

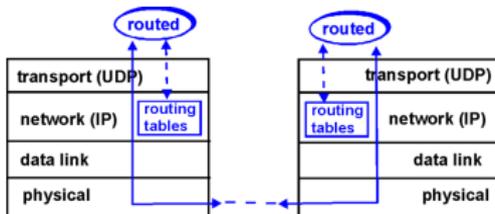
Redes de Datos II

Ruteo Interno

Luis Marrone

LINTI-UNLP

29 de octubre de 2021



1 Ruteo

2 Protocolos de Ruteo Interno

- Algoritmo de Distancia Vectorial
- RIP
- Algoritmos de Link State
- OSPF

Estamos en:

1 Ruteo

2 Protocolos de Ruteo Interno

- Algoritmo de Distancia Vectorial
- RIP
- Algoritmos de Link State
- OSPF

Ruteo vs. "Forwarding"

- ✓ Forwarding: selecciona un port de salida en función de la dirección de destino y tabla de ruteo
- ✓ Ruteo: proceso mediante el cual se construye la tabla de ruteo

Ruteo vs. "Forwarding"

- ✓ Forwarding: selecciona un port de salida en función de la dirección de destino y tabla de ruteo

- ✓ Ruteo: proceso mediante el cual se construye la tabla de ruteo

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global
- ✓ Estado global:
 - "time consuming"
 - Dinámico
 - Difícil de recolectar
- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global
- ✓ Estado global:
 - “time consuming”
 - Dinámico
 - Difícil de recolectar
- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global

- ✓ Estado global:
 - “time consuming”
 - Dinámico
 - Difícil de recolectar

- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global

- ✓ Estado global:
 - “time consuming”
 - Dinámico
 - Difícil de recolectar

- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global
- ✓ Estado global:
 - “time consuming”
 - Dinámico
 - Difícil de recolectar
- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global

- ✓ Estado global:
 - “time consuming”
 - Dinámico
 - Difícil de recolectar

- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global

- ✓ Estado global:
 - “time consuming”
 - Dinámico
 - Difícil de recolectar

- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global

- ✓ Estado global:
 - “time consuming”
 - Dinámico
 - Difícil de recolectar

- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

¿Cuál es el problema?

- ✓ Recolectar y procesar un estado global

- ✓ Estado global:
 - “time consuming”
 - Dinámico
 - Difícil de recolectar

- ✓ Cuestiones a resolver
 - Consistencia
 - Completitud
 - Escalabilidad

Consistencia

- ✓ Decisiones de “forwarding” local deben llevar a una conectividad entre cualquier origen y destino de la red
- ✓ Si los estados son inconsistentes la red no habrá convergido a un estado estable
- ✓ La inconsistencia genera loops

Consistencia

- ✓ Decisiones de “forwarding” local deben llevar a una conectividad entre cualquier origen y destino de la red
- ✓ Si los estados son inconsistentes la red no habrá convergido a un estado estable
- ✓ La inconsistencia genera loops

Consistencia

- ✓ Decisiones de “forwarding” local deben llevar a una conectividad entre cualquier origen y destino de la red
- ✓ Si los estados son inconsistentes la red no habrá convergido a un estado estable
- ✓ La inconsistencia genera loops

Completitud

- ✓ La red como un todo y cada nodo deben tener la información suficiente para poder hallar todos los caminos
- ✓ Cuanto mayor sea la información disponible localmente mayor es la rapidez con que convergen los algoritmos de ruteo
- ✓ Eso significa que se debe recolectar mayor información en cada nodo
- ✓ Esto también limita la escalabilidad del algoritmo

Completitud

- ✓ La red como un todo y cada nodo deben tener la información suficiente para poder hallar todos los caminos
- ✓ Cuanto mayor sea la información disponible localmente mayor es la rapidez con que convergen los algoritmos de ruteo
- ✓ Eso significa que se debe recolectar mayor información en cada nodo
- ✓ Esto también limita la escalabilidad del algoritmo

Completitud

- ✓ La red como un todo y cada nodo deben tener la información suficiente para poder hallar todos los caminos
- ✓ Cuanto mayor sea la información disponible localmente mayor es la rapidez con que convergen los algoritmos de ruteo
- ✓ Eso significa que se debe recolectar mayor información en cada nodo
- ✓ Esto también limita la escalabilidad del algoritmo

Completitud

- ✓ La red como un todo y cada nodo deben tener la información suficiente para poder hallar todos los caminos
- ✓ Cuanto mayor sea la información disponible localmente mayor es la rapidez con que convergen los algoritmos de ruteo
- ✓ Eso significa que se debe recolectar mayor información en cada nodo
- ✓ Esto también limita la escalabilidad del algoritmo

Ruteo estático vs. dinámico

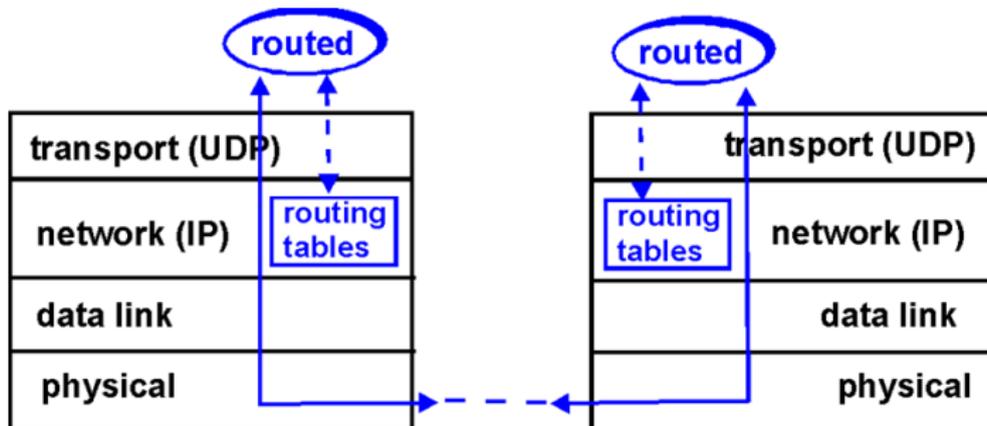
Estático

- Configuración manual de las tablas
- Mayor control
- No es escalable
- Lento para adaptarse a los cambios en la red

Dinámico

- Se obtienen caminos óptimos
- Rápida adaptación a los cambios
- Escalables
- Consumo de CPU, BW y Memoria
- “Debugging” complejo

Esquema Ruteo Dinámico



Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Modelo de Ruteo en Internet

- ✓ Ruteo dinámico
- ✓ Ruteo inter e intra SA (Sistema Autónomo)
- ✓ Sistema Autónomo: Control Administrativo
- ✓ Internet se organiza en base a SA
- ✓ Protocolos de ruteo interno (IGP)
 - Dentro de un SA
 - RIP, OSPF
- ✓ Protocolos de ruteo externo (EGP)
 - Entre SA
 - EGP, BGP-4

Estamos en:

1 Ruteo

2 Protocolos de Ruteo Interno

- Algoritmo de Distancia Vectorial
- RIP
- Algoritmos de Link State
- OSPF

Algoritmos de Ruteo Dinámico

✓ Distancia Vectorial

- Los nodos intercambian tablas con sus vecinos

✓ “Link-state”: información basada en los enlaces

- Obtienen un mapa completo de la red
- Encuentran el camino óptimo y determinan el próximo salto localmente(Algoritmo de Dijkstra).

