Enrutamiento

Problemas de Distance-Vector...

Area de Ingeniería Telemática
http://www.tlm.unavarra.es

Arquitectura de Redes, Sistemas y Servicios
Grado en Ingeniería en Tecnologías de Telecomunicación, 2º

Temario

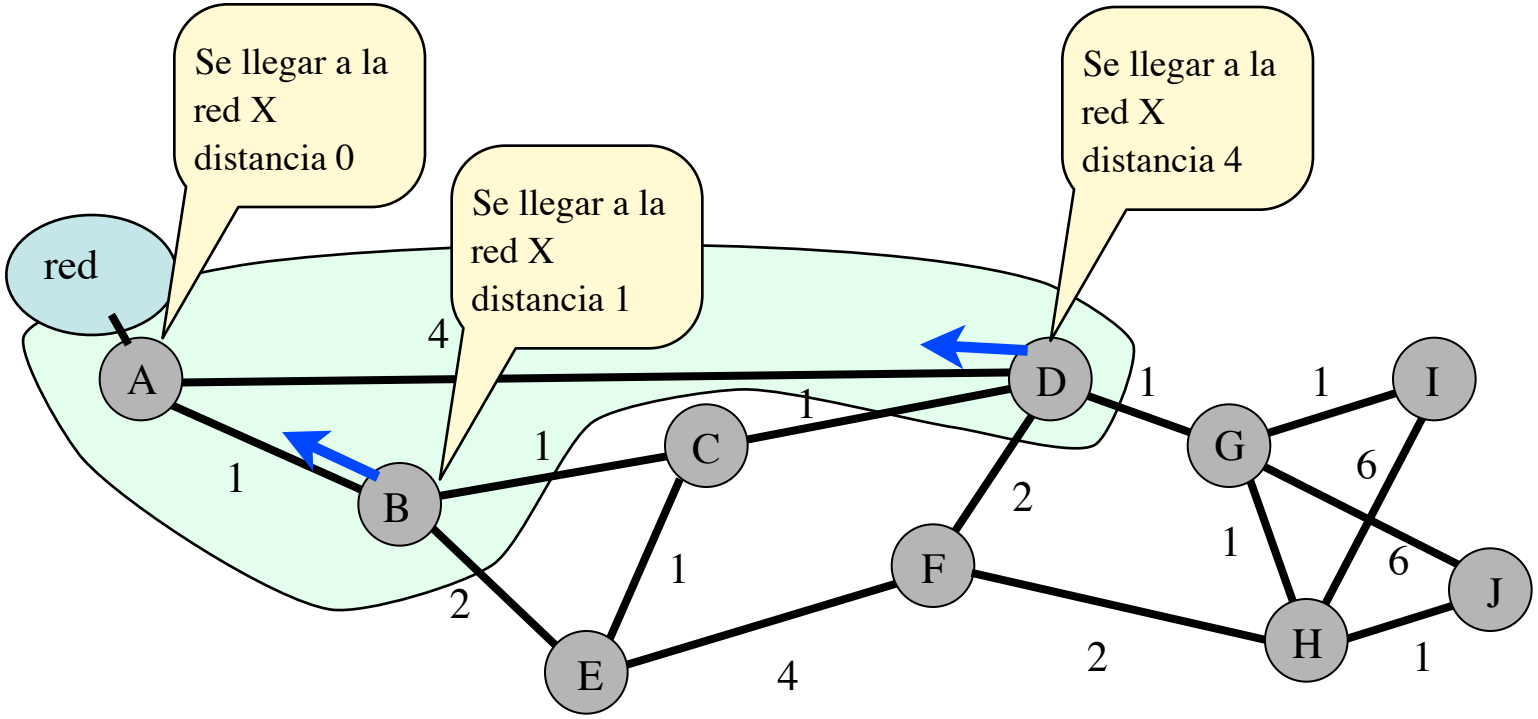
1. Introducción
2. Arquitecturas de conmutación y protocolos
3. Introducción a las tecnologías de red
4. Control de acceso al medio
5. Conmutación de circuitos
6. Transporte fiable
7. Encaminamiento

Temario

1. Introducción
2. Arquitecturas de conmutación y protocolos
3. Introducción a las tecnologías de red
4. Control de acceso al medio
5. Conmutación de circuitos
6. Transporte fiable
7. **Encaminamiento**

