

Protocolos de Routing. Routing entre robots móviles

Redes de Ordenadores para Robots y Máquinas Inteligentes

Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación (GSyC)

Abril 2024



©2024 GSyC
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike
disponible en <http://creativecommons.org/licenses/by-sa/4.0/deed.es>

- 1 Protocolos de Routing
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

Contenidos

- 1 Protocolos de Routing
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

Contenidos

- 1 Protocolos de Routing
 - Introducción
 - Protocolos de routing triviales
 - Algoritmo de Dijkstra
 - Familias de Protocolos de Routing
 - Protocolos basados en Vector de Distancia
 - Protocolos basados en Estado de Enlace
 - Alternativas al routing tradicional en IP
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

Tabla de *routing*

Tabla de routing (tabla de encaminamiento)

Tabla que consulta:

- una máquina (router o no) cada vez que tiene que enviar un paquete a un destino
- un router cada vez que recibe un paquete que no es para él, y así poder reenviarlo

Destino	Gateway	Máscara	Interfaz
112.23.24.0	17.0.0.2	255.255.255.0	eth0
141.12.53.0	18.0.0.3	255.255.255.0	eth1
...

- La tabla de routing puede rellenarse:
 - **Estáticamente**: mediante entradas añadidas manualmente por el administrador
 - **Dinámicamente**: mediante un protocolo de **routing**

Algoritmos y protocolos de routing

Algoritmo de routing

Procedimiento por el cual los routers descubren las mejores rutas para cada destino.

- Como parte del algoritmo de routing, los routers tienen que enviarse información entre ellos mediante mensajes. Es decir: los algoritmos de routing incluyen un **protocolo**, que recibe el nombre de **protocolo de routing**.
- Muchas veces se utiliza el término “protocolo de routing” como sinónimo de “algoritmo de routing”.
- El propósito de los algoritmos/protocolos de routing es que en cada router en todo momento su **tabla de routing** esté actualizada con las mejores rutas dadas las circunstancias de la red.

Objetivos de un algoritmo/protocolo de routing

- Generar rutas óptimas, es decir, de mínimo **coste**, definiendo el coste en base a uno o más de estos parámetros:
 - número de routers intermedios
 - retardo
 - coste económico
 - aprovechamiento de la capacidad de la red
- Minimizar el tiempo en que tarda en recalcularse la tabla en cada router.
- Minimizar la información adicional que debe almacenar cada router para ejecutar el algoritmo
- Minimizar el número y frecuencia de mensajes entre routers que es necesario enviar para tener actualizadas las rutas
- Robustez: evitar agujeros negros, evitar bucles, evitar oscilaciones en las rutas

Contenidos

- 1 Protocolos de Routing
 - Introducción
 - Protocolos de routing triviales
 - Algoritmo de Dijkstra
 - Familias de Protocolos de Routing
 - Protocolos basados en Vector de Distancia
 - Protocolos basados en Estado de Enlace
 - Alternativas al routing tradicional en IP
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

Protocolo de Inundación

- Es un protocolo muy simple:
 - ① Cada paquete recibido por un nodo es reenviado a todos los vecinos excepto al que se lo envió a él.
 - ② Los paquetes van etiquetados de forma que un nodo pueda descartar un paquete recibido si ya lo recibió y reenvió anteriormente.
- Suele utilizarse cuando no se tiene información alguna de encaminamiento, o como parte de otro algoritmo más complejo.

Protocolo de Aprendizaje

- Funcionamiento:
 - ❶ Cada nodo mantiene una tabla con parejas **(Destino, enlace por el que se alcanza)** que va actualizando según los paquetes que va recibiendo.
 - ❷ Al recibir un paquete, se fija en su origen y en el enlace por el que le ha llegado, apuntando en su tabla que cuando ese nodo sea destino de un paquete lo enviará por ese enlace
 - ❸ Cuando para un destino no hay entrada en la tabla, se envía por inundación.
- Es el protocolo que utilizan los *switches*, pero también pueden utilizarlo *routers*.

Contenidos

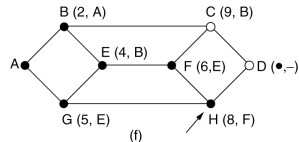
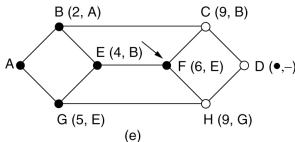
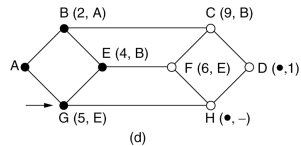
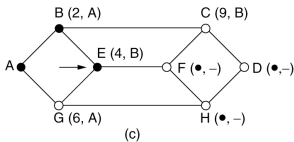
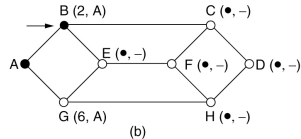
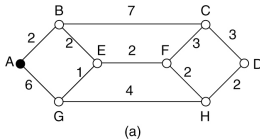
- 1 Protocolos de Routing
 - Introducción
 - Protocolos de routing triviales
 - **Algoritmo de Dijkstra**
 - Familias de Protocolos de Routing
 - Protocolos basados en Vector de Distancia
 - Protocolos basados en Estado de Enlace
 - Alternativas al routing tradicional en IP
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

Algoritmo de Dijkstra

- Cuando se conocen los costes (o distancias) entre los nodos adyacentes, el Algoritmo de Dijkstra permite encontrar los caminos de **distancia mínima desde un nodo al resto**.
- En una red, cada router ejecutará el algoritmo para encontrar rutas desde él al resto.
- Algoritmo:
 - 1 Se trabaja con dos conjuntos de nodos:
 - P: Nodos con su encaminamiento ya resuelto (permanentes)
 - T: Nodos aún no resueltos (tentativos)
 - 2 Inicialmente P sólo contiene el nodo inicial, p_0 .
 - 3 Para cada nodo t_i de T se recalcula su distancia a p_0 :
 - si t_i no está directamente conectado a ningún nodo de P, su distancia a p_0 es infinita (por ahora)
 - en caso contrario, se elige el menor valor entre la distancia calculada hasta ahora de p_0 a t_i y el valor suma de la distancia de p_0 al p_z (último nodo añadido a P) y la distancia directa de p_z a t_i .
 - 4 El nodo de T que presente una menor distancia a p_0 se pasa a P. Si aún quedan nodos en T, se repite el paso anterior.

Algoritmo de Dijkstra: Ejemplo

La figura muestra los 5 primeros pasos utilizados en calcular el camino más corto desde A al resto de nodos. La flecha indica el nodo sobre el que se está actuando:



Contenidos

- 1 Protocolos de Routing
 - Introducción
 - Protocolos de routing triviales
 - Algoritmo de Dijkstra
 - **Familias de Protocolos de Routing**
 - Protocolos basados en Vector de Distancia
 - Protocolos basados en Estado de Enlace
 - Alternativas al routing tradicional en IP
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

Familias de Protocolos de Routing

- Los protocolos de routing más usados en redes pueden clasificarse en dos grandes grupos:
 - Protocolos de **Vector de Distancia** (*Distance Vector Protocols, DVP*)
 - Protocolos de **Estado de Enlace** (*Link State Protocols, LSP*)
- Hay protocolos que no encajan completamente en ninguna de las dos familias, pero sí son “más próximos” a una de las dos.

Contenidos

- 1 Protocolos de Routing
 - Introducción
 - Protocolos de routing triviales
 - Algoritmo de Dijkstra
 - Familias de Protocolos de Routing
 - **Protocolos basados en Vector de Distancia**
 - Protocolos basados en Estado de Enlace
 - Alternativas al routing tradicional en IP
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

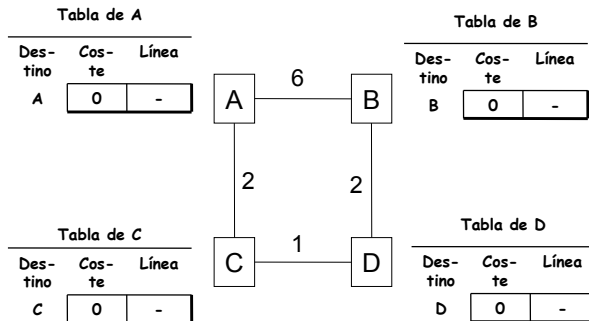
Funcionamiento básico

- 1 Cada nodo conoce el coste para llegar a sus vecinos directos.
- 2 Cada nodo mantiene una tabla con la siguiente información (tabla de routing):
(Destino, Coste, Vecino por el que se alcanza)
- 3 Cada nodo envía periódicamente a sus vecinos su **Vector de Distancia** a todos los destinos, formado por los pares:
(Destino, Coste)
- 4 Cada nodo estudia los vectores de distancia que recibe de sus vecinos para seleccionar para cada destino el vecino por el que tendrá menor coste, y actualiza su tablas de encaminamiento consecuentemente.

