Índice (II)

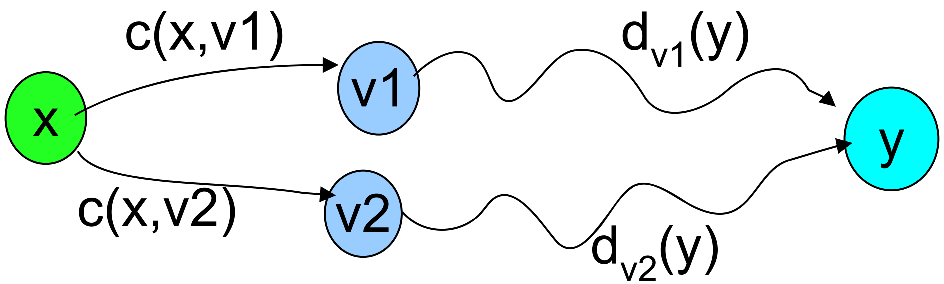
4. Algoritmos de enrutamiento
   - Enrutamiento por estado del enlace
   - ➡ Enrutamiento por vector de distancias
5. Enrutamiento en Internet
   - OSPF
   - BGP

4.3 Enrutamiento por vector de distancias

- Cálculo de la distancia a un destino
  - Coste de la ruta mínima de x a y (ecuación de Bellman-Ford):

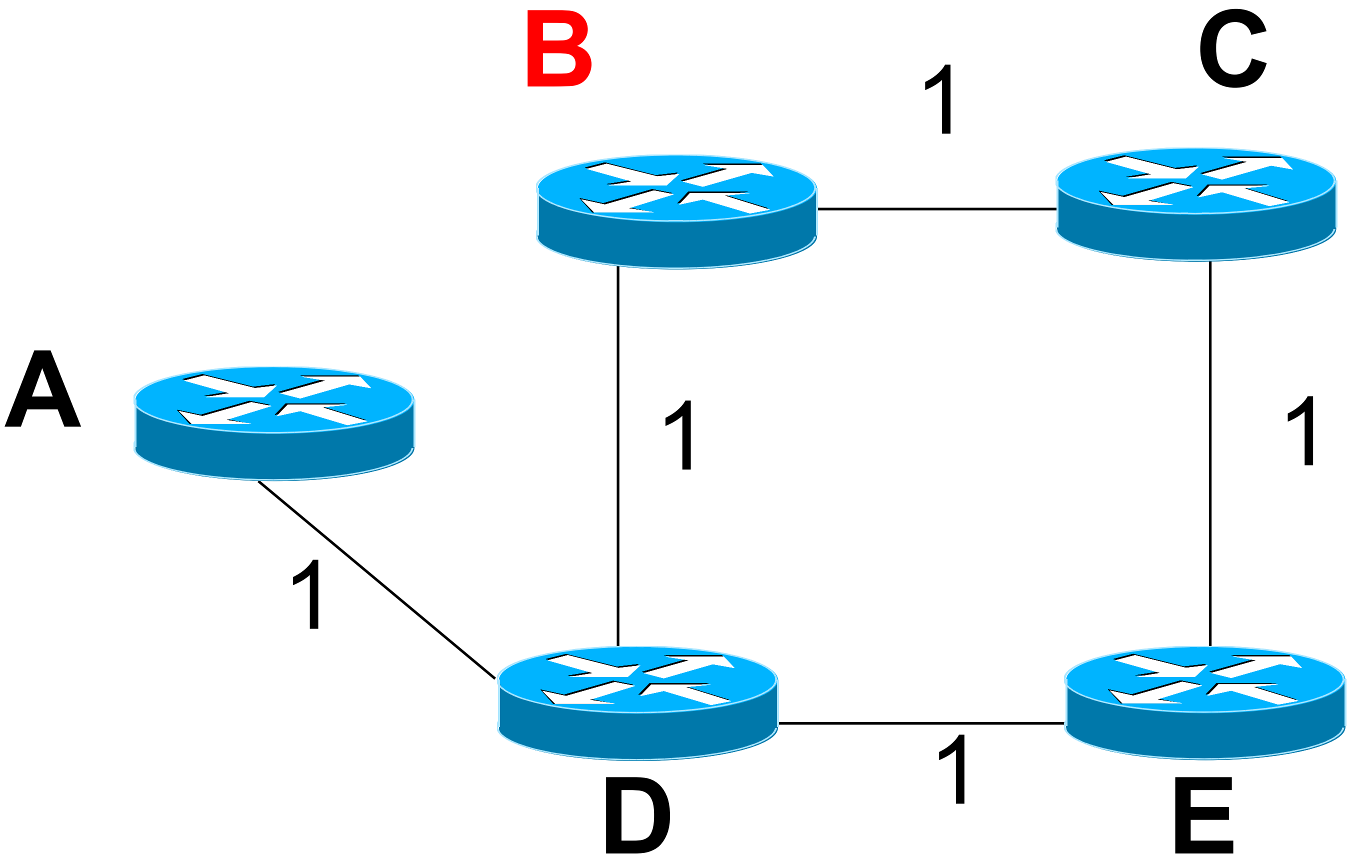
$$\boxed{d_x(y) = \min_v \{c(x,v) + d_v(y)\}}$$