Algoritmos de Ruteo Dinámico

✓ Distancia Vectorial

- Los nodos intercambian tablas con sus vecinos

✓ “Link-state”: información basada en los enlaces

- Obtienen un mapa completo de la red
- Encuentran el camino óptimo y determinan el próximo salto localmente (Algoritmo de Dijkstra).

Algoritmos de Ruteo Dinámico

- ✓ Distancia Vectorial
 - Los nodos intercambian tablas con sus vecinos

- ✓ “Link-state”: información basada en los enlaces
 - Obtienen un mapa completo de la red
 - Encuentran el camino óptimo y determinan el próximo salto localmente(Algoritmo de Dijkstra).

Algoritmos de Ruteo Dinámico

- ✓ Distancia Vectorial
 - Los nodos intercambian tablas con sus vecinos

- ✓ “Link-state”: información basada en los enlaces
 - Obtienen un mapa completo de la red
 - Encuentran el camino óptimo y determinan el próximo salto localmente(Algoritmo de Dijkstra).

Algoritmos de Ruteo Dinámico

- ✓ Distancia Vectorial
 - Los nodos intercambian tablas con sus vecinos

- ✓ “Link-state”: información basada en los enlaces
 - Obtienen un mapa completo de la red
 - Encuentran el camino óptimo y determinan el próximo salto localmente(Algoritmo de Dijkstra).

Distancia Vectorial

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

Distancia Vectorial

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

Distancia Vectorial

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

Distancia Vectorial

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

Distancia Vectorial

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

Distancia Vectorial

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

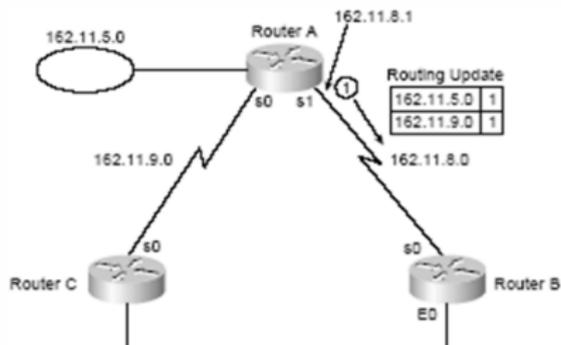
Distancia Vectorial

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

Distancia Vectorial

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

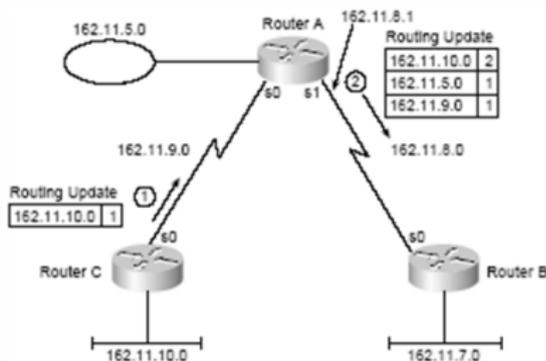
Algoritmo Vectorial en funcionamiento



Router B

Group (Mask Is 255.255.255.0)	Outgoing Interface	Next-Hop Router
162.11.5.0	S0	162.11.8.1
162.11.7.0	E0	—
162.11.8.0	S0	—
162.11.9.0	S0	162.11.8.1

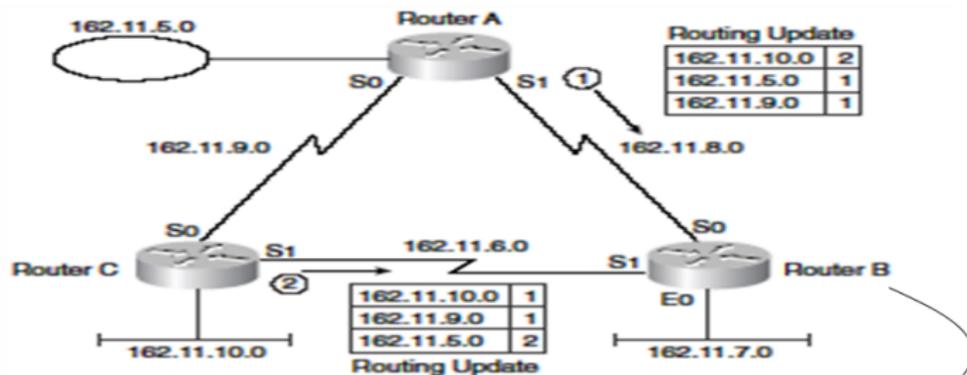
Algoritmo Vectorial en funcionamiento . . .



Router B

Group	Outgoing Interface	Next-Hop Router	Metric
162.11.5.0	S0	162.11.8.1	1
162.11.7.0	E0	—	0
162.11.8.0	S0	—	0
162.11.9.0	S0	162.11.8.1	1
162.11.10.0	S0	162.11.8.1	2

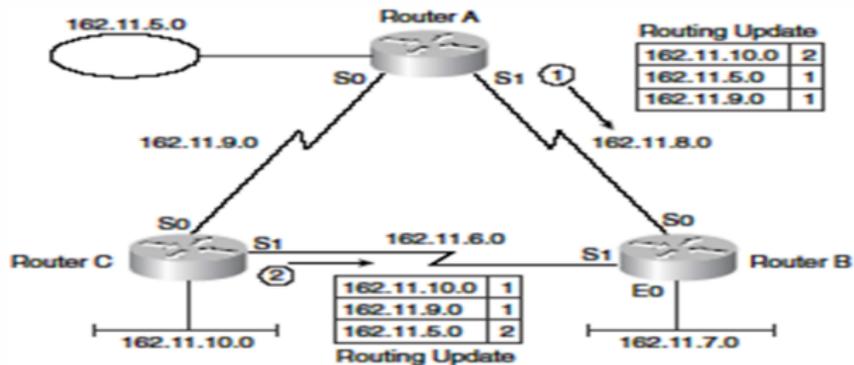
Algoritmo Vectorial en funcionamiento . . .



1

Group	Outgoing Interface	Next Router	Metric
162.11.5.0	S0	162.11.8.1	1
162.11.7.0	E0		0
162.11.8.0	S0		0
162.11.9.0	S0	162.11.8.1	1
162.11.10.0	S0	162.11.8.1	2

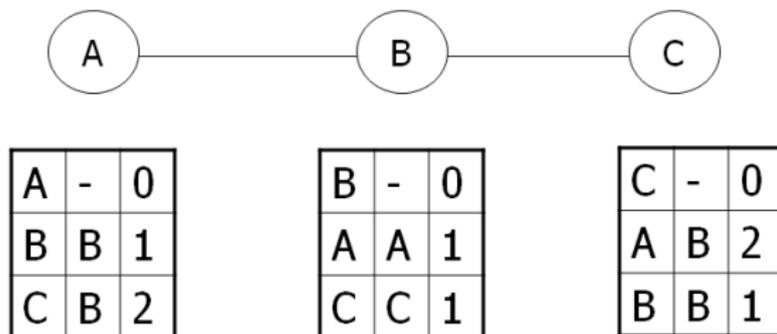
Algoritmo Vectorial en funcionamiento . . .



