

Routing: Algoritmos y arquitectura

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

Grado en Ingeniería en Tecnologías de
Telecomunicación, 3º

Temas de teoría

1. Introducción
2. QoS
3. Encaminamiento dinámico en redes IP
4. Tecnologías móviles
5. Otros temas

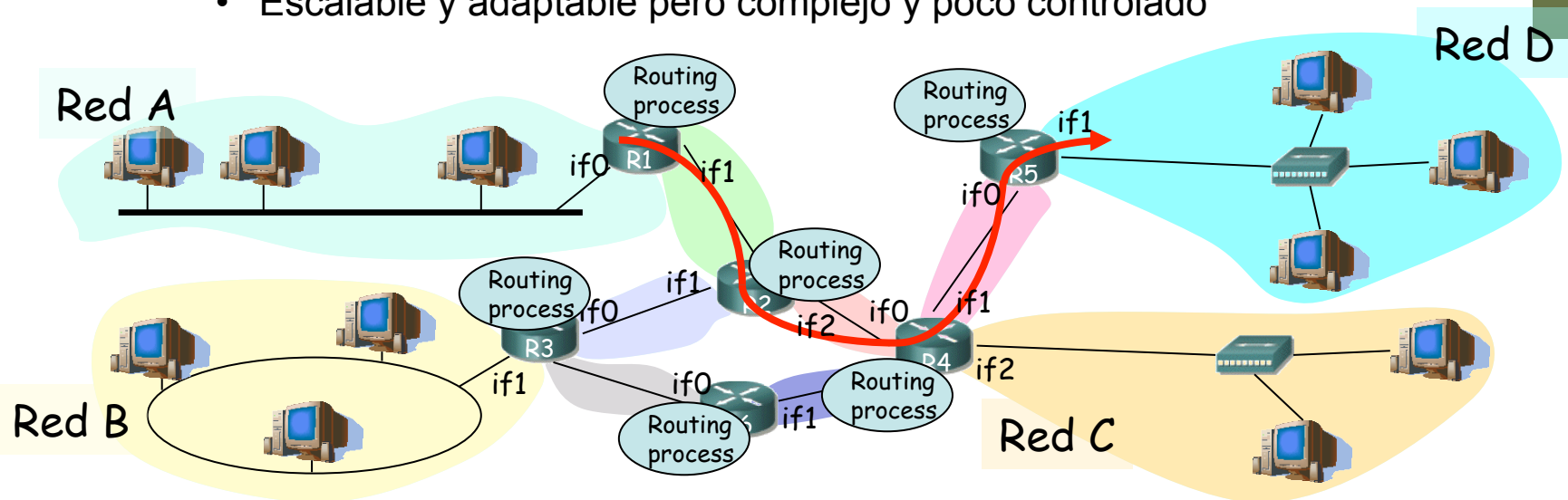
Objetivos

- Conocer los diferentes tipos básicos de protocolos/algoritmos de enrutamiento
- Conocer la estructura jerárquica de enrutamiento en la Internet

Introducción

- IP ofrece un servicio de datagramas
- Encamina salto a salto
- La tabla de rutas se puede especificar:
 - Estática
 - Configuración manual
 - Cambios lentos
 - Dinámica
 - Proceso en cada router
 - Cálculo distribuido
 - Escalable y adaptable pero complejo y poco controlado

Router R1	
Destino	Next-hop
Red A	IP de if1 de R1
Red B	IP de if0 de R3
Red C	IP de if0 de R4
...	...

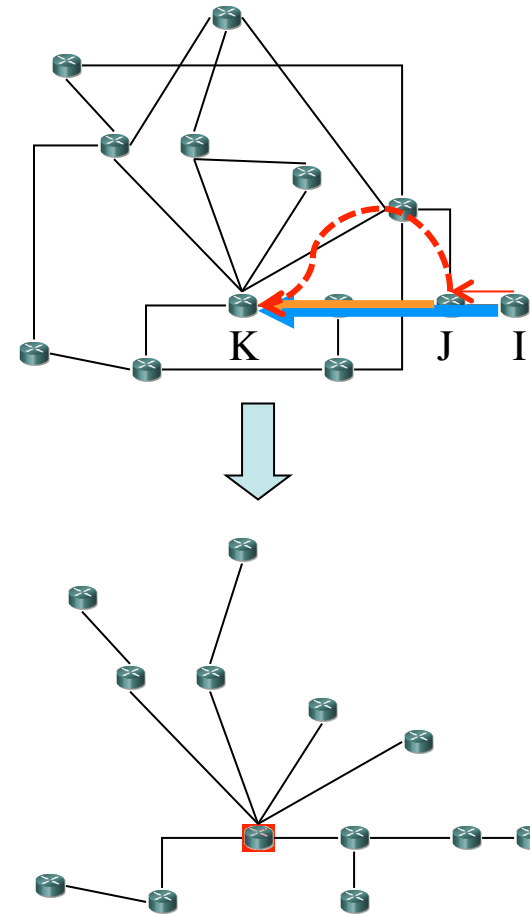


Diferentes alternativas

- **Proactivo** vs reactivo
 - Según se calculen las rutas ante tráfico o previamente
- Centralizado vs. **distribuido**
 - Centralizado es más simple pero no escala
- *Source-routing* vs. **salto a salto**
 - Source-routing camino en la cabecera (*loose* o *hard*)
- **Único camino** vs. múltiples caminos
 - Mantener un camino o varios por cada red destino.
- **Adaptativo** vs. no adaptativo
 - Según reaccione a cambios en la red
- Tráfico **unicast** vs. Multicast
- Información global o descentralizada
 - Según el conocimiento de la red que tengan los routers

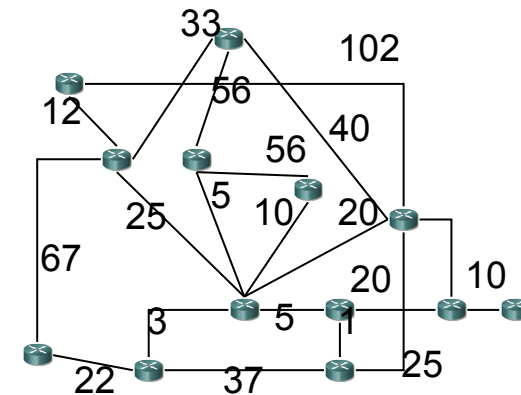
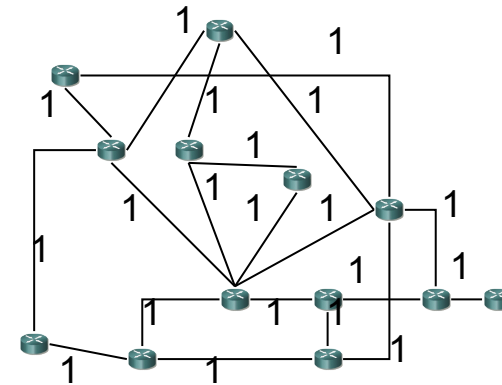
Principio de optimalidad

