



Algoritmos de Ruteo - Introducción

- La función principal de la capa de red es el “ruteo”.
- Mecanismo: Algoritmo de ruteo
- Clasificaciones
 - Donde toman las decisiones
 - Estrategias de ruteo
 - Origen de la información



Algoritmos de Ruteo - Estáticos vs. Dinámicos

Estáticos

Información previamente recopilada

En función de la capacidad de la línea, tráfico promedio, etc.

Tablas cargadas por el administrador estáticamente en cada router

No es posible responder a situaciones cambiantes

Algoritmos sofisticados

Dinámicos

Información recopilada en tiempo real

En función de la info actual de la red recibida de otros routers

Protocolo de routing

Mecanismo autoadaptativo

Algoritmos poco complejos



Flooding

- ❑ Ruteo estático, aislado
- ❑ Cada paquete es enviado por todos los vínculos excepto por el que llegó
- ❑ Muy resistente a fallas
- ❑ Llega por el camino más corto
- ❑ Genera gran overhead en la red
- ❑ Usos
 - ❑ Aplicaciones militares
 - ❑ Bases de datos distribuidas



Flooding

□ Mejoras

□ Utilización del Spanning Tree

Los routers pueden calcular el mínimo árbol de recubrimiento, con lo cual desechan ciertos vínculos para evitar ciclos.

□ No reenviar un paquete más de una vez

Se supone que cada paquete tiene un id único que es RECORDADO por el router. Cuando ve un paquete cuya id ya conoce, lo descarta.

□ Contador de nodos y eliminación de paquete

Cada paquete sale con un TTL máximo igual la diámetro de la red. Cuando el TTL llega a 0, el paquete es descartado.

□ Flooding selectivo

Cada router lo envía no necesariamente por todas las interfaces, sino por las que supone que mas lo acercan.



Backward Learning - Características

- ❑ No es completamente aislado, utiliza tablas de ruteo
- ❑ Son utilizados en redes locales
- ❑ Funcionamiento:
 - ❑ Si no se conoce el destino, se hace flooding y se incrementa el nro de saltos dado por el paquete
 - ❑ Si el destino se conoce, se envía el paquete por la ruta que se indica en la tabla.
 - ❑ El aprendizaje se basa en leer por cada paquete:
 - ❑ Origen.
 - ❑ Interfaz a través de la cual ha llegado.
 - ❑ numero de saltos dados por el paquete.Y quedarse con la entrada que mas convenga.



Backward Learning - Consideraciones

- Para evitar que un paquete esté dando vueltas eternamente en la red se limita el número de saltos que éste puede dar.
- A las entradas se les asocia un tiempo de vida, que se renueva cada vez que se hace uso de la entrada.