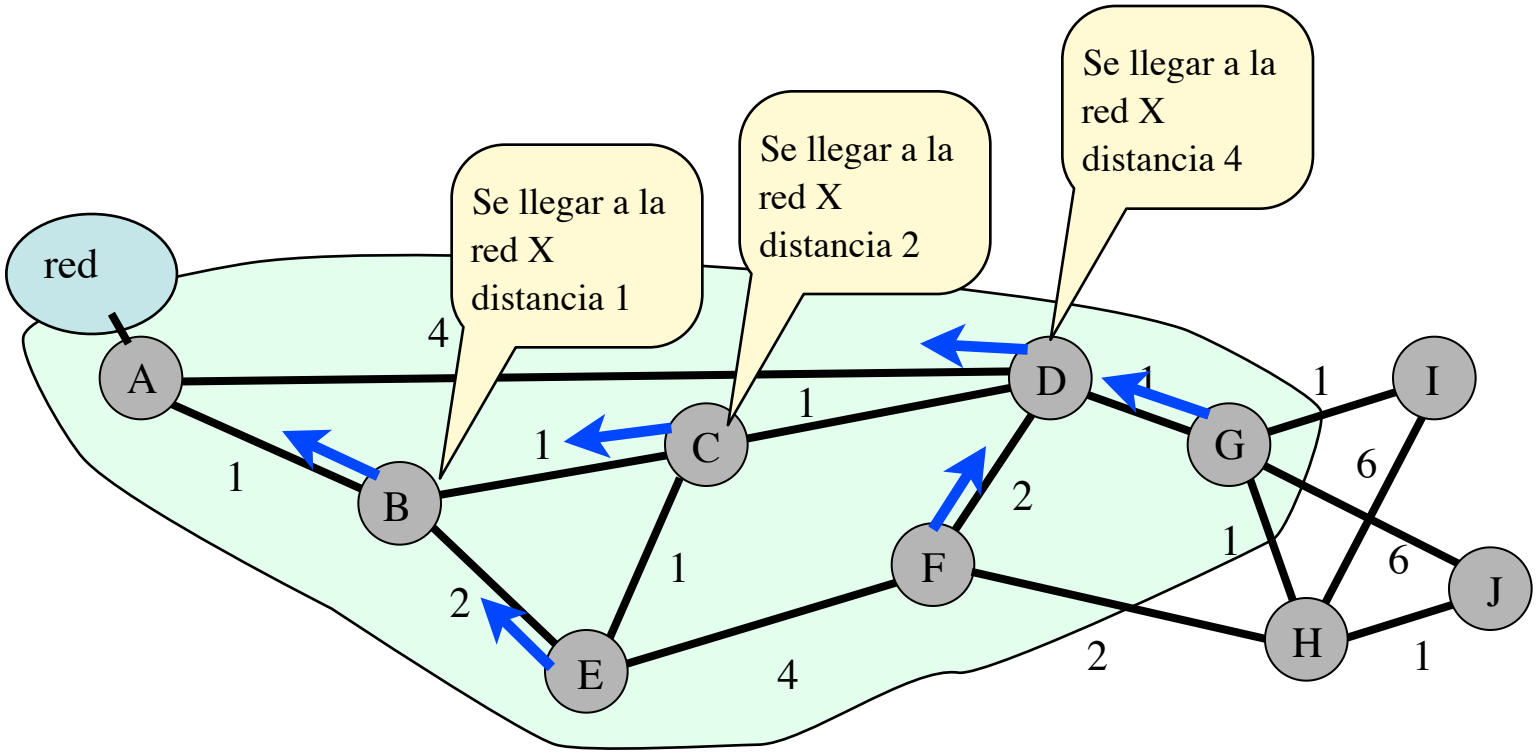
Funciona Distance-Vector?

- La información para cada destino se propaga desde los routers que la saben...
-



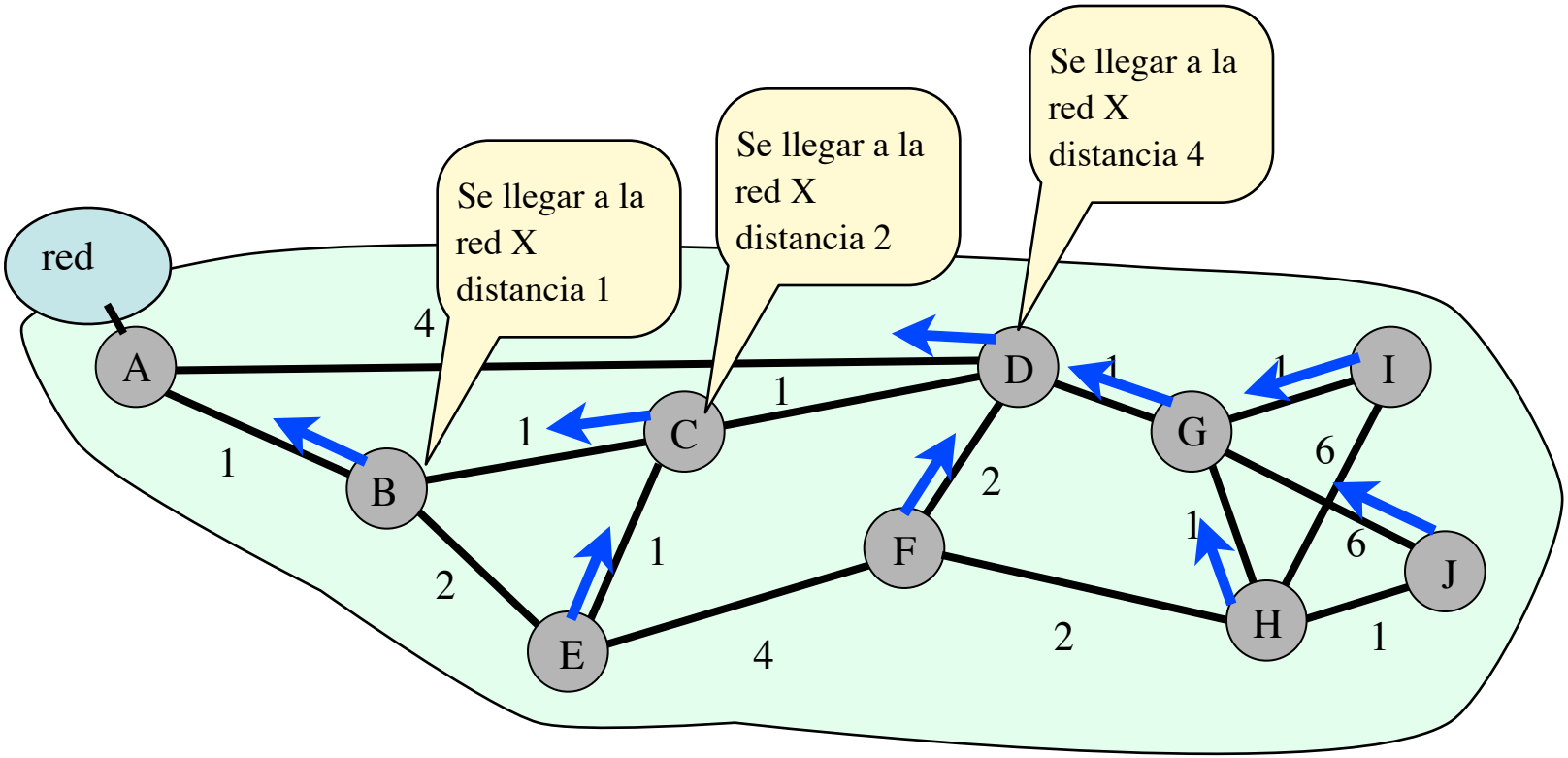
Funciona Distance-Vector?