- Si router *J* está en el camino óptimo desde *I* a *K* entonces el camino óptimo de *J* a *K* está en la misma ruta (...)
- Si existiera una ruta mejor de *J* a *K* se podría concatenar con la de *I* a *J* (...)
- El conjunto de rutas óptimas a un destino es un árbol = **sink tree** (...)
- Árbol \Rightarrow sin lazos (*loops*)



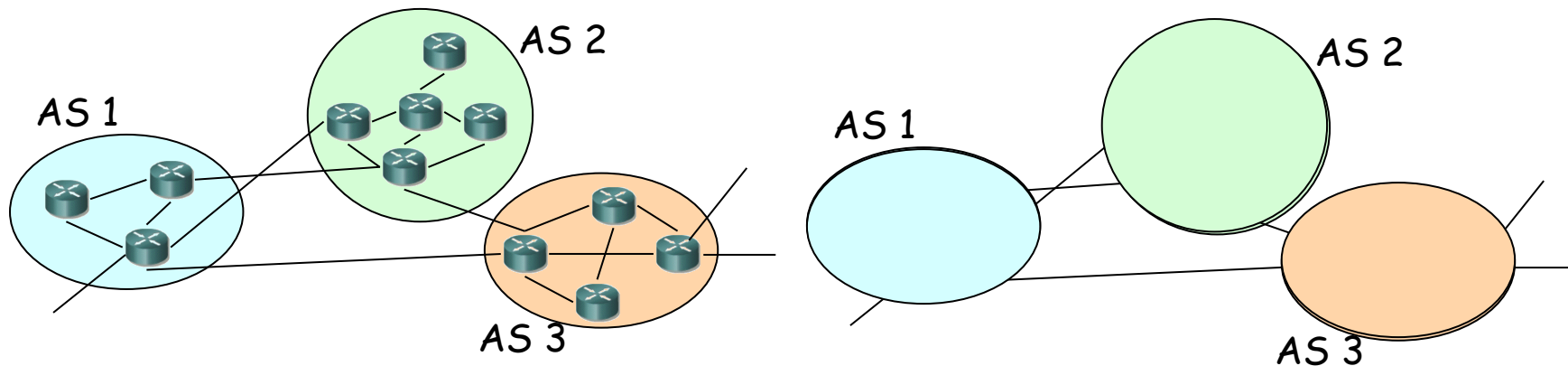
¿Camino óptimo?

- Ante una métrica aditiva: **Shortest paths** (*minimum spanning tree*)
- Métrica / pesos
 - Número de saltos (peso 1)
 - Distancia geográfica
 - Retardo de propagación
 - Longitud media de cola (retardo en cola)
 - Coste (€€)
- Métricas no aditivas
 - BW enlace
 - BW libre
 - Fiabilidad
- Combinación de métricas



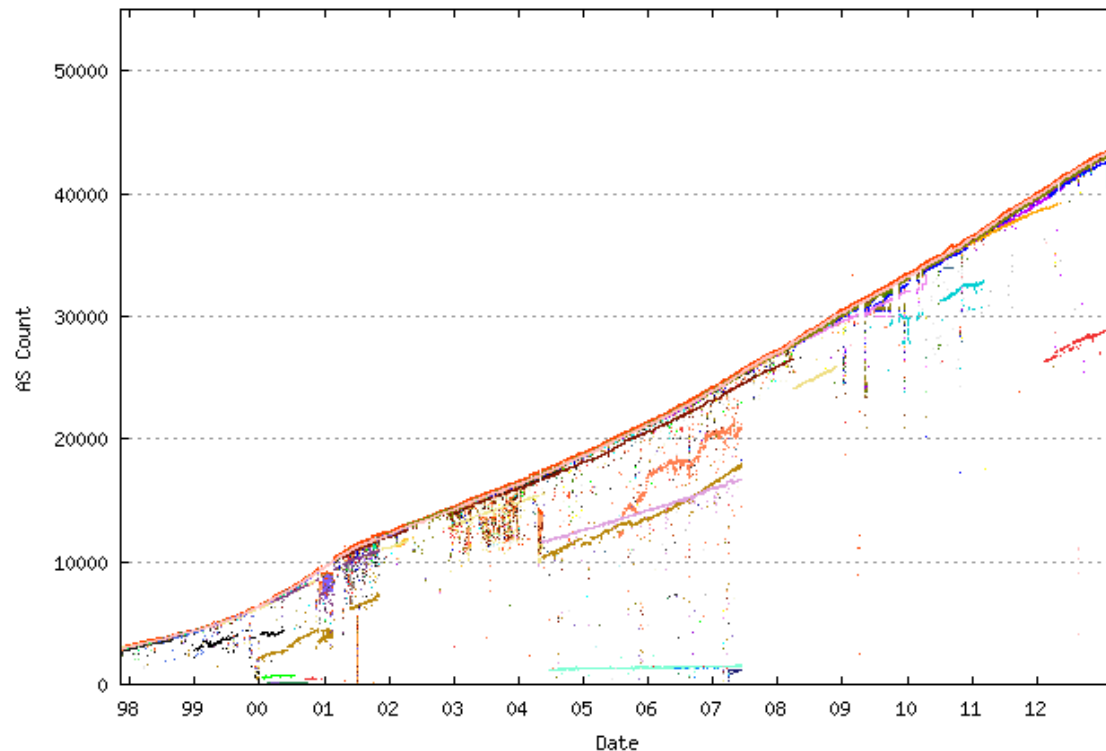
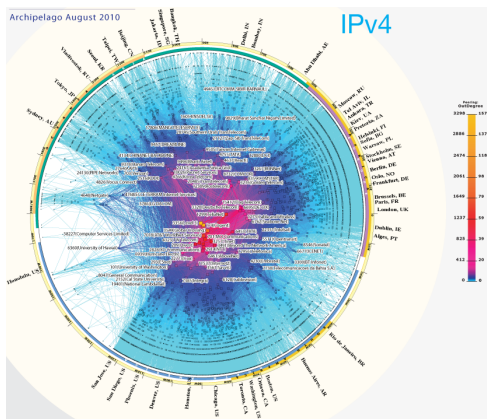
Enrutamiento jerárquico

- ¿Un solo grafo para toda la Internet?
 - Problemas de escala
 - Problemas de coordinación (¿métrica?)
- Enrutamiento jerárquico
 - IGP: Interior Gateway Protocol
 - EGP: Exterior Gateway Protocol
 - Interior/exterior respecto a “sistemas autónomos” (*Autonomous Systems*)
 - “An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy” (BCP 6)



