Tecnología Digital IV: Redes de Computadoras

Clase 17: Protocolos de Ruteo - Parte 2

Lucio Santi & Emmanuel Iarussi

Licenciatura en Tecnología Digital
Universidad Torcuato Di Tella

22 de mayo de 2025

Agenda

- Protocolos de ruteo
  - Introducción
  - Algoritmos *link state*
  - Algoritmos *distance vector*
- Ruteo intra-ISP: **OSPF**
- Ruteo inter-ISP: **BGP**
- *Software Defined Networking* (SDN)

Objetivos

- Entender los conceptos detrás del **plano de control** en Internet
  - Algoritmos de ruteo tradicionales
  - Controladores SDN
- Entender cómo se instancian e implementan en Internet
  - Protocolos OSPF y BGP
  - Controladores SDN: OpenFlow

Algoritmo de ruteo *distance vector*

Basado en la ecuación de Bellman-Ford (programación dinámica)

Si $D_x(y)$ es el costo del camino mínimo de x a y:

$$\boxed{D_x(y) = \text{mín}_v \{ c_{x,v} + D_v(y) \}}$$

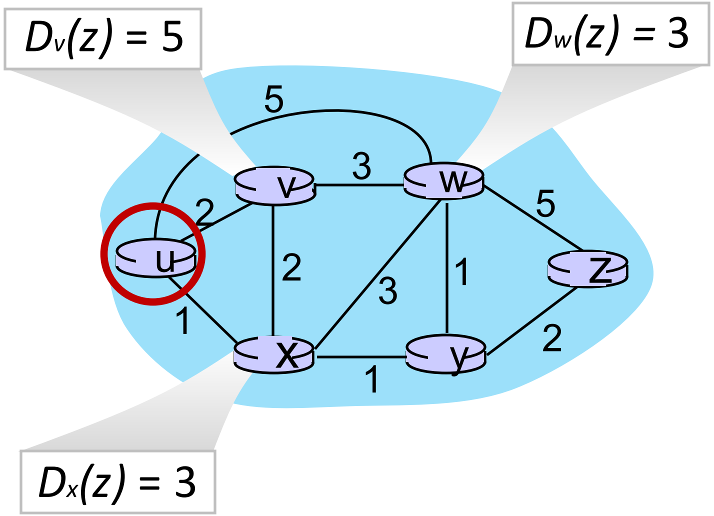
costo del camino mínimo de v a y

costo del enlace de x a v

mínimo tomado sobre
todos los vecinos v de x

Bellman-Ford: ejemplo

Supongamos que los vecinos de *u* (*x*, *v* y *w*) conocen sus caminos mínimos hacia *z*:



De la ecuación de Bellman-Ford,

$$
\begin{aligned}
D_u(z) &= \text{mín}\ \{\, c_{u,v} + D_v(z), \\
&\qquad\quad c_{u,x} + D_x(z), \\
&\qquad\quad c_{u,w} + D_w(z) \} \\
&= \text{mín}\ \{\, 2 + 5, \\
&\qquad\quad 1 + 3, \\
&\qquad\quad 5 + 3 \} = 4
\end{aligned}
$$

el nodo que ofrece el mínimo (*x*) es el *next hop* en el camino mínimo estimado hacia el destino (*z*)

Algoritmo *distance vector*

Idea:

- Cada cierto tiempo, los nodos envían a sus vecinos sus propias estimaciones de caminos mínimos hacia los demás nodos
- Cuando x recibe una estimación, actualiza la propia a partir de la ecuación de Bellman-Ford:

$$D_x(y) \leftarrow \text{mín}_v\{\, c_{x,v} + D_v(y) \,\} \quad \text{para cada nodo } y \text{ en } N$$

- Bajo ciertas condiciones, $D_x(y)$ **converge** al costo mínimo real, $d_x(y)$

Algoritmo *distance vector*

Cada nodo:

Espera: cambios en costos de enlaces locales o mensaje de algún vecino

↓

Recomputa estimaciones a partir de la información recolectada

↓

Notifica a sus vecinos si la estimación a algún destino sufrió cambios

Iterativo y asincrónico: cada iteración se genera por:

- Cambios en los costos locales
- Actualización de los vecinos

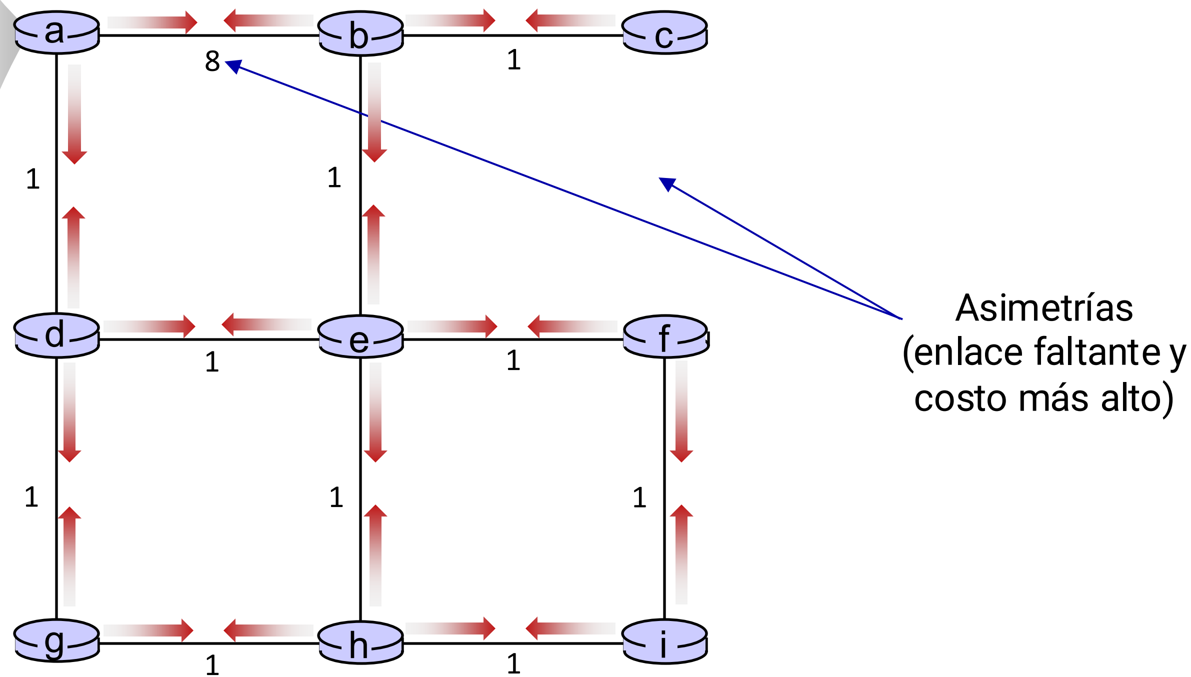
Distribuido y auto-regulado: cada nodo notifica a sus vecinos sólo cuando sus estimaciones cambian:

- No se producen acciones si no se reciben notificaciones

Distance vector: ejemplo

t_0

DV de *a*
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

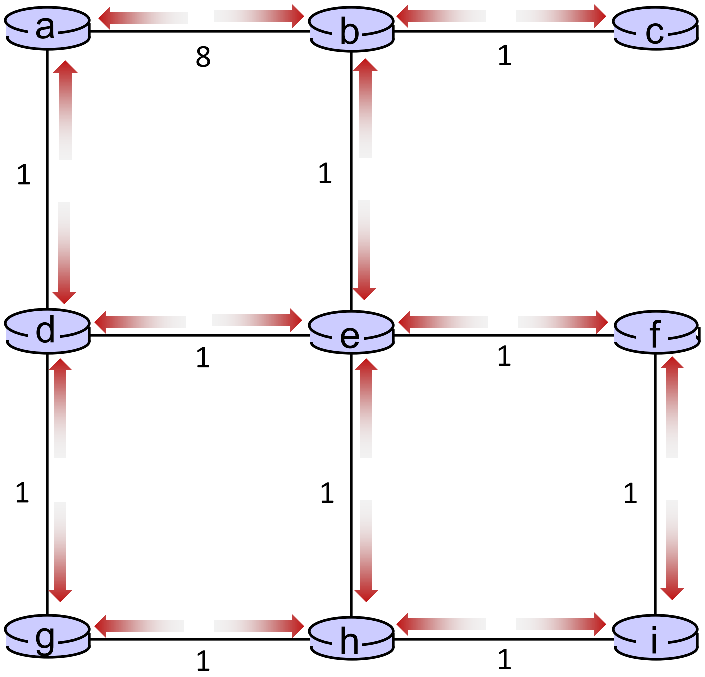


- Los nodos sólo tienen estimaciones hacia sus vecinos directos
- Envían su vector de distancias (**DV**) local a sus vecinos

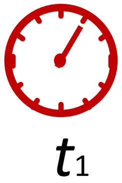
Distance vector: ejemplo - iteración

t_1

Los nodos:
- Reciben DVs de los vecinos
- Computan su nuevo DV local
- Envían el nuevo DV local a sus vecinos

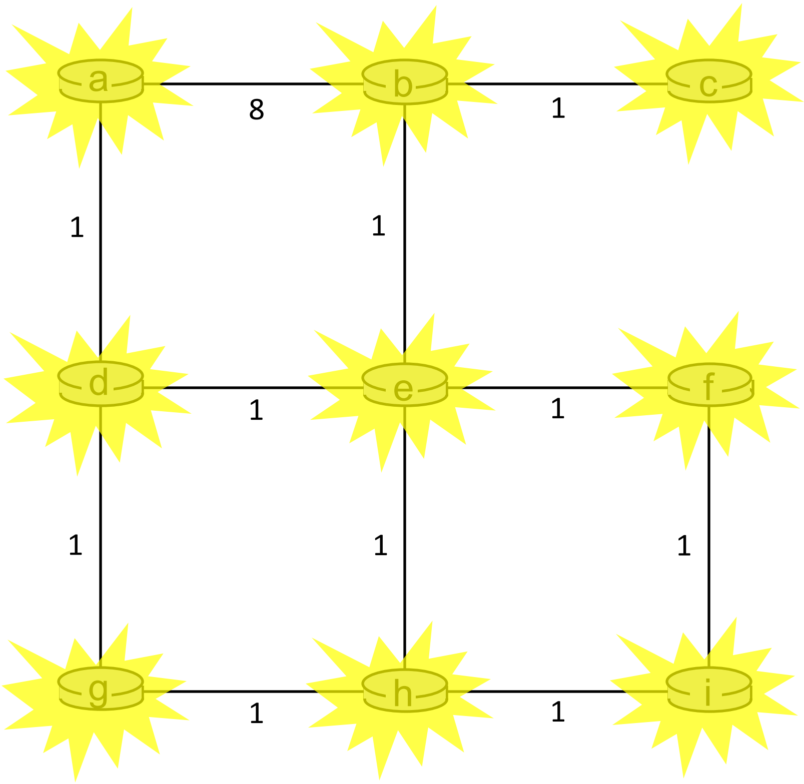


Distance vector: ejemplo - iteración

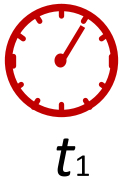


Los nodos:

- Reciben DVs de los vecinos
- Computan su nuevo DV local
- Envían el nuevo DV local a sus vecinos