- La información para cada destino se propaga desde los routers que la saben...
-



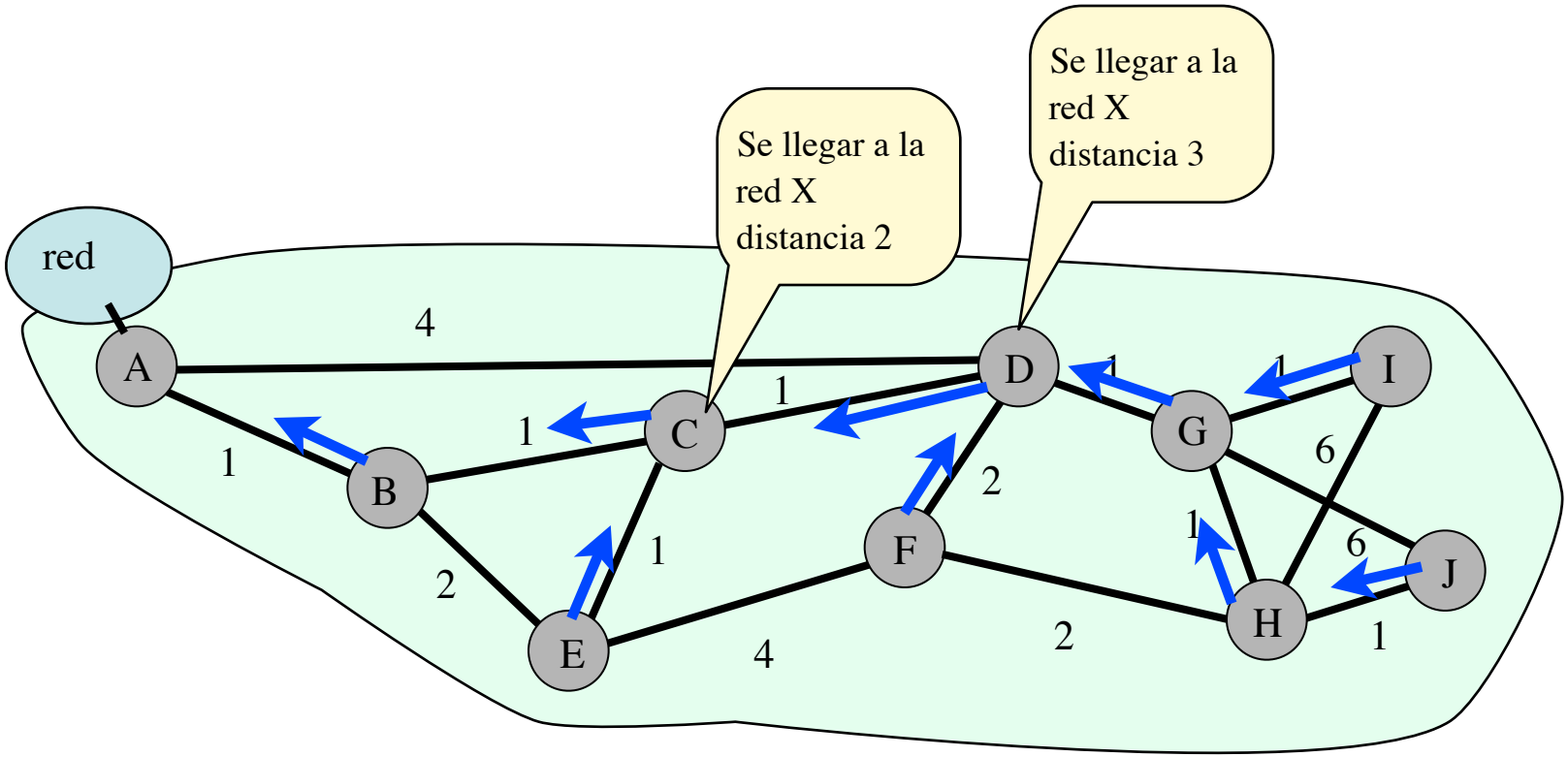
Funciona Distance-Vector?

- La información para cada destino se propaga desde los routers que la saben...
- El tiempo de propagación depende de cuantos routers hay hasta el destino
- El tiempo de propagación no es tanto si cada router envía la información a sus vecinos cada vez que hay cambios (triggered updates)



Funciona Distance-Vector?

- La información para cada destino se propaga desde los routers que la saben...
- No necesariamente la mejor ruta es la que oigo la primera vez por eso el algoritmo debe funcionar de forma continua
- Lo mismo pasa a la vez con todos los demas destinos



Funciona Distance-Vector?

