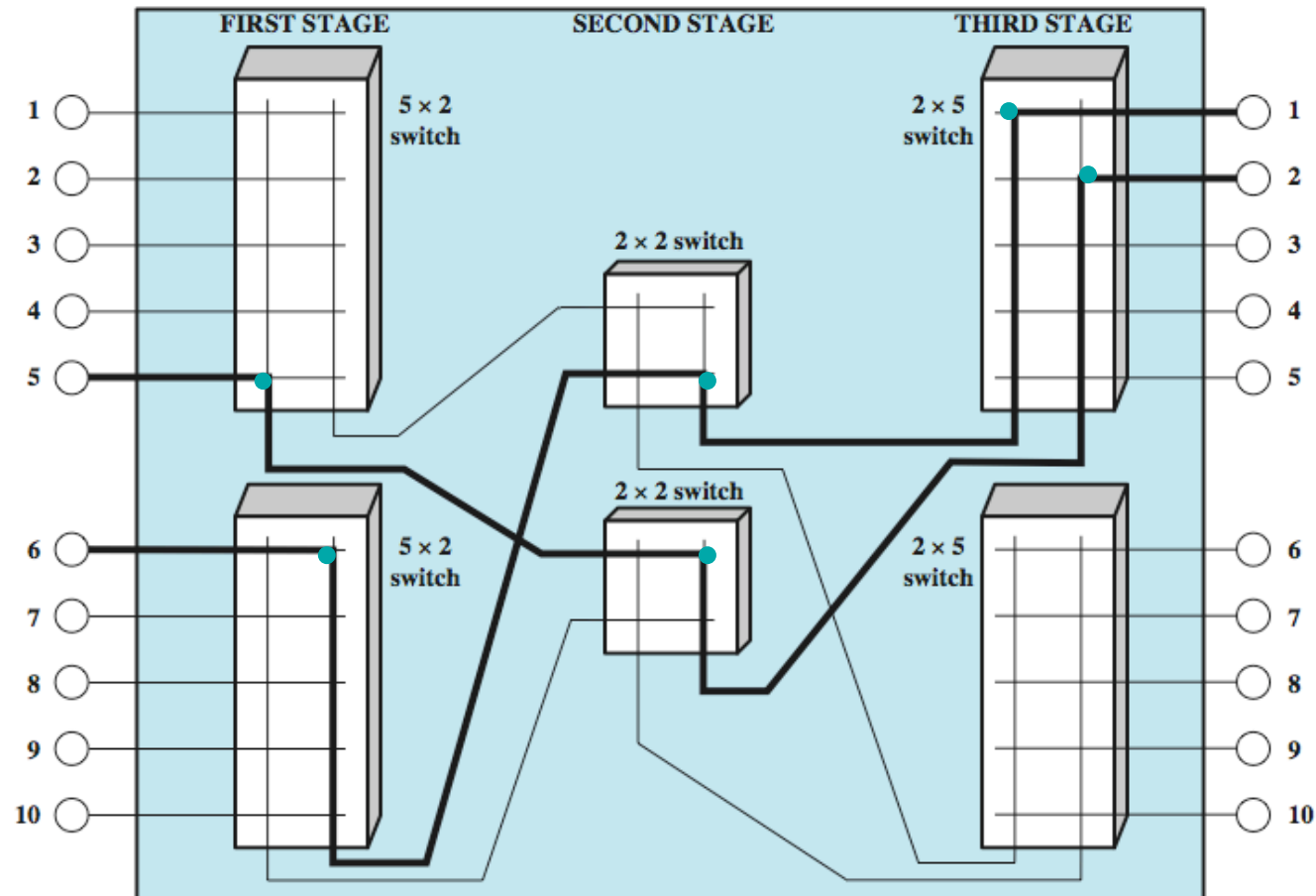
$$r = \frac{N}{n}$$



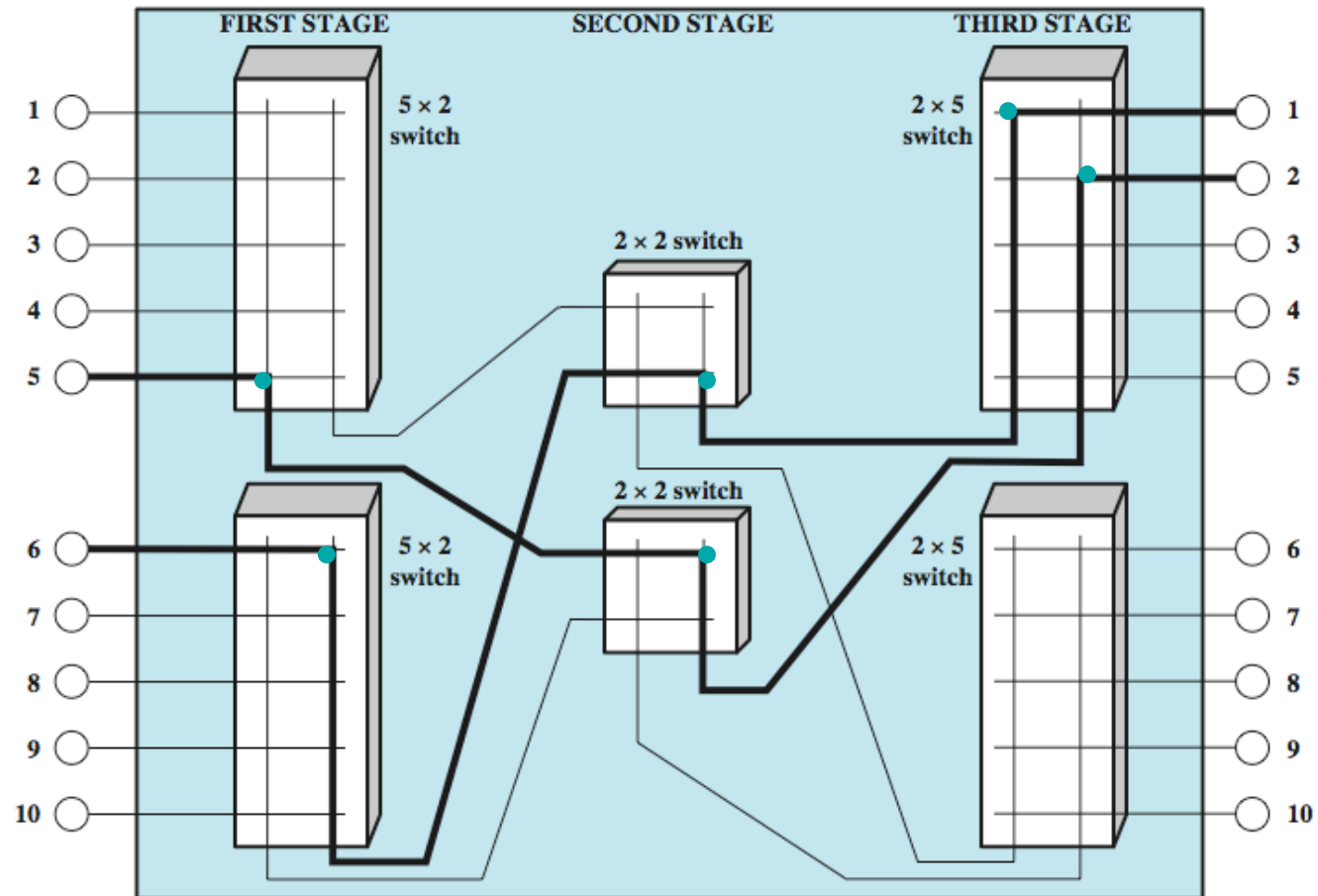


# Conmutador 3 etapas

- (3 stage space division switch SSS)
- ¿ Qué problema tiene este conmutador ?

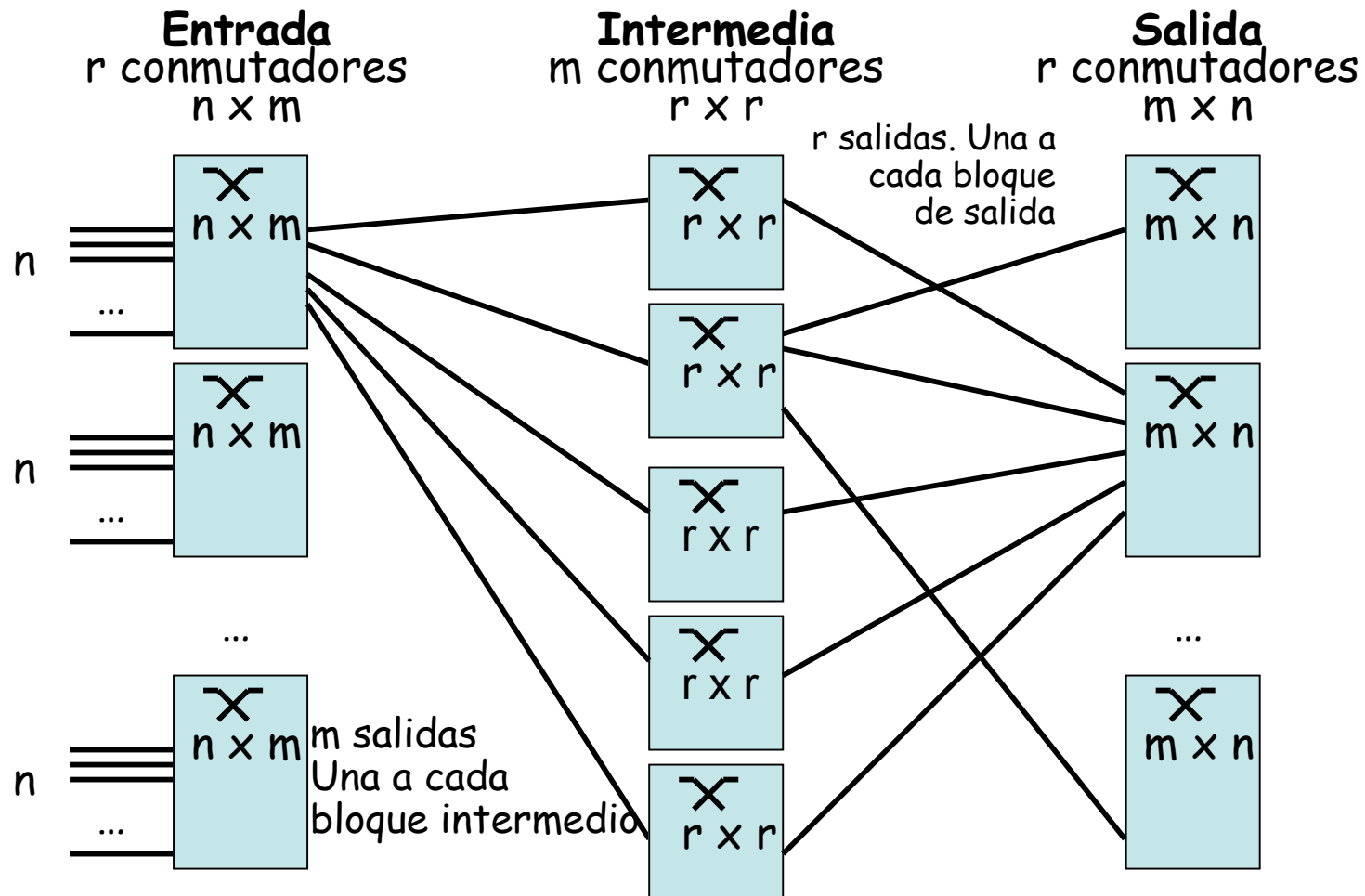


# Conmutador 3 etapas



# Redes de Clos

- ¿ Cuantos conmutadores intermedios  $m$  necesito para que no haya posibilidad de bloqueo interno ?
- Se entiende que al menos  $m \geq n$  (evitar bloqueo de la entrada)

