Tipos de algoritmos de enrutamiento

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

Laboratorio de Programación de Redes
3º Ingeniería Técnica en Informática de Gestión

Objetivo

- Características de los tipos de algoritmos de enrutamiento

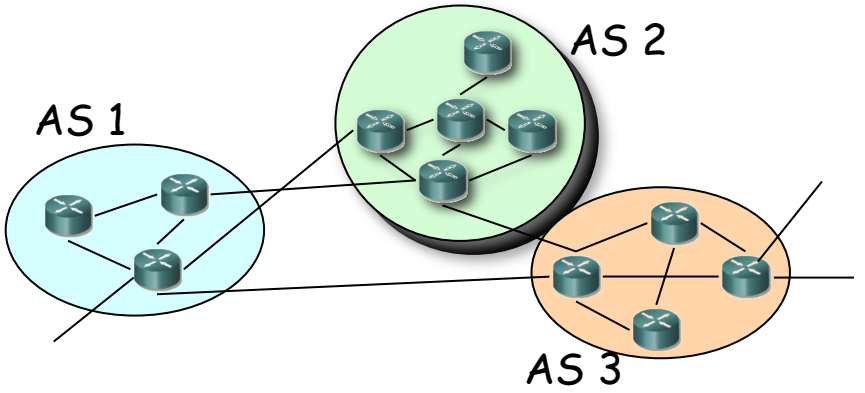
Contenido

- Introducción
- Algoritmos Link-State
- Algoritmos Distance-Vector
  - Descripción
  - Bellman-Ford
- Algoritmos Path-Vector

Tipos de Protocolos de Enrutamiento

Enrutamiento jerárquico

- Sistemas Autónomos (AS)
- Dentro de un AS:
  - ***IGP = Interior Gateway Protocol***
- Entre ASs:
  - ***EGP = Exterior Gateway Protocol***



Tipos de Algoritmos de Enrutamiento

- Deben informar de la topología y los cambios en la misma
- Según cómo diseminan la información

Link State:

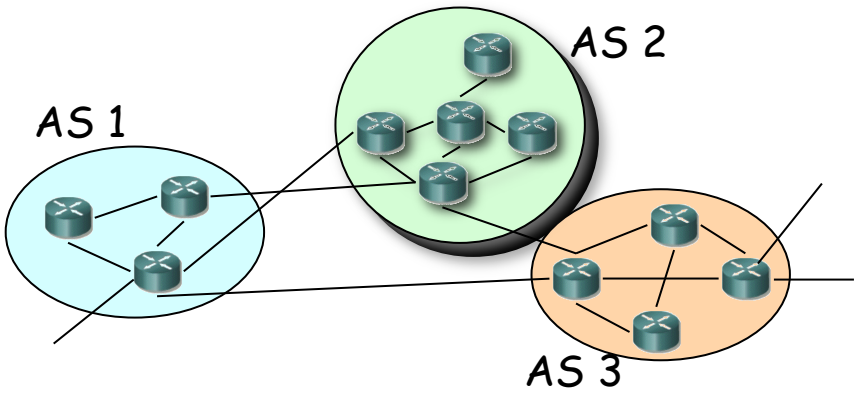
- Comunican qué vecinos tienen y el coste
- Inundan la red
- Cada nodo conoce la topología entera

Distance Vector:

- Comunican estimación de distancia a destinos
- Informan a vecinos

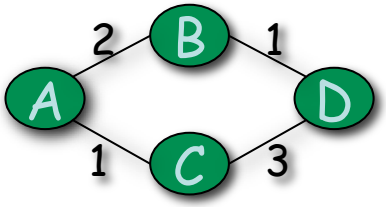
Path Vector:

- Comunican estimación de caminos preferidos a destinos



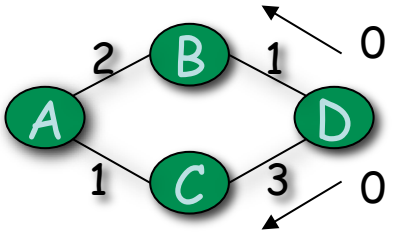
Tipos de Algoritmos de Enrutamiento

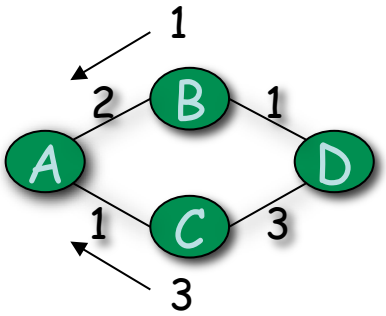
Link State

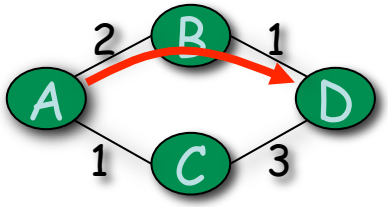


A: [B, 2], [C, 1]
B: [A, 2], [D, 1]
C: [A, 1], [D, 3]
D: [B, 1], [C, 3]

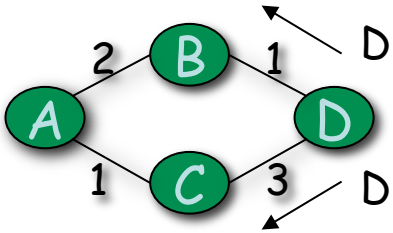
Distance Vector

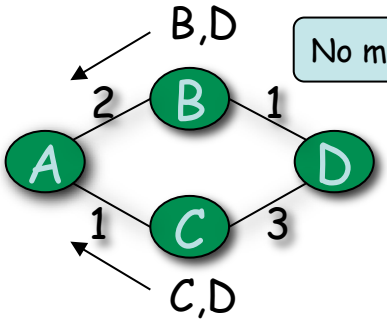


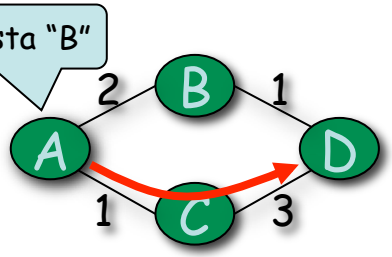




Path Vector





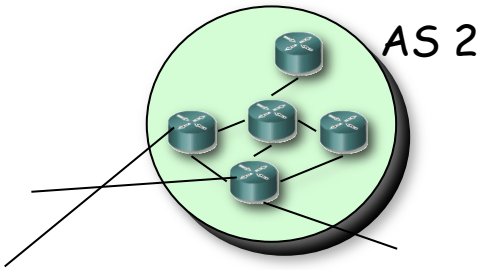


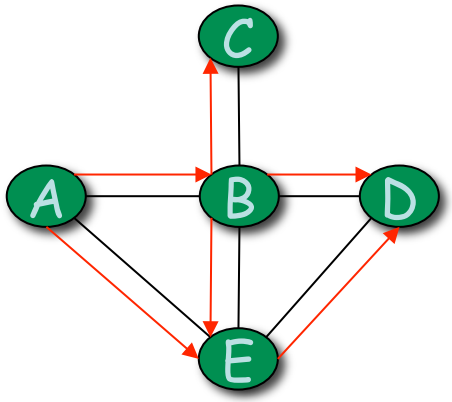
Link State

Tres pasos

1. Descubrir a los vecinos
2. Diseminar la información sobre los enlaces
   - ***Flooding*** (… …)
   - Todos conocen la topología (…)
3. Calcular las rutas
   - Caminos de menor coste
   - Todos calculan los mismos
   - Algoritmo de *Dijkstra*