mín se calcula para todos los vecinos de x

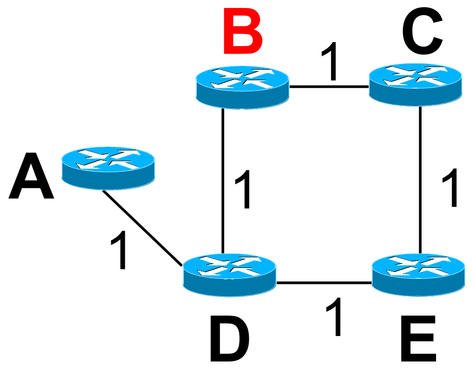


- Con las distancias desde cada vecino se construye la tabla de distancias
- Y a partir de la tabla de distancias la de reenvío

Ejemplo: tabla de distancias del nodo B



Ejemplo: tabla de distancias del nodo B



- **Información disponible en el nodo B:**
  - **Vecinos:** C y D
  - **Distancia a los vecinos**: $d_B(C) = 1$, $d_B(D) = 1$
- **Distancia del nodo B al nodo destino A**

$$d_B(A) = \min\{c(B,C) + d_C(A), c(B,D) + d_D(A)\} = \min\{1 + 3, \mathbf{1 + 1}\} = 2$$

Vectores de distancias de los nodos C y D

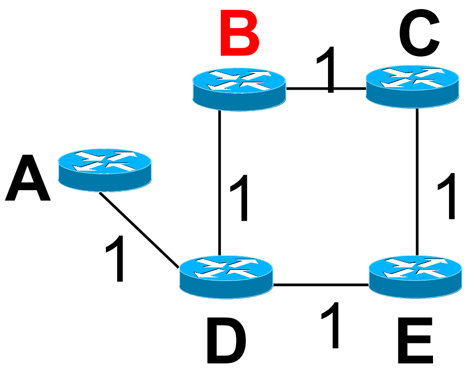
C	
A	3
B	1
D	2
E	1

$d_B(C) = 1$

D	
A	1
B	1
C	2
E	1

$d_B(D) = 1$

Ejemplo: tabla de distancias del nodo B



- **Información disponible en el nodo B:**
  - **Vecinos:** C y D
  - **Distancia a los vecinos**: $d_B(C) = 1$, $d_B(D) = 1$
- **Distancia del nodo B al nodo destino A**

$d_B(A) = \min\{c(B,C) + d_C(A), c(B,D) + d_D(A)\} = \min\{1 + 3, \mathbf{1 + 1}\} = 2$