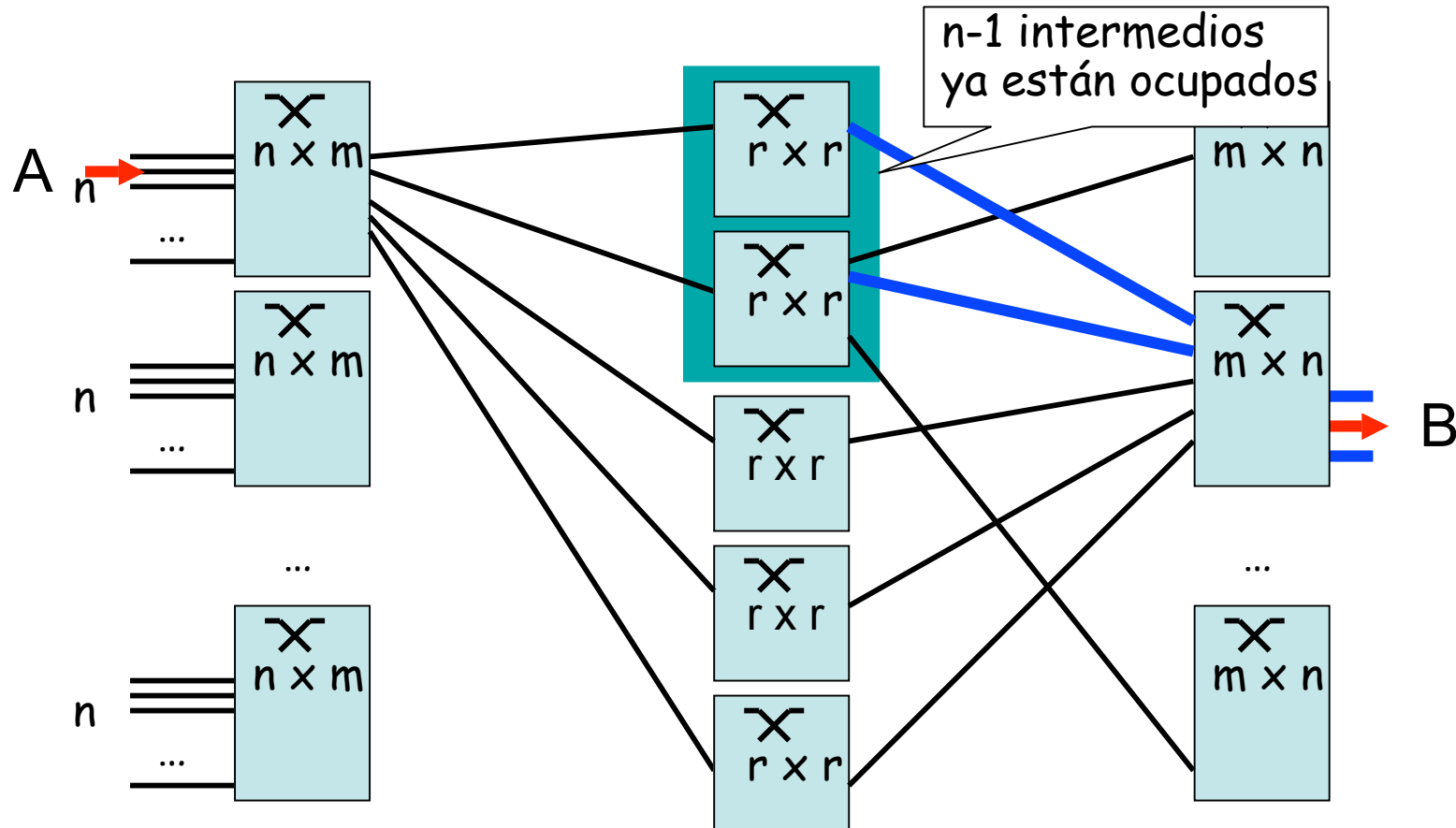


# Redes de Clos

## Condición en el caso peor

- De la entrada A a la salida B tiene que haber un camino posible
- Tiene que haber al menos un conmutador intermedio que tenga una línea libre al bloque de B
- En el caso peor habrá  $n-1$  ocupados, es decir, todas las otras salidas del bloque de salida ocupadas)

Linea A, conectar a la salida B

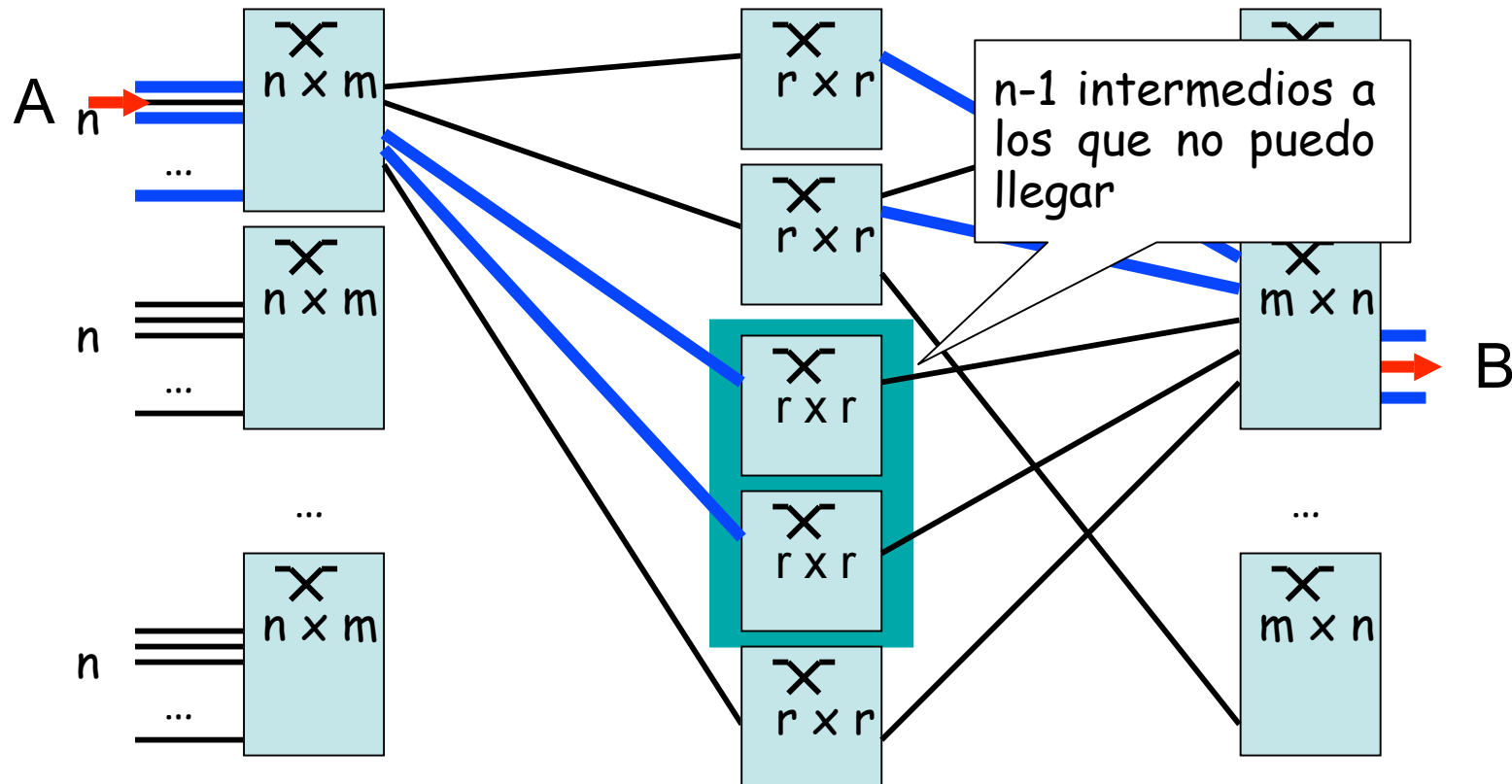


# Redes de Clos

## Condición en el caso peor

- Tiene que ser un conmutador intermedio al que esté libre la línea desde el bloque de entrada
- En el peor caso los conmutadores intermedios que no puedo usar por las dos razones no serán los mismos
- Tiene que haber conmutadores intermedios suficientes para los dos casos y uno más para tener camino para la llamada nueva

Linea A, conectar a la salida B

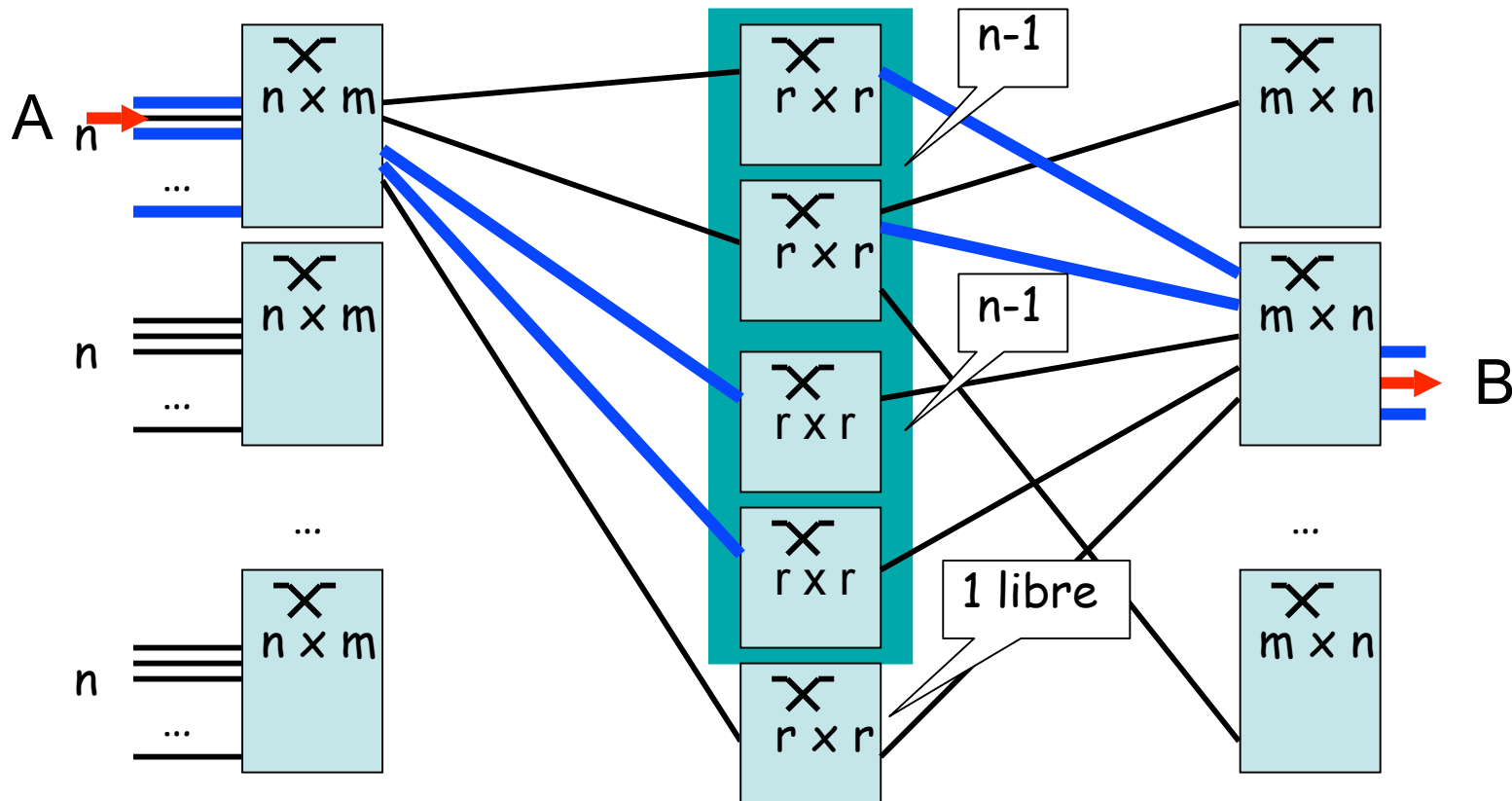


# Condición de Clos

- En un conmutador  $(r \ n) \times (r \ n)$  formado con  $m$  conmutadores intermedios el número  $m$  de conmutadores intermedios necesarios para que no exista probabilidad de bloqueo tiene que ser al menos
 
