



**UNIVERSIDAD
DE BURGOS**

TEMA 2

Arquitecturas multiprocesador: MIMD de memoria distribuida (multicomputadores)

José M. Cámara (checam@ubu.es)

v. 2.0



Introducción

- Máquinas de hardware débilmente acoplado.
- Existen diferentes niveles de acoplamiento en esta categoría en función del hardware compartido: almacenamiento, entrada/salida...
- Al no disponer de espacio común en memoria, la comunicación se va a llevar a cabo mediante paso de mensajes.
 - Mensaje: unidad de intercambio de información.
 - Paquete: unidad de conmutación.



Conmutación

- Conmutación de circuitos: se establece una conexión permanente para la transmisión del mensaje completo.
- Conmutación de paquetes (control de flujo):
 - Sin división: $T = (N-1)(L/W)$ store & forward
 - Con división en flits: $T = (L/W) + (F/W)(N-2)$ wormhole
 - Con división en phits: $T = L/W + (P/W)(N-2)$ virtual cut through

Paquetes: disponen de información de destino y nº de orden dentro del mensaje.

Flits (flow control unit): no disponen de ninguna información salvo el de cabecera, por lo que todos deben seguirle.

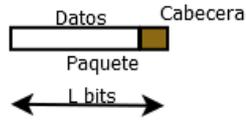
Phits: unidad mínima de información posible; equivale al nº de líneas de datos del enlace.

El pipeline en VCT es análogo a Wormhole pero requiere espacio de almacenamiento intermedio para paquetes completos, como S&F.

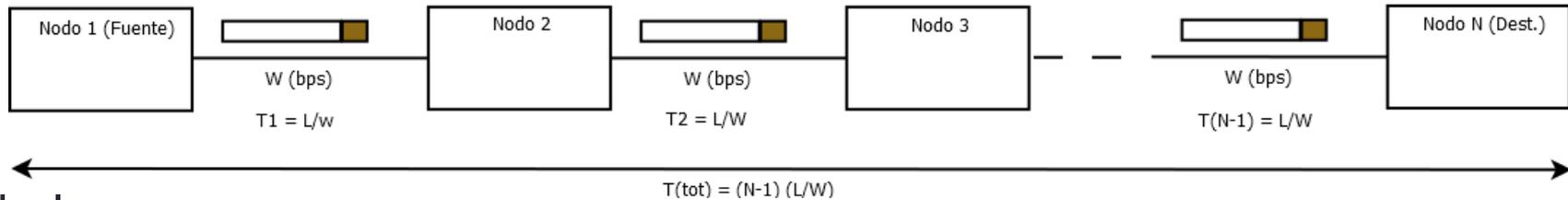


Estrategias de conmutación

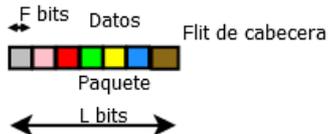
Store & forward



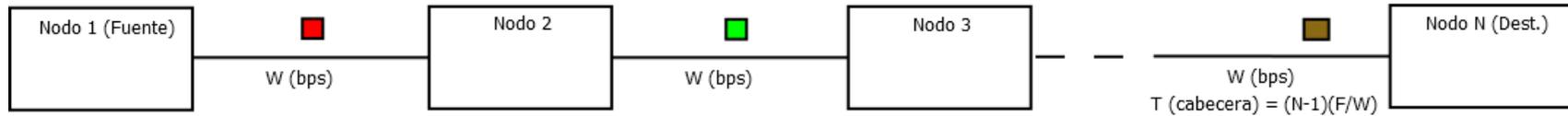
Los nodos almacenan los paquetes en buffers locales antes de su reenvío al siguiente nodo.



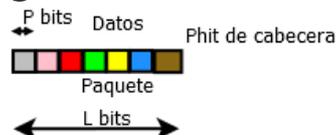
Wormhole



El flit de cabecera realiza todo el recorrido. A partir de su llegada, cada F/W segundos llega un flit.



Virtual cut through



El phit de cabecera realiza todo el recorrido. A partir de su llegada, cada P/W segundos llega un phit.



$T (Tot) = (N-2)(P/W) + (L/W) = \text{phit de cabecera hasta el penúltimo nodo} + 1 \text{ solo salto del paquete completo.}$



Conmutación de paquetes

■ Aspectos relevantes:

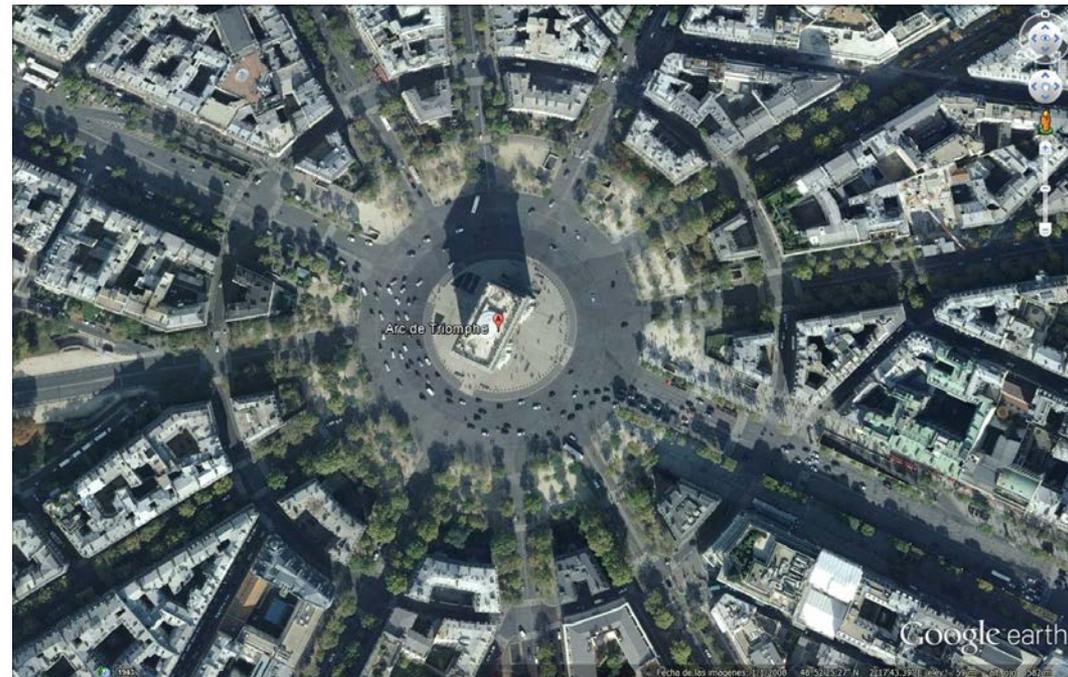
- Control de flujo: cómo y cuando se mueven los paquetes por la red. Incluye:
 - Resolución de colisiones: entre paquetes entrantes cuando los recursos de almacenamiento son escasos.
 - Encaminamiento: cómo hacer que los paquetes lleguen a su destino.
- Bloqueo (deadlock): los paquetes se bloquean formando un bucle -> la red se colapsa.
- Livelock: un paquete se mueve por la red sin llegar nunca a destino.
- Inanición (starvation): un paquete no recibe nunca servicio y queda anclado en un elemento de almacenamiento.

Abrazo mortal e inanición

Abrazo mortal de tráfico



http://fedoraproject.org/wiki/FAD_SP_2013



Situación potencial de inanición
Los coches procedentes de las calles en la mitad inferior podrían no conseguir acceder a la rotonda.

Conmutación de paquetes

■ Soluciones:

■ Bloqueo

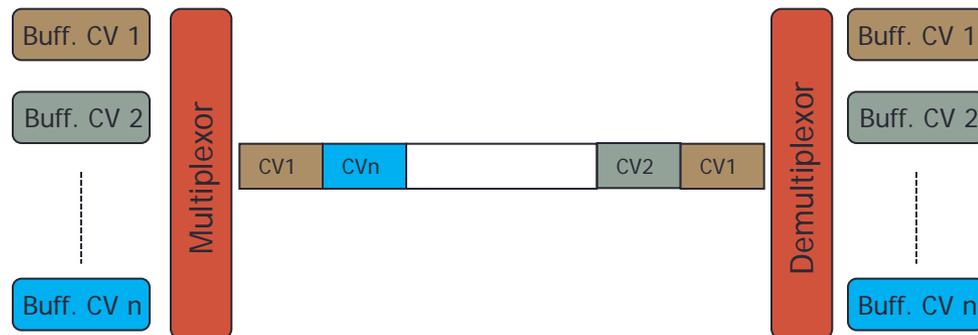
- Canales virtuales para demorar la aparición del bloqueo
- Ciertas combinaciones de interconexión – algoritmo de encaminamiento están libres de bloqueo.
- Recuperación: si el bloqueo aparece, el sistema puede descartar un paquete para deshacerlo. Debe notificarlo al emisor para que pueda reintentar.

■ Resolución de colisiones

- Descartar
- Almacenar
- Reenviar -> livelock
- Bloquear

■ Encaminamiento

- Algoritmos de encaminamiento



Canales virtuales multiplexados en el tiempo



[Contenido multimedia disponible](#)



Algoritmos de encaminamiento

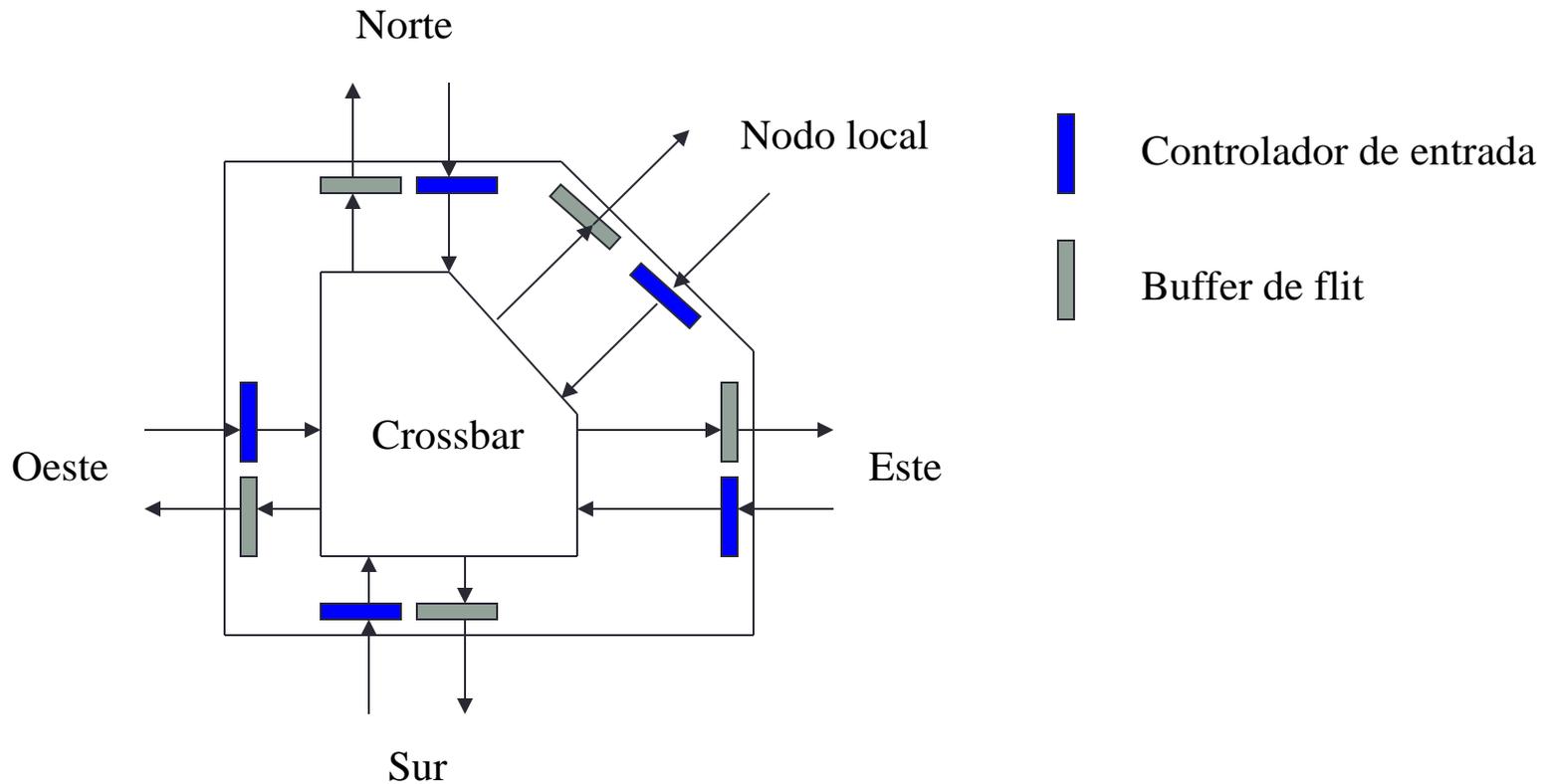
- Criterios de clasificación:
 - Número de destino: unicast, multicast
 - Lugar donde se toman las decisiones: centralizados, fuente, distribuidos
 - Forma de implementación: tabla, máquina de estados
 - Adaptatividad: deterministas, adaptativos
 - N° de caminos alternativos: parcialmente adaptativos, totalmente adaptativos
 - Progresividad: progresivos, regresivos
 - Minimalidad: mínimos, no mínimos -> livelock



