

Movilidad...

802.11

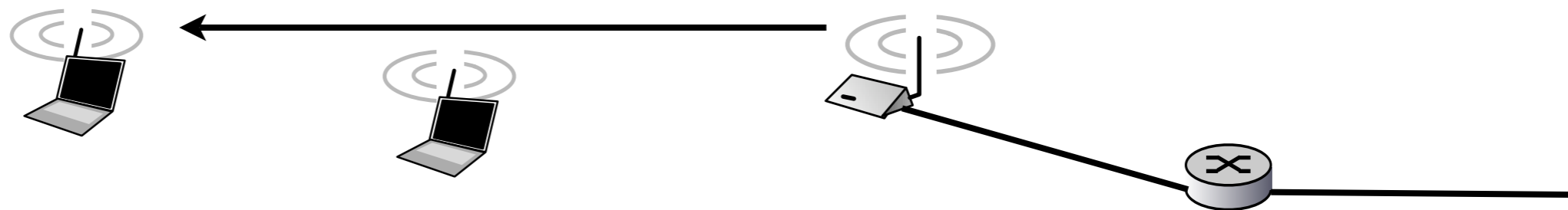
Redes inalámbricas

- ▶ Cada vez más importancia ofrecen: movilidad, facilidad de instalación, flexibilidad
- ▶ Evolución hacia comunicaciones inalámbricas
Telefonía (GPRS,3G...), dispositivos WPAN (Bluetooth, wirelessUSB...) y **redes de datos** (802.11, WiMax?...)
- ▶ Nos centraremos en **IEEE 802.11** (vulgarmente wifi)

Wifi 802.11: Nivel físico

- ▶ NICs y puntos de acceso, transmiten y reciben señales de radio/microondas a través del aire
 - > Varios estándares de modulación
 - + DSSS, FHSS, Luz infrarroja en BB (no se utiliza)
 - > Y frecuencias
 - + 2.4GHz, 5GHz, 3GHz

Paquete modulado sobre portadora de 2.4GHz con DSSS
La velocidad de datos en el canal es 11Mbps



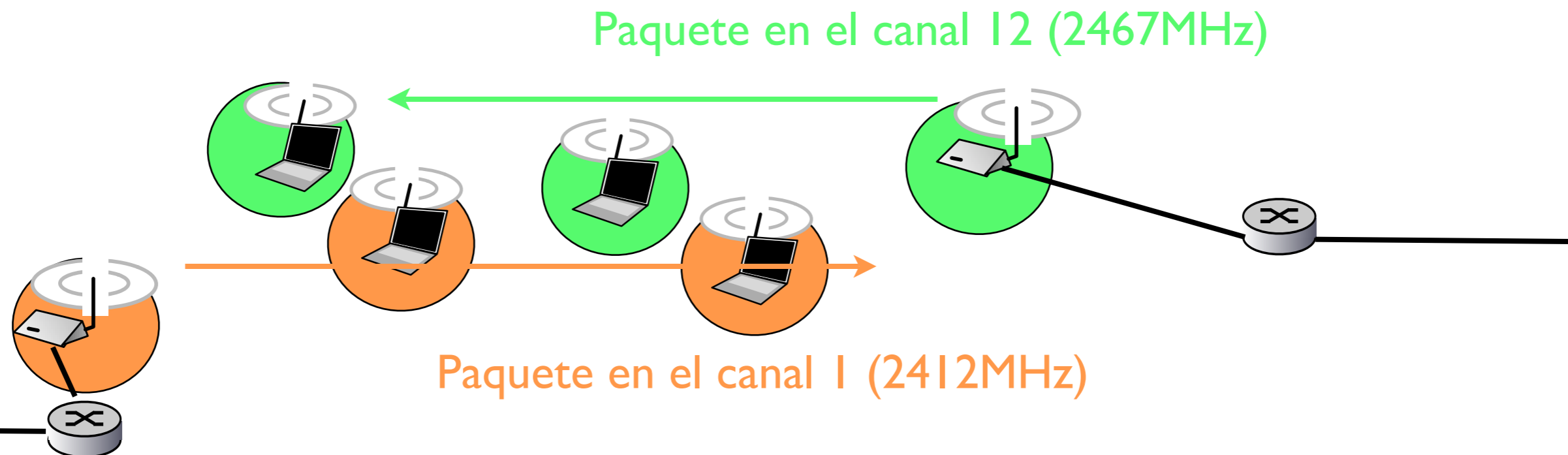
Medio compartido de broadcast

Las NICs oyen el paquete y según la cabecera lo procesan

Wifi 802.11: Nivel físico

- ▶ Versiones con el tiempo (definidos por diferentes estándares del IEEE)
- ▶ 802.11a 5GHz velocidad de datos hasta 54Mbps
- ▶ 802.11b 2.4GHz velocidad de datos hasta 11Mbps
- ▶ 802.11g 2.4GHz velocidad de datos hasta 54Mbps
- ▶ 802.11n 2.4,5GHz velocidad de datos hasta 248Mbps
- ▶ El espectro en torno a la frecuencia utilizada se divide en varios canales utilizando frecuencias cercanas.

Permite tener varias redes en el mismo espacio



Wifi 802.11: Nivel físico

- ▶ Canales en 802.11b
 - > En la banda libre de 2.4GHz
 - > Algunos son ilegales en algunos países
 - EEUU 1-11
 - EMEA 1-13

Canal	Frecuencia	Canal	Frecuencia
1	2412 MHz	8	2447 MHz
2	2417 MHz	9	2452 MHz
3	2422 MHz	10	2457 MHz
4	2427 MHz	11	2462 MHz
5	2432 MHz	12	2467 MHz
6	2437 MHz	13	2472 MHz
7	2442 MHz	14	2484 MHz

- ▶ Aún así los canales cercanos se interfieren
- ▶ De todas formas el mecanismo de acceso al medio es capaz de soportar varias redes en el mismo canal cercanas utilizando colisiones y CSMA
- ▶ Nos interesa más el nivel de enlace

