

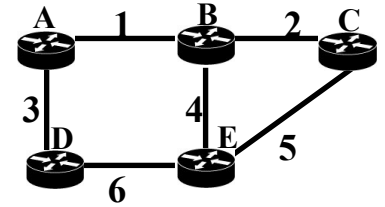
Ruteo Distance vector (RIP)

- **Ruteo dinámico, distribuido**
- **Utilizado en la Internet, RIPv1(Dominio público), RIPv2(DP), IGRP (Cisco), EIGRP (Cisco)**
- **En la actualidad, reemplazado por ruteo link state (OSPF)**

- **Simple de instalar y administrar**
- **Puede soportar una métrica (pura o combinada)**
- **Limitado en cuanto al diámetro máximo de la red**
- **Presenta problemas de convergencia**
- **Cada nodo conoce el costo de cada uno de sus vínculos (distancia, demora, etc)**
- **Un nodo intercambia periódicamente sus tablas de ruteo con sus nodos adyacentes**
- **El cálculo del camino más corto se realiza distribuido entre los nodos**

Distance vector-intercambio de tablas

- Ejemplo Nodos A, B, C, D, E; vínculos 1, 2, 3, 4, 5.
- Se asume como métrica distancia (todos los vínculos tienen costo 1)



1. Estado de las tablas en el arranque (aún no se intercambió información)

NODO A			NODO B			NODO C			NODO D			NODO E			
A	Nodo	Costo	Por link	A	Nodo	Costo	Por link	A	Nodo	Costo	Por link	A	Nodo	Costo	Por link
A		0	local	B		0	local	C		0	local	D		0	local

2. Primer intercambio, cada nodo envía su tabla a los adyacentes (p.ej. A envía: A=0 a B y a D)

INTERCAMBIO 1

A→B(L1): (A,0)

A→D(L3): (A,0)

B→A(L1): (B,0)

B→C(L2): (B,0)

B→E(L4): (B,0)

C→B(L2): (C,0)

C→E(L5): (C,0)

D→A(L3): (D,0)

D→E(L6): (D,0)

E→B(L4): (E,0)

E→C(L5): (E,0)

E→D(L6): (E,0)

NODO A		
A	Nodo	Costo
A		0
B	1	1
D	1	3

NODO B		
A	Nodo	Costo
B		0
A	1	1
C	1	2
E	1	4

NODO C		
A	Nodo	Costo
C		0
B	1	2
E	1	5

NODO D		
A	Nodo	Costo
D		0
A	1	3
E	1	6

NODO E		
A	Nodo	Costo
E		0
B	1	4
C	1	5
D	1	6

B: no esta en tabla, se agrega
 Costo a B:
 Costo pasado por B + costo Link 1

D: no esta en tabla, se agrega
 Costo a D:
 Costo pasado por D + costo Link 3

Distance vector-intercambio de tablas

NODO A		
A	Nodo	Costo Por link
A	0	local
B	1	1
D	1	3

NODO B		
A	Nodo	Costo Por link
B	0	local
A	1	1
C	1	2
E	1	4

NODO C		
A	Nodo	Costo Por link
C	0	local
B	1	2
E	1	5

NODO D		
A	Nodo	Costo Por link
D	0	local
A	1	3
E	1	6

NODO E		
A	Nodo	Costo Por link
E	0	local
B	1	4
C	1	5
D	1	6

2. Segundo intercambio

INTERCAMBIO 2

A→B(L1): (A,0), (B,1), (D,1)
 A→D(L3): (A,0), (B,1), (D,1)
 B→A(L1): (B,0), (A,1), (C,1), (E,1)
 B→C(L2): (B,0), (A,1), (C,1), (E,1)
 B→E(L4): (B,0), (A,1), (C,1), (E,1)
 C→B(L2): (C,0), (B,1), (E,1)
 C→E(L5): (C,0), (B,1), (E,1)
 D→A(L3): (D,0), (A,1), (E,1)
 D→E(L6): (D,0), (A,1), (E,1)
 E→B(L4): (E,0), (B,1), (C,1), (D,1)
 E→C(L5): (E,0), (B,1), (C,1), (D,1)
 E→D(L6): (E,0), (B,1), (C,1), (D,1)

