

uc3m

Universidad  
**Carlos III**  
de Madrid

# *RIP: Routing Information Protocol*

Redes y Servicios de Comunicaciones Avanzadas

Departamento de Ingeniería Telemática

Carlos J. Bernardos

[cjbc@it.uc3m.es](mailto:cjbc@it.uc3m.es)

Manuel Urueña

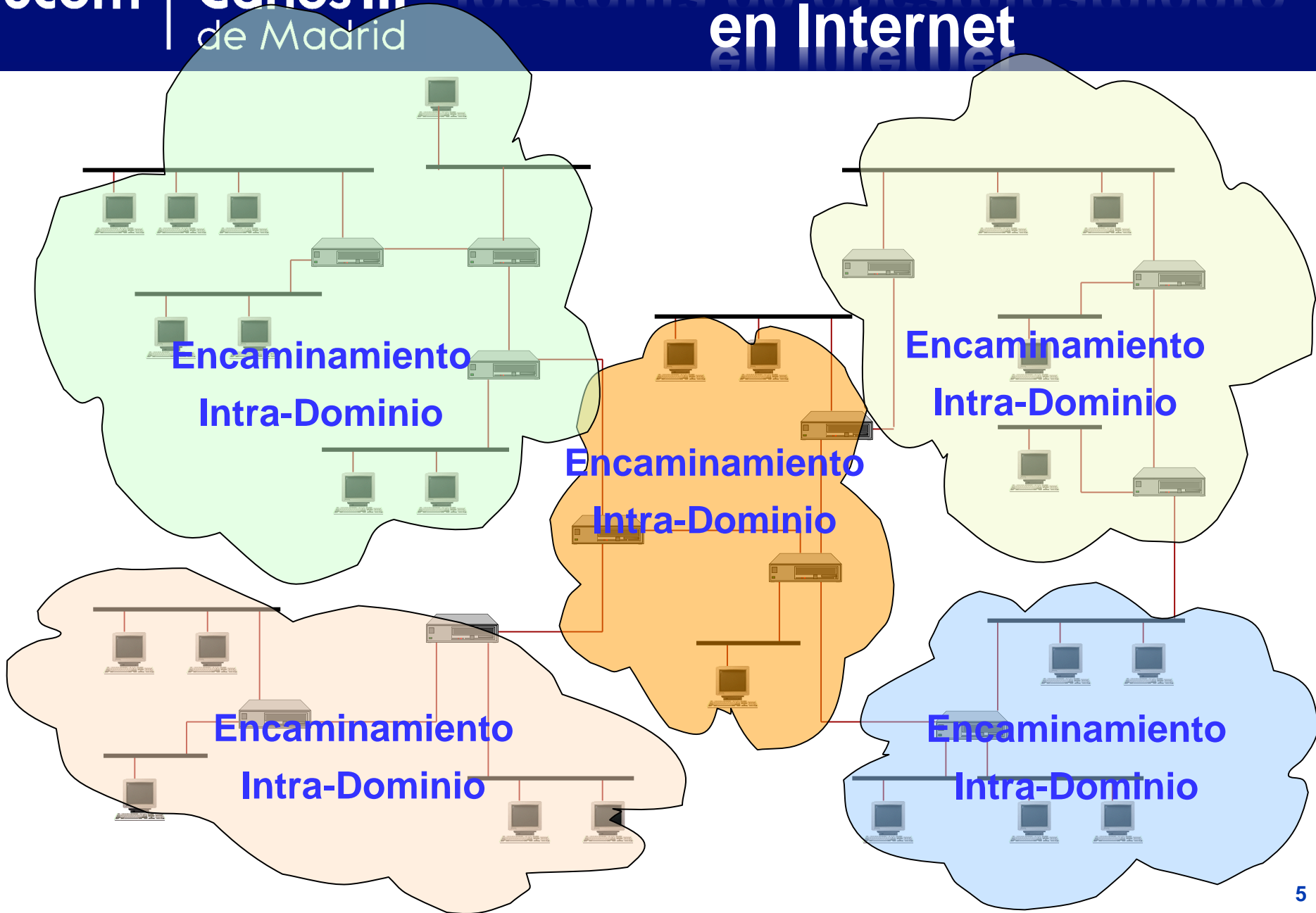
[muruenya@it.uc3m.es](mailto:muruenya@it.uc3m.es)

- 1. Protocolos de encaminamiento**
- 2. Jerarquía de encaminamiento**
- 3. El protocolo RIP**

- ◆ **El incremento de tamaño de las redes hace más necesaria la utilización de protocolos de encaminamiento dinámico.**
- ◆ **Un protocolo de encaminamiento se compone de dos fases:**
  1. **Comunicación con otros *routers*.**
  2. **Utilización de la información obtenida para la actualización de la tabla de encaminamiento.**
- ◆ **Es importante que todo el proceso de encaminamiento no sobrecargue en exceso la red.**

- ◆ **El encaminamiento dinámico presenta grandes ventajas frente al estático; entonces, ¿por qué se sigue utilizando el encaminamiento estático?**
  - ❖ **Para controlar las rutas empleadas.**
  - ❖ **No resulta adecuado en enlaces por marcación (*dial-up links*).**
  - ❖ **Los enlaces de baja capacidad pueden verse desbordados.**
  - ❖ **Por seguridad.**

# Jerarquía de encaminamiento en Internet





The diagram illustrates the hierarchy of routing in the Internet. It features five cloud-shaped regions representing different domains, arranged around a central orange cloud. The central orange cloud is labeled "Encaminamiento Inter-Dominio" in blue text. Four red lines radiate from this central cloud to the four surrounding clouds: a green cloud (top-left), a yellow-green cloud (top-right), a light orange cloud (bottom-left), and a light blue cloud (bottom-right). This represents the flow of traffic between different administrative domains through a central inter-domain routing structure.