CONMUTACIÓN DE PAQUETES

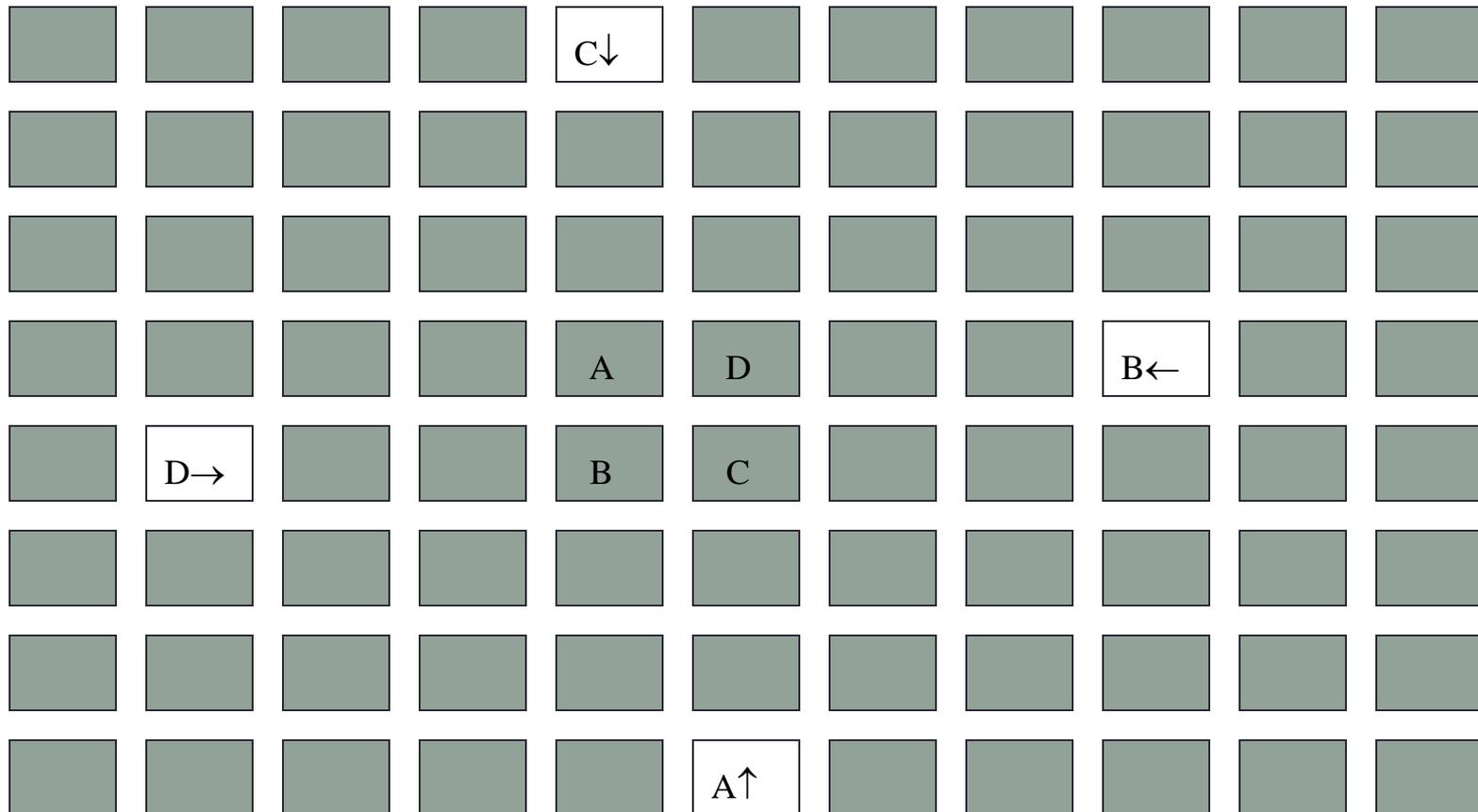
ENCAMINAMIENTO

ROUTER GENÉRICO (Intel Paragon)