$$m \geq 2 \times (n-1) + 1$$

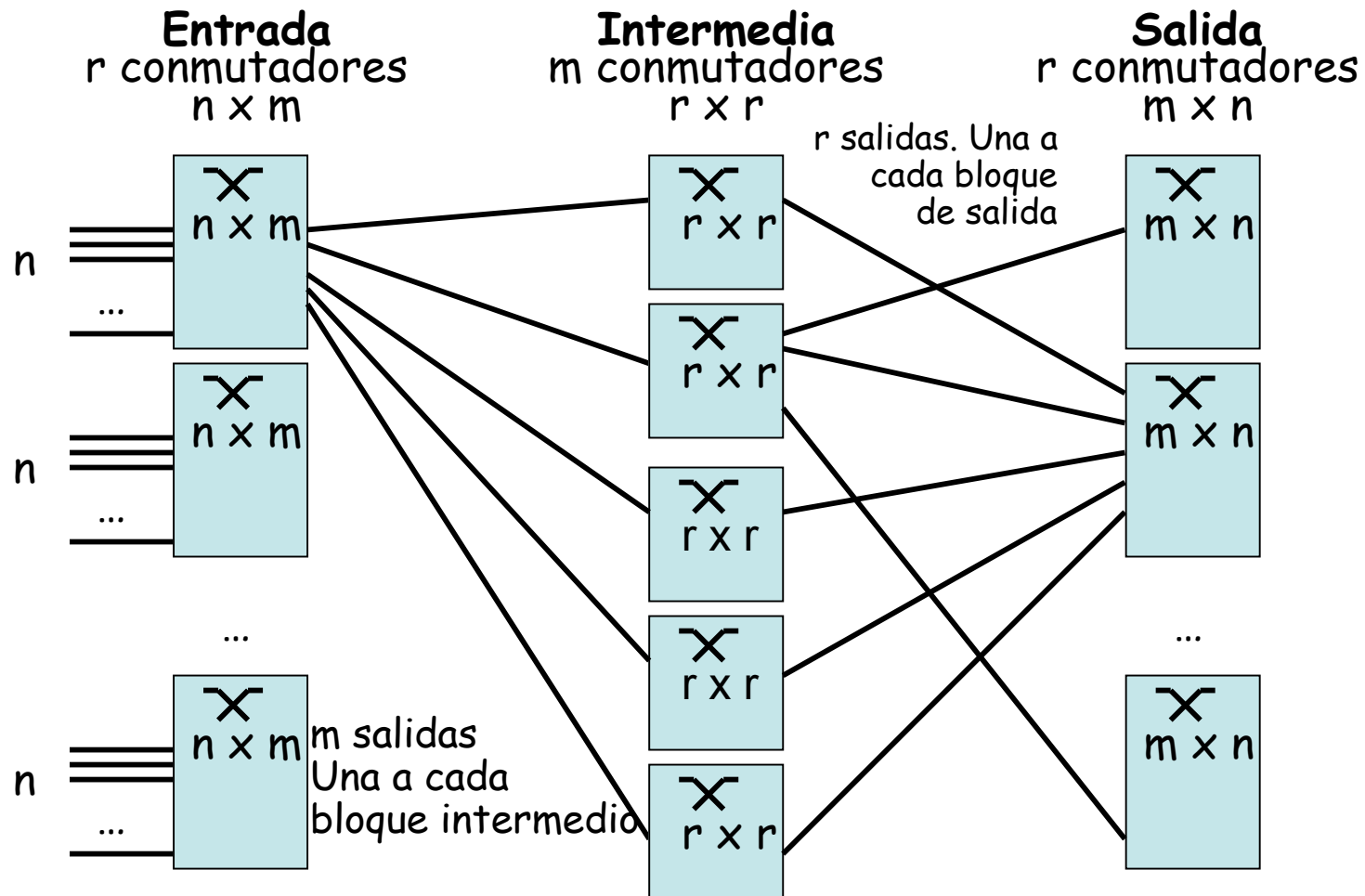
$$m \geq 2n - 1$$
- Un conmutador construido así no tiene bloqueo interno y tendrá normalmente menos puntos de cruce que un *crossbar* entero

Linea A, conectar a la salida B



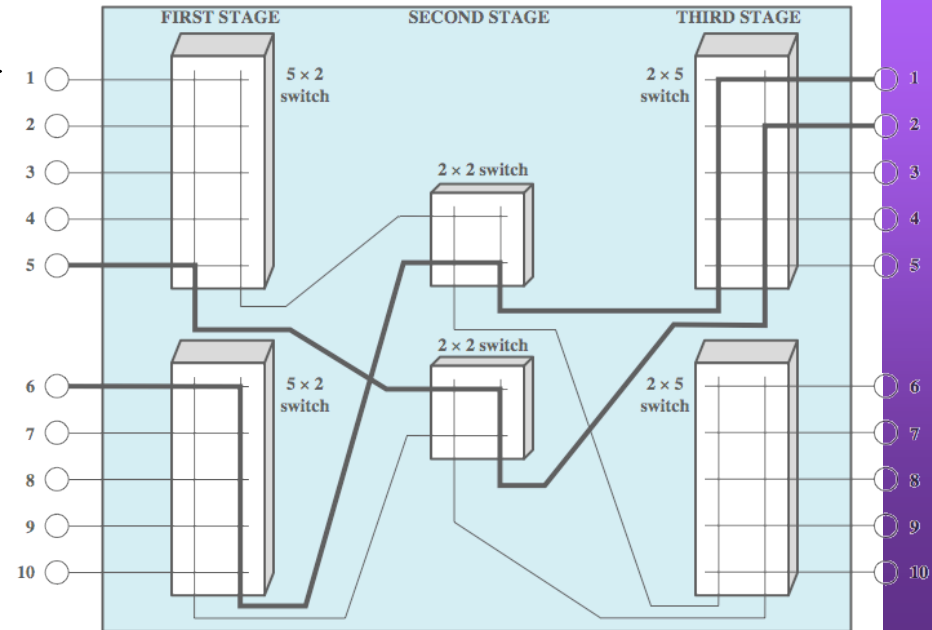
# Número de puntos de cruce

- Como ya se vio:



# Conmutador 3 etapas

- El ejemplo anterior no cumplía la condición de Clos →
- ¿Cuántos conmutadores intermedios necesitamos para que no pueda tener bloqueo?

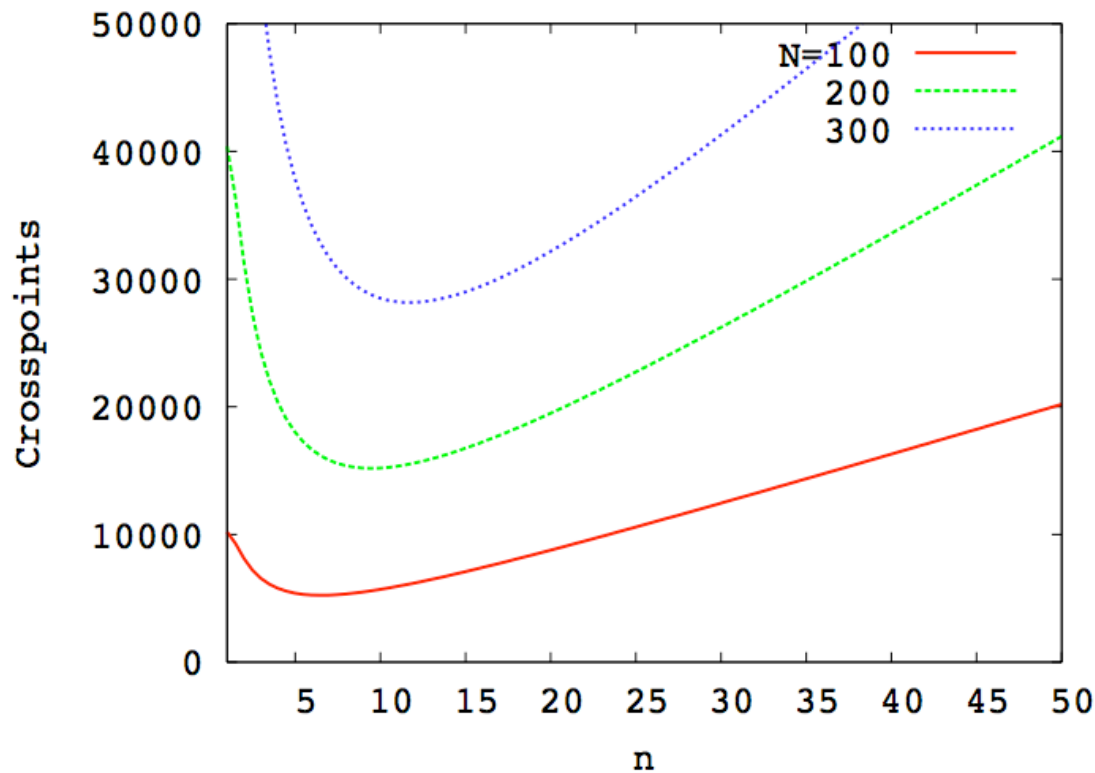


- *Crosspoints* en el ejemplo: \_\_\_\_.
- *Crosspoints* necesarios para un *crossbar* 10x10 : \_\_\_\_.
- Cumpliendo la condición de Clos: \_\_\_\_\_.
- ¡ El ser multietapa no garantiza menos puntos de cruce !



# Crosspoints

- Conmutador 3 etapas que cumple la condición de Clos mínima
- $C$  = número de *crosspoints*



¿Mínimo número de *crosspoints*?