**Encaminamiento Inter-Dominio**

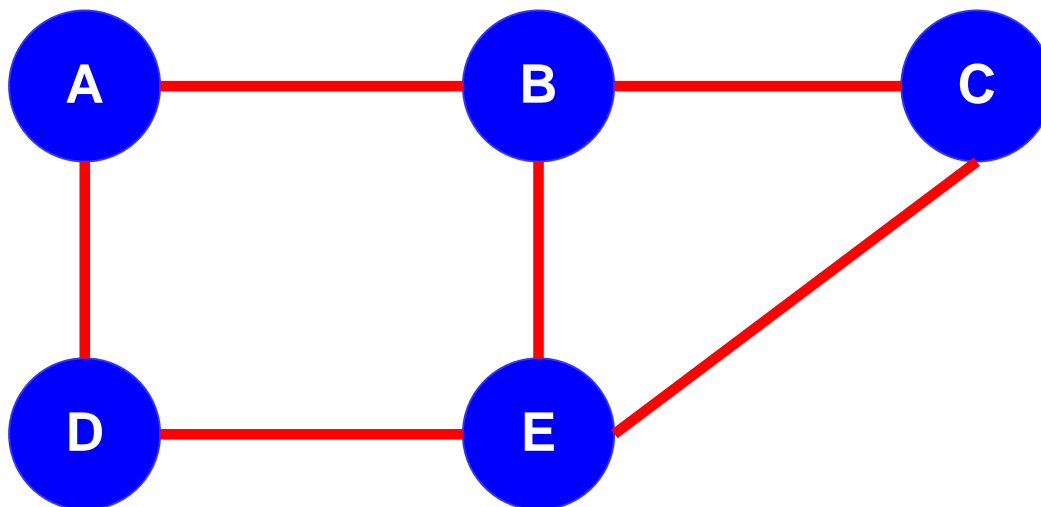
- ◆ Los protocolos de encaminamiento intra-dominio (también conocidos como IGP: *Interior Gateway Protocol*) se clasifican principalmente en 2 familias:
  - ❖ Protocolos de **Vector-Distancia**:
    - ✓ Ej. RIP
  - ❖ Protocolos de **Estado-de-Enlaces**:
    - ✓ Ej. OSPF, IS-IS

Protocolos de Vector-Distancia:

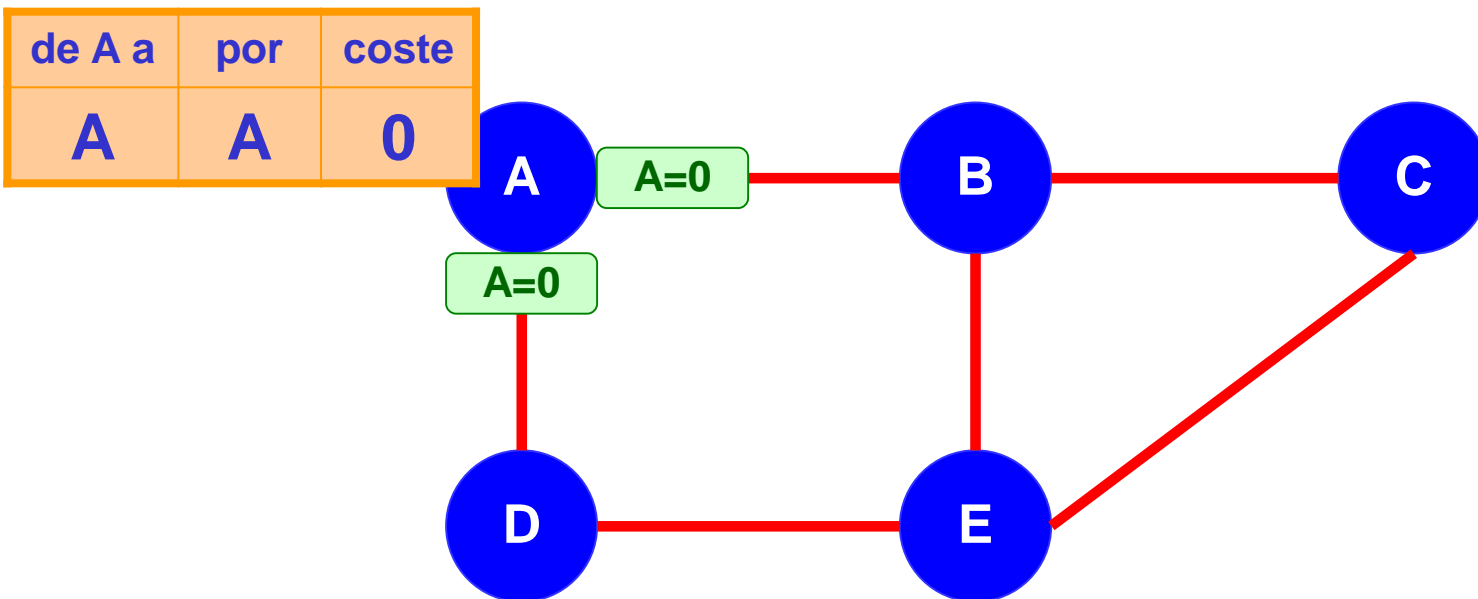
**RIP**



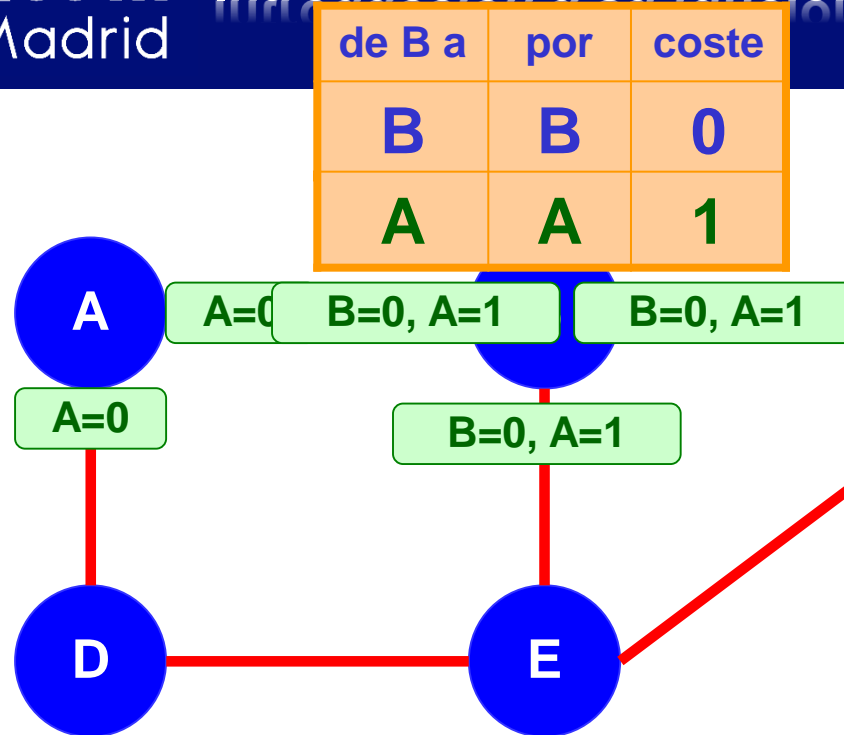
- ◆ **El *Routing Information Protocol* (RIP) es el protocolo de Vector-Distancia más conocido.**
- ◆ **Orígenes:**
  - ❖ **Conocidos como protocolos “Bellman-Ford”.**
  - ❖ **Utilizados en varios tipos de redes de paquetes:**
    - ✓ **ARPANET y Cyclades.**
    - ✓ **XNS *Routing Information Protocol* (de Xerox).**
    - ✓ **Internet RIP desarrollado como un componente del código del UNIX BSD, inspirado en XNS RIP.**



- Supongamos que cada *router* tiene una subred asociada del mismo nombre.
  - En RIP se anuncian subredes no *routers*
- Suponemos enlaces simétricos.
- Coste idéntico de todos los enlaces = 1



- **Funcionamiento al arranque:**
  - **Sólo es necesario tener conocimiento local:**
    - ◆ Los nodos conocen sus condiciones locales...
    - ◆ ...pero desconocen la topología de la red.
  - **Vector-Distancia inicial de A:  $A=0$**



¿Destino desconocido?

• **SÍ:** Añado ruta, con coste = (coste recibido + coste del enlace)

• **NO:** ¿(coste recibido + coste del enlace)  $\geq$  coste actual?

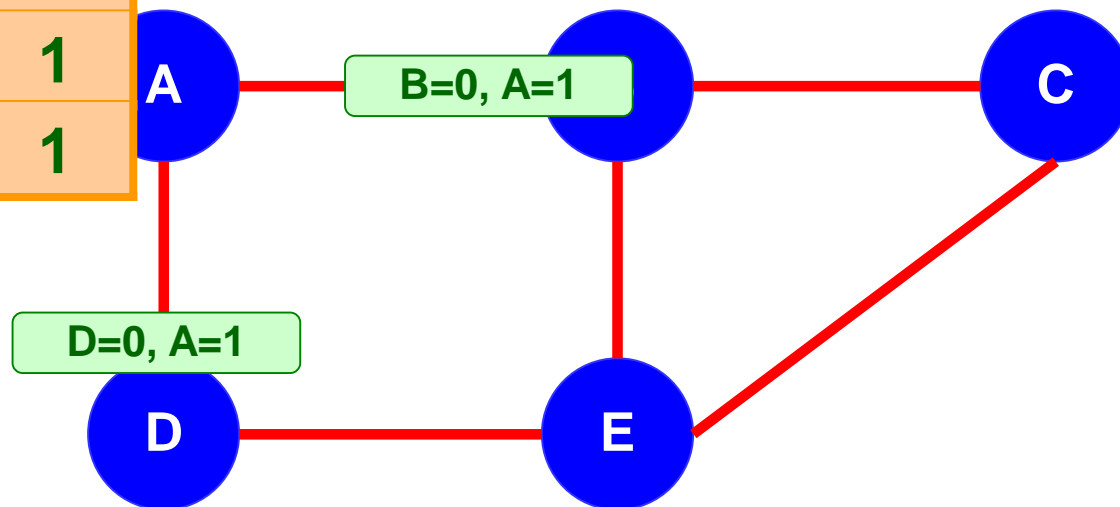
• **NO:** Añado ruta

• **SÍ:** Ignoro ruta

## ● Cada nodo:

- ◆ Recibe Vector-Distancia (VD) de sus vecinos.
- ◆ Recalcula su tabla de rutas local.
- ◆ Transmite su Vector-Distancia a sus vecinos.

| de A a | por | coste |
|--------|-----|-------|
| A      | A   | 0     |
| B      | B   | 1     |
| D      | D   | 1     |



| de D a | por | coste |
|--------|-----|-------|
| D      | D   | 0     |
| A      | A   | 1     |

- Llega un momento en que la recepción de un Vector-Distancia no genera cambios en la tabla de rutas local de los nodos:
  - ◆ **Convergencia** inicial del protocolo.