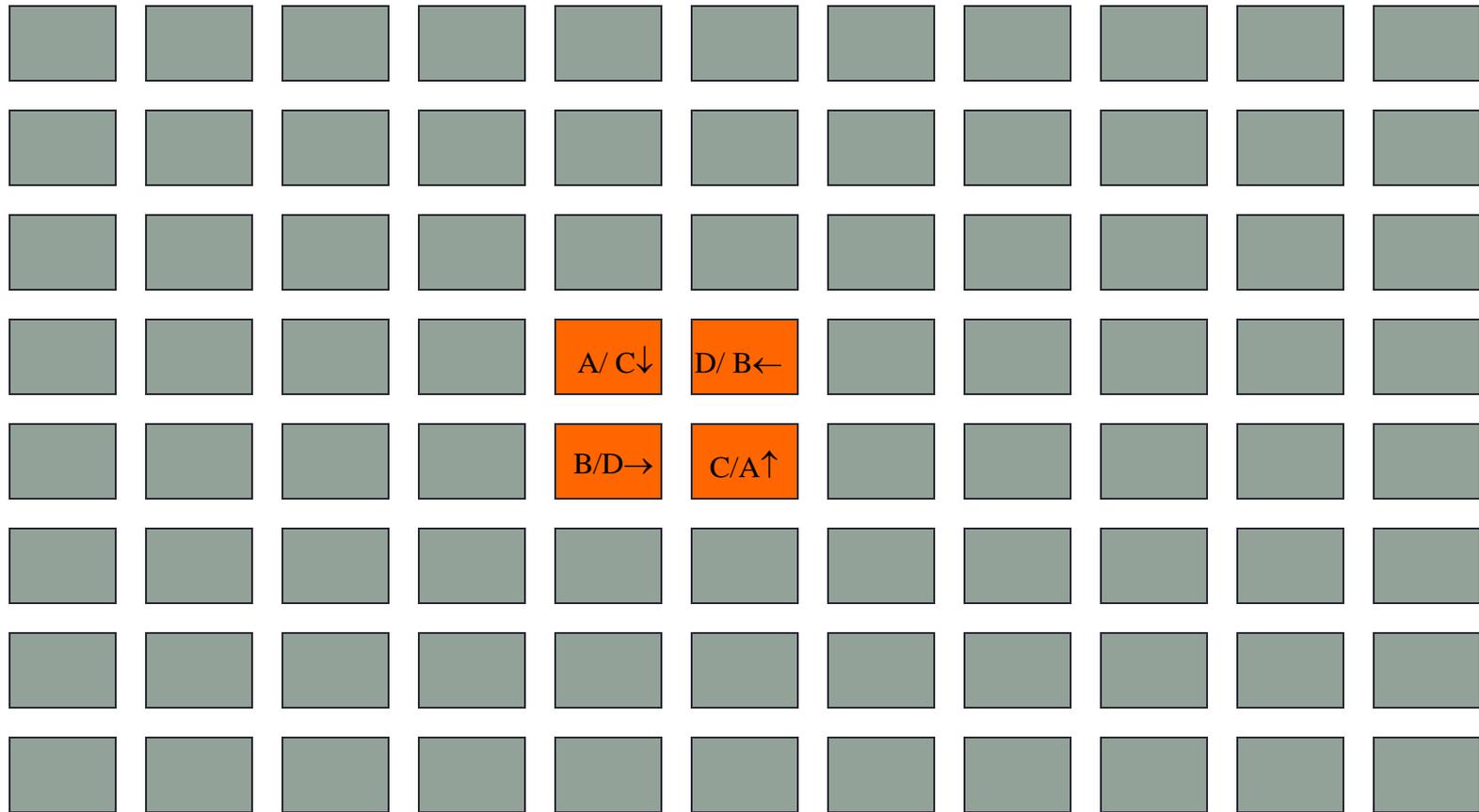


SITUACIÓN DE ABRAZO MORTAL I



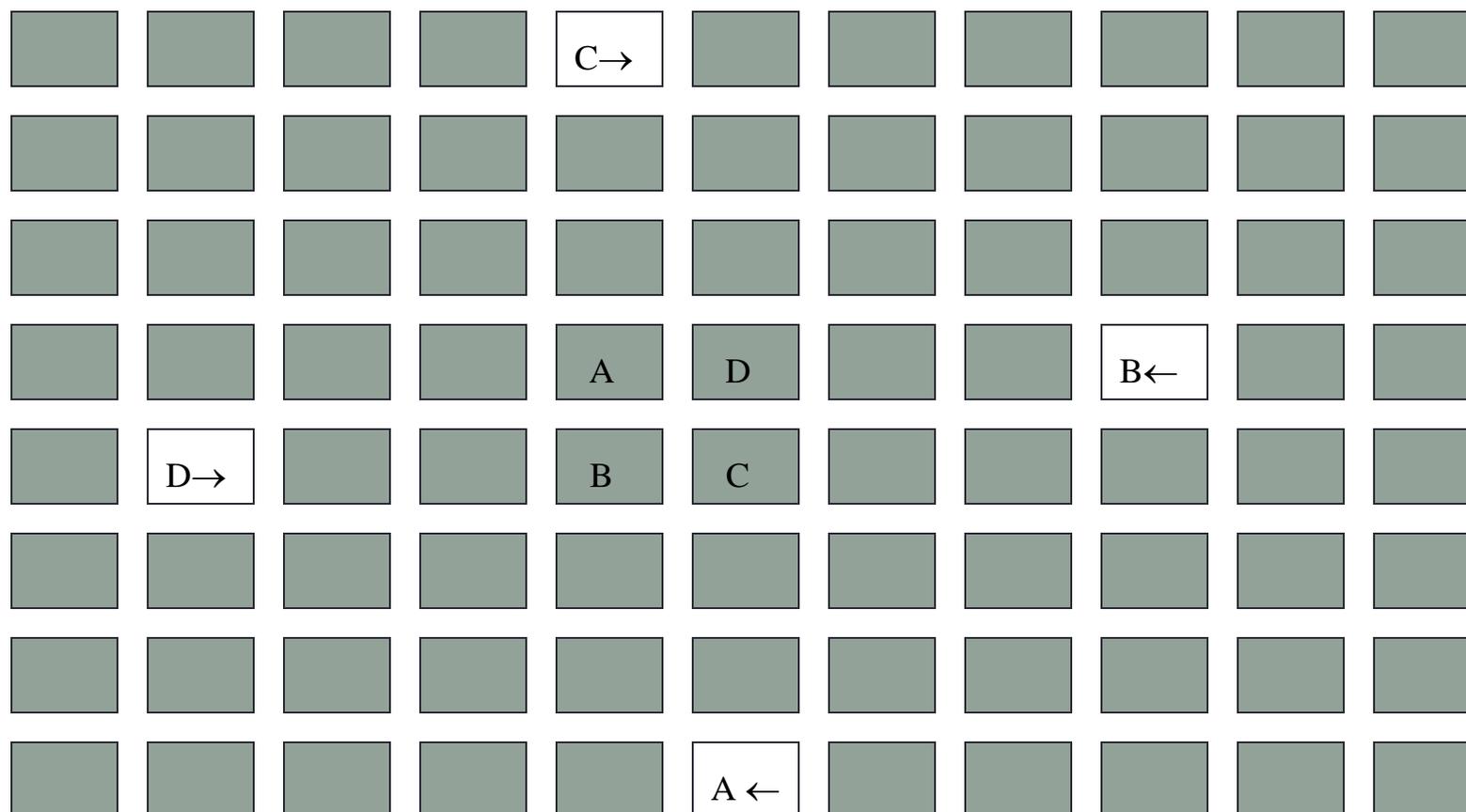


SITUACIÓN DE ABRAZO MORTAL IV



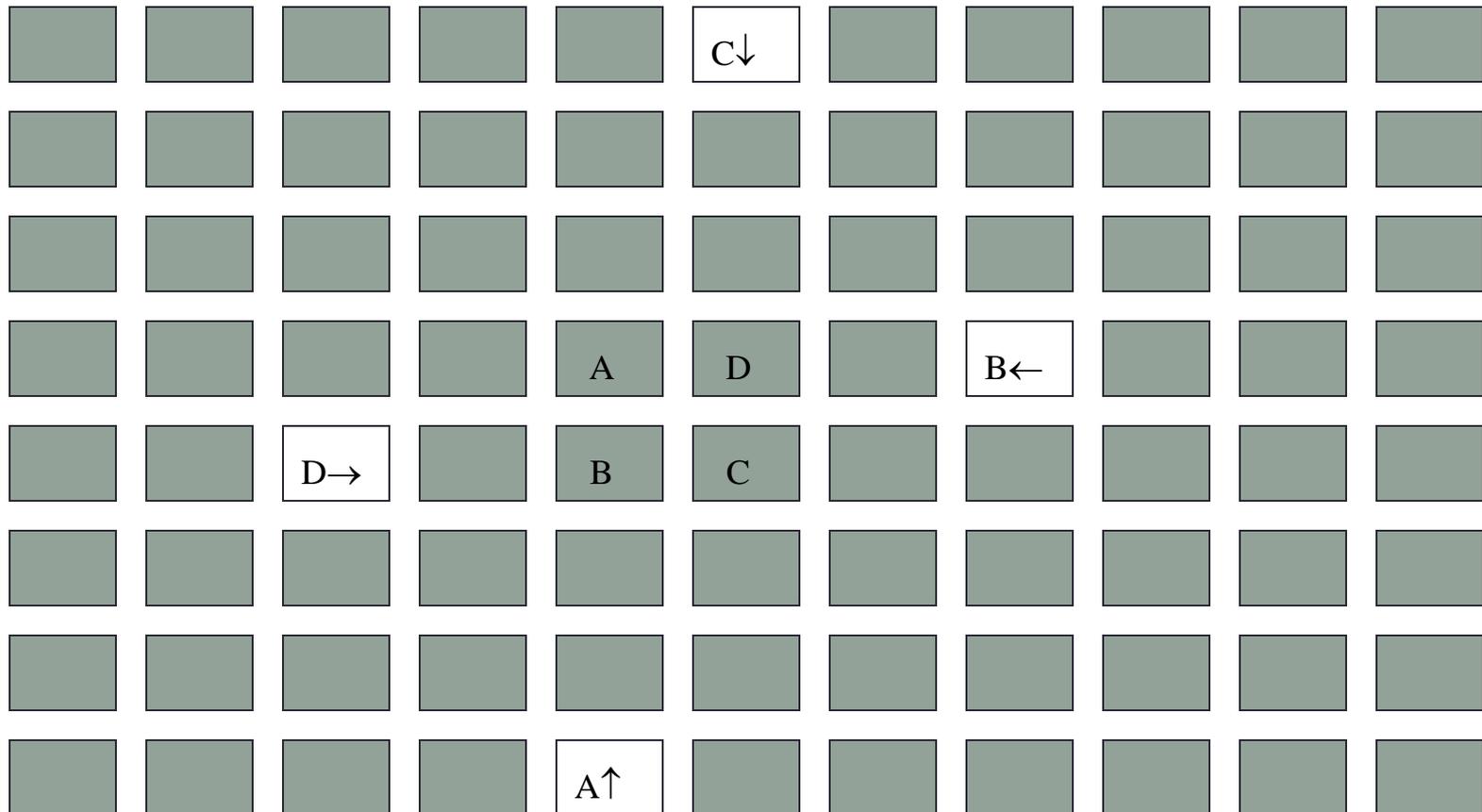


ENCAMINAMIENTO DOR I

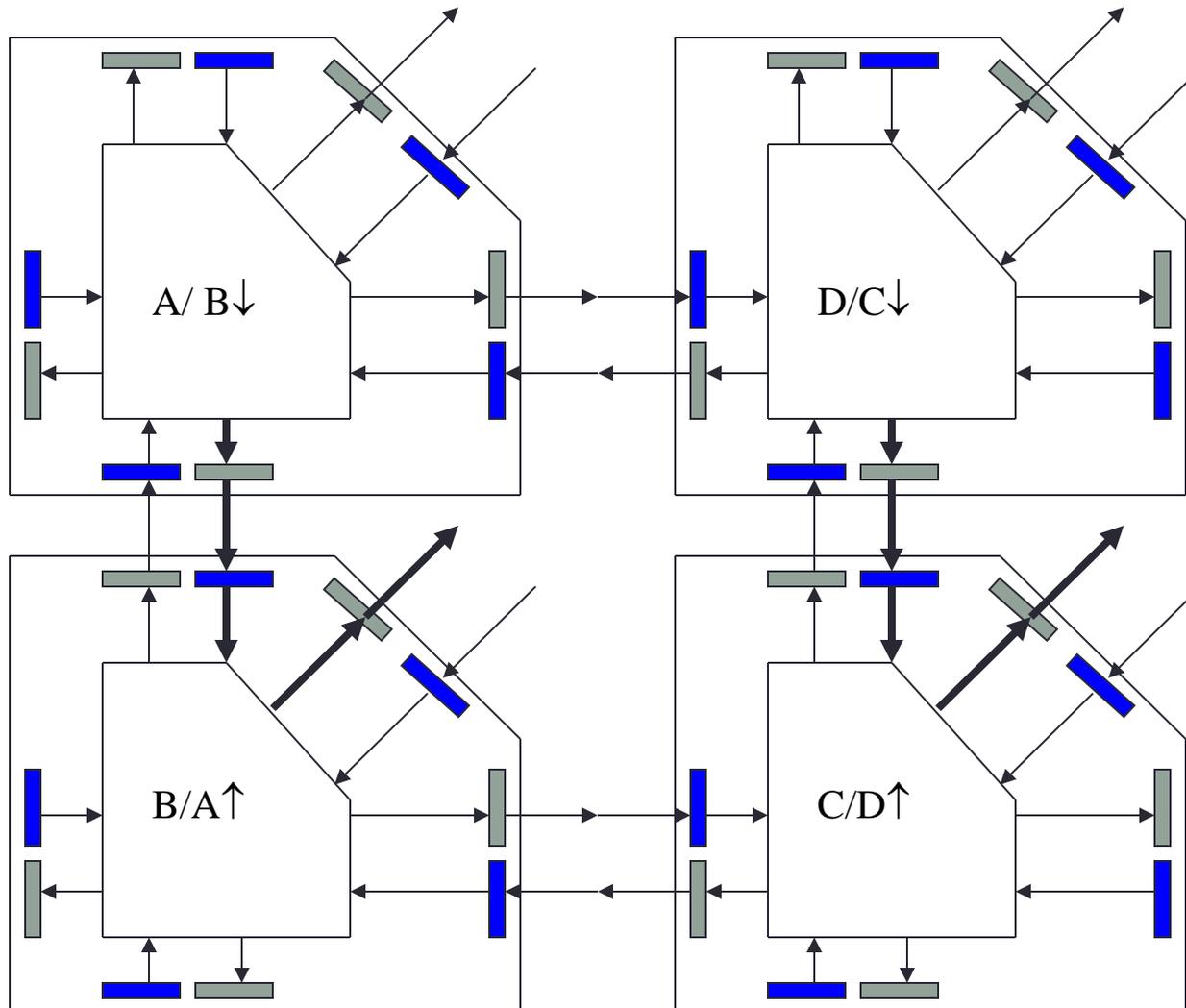




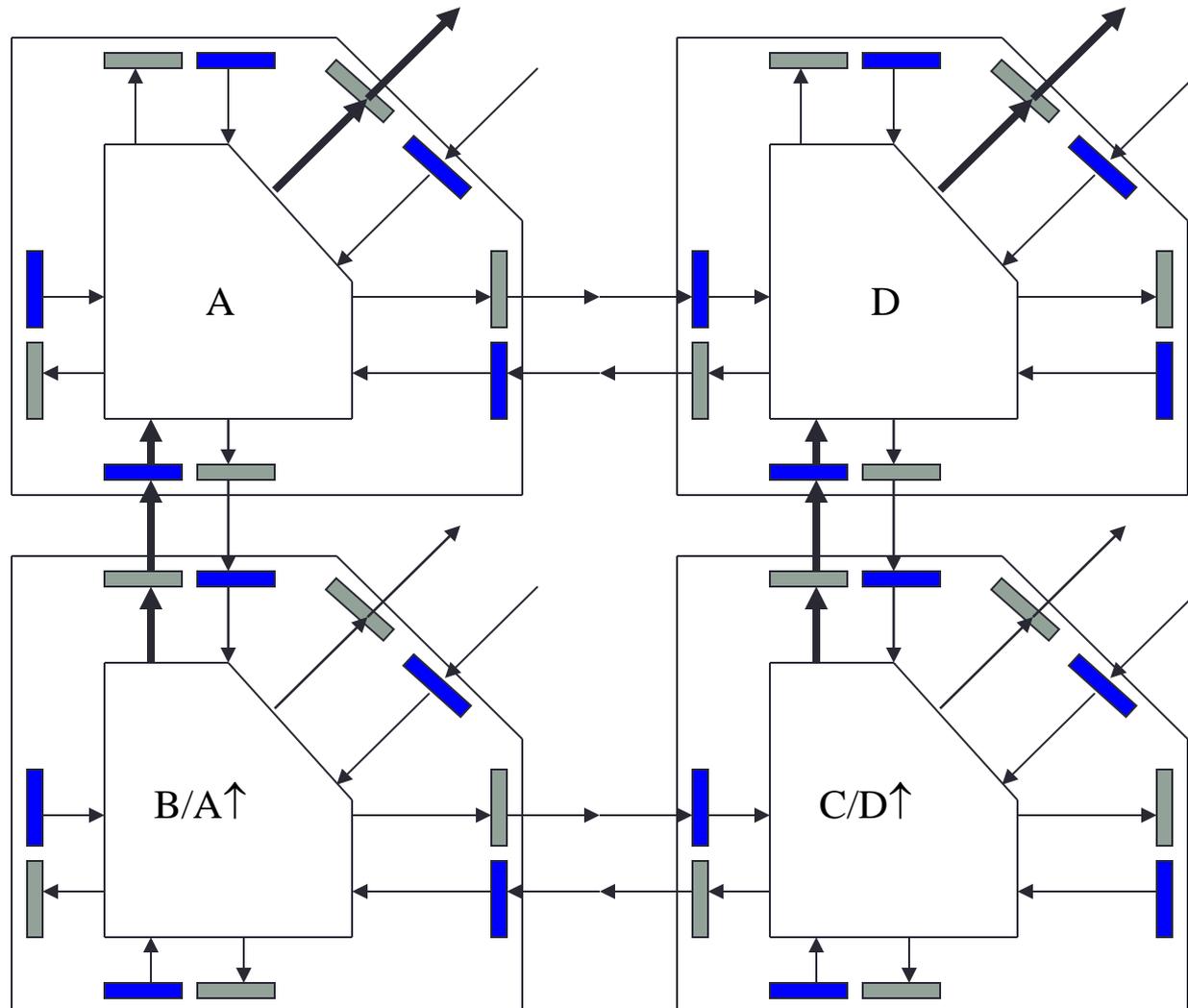