NODO A		
A	Nodo	Costo Por link
A	0	local
B	1	1
D	1	3
C	2	1
E	2	3

NODO B		
A	Nodo	Costo Por link
B	0	local
A	1	1
C	1	2
E	1	4
D	2	4

NODO C		
A	Nodo	Costo Por link
C	0	local
B	1	2
E	1	5
A	2	2
D	2	5

NODO D		
A	Nodo	Costo Por link
D	0	local
A	1	3
E	1	6
C	2	6
B	2	6

NODO E		
A	Nodo	Costo Por link
E	0	local
B	1	4
C	1	5
D	1	6
A	2	6

PROCESO EN NODO A

Rutas a A:

de B, costo 1 → C=2, se queda con la de la tabla (C=0)

de D, costo 1 → C=2, se queda con la de la tabla (C=0)

Rutas a B:

de B, costo 0 → C=1, igual tabla por el mismo link, no cambia

Rutas a C:

de B, costo 1 → C=2, agrega entrada en tabla

Rutas a D:

de D, costo 0 → C=1, igual tabla por el mismo link, no cambia

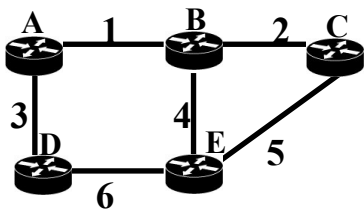
Rutas a E:

de B, costo 1 → C=2

elige cualquiera de las dos

de D, costo 1 → C=2

ya que tienen igual costo

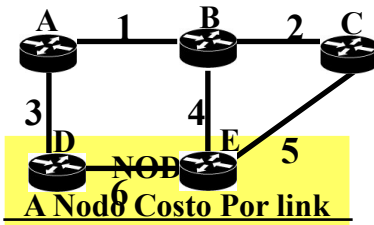


Distance vector: proceso de entradas

- **Vector de elementos costo del nodo vecino al destino: costo, destino (C, D)**
- **Link de recepción del vector LR**
- **Entrada en tabla: destino, costo, link de recepción (D, CT, LT)**

- **Por cada elemento del vector**
 - **Si no existe entrada para D**
 - **Calcular costo $CN = C + \text{costo}(\text{LR})$**
 - **Agregar entrada (D, CN, LR)**
 - **Si existe entrada**
 - **Calcular costo $CN = C + \text{costo}(\text{LR})$**
 - **Si $CN < CT$, modificar costo y link en la entrada (D, CN, LR)**
 - **Si $CN > CT$**
 - **Si $LR == LT$, modificar costo en tabla (D, CN, LT)**

Distance vector-caída de un link(1)



1-Estado de convergencia

NODO A		
A	Nodo	Costo Por link
A	0	local
B	1	1
D	1	3
C	2	1
E	2	3

NODO B		
A	Nodo	Costo Por link
B	0	local
A	1	1
C	1	2
E	1	4
D	1	4

NODO C		
A	Nodo	Costo Por link
C	0	local
B	1	2
E	1	5
A	2	2
D	2	5

NODO D		
A	Nodo	Costo Por link
D	0	local
A	1	3
E	1	6
C	2	6
B	2	6

NODO E		
A	Nodo	Costo Por link
E	0	local
B	1	4
C	1	5
D	1	6
A	2	6

2-Caída del link 1, detectada por A v por B

NODO A		
A	Nodo	Costo Por link
A	0	local
B	*	1
D	1	3
C	*	1
E	2	3

NODO B		
A	Nodo	Costo Por link
B	0	local
A	*	1
C	1	2
E	1	4
D	2	4

NODO C		
A	Nodo	Costo Por link
C	0	local
B	1	2
E	1	5
A	2	2
D	2	5

NODO D		
A	Nodo	Costo Por link
D	0	local
A	1	3
E	1	6
C	2	6
B	2	6

NODO E		
A	Nodo	Costo Por link
E	0	local
B	1	4
C	1	5
D	1	6
A	2	6

INTERCAMBIO 1

- A→D(L3): (A,0), (B,*), (D,1), (C,*), (E,2)
- B→C(L2): (B,0), (A,*), (C,1), (E,1), (D,*)
- B→E(L4): (B,0), (A,*), (C,1), (E,1), (D,*)
- C→B(L2): (C,0), (B,1), (E,1), (A,2), (D,2)
- C→E(L5): (C,0), (B,1), (E,1), (A,2), (D,2)
- D→A(L3): (D,0), (A,1), (E,1), (C,2), (B,2)
- D→E(L6): (D,0), (A,1), (E,1), (C,2), (B,2)
- E→B(L4): (E,0), (B,1), (C,1), (D,1), (A,2)
- E→C(L5): (E,0), (B,1), (C,1), (D,1), (A,2)
- E→D(L6): (E,0), (B,1), (C,1), (D,1), (A,2)