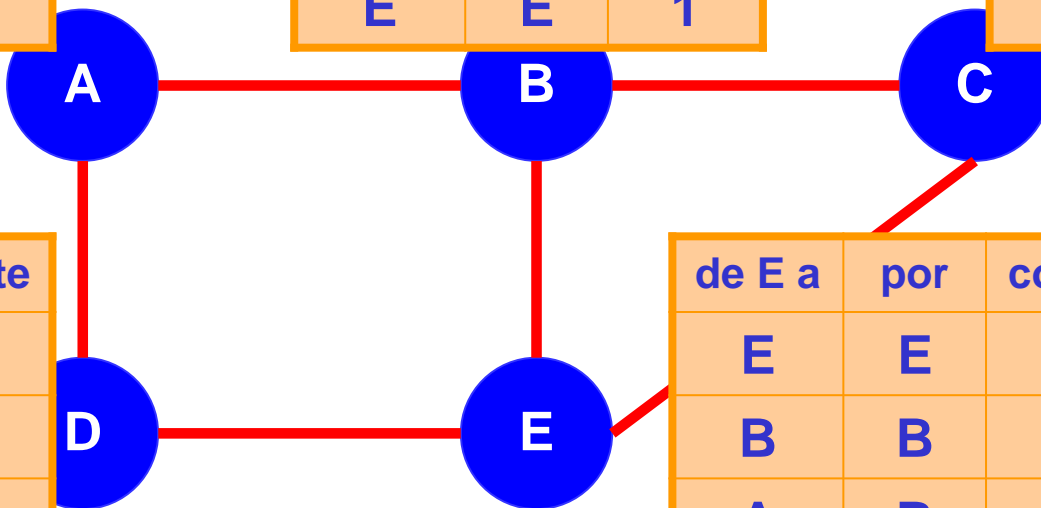
# Protocolos de Vector-Distancia

## Introducción a su funcionamiento: "Cold start"

| de A a | por | coste |
|--------|-----|-------|
| A      | A   | 0     |
| B      | B   | 1     |
| D      | D   | 1     |
| C      | B   | 2     |
| E      | B   | 2     |

| de B a | por | coste |
|--------|-----|-------|
| B      | B   | 0     |
| A      | A   | 1     |
| D      | A   | 2     |
| C      | C   | 1     |
| E      | E   | 1     |

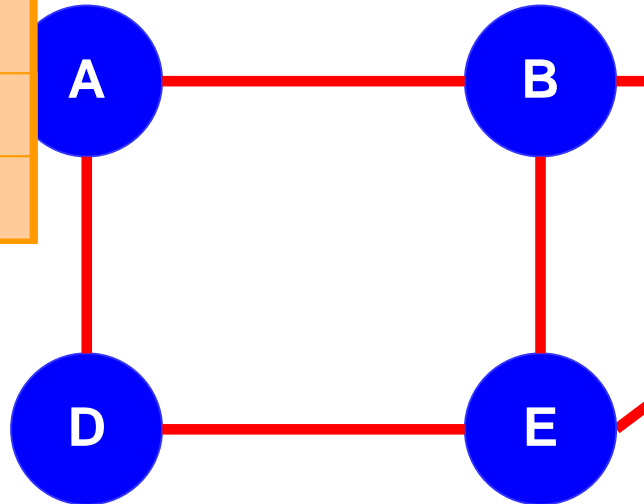
| de C a | por | coste |
|--------|-----|-------|
| C      | C   | 0     |
| B      | B   | 1     |
| A      | B   | 2     |
| E      | E   | 1     |
| D      | E   | 2     |



| de D a | por | coste |
|--------|-----|-------|
| D      | D   | 0     |
| A      | A   | 1     |
| B      | A   | 2     |
| E      | E   | 1     |
| C      | E   | 2     |

| de E a | por | coste |
|--------|-----|-------|
| E      | E   | 0     |
| B      | B   | 1     |
| A      | B   | 2     |
| D      | D   | 1     |
| C      | C   | 1     |

| de A a | por | coste |
|--------|-----|-------|
| A      | A   | 0     |
| B      | B   | inf   |
| D      | D   | 1     |
| C      | B   | inf   |
| E      | B   | inf   |



| de B a | por | coste |
|--------|-----|-------|
| B      | B   | 0     |
| A      | A   | inf   |
| D      | A   | inf   |
| C      | C   | 1     |
| E      | E   | 1     |

- Se “rompe” el enlace A-B:
  - A y B descubren la rotura/fallo en el enlace:
    - ❖ Por indicación de nivel 2: *Link-down*.
    - ❖ Por *timeout* de mensajes periódicos.
  - Establecen el coste para llegar a los destinos que utilizaban ese enlace a infinito ( $\infty$ ).

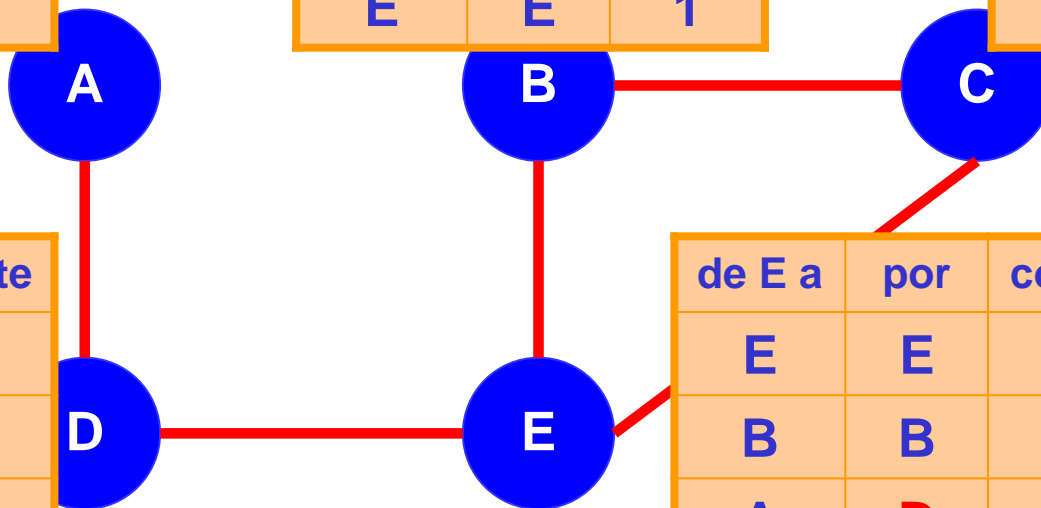
# Protocolos de Vector-Distancia

## Introducción a su funcionamiento: Fallo enlace

| de A a | por | coste |
|--------|-----|-------|
| A      | A   | 0     |
| B      | D   | 3     |
| D      | D   | 1     |
| C      | D   | 3     |
| E      | D   | 2     |