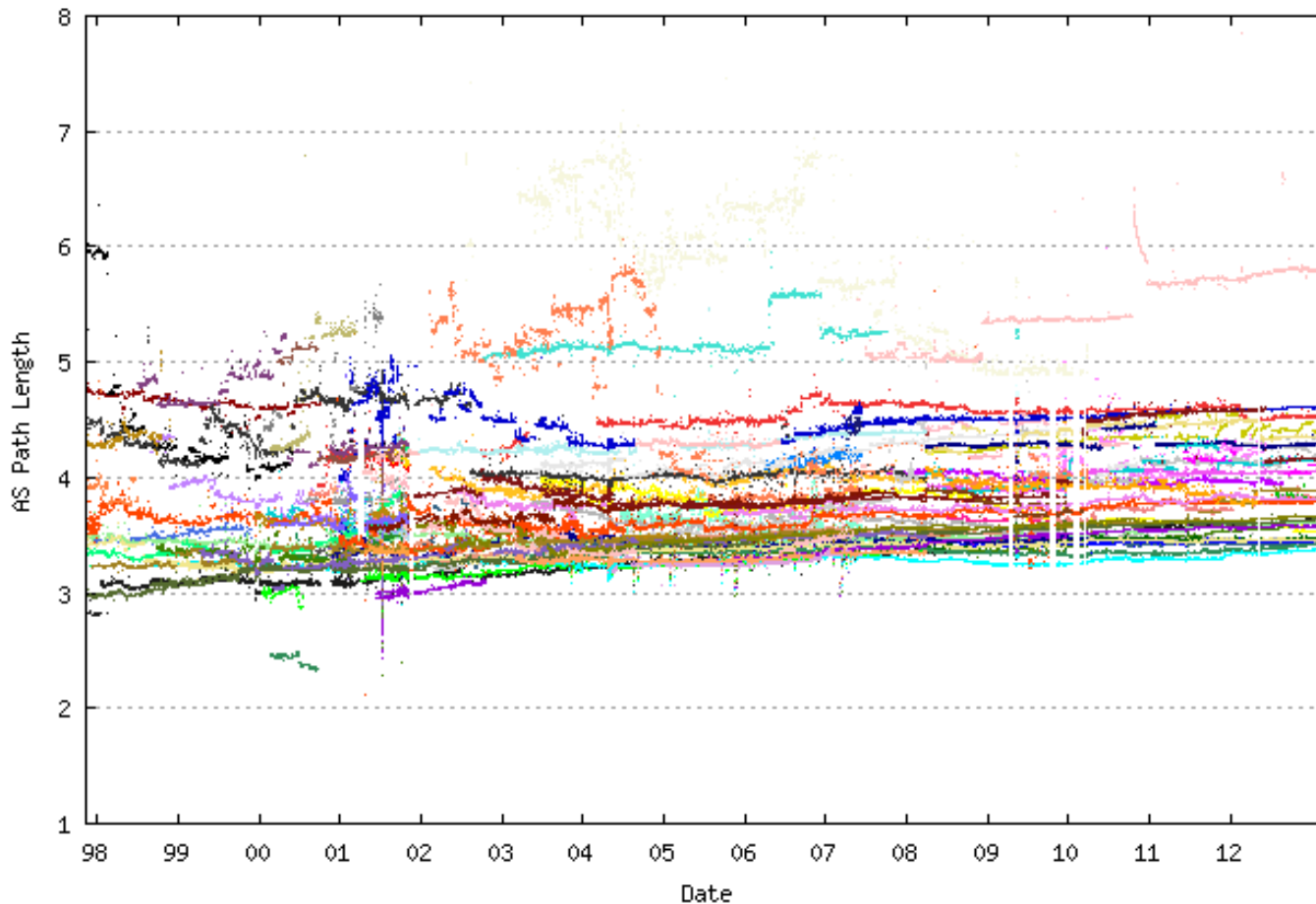
Sistemas Autónomos

- Cada AS calcula rutas internas de forma independiente
- Otorga autonomía a cada AS para elegir y configurar su protocolo de enrutamiento interior
- ASN = Autonomous System Number (16 ó 32 bits)
- Más de 60K+ ASNs asignados (2013), 40K+ anunciados
- Un ISP puede tener asignado uno o más ASNs



Longitud del camino exterior

- Ha crecido el número de ASs pero no las distancias entre ellos (en nº de ASs atravesados)
- Ha crecido el grado de conectividad



ASNs: Ejemplos

```
$ whois -h whois.cymru.com 130.206.164.68
AS      | IP                | AS Name
766     | 130.206.164.68   | REDIRIS Entidad Publica Empresarial Red.es
```

```
$ whois -h whois.cymru.com 169.229.216.200
AS      | IP                | AS Name
25      | 169.229.216.200 | UCB - University of California at Berkeley
```

```
$ whois -h whois.cymru.com 72.21.194.212
AS      | IP                | AS Name
16509   | 72.21.194.212   | AMAZON-02 - Amazon.com, Inc.
```

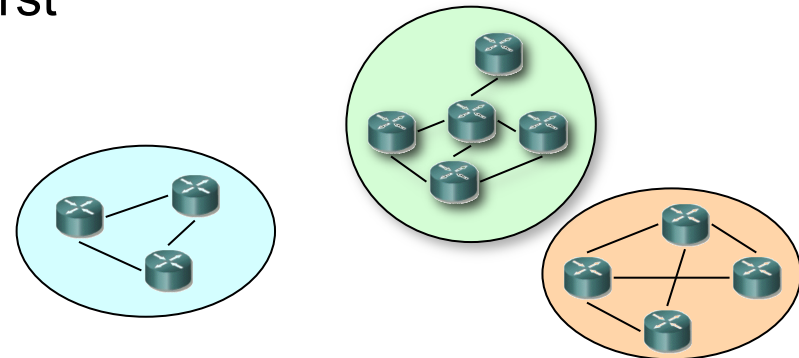
```
$ whois -h whois.cymru.com 8.8.8.8
AS      | IP                | AS Name
15169   | 8.8.8.8          | GOOGLE - Google Inc.
```

```
$ whois -h whois.cymru.com 81.47.192.13
AS      | IP                | AS Name
3352    | 81.47.192.13    | TELEFONICA-DATA-ESPANA TELEFONICA DE ESPANA
```

```
$ whois -h whois.cymru.com 187.8.1.1
AS      | IP                | AS Name
10429   | 187.8.1.1       | Telefonica Data S.A.
```

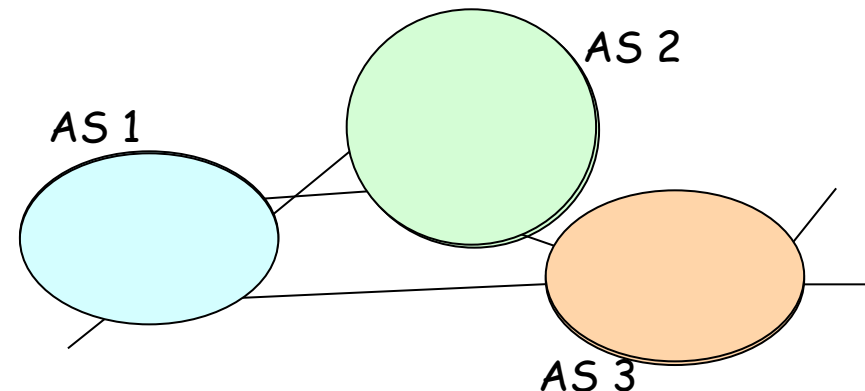
Interior Gateway Protocols (IGP)

- **Características:**
 - Simples
 - Calculan caminos eficientes respecto a una métrica
 - Recalculan rápidamente ante cambios
 - No escalan bien para redes grandes
- **Los más comunes:**
 - *RIP*: Routing Information Protocol
 - *(E)IGRP*: (Enhanced) Interior Gateway Routing Protocol (propietario de Cisco)
 - *IS-IS*: Intermediate System to Intermediate System (OSI)
 - *OSPF*: Open Shortest Path First