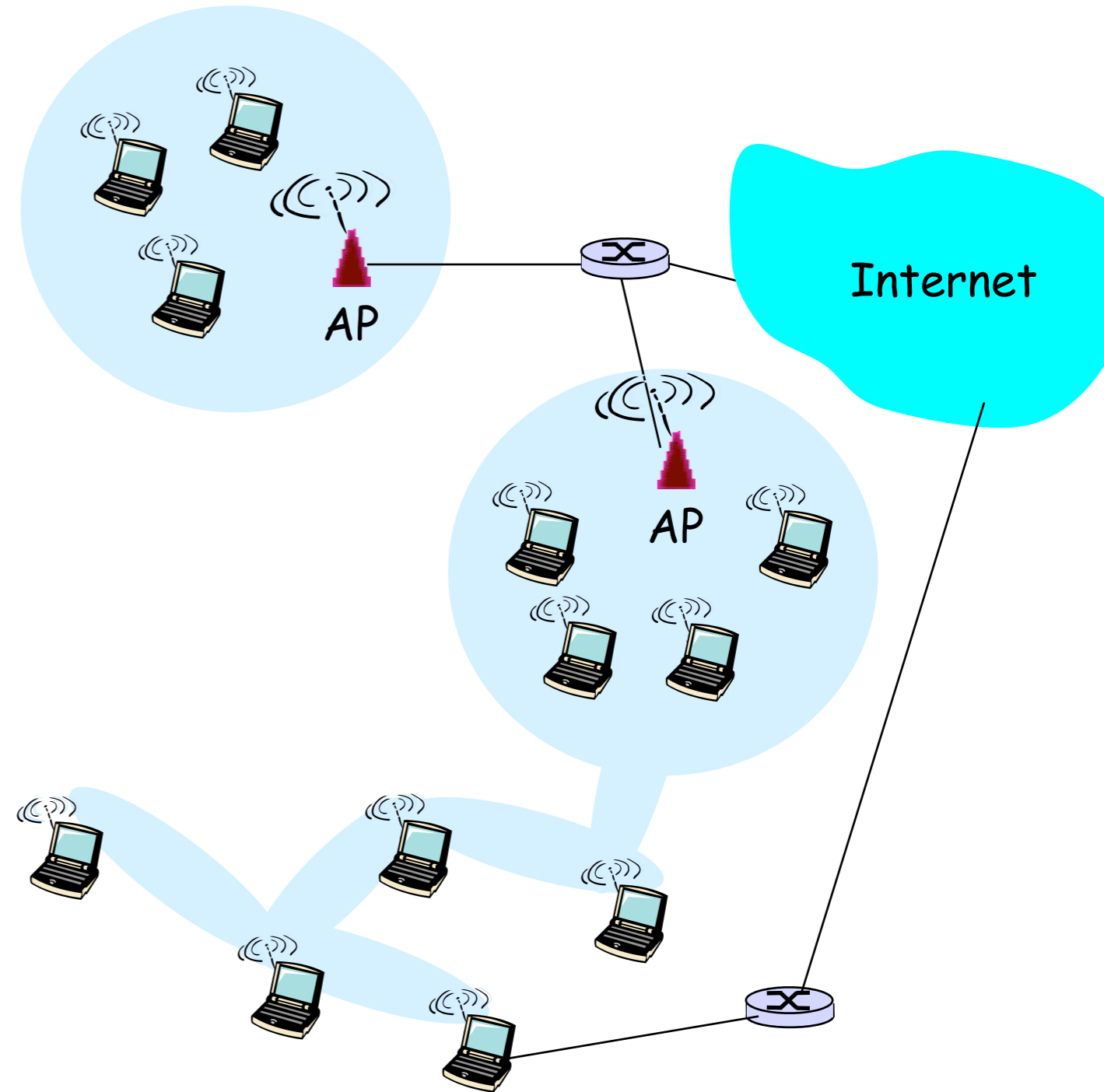
2 modos de funcionamiento

▶ **Base-station**

- > Infraestructura: estaciones base (access point) conectadas a una red fija

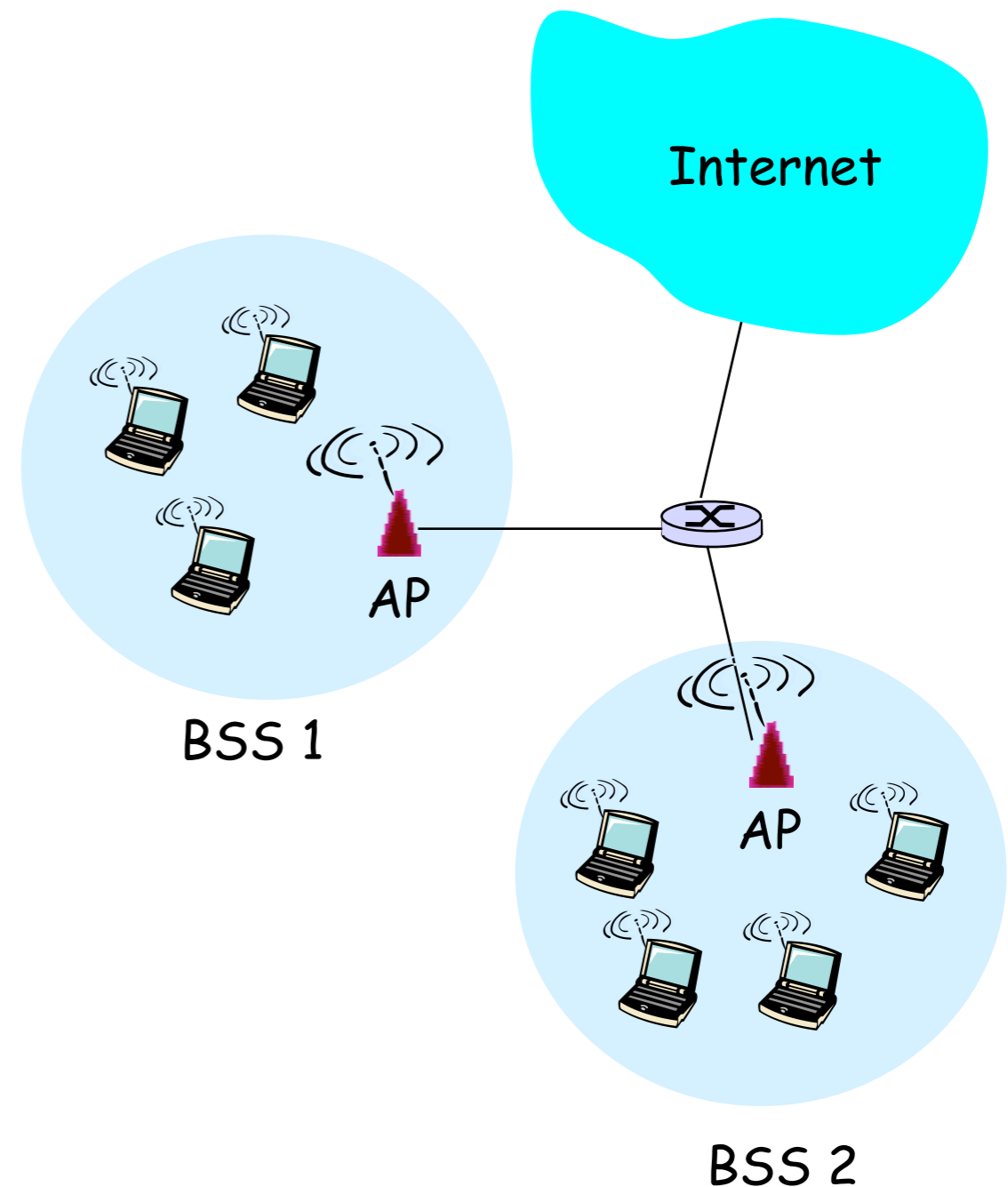
▶ **Ad-hoc**

- > punto-a-punto
- Los terminales inalámbricos se comunican entre si
- > Corren algoritmos de enrutamiento y extienden la red más allá del alcance de uno
- > parecido a peer-to-peer



Basic Service Set

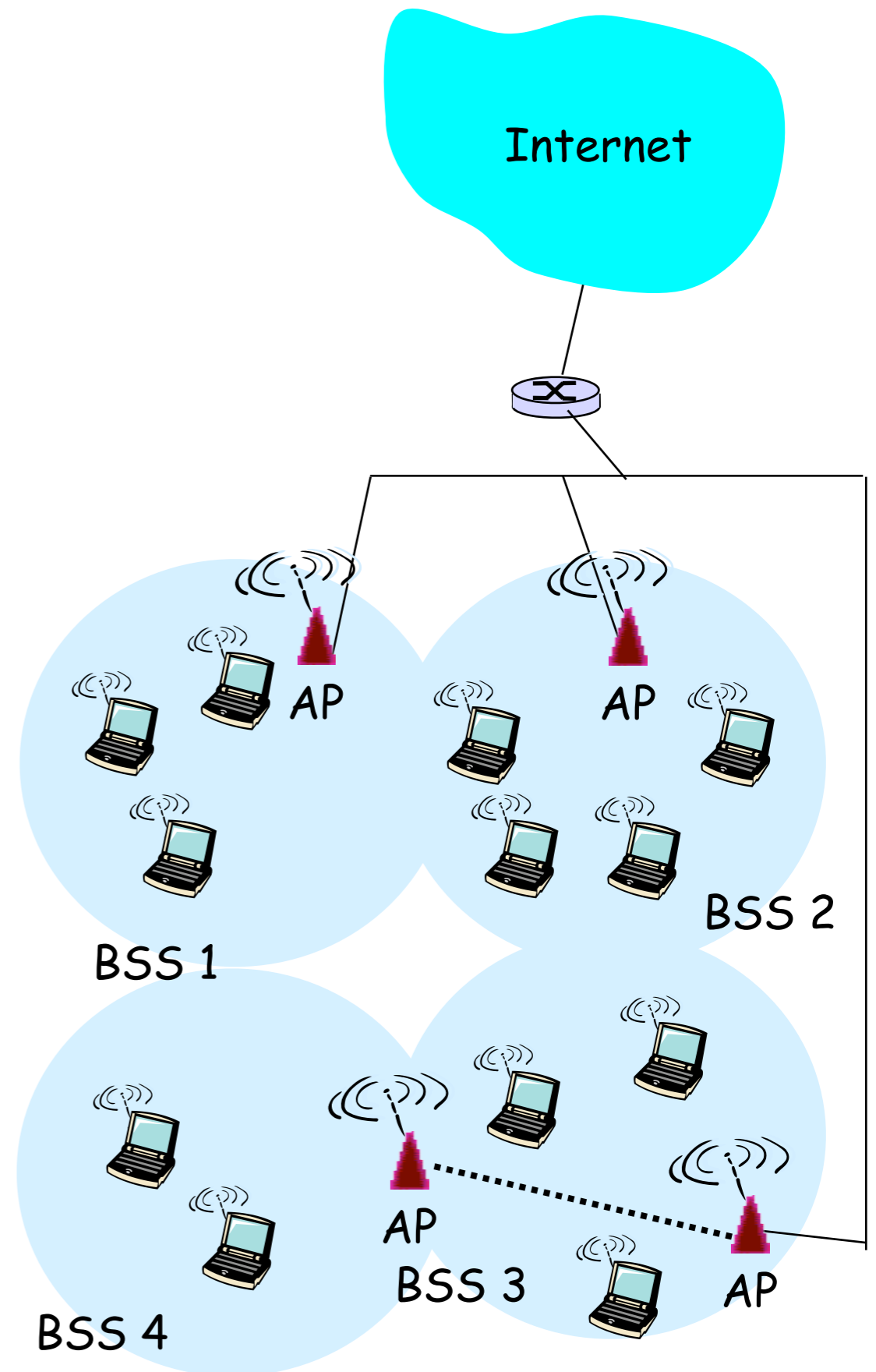
- ▶ Al conjunto formado por
 - > Hosts wireless
 - > 1 access point
 - > su router de acceso
- ▶ Le llamaremos Basic Service Set (BSS)
- ▶ Equivalentes a las celdas de la telefonía móvil



Extended Service Set

- ▶ Varios BSSs unidos para dar un servicio común en una zona mayor
- ▶ Le llamaremos Extended Service Set (ESS)
- ▶ La interconexión entre puntos de acceso puede ser por una red de cable o incluso wireless (WDS)
- ▶ El ESS tiene un identificador común de forma que el usuario no sabe si es un BSS o un ESS

El Service Set Identifier (SSID)



802.11 Asociación

- ▶ Para poder comunicarse en un BSS los hosts deben primero asociarse a la red deseada (identificada por su SSID)

- ▶ ¿Como conocen el SSID?