Distance Vector -Introducción

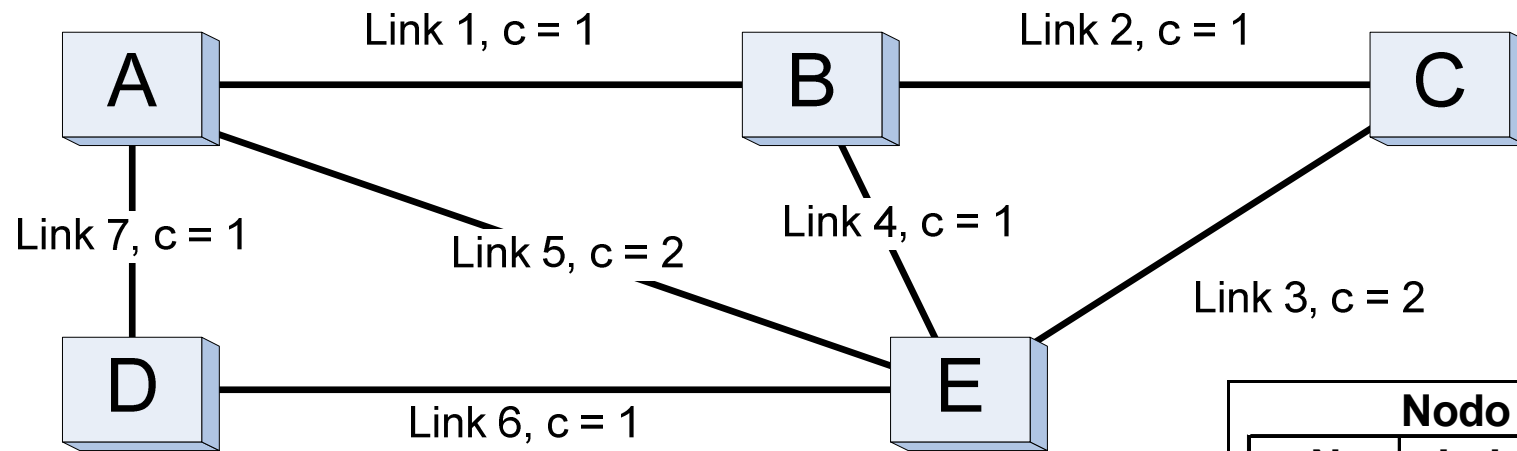
- Cada router mantiene una tabla de ruteo con:
 - La mejor distancia a cada destino
 - Link que debe usar
- Ruteo dinámico
- Cálculo de manera distribuida
- RIPv1, RIPv2, IGRP y EIGRP



Distance Vector – Información en el router

- Cada router mantiene una tabla de ruteo
 - Una entrada por cada router de la red
 - Cada entrada tiene dos valores:
 - Link
 - Costo

Distance Vector – Información en el router



Nodo A		
N	Lnk	C
A	loc	0
B	1	1
E	5	2
D	7	1
C	1	2



Distance Vector – Algoritmo

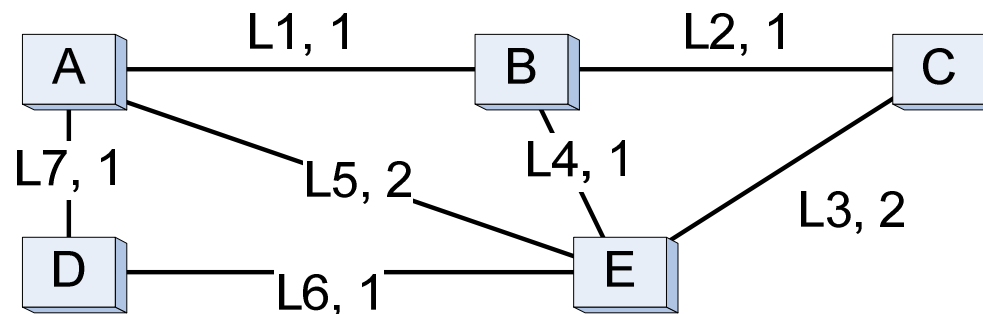
- Inicialmente cada tabla tiene una entrada indicando a el mismo con costo cero
- Por cada tabla que recibe de los nodos adyacentes
 - Para cada destino de las entradas de las tablas
 - calcula el costo (costo al nodo adyacente + costo de ese nodo adyacente al destino)
 - Si el nodo destino no esta en la tabla lo agrega
 - Si esta en la tabla deja el de menor costo
 - Si el destino es alcanzado a través del vínculo que recibió la actualización, el costo es modificado