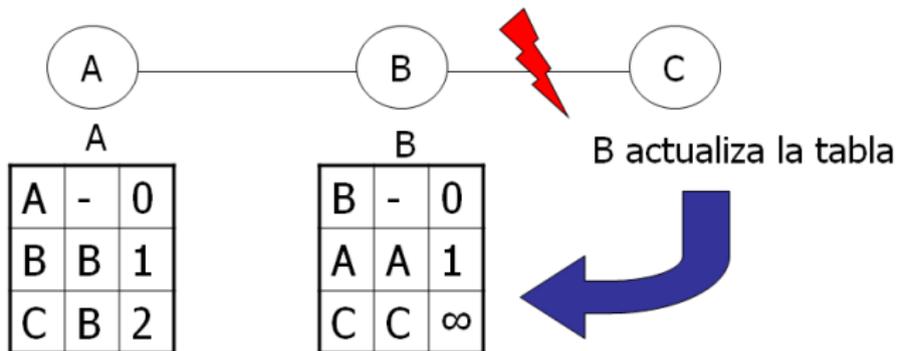
2

Group	Outgoing Interface	Next Router	Metric
162.11.5.0	S0	162.11.8.1	1
162.11.6.0	S1		0
162.11.7.0	E0		0
162.11.8.0	S0		0
162.11.9.0	S0	162.11.8.1	1
162.11.10.0	S1	162.11.6.2	1

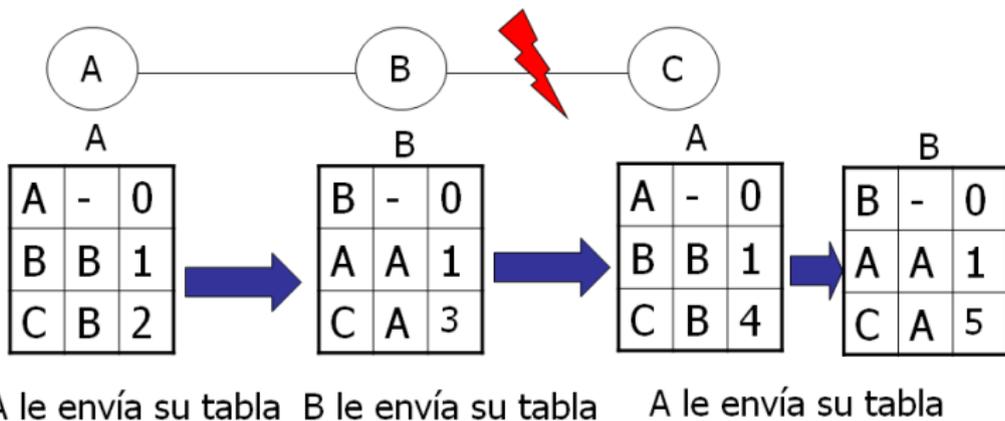
Cuenta a Infinito



Cuenta a Infinito ...



Cuenta a Infinito ...

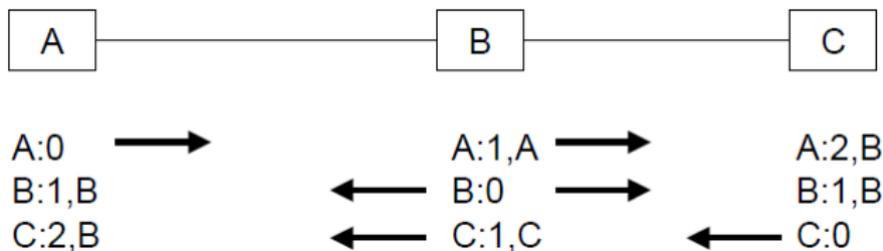


Para acelerar la convergencia se toma infinito=16

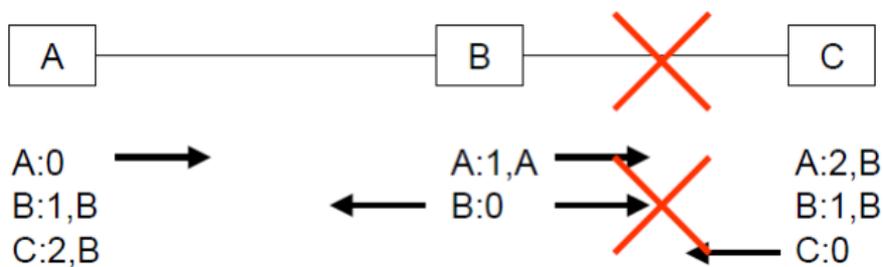
Limita la escalabilidad

Split Horizon

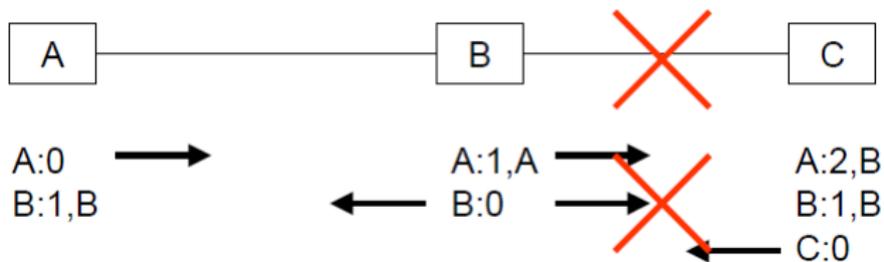
Compensa los efectos de la cuenta a infinito
 Los routers no anuncian rutas por las interfaces que las aprendieron(recibieron)



Split Horizon ...

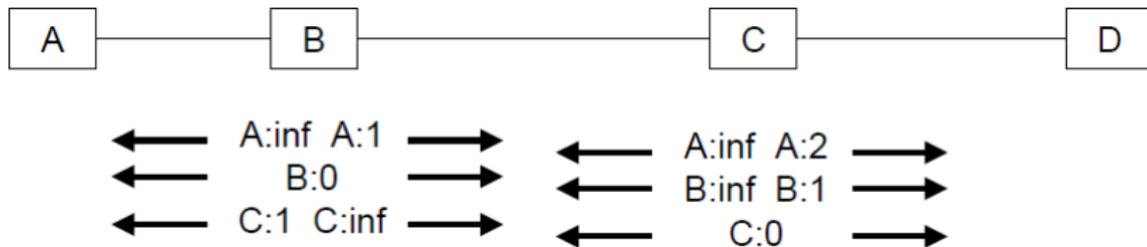


Split Horizon ...

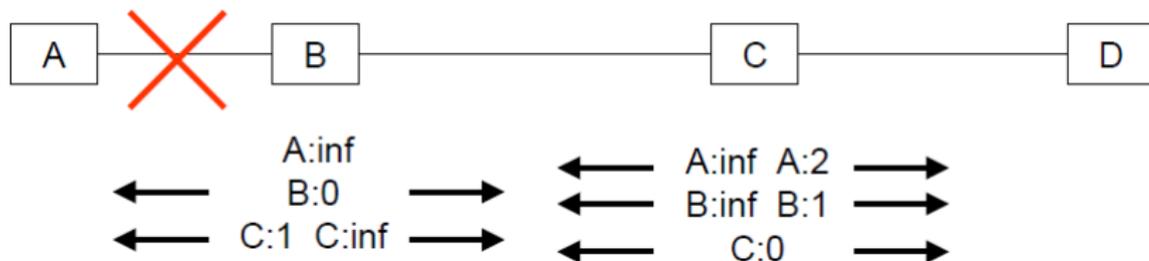


Split Horizon con Poison Reverse

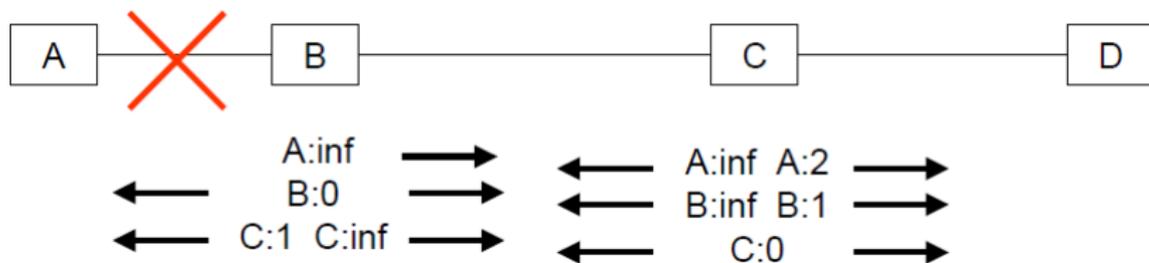
Las rutas se publican pero con métrica infinito