- OSPF, IS-IS



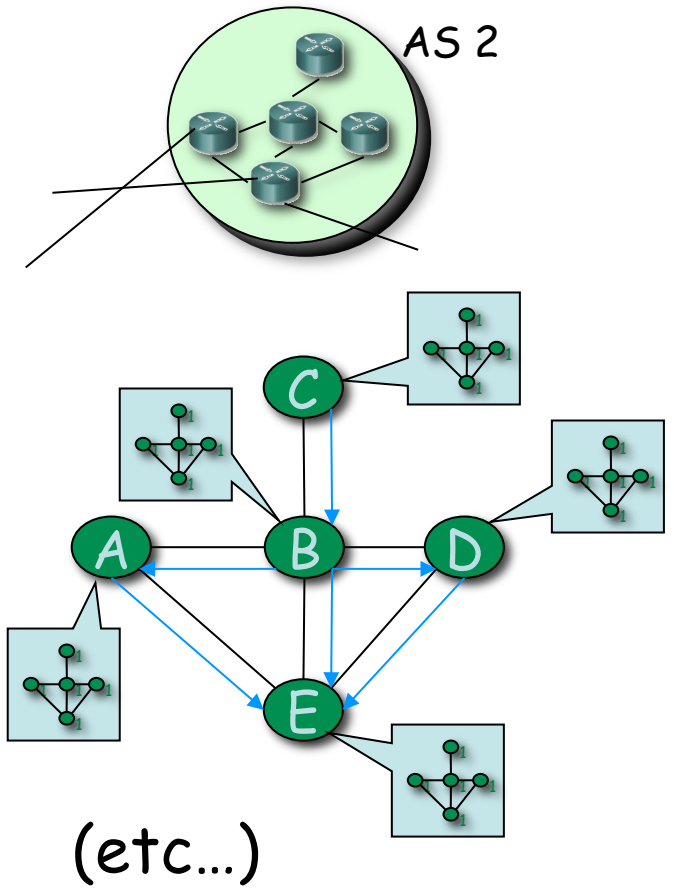


Link State

Tres pasos

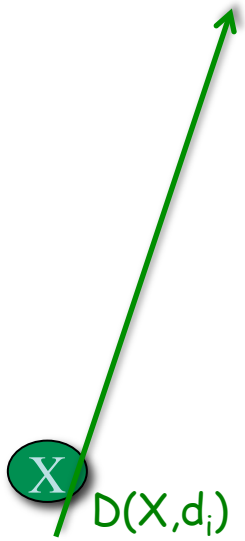
1. Descubrir a los vecinos
2. Diseminar la información sobre los enlaces
   - ***Flooding*** (… …)
   - Todos conocen la topología (…)
3. Calcular las rutas
   - Caminos de menor coste
   - Todos calculan los mismos
   - Algoritmo de *Dijkstra*

- OSPF, IS-IS



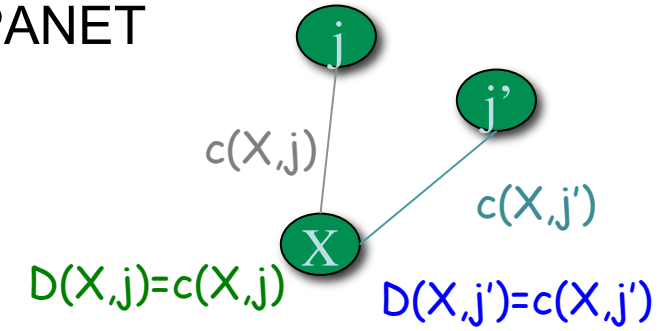
Distance Vector

- Cada nodo llega a conocer la distancia desde él a todos los destinos
  - $D(X,d_i)$
- Inicialmente cada nodo solo conoce distancia a sus vecinos (…)
  - $D(X,d)=c(X,d)$
- Periódicamente comunica D(X,d) a todos sus vecinos
  - Informan con un ***vector*** con las ***distancias*** a los destinos
    ($D(X,d_1)$, $D(X,d_2)$, $D(X,d_3)$, $D(X,d_4)$…)
  - Asíncrono
- Al recibir información actualiza (… …):
  - $D(X,d) \leftarrow \min_{j/c(X,j)<\infty}\{c(X,j)+D(j,d)\}$
- Algoritmo de ***Bellman-Ford*** distribuido
- Empleado desde los comienzos de la ARPANET