Ejercicio 4 - Enunciado

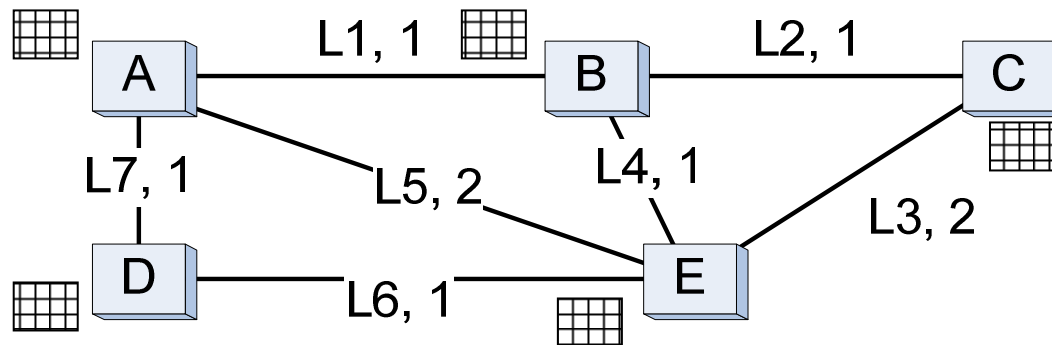
Para la siguiente red suponiendo que se utiliza distance vector y que se realiza el intercambio de tablas de ruteo cada 30 segundos, y una entrada en la tabla es descartada luego de 180 segundos. (RIP)

A) ¿Cuánto tiempo desde el cold start tarda en converger el algoritmo?



Ejercicio 4 – Inciso A

□ Cold Start, $t=0$.



Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0	D	loc	0	E	loc	0

Ejercicio 4 – Inciso A

□ Primera Transición $t= 30$.

A través de L1 recibo:

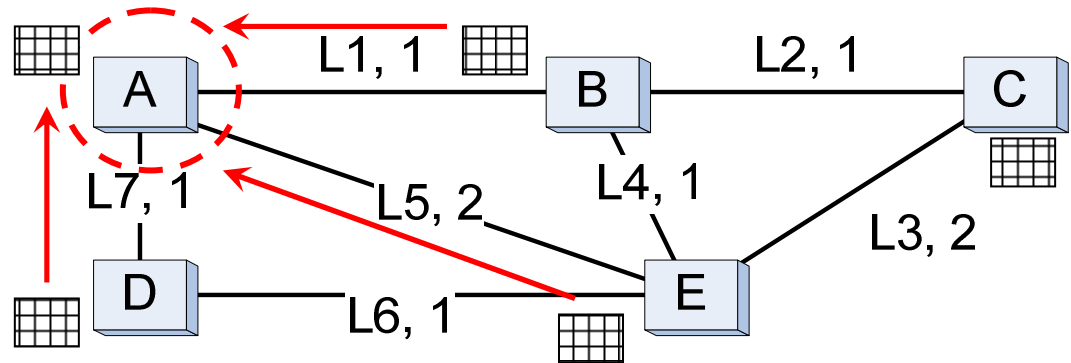
B	0
---	---

A través de L5 recibo:

E	0
---	---

A través de L7 recibo:

D	0
---	---

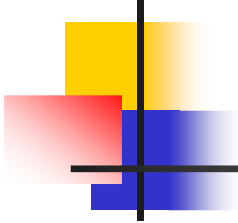


N	Lnk	C
A	loc	0

N	Lnk	C
A	loc	0
B	1	1

N	Lnk	C
A	loc	0
B	1	1
E	5	2

Nodo A		
N	Lnk	C
A	loc	0
B	1	1
E	5	2
D	7	1



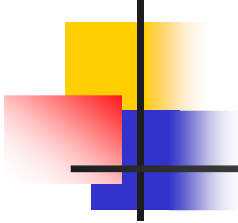
Ejercicio 4 – Inciso A

□ Cold Start, $t=0$.

Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0	D	loc	0	E	loc	0

□ Primera Transición, $t=30$.

Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0	D	loc	0	E	loc	0
B	1	1	A	1	1	B	2	1	A	7	1	A	5	2
E	5	2	C	2	1	E	3	2	E	6	1	B	4	1
D	7	1	E	4	1							C	3	2
												D	6	1



Ejercicio 4 – Inciso A

□ Primera Transición, $t=30$.

Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0	D	loc	0	E	loc	0
B	1	1	A	1	1	B	2	1	A	7	1	A	5	2
E	5	2	C	2	1	E	3	2	E	6	1	B	4	1
D	7	1	E	4	1							C	3	2
												D	6	1

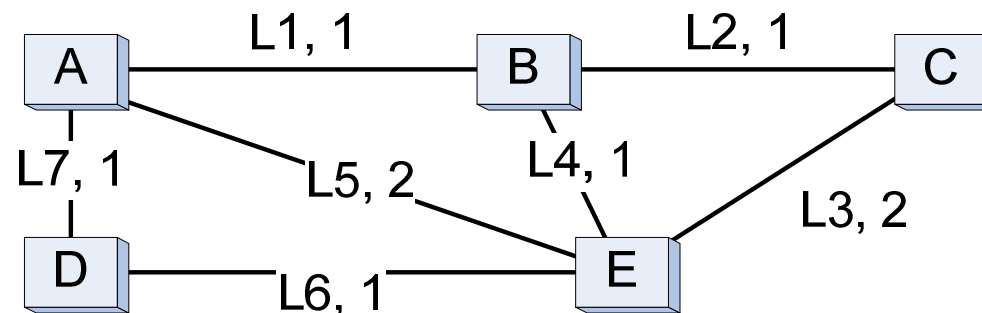
□ Segunda Transición, $t=60$.

Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0	D	loc	0	E	loc	0
B	1	1	A	1	1	B	2	1	A	7	1	A	5	2
E	5	2	C	2	1	E	3	2	E	6	1	B	4	1
D	7	1	E	4	1	A	2	2	B	7	2	C	3	2
C	1	2	D	1	2	D	3	3	C	6	3	D	6	1

Ejercicio 4 – Inciso B

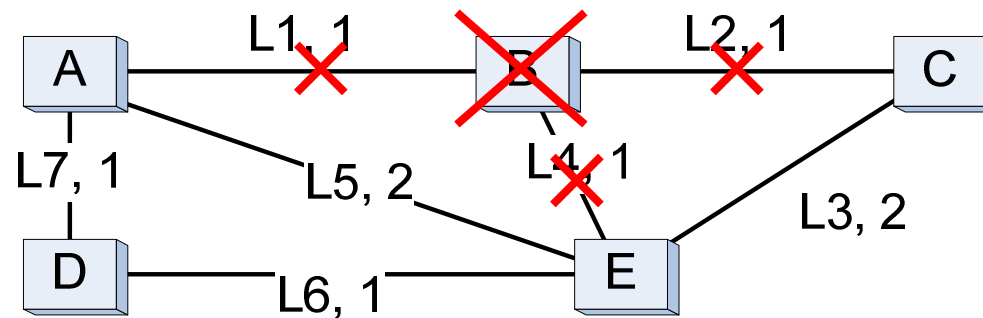
¿Qué overhead (bps) introduce en la red el ruteo (suponer que la información destino-costo insume 10 bytes)? ¿Cómo será la carga en cada uno de los vínculos?

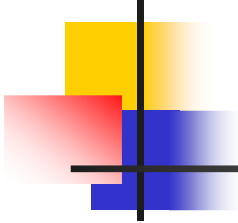
Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0	D	loc	0	E	loc	0
B	1	1	A	1	1	B	2	1	A	7	1	A	5	2
E	5	2	C	2	1	E	3	2	E	6	1	B	4	1
D	7	1	E	4	1	A	2	2	B	7	2	C	3	2
C	1	2	D	1	2	D	3	3	C	6	3	D	6	1



Ejercicio 4 – Inciso C

- c) Suponga que cae el router B. Describa qué ocurre (intercambios de tablas) hasta que el algoritmo converge. ¿Cuánto tiempo tarda en converger? ¿Cómo mejora este tiempo el utilizar triggered updates?





Ejercicio 4 – Inciso C

- Tablas antes de la caída del router.

Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0	D	loc	0	E	loc	0
B	1	1	A	1	1	B	2	1	A	7	1	A	5	2
E	5	2	C	2	1	E	3	2	E	6	1	B	4	1
D	7	1	E	4	1	A	2	2	B	7	2	C	3	2
C	1	2	D	1	2	D	3	3	C	6	3	D	6	1

- Se cae el router B con los vínculos 1, 2 y 4.

Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0				C	loc	0	D	loc	0	E	loc	0
B	1	inf				B	2	inf	A	7	1	A	5	2
E	5	2				E	3	2	E	6	1	B	4	inf
D	7	1				A	2	inf	B	7	2	C	3	2
C	1	inf				D	3	3	C	6	3	D	6	1

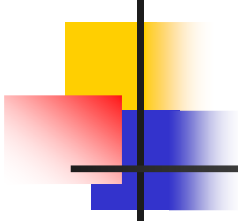
Ejercicio 4 – Inciso C

- Se cae el router B con los vínculos 1, 2 y 4.

Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0				C	loc	0	D	loc	0	E	loc	0
B	-	inf				B	-	inf	A	7	1	A	5	2
E	5	2				E	3	2	E	6	1	B	-	inf
D	7	1				A	-	inf	B	7	2	C	3	2
C	-	inf				D	3	3	C	6	3	D	6	1

- Primera Transición.

Nodo A			Nodo B			Nodo C			Nodo D			Nodo E		
N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0				C	loc	0	D	loc	0	E	loc	0
B	7	3				B	2	inf	A	7	1	A	5	2
E	5	2				E	3	2	E	6	1	B	6	3
D	7	1				A	3	4	B	7	inf	C	3	2
C	7	4				D	3	3	C	6	3	D	6	1



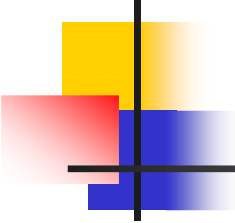
Ejercicio 4 – Inciso C

□ Primera Transición.

Nodo A			Nodo B	Nodo C			Nodo D			Nodo E		
N	Lnk	C		N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0		C	loc	0	D	loc	0	E	loc	0
B	-	inf		B	-	inf	A	7	1	A	5	2
E	5	2		E	3	2	E	6	1	B	-	inf
D	7	1		A	-	inf	B	7	2	C	3	2
C	-	inf		D	3	3	C	6	3	D	6	1

□ Segunda Transición.

Nodo A			Nodo B	Nodo C			Nodo D			Nodo E		
N	Lnk	C		N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0		C	loc	0	D	loc	0	E	loc	0
B	5	5		B	3	5	A	7	1	A	5	2
E	5	2		E	3	2	E	6	1	B	5	5
D	7	1		A	3	4	B	7	4	C	3	2
C	7	4		D	3	3	C	6	3	D	6	1



Ejercicio 4 – Inciso C

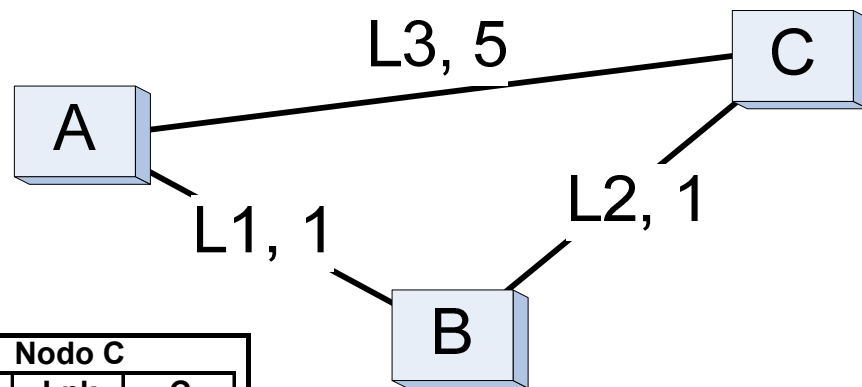
Triggered Updates

- ❑ **Si una métrica cambia el router inmediatamente envía una actualización de su tabla a sus adyacentes.**
- ❑ **Converge mucho mas rápido.**
- ❑ **“Counting to Infinity” se reduce drásticamente.**
- ❑ **Incompatibilidades entre actualizaciones regulares y “triggered updates”**