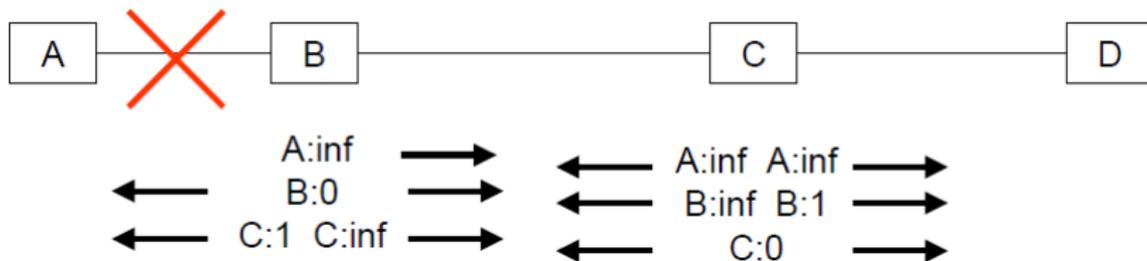
Split Horizon con Poison Reverse ...



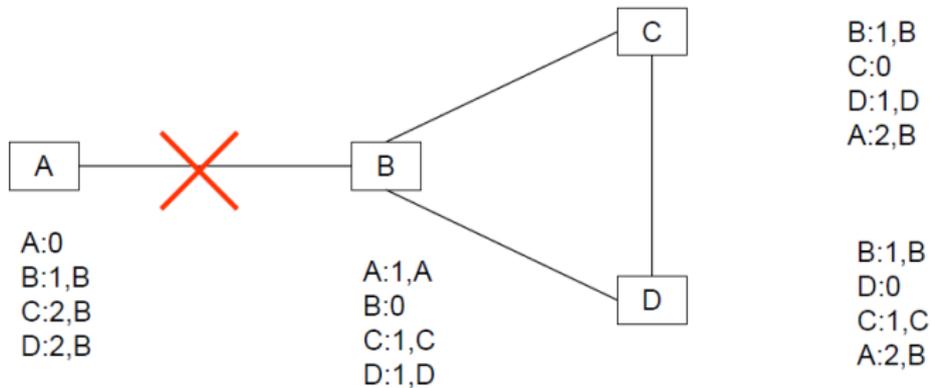
Split Horizon con Poison Reverse ...



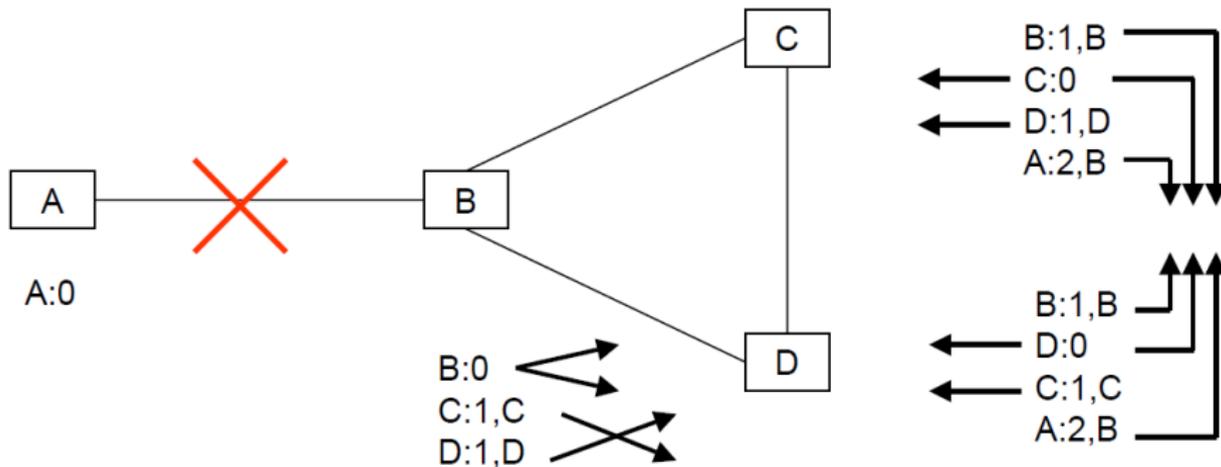
Split Horizon con Poison Reverse ...



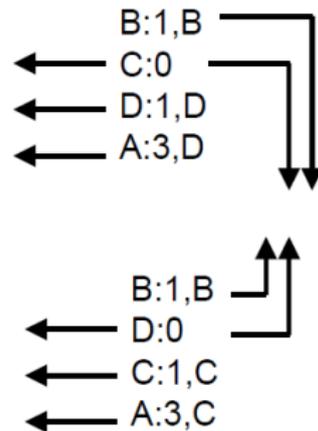
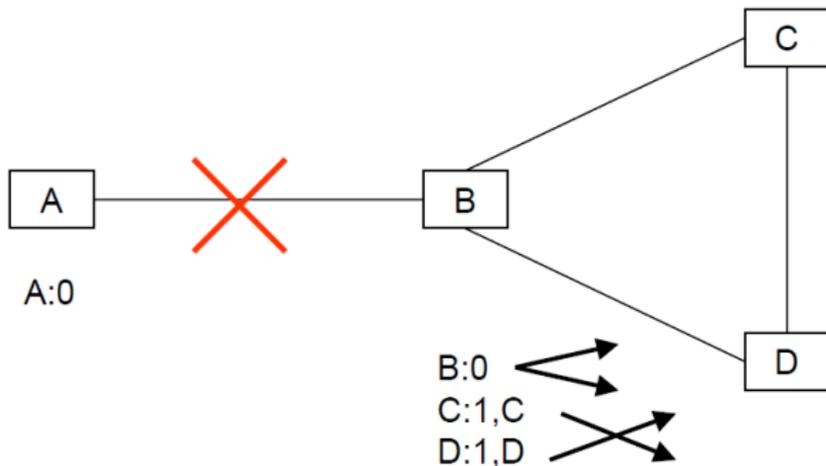
Pero ...



Pero ...



Pero ...