Ejemplo

INICIALMENTE

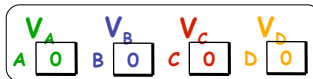
Inicialmente cada nodo sólo se conoce a sí mismo.



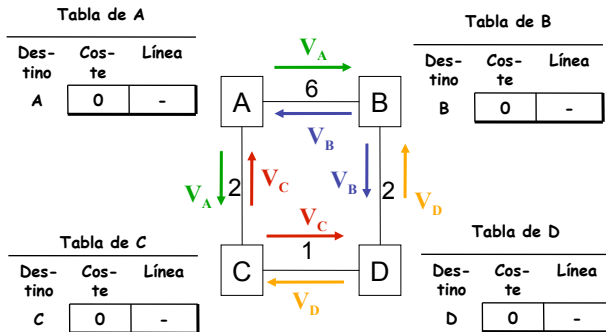
Ejemplo

INTERCAMBIO 1

Cada nodo intercambia su vector distancia con sus vecinos.



Vectores distancia iniciales que envía cada una de las máquinas



Ejemplo

INTERCAMBIO 1

Cada nodo intercambia su vector distancia con sus vecinos:
- A actualiza su tabla con los vectores recibidos.

Vectores distancia
que recibe A

Tabla de A

Des- tino	Cos- te	Línea
A	0	-
B	6	B
C	2	C

Tabla de B

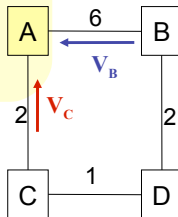
Des- tino	Cos- te	Línea
B	0	-

Tabla de C

Des- tino	Cos- te	Línea
C	0	-

Tabla de D

Des- tino	Cos- te	Línea
D	0	-



Ejemplo

INTERCAMBIO 1

Cada nodo intercambia su vector distancia con sus vecinos:

- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos

Tabla de A

Des- tino	Cos- te	Línea
A	0	-
B	6	B
C	2	C

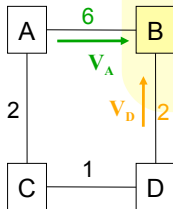


Tabla de C

Des- tino	Cos- te	Línea
C	0	-

Tabla de B

Des- tino	Cos- te	Línea
B	0	-
A	6	A
D	2	D

Vectores distancia
que recibe B



Tabla de D

Des- tino	Cos- te	Línea
D	0	-

Ejemplo

INTERCAMBIO 1

Cada nodo intercambia su vector distancia con sus vecinos:

- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos.
- **C actualiza su tabla con los vectores recibidos**

Tabla de A

Des-tino	Cos-te	Línea
A	0	-
B	6	B
C	2	C

Tabla de B

Des-tino	Cos-te	Línea
B	0	-
A	6	A
D	2	D

Vectores distancia
que recibe C

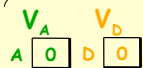
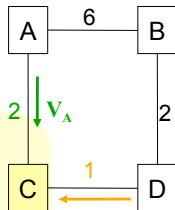


Tabla de C

Des-tino	Cos-te	Línea
C	0	-
A	2	A
D	1	D

Tabla de D

Des-tino	Cos-te	Línea
D	0	-

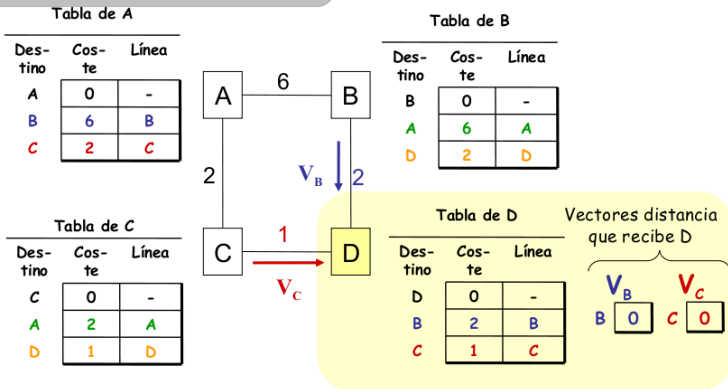


Ejemplo

INTERCAMBIO 1

Cada nodo intercambia su vector distancia con sus vecinos:

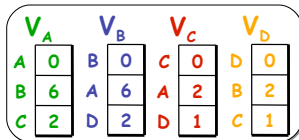
- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos.
- C actualiza su tabla con los vectores recibidos.
- **D actualiza su tabla con los vectores recibidos**



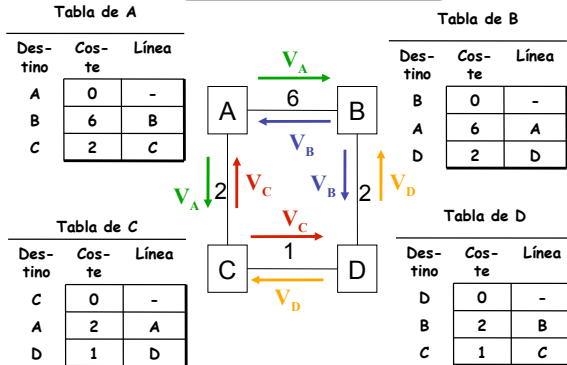
Ejemplo

INTERCAMBIO 2

Cada nodo intercambia su vector distancia con sus vecinos.



Vectores distancia que envía cada una de las máquinas



Ejemplo

INTERCAMBIO 2

Cada nodo intercambia su vector distancia con sus vecinos:
 - A actualiza su tabla con los vectores recibidos.

Vectores distancia
que recibe A

	V_B		V_C
B	0	C	0
A	6	A	2
D	2	D	1

Tabla de A

Des-tino	Cos-te	Línea
A	0	-
B	6	B
C	2	C
D	3	C

Tabla de C

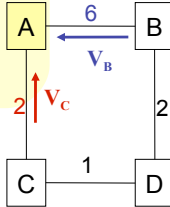
Des-tino	Cos-te	Línea
C	0	-
A	2	A
D	1	D

Tabla de B

Des-tino	Cos-te	Línea
B	0	-
A	6	A
D	2	D

Tabla de D

Des-tino	Cos-te	Línea
D	0	-
B	2	B
C	1	C

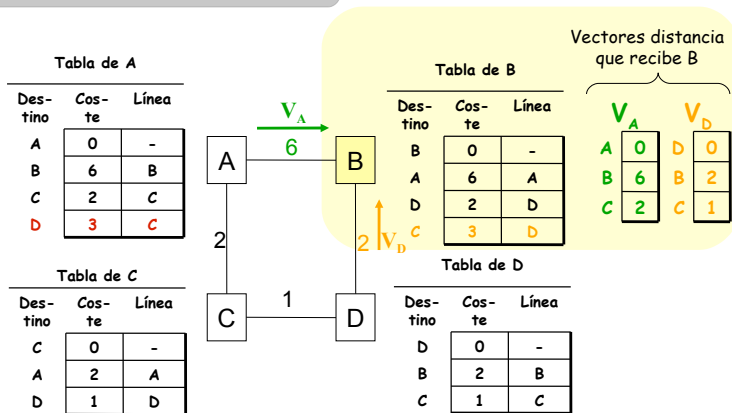


Ejemplo

INTERCAMBIO 2

Cada nodo intercambia su vector distancia con sus vecinos:

- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos



Ejemplo

INTERCAMBIO 2

Cada nodo intercambia su vector distancia con sus vecinos:

- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos.
- **C actualiza su tabla con los vectores recibidos**

Tabla de A

Des- tino	Cos- te	Línea
A	0	-
B	6	B
C	2	C
D	3	C

Tabla de B

Des- tino	Cos- te	Línea
B	0	-
A	6	A
D	2	D
C	3	D

Tabla de D

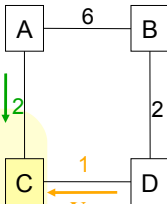
Des- tino	Cos- te	Línea
D	0	-
B	2	B
C	1	C

Tabla de C

Des- tino	Cos- te	Línea
C	0	-
A	2	A
D	1	D
B	3	D

Vectores distancia
que recibe C

V_A		V_D	
A	0	D	0
B	6	B	2
C	2	C	1

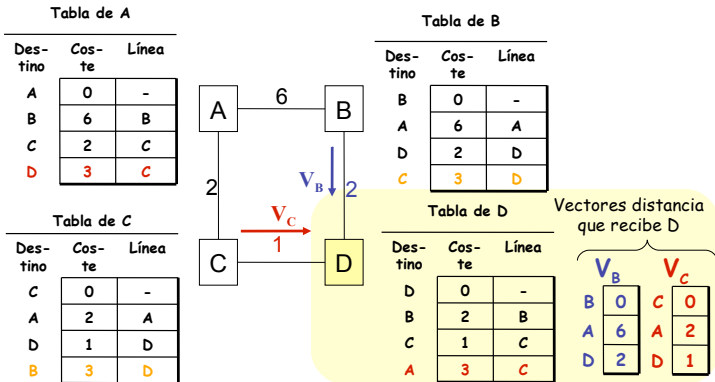


Ejemplo

INTERCAMBIO 2

Cada nodo intercambia su vector distancia con sus vecinos:

- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos.
- C actualiza su tabla con los vectores recibidos.
- **D actualiza su tabla con los vectores recibidos**



Ejemplo

INTERCAMBIO 3

Cada nodo intercambia su vector distancia con sus vecinos.

V_A	V_B	V_C	V_D
A 0	B 0	C 0	D 0
B 6	A 6	A 2	B 2
C 2	D 2	D 1	C 1
D 3	C 3	B 3	A 3

Vectores distancia que envía cada una de las máquinas

Tabla de A

Des- tino	Cos- te	Línea
A	0	-
B	6	B
C	2	C
D	3	C

Tabla de B

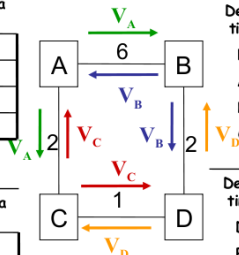
Des- tino	Cos- te	Línea
B	0	-
A	6	A
D	2	D
C	3	D

Tabla de C

Des- tino	Cos- te	Línea
C	0	-
A	2	A
D	1	D
B	3	D

Tabla de D

Des- tino	Cos- te	Línea
D	0	-
B	2	B
C	1	C
A	3	C



Ejemplo

INTERCAMBIO 3

Cada nodo intercambia su vector distancia con sus vecinos:

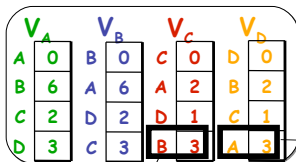
- A y B actualizan sus tablas con los vectores recibidos
- C y D no necesitan actualizarlas

Tabla de A

Des- tino	Cos- te	Línea
A	0	-
B	5	C
C	2	C
D	3	C

Tabla de C

Des- tino	Cos- te	Línea
C	0	-
A	2	A
D	1	D
B	3	D



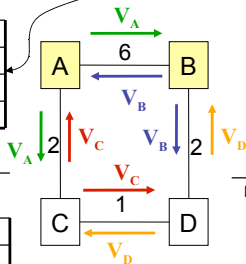
Vectores distancia que envía cada una de las máquinas

Tabla de B

Des- tino	Cos- te	Línea
B	0	-
A	5	D
D	2	D
C	3	D

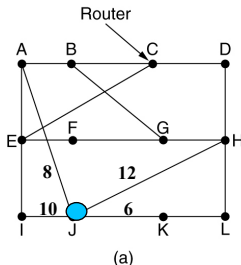
Tabla de D

Des- tino	Cos- te	Línea
D	0	-
B	2	B
C	1	C
A	3	C



Ejercicio

En la siguiente figura, J recibe los vectores distancia de sus nodos vecinos A, I, H y K. Según los datos que aparecen en la figura, actualizar los costes para llegar a todos los destinos en el nodo J:

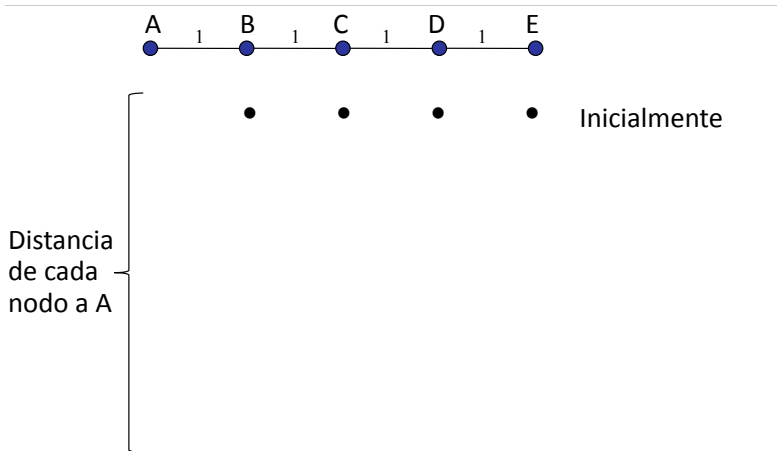


To	A	I	H	K	New estimated delay from J	
A	0	24	20	21		
B	12	36	31	28		
C	25	18	19	36		
D	40	27	8	24		
E	14	7	30	22		
F	23	20	19	40		
G	18	31	6	31		
H	17	20	0	19		
I	21	0	14	22		
J	9	11	7	10		
K	24	22	22	0		
L	29	33	9	9		
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

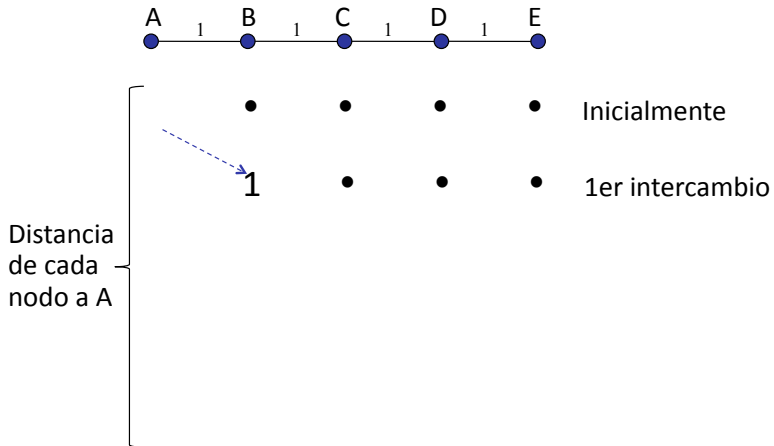
Propagación de rutas (I)

- La información acerca de mejores rutas se propaga poco a poco, consiguiéndose al cabo de un rato que todos los encaminadores tengan tablas óptimas.



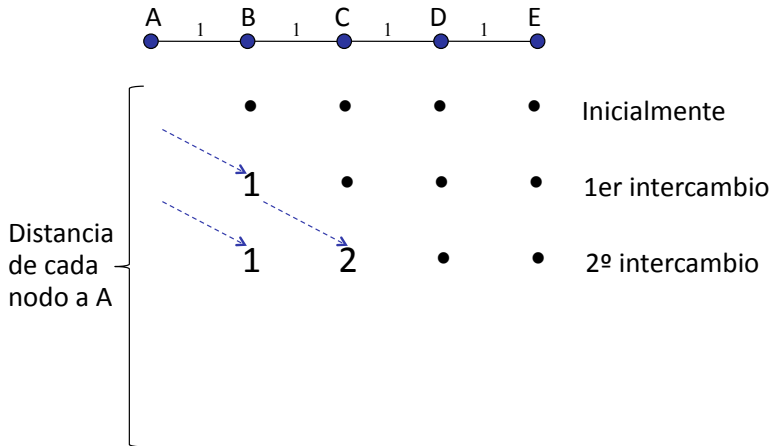
Propagación de rutas (II)

- La información acerca de mejores rutas se propaga poco a poco, consiguiéndose al cabo de un rato que todos los encaminadores tengan tablas óptimas.