- > La estación base envía periódicamente tramas (beacon) con su nombre (SSID) y su dirección MAC

Eso permite a los hosts escanear los canales y presentar al usuario los SSIDs observados para que elija

- > La estación base no envía tramas beacon (SSID oculto) y el administrador es responsable de comunicar el SSID

Esto a veces se ve como una medida de seguridad pero es una medida de seguridad muy ligera. El SSID no se protege y si observas el canal y ves a otro host asociarse ves el SSID

802.11 Asociación

- ▶ Antes de transmitir un host sigue los pasos:
 - > Escanea permanentemente los canales en busca de tramas beacon (y los presenta al usuario para que elija o está configurado para buscar unos SSIDs que conoce)
 - > Una vez elegido el SSID realiza autenticación y asociación
 - + Pide autorización al Access Point para estar en la red
 - Varios protocolos que permiten comprobar si el usuario tiene acceso a la red (con contraseña (SKA), autenticación abierta (OSA) que siempre se concede)
 - + Pide al Access Point que lo considere asociado a la red
 - Al completar este protocolo el host está en el BSS
 - > Una vez realizada el host forma parte del BSS y puede enviar tramas (de nivel de enlace 802.11) a otros hosts del BSS o al router.

Normalmente lo primero que hace el host es usar protocolos de configuración de IP, enviar petición de DHCP para obtener IP y parámetros de configuración IP en la red de la estación

802.11 Asociación

existe una red llamada

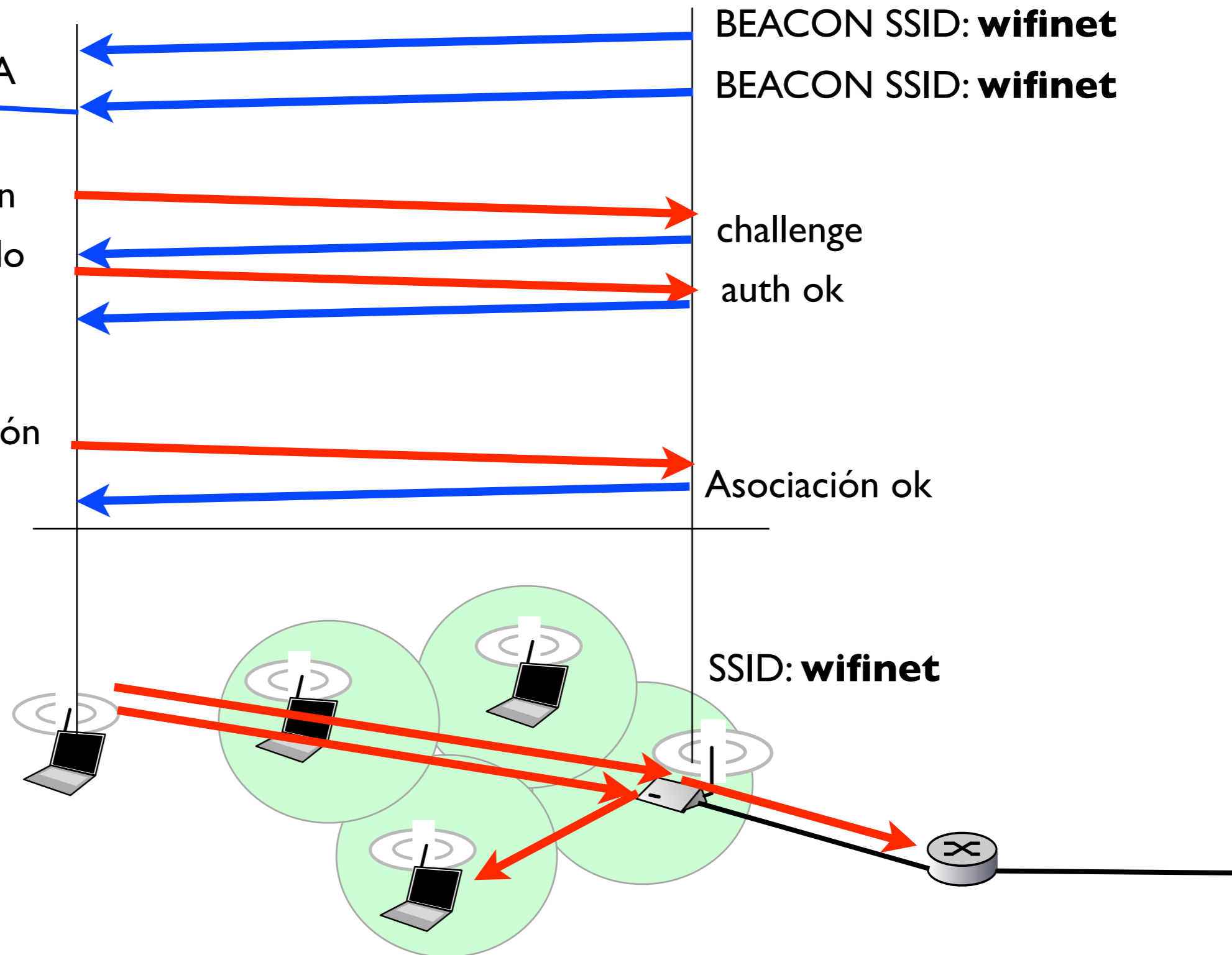
wifinet

y usa autenticación SKA
(shared key auth)