Más mejoras

- ✓ Se asocia un timer con cada entrada de la tabla
- ✓ Si no se actualiza la entrada al cabo del temporizador entonces se marca con infinito dicha entrada
- ✓ Normalmente se lo configura 6 veces el intervalo de transmisión
- ✓ Un cambio en la tabla provoca la publicación del mismo, “triggered update”
- ✓ Publicación de tablas cada 90 segundos
- ✓ Hold-down: Si se recibe un incremento en alguna ruta se pasa su métrica a infinito durante el timer de hold-down
- ✓ Si luego se recibe el incremento nuevamente se lo adopta

Más mejoras

- ✓ Se asocia un timer con cada entrada de la tabla
- ✓ Si no se actualiza la entrada al cabo del temporizador entonces se marca con infinito dicha entrada
- ✓ Normalmente se lo configura 6 veces el intervalo de transmisión
- ✓ Un cambio en la tabla provoca la publicación del mismo, “triggered update”
- ✓ Publicación de tablas cada 90 segundos
- ✓ Hold-down: Si se recibe un incremento en alguna ruta se pasa su métrica a infinito durante el timer de hold-down
- ✓ Si luego se recibe el incremento nuevamente se lo adopta

Más mejoras

- ✓ Se asocia un timer con cada entrada de la tabla
- ✓ Si no se actualiza la entrada al cabo del temporizador entonces se marca con infinito dicha entrada
- ✓ Normalmente se lo configura 6 veces el intervalo de transmisión
- ✓ Un cambio en la tabla provoca la publicación del mismo, “triggered update”
- ✓ Publicación de tablas cada 90 segundos
- ✓ Hold-down: Si se recibe un incremento en alguna ruta se pasa su métrica a infinito durante el timer de hold-down
- ✓ Si luego se recibe el incremento nuevamente se lo adopta

Más mejoras

- ✓ Se asocia un timer con cada entrada de la tabla
- ✓ Si no se actualiza la entrada al cabo del temporizador entonces se marca con infinito dicha entrada
- ✓ Normalmente se lo configura 6 veces el intervalo de transmisión
- ✓ Un cambio en la tabla provoca la publicación del mismo, “triggered update”
- ✓ Publicación de tablas cada 90 segundos
- ✓ Hold-down: Si se recibe un incremento en alguna ruta se pasa su métrica a infinito durante el timer de hold-down
- ✓ Si luego se recibe el incremento nuevamente se lo adopta

Más mejoras

- ✓ Se asocia un timer con cada entrada de la tabla
- ✓ Si no se actualiza la entrada al cabo del temporizador entonces se marca con infinito dicha entrada
- ✓ Normalmente se lo configura 6 veces el intervalo de transmisión
- ✓ Un cambio en la tabla provoca la publicación del mismo, “triggered update”
- ✓ Publicación de tablas cada 90 segundos
- ✓ Hold-down: Si se recibe un incremento en alguna ruta se pasa su métrica a infinito durante el timer de hold-down
- ✓ Si luego se recibe el incremento nuevamente se lo adopta

Más mejoras

- ✓ Se asocia un timer con cada entrada de la tabla
- ✓ Si no se actualiza la entrada al cabo del temporizador entonces se marca con infinito dicha entrada
- ✓ Normalmente se lo configura 6 veces el intervalo de transmisión
- ✓ Un cambio en la tabla provoca la publicación del mismo, “triggered update”
- ✓ Publicación de tablas cada 90 segundos
- ✓ Hold-down: Si se recibe un incremento en alguna ruta se pasa su métrica a infinito durante el timer de hold-down
- ✓ Si luego se recibe el incremento nuevamente se lo adopta

Más mejoras

- ✓ Se asocia un timer con cada entrada de la tabla
- ✓ Si no se actualiza la entrada al cabo del temporizador entonces se marca con infinito dicha entrada
- ✓ Normalmente se lo configura 6 veces el intervalo de transmisión
- ✓ Un cambio en la tabla provoca la publicación del mismo, “triggered update”
- ✓ Publicación de tablas cada 90 segundos
- ✓ Hold-down: Si se recibe un incremento en alguna ruta se pasa su métrica a infinito durante el timer de hold-down
- ✓ Si luego se recibe el incremento nuevamente se lo adopta

Routing Information Protocol

- ✓ 1960: Se realizó el ruteo por distancia vectorial en ARPANET
- ✓ 1982: Desarrollo de routed para BSD Unix
- ✓ 1988: RIP v1 (RFC 1058)
- ✓ Ruteo Classful
- ✓ 1993: RIP v2 (RFC 1388)
- ✓ 1998: RIP v2 (RFC 2453). STD 56

Routing Information Protocol

- ✓ 1960: Se realizó el ruteo por distancia vectorial en ARPANET
- ✓ 1982: Desarrollo de routed para BSD Unix
- ✓ 1988: RIP v1 (RFC 1058)
- ✓ Ruteo Classful
- ✓ 1993: RIP v2 (RFC 1388)
- ✓ 1998: RIP v2 (RFC 2453). STD 56

Routing Information Protocol

- ✓ 1960: Se realizó el ruteo por distancia vectorial en ARPANET
- ✓ 1982: Desarrollo de routed para BSD Unix
- ✓ 1988: RIP v1 (RFC 1058)
- ✓ Ruteo Classful
- ✓ 1993: RIP v2 (RFC 1388)
- ✓ 1998: RIP v2 (RFC 2453). STD 56

Routing Information Protocol

- ✓ 1960: Se realizó el ruteo por distancia vectorial en ARPANET
- ✓ 1982: Desarrollo de routed para BSD Unix
- ✓ 1988: RIP v1 (RFC 1058)
- ✓ Ruteo Classful
- ✓ 1993: RIP v2 (RFC 1388)
- ✓ 1998: RIP v2 (RFC 2453). STD 56

Routing Information Protocol

- ✓ 1960: Se realizó el ruteo por distancia vectorial en ARPANET
- ✓ 1982: Desarrollo de routed para BSD Unix
- ✓ 1988: RIP v1 (RFC 1058)
- ✓ Ruteo Classful
- ✓ 1993: RIP v2 (RFC 1388)
- ✓ 1998: RIP v2 (RFC 2453). STD 56

Routing Information Protocol

- ✓ 1960: Se realizó el ruteo por distancia vectorial en ARPANET
- ✓ 1982: Desarrollo de routed para BSD Unix
- ✓ 1988: RIP v1 (RFC 1058)
- ✓ Ruteo Classful
- ✓ 1993: RIP v2 (RFC 1388)
- ✓ 1998: RIP v2 (RFC 2453). STD 56

RIP

- ✓ Métrica: hop count. Infinito = 16
- ✓ Convergencia lenta
- ✓ Soporta “subnetting”
- ✓ No soporta VLSM
- ✓ No soporta autenticación

RIP

- ✓ Métrica: hop count. Infinito = 16
- ✓ Convergencia lenta
- ✓ Soporta “subnetting”
- ✓ No soporta VLSM
- ✓ No soporta autenticación

RIP

- ✓ Métrica: hop count. Infinito = 16
- ✓ Convergencia lenta
- ✓ Soporta “subnetting”
- ✓ No soporta VLSM
- ✓ No soporta autenticación

RIP

- ✓ Métrica: hop count. Infinito = 16
- ✓ Convergencia lenta
- ✓ Soporta “subnetting”
- ✓ No soporta VLSM
- ✓ No soporta autenticación

RIP

- ✓ Métrica: hop count. Infinito = 16
- ✓ Convergencia lenta
- ✓ Soporta “subnetting”
- ✓ No soporta VLSM
- ✓ No soporta autenticación

RIP – Estructura



- Port UDP
520

Command	Versión	000.....
Address family		000.....
Dirección IP		
00000000		
00000000		
Métrica (1-16)		
Hasta 24 rutas más, 20 bytes por ruta		