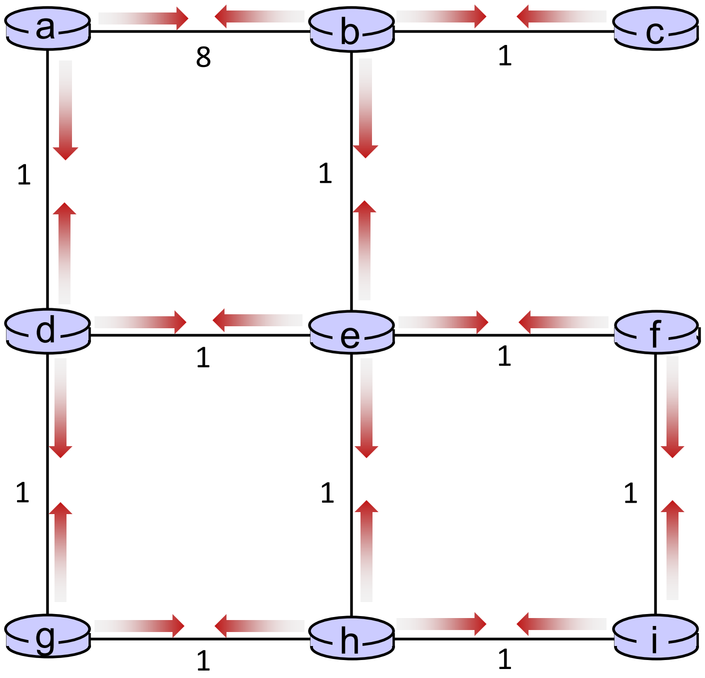


Distance vector: ejemplo - iteración



Los nodos:

- Reciben DVs de los vecinos
- Computan su nuevo DV local
- Envían el nuevo DV local a sus vecinos



Distance vector: ejemplo - iteración

… y así sucesivamente en t_2, t_3, ...

Veamos ahora algunos ejemplos de los cómputos en los nodos

Distance vector: ejempl

t_0

DV en *a*

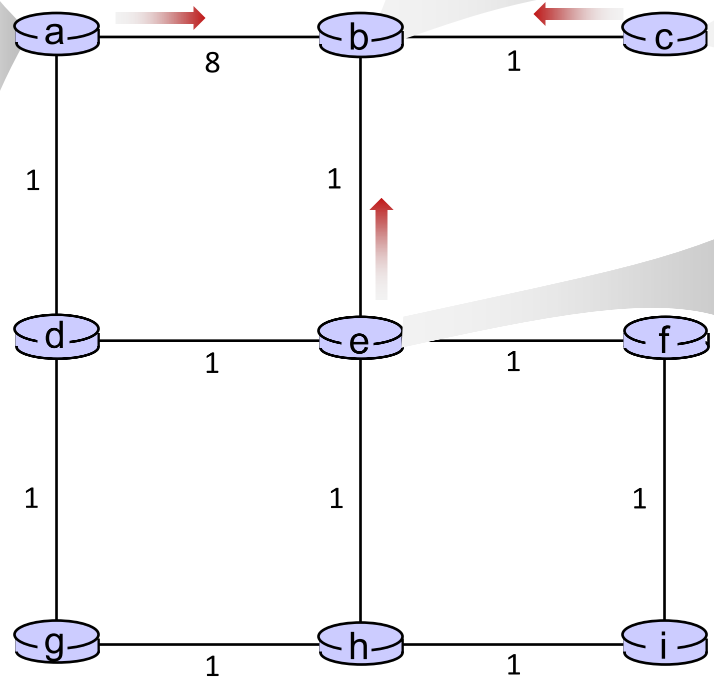
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