Peticion autenticación
challenge cifrado

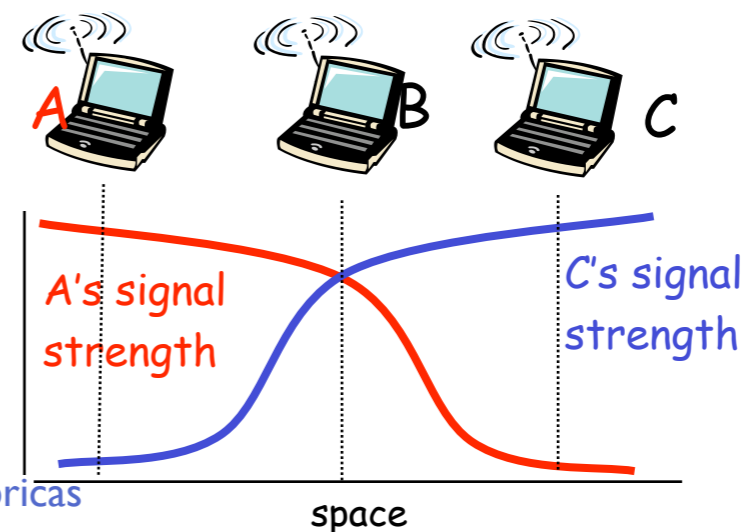
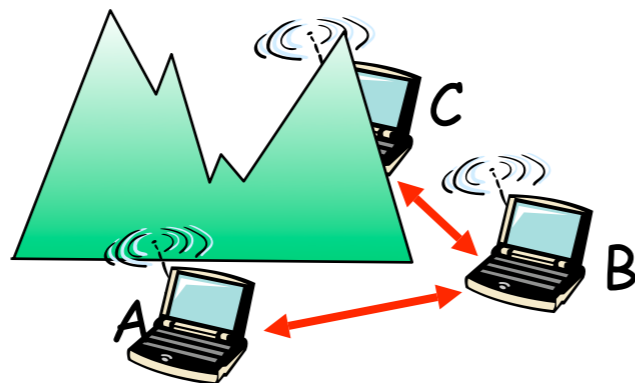
Petición asociación

A partir de aqui puedo
enviar a los demas hosts
y al router



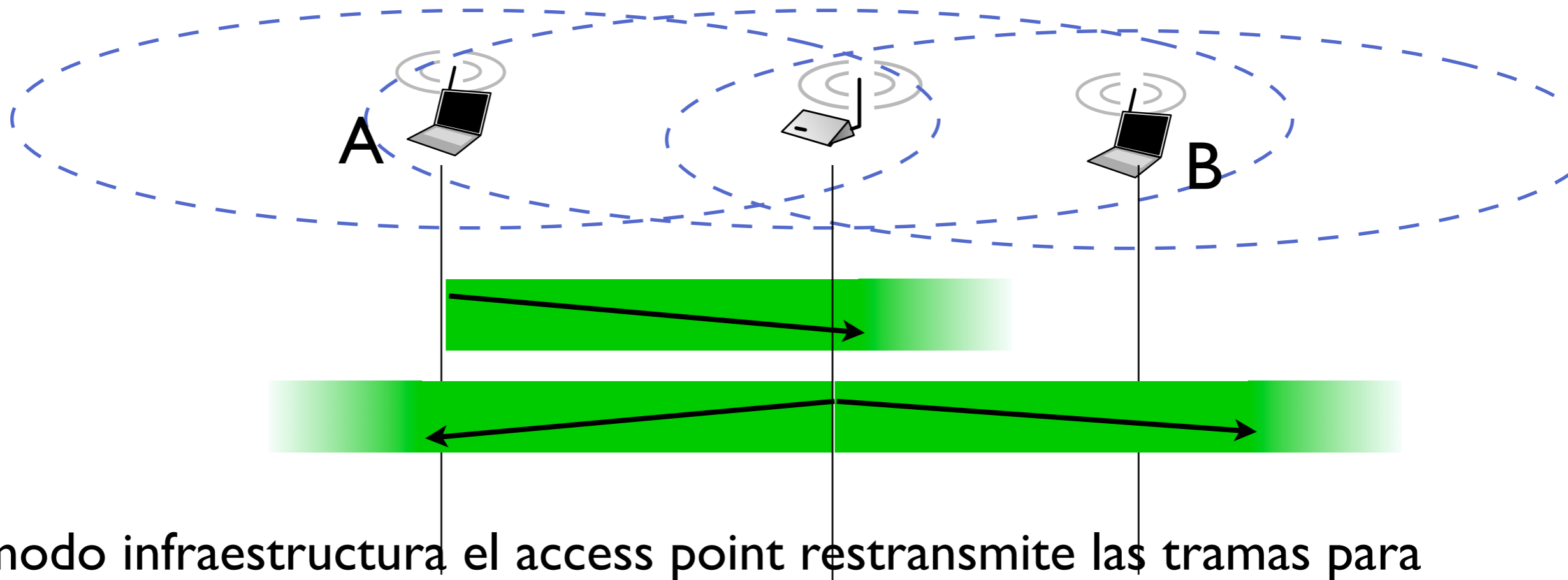
802.11 Acceso múltiple

- ▶ Acceso múltiple con problemas propios del medio inalámbrico
- ▶ Usa CSMA (carrier sense, si veo que alguien está enviando no envío)
 - > No colisiona con transmisiones en curso
- ▶ Pero la detección de colisión es un problema
 - > La señal se atenúa muy rápido por lo que es difícil comparar lo enviado con lo recibido. De hecho normalmente las NIC no pueden escuchar mientras envían
 - > Existe el problema de terminales ocultos
 - A y C no se oyen entre si
 - No pueden saber que B ve una colisión