Ejemplo: si  $N=100.000$

$$C_{crossbar} = 10^{10}$$

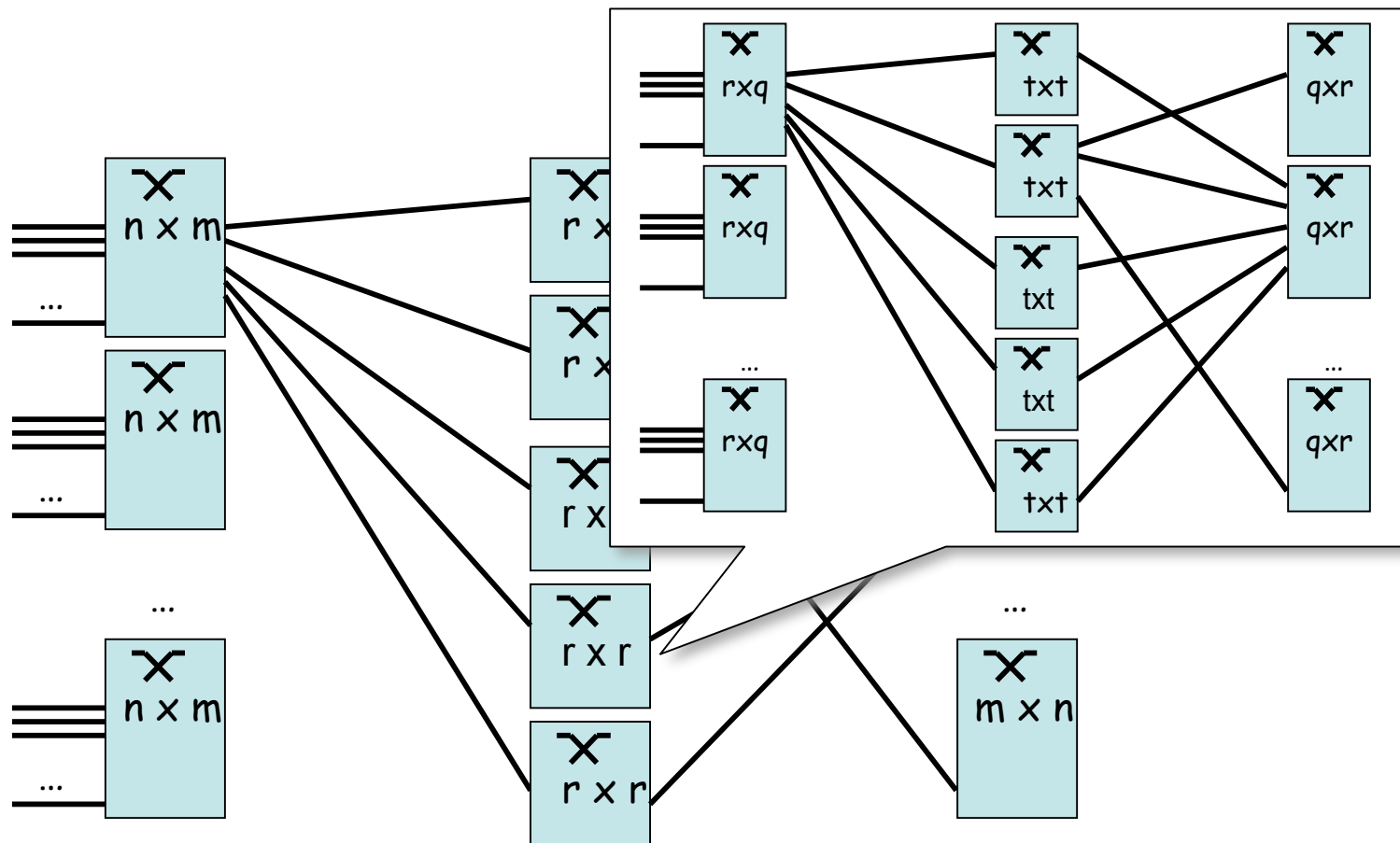
$$C_{nb, \acute{o}pt} =$$

# Conmutadores espaciales multietapa

- Los conmutadores que cumplen la condición de Clos no tienen bloqueo interno
- Los conmutadores que cumplen la condición  $m \geq n$ 
  - Se puede hacer que no tengan bloqueo interno
  - Hace falta que el sistema de control sea capaz de recolocar llamadas ya establecidas
  - *Rearrangeably nonblocking Clos networks*
- Escalan mejor que un *crossbar* al aumentar el número de entradas
- Seguimos necesitando muchos puntos de cruce para conmutar centenares de miles de canales telefónicos
- Optimizando aún más

# Reducir más el nº de crosspoints

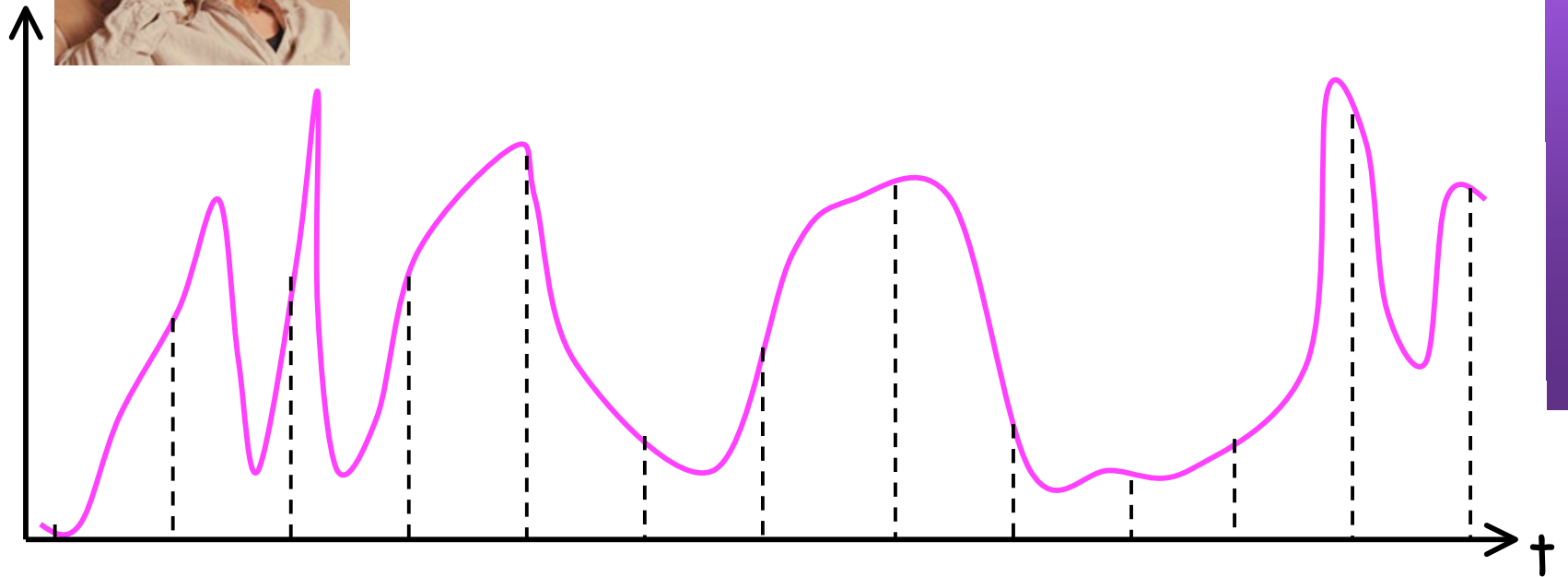
- a) Permitir cierto grado de bloqueo (pequeña probabilidad)
- b) Extender el número de etapas



# Voz digital



- 8.000 muestras por segundo
- $T_s = 125 \mu s$
- 8 bits/muestra



5 30 37 37 45 20 28 41 26 18 19 50 42



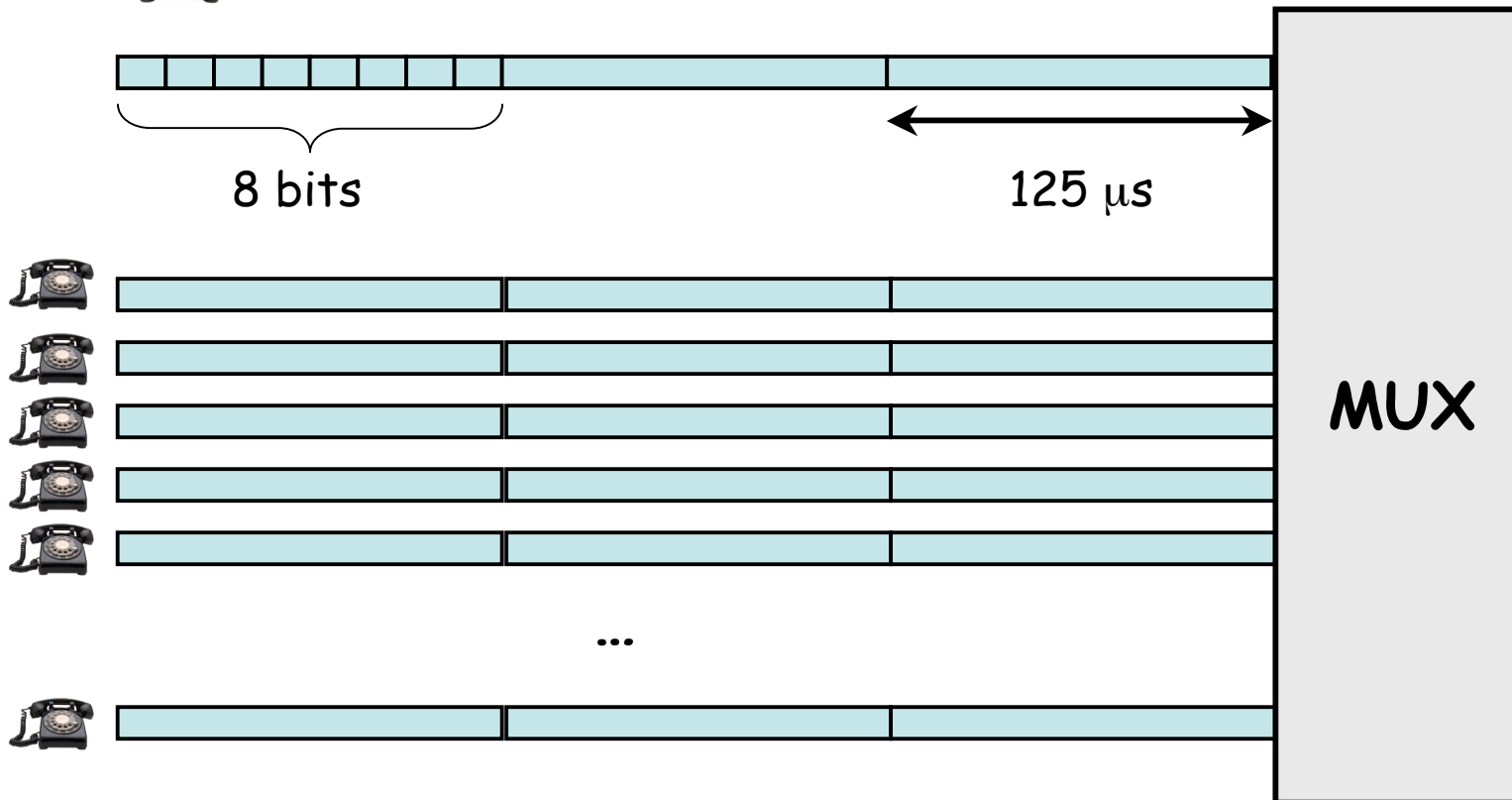
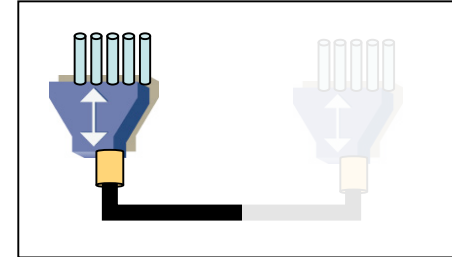
8 bits