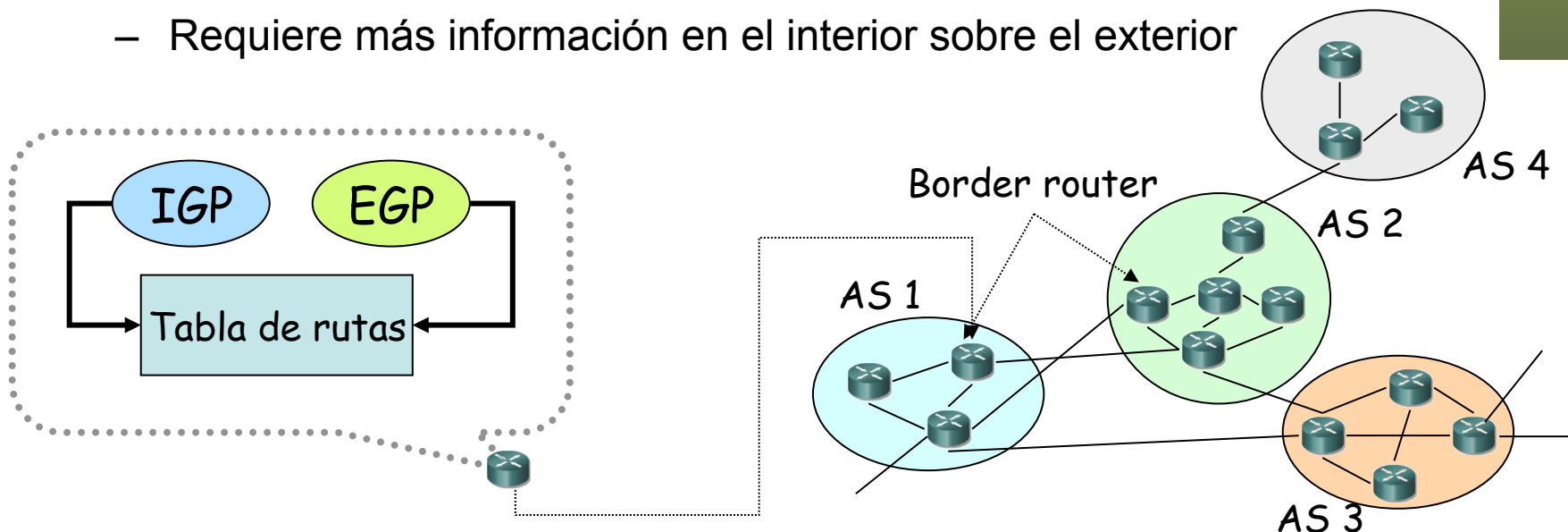
Exterior Gateway Protocols (EGP)

- **Características:**
 - Mejor escalabilidad
 - Habilidad para agregar rutas
 - Habilidad para expresar políticas
 - Mayor carga en el router
- **BGP (Border Gateway Protocol):**
 - *Estándar de facto*
 - Algoritmo *path-vector* : anuncia el camino completo al destino (como una secuencia de ASs)
 - Los anuncios emplean conexiones TCP entre los routers



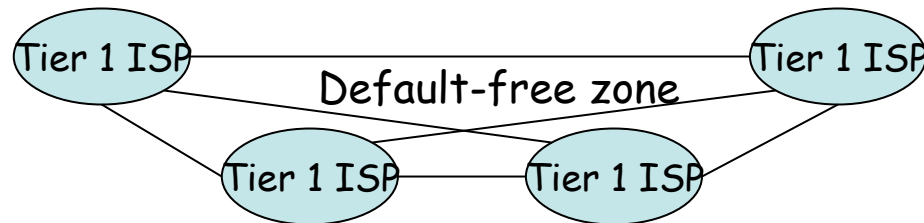
Routers frontera

- En ellos la tabla de rutas es configurada por ambos protocolos
- IGP: rutas a destinos internos, EGP: rutas a destinos externos
- Si hay un solo enlace al exterior: routers internos solo necesitan una ruta por defecto hacia otros ASs
- Más de un enlace al exterior
 - Red multi-homed o de tránsito
 - Cada router interno necesita rutas para cada red externa para poder elegir la mejor salida del AS
 - O se puede mandar todo a la misma (la más cercana)
 - Requiere más información en el interior sobre el exterior



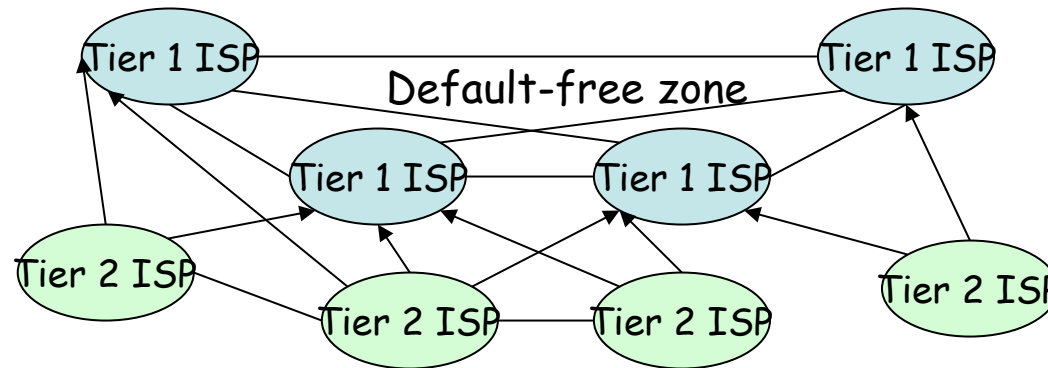
Estructura de Internet

- **Tier-1 ISPs o Internet backbone networks**
 - Grandes proveedores internacionales (no claro cuáles, AT&T, NTT, Telefónica, Sprint, Verizon, etc., aproximadamente una docena)
 - Conexión completamente mallada (*peering agreements*)
 - No pagan a nadie por el tránsito
 - No emplean “ruta por defecto”, tienen rutas a todas las redes (2013: más de 400K rutas)