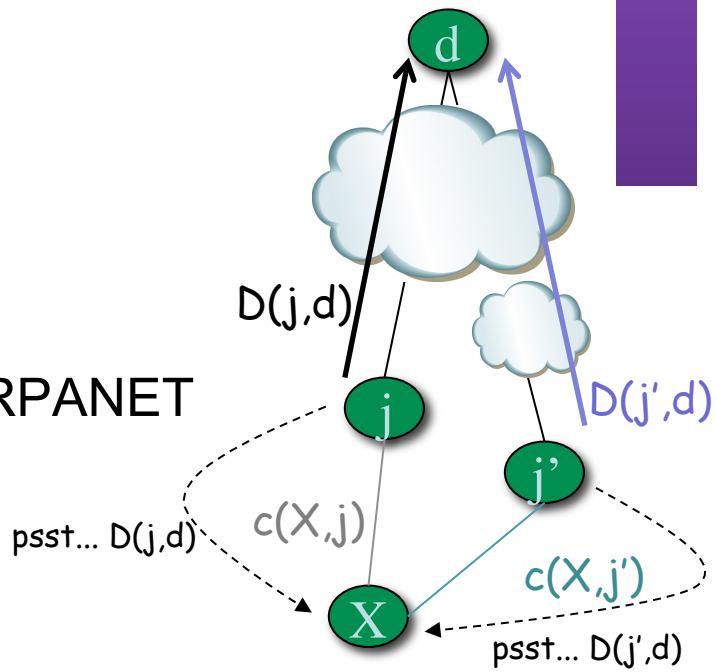
Distance Vector

- Cada nodo llega a conocer la distancia desde él a todos los destinos
  - $D(X,d_i)$
- Inicialmente cada nodo solo conoce distancia a sus vecinos (…)
  - $D(X,d)=c(X,d)$
- Periódicamente comunica D(X,d) a todos sus vecinos
  - Informan con un ***vector*** con las ***distancias*** a los destinos
    ($D(X,d_1)$, $D(X,d_2)$, $D(X,d_3)$, $D(X,d_4)$…)
  - Asíncrono
- Al recibir información actualiza (… …):
  - $D(X,d) \leftarrow \min_{j/c(X,j)<\infty}\{c(X,j)+D(j,d)\}$
- Algoritmo de ***Bellman-Ford*** distribuido
- Empleado desde los comienzos de la ARPANET



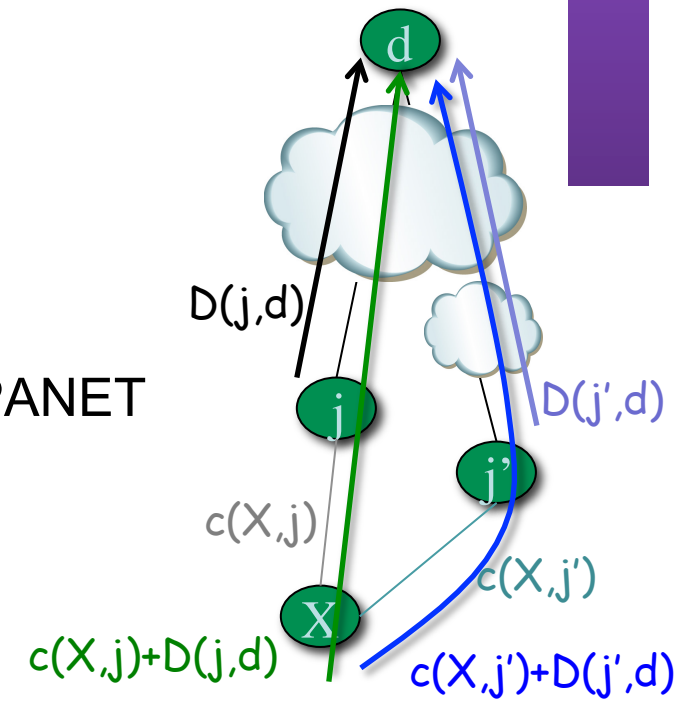
Distance Vector

- Cada nodo llega a conocer la distancia desde él a todos los destinos
  - $D(X,d_i)$
- Inicialmente cada nodo solo conoce distancia a sus vecinos (…)
  - $D(X,d)=c(X,d)$
- Periódicamente comunica D(X,d) a todos sus vecinos
  - Informan con un ***vector*** con las ***distancias*** a los destinos
    ($D(X,d_1)$, $D(X,d_2)$, $D(X,d_3)$, $D(X,d_4)$…)
  - Asíncrono
- Al recibir información actualiza (… …):
  - $D(X,d) \leftarrow \min_{j/c(X,j)<\infty}\{c(X,j)+D(j,d)\}$
- Algoritmo de ***Bellman-Ford*** distribuido
- Empleado desde los comienzos de la ARPANET