125  $\mu s$

$\Rightarrow$  64Kbps

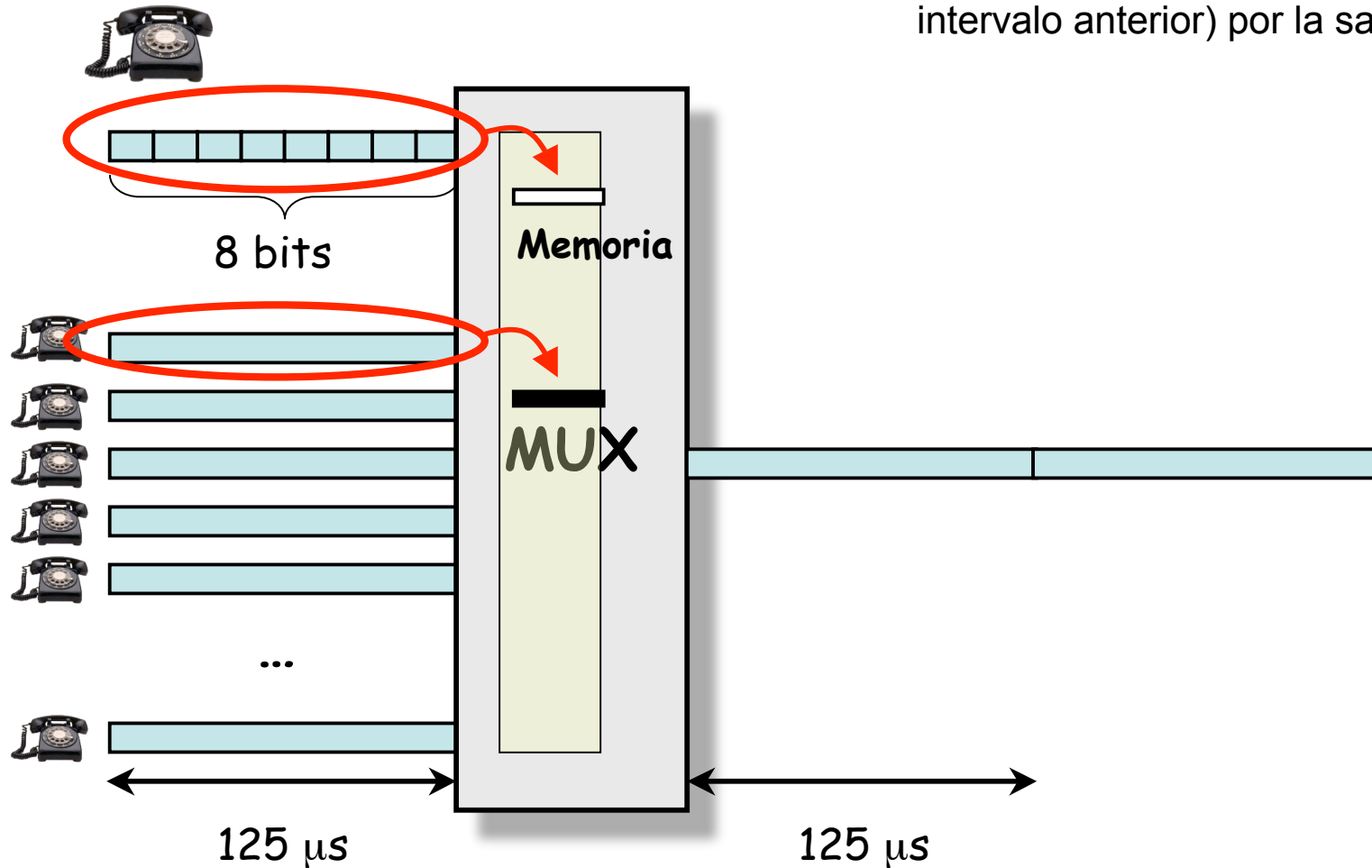
# Multiplexación TDM

- Las líneas troncales multiplexan los canales de voz en un mismo canal espacial



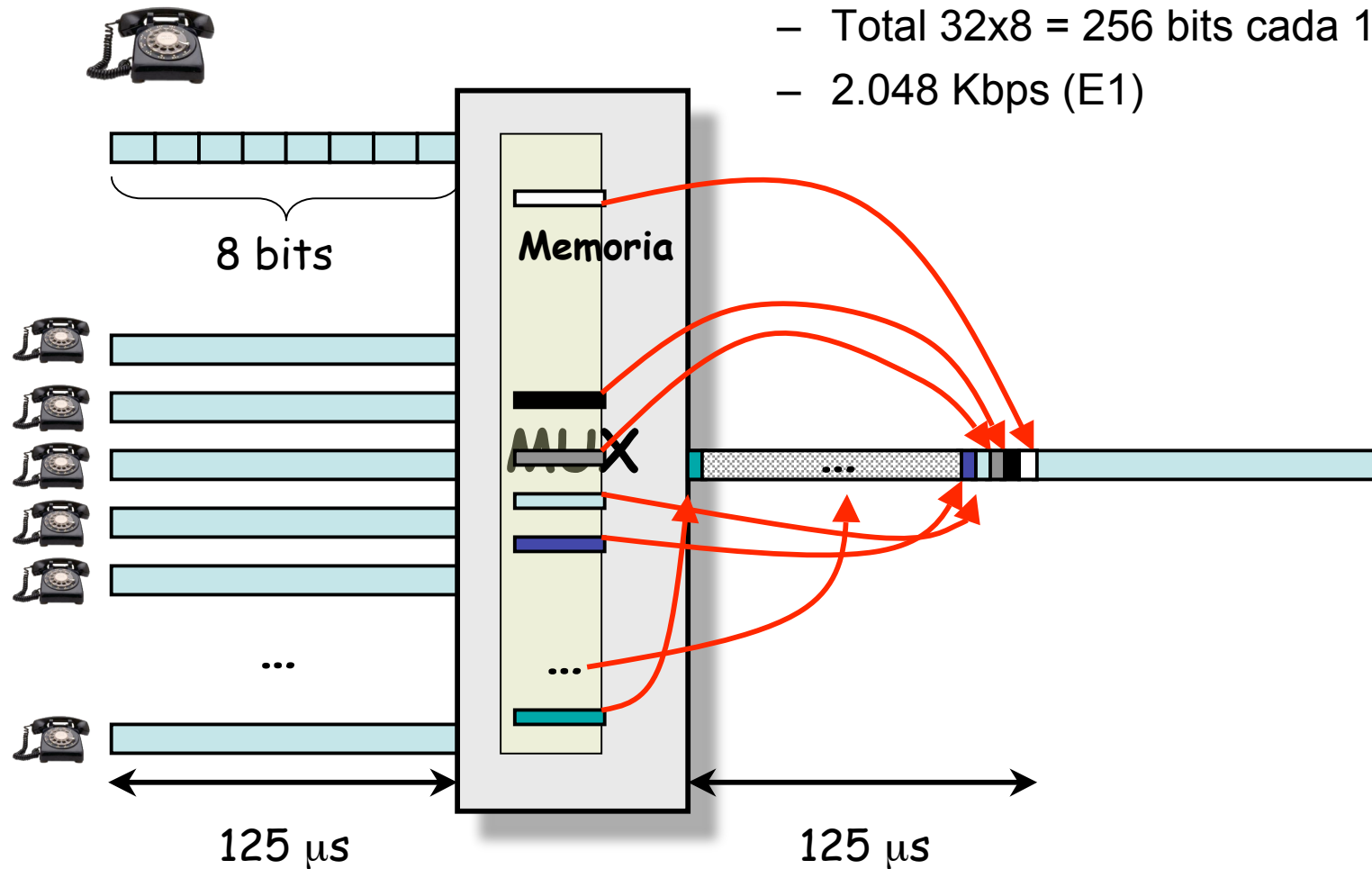
# Multiplexación TDM

- Las líneas troncales multiplexan los canales de voz en un mismo canal espacial
- En cada  $\partial t$  el MUX
  - Recibe una muestra de voz de cada una de las líneas
  - Envía N muestras de voz (del intervalo anterior) por la salida



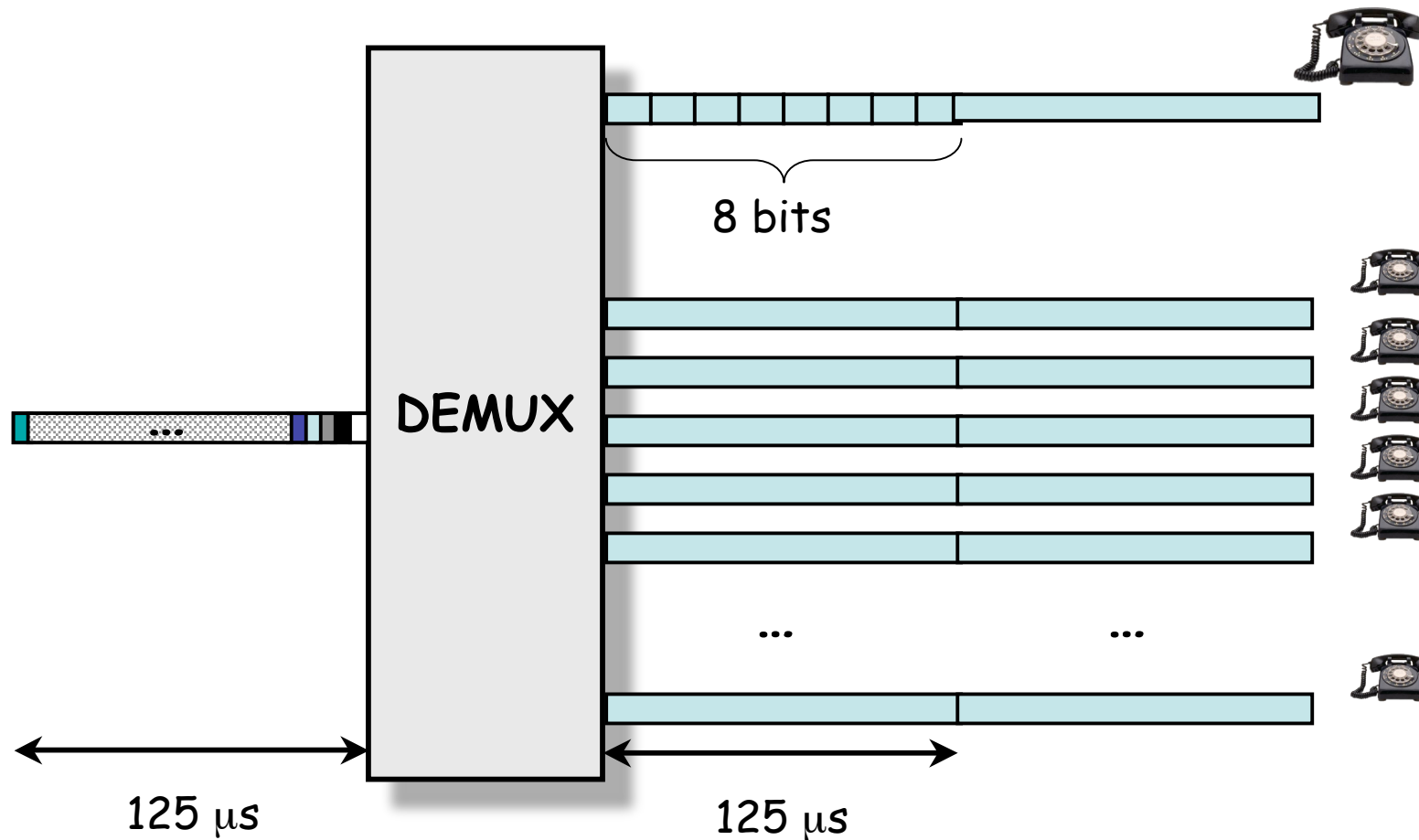
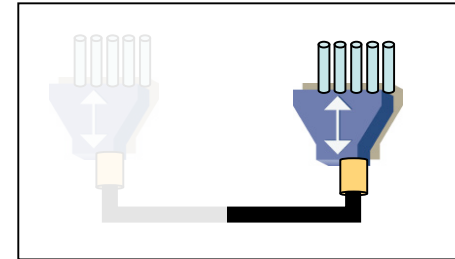
# Multiplexación TDM

- Ejemplo
  - 32 canales de voz
  - Cada canal 8bits cada  $125 \mu\text{s}$
  - Total  $32 \times 8 = 256$  bits cada  $125 \mu\text{s}$
  - 2.048 Kbps (E1)