| de B a | por | coste |
|--------|-----|-------|
| B      | B   | 0     |
| A      | E   | 3     |
| D      | E   | 2     |
| C      | C   | 1     |
| E      | E   | 1     |

| de C a | por | coste |
|--------|-----|-------|
| C      | C   | 0     |
| B      | B   | 1     |
| A      | E   | 3     |
| E      | E   | 1     |
| D      | E   | 2     |



| de D a | por | coste |
|--------|-----|-------|
| D      | D   | 0     |
| A      | A   | 1     |
| B      | E   | 2     |
| E      | E   | 1     |
| C      | E   | 2     |

| de E a | por | coste |
|--------|-----|-------|
| E      | E   | 0     |
| B      | B   | 1     |
| A      | D   | 2     |
| D      | D   | 1     |
| C      | C   | 1     |

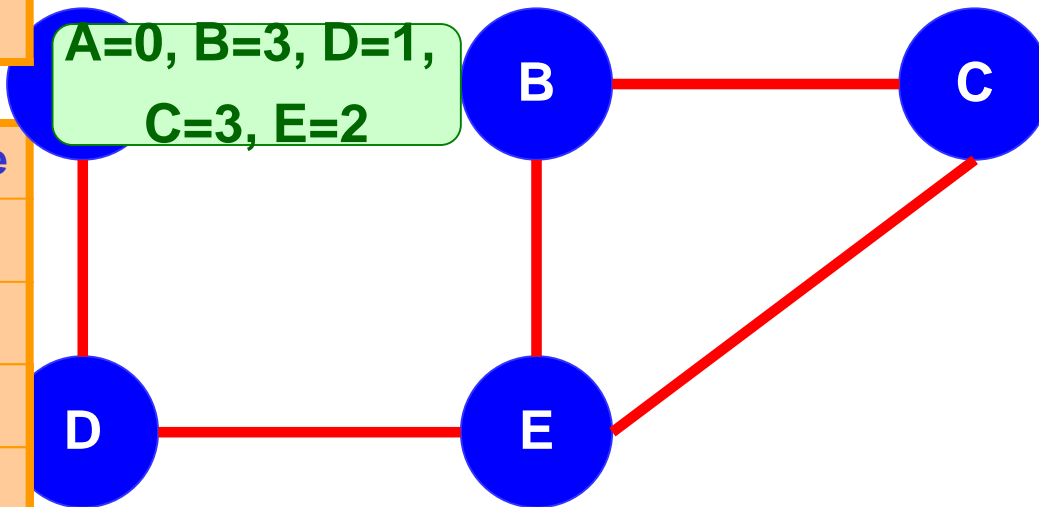


# Protocolos de Vector-Distancia

Introducción a su funcionamiento: Conteo a  $\infty$

| de A a | por | coste |
|--------|-----|-------|
| A      | A   | 0     |
| B      | D   | 3     |
| D      | D   | 1     |
| C      | D   | 3     |
| E      | D   | 2     |

| de D a | por | coste |
|--------|-----|-------|
| D      | D   | 0     |
| A      | A   | 1     |
| B      | A   | 4     |
| E      | A   | 3     |
| C      | A   | 4     |



- Pero el enlace D-E también falla:

- ◆ A y D se quedan aislados.

- ◆ D se da cuenta del fallo, pero:

- Si A manda su VD antes de darse cuenta del fallo (recibir el VD de D) → **Conteo a Infinito**.

- ◆ Solución para el conteo a infinito: Limitar la distancia máxima (Ej. infinito = 16).
- ◆ Mitigación 1: **Triggered updates:**
  - ❖ Forzar el envío de información actualizada tan pronto como se detecte un cambio.
- ◆ Mitigación 2: **Split Horizon**
  - ❖ Si A encamina paquetes hacia X a través de B, no tiene sentido que B intente alcanzar X a través de A.
  - ❖ 2 variantes:
    - ✓ **Split Horizon:** Nunca anunciar una ruta por la interfaz por la que se aprendió.
    - ✓ **Split Horizon with Poisoned Reverse:** En cada enlace, se informa como rutas inalcanzables (distancia infinita) aquellas que utilizan dicho enlace para alcanzar el destino.