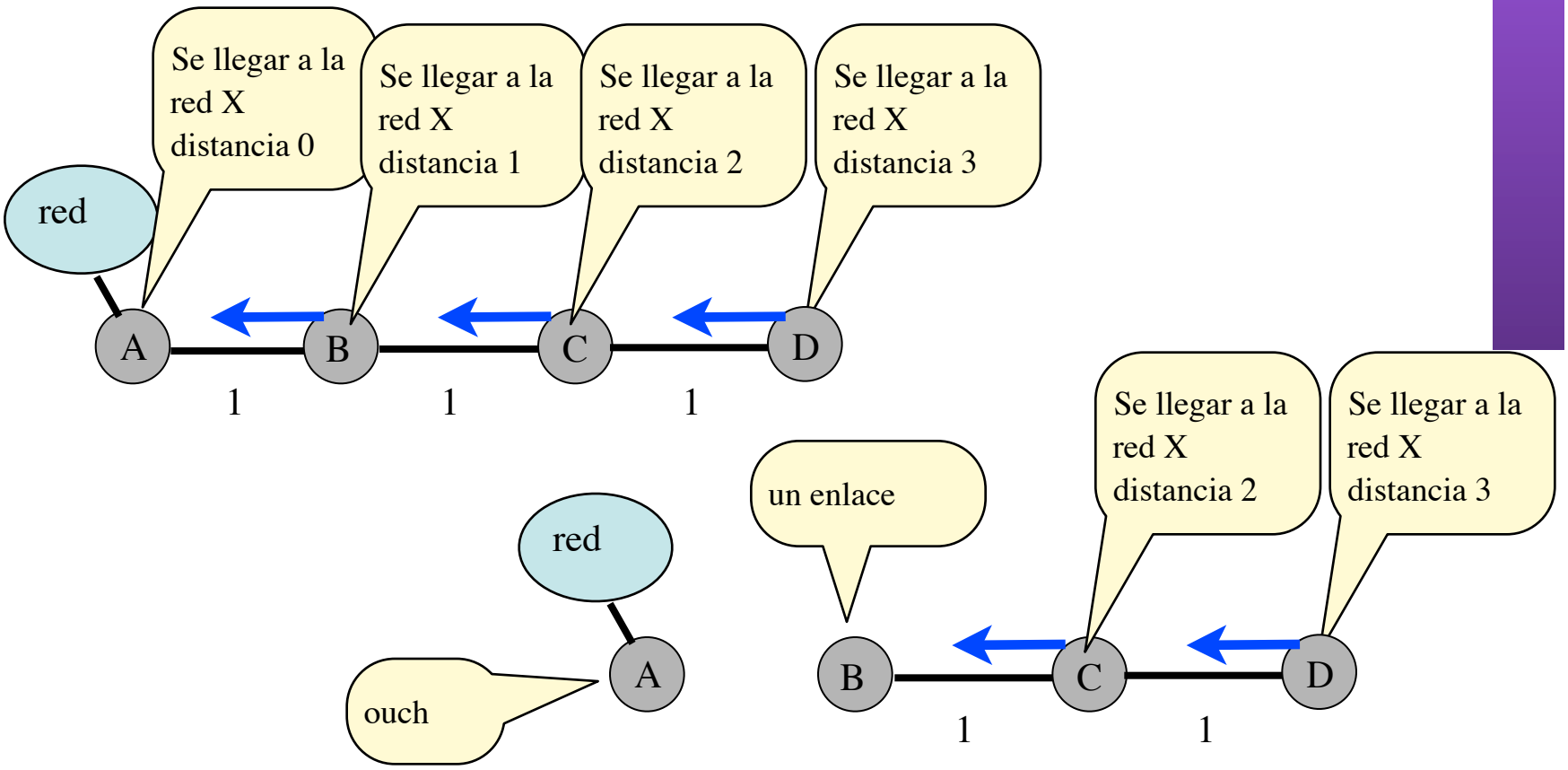
- El algoritmo no sabe cuando ha terminado
- Pero no importa mucho que se ejecute de forma continua
- Si usamos envío de información periódica controlamos el trafico de enrutamiento que se envía... pero el tiempo en propagar rutas es mas largo
- Si enviamos en cuanto hay cambios (triggered updates) la propagación es rapida y se envía más tráfico cuando hay un cambio pero es self-stoping se autocontrola y deja de enviar cuando las rutas se estabilizan

- Normalmente se utiliza triggered updates con y envio periodico no demasiado frecuente
  - Envío rapido de cambios
  - El envio periodico ayuda si se pierden mensajes o para descubrir vecinos cuando un router se conecta a la red

- Parece razonable. Funciona, se propaga rapido y no crea mucho trafico...
- Y entonces por que es el sistama de **enrutamiento *antiguo* de Internet**