Mensaje RIP

- ✓ Los comandos son:
 - 1: request
 - 2: response

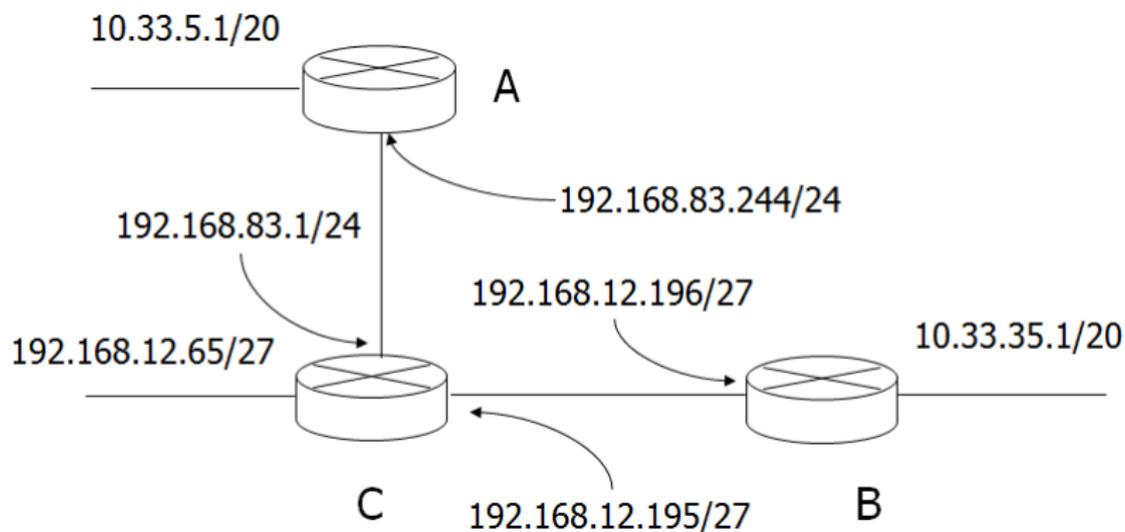
- ✓ Version : 1

- ✓ Address family:
 - 2 para IP
 - 0 para requerir la tabla completa

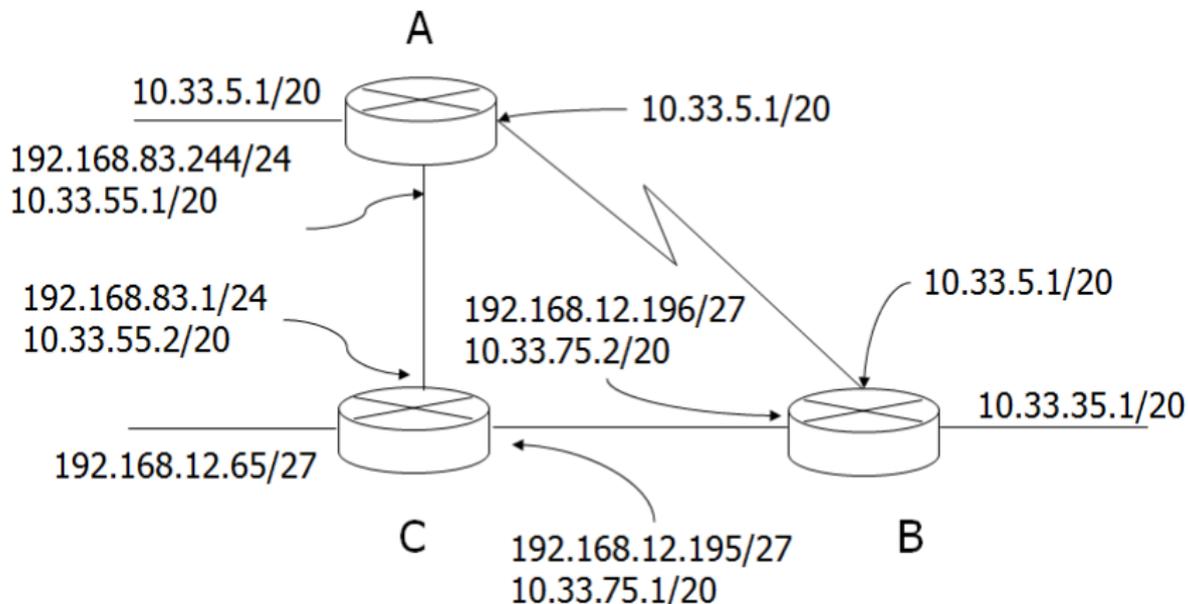
RIP – Subredes

- ✓ Los updates enviados a través de una interfaz de la red X, si contienen rutas de subredes de la red X, contienen los números de las subredes pero no las máscaras correspondientes
- ✓ Los updates enviados a través de una interfaz de la red X, si contienen rutas de subredes de la red Y, se suman en una sola ruta a la red Y
- ✓ Al recibir un update que contiene rutas de subredes de la red X, el receptor asume que la máscara correspondiente es la misma que la que él usa en una interfaz con dirección de la red X
- ✓ Al recibir un update de la red X, si el receptor no tiene interfaces en la red X, considera la ruta como ruta a la clase A, B, o C completa de X

Subredes Discontinuas



Alternativas



RIP v2

- ✓ Es una extensión de RIP v1
- ✓ Las extensiones de RIP v2 se colocan en los campos sin usar de la versión 1
- ✓ En lugar de broadcast utiliza multicast 224.0.0.9
- ✓ Soporta:
 - VLSM
 - Autenticación
- ✓ RFCs 1387-88-89.(1993)
- ✓ RFC 2453 (1998)

RIP v2 - Estructura del mensaje

IP Header	UDP Header	Mensaje RIP v 2
Command	Versión	000.....
Address family		Route tag
Dirección IP		
Máscara de subred		
Dirección de próximo salto		
Métrica (1-16)		
Hasta 24 rutas más, 20 bytes por ruta		

RIP v2 con Autenticación

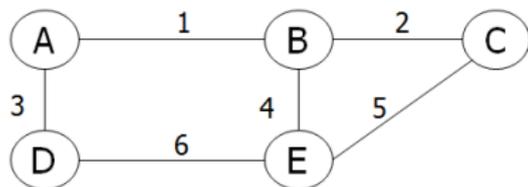
Command	Version (2)	Reserved (0)
FFFF		Auth type
Auth Data (bytes 0-3)		
Auth Data (bytes 4-7)		
Auth Data (bytes 8-11)		
Auth Data (bytes 12-15)		

- ✓ La codificación del Address Family es FFFF
- ✓ Ahora le seguirán a esta autenticación 24 rutas RIP
- ✓ Auth type:
 - 2: Password plana
 - 3: MD-5; SHA

Sincronización

- ✓ En 1993 se reportó un pico de tráfico y/o pérdida de paquetes en una red cada 30 seg aprox
- ✓ Los routers RIP se habían sincronizado
- ✓ Todos enviaban simultáneamente su tabla a los vecinos
- ✓ Se agregó un intervalo de aleatorización al timer tradicional

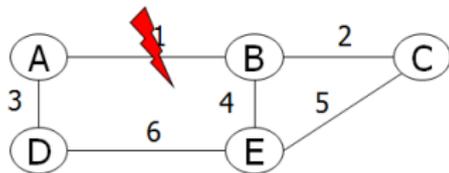
Link State



Todos los nodos tienen una copia de la red
Calculan el camino óptimo a todos los destinos

Desde	Hasta	Link	Distancia
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Flooding



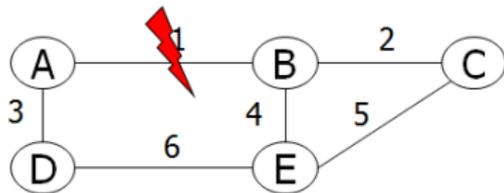
Si se cae el enlace 1 el nuevo estado se carga en la base
 A y B detectan la caída
 El nuevo estado se carga en la base

- ✓ Se transmite a los demás nodos
- ✓ La transmisión es por flooding
- ✓ A envía a D
 - Desde A a B, link=1, distancia = inf
- ✓ D lo pasa a E
- ✓ E a B y C
- ✓ El mensaje puede circular
- ✓ Para evitarlo → número identificador

Algoritmo de Flooding

- ✓ Recibir el mensaje. Buscar el registro en la base
- ✓ Si no existe, agregarlo y hacer “broadcast” del mensaje
- ✓ Si el número en la base es menor que el recibido, reemplazarlo y enviarlo
- ✓ Si el número es mayor, transmitir el registro de la base a través del enlace por donde se recibió el mensaje
- ✓ Si son iguales no hacer nada
- ✓ El “broadcast” se hace sobre todos los enlaces, excepto sobre el que se recibió el mensaje

Flooding...