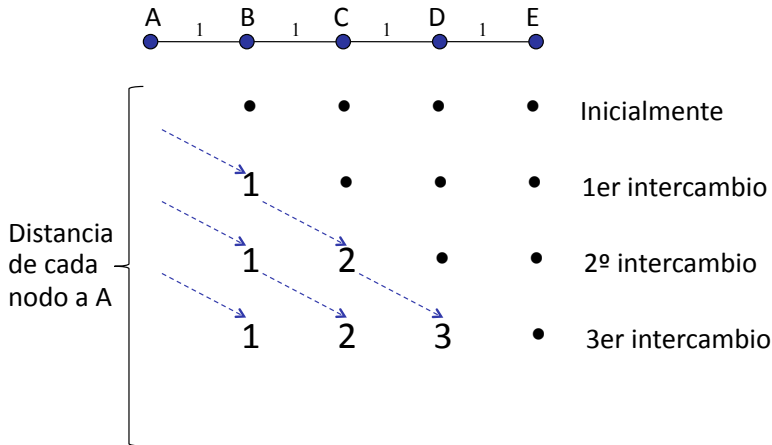
Propagación de rutas (III)

- La información acerca de mejores rutas se propaga poco a poco, consiguiéndose al cabo de un rato que todos los encaminadores tengan tablas óptimas.



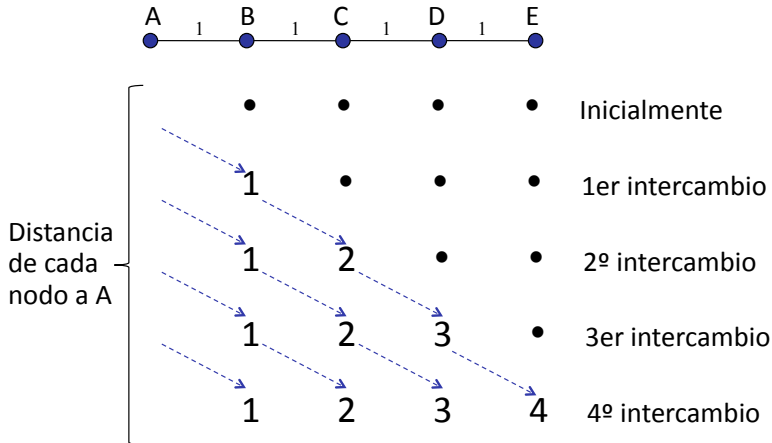
Propagación de rutas (IV)

- La información acerca de mejores rutas se propaga poco a poco, consiguiéndose al cabo de un rato que todos los encaminadores tengan tablas óptimas.



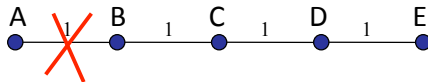
Propagación de rutas (V)

- La información acerca de mejores rutas se propaga poco a poco, consiguiéndose al cabo de un rato que todos los encaminadores tengan tablas óptimas.



Problema: Cuenta al infinito (I)

- B pierde el contacto con A

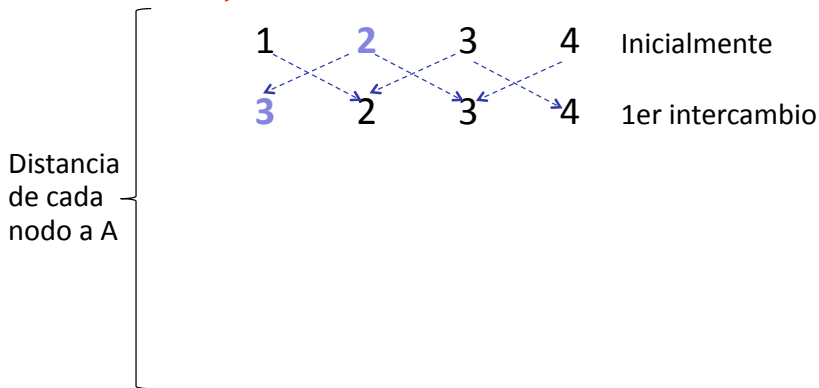
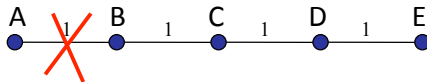


1 2 3 4 Inicialmente

Distancia
de cada
nodo a A

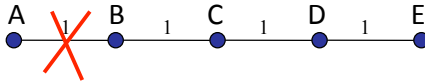
Problema: Cuenta al infinito (II)

- B pierde el contacto con A

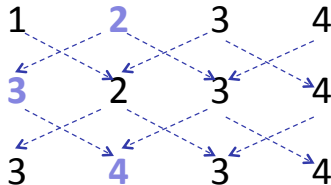


Problema: Cuenta al infinito (III)

- B pierde el contacto con A



Distancia
de cada
nodo a A



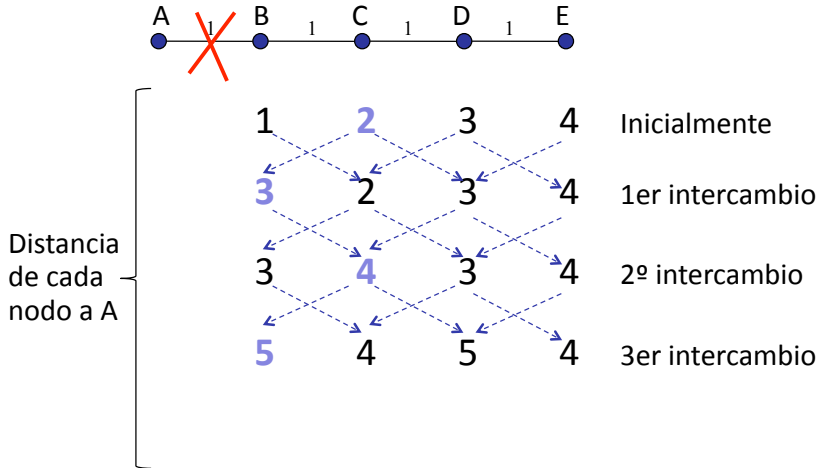
Inicialmente

1er intercambio

2º intercambio

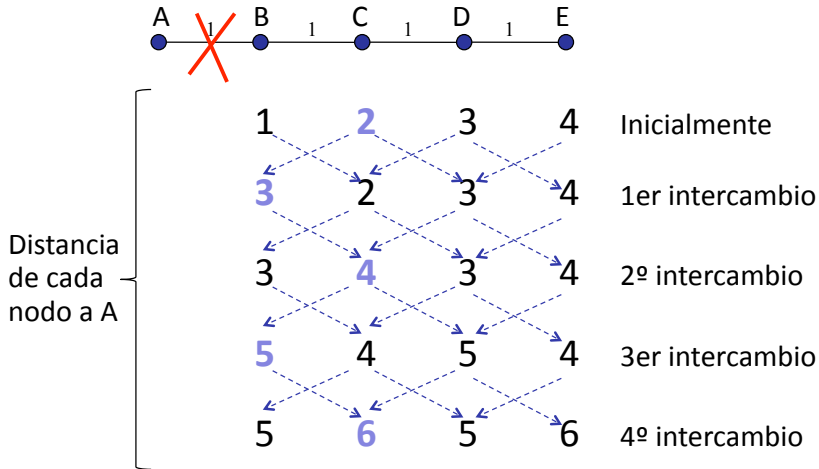
Problema: Cuenta al infinito (IV)

- B pierde el contacto con A



Problema: Cuenta al infinito (V)

- B pierde el contacto con A



Problema: Cuenta al infinito (VI)

- El problema es que todos los routers creen que sigue habiendo un camino para llegar a A, cuando no lo hay.
- Los paquetes rebotarán entre los routers, formándose un **bucle** de encaminamiento (*routing loop*), en vez de ser directamente descartados por no haber ruta al destino.
- Todos los protocolos de vector de distancia tienen algún mecanismo para paliar o solucionar este problema.
- El corolario es que con estos protocolos:
 - Las “buenas noticias” (información de nuevas rutas) se propagan rápidamente.
 - Las “malas noticias” (información de rutas que empeoran) se propagan lentamente.