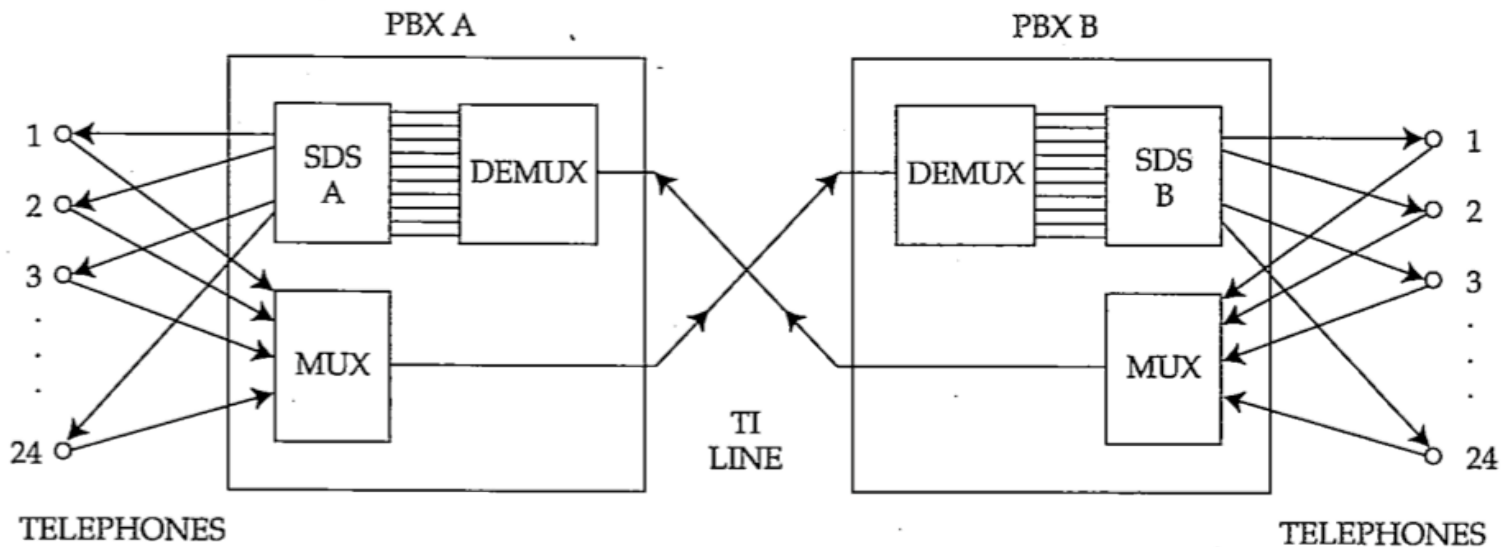
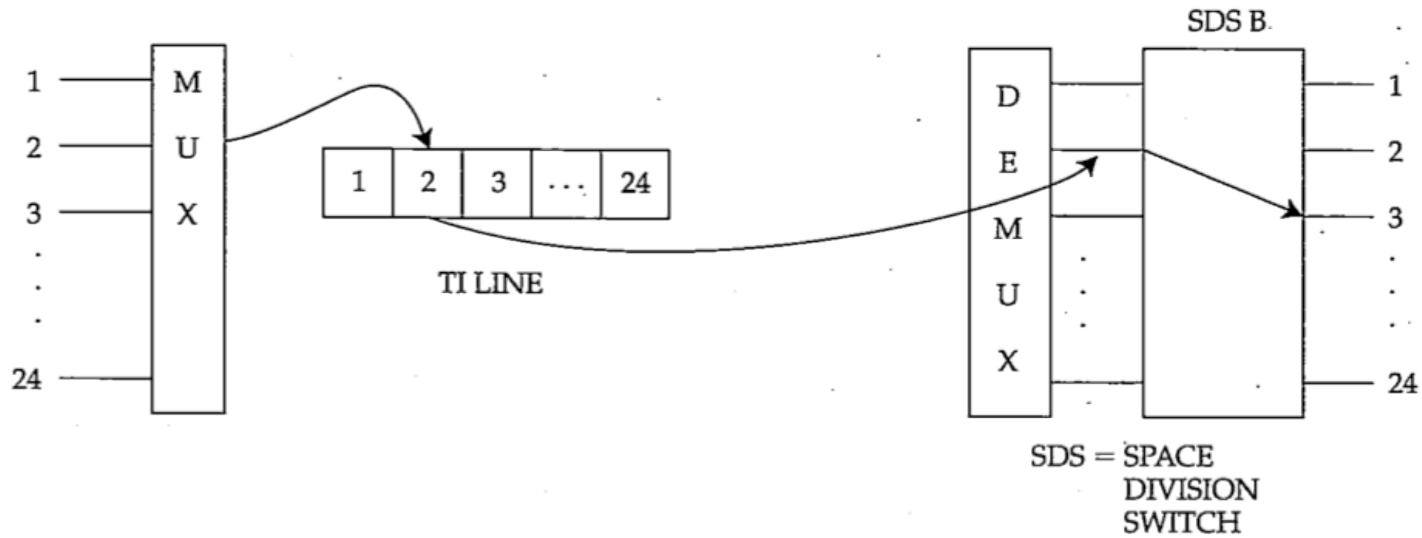
# Demultiplexación TDM

- Proceso inverso
- Una entrada
- N salidas de velocidad N veces menor





# Space Digital Switch (SDS)

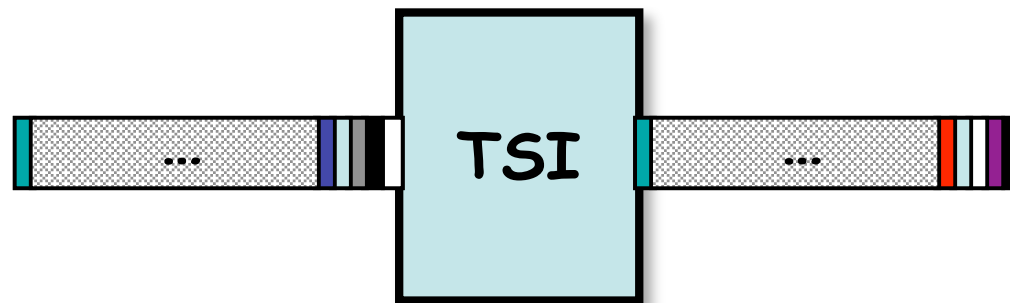


# Conmutación temporal

(Time-division Switching)

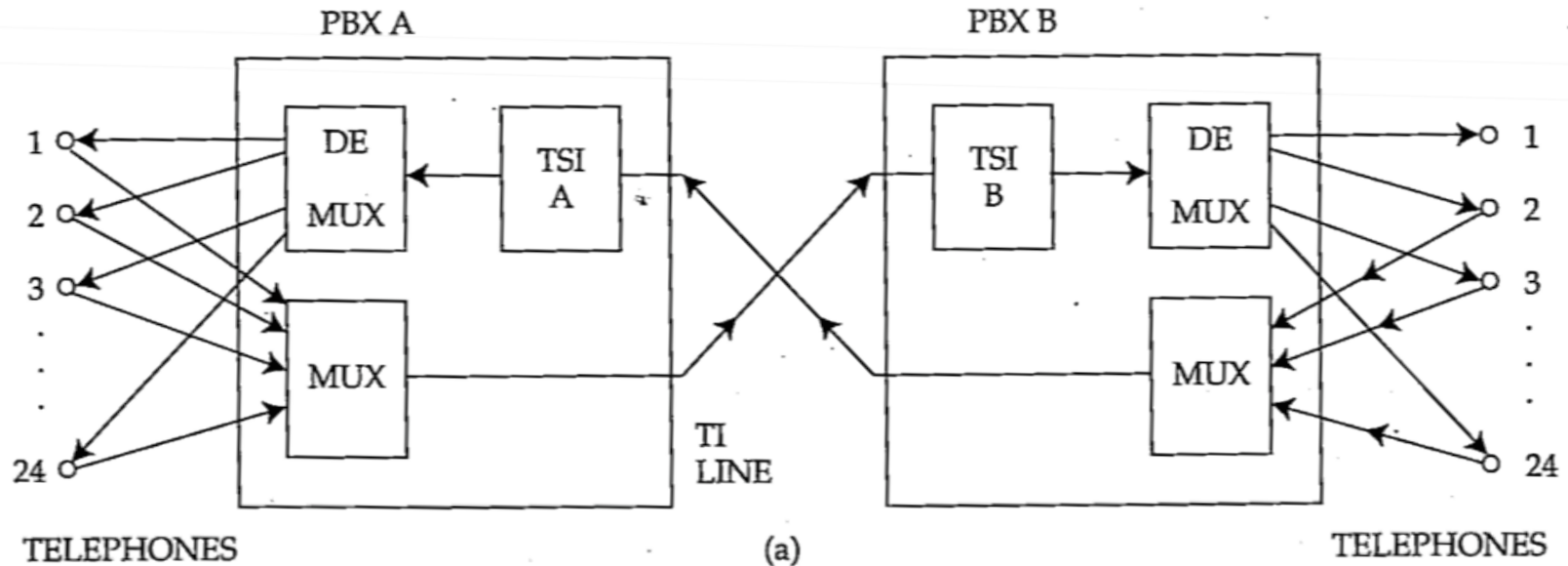
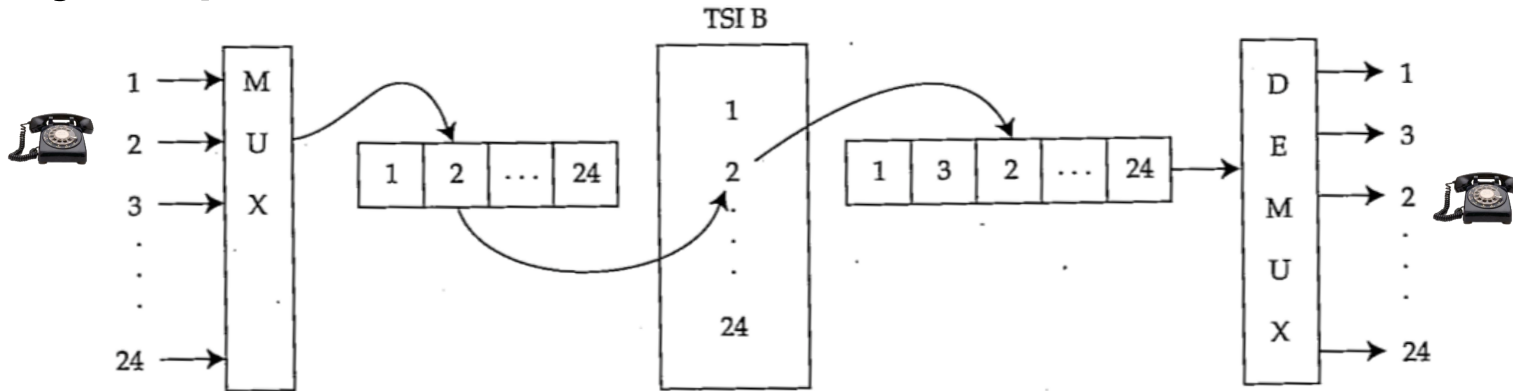
## TSI = Time Slot Interchanger

- Una entrada y una salida
- Ambas llevan N canales de voz multiplexados
- Guarda las muestras de entrada en un buffer de N bytes (una por circuito)
- Las reescribe en diferente orden
- Combinado con MUX/DEMUX el resultado es conmutación



# Time Slot Interchanger

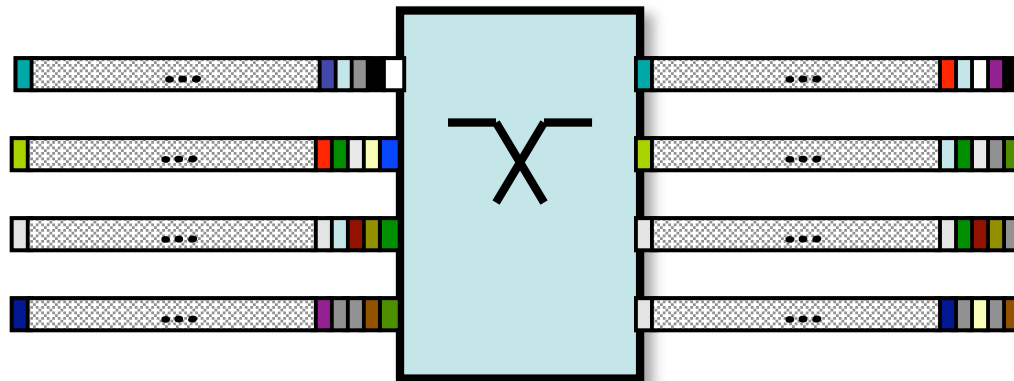
- Ejemplo



(a)