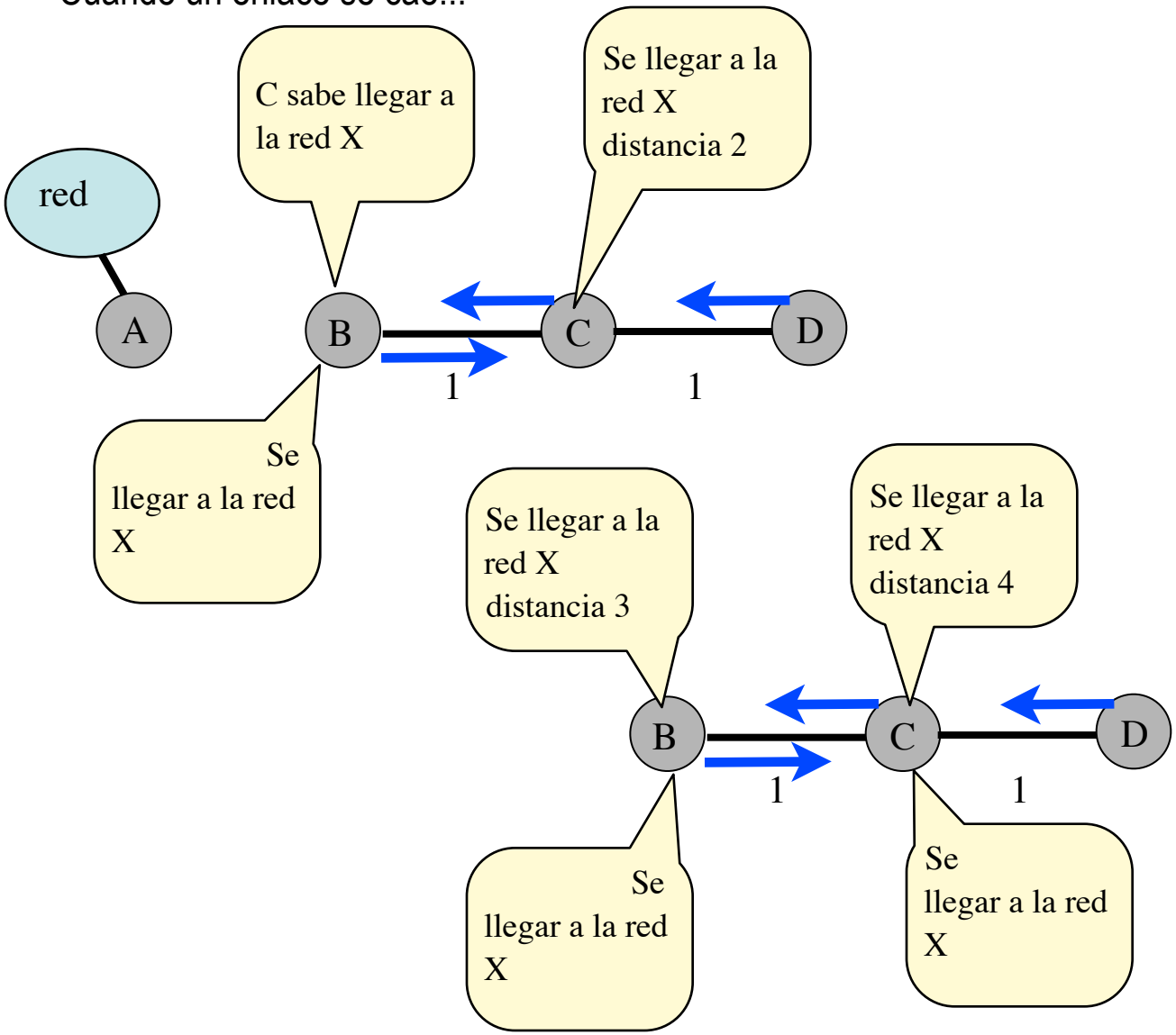
Problemas

- Es lento en reaccionar !!!
- No todos los cambios se propagan rápido hay situaciones anómalas
- Un caso muy simple y bien conocido
- Qué pasa si se cae el primer enlace?



La cuenta a infinito

- Cuando un enlace se cae...



La cuenta a infinito

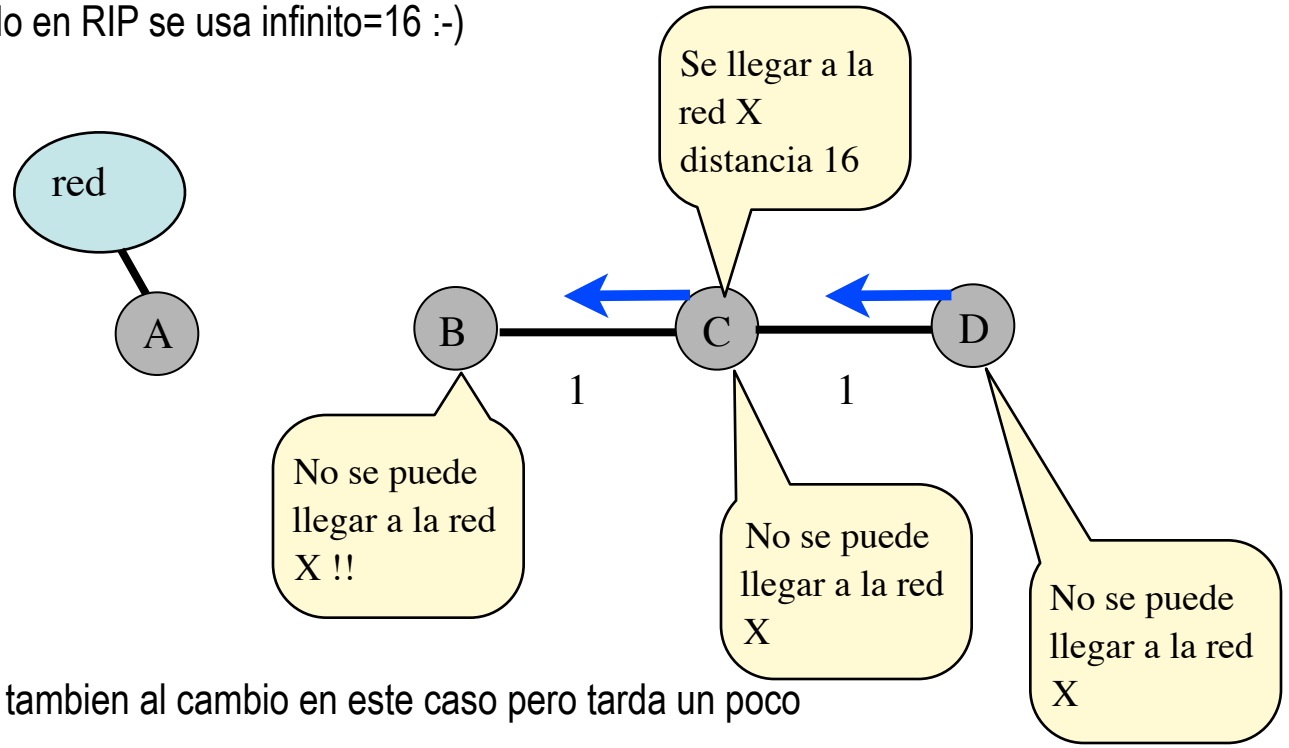
- Como acaba esto?
- Se hace que el campo para indicar la distancia tendrá bits limitados