Contenidos

- 1 Protocolos de Routing
 - Introducción
 - Protocolos de routing triviales
 - Algoritmo de Dijkstra
 - Familias de Protocolos de Routing
 - Protocolos basados en Vector de Distancia
 - **Protocolos basados en Estado de Enlace**
 - Alternativas al routing tradicional en IP
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

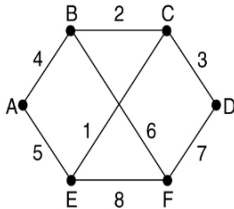
Funcionamiento básico

- 1 Cada router conoce el coste para llegar a sus nodos vecinos y construye un paquete de **Estado de Enlace** (LSP, *Link State Packet*) con esta información.
- 2 Cada router envía periódicamente **a todos** los routers de la red el paquete de Estado de Enlace con el coste para llegar a **sus vecinos**. Estos mensajes se difunden por **inundación**.
- 3 Cada router, con la información recibida, conoce la topología completa de la red y calcula el mejor camino a todos sus destinos (aplicando, por ejemplo, el algoritmo de Dijkstra).

Inundación en los protocolos de estado del enlace

- El envío de los LSP por inundación es la parte más compleja de estos protocolos: los LSP son paquetes más pequeños que los vectores de distancia, pero se genera mucho tráfico al enviarlos por inundación.
- Para controlar la inundación, los LSP llevan un número de secuencia (que asigna su creador).
- Si en una subred hay varios routers, sólo uno de ellos será elegido como encargado de enviar los paquetes por inundación.

Ejemplo



(a)

Link		State		Packets	
A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

(a) Ejemplo de subred

(b) Paquetes de estado de enlace para esa subred

Seq: número de secuencia

Age: Tiempo desde su creación

Ejemplo

- Cada nodo recibe el LSP de todos los demás nodos.
- Cada nodo aplica Dijkstra con la información recibida y construye su tabla de routing:

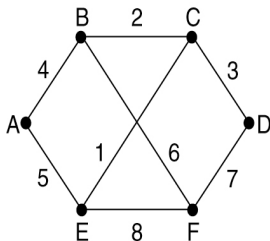


Tabla de A

Destino	Coste	Gateway
A	0	—
B	4	B
C	6	B
D	9	E
E	5	E
F	10	B

Ventajas

- Todos los nodos conocen todo el estado de toda la red. El resto de ventajas se deducen de ésta. El los protocolos de Vector de Distancias cada nodo sólo tiene una información parcial de la red: cuál es su mejor vecino para cada destino; nada más.
- Convergen más rápido y sin bucles.
- Permiten usar varias métricas para calcular el mejor camino.
- Permiten obtener varias rutas alternativas para un mismo destino, y compararlas: permiten balanceo de tráfico.

Inconvenientes

- Es imprescindible asegurar la consistencia de la información sobre la que se calculan las rutas.
- La información que se recibe de cada *router* hay que guardarla en una base de datos y ésta puede ser grande. Sobre esa base de datos se aplica Dijkstra.
- Se generan muchos mensajes para propagar la información de esa base de datos, y la gestión del envío por inundación es compleja.
- Su implementación es mucho, mucho más compleja que la de los protocolos de Vector de Distancia.

Contenidos

- 1 Protocolos de Routing
 - Introducción
 - Protocolos de routing triviales
 - Algoritmo de Dijkstra
 - Familias de Protocolos de Routing
 - Protocolos basados en Vector de Distancia
 - Protocolos basados en Estado de Enlace
 - Alternativas al routing tradicional en IP
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

Routing IP

- Funcionamiento habitual de los routers IP:
 - ① Un protocolo de routing mantiene una tabla con rutas a los posibles destinos.
 - ② Por cada paquete que se recibe, se consulta la dirección de destino en la tabla buscando la **ruta más específica** que encaja.
 - (este paso suele denominarse *forwarding* o reenvío)
- A veces el paso 1 no es sencillo de conseguir, o el paso 2 resulta demasiado lento, y se utilizan otras técnicas.

Source routing (encaminamiento en origen)

- Los paquetes enviados llevan en su cabecera toda o parte de la ruta para llegar al destino, y los routers la aplican.
- En IP una opción de la cabecera permite hacerlo de forma sencilla:
 - ① El paquete incluye una lista ordenada de direcciones IP de routers por los que debe viajar
 - ② Cada router busca en su tabla la IP del siguiente punto de paso en vez de la IP de destino
- (está opción de la cabecera IP se usa muy, muy poco)

Routing basado en etiquetas

- Los paquetes llevan una etiqueta (un número) que determina la ruta que deben seguir.
- Cada router, en vez de consultar la IP de destino en la tabla de routing IP, consulta la etiqueta en una tabla que dictamina el siguiente salto.
 - Una búsqueda de un número en una tabla es muchísimo más rápida que buscar la ruta más específica (ruta que encaja con el prefijo más largo) en la tabla de routing IP.

Contenidos

- 1 Protocolos de Routing
- 2 Routing en Internet**
- 3 Routing en redes ad-hoc móviles
- 4 Referencias

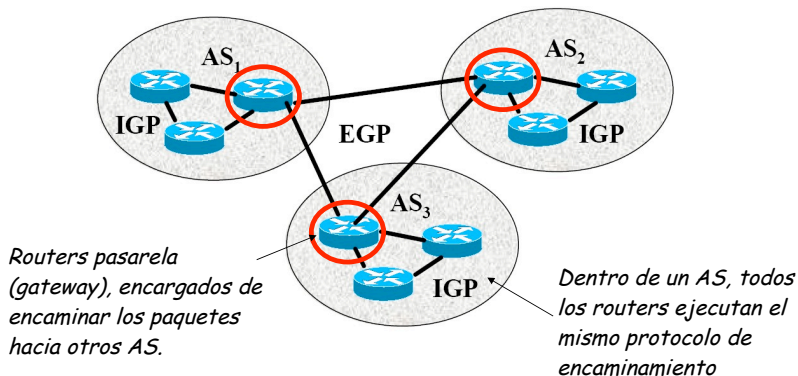
Protocolos de routing en Internet:

Protocolos Interiores y Exteriores

- A principios de los 80 Internet era una sola red desde el punto de vista administrativo. Las tablas mantenían entradas para todas las subredes. Problemas:
 - Escalabilidad
 - Autonomía administrativa
- En 1982 se decide agrupar subredes en **Sistemas Autónomos (AS)** y eliminar la centralización administrativa:
 - Cada AS ejecuta un **Protocolo Interior de Encaminamiento (IGP)**, Interior Gateway Protocol) para sus subredes:
 - RIP (Routing Information Protocol) (DVP)
 - IGRP (Interior Gateway Routing Protocol) (DVP)
 - OSPF (Open Shortest Path First) (LSP)
 - IS-IS (Intermediate System to Intermediate System) (LSP)
 - Algunos routers de cada AS ejecutan entre sí un **Protocolo Exterior de Encaminamiento (EGP)**, Exterior Gateway Protocols) para la interconexión de un AS con el resto:
 - BGP (Border Gateway Protocol) (estilo DVP)

AS, IGP y EGP

- Relación entre sistemas autónomos y protocolos interiores y exteriores de encaminamiento:



Contenidos

- 1 Protocolos de Routing
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles**
- 4 Referencias

Contenidos

- 1 Protocolos de Routing
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles**
 - **Routing entre robots**
 - Clasificación de los protocolos de routing en MANETs
 - Protocolos
- 4 Referencias

Escenarios

- **Robots fijos conectados por red cableada:**
 - Pueden usar rutas estáticas o los protocolos de routing (interiores) habituales en Internet: RIP, IGRP, OSPF, IS-IS.
- **Robots fijos o móviles conectados por red inalámbrica:**
 - **Comunicaciones centralizadas:** Los distintos robots se comunican con un elemento central (servidor o un robot central)
 - **Comunicaciones con infraestructura de puntos de acceso**
 - **Comunicaciones en red ad-hoc**

Robots con comunicaciones centralizadas

- Los robots se comunican con un servidor o con un robot central
- Situaciones:
 - Los mensajes que envía un robot van destinados siempre al elemento central, y los mensajes del elemento central van destinados a un robot
 - Los mensajes que envía un robot van destinados a otro robot, pero se decide orquestar el envío de mensajes a través del elemento central: un mensaje de un robot a otro se envía primero al elemento central
- Todos los robots tienen conectividad con el elemento central
- No se necesitan protocolos de routing

Robots con infraestructura de puntos de acceso

- Todos los robots tienen conectividad con algún *access point* y los *access point* entre sí por una red cableada o por un WDS.
- Los robots se configuran en la misma subred IP, como adyacentes
- Cada robot transmite a otro a través de uno (o más) AP.
- No se necesitan protocolos de routing.