### ◆ **Routing Information Protocol (RIP)**

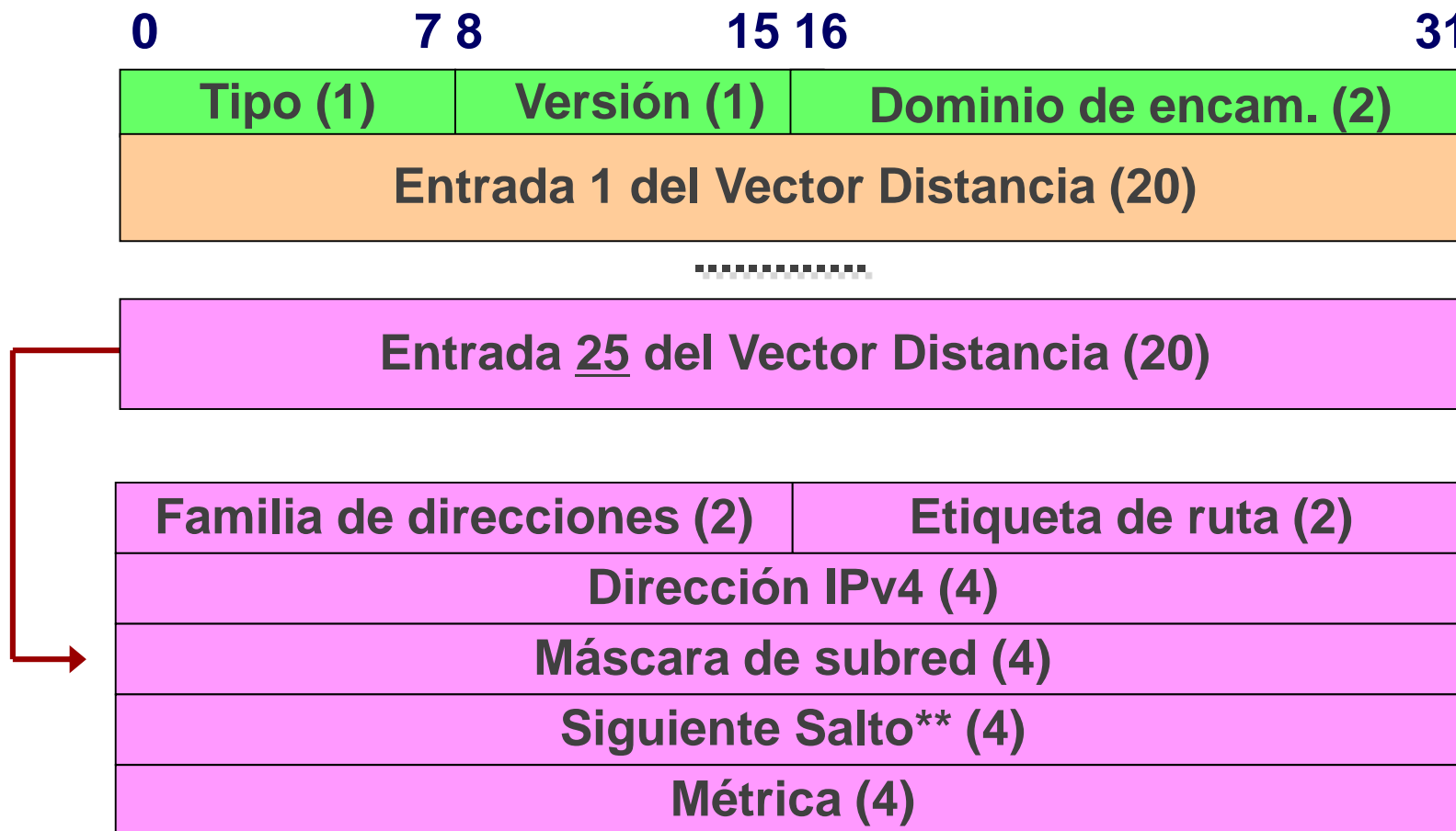
- ❖ **RIPv1:** Definido en la RFC 1058 (Junio 1988, 33 págs.)
- ❖ **RIPv2:** Definido en la RFC 2453 (Noviembre 1998, 39 págs.)

### ◆ **Características:**

- ❖ **Vector-Distancia.**
- ❖ **Sobre puerto 520 de UDP (origen y destino).**
- ❖ **RIP utiliza mensajes de difusión (*broadcast* y/o *multicast*).**
- ❖ **Métrica = Número de saltos (de 1 a 15; 16 =  $\infty$ ).**
  - ✓ No ponen la red local en el VD con distancia 0.
- ❖ **Dos tipos de participantes: Activos ó Pasivos.**
  - ✓ Cada 30 segs. Los nodos Activos difunden su Vector de Distancias: Conjunto de pares (subred IP, distancia).
  - ✓ Los nodos Activos y Pasivos escuchan los mensajes RIP y actualizan sus tablas.

### ◆ Características (cont.):

- ❖ Para cada destino sólo se guarda la mejor métrica recibida hasta el momento y el vecino que la mandó (siguiente salto de la ruta).
- ❖ Existe un proceso de borrado de rutas:
  - ✓ Las rutas se borran si dejan de actualizarse o se anuncian con una distancia infinita.
- ❖ Dos tipos de paquetes:
  - ✓ **Request**: (opcional) enviados por routers que acaban de conectarse, con información caducada, para monitorizar, etc.
  - ✓ **Response**: enviados periódicamente (30 segs.), en respuesta a un *Request* o cuando cambia algún coste (*Triggered Updates*):
    - Contienen los Vectores Distancia (VD).



\* Formato: Nombre del campo (Tamaño en octetos)

\*\* Sólo si es diferente al *router* origen (0.0.0.0 por defecto)

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filtter: rip Expression... Clear Apply

| No. . | Time      | Source      | Destination | Protocol | Info     |
|-------|-----------|-------------|-------------|----------|----------|
| 643   | 51.245932 | 192.168.4.1 | 224.0.0.9   | RIPv2    | Response |