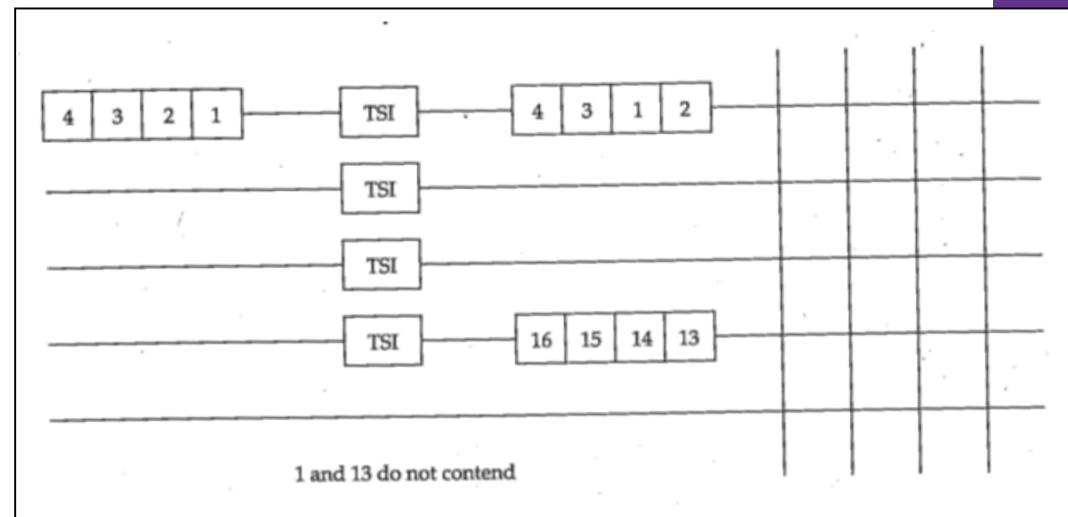
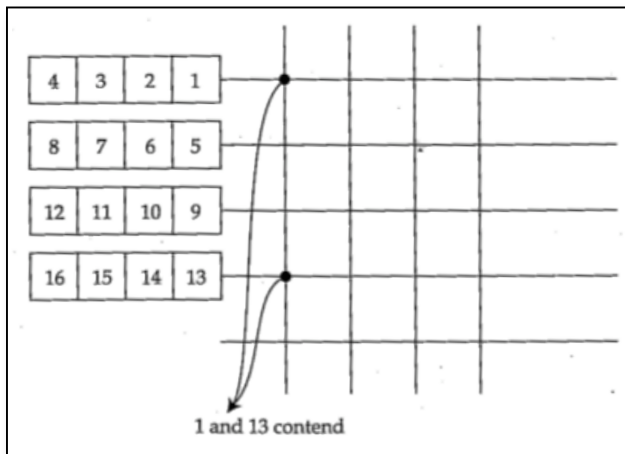
# Problemas construyendo un TSI

- El límite es el tiempo necesario para leer y escribir a memoria
- Para 120000 circuitos
  - Leer y escribir una muestra de cada uno una vez cada 125 microsegundos
  - Cada operación necesitaría hacerse en menos de 0.5 ns
- Otras técnicas...



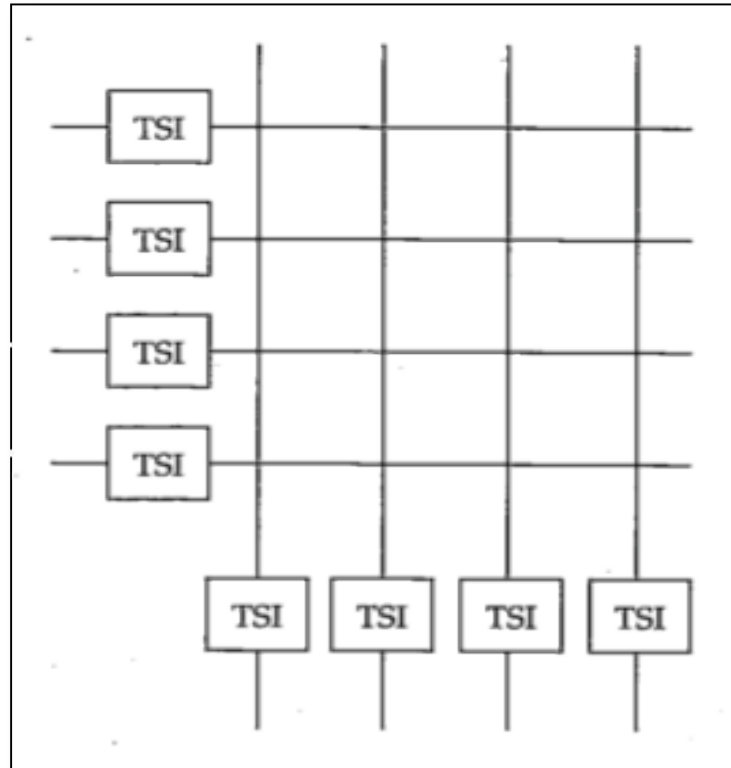
# Time-Space (TS) Switch

- Entradas multiplexadas
  - Crossbar con un TSI previo a cada entrada
  - Retrasar las muestras para que lleguen en el momento adecuado para la conmutación espacial
- Ejemplo: Sin TSIs
  - Ejemplo: Con TSIs



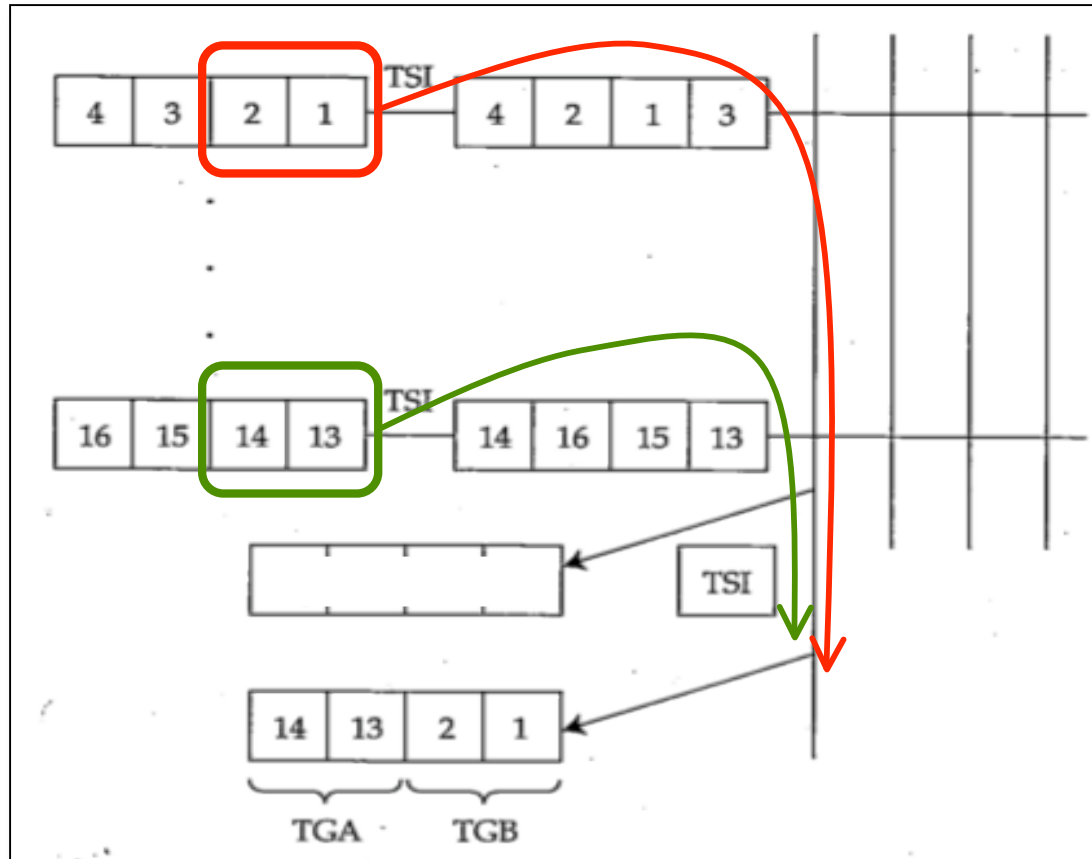
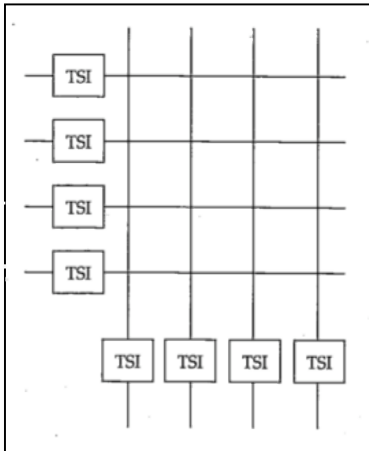
# Time-Space-Time (TST) Switch

- Similar a un conmutador espacial de 3 etapas
- Las etapas de entrada y de salida son TSIs
- Reordenar en entrada para evitar bloqueo en crossbar
- Reordenar en salida para asignar a slot correcto
- Ejemplo:



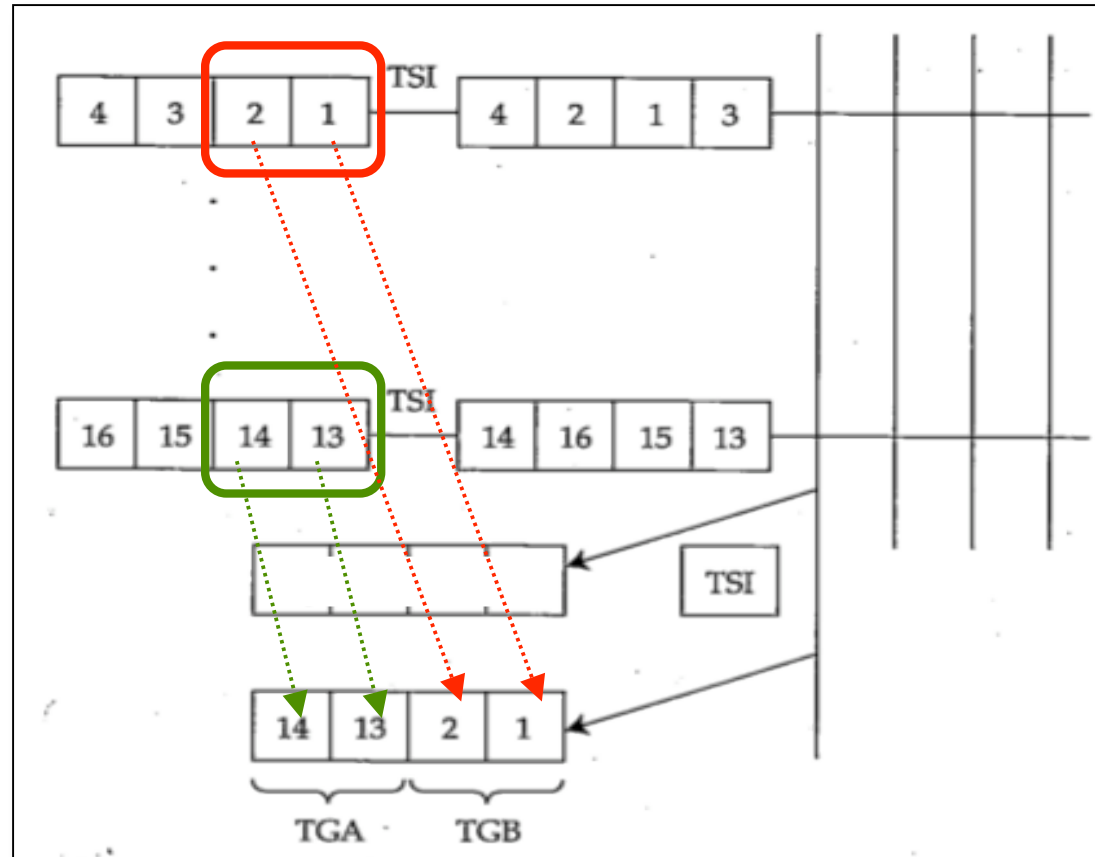
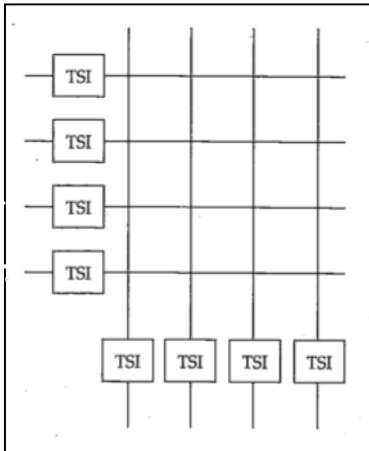
# Time-Space-Time (TST) Switch