Robots con red ad-hoc

- Los robots se comunican directamente de robot a robot
- No todos los robots están accesibles directamente entre sí: típicamente un robot puede alcanzar a algunos otros.
- Si los robots se mueven, la alcanzabilidad cambia
- Aunque los robots se configuren con direcciones de la misma subred IP, para comunicarse un robot A con otro B se necesitará con frecuencia utilizar otros robots intermedios como routers.
- Si los robots se mueven, no sirven rutas fijas: se necesitan protocolos de routing.
- Los protocolos que se utilizan son los utilizados en MANETs (Mobile Ad-hoc NETWORKs)

MANETs: Mobile Ad-hoc Networks

- VANETs: *Vehicular Ad-hoc Networks*
- SPANs: *Smartphone Ad-hoc Networks*
- iMANETs: *Internet based MANETs*
- *Wireless Mesh Networks*
- *Wireless Sensor Networks*
- *Ad-hoc Robots Networks*
- ...

Contenidos

- 1 Protocolos de Routing
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
 - Routing entre robots
 - Clasificación de los protocolos de routing en MANETs
 - Protocolos
- 4 Referencias

Clasificación de los protocolos de routing en MANETs

- **Proactivos:** Los nodos tienen una tabla con rutas a todos los destinos.
- **Reactivos:** Cada nodo sólo calcula una ruta a un destino cuando la necesita para enviar un mensaje.
- **Híbridos:** Dividen la red en zonas, para los nodos más cercanos construyen rutas de forma proactiva, para los nodos más alejados de forma reactiva.
- **Geográficos:** Basados en las coordenadas geográficas de los nodos, que son radiadas periódicamente por ellos.

Protocolos Proactivos

- También llamados *table driven*, **guiados por tablas**: los nodos mantienen constantemente una tabla de rutas a todos los destinos.
- Los nodos intercambian información para mantener su tabla siempre actualizada.
- Cuando se va a enviar un mensaje a un destino, simplemente hay que consultar la tabla, porque tendrá una ruta actualizada.
- Cada nodo debe almacenar una cantidad significativa de información.
- Los nodos envían periódicamente mensajes para mantener la tabla actualizada. . . incluso aunque la topología no cambie
- Ejemplos:
 - DSDV: Destination Sequenced Distance Vector
 - OLSR: Optimized Link State Routing
 - B.A.T.M.A.N.: Better Approach To Mobile Ad-hoc Networking
 - BABEL

Protocolos Reactivos

- También llamados *on-demand driven*, guiados por demanda.
- No hay tablas a priori: cuando un nodo quiere enviar un mensaje a un destino construye en ese momento una ruta para él.
- Cuando se necesita una ruta, se envían mensajes (normalmente por inundación) para descubrirla.
- No se necesita almacenar demasiada información en cada nodo.
- No se genera demasiado tráfico para construir rutas, sólo bajo demanda y para las rutas nuevas que se necesitan.
- Aparece una latencia significativa antes de enviar el primer mensaje por una nueva ruta.
- Ejemplos:
 - DSR: Dynamic Source Routing
 - AODV: Ad-hoc On-demand Distance Vector
 - TORA: Temporally Ordered Routing Algorithm

Protocolos Híbridos

- Usan los dos enfoques, proactivo y reactivo, pero tratando de evitar los inconvenientes de ambos.
- Son jerárquicos: dividen la red en zonas
- Cada nodo aplica:
 - mecanismos proactivos para tener una tabla con rutas a los nodos que hay en ese momento dentro de su zona
 - mecanismos reactivos para comunicarse con nodos de fuera de su zona: cada vez que lo necesita, construye una ruta.
- Ejemplos:
 - ZRP: Zone Routing Protocol
 - OORP: Order One Routing Protocol

Protocolos Geográficos

- Los nodos radian sus coordenadas geográficas
- Se calculan las rutas teniendo en cuenta esas coordenadas geográficas
- Se aplican en entornos en que los nodos pueden obtener sus coordenadas, por ejemplo, por disponer de GPS.
- Ejemplos:
 - GPSR: Greedy Perimeter Stateless Routing
 - LAR: Location Aided Routing
 - GSR: Geography Source Routing

Contenidos

- 1 Protocolos de Routing
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles**
 - Routing entre robots
 - Clasificación de los protocolos de routing en MANETs
 - **Protocolos**
- 4 Referencias

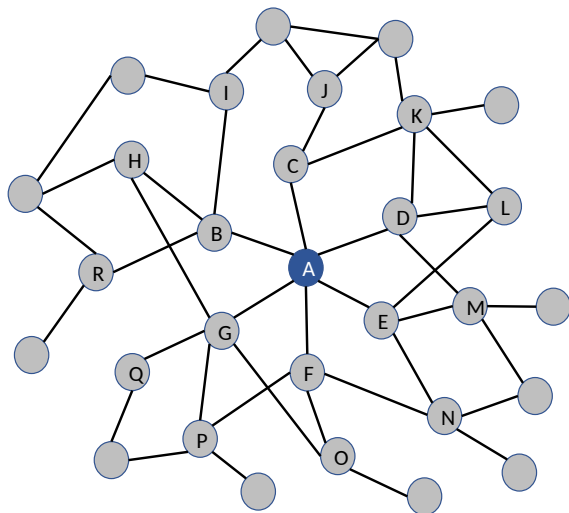
DSDV (Destination Sequenced Distance Vector)

- Protocolo **proactivo** tipo **vector de distancias**
- Cada entrada en la tabla contiene:
 - Destino
 - Siguiendo salto
 - Número de saltos (coste)
 - Número de secuencia
- Cada nodo radia a sus vecinos:
 - toda la tabla (enviada de vez en cuando)
 - los cambios en la tabla (envíos frecuentes)
- Se cambia una entrada cuando se recibe de un vecino una ruta “mejor”: con mayor número de secuencia (aunque el coste aumente) o con menor número de saltos (si mantiene el mismo número de secuencia).
- El uso de los números de secuencia permite evitar bucles y la cuenta al infinito.

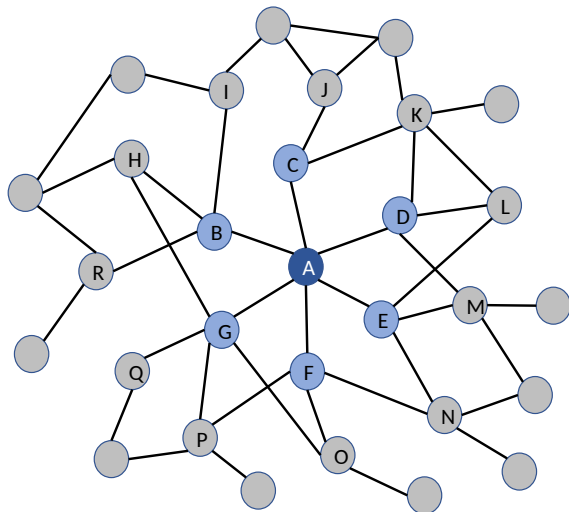
OLSR: Optimized Link State Protocol (I)

- Protocolo **proactivo** tipo **estado de enlace**.
- Trata de reducir el número de mensajes que se envían por inundación en los protocolos “clásicos” (OSPF, IS-IS).
- Para un nodo, se definen:
 - **vecinos a 1 salto**: los vecinos que oyen al nodo
 - **vecinos a 2 saltos**: los vecinos que oyen a los vecinos a 1 salto del nodo.
- **Mensajes HELLO**: Enviados periódicamente por un nodo a sus vecinos a 1 salto. Contienen la lista de vecinos a 1 salto y a 2 saltos del nodo que los envía.
- Con los mensajes HELLO recibidos, cada nodo va conociendo a sus vecinos a 2 saltos.
- Cada nodo elige un subconjunto de sus vecinos a 1 salto: mínimo subconjunto que da acceso a todos los vecinos a 2 saltos: son los **MPR** (Multi Point Relays) del nodo.