DV en *b*

$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV en *c*

$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$



DV en *e*

$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

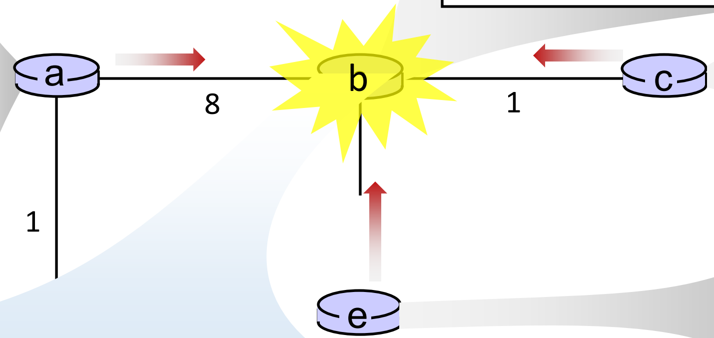
Distance vector: ejempl

t_1

- *b* recibe DVs de *a*, *c*, *e;* computa:

DV en *a*

$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$



DV en *b*

$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV en *c*

$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$

DV en *e*

$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

$D_b(a) = \text{mín}\{c_{b,a}+D_a(a),\ c_{b,c}+D_c(a),\ c_{b,e}+D_e(a)\} = \text{mín}\{8,\infty,\infty\} = 8$

$D_b(c) = \text{mín}\{c_{b,a}+D_a(c),\ c_{b,c}+D_c(c),\ c_{b,e}+D_e(c)\} = \text{mín}\{\infty,1,\infty\} = 1$

$D_b(d) = \text{mín}\{c_{b,a}+D_a(d),\ c_{b,c}+D_c(d),\ c_{b,e}+D_e(d)\} = \text{mín}\{9,\infty,2\} = 2$

$D_b(e) = \text{mín}\{c_{b,a}+D_a(e),\ c_{b,c}+D_c(e),\ c_{b,e}+D_e(e)\} = \text{mín}\{\infty,\infty,1\} = 1$

$D_b(f) = \text{mín}\{c_{b,a}+D_a(f),\ c_{b,c}+D_c(f),\ c_{b,e}+D_e(f)\} = \text{mín}\{\infty,\infty,2\} = 2$

$D_b(g) = \text{mín}\{c_{b,a}+D_a(g),\ c_{b,c}+D_c(g),\ c_{b,e}+D_e(g)\} = \text{mín}\{\infty, \infty, \infty\} = \infty$

$D_b(h) = \text{mín}\{c_{b,a}+D_a(h),\ c_{b,c}+D_c(h),\ c_{b,e}+D_e(h)\} = \text{mín}\{\infty, \infty, 2\} = 2$

$D_b(i) = \text{mín}\{c_{b,a}+D_a(i),\ c_{b,c}+D_c(i),\ c_{b,e}+D_e(i)\} = \text{mín}\{\infty, \infty, \infty\} = \infty$

DV en *b*

$D_b(a) = 8$	$D_b(f) =2$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = 2$	$D_b(h) = 2$
$D_b(e) = 1$	$D_b(i) = \infty$

Distance vector: ejempl

t_1

- *c* recibe DV de *b*

DV en *a*

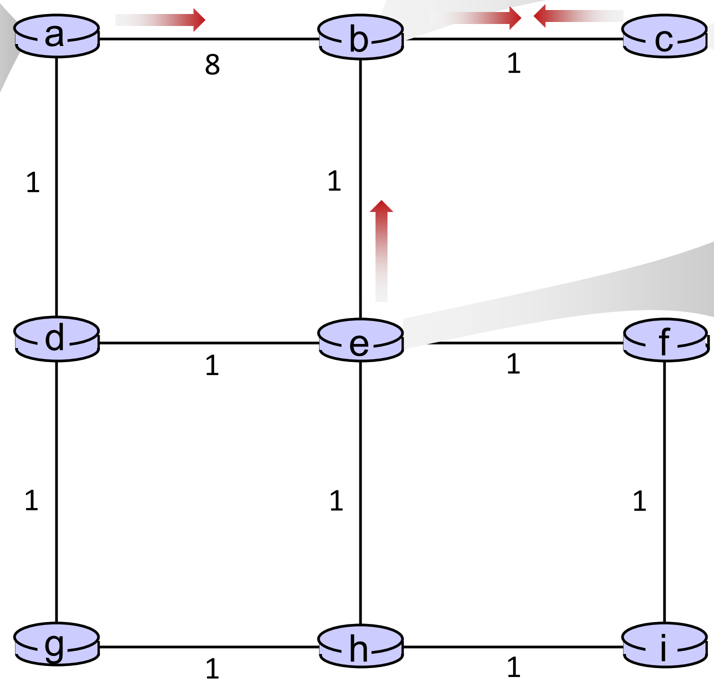
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