▶ Frame 643 (126 bytes on wire, 126 bytes captured)  
 ▶ Ethernet II, Src: 02:14:04:d1:eb:63 (02:14:04:d1:eb:63), Dst: 01:00:5e:00:00:09 (01:00:5e:00:00:09)  
 ▶ Internet Protocol, Src: 192.168.4.1 (192.168.4.1), Dst: 224.0.0.9 (224.0.0.9)  
 ▼ User Datagram Protocol, Src Port: 520 (520), Dst Port: 520 (520)  
     Source port: 520 (520)  
     Destination port: 520 (520)  
     Length: 92  
     ▶ Checksum: 0x4abe [validation disabled]  
 ▼ Routing Information Protocol  
     Command: Response (2)  
     Version: RIPv2 (2)  
     Routing Domain: 0  
     ▼ IP Address: 192.168.1.0, Metric: 1  
         Address Family: IP (2)  
         Route Tag: 0  
         IP Address: 192.168.1.0 (192.168.1.0)  
         Netmask: 255.255.255.0 (255.255.255.0)  
         Next Hop: 0.0.0.0 (0.0.0.0)  
         Metric: 1  
     ▼ IP Address: 192.168.2.0, Metric: 1  
         Address Family: IP (2)  
         Route Tag: 0  
         IP Address: 192.168.2.0 (192.168.2.0)  
         Netmask: 255.255.255.0 (255.255.255.0)  
         Next Hop: 0.0.0.0 (0.0.0.0)  
         Metric: 1  
     ▼ IP Address: 192.168.3.0, Metric: 1  
         Address Family: IP (2)  
         Route Tag: 0  
         IP Address: 192.168.3.0 (192.168.3.0)  
         Netmask: 255.255.255.0 (255.255.255.0)  
         Next Hop: 0.0.0.0 (0.0.0.0)  
         Metric: 1  
     ▼ IP Address: 192.168.5.0, Metric: 1  
         Address Family: IP (2)  
         Route Tag: 0  
         IP Address: 192.168.5.0 (192.168.5.0)  
         Netmask: 255.255.255.0 (255.255.255.0)  
         Next Hop: 0.0.0.0 (0.0.0.0)  
         Metric: 1

```

0000 01 00 5e 00 00 09 02 14 04 d1 eb 63 08 00 45 00  .^.....C..E.
0010 00 70 00 00 40 00 01 11 d4 ca c0 a8 04 01 e0 00  .p.@.....
0020 00 09 02 08 02 08 00 5c 4a be 02 02 00 00 00 02  ..... \ J.....
0030 00 00 c0 a8 01 00 ff ff ff 00 00 00 00 00 00 00  .....
0040 00 01 00 02 00 00 c0 a8 02 00 ff ff ff 00 00 00  .....
0050 00 00 00 00 00 01 00 02 00 00 c0 a8 03 00 ff ff  .....
0060 ff 00 00 00 00 00 00 00 00 01 00 02 00 00 c0 a8  .....
0070 05 00 ff ff ff 00 00 00 00 00 00 00 00 01  .....
  
```

eth0: <live capture in progress> Fi... Packets: 2972 Displayed: 26 Marked: 0 Profile: Default

1. Al llegar un VD de R' a R, se le suma el coste de la red que ambos comparten y se compara con la tabla de R.
2. Si la entrada no existe y el valor de la métrica recibida no es  $\infty$ , se añade la entrada y como siguiente salto se pone a R'.
3. Si la entrada existía con una métrica mayor, se actualiza dicha métrica, el siguiente salto y se inicializa el temporizador asociado a la entrada.
4. Si la entrada existía y su siguiente salto era R', se actualiza la métrica – si difiere del valor almacenado (ya sea mayor o menor) – y se reinicializa el temporizador.
5. En el resto de los casos se ignora el mensaje.

### ◆ Envío:

- ❖ **Update timer:** se generan mensajes *Response* cada 30 segundos.

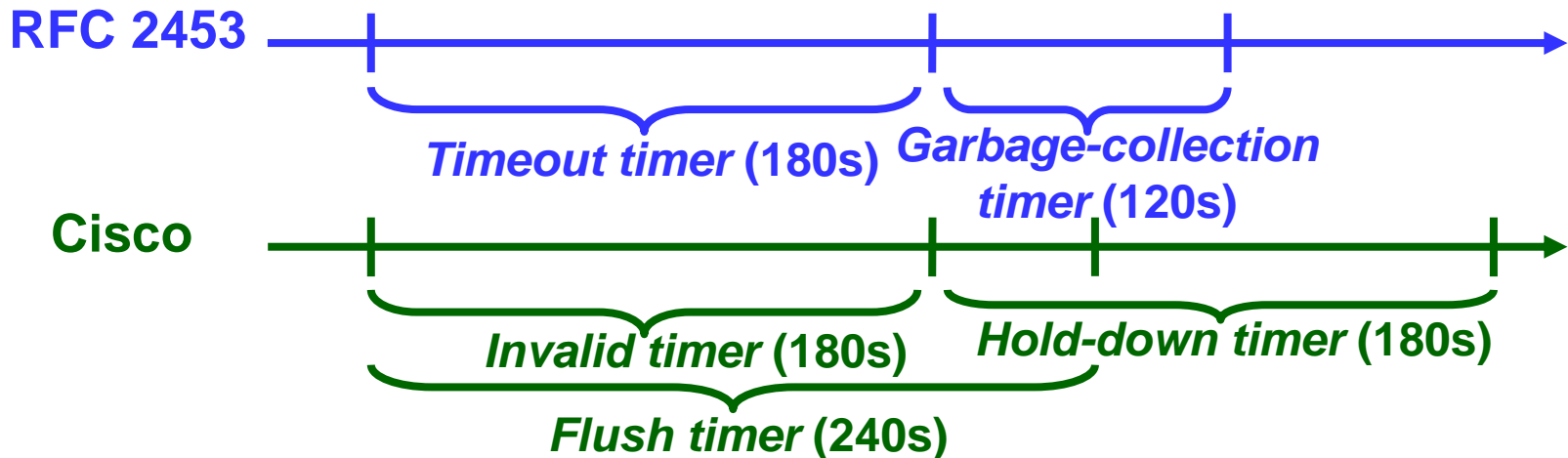
### ◆ Recepción:

- ❖ **Timeout timer:** Si durante 180 segundos (6 veces el *update-timer*) no se recibe un mensaje que actualice la ruta (o el siguiente salto envía un VD con distancia  $\infty$ ), se invalida dicha ruta:
  - ✓ Se establece distancia  $\infty$  para dicha ruta.
  - ✓ Se sigue incluyendo dicha ruta en los VD periódicos a los vecinos con distancia  $\infty$ .
  - ✓ Se lanza un temporizador de recolección de basura (*garbage-collection*).
- ❖ **Garbage-collection timer:** Si durante 120 segundos no se recibe un mensaje que actualice la ruta inválida, se borra definitivamente de las tablas.