Vectores de distancias de los nodos C y D

C	
A	3
B	1
D	2
E	1

$d_B(C) = 1$

D	
A	1
B	1
C	2
E	1

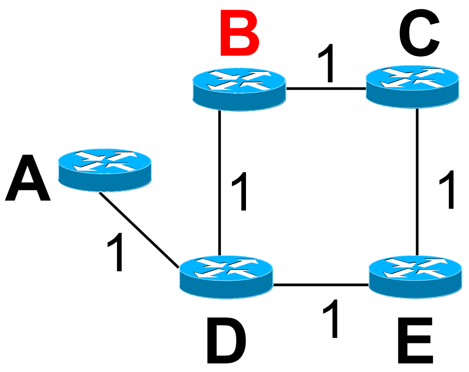
$d_B(D) = 1$

Tabla de distancias del nodo B

	$D_B()$	Coste al destino vía	
		C	D
Destino	A	4	2
	C	1	3
	D	3	1
	E	2	2

Ejemplo: tabla de distancias del nodo B

- **Información disponible en el nodo B:**
  - **Vecinos:** C y D
  - **Distancia a los vecinos**: $d_B(C) = 1$, $d_B(D) = 1$



- **Distancia del nodo B al nodo destino A**

$d_B(A) = \min\{c(B,C) + d_C(A), c(B,D) + d_A(A)\} = \min\{1 + 3, \mathbf{1 + 1}\} = 2$

Vectores de distancias de los nodos C y D

C	
A	3
B	1
D	2
E	1

$d_B(C) = 1$

D	
A	1
B	1
C	2
E	1

$d_B(D) = 1$

Tabla de distancias del nodo B

Coste al destino vía

$D_B()$	C	D
A	4	2
C	1	3
D	3	1
E	2	2

Destino

Tabla de reenvío

Destino	Siguiente salto
A	D
C	C
D	D
E	C

Otro ejemplo de una tabla de reenvío

Coste al destino vía

Destino

$D_D()$	A	B	E
A	1	3	3
B	3	1	3
C	4	2	2
E	3	3	1

Destino	Siguiente
A	A
B	B
C	B
E	E

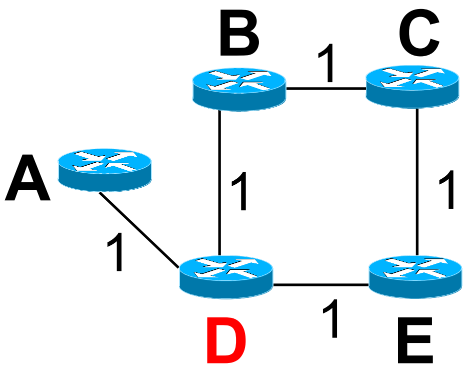


Tabla de distancias → **Tabla de reenvío**

El nodo vecino que proporciona el camino mínimo es el siguiente salto en la tabla de reenvío

Algoritmo de vector de distancias (I)

- $D_x(y)$ = coste mínimo **estimado** desde x hasta y
  - El nodo x mantiene el vector de distancias $D_x=[D_x(y);y\in N]$
- **Nodo x:**
  - Conoce el coste a cada vecino **v**: $c(x,v)$
  - Mantiene los vectores de distancia de sus vecinos. Para cada vecino **v**, **x** almacena $D_v=[D_v(y); y\in N]$
  - Si no conoce la distancia a un destino $D_x(y) =\infty$

Algoritmo de vector de distancias (II)

- Idea básica:
  - Cada nodo envía periódicamente su vector de distancias (VD) a sus vecinos
  - Cuando **x** recibe el VD de un vecino, actualiza, su propio VD (si procede) mediante la ecuación de B-F:

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \text{ para cada nodo } y \in N$$