802.11 Acceso múltiple

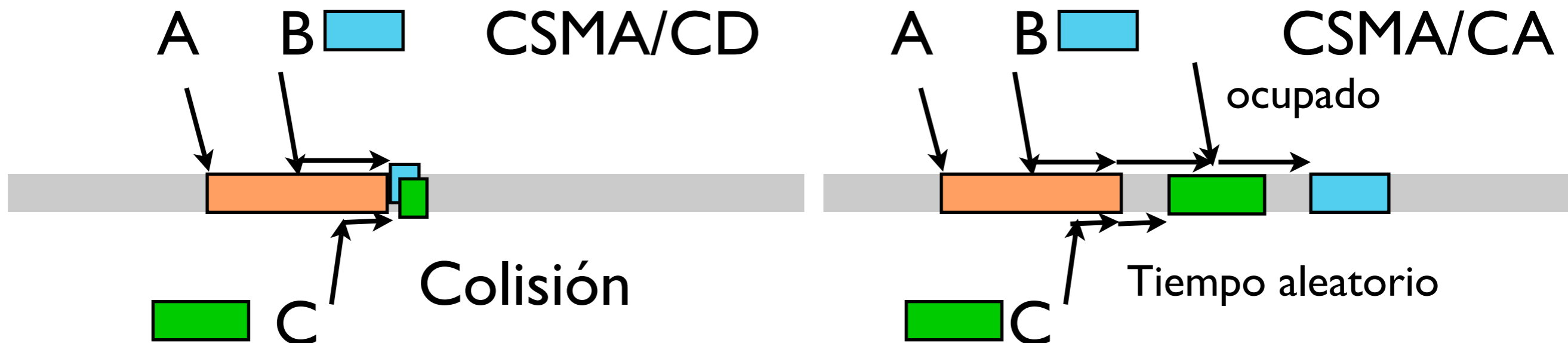
- ▶ Problemas de potencia:
 - > A oye al Access Point pero no a B



- ▶ En modo infraestructura el access point retransmite las tramas para que las oigan todos los hosts del BSS
Las transmisiones host-host pasan siempre por el access point
- ▶ Esto no soluciona el problema del terminal oculto

CSMA/CA

- ▶ Collision avoidance (evitación) en lugar de detección
- ▶ El receptor confirma (ACK) las tramas (ante los problemas para detectar si ha habido colisión)
- ▶ Se utilizan tiempos aleatorios cuando voy a transmitir
 - > Objetivo: evitar las colisiones causadas entre las estaciones que esperan que el medio quede libre



CSMA/CA

▶ Emisor 802.11

> Si el canal está vacío por un tiempo DIFS

- + Envía la trama entera (sin CD)

> Si el canal está ocupado

- + Inicia un temporizador aleatorio (con backoff)