ENCAMINAMIENTO DOR II



ENCAMINAMIENTO DOR V

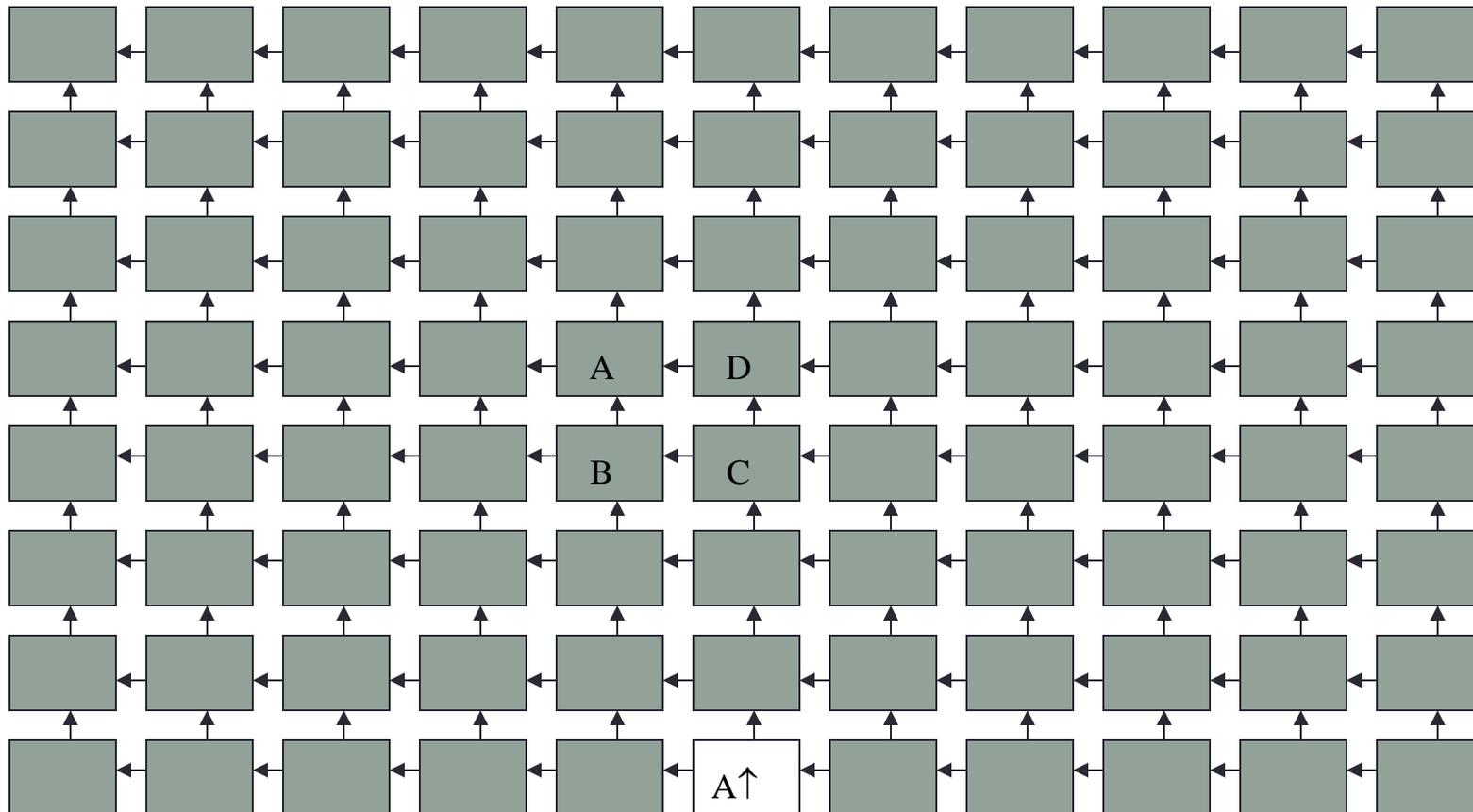


ENCAMINAMIENTO DOR VI





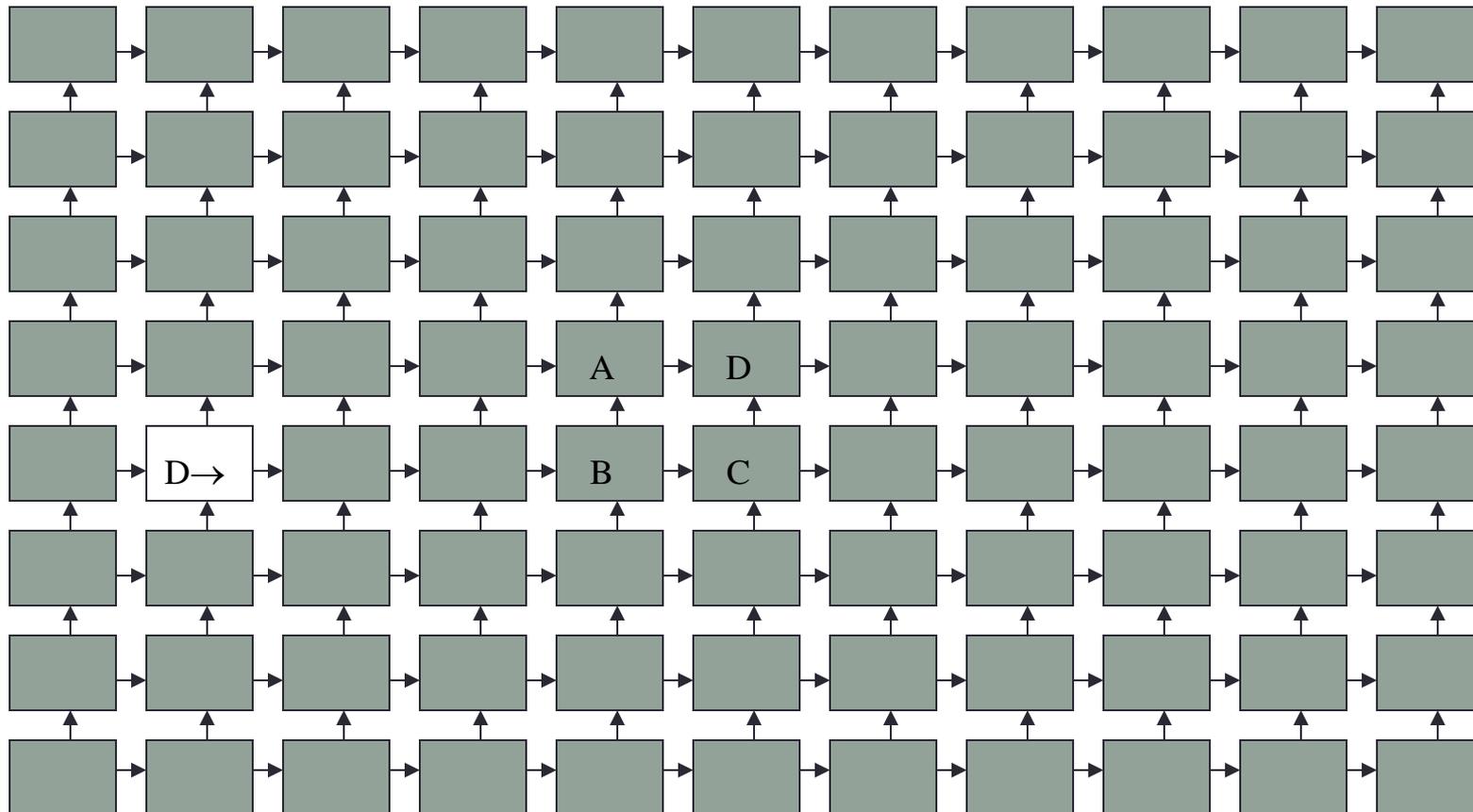
REDES VIRTUALES I (noroeste)





REDES VIRTUALES I

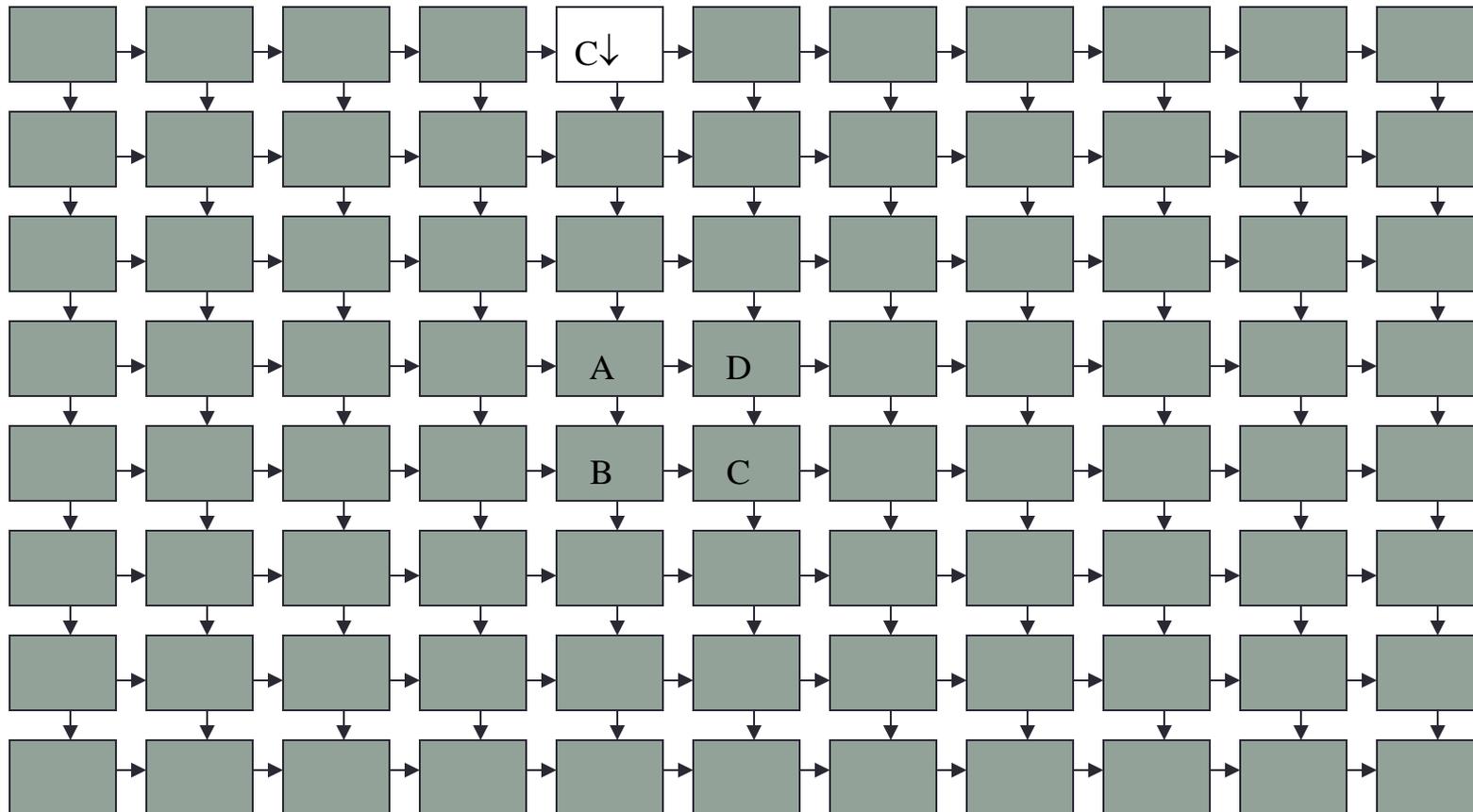
(noreste)