DV en *b*

$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV en *c*

$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$



DV en *e*

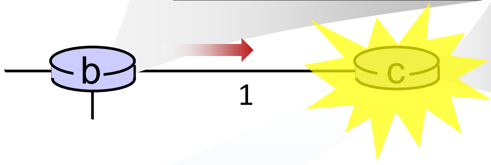
$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

Distance vector: ejemplo

DV en *b*	
$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV en *c*
$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$

t_1



- *c* recibe DV de *b;* computa:

$D_c(a) = \text{mín}\{c_{c,b}+D_b(a)\} = 1 + 8 = 9$

$D_c(b) = \text{mín}\{c_{c,b}+D_b(b)\} = 1 + 0 = 1$

$D_c(d) = \text{mín}\{c_{c,b}+D_b(d)\} = 1+ \infty = \infty$

$D_c(e) = \text{mín}\{c_{c,b}+D_b(e)\} = 1 + 1 = 2$

$D_c(f) = \text{mín}\{c_{c,b}+D_b(f)\} = 1+ \infty = \infty$

$D_c(g) = \text{mín}\{c_{c,b}+D_b(g)\} = 1+ \infty = \infty$

$D_c(h) = \text{mín}\{c_{c,b}+D_b(h)\} = 1+ \infty = \infty$

$D_c(i) = \text{mín}\{c_{c,b}+D_b(i)\} = 1+ \infty = \infty$

DV en *c*
$D_c(a) = 9$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = 2$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$

Distance vector: difusión de información

El estado de cada nodo se propaga a través de la red de manera **iterativa**:

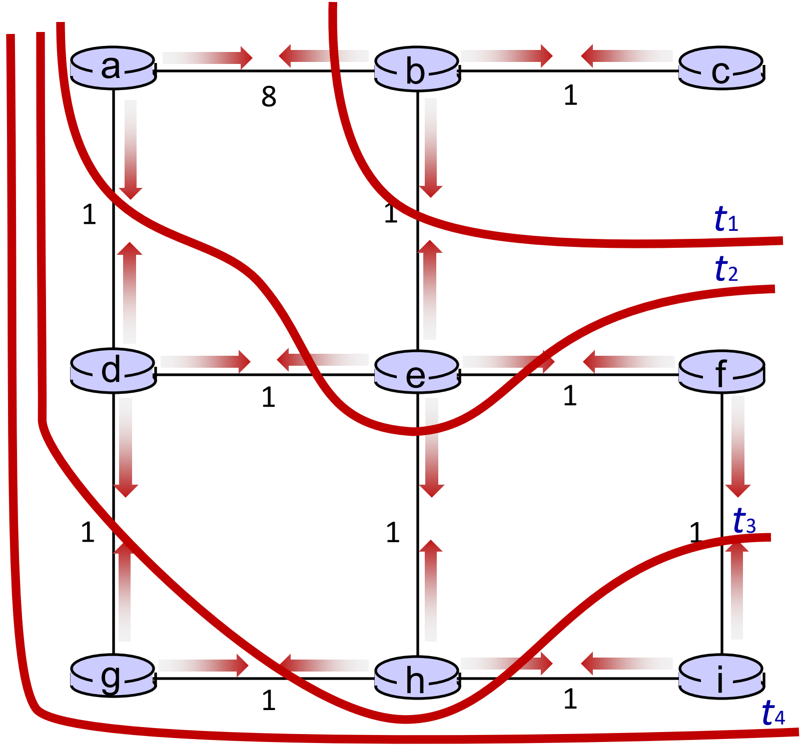
t_0 El estado de *c* en t_0 sólo se ve en *c*