- + El temporizador solo descuenta tiempo con canal libre

- + Transmite cuando expire

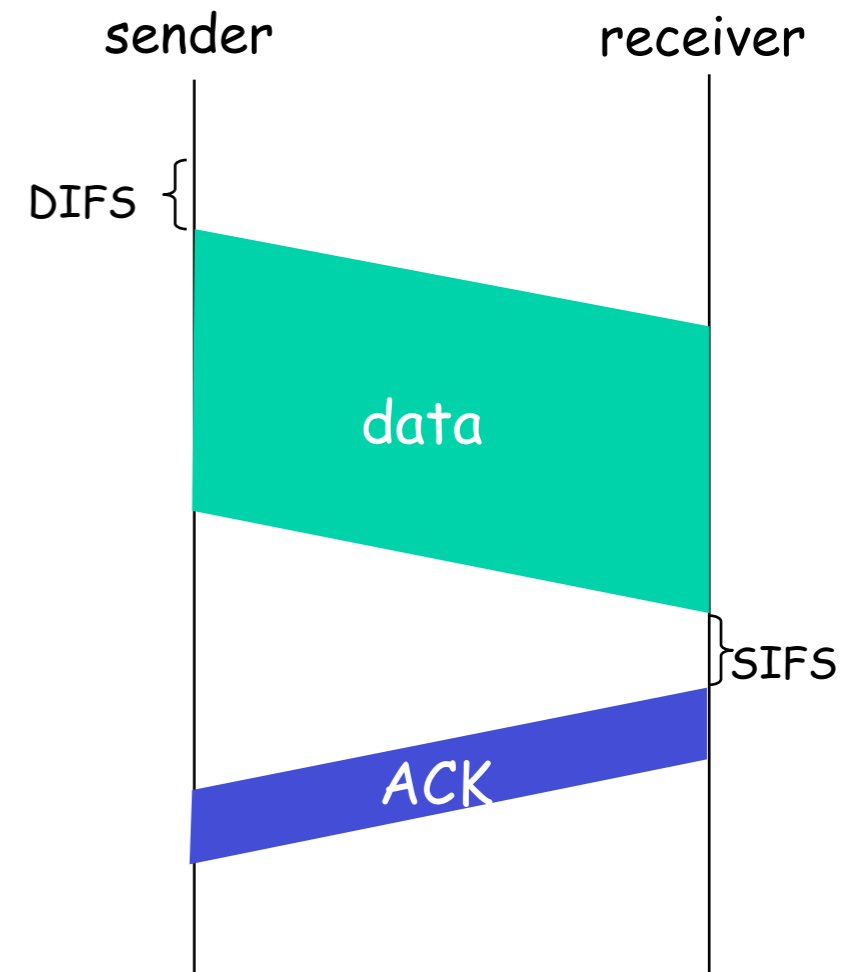
- + Si no recibe ACK aumenta el backoff

▶ Receptor 802.11

> Si recibo una trama

- + Envía ACK después de un SIFS

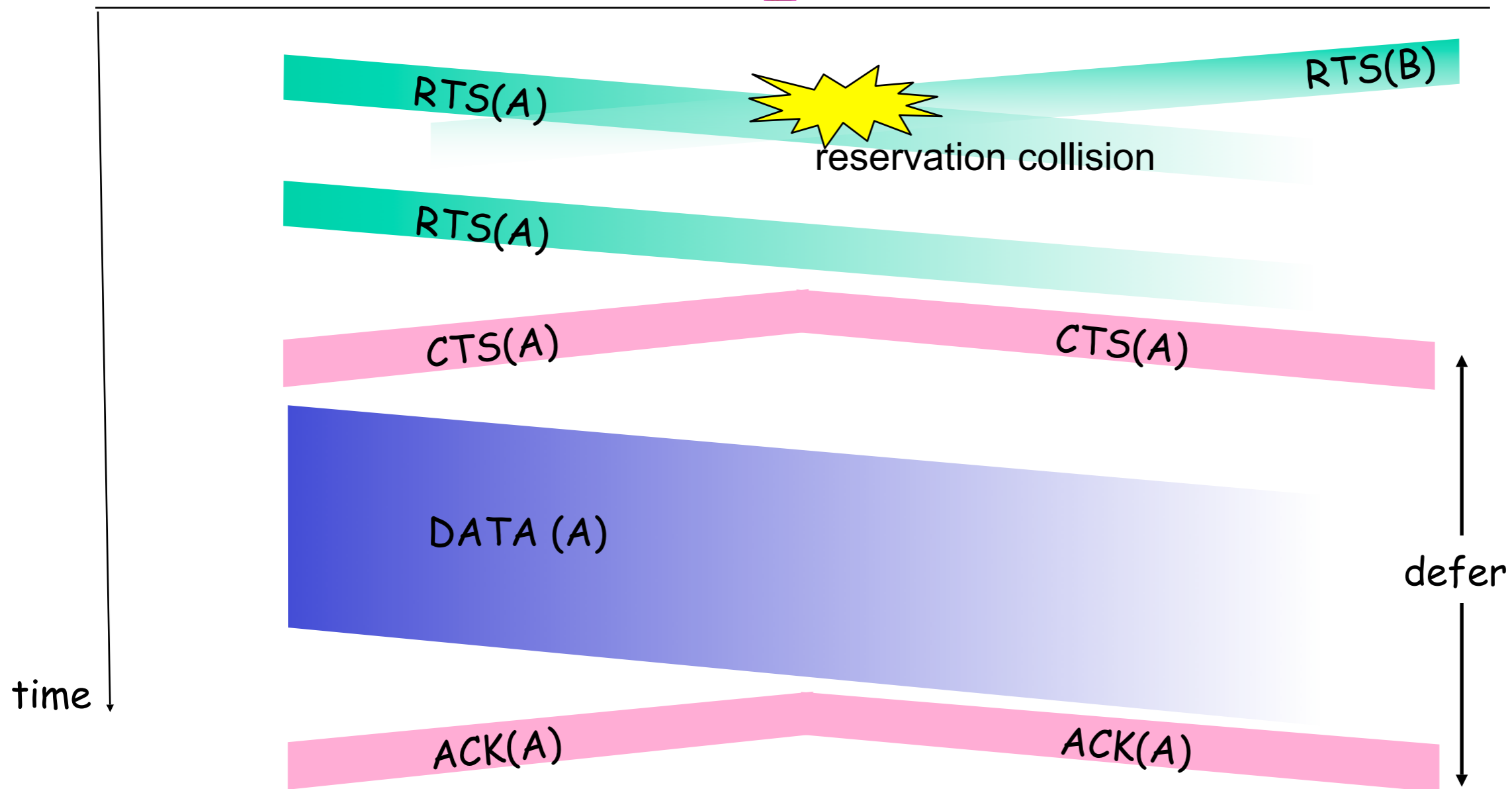
($SIFS < DIFS$ los ACKs tienen prioridad)



CSMA/CA

- ▶ Mejora: permitir al emisor reservar el canal para evitar colisiones en las tramas muy largas
 - > El emisor envía una trama de RTS (request to send) a la estación base pidiendo el canal (usando CSMA/CA)
Los RTS pueden colisionar con otras tramas pero al menos son cortas
 - > La estación base envía el permiso en una trama CTS (Clear to send)
 - > Todos los nodos reciben la CTS
 - + El solicitante envía la trama
 - + El resto dejan libre el canal
- ▶ Evita completamente las colisiones
 - > A costa de más retardo
 - > Normalmente se activa sólo para tramas por encima de una longitud

Ejemplo



Coordination function

- ▶ Esto es conocido como funcionamiento con función de coordinación distribuida

DCF

- ▶ El estándar también soporta tipo polling

Point Coordination Function (PCF)

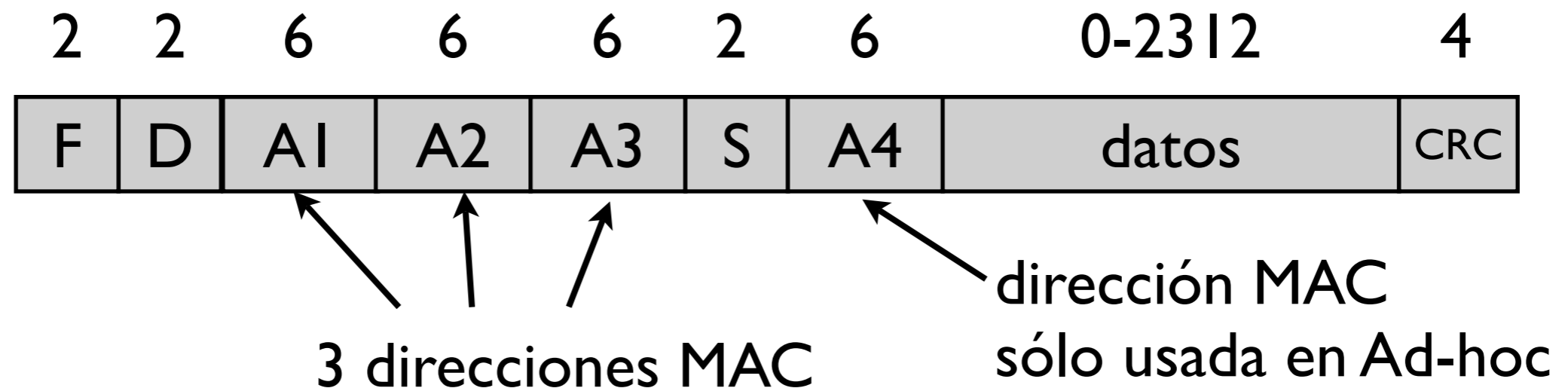
- ▶ En modo Adhoc solo se usa la DCF
- ▶ En modo infraestructura se puede usar DCF o DCF+PCF
 - > Contention Free Periods (con PCF) + Contention Periods (con DCF)
- ▶ Pero PCF no se usa mucho