El mensaje que A contiene es:
 $A \rightarrow B$, link=1, dist=1, núm= 1

- ✓ El mensaje que A transmite a D es:
 Desde A a B, link =1, dist = inf, núm = 2
- ✓ D actualiza su base y hace flooding a E, a su vez E a B y C
- ✓ B y C se transmiten el mensaje entre sí pero como el núm es igual se detiene el flooding

Algoritmo de Dijkstra

- ✓ Algoritmo SPF, (Shortest Path First)
- ✓ Computa el camino más corto entre un nodo y el resto
- ✓ Divide a los nodos en dos conjuntos:
 - {E}: El conjunto de nodos evaluados para cuyos elementos se conoce el camino mínimo
 - {R}: El resto
- ✓ Conjunto {O}: Lista ordenada de rutas

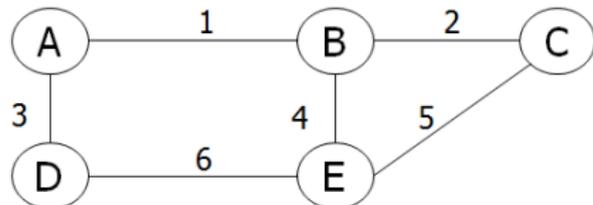
Algoritmo de Dijkstra. . .

- 1 Inicio
Inicializar E con el nodo origen
Inicializar R con el resto de los nodos
Inicializar O con la lista de caminos a partir del origen por orden creciente de métrica
- 2 Si la lista O está vacía o la métrica del primer elemento es inf, entonces marcar los nodos que queden en R como inalcanzables y el algoritmo finaliza
- 3 Examinar P, el camino más corto en O. Quitar P de O. Si V es el nodo extremo de P y V ya está en E, continuar 2.-; Caso contrario P es el camino más corto a V y pasar V de R a E
- 4 Construir un nuevo conjunto de caminos posibles, concatenando P y cada uno de los links a partir de V. Al costo anterior sumarle el del link que se agregó. Insertarlos en la lista O en su orden correspondiente y pasar a 2

Algoritmo de Dijkstra. . .

- ✓ El número de caminos a considerar es el número de enlaces en la red
- ✓ El número de iteraciones es $M \cdot \log M$; donde M es el número de enlaces
- ✓ En el caso vectorial ese número es $M \cdot N$; dónde N es el número de nodos

Algoritmo de Dijkstra – Ejemplo



Tomamos como nodo origen a D

- $E=\{D\}$; $R=\{A,B,C,E\}$; $O=\{D-A(1); D-E(1)\}$
- Examinamos D-A; A no está en E \rightarrow es la ruta más corta
- $P=\{D-A\}$ y movemos A de R a E
- $E=\{D,A\}$; $R=\{B,C,E\}$
- Incorporamos a O las rutas a partir de D-A,
 $O=\{D-E(1);D-A-B(2)\}$
- Volvemos al paso 2 del algoritmo

Algoritmo de Dijkstra – Ejemplo. . .

- Tomamos el camino D-E
E no está en E \rightarrow D-E es el camino más corto a E,
 $P=\{D-A, D-E\}$,
Movemos E de R a E
 $E=\{D,A,E\}$; $R=\{B,C\}$
- Incorporamos los nuevos caminos a O a partir de D-E
 $O=\{D-A-B(2); D-E-B(2); D-E-C(2)\}$
Volvemos al paso 2
- $O=\{D-E-B(2); D-E-C(2)\}$, D-A-B , B no está en E \rightarrow D-A-B es camino
- $P=\{D-A; D-E; D-A-B\}$

Algoritmo de Dijkstra – Ejemplo. . .

- Pasamos B de R a E
 $E = \{D, A, E, B\}$; $R = \{C\}$
- Incorporamos los nuevos caminos en O a partir de D-A-B
 $O = \{D-E-B(2); D-E-C(2); D-A-B-E(3); D-A-B-C(3)\}$
- Quitamos D-E-B de O, al analizarla comprobamos que B ya está en E por lo que volvemos al paso 2
 $O = \{D-E-C(2); D-A-B-E(3); D-A-B-C(3)\}$
- Quitamos D-E-C de O; como C no está en E, es un camino:
 $P = \{D-A; D-E; D-A-B; D-E-C\}$ y pasamos C de R a E

Algoritmo de Dijkstra – Ejemplo. . .

- $E = \{D, A, E, B, C\} ; R \{\}$
- Incorporamos los nuevos caminos a partir de D-E-C
- $O = \{D-A-B-E(3); D-A-B-C(3); D-E-C-B(3)\}$
- Volvemos al paso 2
- Todos los caminos de O tienen el último nodo en E por lo tanto O queda vacío y finaliza el algoritmo

SPF

- ✓ Converge más rápido
- ✓ No genera lazos cerrados
- ✓ Soporta múltiples métricas
- ✓ Soporta múltiples caminos. Modificar el algoritmo
- ✓ Mejor interfase al ruteo externo
- ✓ Menor tráfico entre routers

OSPF

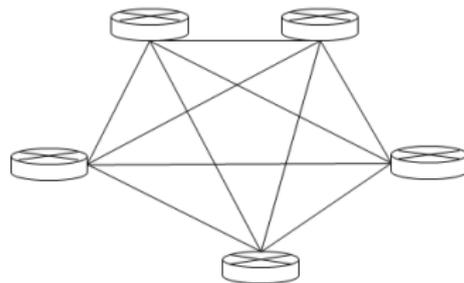
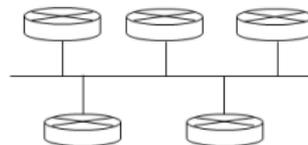
- ✓ Versión 2, abril 1998. RFC 2328. STD 54
- ✓ Encapsulado en IP directamente, protocolo = 89
- ✓ Soporta sistemas jerárquicos
- ✓ No necesita conocer la topología completa de la red

OSPF – Características

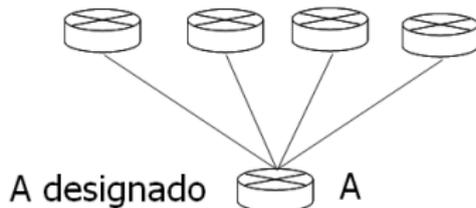
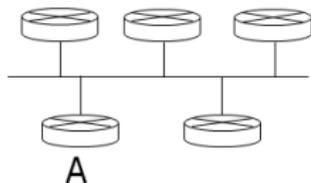
- ✓ Métrica = costo
- ✓ Costo = C_i (Costo de cada enlace)
- ✓ C_i = función del ancho de banda
- ✓ $C_i = 10^8 / \text{ancho de banda (bps)}$
- ✓ Se elige el costo menor
- ✓ Convergencia rápida
- ✓ VLSM
- ✓ Mensajes cortos
- ✓ Triggered updates
- ✓ Full flooding cada 30 minutos
- ✓ Consumo de procesamiento y memoria