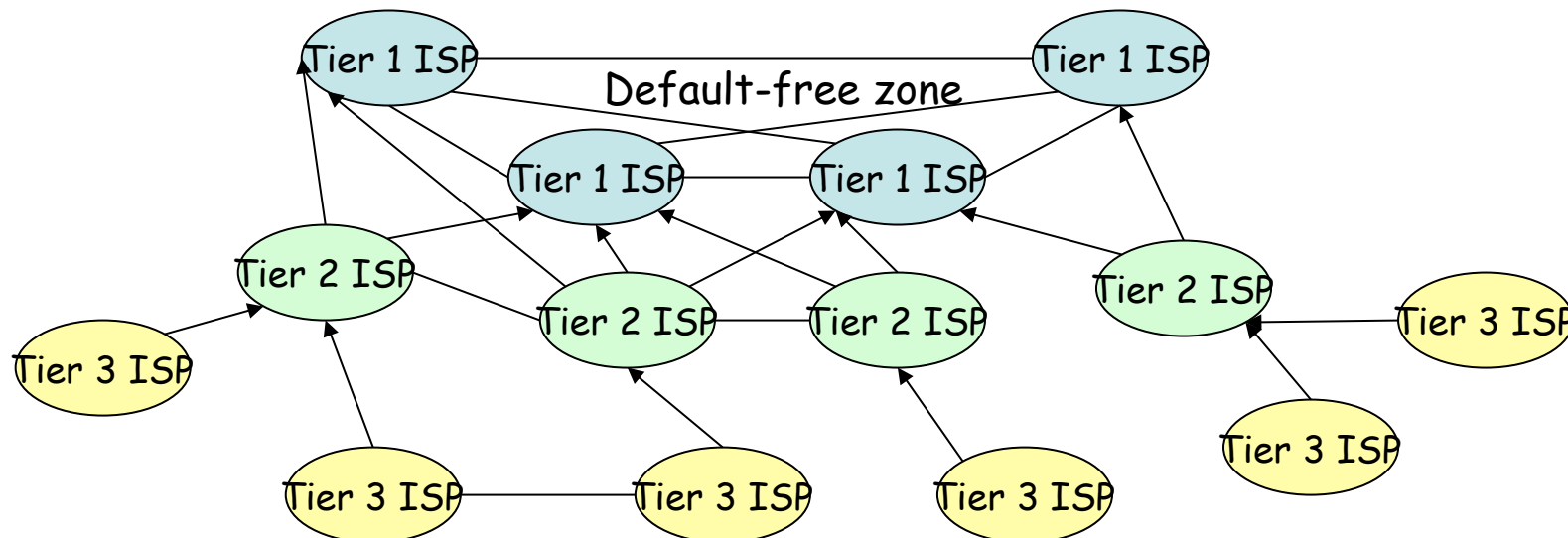
Estructura de Internet

- **Tier-2 ISPs**
 - Regionales o nacionales
 - Contratan tránsito a unos pocos tier-1 ISPs (ellos son los clientes y el tier-1 el proveedor de tránsito)
 - Pueden establecer *peering agreements* con otros tier-2



Estructura de Internet

- **Tier-3 ISPs**
 - ISPs locales de acceso
 - Contratan tránsito a uno o más tier-2 y pueden hacer *peering agreements* entre ellos

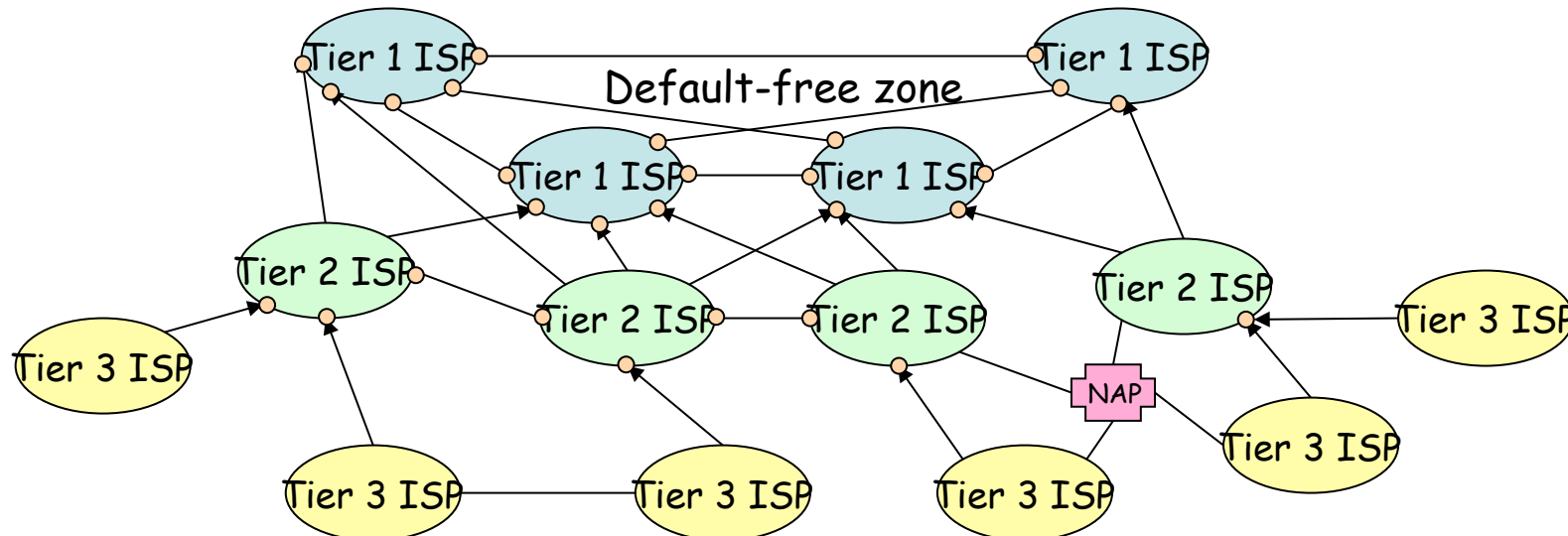
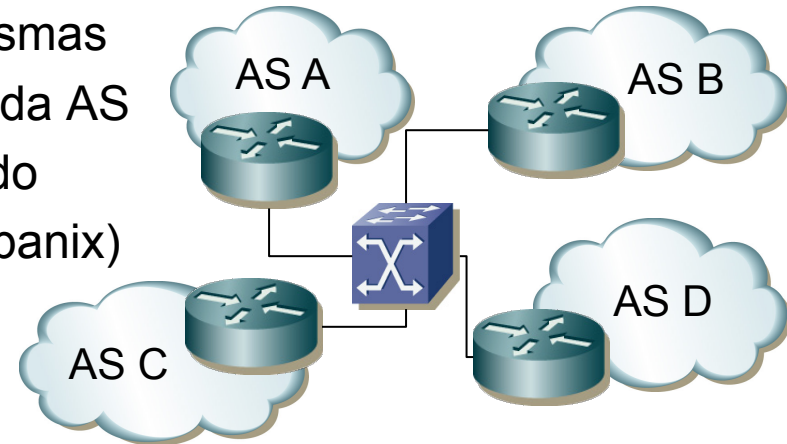


Estructura de Internet

Points of Presence (POPs)

NAPs (Network Access Points) o IXP (Internet eXchange Point)

- Son redes de alta velocidad en sí mismas
- Por ejemplo switch L2 + router de cada AS
- Pretenden ahorrar €€ y reducir retardo
- Mantener local el tráfico local (ej: Espanix)



Algoritmos de Enrutamiento

Link State:

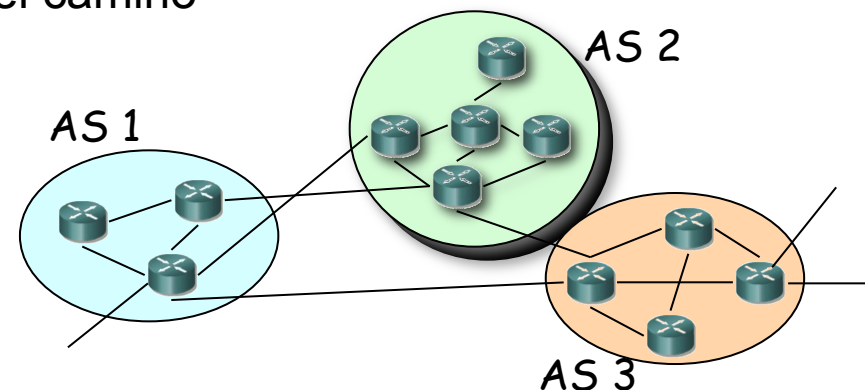
- Comunican qué vecinos tienen y el coste
- Inundan la red
- Cada nodo conoce la topología entera
- Protocolos: OSPF, IS-IS, PNNI...