- ◆ Cisco utiliza una terminología diferente y define nuevos temporizadores:
  - ❖ *Update timer*
  - ❖ *Timeout timer*:
    - ✓ Utiliza el término *Invalid timer* (180 s)
  - ❖ *Garbage-collection timer*:
    - ✓ Utiliza el término *Flush timer*
      - $Flush\ timer = Garbage\text{-}collection\ timer + Timeout\ timer$
      - Valor por defecto: 240s (ya que utiliza un valor para el *Garbage-collection timer* de 60s en vez de 120s)

- ◆ **Hold-down timer** (nuevo, definido por Cisco)
  - ❖ Se inicializa cuando el *Invalid timer* expira (o cuando el router actual de la ruta envía un VD con distancia  $\infty$ ).
  - ❖ Valor por defecto: 180 s
  - ❖ Durante este tiempo, se ignora toda actualización de la ruta que se reciba:
    - ✓ Se considera que la red puede no haber convergido aún y que la información recibida no es válida.



|  |                    |                   |
|--|--------------------|-------------------|
| <b>Tipo (1)</b>                            | <b>Versión (1)</b> | <b>A Cero (2)</b> |
| <b>Entrada 1 del Vector Distancia (20)</b> |                    |                   |

.....

|   |
|---|
| <b>Entrada 25 del Vector Distancia (20)</b> |
|---|

|                                   |                   |
|-----------------------------------|-------------------|
| <b>Familia de Direcciones (2)</b> | <b>A Cero (2)</b> |
| <b>Dirección IPv4 (4)</b>         |                   |
| <b>A Cero (4)</b>                 |                   |
| <b>A Cero (4)</b>                 |                   |
| <b>Métrica (4)</b>                |                   |



### ◆ Limitaciones RIPv1:

#### ❖ Información mínima para encaminar:

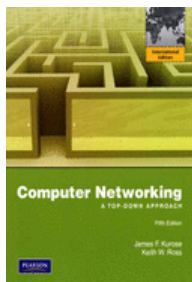
- ✓ No soporta AS, IGP/EGP, autenticación.
- ✓ No soporta máscaras: No permite *subnetting*, etc.
- ✓ Malgasta mucho espacio (campos a cero, etc.).
- ✓ Ningún mecanismo de seguridad.

### ◆ Justificación RIPv2 (en la RFC de RIPv2):

- ❖ Poca sobrecarga (*overhead*) en redes pequeñas (BW, gestión, etc.).
- ❖ Fácil de implementar.
- ❖ RIPv1 estaba muy extendido.

- ◆ **Definido por primera vez en RFC 1388 (Enero 1993):**
  - ❖ Actualizado en RFC 1723 (Nov. 1994) y RFC 2453 (Nov. 1998).
- ◆ **Nuevas características frente a RIPv1 (RFC 1058):**
  - ❖ Encaminamiento en subredes:
    - ✓ Incorpora un nuevo campo con la máscara de la subred.
  - ❖ Autenticación:
    - ✓ Un nuevo tipo de “Entrada RIP” permite autenticar al emisor de un mensaje RIP.
  - ❖ Etiqueta de Dominio de Encaminamiento, campo de Siguiente Salto:
    - ✓ Permite ejecutar varios “dominios” RIP en la misma red compartida.
  - ❖ Soporte de *multicast*:
    - ✓ Los mensajes pueden enviarse a un grupo *multicast* (224.0.0.9), en lugar de *broadcast* (que llega a todos los nodos).
  - ❖ Cierta compatibilidad:
    - ✓ Utiliza los campos a cero de RIPv1.

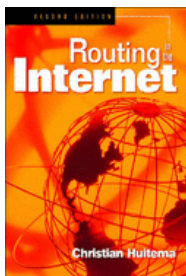
- ◆ RIP es un protocolo Vector-Distancia.
- ◆ Su funcionamiento puede resumirse en:
  - ❖ “Dile a tus vecinos cómo ves el mundo”
- ◆ Es un protocolo bastante simple.
- ◆ Tiene la ventaja de que “las buenas noticias” viajan rápido.
- ◆ Pero, por el contrario, “las malas noticias” viajan despacio.
- ◆ Carga las redes y puede crear bucles.
- ◆ Para redes pequeñas:
  - ❖ 15 saltos máximo.
- ◆ La métrica no tiene en cuenta (típicamente) la velocidad de los enlaces, las cargas, etc.



**J. F. Kurose, K. W. Ross; “*Computer Networking, a top-down approach*”, 5th Edition, Pearson – Addison Wesley, 2009.**

❖ Secciones 4.5.2 y 4.6.1

📖 L/S 004.7 KUR



**C. Huitema, “*Routing in the Internet*”, Upper Saddle River : Prentice Hall PTR, 2<sup>nd</sup> Edition, 2000.**

❖ Capítulos 5 y 6

📖 L/D 004.738.5.057.4 HUI



**G. Malkin, “*RIP Version 2*”, RFC 2453, Noviembre 1998.**

🖨️ <http://www.ietf.org/rfc/rfc2453.txt>