  Cuando llegue al máximo se considera infinito y la ruta se descarta

  Establcer un valor de infinito pequeño hace que estos casos se detecten antes

  Pero entonces solo podremos calcular distancias menores a ese valor

  Por ejemplo en RIP se usa infinito=16 :-)



- Se adapta tambien al cambio en este caso pero tarda un poco

  y genera unos cuantos mensajes de actualización en el proceso

Problemas distance-vector

- Las cuentas a infinito hacen que los protocolos distance-vector puedan tardar en bastante en converger a la solución
- Mientras convergen las rutas puede no ser buenas

  (ciclos de enrutamiento y perdidas)

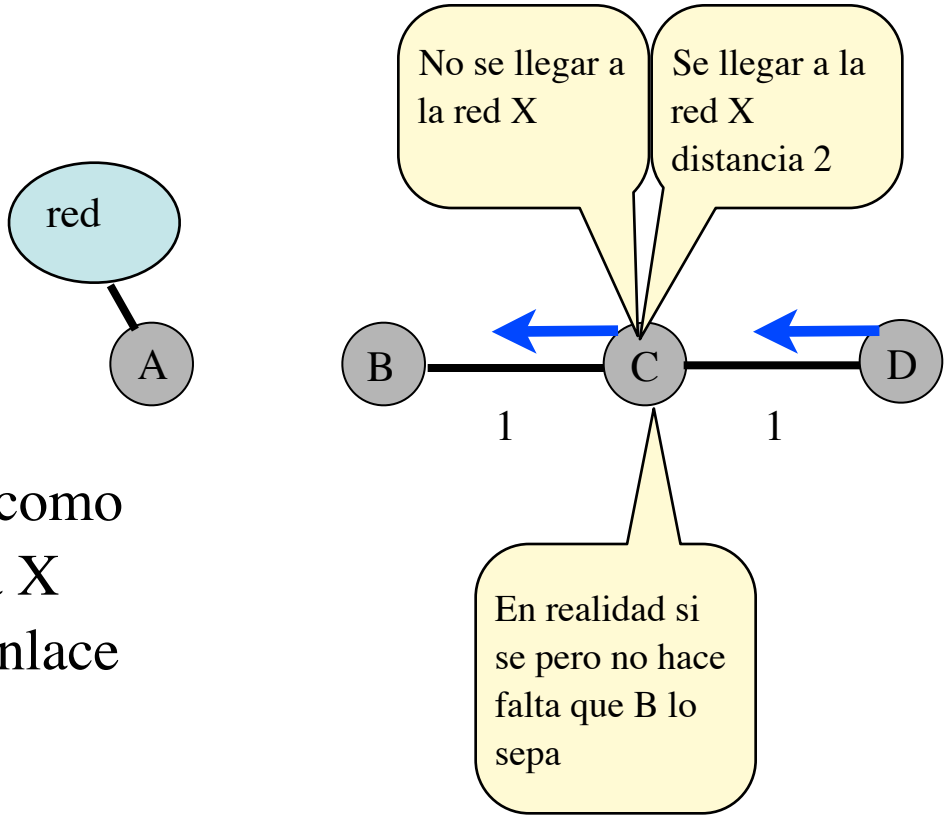
- Se puede resolver el problema de las cuentas a infinito?

  Hay algunas optimizaciónes que parecen obvias...
  - Mejor no anunciar una ruta a un nodo si la ruta pasa por el (split-horizon)
  - Cuando una ruta se vuelve es descartada no aceptar nuevas (hold-down)
  - ...

Soluciones: split horizon

- No anuncio las rutas a un destino a mi siguiente salto para ese destino

  El deberia saber como llegar porque yo le estoy mandando a el los paquetes que van a ese destino
- Esto resuelve el escenario anterior