  - En condiciones normales, la estimación $D_x(y)$ converge al coste real mínimo $d_x(y)$
  - Si el vector de distancias cambia, **x** envía su nuevo vector a sus vecinos, y ellos a su vez pueden actualizar sus vectores de distancia

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$
$$= \min\{2+0 , 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$
$$= \min\{2+1 , 7+0\} = 3$$

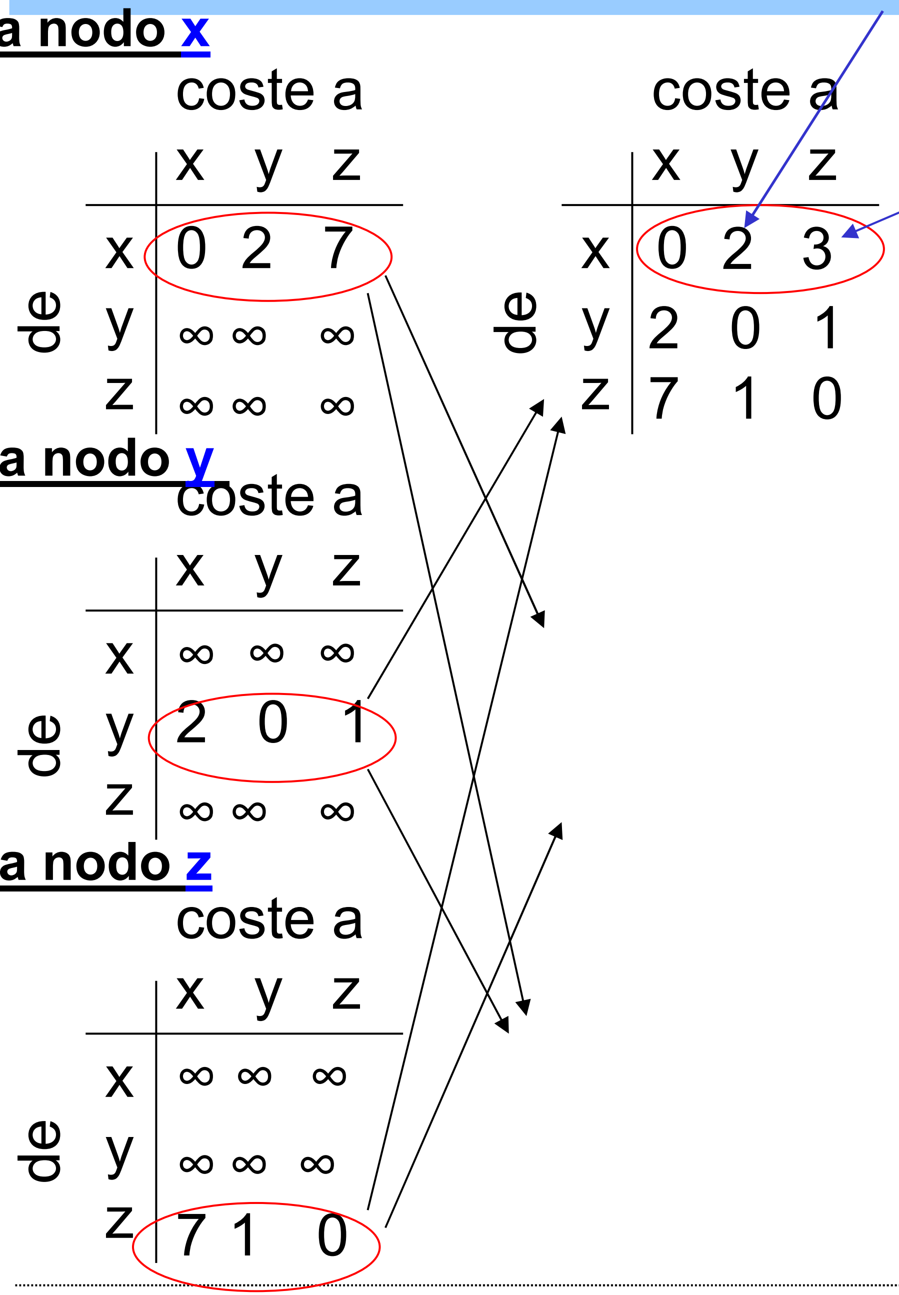


Tabla nodo x

coste a

de	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

coste a

de	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

Tabla nodo y

coste a

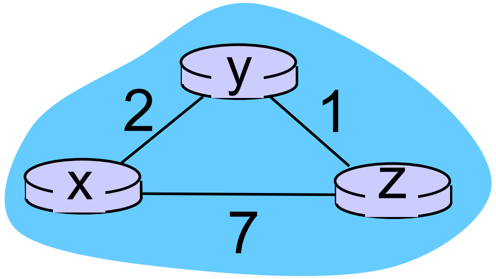
de	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

Tabla nodo z

coste a

de	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

tiempo



$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$
$$= \min\{2+0 , 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$
$$= \min\{2+1 , 7+0\} = 3$$