Distance Vector:

- Comunican estimación de distancia a destinos
- Informan a vecinos
- Protocolos: RIP, IGRP, DECnet RP, DSDV, IPX-RIP...

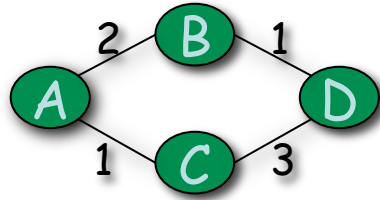
Path Vector:

- Comunican estimación de caminos preferidos a destinos
- Se evitan bucles evitando repetir en el camino
- Protocolos: BGP



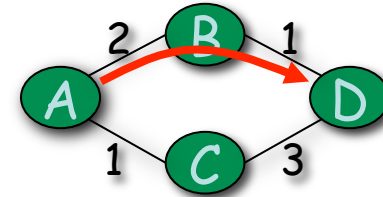
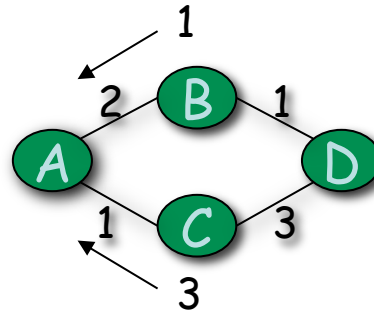
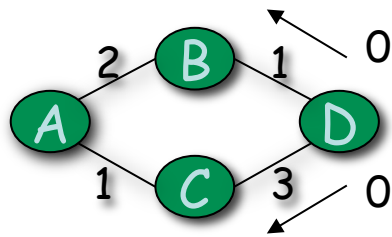
Algoritmos de Enrutamiento

Link State

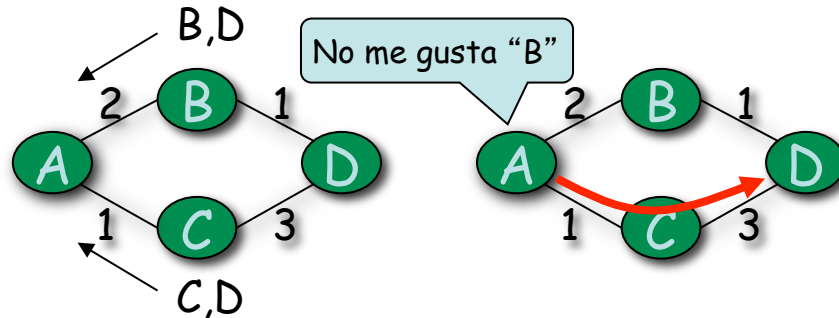
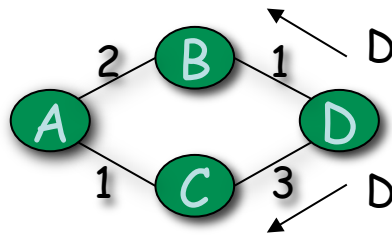


A: [B, 2], [C, 1]
 B: [A, 2], [D, 1]
 C: [A, 1], [D, 3]
 D: [B, 1], [C, 3]