OLSR: Elección de MPR (I)

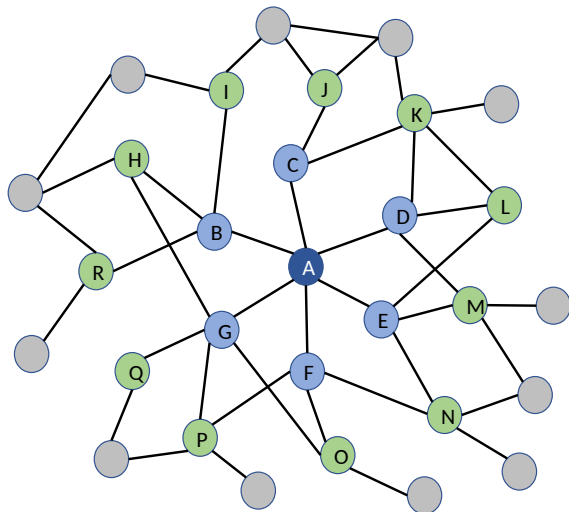


OLSR: Elección de MPR (II)



A: vecinos a 1 salto: B, C, D, E, F, G

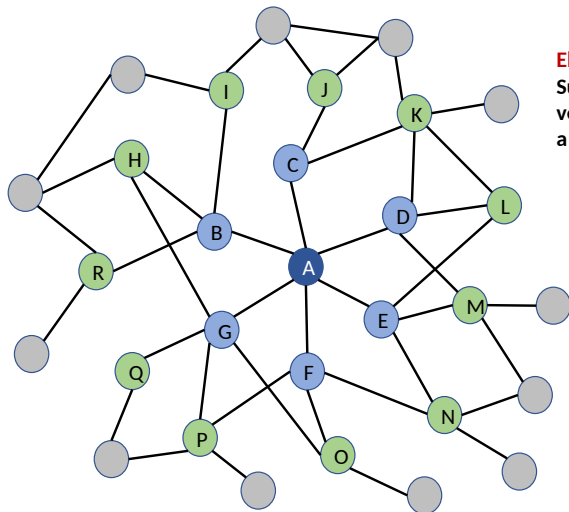
OLSR: Elección de MPR (III)



A: vecinos a 1 salto: B, C, D, E, F, G

A: vecinos a 2 saltos: H, I, J, K, L, M, N, O, P, Q, R

OLSR: Elección de MPR (IV)



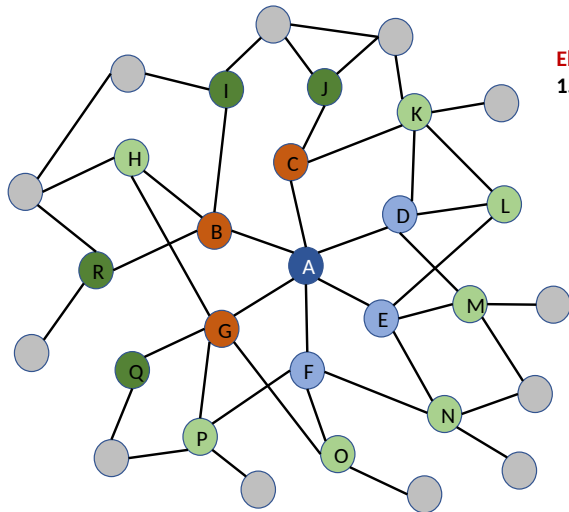
Elección de MPRs de A:

Subconjunto mínimo de sus vecinos-a-1-salto que dan acceso a todos sus vecinos-a-2-saltos

A: vecinos a 1 salto: B, C, D, E, F, G

A: vecinos a 2 saltos: H, I, J, K, L, M, N, O, P, Q, R

OLSR: Elección de MPR (V)



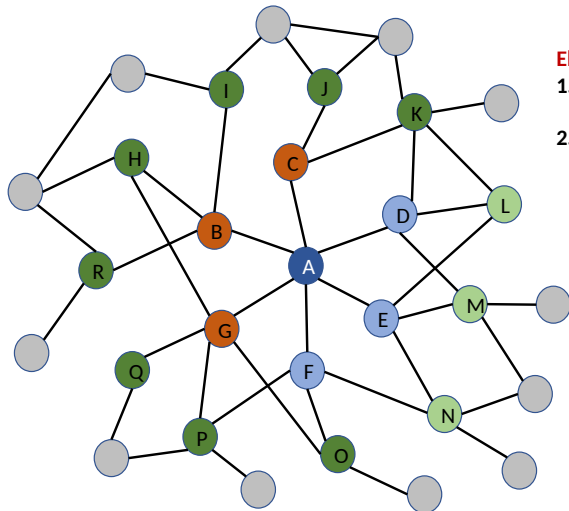
Elección de MPRs de A:

1. se elige los que dan acceso a lo vecinos-a-2-saltos aislados

A: vecinos a 1 salto: B, C, D, E, F, G

A: vecinos a 2 saltos: H, I, J, K, L, M, N, O, P, Q, R

OLSR: Elección de MPR (VI)



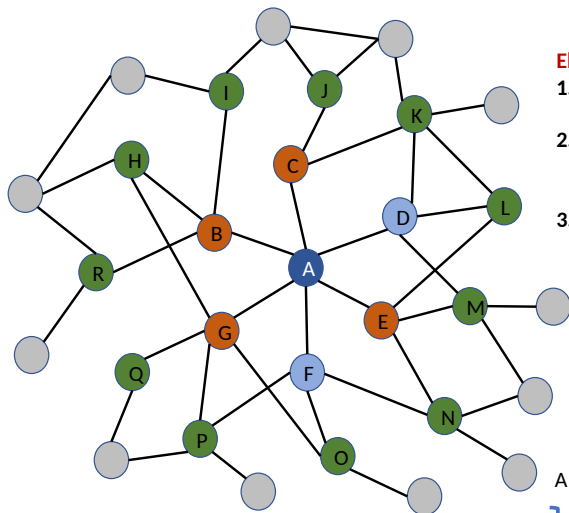
Elección de MPRs de A:

1. se elige los que dan acceso a lo vecinos-a-2-saltos aislados
2. se ve qué vecinos-a-2-saltos quedan ya cubiertos y cuáles faltan por cubrir

A: vecinos a 1 salto: B, C, D, E, F, G

A: vecinos a 2 saltos: H, I, J, K, L, M, N, O, P, Q, R

OLSR: Elección de MPR (VII)



Elección de MPRs de A:

1. se elige los que dan acceso a los vecinos-a-2-saltos aislados
2. se ve qué vecinos-a-2-saltos quedan ya cubiertos y cuáles faltan por cubrir
3. se elige al mínimo número de vecinos a 1-salto que cubre a los vecinos-a-2-saltos que faltan

A recibirá los HELLO de B, C, D, E, F, G

A: vecinos a 1 salto: B, C, D, E, F, G. **MPRs:** B, C, E, G

A: vecinos a 2 saltos: H, I, J, K, L, M, N, O, P, Q, R

Mensaje HELLO
enviado por A

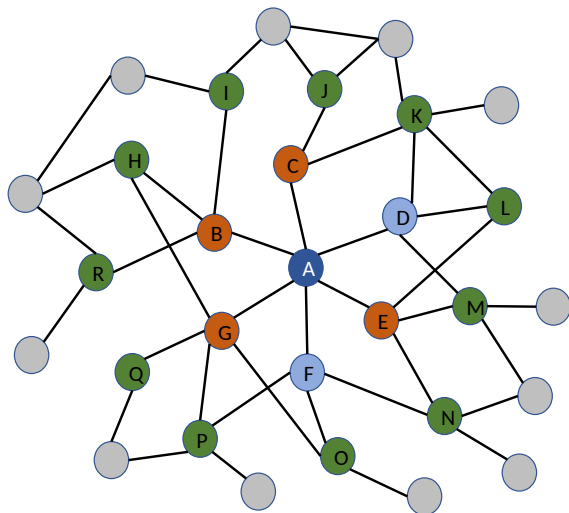
OLSR: Optimized Link State Protocol (II)