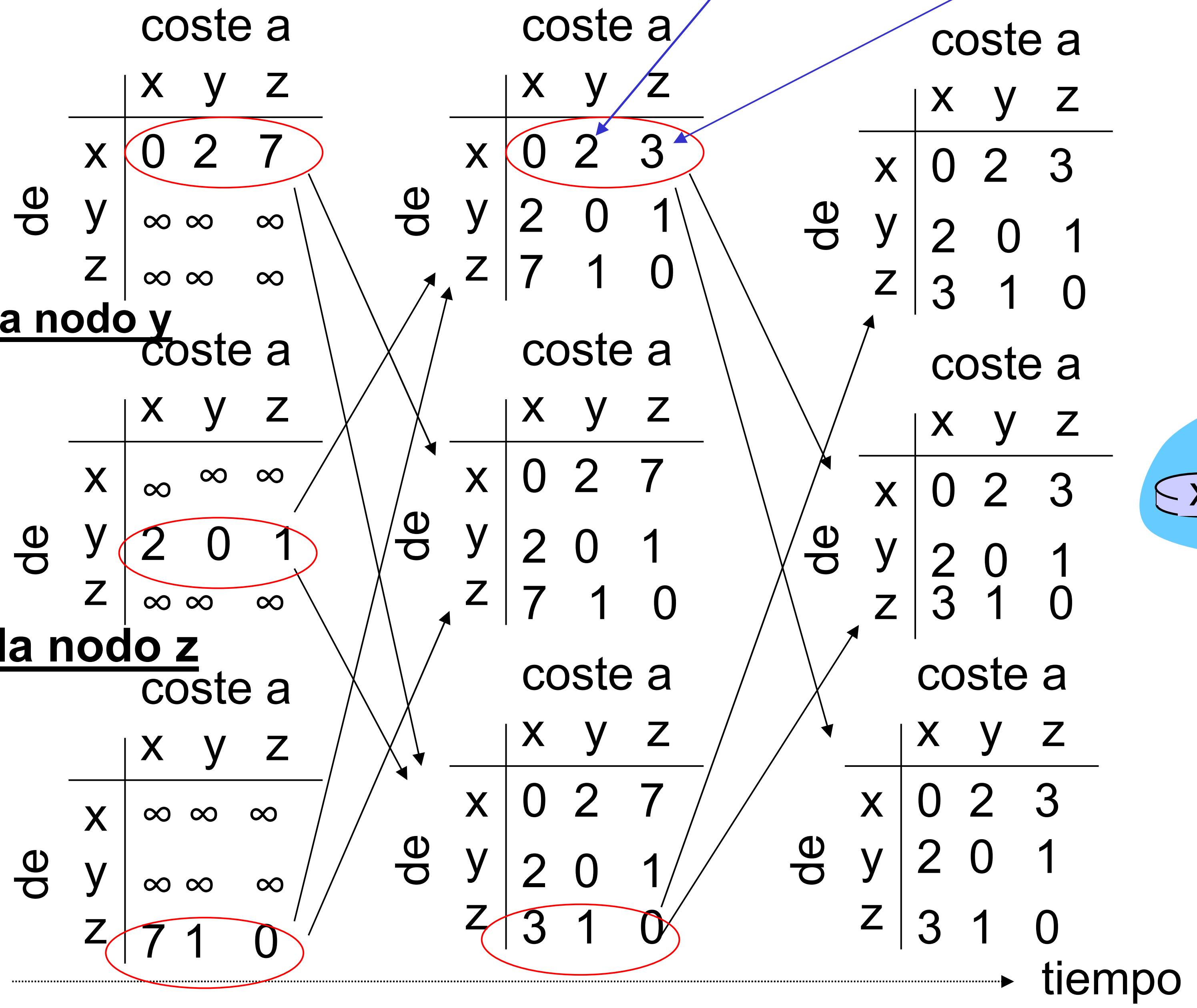


Tabla nodo x

coste a

de	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

coste a

de	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

coste a

de	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Tabla nodo y

coste a

de	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

coste a

de	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

coste a

de	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Tabla nodo z

coste a

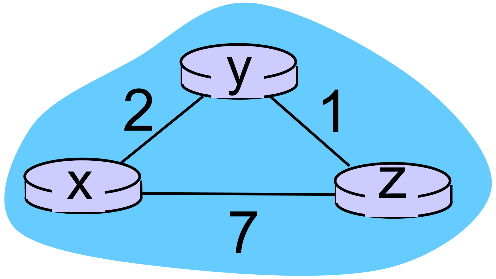
de	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

coste a

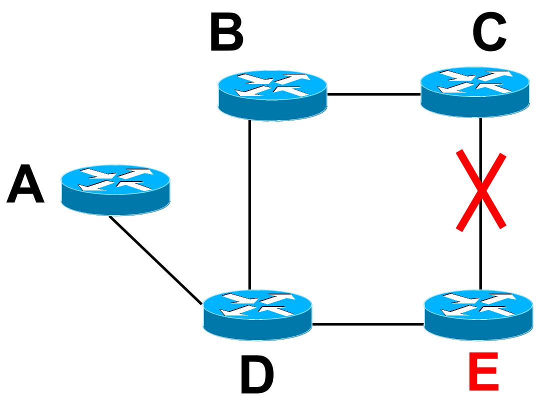
de	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