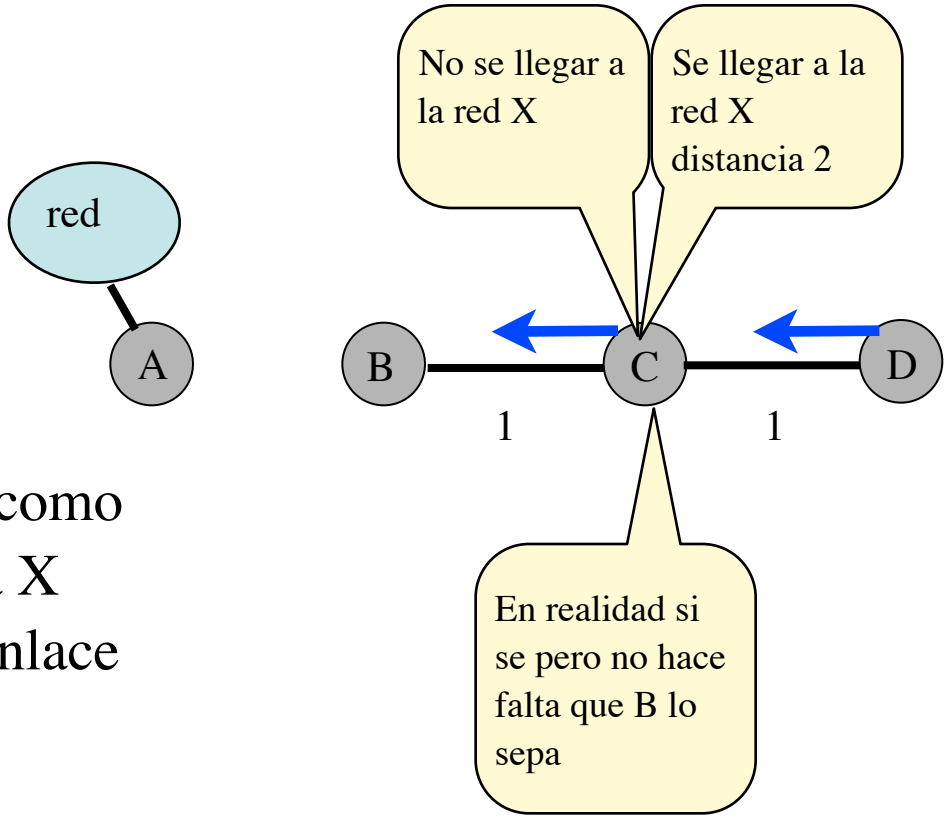


B ya no elegira a C como siguiente salto hacia X cuando se caiga el enlace

Soluciones: split horizon

- No anuncio las rutas a un destino a mi siguiente salto para ese destino

  El deberia saber como llegar porque yo le estoy mandando a el los paquetes que van a ese destino
- Esto resuelve el escenario anterior



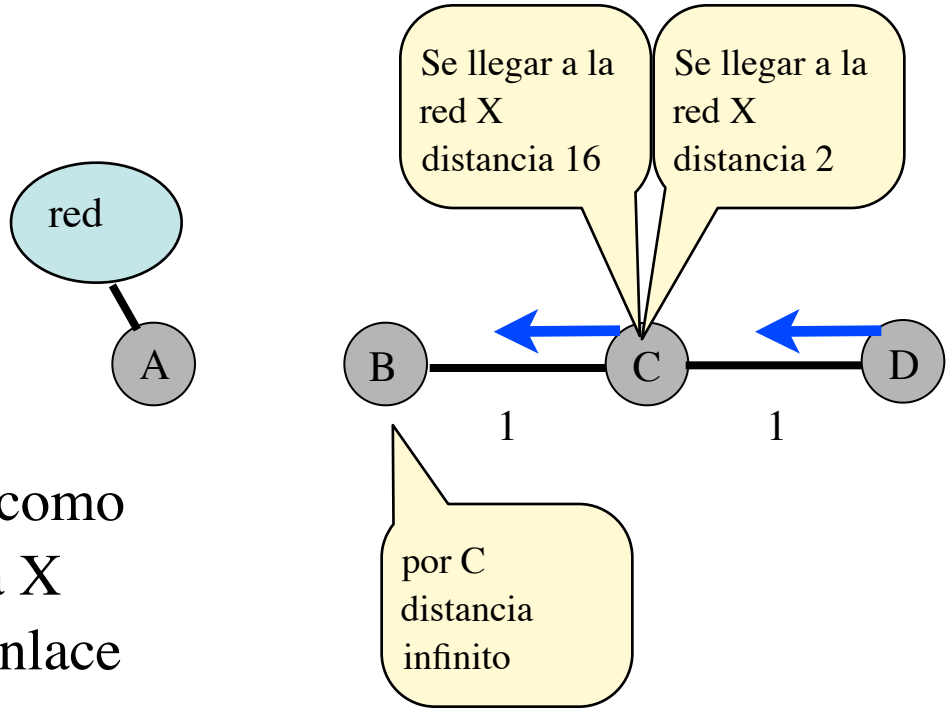
B ya no elegira a C como siguiente salto hacia X cuando se caiga el enlace

Soluciones: poison reverse

- Al siguiente salto para un destino le miento y le digo que la distancia es infinito

  Parecido a Split-horizon

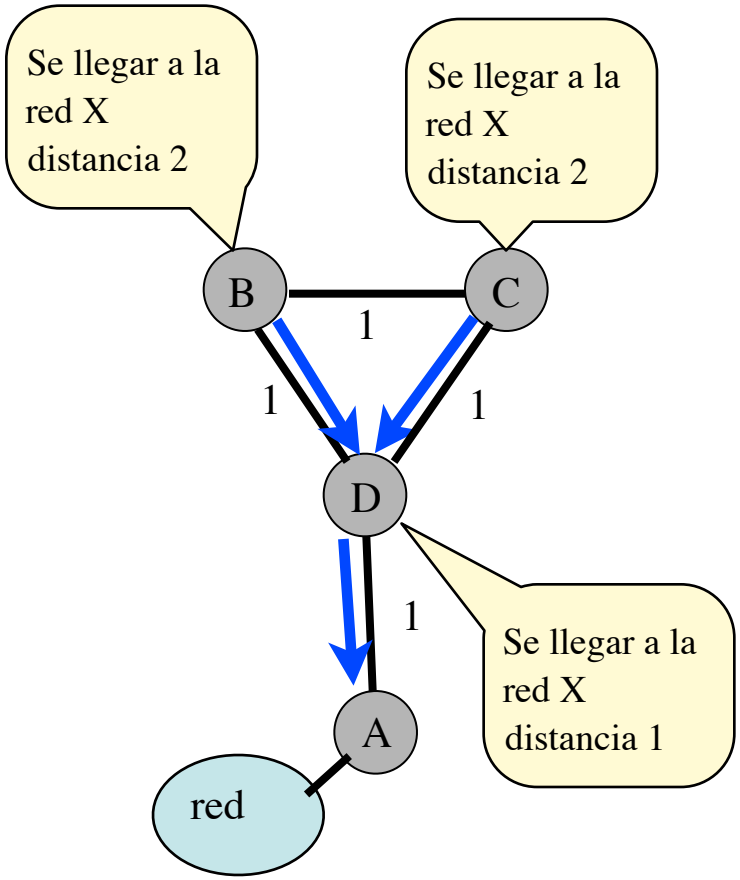
  Un poco mejor porque si por alguna otra cosa habia un ciclo de enrutamiento lo rompe
- Tambien resuelve el escenario anterior