NODO A		
A	Nodo	Costo Por link
A	0	local
B	3	3
D	1	3
C	3	3
E	2	3

NODO B		
A	Nodo	Costo Por link
B	0	local
A	1	1
C	1	2
E	1	4
D	2	4

NODO C		
A	Nodo	Costo Por link
C	0	local
B	1	2
E	1	5
A	2	2
D	2	5

NODO D		
A	Nodo	Costo Por link
D	0	local
A	1	3
E	1	6
C	2	6
B	2	6

NODO E		
A	Nodo	Costo Por link
E	0	local
B	1	4
C	1	5
D	1	6
A	2	6

Nodo A, recibe rutas a C y B por L3, reemplaza porque son de menor costo y por otro link

Nodo D, recibe rutas de A, hacia B y C, costo *; no modifica sus rutas porque tiene costo menor y son por otro link

Distance vector-caída de un link(2)

2-Caída del link 1, detectada por A v por B

NODO A			NODO B			NODO C			NODO D			NODO E		
A	Nodo	Costo Por link	A	Nodo	Costo Por link	A	Nodo	Costo Por link	A	Nodo	Costo Por link	A	Nodo	Costo Por link
A	0	local	B	0	local	C	0	local	D	0	local	E	0	local
B	*	1	A	*	1	B	1	2	A	1	3	B	1	4
D	1	3	C	1	2	E	1	5	E	1	6	C	1	5
C	*	1	E	1	4	A	2	2	C	2	6	D	1	6
E	2	3	D	2	4	D	2	5	B	2	6	A	2	6

INTERCAMBIO 1

A→D(L3): (A,0), (B,*), (D,1), (C,*), (E,2)

B→C(L2): (B,0), (A,*), (C,1), (E,1), (D,*)

B→E(L4): (B,0), (A,*), (C,1), (E,1), (D,*)

C→B(L2): (C,0), (B,1), (E,1), (A,2), (D,2)

C→E(L5): (C,0), (B,1), (E,1), (A,2), (D,2)

D→A(L3): (D,0), (A,1), (E,1), (C,2), (B,2)

D→E(L6): (D,0), (A,1), (E,1), (C,2), (B,2)

E→B(L4): (E,0), (B,1), (C,1), (D,1), (A,2)

E→C(L5): (E,0), (B,1), (C,1), (D,1), (A,2)

E→D(L6): (E,0), (B,1), (C,1), (D,1), (A,2)

NODO A			NODO B			NODO C		
A	Nodo	Costo Por link	A	Nodo	Costo Por link	A	Nodo	Costo Por link
A	0	local	B	0	local	C	0	local
B	3	3	A	3	4	B	1	2
D	1	3	C	1	2	E	1	5
C	3	3	E	1	4	A	3	5
E	2	3	D	2	4	D	2	5

NODO D			NODO E		
A	Nodo	Costo Por link	A	Nodo	Costo Por link
D	0	local	E	0	local
A	1	3	B	1	4
E	1	6	C	1	5
C	2	6	D	1	6
B	2	6	A	2	6

Nodo C, recibe de B, por link 2: (A,*)

recibe de E, por link 5: (A,2)

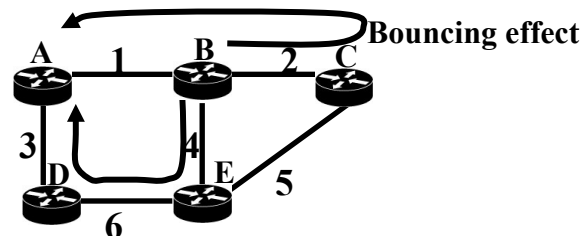
Nueva entrada para A: link 5, costo 3

Nodo B, recibe de C, por link 2: (A,2)

recibe de E, por link 4: (A, 2)

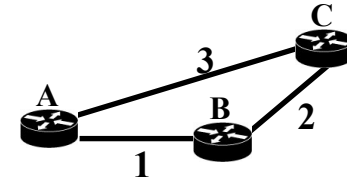
Si elige link 4 (por E), el protocolo converge