coste a

de	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



Vector de distancias: cambios en el coste de un enlace (I)



- Los cambios en los costes a veces se resuelven rápidamente

Coste al destino vía:

Destino

$D_E()$	C	D
A	4	2
B	2	2
C	1	3
D	3	1

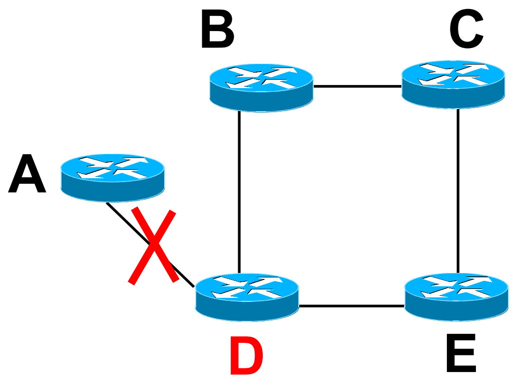
Coste al destino vía:

Destino

$D_E()$	C	D
A	∞	2
B	∞	2
C	∞	3
E	∞	1

Vector de distancias: cambios en el coste de un enlace (II)

En ocasiones la red puede tener problemas para estabilizarse



	Coste al destino vía		
$D_D()$	A	B	E
A	∞	3	3
B	∞	1	3
C	∞	2	2
E	∞	3	1

Destino

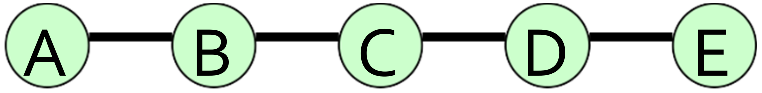
¡La ruta que tiene B utiliza el enlace DA!