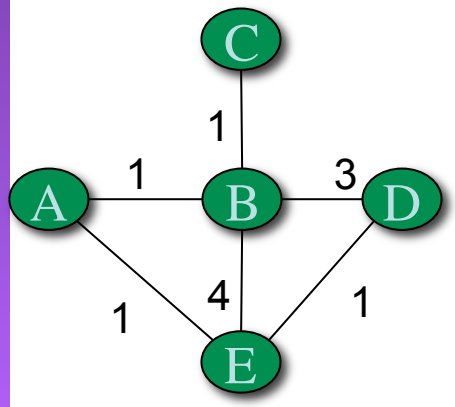
Distance Vector

- Cada nodo llega a conocer la distancia desde él a todos los destinos
  - $D(X,d_i)$
- Inicialmente cada nodo solo conoce distancia a sus vecinos (…)
  - $D(X,d)=c(X,d)$
- Periódicamente comunica D(X,d) a todos sus vecinos
  - Informan con un ***vector*** con las ***distancias*** a los destinos
    ($D(X,d_1)$, $D(X,d_2)$, $D(X,d_3)$, $D(X,d_4)$…)
  - Asíncrono
- Al recibir información actualiza (… …):
  - $D(X,d) \leftarrow \min_{j/c(X,j)<\infty}\{c(X,j)+D(j,d)\}$
- Algoritmo de ***Bellman-Ford*** distribuido
- Empleado desde los comienzos de la ARPANET



Algoritmo de Bellman-Ford

- Comienzo



A

Dest	Next	Cost
B	B	1
C	-	∞
D	-	∞
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	E	4

C

Dest	Next	Cost
A	-	∞
B	B	1
D	-	∞
E	-	∞

D

Dest	Next	Cost
A	-	∞
B	B	3
C	-	∞
E	E	1

E

Dest	Next	Cost
A	A	1
B	B	4
C	-	∞
D	D	1

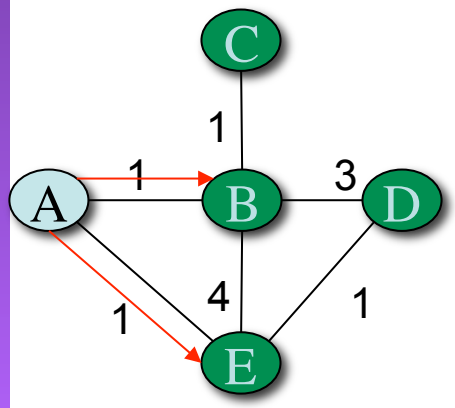
Algoritmo de Bellman-Ford

A envía

D(E,d)←mín{c(E,A)+D(A,d)}

D(B,d)←mín{c(B,A)+D(A,d)}

(…)



A

Dest	Next	Cost
B	B	1
C	-	∞
D	-	∞
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	E	4

C

Dest	Next	Cost
A	-	∞
B	B	1
D	-	∞
E	-	∞

D

Dest	Next	Cost
A	-	∞
B	B	3
C	-	∞
E	E	1

E

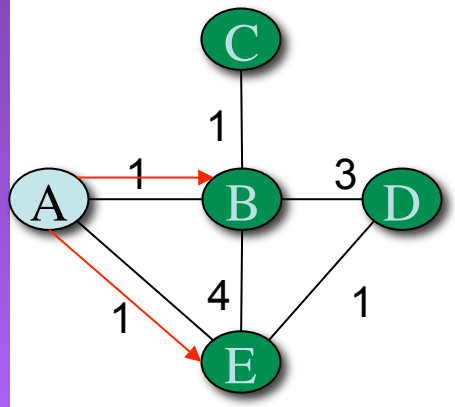
Dest	Next	Cost
A	A	1
B	B	4
C	-	∞
D	D	1

Algoritmo de Bellman-Ford

A envía

D(E,d)←mín{c(E,A)+D(A,d)}

D(B,d)←mín{c(B,A)+D(A,d)}



A

Dest	Next	Cost
B	B	1
C	-	∞
D	-	∞
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	**A** (E)	**2** (4)