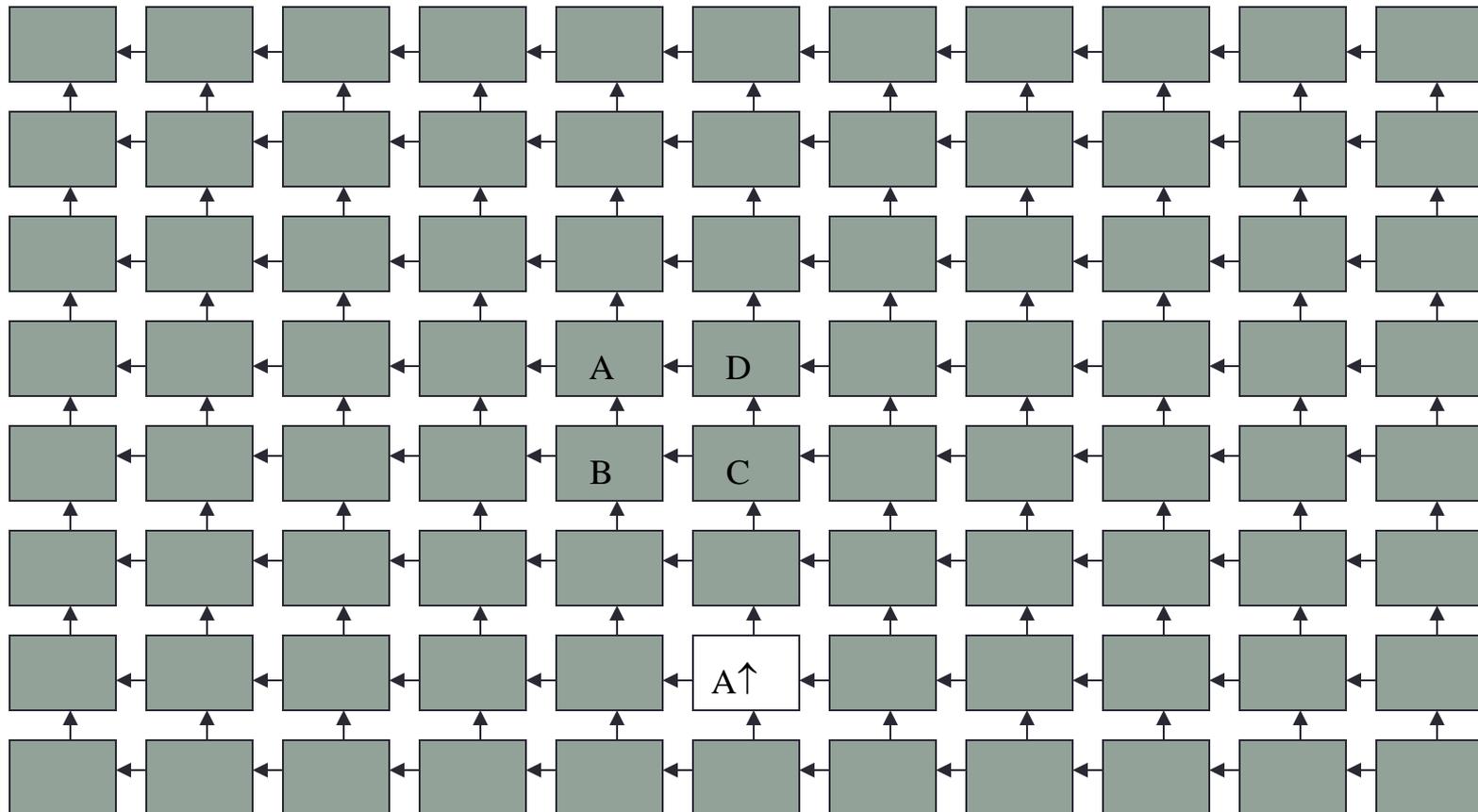
REDES VIRTUALES I

(sureste)



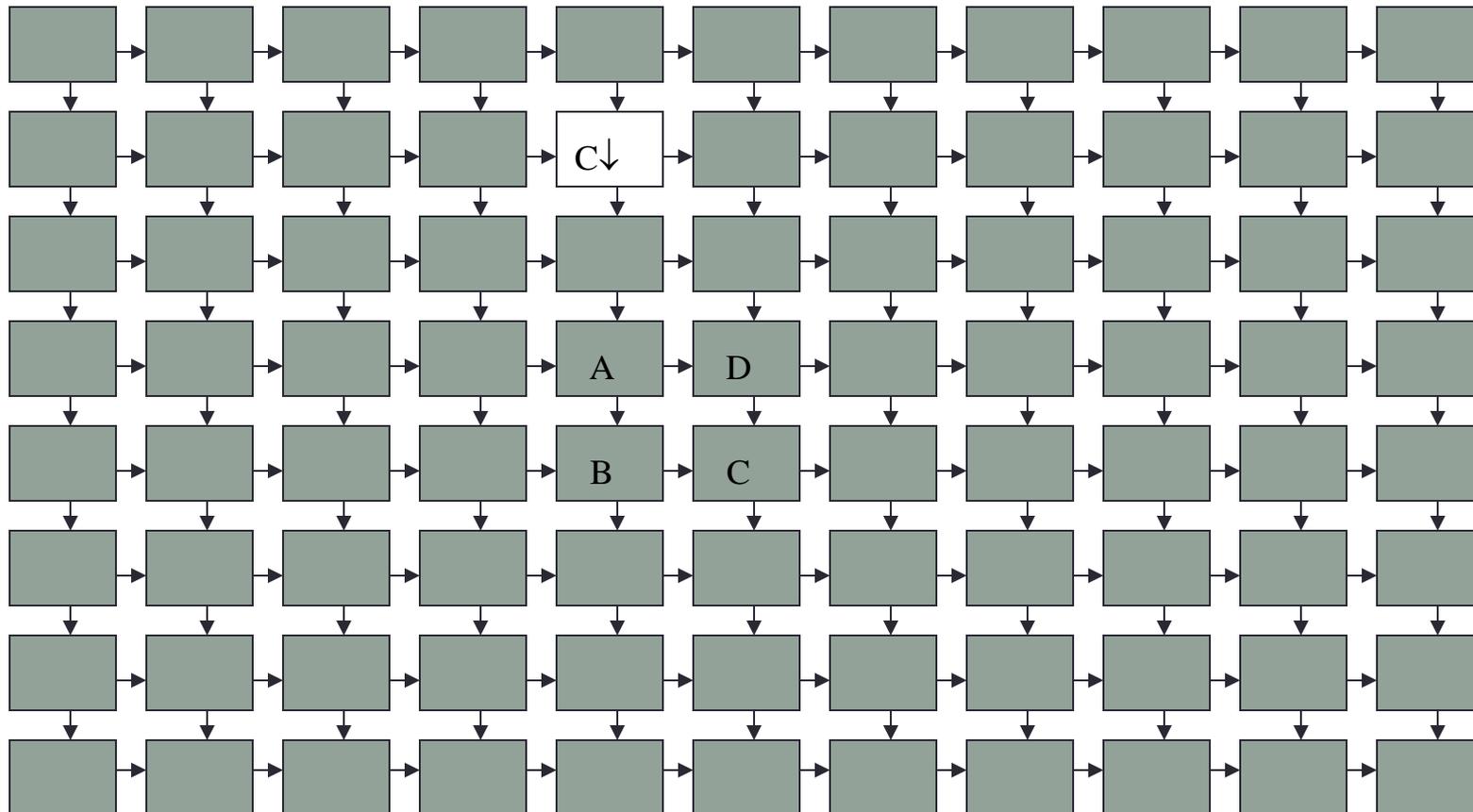
REDES VIRTUALES II

(noroeste)



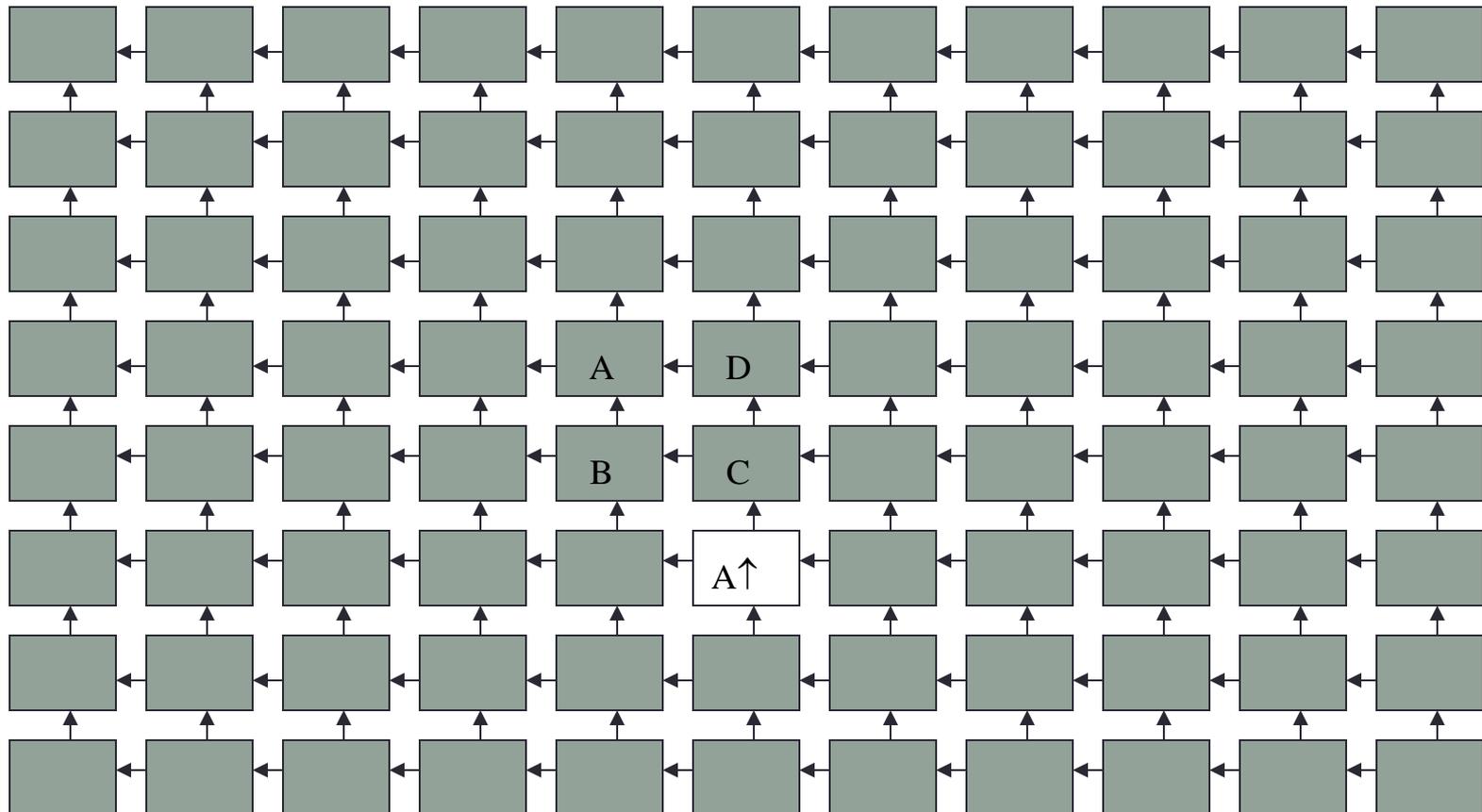
REDES VIRTUALES II

(sureste)