- Aparecen bucles en el encaminamiento
- Problema de la **cuenta al infinito**
  - Soluciones:
    - Limitación del diámetro de la red
    - División horizontal (Horizonte dividido) con inversa envenenada

Vector de distancia: problema cuenta hasta infinito

Los valores representan la alcanzabilidad (en “hops”) al **nodo A**

En 4 intercambios TODOS saben que pueden alcanzar al nodo A

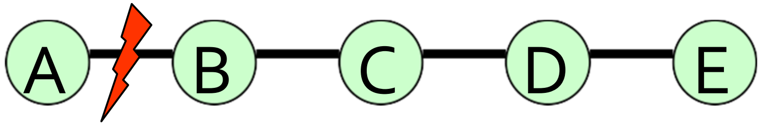


Inf	Inf	Inf	Inf	Inicialmente
1	Inf	Inf	Inf	Tras 1 intercambio
1	2	Inf	Inf	Tras 2 intercambios
1	2	3	Inf	Tras 3 intercambios
1	2	3	4	Tras 4 intercambios

Malas noticias, el nodo A cae

El vecino C comunica a B una alternativa de alcance a A que es dependiente de B

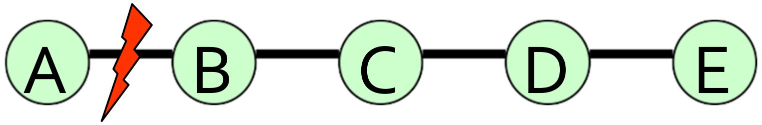
Problema de la cuenta a infinito



1	2	3	4	Inicialmente
3	2	3	4	Tras 1 intercambio
3	4	3	4	Tras 2 intercambios
5	4	5	4	Tras 3 intercambios
5	6	5	6	Tras 4 intercambios
7	6	7	6	Tras 5 intercambios
7	8	7	8	Tras 6 intercambios
	.			
	.			
Inf	Inf	Inf	Inf	

Vector de distancia: problema cuenta hasta infinito

- **Horizonte dividido con Inversa envenenada**:



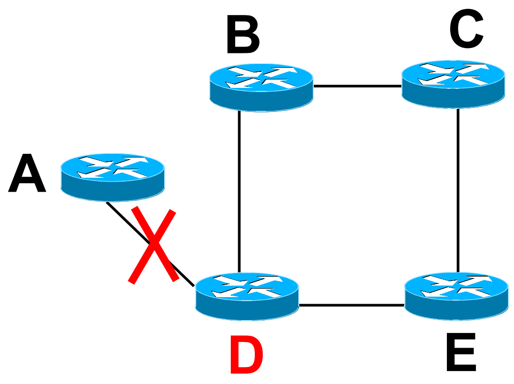
1	2	3	4
∞	2	3	4
∞	∞	3	4
∞	∞	∞	4
∞	∞	∞	∞

Si C sabe que para alcanzar a A lo hace por B ➔ No manda a B esa info. (o manda **[A, ∞]**)

- ¿Resuelve esto completamente el problema de la "cuenta hasta el infinito"?
  - Los bucles que implican a tres o más nodos (en lugar de a dos nodos vecinos) no serán detectados por esta técnica.

Vector de distancias: cambios en el coste de un enlace (III)

- Solución con inversa envenenada



Coste al destino vía

Destino

$D_D()$	A	B	E
A	∞	∞	∞
B	∞	1	3
C	∞	2	2
E	∞	3	1

Pero C ofrecerá a B una ruta a A a través de E

↓

Más iteraciones hasta que C detecte que ni B ni E tienen ruta a A