Si elige link 2 (por C), el protocolo no converge (bouncing effect)



Distance vector-Bouncing effect

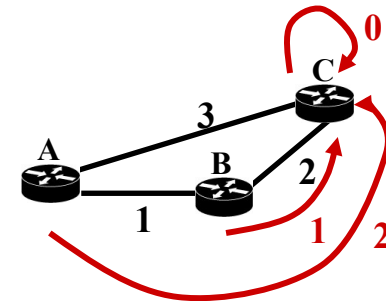
- Ejemplo de rebote (bouncing effect): Casos de fallas en los que el algoritmo converge, pero de manera lenta, produciendo situaciones de mal uso de los vínculos. Los vínculos 1 y 2 tienen costo 1, el 3 tiene costo 5. Se supone que cae el vínculo 2.



Los intercambios de tablas se producen cada 30 segundos, en el orden C a A; B a A; A a B y A a C. La información que se intercambia está aún sin actualizar

1- Rutas a C cuando el algoritmo converge inicialmente

Desde	Link	Costo
A a C	1	2
B a C	2	1
C a C	local	0



2- Rutas a C cuando cae el vínculo 2, en el momento en que B actualiza

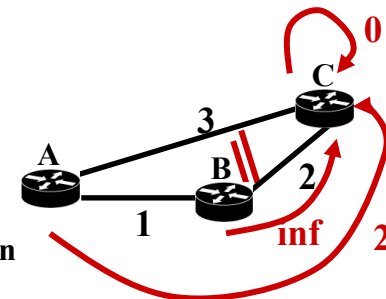
Desde	Link	Costo
A a C	1	2
B a C	2	inf
C a C	local	0

Se produce el intercambio de tablas con información de Tabla 2:

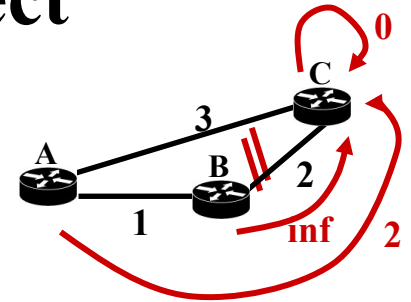
C a A: A no actualiza, tiene un costo menor

B a A: A a C, Link 1, Costo inf; actualiza pues aumenta costo en vínculo

A a B: B a C, Link 1, Costo 3;



Distance vector-Bouncing effect



3- Rutas a C actualizadas según Tabla 2

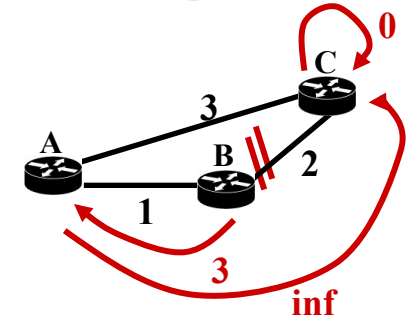
Desde	Link	Costo
A a C	1	inf
B a C	1	3
C a C	local	0

Se produce el intercambio de tablas con información de Tabla 3:

C a A: A a C, Link 3, Costo 5; actualiza porque tiene un costo mayor(inf)

B a A: A a C, Link 1, Costo 4; actualiza porque tiene costo mayor(5)

A a B: B a C, Link 1, Costo inf; actualiza porque el link aumentó el costo



4- Rutas a C actualizadas según Tabla 3

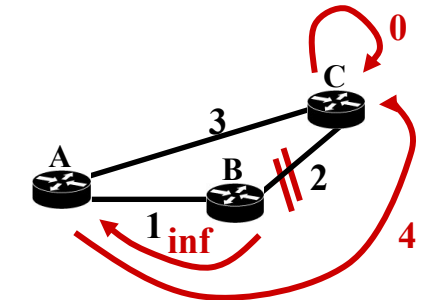
Desde	Link	Costo
A a C	1	4
B a C	1	inf
C a C	local	0

Se produce el intercambio de tablas con información de Tabla 4:

C a A: A A no actualiza porque tiene un costo menor(4)

B a A: A a C, Link 1, Costo inf; actualiza porque el link aumentó el costo

A a B: B a C, Link 1, Costo 5; actualiza porque tenía un costo mayor(inf)