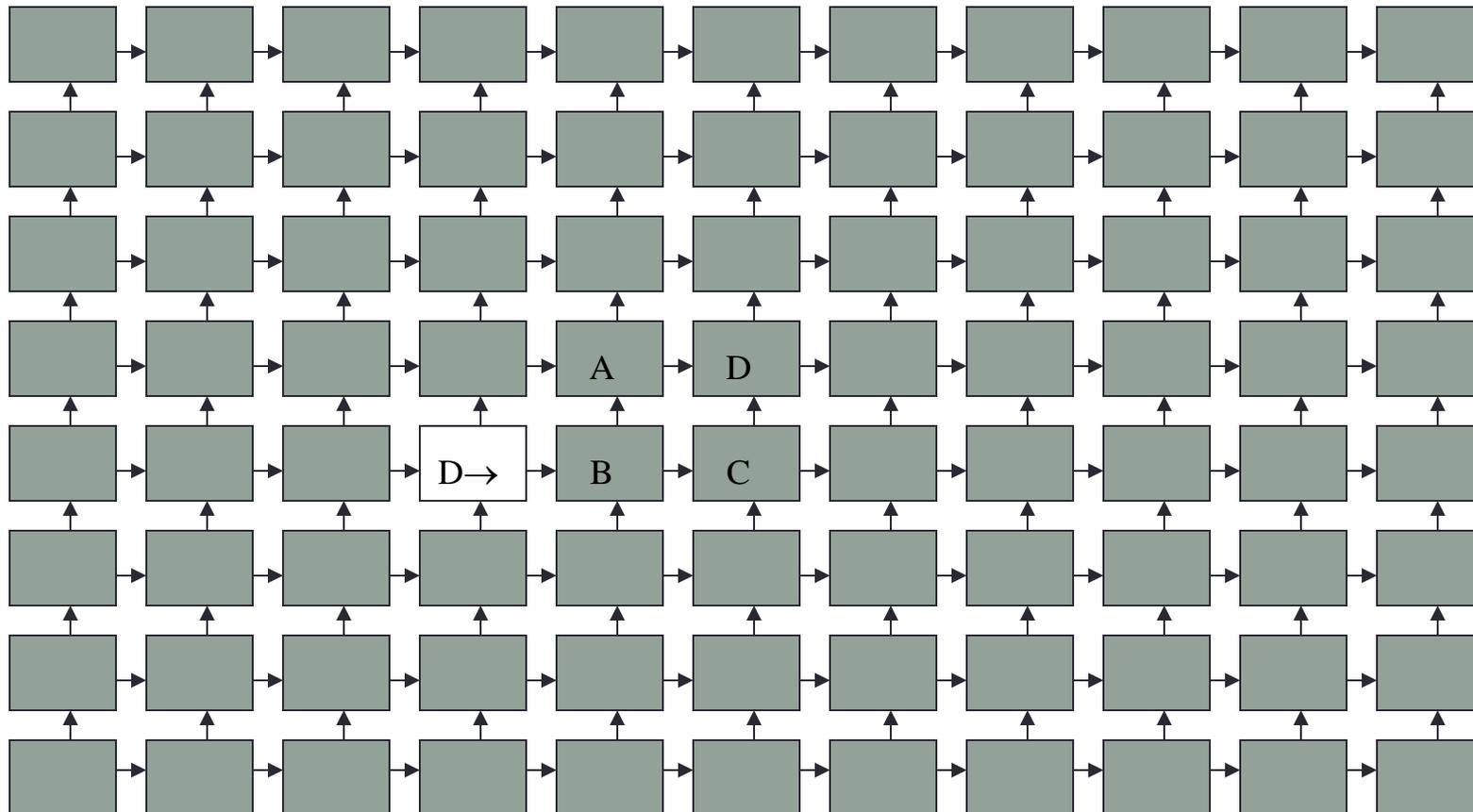
REDES VIRTUALES III

(noroeste)





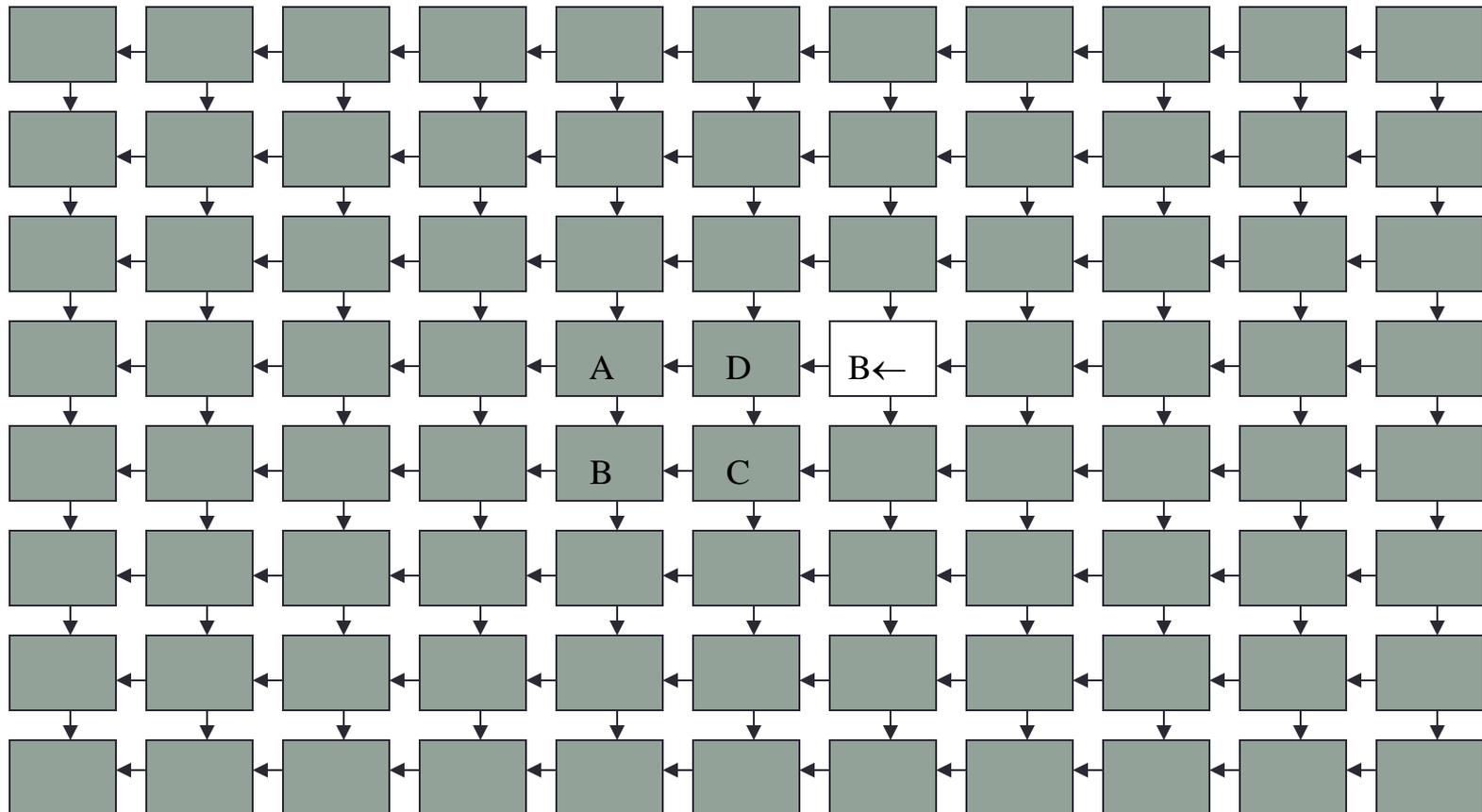
REDES VIRTUALES III (noreste)





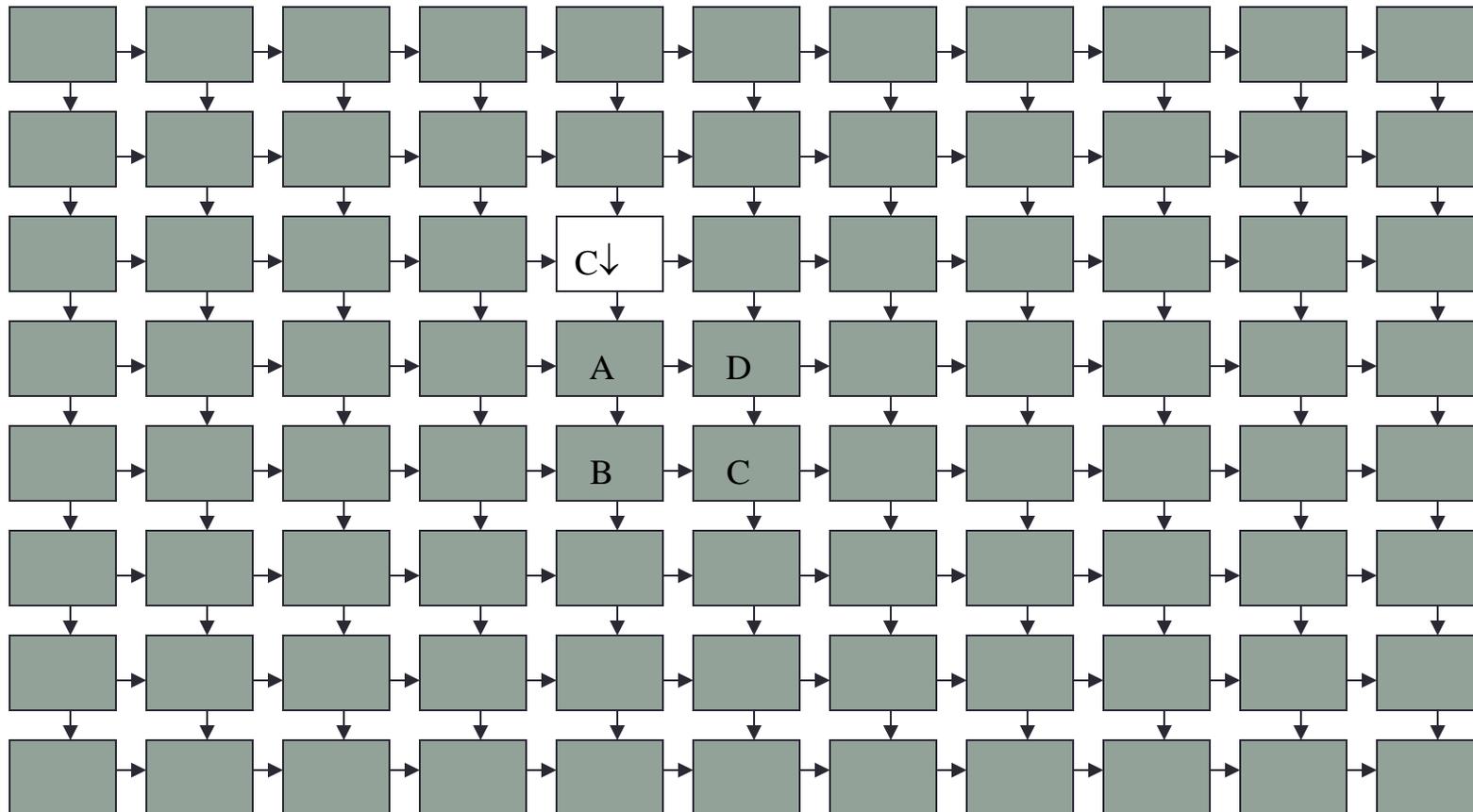
REDES VIRTUALES III

(suroeste)