C

Dest	Next	Cost
A	-	∞
B	B	1
D	-	∞
E	-	∞

D

Dest	Next	Cost
A	-	∞
B	B	3
C	-	∞
E	E	1

E

Dest	Next	Cost
A	A	1
B	**A** (B)	**2** (4)
C	-	∞
D	D	1

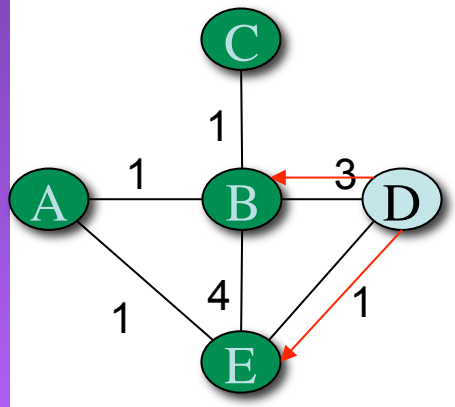
Algoritmo de Bellman-Ford

D envía

D(E,d)←mín{c(E,D)+D(D,d)}

D(B,d)←mín{c(B,D)+D(D,d)}

No hay cambios



A

Dest	Next	Cost
B	B	1
C	-	∞
D	-	∞
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

C

Dest	Next	Cost
A	-	∞
B	B	1
D	-	∞
E	-	∞

D

Dest	Next	Cost
A	-	∞
B	B	3
C	-	∞
E	E	1

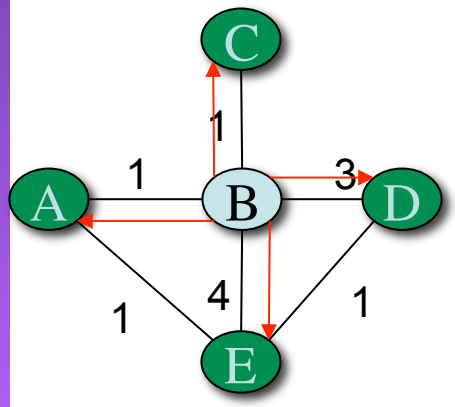
E

Dest	Next	Cost
A	A	1
B	A	2
C	-	∞
D	D	1

Algoritmo de Bellman-Ford

B envía

D(A,d)←mín{c(A,B)+D(B,d)}
D(C,d)←mín{c(C,B)+D(B,d)}
D(D,d)←mín{c(D,B)+D(B,d)}
D(E,d)←mín{c(E,B)+D(B,d)}
(…)



A

Dest	Next	Cost
B	B	1
C	-	∞
D	-	∞
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

C

Dest	Next	Cost
A	-	∞
B	B	1
D	-	∞
E	-	∞

D

Dest	Next	Cost
A	-	∞
B	B	3
C	-	∞
E	E	1

E

Dest	Next	Cost
A	A	1
B	A	2
C	-	∞
D	D	1

Algoritmo de Bellman-Ford

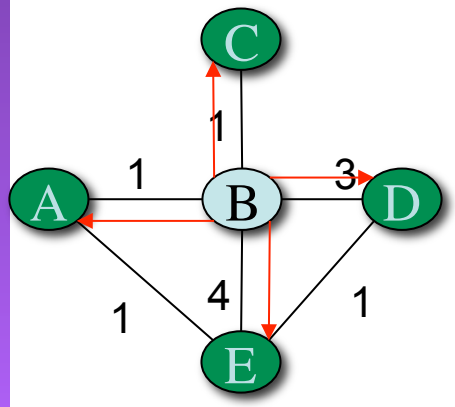
B envía

$D(A,d) \leftarrow mín\{c(A,B)+D(B,d)\}$

$D(C,d) \leftarrow mín\{c(C,B)+D(B,d)\}$

$D(D,d) \leftarrow mín\{c(D,B)+D(B,d)\}$

$D(E,d) \leftarrow mín\{c(E,B)+D(B,d)\}$



Dest	Next	Cost
B	B	1
C	B (-)	2 (∞)
D	B (-)	4 (∞)
E	E	1

A

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

B

Dest	Next	Cost
A	B (-)	2 (∞)
B	B	1
D	B (-)	4 (∞)
E	B (-)	3 (∞)