Distance Vector



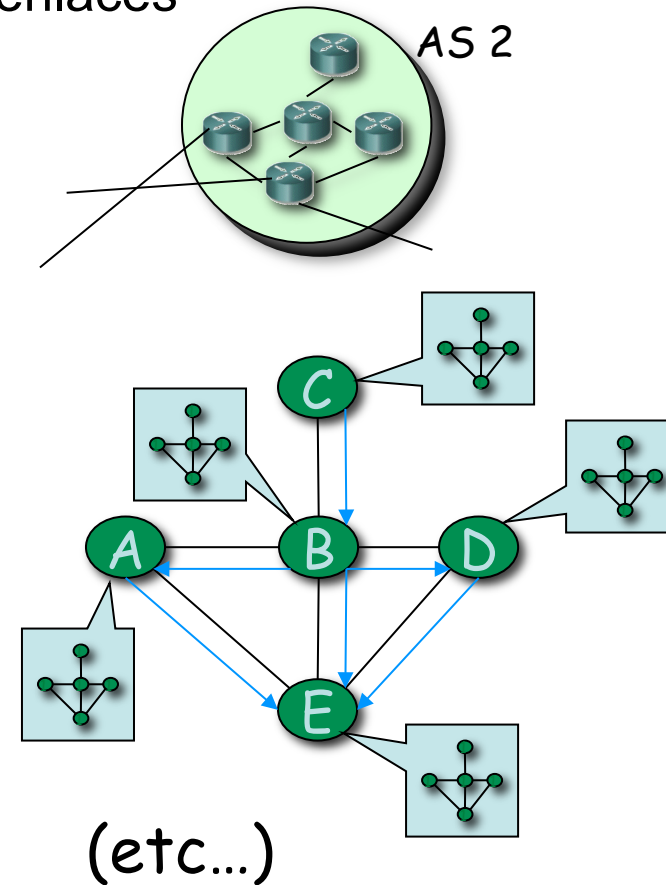
Path Vector



Link State

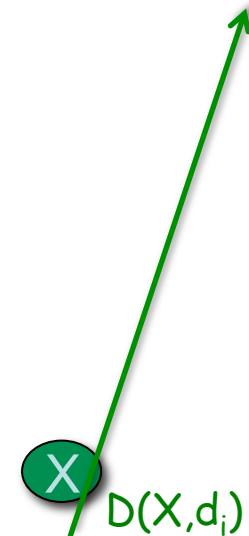
Tres pasos

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*



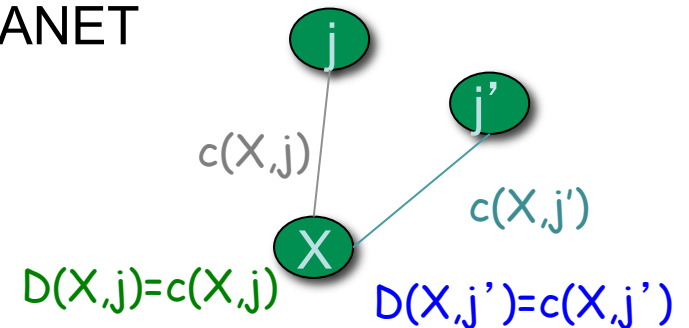
Distance Vector

- Cada nodo X 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



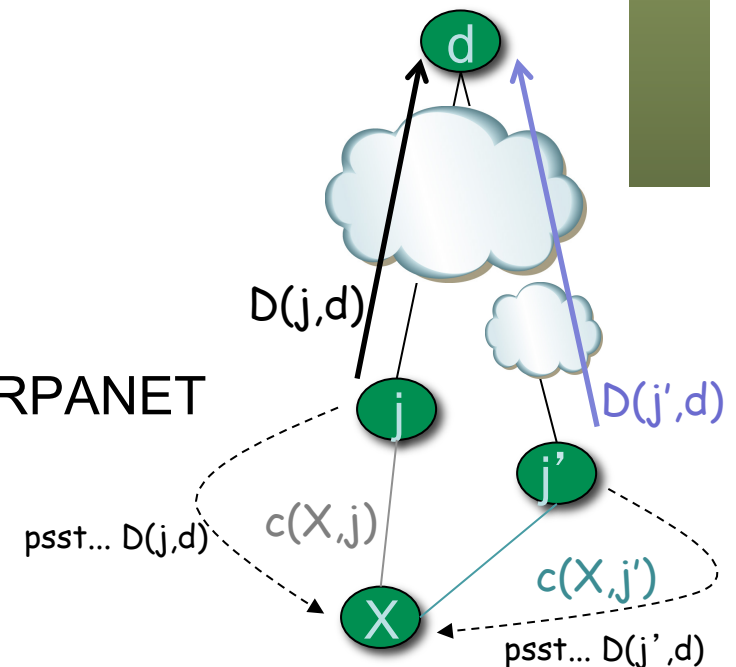
Distance Vector

- Cada nodo X 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



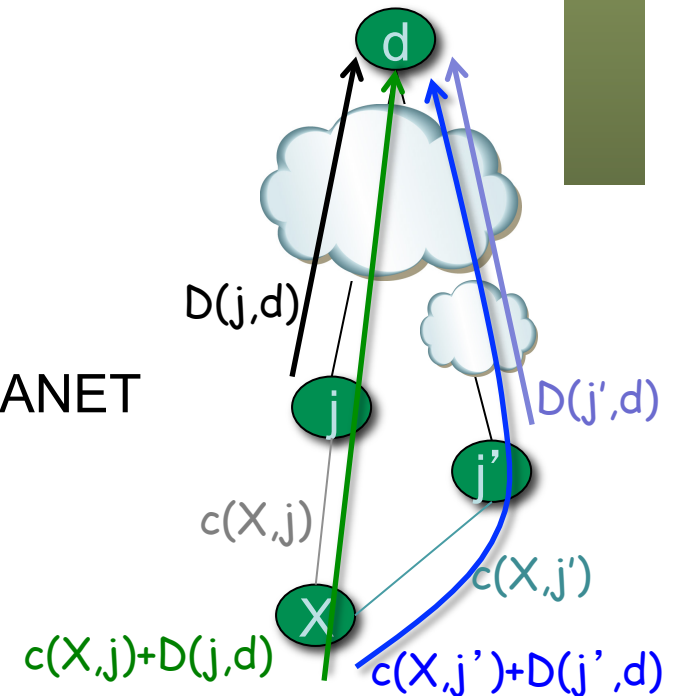
Distance Vector

- Cada nodo X 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
 ($D(X, d_1), D(X, d_2), D(X, d_3), D(X, d_4), \dots$)
 - 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



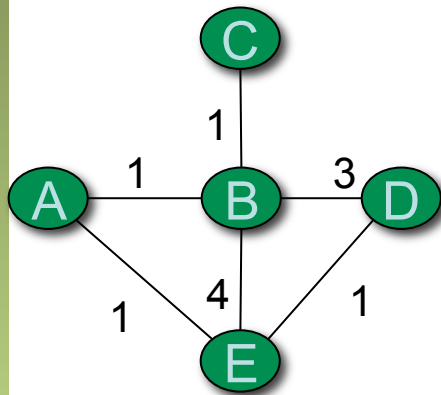
Distance Vector

- Cada nodo X 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
 ($D(X,d_1), D(X,d_2), D(X,d_3), D(X,d_4), \dots$)
 - 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



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

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

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

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

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

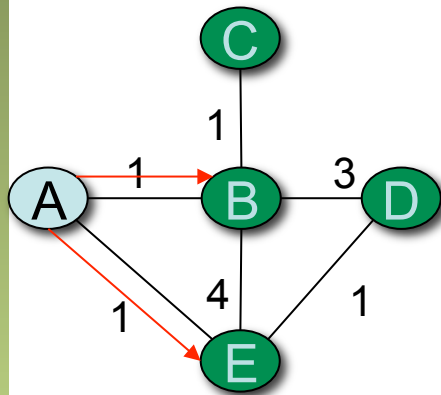
Algoritmo de Bellman-Ford

A envía

$$D(E,d) \leftarrow \min\{c(E,A)+D(A,d)\}$$

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

(...)



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

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

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

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

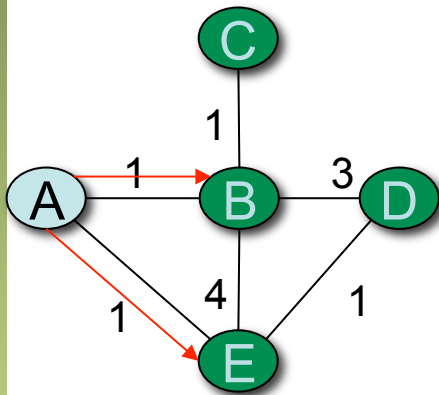
Dest	Next	Cost
A	A	1
B	B	4
C	-	∞
D	D	1

Algoritmo de Bellman-Ford

A envía

$$D(E,d) \leftarrow \min\{c(E,A)+D(A,d)\}$$

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



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

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

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

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

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

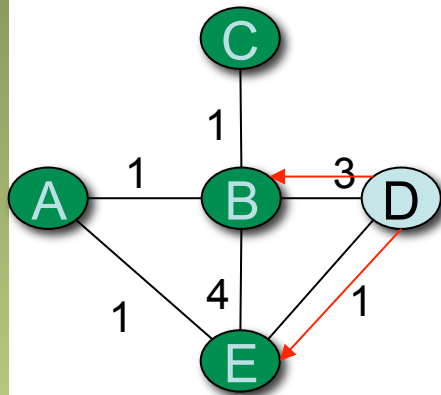
Algoritmo de Bellman-Ford

D envía

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

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

No hay cambios



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

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

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

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

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

Algoritmo de Bellman-Ford

B envía

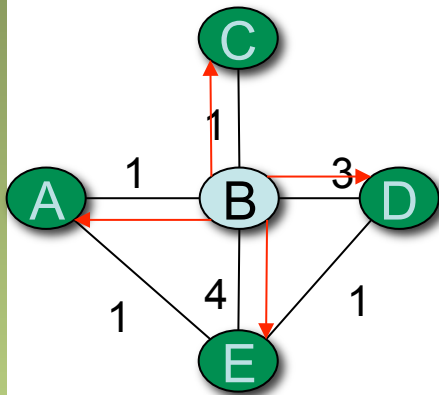
$$D(A,d) \leftarrow \min\{c(A,B)+D(B,d)\}$$

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

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

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

(...)