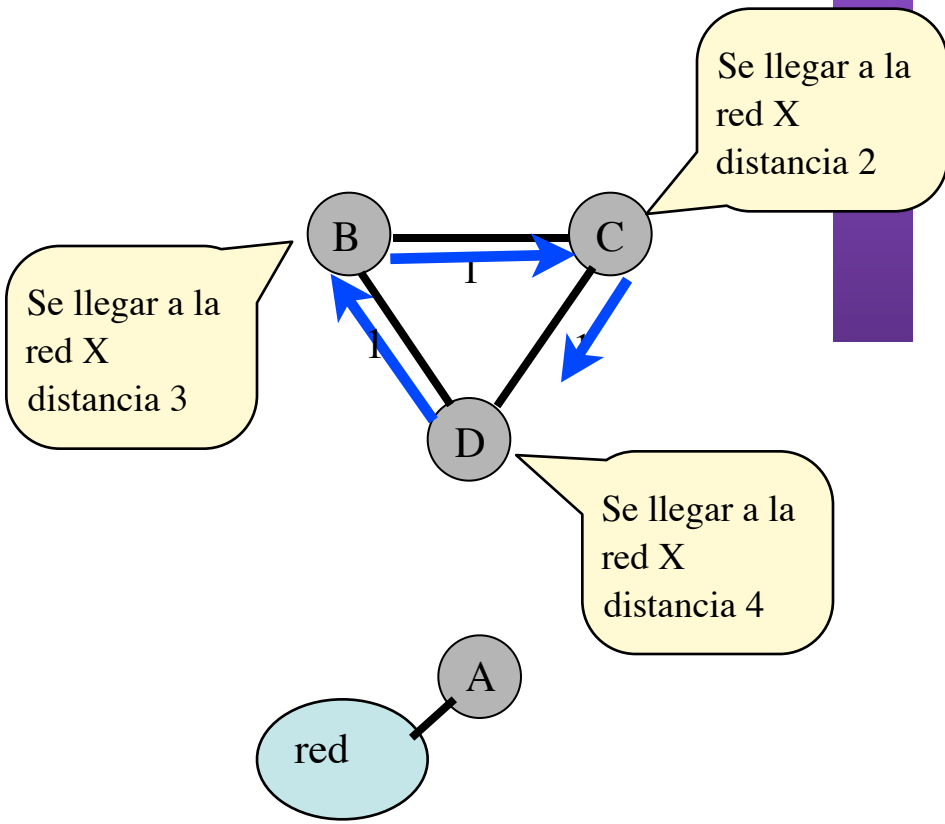


B ya no elegira a C como siguiente salto hacia X cuando se caiga el enlace

Split-horizon/poison-reverse

- Fáciles de implementar y resuelven el escenario anterior
- Tienen problemas en redes de area local cuando envio anuncios a broadcast como enveneno las rutas a mi siguiente salto pero no a los demas
- Y siguen pudiendo ocurrir cuentas a infinito de otros tipos





Más soluciones

- Hold-down: si una ruta se hace invalida no aceptar nuevos caminos a ese destino en un tiempo de hold-down

  Confia en que en ese tiempo la no alcanzabilidad se extienda a toda la red. si no da tiempo y algún nodo tiene aun ruta al destino puede crearse una cuenta a infinito
- Anunciar la distancia y el siguiente salto ~parecido a poison reverse
- Anunciar la distancia y el camino entero

  Este funciona bien pero requiere que los routers almacenen todo el camino para cada destino. No escala bien
- Usar dos métricas y mantener dos distancias

  Una para comparar caminos y la otra con el numero de saltos para las cuentas a infinito
- DUAL (Diffusing Update Algorithm)

  razonamientos con las distancias para decidir si es posible o no que el camino anunciado pase por ti

  i.e. si la distancia que anuncia C es menor que la que yo tenia al destino antes de hacerse invalida la ruta es seguro cambiar a C

Resumen hasta ahora

- Algoritmo distance-vector teorico
- Calculo de rutas distribuidas
- Adaptación a los cambios
- Convergencia rápida a los cambios y auto-parada en algunos casos
  - Normalmente los cambios a mejor se propagan rápido
- Problemas de convergencia/estabilidad y cuentas a infinito en otros casos
  - Normalmente los cambios a peor se propagan despacio
- Soluciones que alivian estos problemas pero no los resuelven totalmente

- Lo suficiente para que los algoritmos distance-vector sean útiles
- Qué protocolos reales distance-vector se utilizan? (RIP, IGRP, EIGRP...)
- Como conseguir algoritmos con menos problemas de convergencia?

Conclusiones

- Algoritmos Distance-vector
  - Simples
  - Se adaptan rápido a los cambios a mejor
  - Tienen problemas de convergencia en los cambios a peor

- Otros algoritmos?