- Slots 1 y 2 de entrada 1 y slots 1 y 2 (valores 13 y 14) de entrada 4 van a la salida 1



# Time-Space-Time (TST) Switch

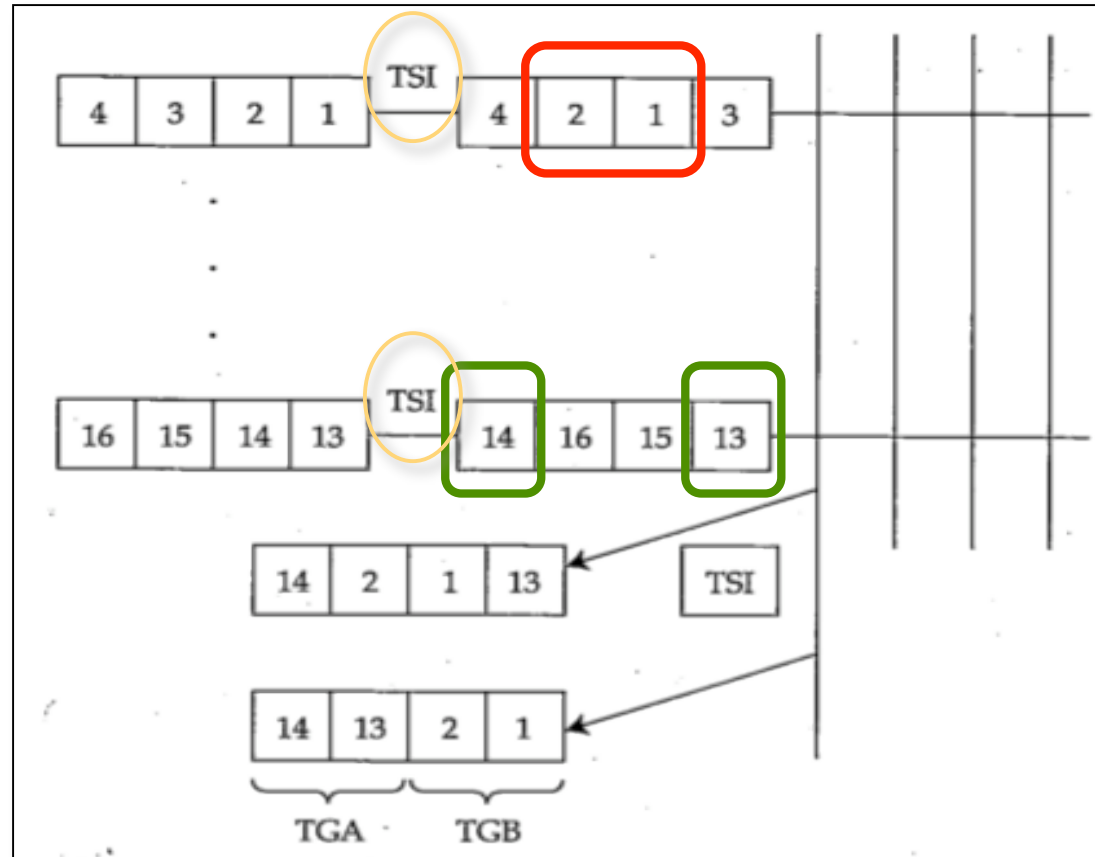
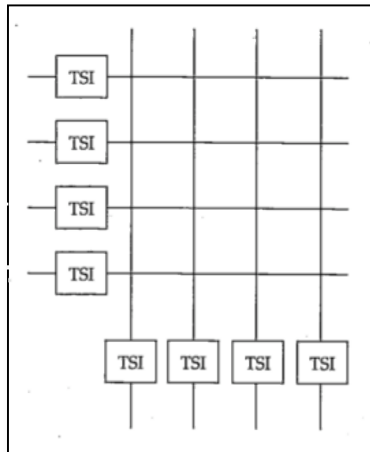
- Slots 1 y 2 de entrada 1 y slots 1 y 2 (valores 13 y 14) de entrada 4 van a la salida 1
- En la salida slot 1 de entrada 1 va a slot 4, slot 2 a slot 3
- En la salida slot 1 de entrada 4 va a slot 2 y slot 2 a slot 1





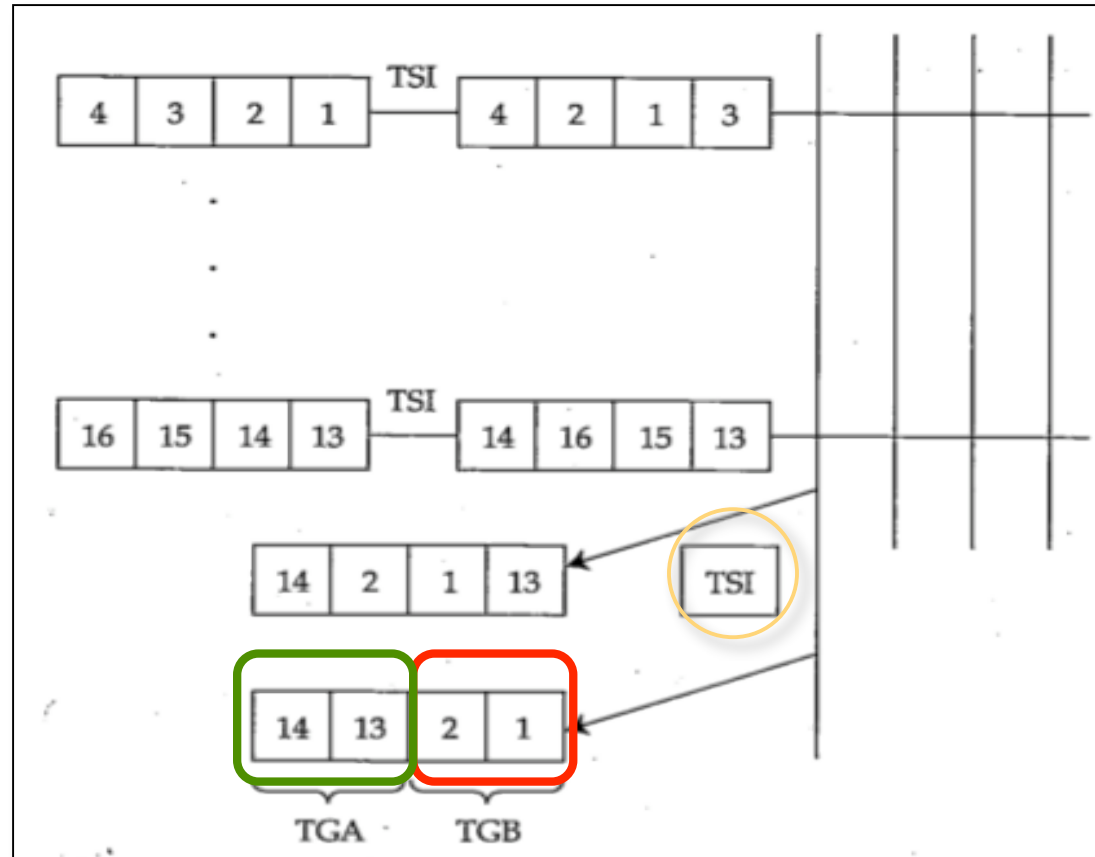
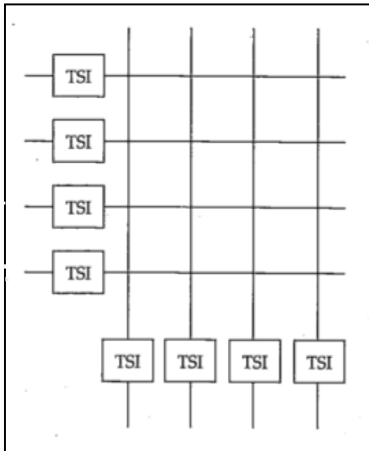
# Time-Space-Time (TST) Switch

- TSIs en la entrada reordenan para que no haya bloqueo
- Pueden colocar los slots en la posición que quieran que no produzca bloqueo
- Mayor flexibilidad para usar el conmutador espacial y evitar bloqueo



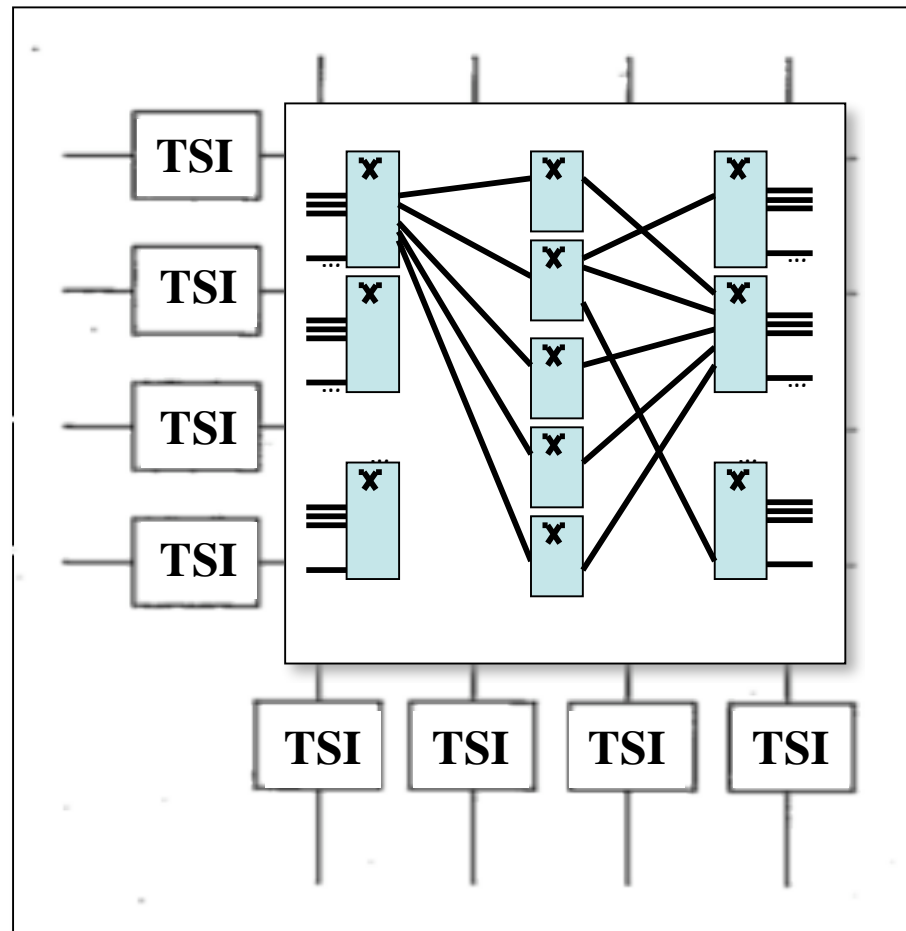
# Time-Space-Time (TST) Switch

- TSIs en la entrada reordenan para que no haya bloqueo
- Pueden colocar los slots en la posición que quieran que no produzca bloqueo
- Mayor flexibilidad para usar el conmutador espacial y evitar bloqueo
- TSI en la salida reordena



# Time-Space-Time (TST) Switch

- La etapa espacial puede sustituirse por un conmutador de 3 etapas  $\Rightarrow$  TSSST



# Conclusiones

- Conmutación de circuitos
  - Establecer caminos *físicos* para conectar dos terminales
  - Señalización para control entre nodos y entre usuarios y nodos
- Conmutadores
  - Bloqueo
  - *Crossbar* y conmutadores multietapa
  - Redes sin bloqueo: condición de Clos
  - Conmutadores espaciales y temporales
  - S, T, TST

## **Próxima clase:**

- Prestaciones