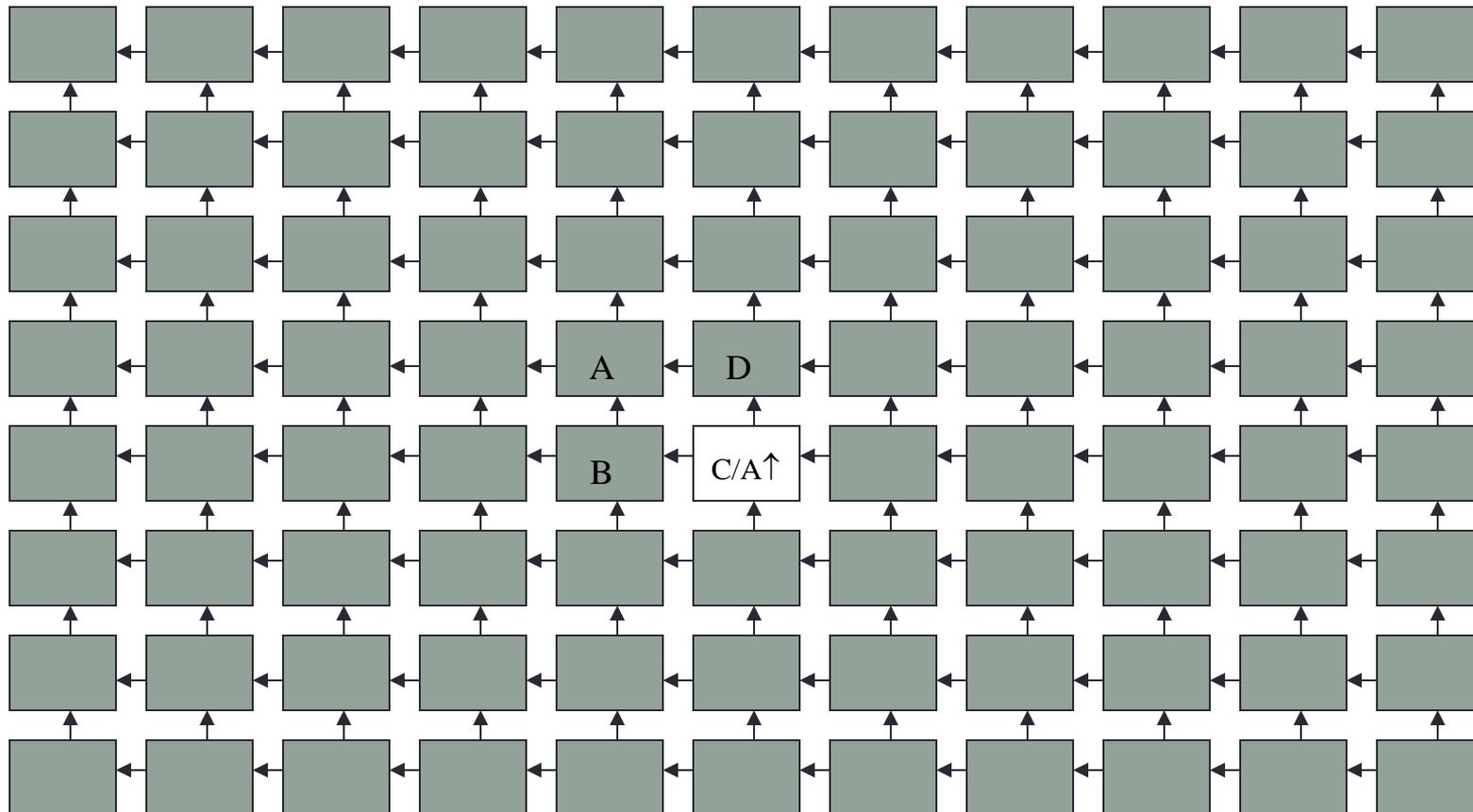
REDES VIRTUALES III

(sureste)

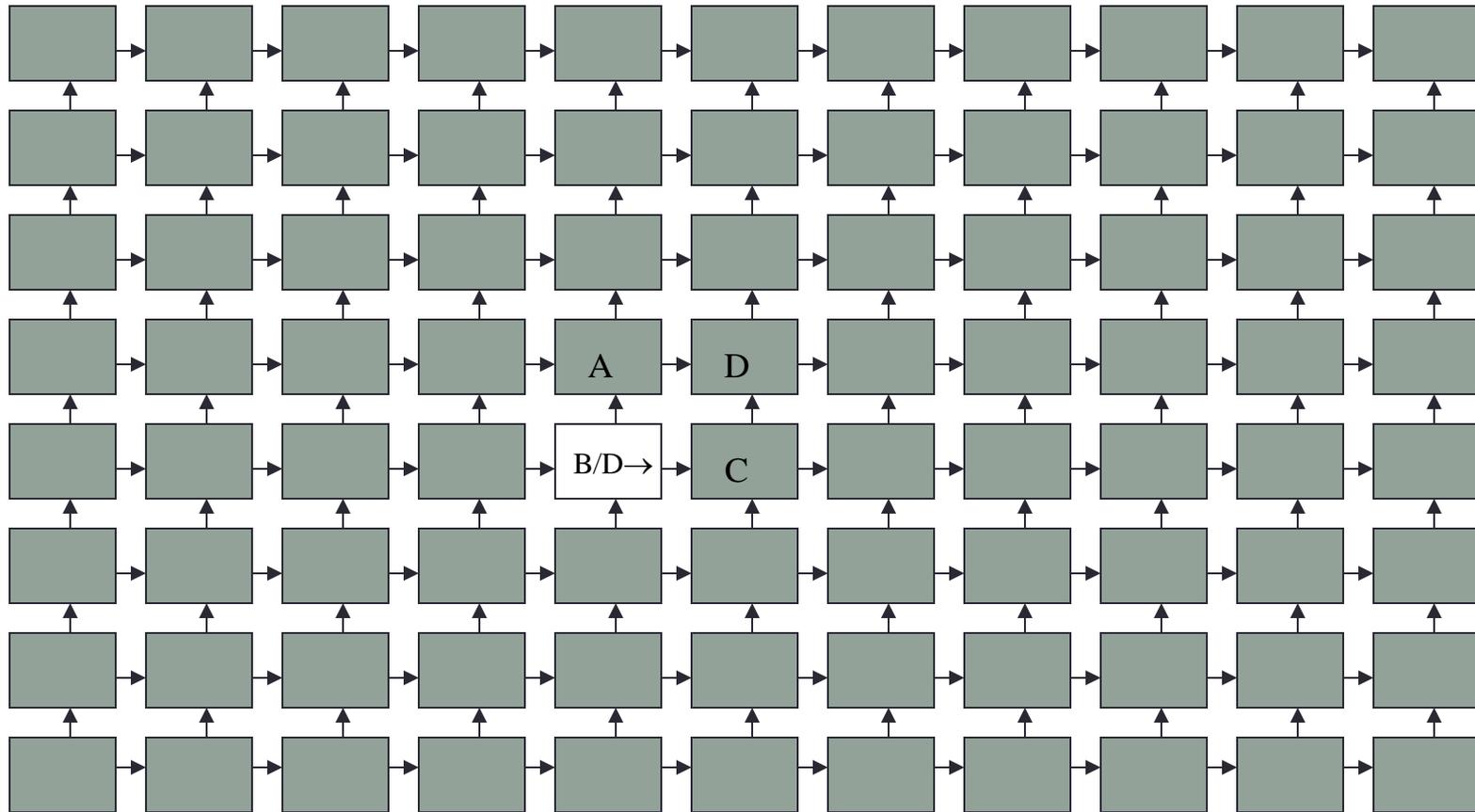


REDES VIRTUALES IV

(noroeste)

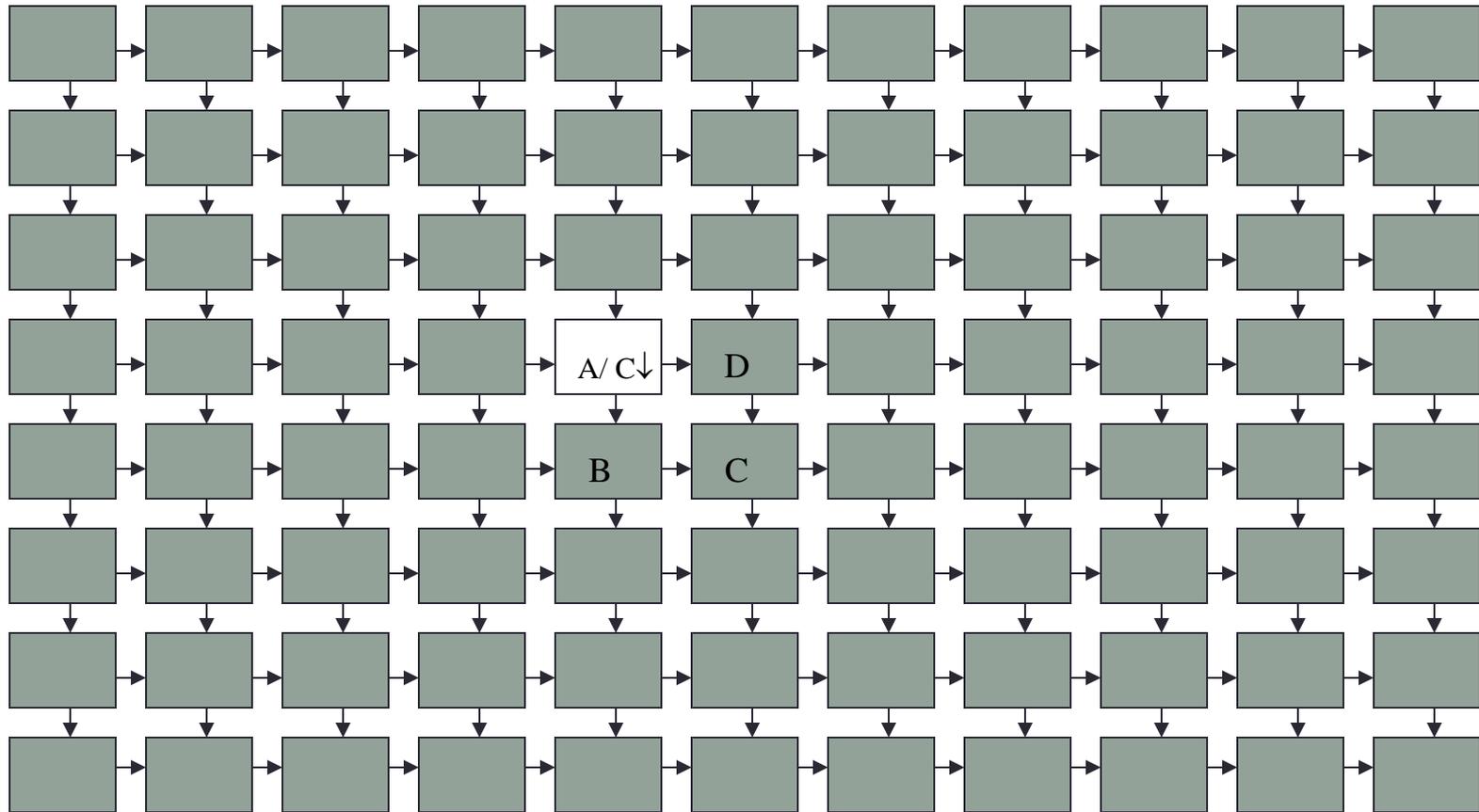


REDES VIRTUALES IV (noreste)