5- Rutas a C actualizadas según Tabla 4

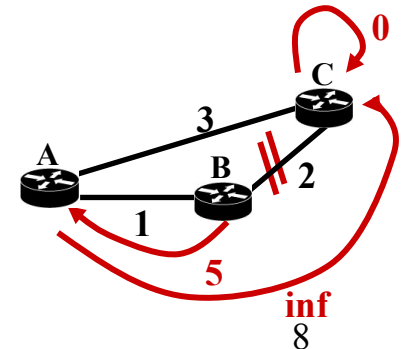
Desde	Link	Costo
A a C	1	inf
B a C	1	5
C a C	local	0

Se produce el intercambio de tablas con información de Tabla 5:

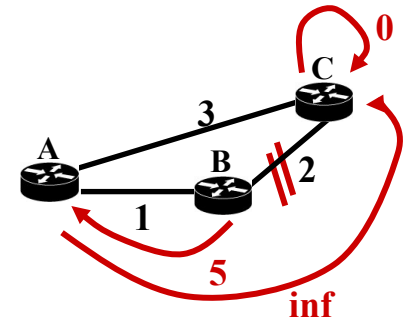
C a A: A a C, Link 3, Costo 5; actualiza porque tiene un costo mayor(inf)

B a A: A no actualiza porque el cálculo de costos es mayor (6)

A a B: B a C, Link 1, Costo inf; actualiza porque el link aumentó el costo



Distance vector-Bouncing effect



6- Rutas a C actualizadas según Tabla 5

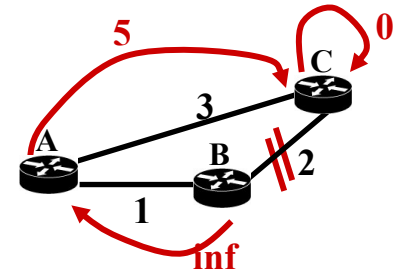
Desde	Link	Costo
A a C	3	5
B a C	1	inf
C a C	local	0

Se produce el intercambio de tablas con información de Tabla 6:

C a A: A no actualiza porque es el mismo costo

B a A: A no actualiza porque el cálculo de costos es mayor (inf)

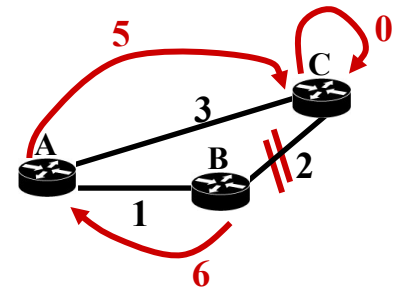
A a B: B a C, Link 1, Costo 6; actualiza porque el costo es menor



7- Rutas a C actualizadas según Tabla 6

Desde	Link	Costo
A a C	1	4
B a C	1	6
C a C	local	0

En este momento al algoritmo converge nuevamente



•El tiempo de convergencia depende del costo del vínculo 3

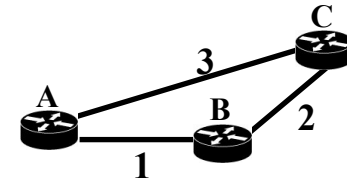
•El algoritmo oscila, teniendo períodos de inconsistencia (loop de paquetes, destinos no alcanzables)

Distance vector-Counting to infinity

La situación anterior con una caída del vínculo 3:

Los nodos A y B quedan aislados

No se produce la convergencia forzada por el costo de vínculo 3



Proceso conocido como “count to infinity”

Solución:

Límite impuesto dando un valor (de métrica) a infinito

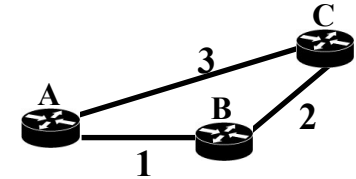
Problemas:

Limita el diámetro de la red

Produce inconsistencias temporarias hasta que se alcanza dicho valor

Distance vector -Split Horizon

- **Técnica para minimizar bouncing effect**



- **Modificación al algoritmo:**
 - Si el nodo A está enviando paquetes al nodo X a través del nodo B, B no podrá tratar de llegar a X a través de A
 - **Dos variantes**
 - Un nodo A que está enviando paquetes a otro X a través de uno B, no anunciará a B (por ese vínculo) ruta a X (técnica conservadora, basada en timeouts de entradas)
 - Un nodo A que está enviando paquetes a otro X a través de B, anunciará por ese vínculo un costo infinito a X (Split horizon with poisonus reverse).Técnica más agresiva, inmediatamente cancela la ruta