C

Dest	Next	Cost
A	B (-)	4 (∞)
B	B	3
C	B (-)	4 (∞)
E	E	1

D

Dest	Next	Cost
A	A	1
B	A	2
C	B (-)	5 (∞)
D	D	1

E

Algoritmo de Bellman-Ford

C envía

D(B,d)←mín{c(B,C)+D(C,d)}

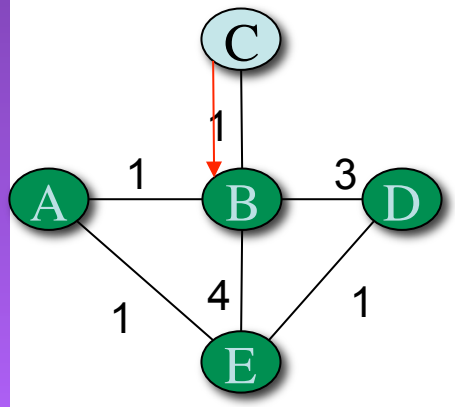
No hay cambios

Dest	Next	Cost
B	B	1
C	B	2
D	B	4
E	E	1

A

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

B



Dest	Next	Cost
A	B	2
B	B	1
D	B	4
E	B	3

C

Dest	Next	Cost
A	B	4
B	B	3
C	B	4
E	E	1

D

Dest	Next	Cost
A	A	1
B	A	2
C	B	5
D	D	1

E

Algoritmo de Bellman-Ford

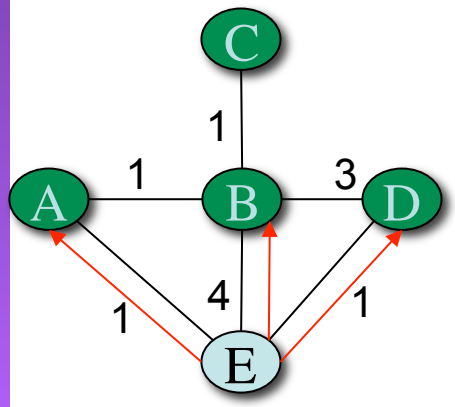
E envía

D(A,d)←mín{c(A,E)+D(E,d)}

D(B,d)←mín{c(B,E)+D(E,d)}

D(D,d)←mín{c(D,E)+D(E,d)}

(…)



A

Dest	Next	Cost
B	B	1
C	B	2
D	B	4
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