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

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

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

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

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

Algoritmo de Bellman-Ford

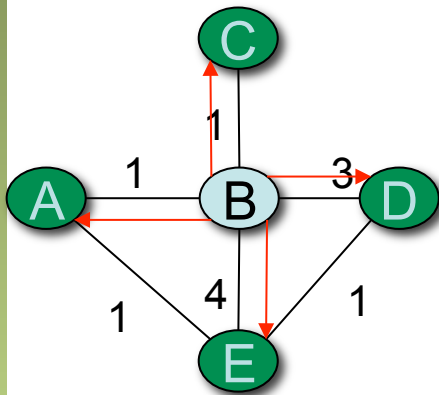
B envía

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

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

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

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



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

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

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

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

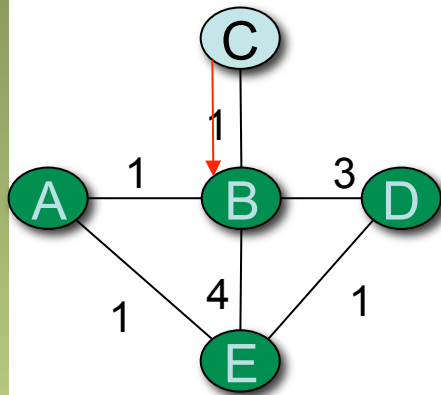
Dest	Next	Cost
A	A	1
B	A	2
C	B (-)	5 (∞)
D	D	1

Algoritmo de Bellman-Ford

C envía

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

No hay cambios



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

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

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

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

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

Algoritmo de Bellman-Ford

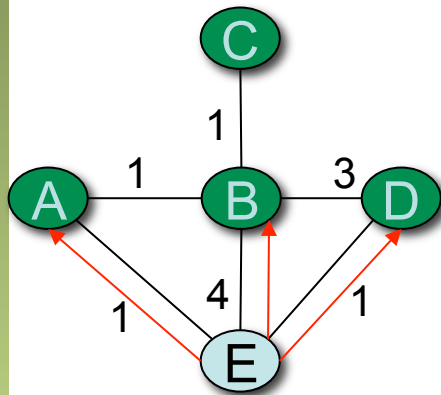
E envía

$$D(A,d) \leftarrow \min\{c(A,E)+D(E,d)\}$$

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

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

(...)



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

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

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

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

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

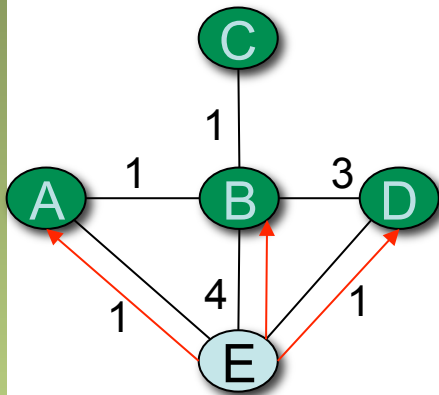
Algoritmo de Bellman-Ford

E envía

$$D(A,d) \leftarrow \min\{c(A,E)+D(E,d)\}$$

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

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



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

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

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

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

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

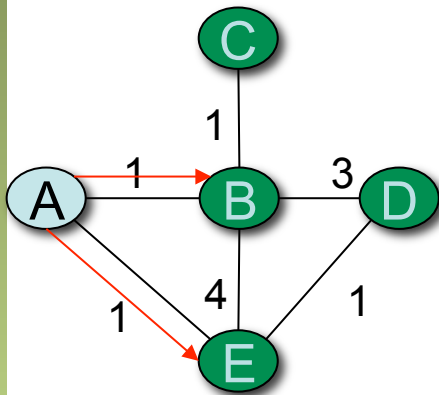
Algoritmo de Bellman-Ford

A envía

$$D(E,d) \leftarrow \min\{c(E,A)+D(A,d)\}$$

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

(...)



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

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

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

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

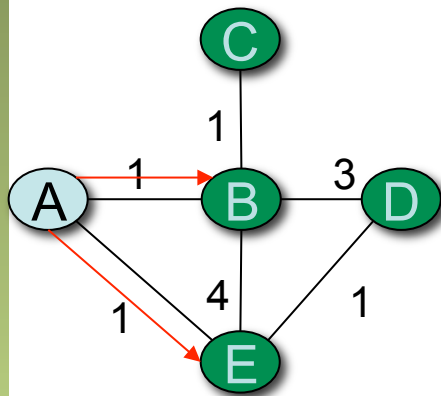
Dest	Next	Cost
A	A	1
B	A	2
C	B	5
D	D	1

Algoritmo de Bellman-Ford

A envía

$$D(E,d) \leftarrow \min\{c(E,A)+D(A,d)\}$$

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



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

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

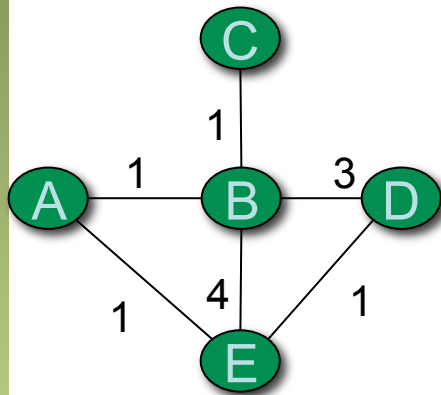
Dest	Next	Cost
A	B	2
B	B	1
D	B	4
E	B	3

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

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

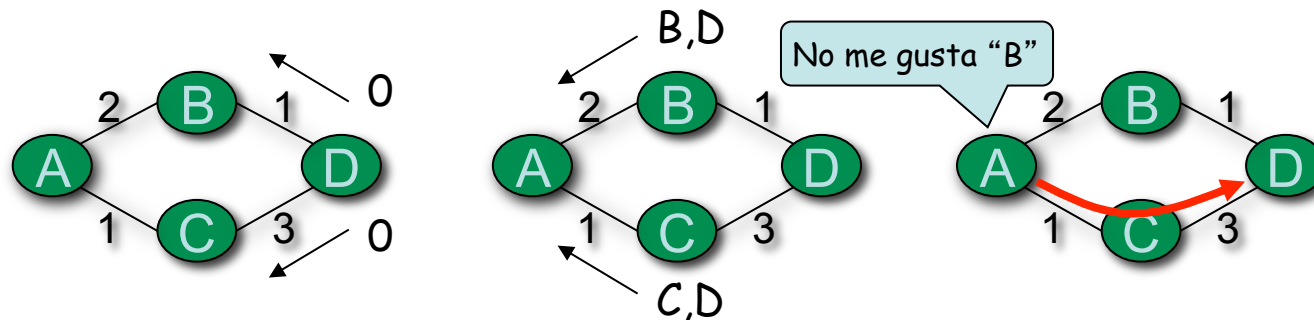


Dest	Next	Cost	Dest	Next	Cost
B	B	1	A	A	1
C	B	2	C	C	1
D	E	2	D	D	3
E	E	1	E	A	2

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

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



Resumen

- La estructura de Internet es jerárquica
- Esto lleva a un enrutamiento jerárquico dividido en:
 - Intradomain
 - Interdomain
 - Diferente problemática para ambos
- Algoritmos empleados por los protocolos
 - Distance-vector
 - Link-State
 - Path-vector