Ejercicio 6

Encuentre un ejemplo de efecto de rebote y otro de counting to infinity. Muestre el intercambio de tablas en los casos en que se utilice split horizon con anuncio de métrica infinita y con poisonus reverse.

Efecto de rebote

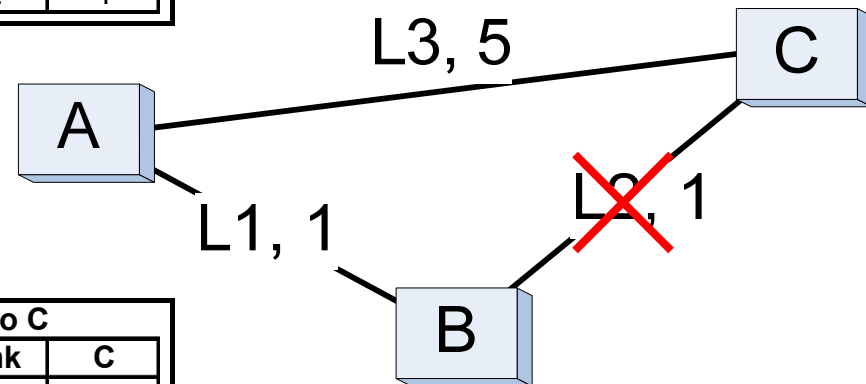


Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	2	2
C	1	2	C	2	1	B	2	1

Ejercicio 6 – Efecto Rebote

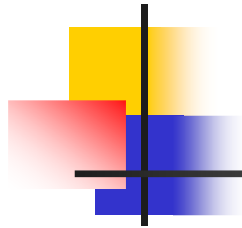
Estado Inicial

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	2	2
C	1	2	C	2	1	B	2	1



Cae el vínculo 2

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	2	inf
C	1	2	C	2	inf	B	2	inf



Ejercicio 6 – Efecto Rebote

Cae el vínculo 2

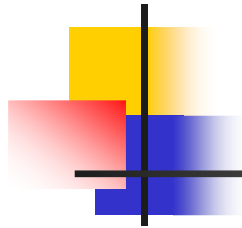
Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	2	inf
C	1	2	C	2	inf	B	2	inf

Primer Transición

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	3	5
C	3	5	C	1	3	B	3	6

Segunda Transición

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	3	5
C	1	4	C	1	3	B	3	6



Ejercicio 6 – Efecto Rebote

Segunda Transición

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	3	5
C	1	4	C	1	3	B	3	6

Tercera Transición

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	3	5
C	3	5	C	1	5	B	3	6

Cuarta Transición

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	3	5
C	3	5	C	1	6	B	3	6



Ejercicio 6 – Split Horizon

- Si el nodo A está enviando 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 vinculo) ruta a X.
 - Un nodo A que está enviando paquetes a otro X a través de B, anunciará por ese vinculo un costo infinito a X (with poisonous reverse). Inmediatamente cancela la ruta.

Ejercicio 6 – Split Horizon with Poisonus Reverse

Cae el vínculo 2

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	2	inf
C	1	2	C	2	inf	B	2	inf

Primer Transición

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	3	5
C	3	5	C	1	inf	B	3	6

Segunda Transición

Nodo A			Nodo B			Nodo C		
N	Lnk	C	N	Lnk	C	N	Lnk	C
A	loc	0	B	loc	0	C	loc	0
B	1	1	A	1	1	A	3	5
C	3	5	C	1	6	B	3	6