- **Mensajes TC (Topology Control):** Cada nodo envía mensajes TC que indican cuáles de sus vecinos le tienen seleccionado a él como MPR, y por lo tanto son el último salto en la ruta hacia él.
- Los mensajes TC se envían por inundación sólo a través de nodos que sean MPRs
- Con los mensajes TC y los HELLO de los vecinos los nodos conocen lo suficiente de la topología de la red y pueden calcular rutas a cualquier destino aplicando una variante de Dijkstra:
 - A quiere calcular la ruta hacia B:
 - En el TC enviado por B estará la lista de los posibles últimos saltos en la ruta hacia B, por ejemplo, X1 y X2.
 - En los TCs enviados por X1 y X2 estarán los posibles penúltimos saltos
 - Se repite recursivamente la búsqueda hasta llegar a un XN que sea vecino a 1 salto (y MPR) de A. Cuanto antes se llegue, más corta será la ruta

OLSR: Optimized Link State Protocol (III)

- Los MPRs tienen por tanto 2 roles:
 - Son los únicos que reenvían los mensajes TC
 - Son los nodos a través de los que se construyen las rutas

OLSR: Mensajes TC



A: vecinos a 1 salto: B, C, D, E, F, G. **MPRs: B, C, E, G**

A: vecinos a 2 saltos: H, I, J, K, L, M, N, O, P, Q, R

Mensaje TC de A:

- **Originador:** A
- **Selectores MPR:** Qué vecinos-a-1-salto han elegido a A como MPR (ojo: no necesariamente son B, C, E, G)
- **Número de secuencia**

Envío de mensajes TC:

- A envía su TC a sus MPRs: B, C, E, G
- B, C, E, G reenvían el TC de A a sus MPRs
- y así sucesivamente

OLSR: Elección de ruta

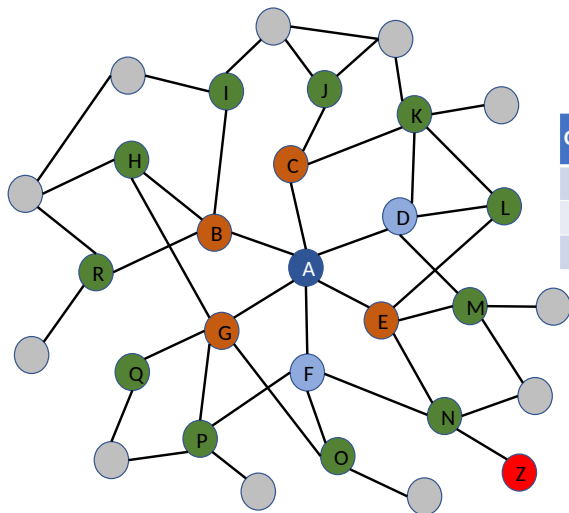


Tabla de Información de MPR

Creador TC	Selectores MPR	Núm Secuencia
Z	N	1
N	E	1
E	A	1

RUTA A → Z:

- Z ← N
 - N ← E
 - E ← A
- } A → E → N → Z

A: vecinos a 1 salto: B, C, D, E, F, G. **MPRs:** B, C, E, G

A: vecinos a 2 saltos: H, I, J, K, L, M, N, O, P, Q, R

B.A.T.M.A.N. (I)

- Protocolo **proactivo** y (en parte) tipo **vector de distancias**.
- Diseñado específicamente como respuesta a la complejidad de OLSR:
 - genera menos tráfico
 - necesita menos tiempo de cómputo en los nodos
 - necesita menos información almacenada en los nodos
- Cada nodo construye **Mensajes OGM** (Originator Messages) para informar al resto de su existencia, enviándolos por inundación. Los mensajes contienen:
 - Originador (creador)
 - Nodo que lo reenvía por inundación
 - TTL (decreciente en cada reenvío por inundación)
 - Número de secuencia

B.A.T.M.A.N. (II)

- Cada nodo que recibe de varios vecinos un mismo OGM, elige al vecino que considera “mejor” como su siguiente salto en la ruta hacia el originador de ese OGM.
 - El “mejor” es aquel vecino de quién recibió más veces ese mismo OGM recientemente (sólo considera los últimos números de secuencia recibidos).
- Un nodo sólo reenvía un OGM si lo recibe del que es su siguiente salto para el originador de ese OGM.
- Dos versiones del protocolo:
 - **batmand**: Funciona sobre UDP, ejecuta en espacio de usuario
 - **batman-adv**: Funciona sobre el nivel de enlace (wifi)

DSR: Dynamic Source Routing (I)

- Protocolo **reactivo** y que utiliza **encaminamiento en origen**.
- Dos partes:
 - **Descubrimiento de rutas**: se envían los mensajes:
 - **RREQ**: Route Request
 - **RREP**: Route Reply
 - **Mantenimiento de rutas**: se envía el mensaje:
 - **RERR**: Route Error

DSR: Dynamic Source Routing (II)

- **RREQ**: Contiene:
 - Identificador único
 - Origen
 - Destino
 - Nodos intermedios, inicialmente vacío
- Cuando el nodo A quiere enviar un mensaje a B, si no conoce su ruta, envía un RREQ por inundación.
- Cada nodo X que reciba el RREQ, si ya lo ha visto lo descarta. Si es la primera vez que lo ve:
 - Se añade a la lista de nodos intermedios.
 - Si no conoce la ruta desde él hasta B, reenvía el RREQ.
 - Si conoce la ruta desde X hasta B, envía un RREP con la ruta completa: (A) + (nodos intermedios del mensaje recibido) + (ruta de X a B). El RREP se envía por la ruta inversa a la contenida como respuesta.

DSR: Dynamic Source Routing (III)

- Si el RREQ llega a B (destino), B compone un RREP con la ruta, que resulta ser la lista de todos los nodos intermedios que lleva el mensaje. El RREP se envía por la ruta inversa a la contenida como respuesta.
- Cuando A reciba un RREP, apunta la ruta en una caché de rutas y ya puede enviar el primer paquete de A a B. El paquete contiene la ruta completa y se encamina por **source routing**.
 - un nodo puede almacenar varias rutas alternativas a un mismo destino

DSR: Dynamic Source Routing (IV)

- **Mantenimiento de rutas:**

- Cuando se envían paquetes por una determinada ruta, cada nodo es responsable de comprobar que su siguiente salto sigue activo.
- Si en el camino de A a B un nodo intermedio X detectara que el siguiente salto Y ya no está disponible, envía un mensaje RERR a A indicando que $X \rightarrow Y$ ya no es un tramo válido.
 - A borrará esa ruta de su caché de rutas hacia D.
 - A borrará también cualquier otra ruta de su caché que incluya ese tramo.
- Existen mecanismos adicionales para detección y borrado de rutas obsoletas o inválidas.

Contenidos

- 1 Protocolos de Routing
- 2 Routing en Internet
- 3 Routing en redes ad-hoc móviles
- 4 Referencias**

Referencias

- Andrew S. Tanenbaum, **Redes de Computadores**, Prentice Hall, 4ª edición: apartado 5.2.
- J.J. Kurose y K.W. Ross, **Redes de Computadores: un enfoque descendente basado en Internet**, Pearson Educación, 2ª edición: apartados 4.2, 4.3.
- RFC 3626, **Optimized Link State Routing Protocol (OLSR)**: <http://www.faqs.org/rfcs/rfc3626.html>
- Internet Draft, **Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)**:
<https://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00>
- RFC 4728, **The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4**:
<http://www.faqs.org/rfcs/rfc4728.html>