t_1 El estado de *c* en t_1 se propagó a *b* y puede influir en los cómputos de hasta 1 *hop* de distancia (i.e., en *b*)

t_2 El estado de *c* en t_2 puede influir en los cómputos de hasta 2 *hops* de distancia (i.e., en *b, a* y *e*)

t_3 El estado de *c* en t_3 puede influir en los cómputos de hasta 3 *hops* de distancia (i.e., en *b, a* y *e,* y también en *d, h* y *f*)

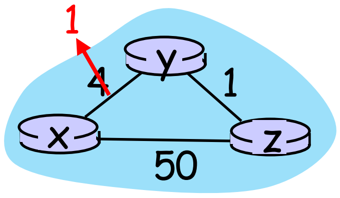
t_4 El estado de *c* en t_4 puede influir en los cómputos de hasta 4 *hops* de distancia (i.e., en *b, a, e, d, h, f, g, h* e *i*)



Distance vector: cambio de costo de enlaces

Cambio de costos en enlaces:

- Un nodo detecta un cambio en el costo de uno de sus enlaces
- Recalcula su DV local
- Si se modifica, envía notificaciones a sus vecinos



t_0 : *y* detecta un cambio de costo, actualiza su DV y notifica a sus vecinos

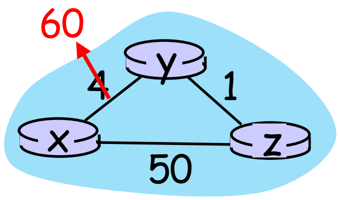
t_1 : *z* recibe el mensaje de *y*, computa el nuevo costo mínimo a *x* y envía a sus vecinos su DV

t_2 : *y* recibe la actualización de *z* pero sus costos mínimos no sufren cambios: *y* **no envía** nuevos mensajes

Distance vector: cambio de costo de enlaces

Cambio de costos en los enlaces:

- Puede suscitar el problema de ***cuenta a infinito***



- *y* nota que el enlace a *x* tiene costo 60 pero *z* informa un camino de costo 5: *y* concluye que puede llegar a *x* con costo 6 vía *z*. Luego informa de esto a *z*.
- *z* observa que el camino a *x* vía *y* tiene nuevo costo 6: *z* actualiza su costo a *x* y lo pone en 7, vía *y*. Luego notifica a *y* de este nuevo costo.
- *y* observa que el camino a *x* vía *z* tiene nuevo costo 7: *y* actualiza su costo a *x* y lo pone en 8, vía *z*. Luego notifica a *z* de este nuevo costo.
- *z* observa que el camino a *x* vía *y* tiene nuevo costo 8: *z* actualiza su costo a *x* y lo pone en 9, vía *y*. Luego notifica a *y* de este nuevo costo.

…

Comparación entre *link state* y *distance vector*

Complejidad en mensajes

LS: *n* routers, O(n^2) mensajes

DV: intercambio de mensajes entre vecinos; tiempo de convergencia variable

Velocidad de convergencia

LS: complejidad *linearítmica*

- Puede tener oscilaciones

DV: tiempo de convergencia variable

- Puede haber ciclos
- Cuenta a infinito

Robustez: resiliencia ante fallas/*hacks* en los routers

LS:

- El router puede informar costos incorrectos de sus enlaces (solamente)
- Cada router sólo computa su propia tabla de ruteo

DV:

- El router puede informar costos incorrectos de **caminos** (fenómeno de *black-holing*)
- Las tablas de los routers son usadas por otros routers: los errores pueden propagarse a través de la red