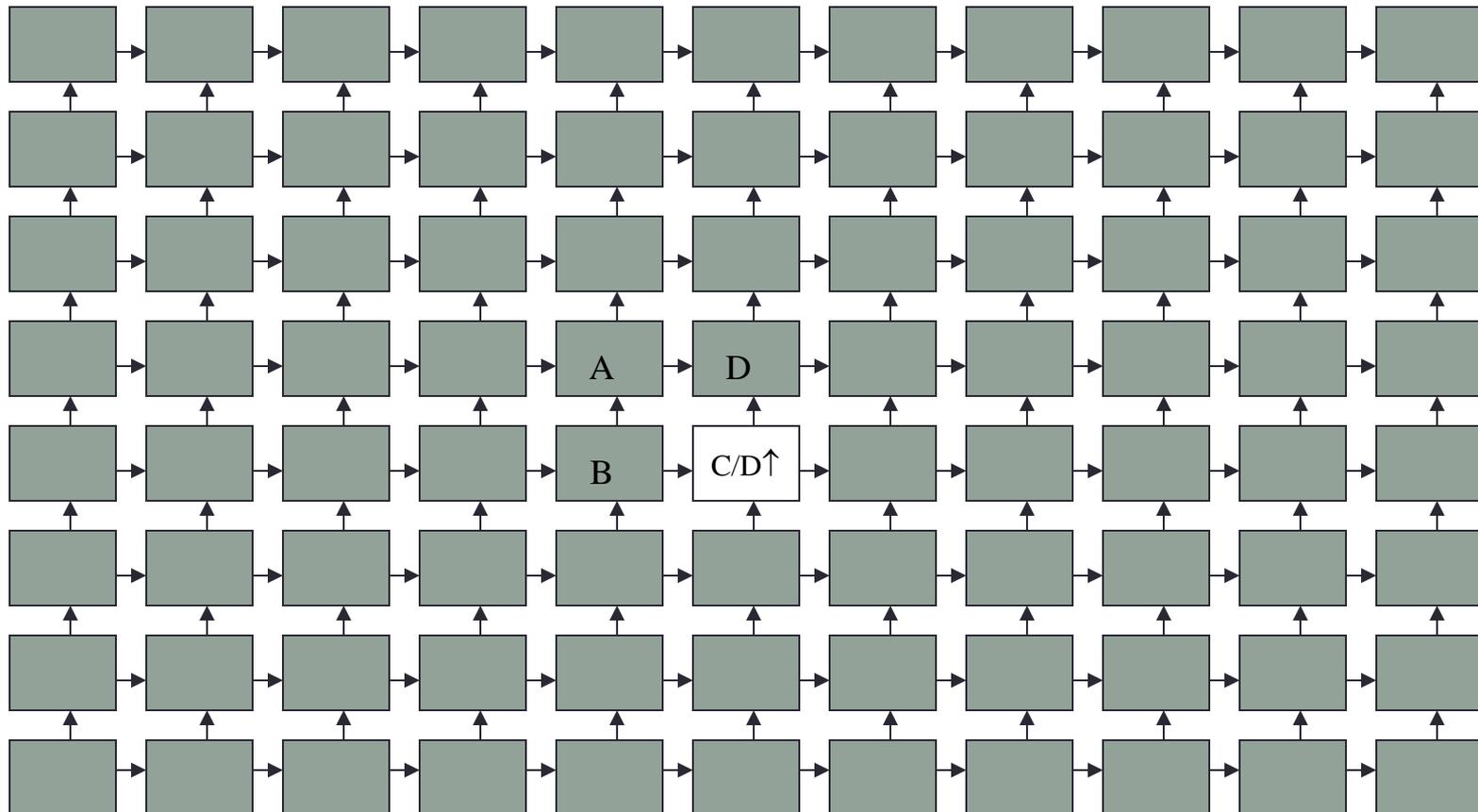




REDES VIRTUALES IV (sureste)

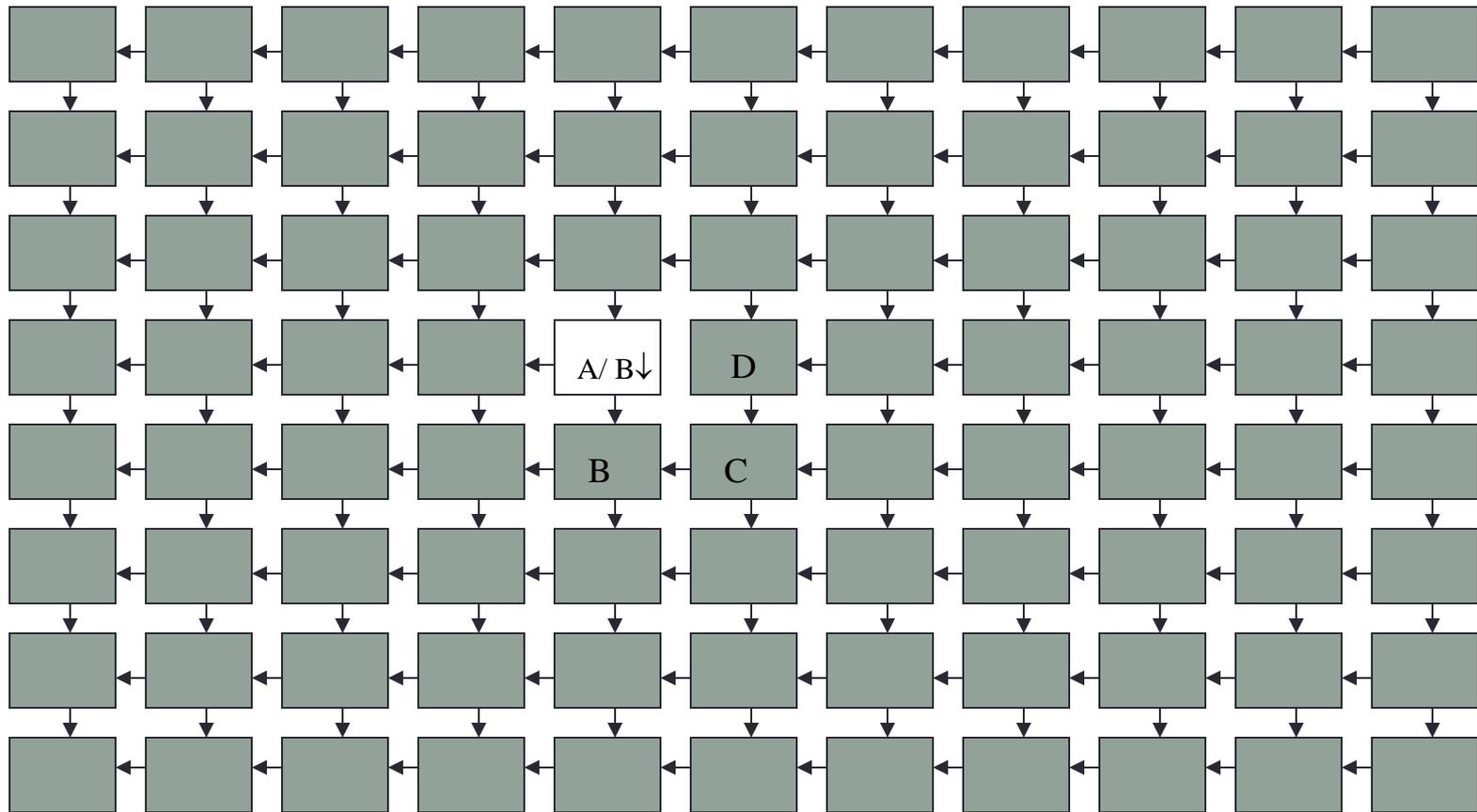


REDES VIRTUALES V (noreste)



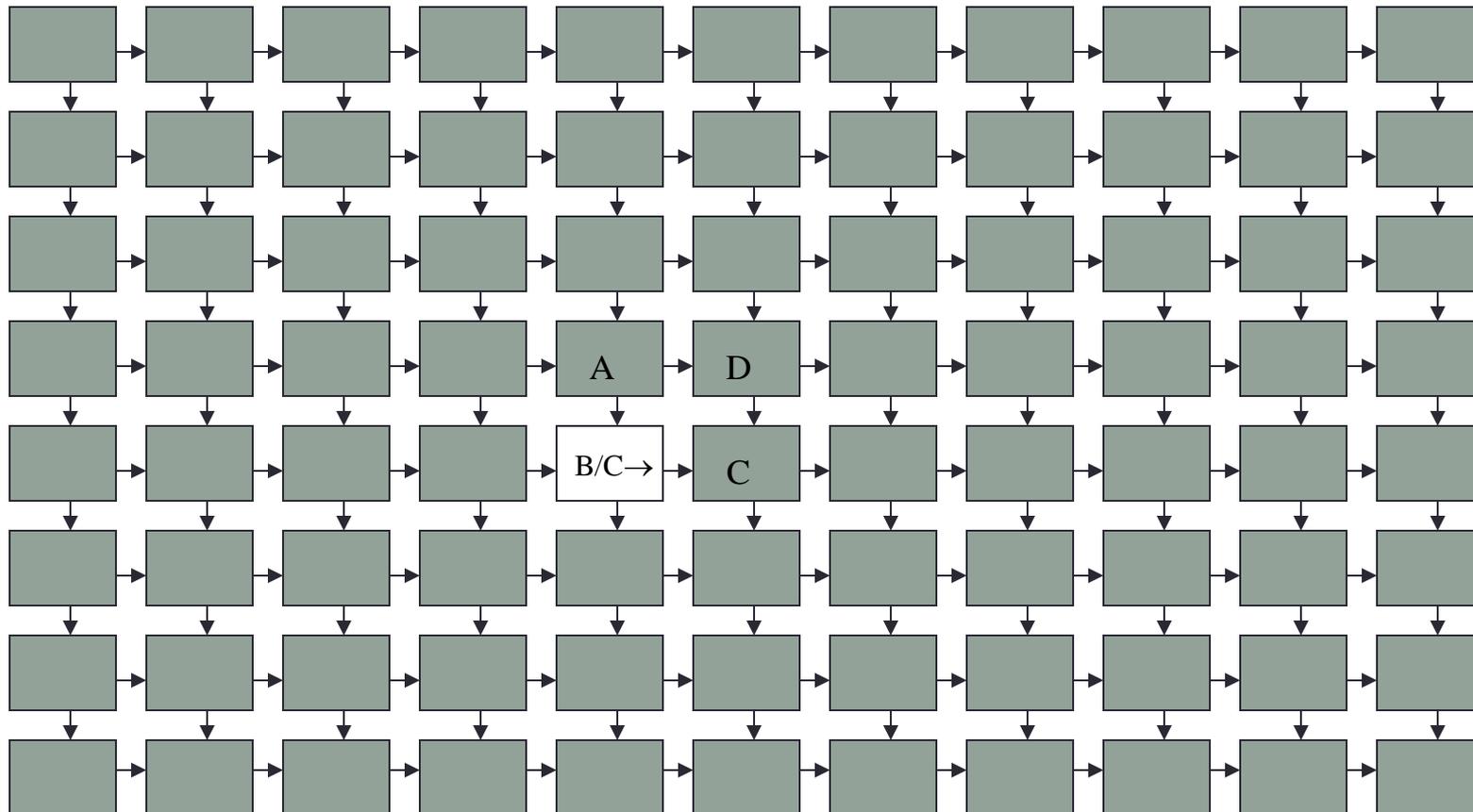


REDES VIRTUALES V (suroeste)





REDES VIRTUALES V (sureste)





Ejemplo de algoritmo de encamianmientto

- Topología: malla.
- Tipo de algoritmo: unicast, progresivo, centralizado, mínimo.
- Según su implementación puede ser:
 - Determinista o adaptativo.
 - Tabla o máquina de estados.
- Compatible con DOR.
- Consiste en restar las coordenadas del nodo fuente de las del nodo destino.
- Para ir del nodo (2,8) al nodo (12, 6), la ruta se compone de $12 - 2 = 10$ saltos en sentido x^+ y $6 - 8 = 2$ saltos en sentido y^- .