OSPF, redes broadcast y multiacceso

Si n es el número de routers
tenemos $n(n - 1)/2$ adyacencias
Cada router enviará $n - 1$ LSA,
resultando $\approx n^2$ LSAs enviados en
total
Flooding caótico. Múltiples copias
del mismo LSA en la misma red
Se crea el Designated Router (DR)



OSPF, redes broadcast

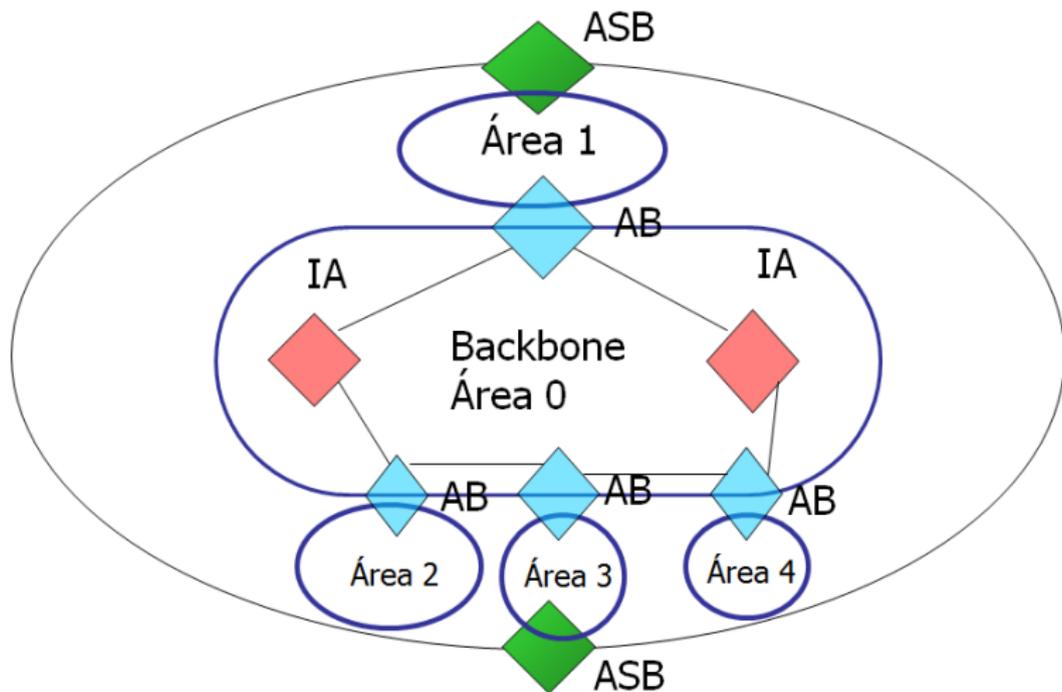


- Si un router debe transmitir el estado de un enlace, sólo lo transmite al designado. Utiliza la dirección multicast 224.0.0.6 (AllDRouters)
- Si es un nuevo aviso el “designado” lo envía por “flooding” a la dirección multicast 224.0.0.5 (AllRouters)

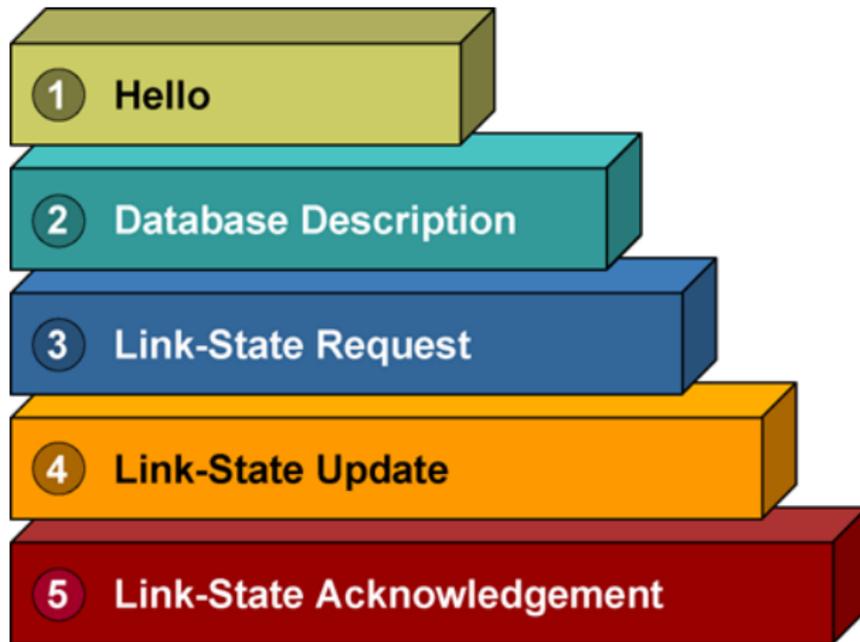
OSPF, jerarquía

- ✓ Dividir la red en sectores independientes interconectados por un “backbone”
- ✓ Los sectores son las “áreas”. Se comportan como redes independientes
- ✓ La base de datos incluye los enlaces del área
- ✓ El “flooding” se detiene en los bordes del área
- ✓ El costo resulta proporcional al tamaño del área y no al de la red
- ✓ Para mantener la integridad de la red existen routers que interconectan el área con el backbone
- ✓ Son los routers de borde de área
- ✓ Mantienen varias bases de datos, una para cada área a la que pertenecen

OSPF – Áreas



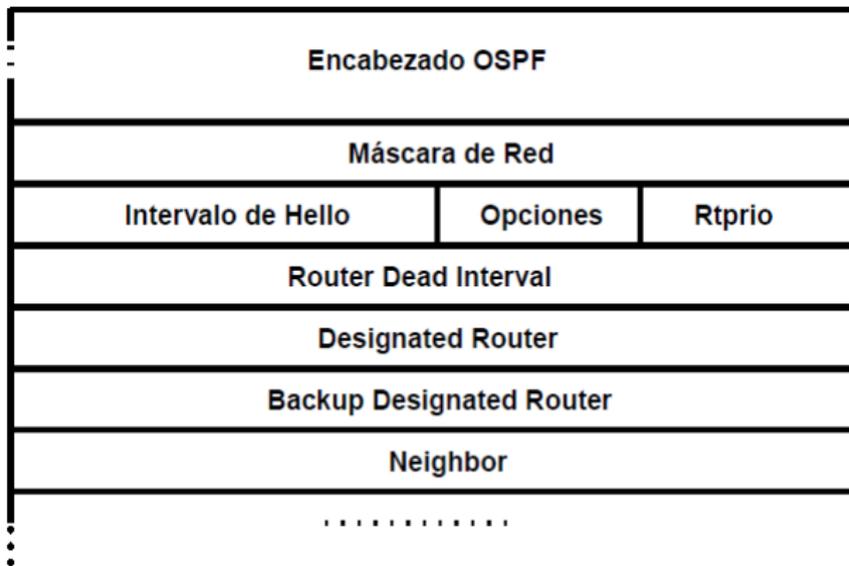
Paquetes OSPF

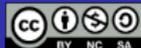


Encabezado OSPF

Versión	Tipo	Largo de paquete
Router Id		
Area Id		
Checksum	Tipo de autenticación	
Datos de Autenticación		
Datos de Autenticación		

Paquete de Hello





**Atribución-NoComercial-CompartirIgual
4.0 Internacional (CC BY-NC-SA 4.0)**

Esta obra está sujeta a la licencia Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) de Creative Commons.

Para detalle de esta licencia visite

<https://creativecommons.org/licenses/by-nc-sa/4.0/>