Distance vector-Split Horizon

Caso de falla de la técnica split horizon

1 Estado inicial de rutas a D

Desde	Link	Costo
A a D	2	2
B a D	3	2
C a D	4	1

2 Estado luego de caer vínculo 4

Desde	Link	Costo
A a D	2	2
B a D	3	2
C a D	4	inf

A propaga ruta destino D a B, ya que está ruteando vía C

Lo mismo ocurre con B

Ni A ni B propagan a C información acerca de D, por split horizon (envían costo infinito)

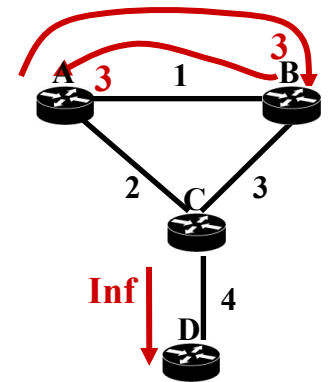
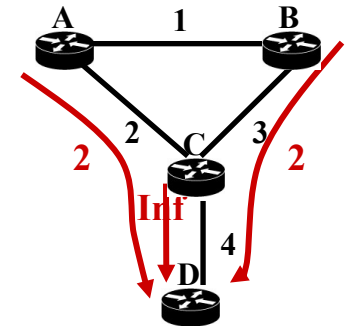
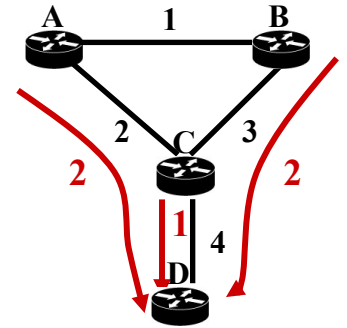
De C a A: D, Costo inf De C a B: D, Costo inf

De B a A: D, Costo 2 De A a B: D, Costo 2

3 Estado luego del intercambio

Desde	Link	Costo
A a D	1	3
B a D	1	3
C a D	4	inf

Luego comienza a incrementarse el costo a D en A y B (count to infinity) ya que el nodo al otro lado del vínculo registrará un incremento en el costo



RIP (Routing Information Protocol)

- **Aplicación de Distance vector y sus mejoras**
- **Difundido como parte del código de red del Unix BSD (routed)**
- **Amplia difusión**
- **Configuración simple**
- **Bajo overhead**
- **Reemplazado por OSPF**

- **Limitaciones**
 - **protocolo tipo IGP, para redes de diámetro menor que 15 saltos**
 - **métrica unica, fija**
 - **Direcciones classful**

- **Mejoras de RIP v2 (Definido en RFC)**
 - **soporta máscaras de red**
 - **soporta Dominios de ruteo**
 - **provee autenticación**

- **RIPng**
 - **Adaptado a IPv6**
 - **Cambios en las direcciones y prefijos de las rutas**

RIP v2: formato de frame

Comando	Versión	Routing domain
Familia de direcciones		Route tag
Dirección IP		
Máscara de subred		
Next Hop		
Métrica		

Comando

Request

Reply

Flia direcciones: 0002 :IP, FFFF: autenticación

Route Tag Ruta externa, uso de los EGP

Dirección IP: dirección destino

Mascara de subred: Soporte classless routing

Routing Domain: Normalmente un id de SA

Next hop: Impide repetición de paquetes

Métrica: Cantidad de saltos

Autenticación

Compatible con v1 (Familia FFFF)

Reemplaza el primer anuncio

Comando	Versión	Routing domain
0XFFFF		Tipo Autenticación
Aut		
Aut		
Aut		
Aut		

RIP-Triggered Updates

- **Información de ruteo enviada asincrónicamente al producirse un cambio**
- **Aceleran tiempos de convergencia del algoritmo**
- **Complementan al intercambio periódico de tablas**
- **Pueden dar lugar a un volumen de tráfico considerable (dependiendo del estado de las tablas)**

RIP

Timers:

Intercambio de información de ruteo : 30 segundos

Validez de una ruta : 180 segs.

Garbage collection: 120 segs

Timer para triggered updates: al azar, entre 1 y 5 segs.

Entradas en la tabla

Dirección del destino (red o host)

Métrica

Dirección del next hop

Silent nodes