C

Dest	Next	Cost
A	B	2
B	B	1
D	B	4
E	B	3

D

Dest	Next	Cost
A	B	4
B	B	3
C	B	4
E	E	1

E

Dest	Next	Cost
A	A	1
B	A	2
C	B	5
D	D	1

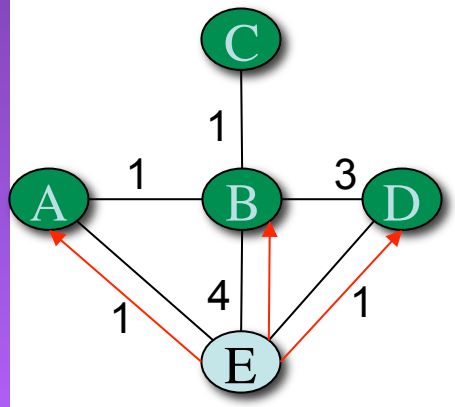
Algoritmo de Bellman-Ford

E envía

D(A,d)←mín{c(A,E)+D(E,d)}

D(B,d)←mín{c(B,E)+D(E,d)}

D(D,d)←mín{c(D,E)+D(E,d)}



Dest	Next	Cost
B	B	1
C	B	2
D	**E** (B)	**2** (4)
E	E	1

A

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

B

Dest	Next	Cost
A	B	2
B	B	1
D	B	4
E	B	3

C

Dest	Next	Cost
A	**E** (B)	**2** (4)
B	B	3
C	B	4
E	E	1

D

Dest	Next	Cost
A	A	1
B	A	2
C	B	5
D	D	1

E

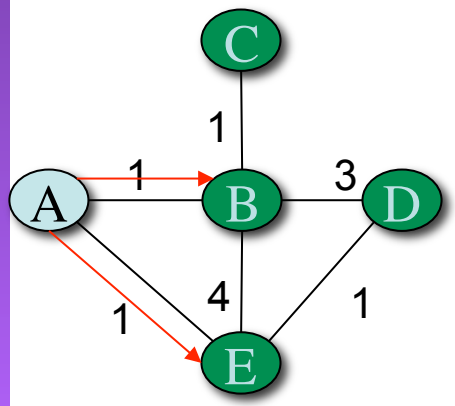
Algoritmo de Bellman-Ford

A envía

D(E,d)←mín{c(E,A)+D(A,d)}

D(B,d)←mín{c(B,A)+D(A,d)}

(…)



A

Dest	Next	Cost
B	B	1
C	B	2
D	E	2
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

C

Dest	Next	Cost
A	B	2
B	B	1
D	B	4
E	B	3

D

Dest	Next	Cost
A	E	2
B	B	3
C	B	4
E	E	1

E

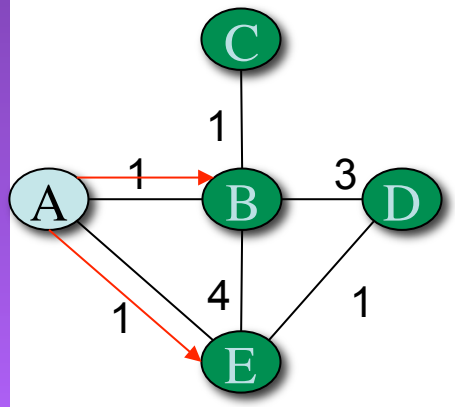
Dest	Next	Cost
A	A	1
B	A	2
C	B	5
D	D	1

Algoritmo de Bellman-Ford

A envía

D(E,d)←mín{c(E,A)+D(A,d)}

D(B,d)←mín{c(B,A)+D(A,d)}



A

Dest	Next	Cost
B	B	1
C	B	2
D	E	2
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

C

Dest	Next	Cost
A	B	2
B	B	1
D	B	4
E	B	3

D

Dest	Next	Cost
A	E	2
B	B	3
C	B	4
E	E	1

E

Dest	Next	Cost
A	A	1
B	A	2
C	**A** (B)	**3** (5)
D	D	1

Algoritmo de Bellman-Ford

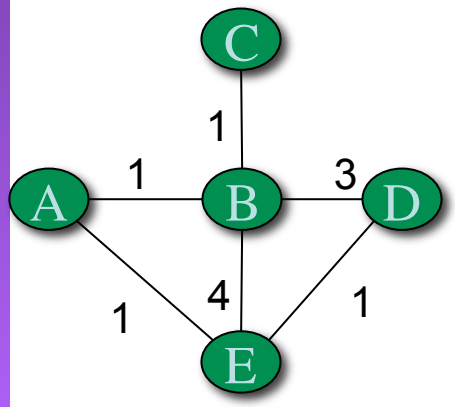
D envía
- No hay cambios

B envía
- No hay cambios

C envía
- No hay cambios

E envía
- No hay cambios

A envía
- No hay cambios



A

Dest	Next	Cost
B	B	1
C	B	2
D	E	2
E	E	1

B

Dest	Next	Cost
A	A	1
C	C	1
D	D	3
E	A	2

C

Dest	Next	Cost
A	B	2
B	B	1
D	B	4
E	B	3

D

Dest	Next	Cost
A	E	2
B	B	3
C	B	4
E	E	1

E

Dest	Next	Cost
A	A	1
B	A	2
C	A	3
D	D	1

Distance Vector

- Cálculo distribuido
- Iterativo e incremental
- Asíncrono
- Converge a los caminos de menor coste
- Protocolos: RIP, IPX-RIP, DECnet, IGRP, EIGRP, DSDV

Path Vector

- Similar a Distance Vector
- Cálculo distribuido
- Informan a sus vecinos de las rutas calculadas
- Incluyen todo el camino hasta el destino para cada ruta
- Protocolos: BGP