- ▶ 802.11e HCF Hybrid Coordination Function y soporte de QoS

Resumiendo

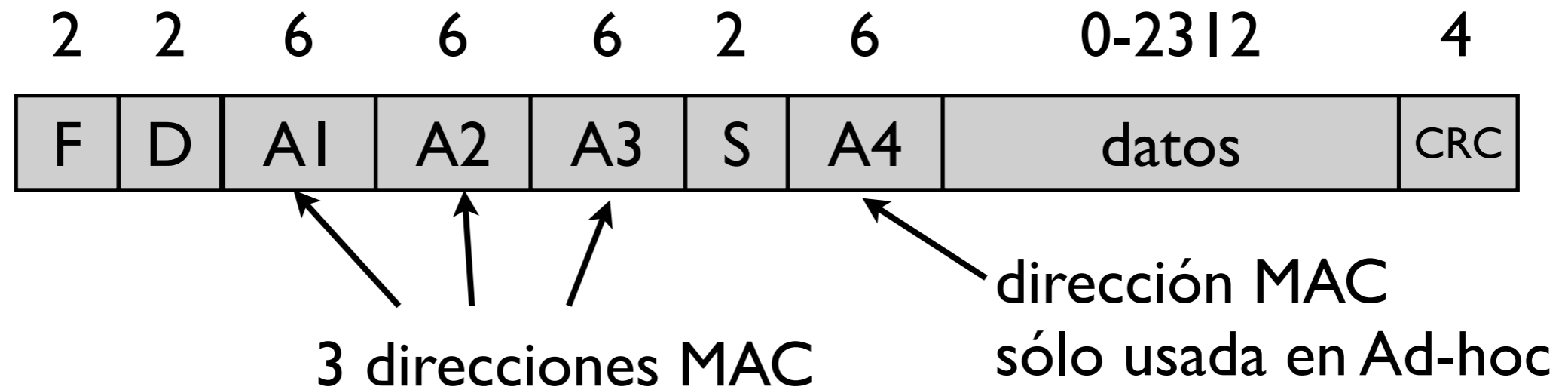
- ▶ Control de acceso al medio más complicado que en Ethernet
 - > Hay ACKs en el nivel de enlace
 - > Hay retransmisiones en el nivel de enlace
 - > Hay autentificación/asociación
 - > El access point retransmite tramas

802.11: formato de trama



- ▶ S: secuencia de la trama
 - > necesario para el ACK
- ▶ 4 direcciones MAC
 - > A1: MAC destino. Wireless host que debe recibir esta trama
 - > A2: MAC origen. Wireless host que envia esta trama
 - > A3: MAC router asociado al access point
 - > A4: usada en modo Ad-hoc o para interconectar access-points a traves de la red inalámbrica (WDS)

802.11: formato de trama



- ▶ F frame control
 - > Flags y tipo de la trama (Data, ACK, RTS o CTS)
- ▶ D duración
 - > Tiempo por el que se solicita el canal
En RTS/CTS

802.11 tipos de tramas

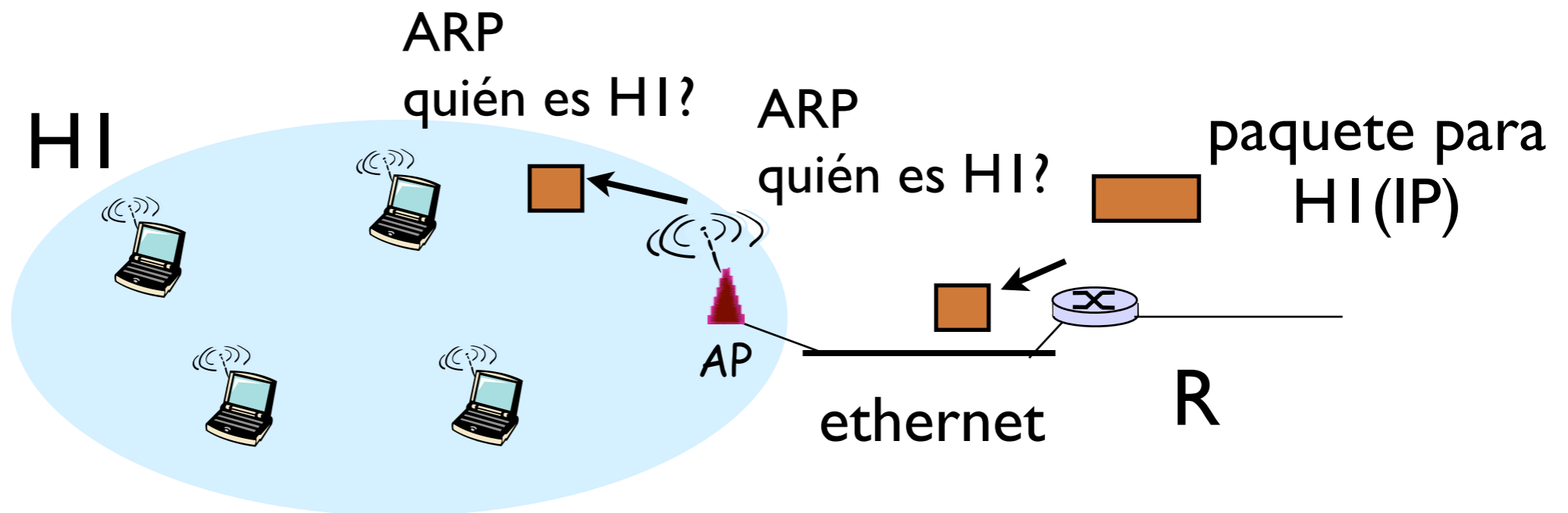
- ▶ El campo de control de la trama permite definir tipos y subtipos de tramas

Tipo	Subtipo	Nombre
00 (Management)	0	Asociación (request)
	1	Asociación (response)
	100	Probe request
	101	Probe response
	1000	Beacon
	otros
01 (Control)	1011	RTS
	1100	CTS
	1101	ACK
	...	otros
10 (Datos)	0	Datos
	otros opciones y QoS
11 (No se usa)		

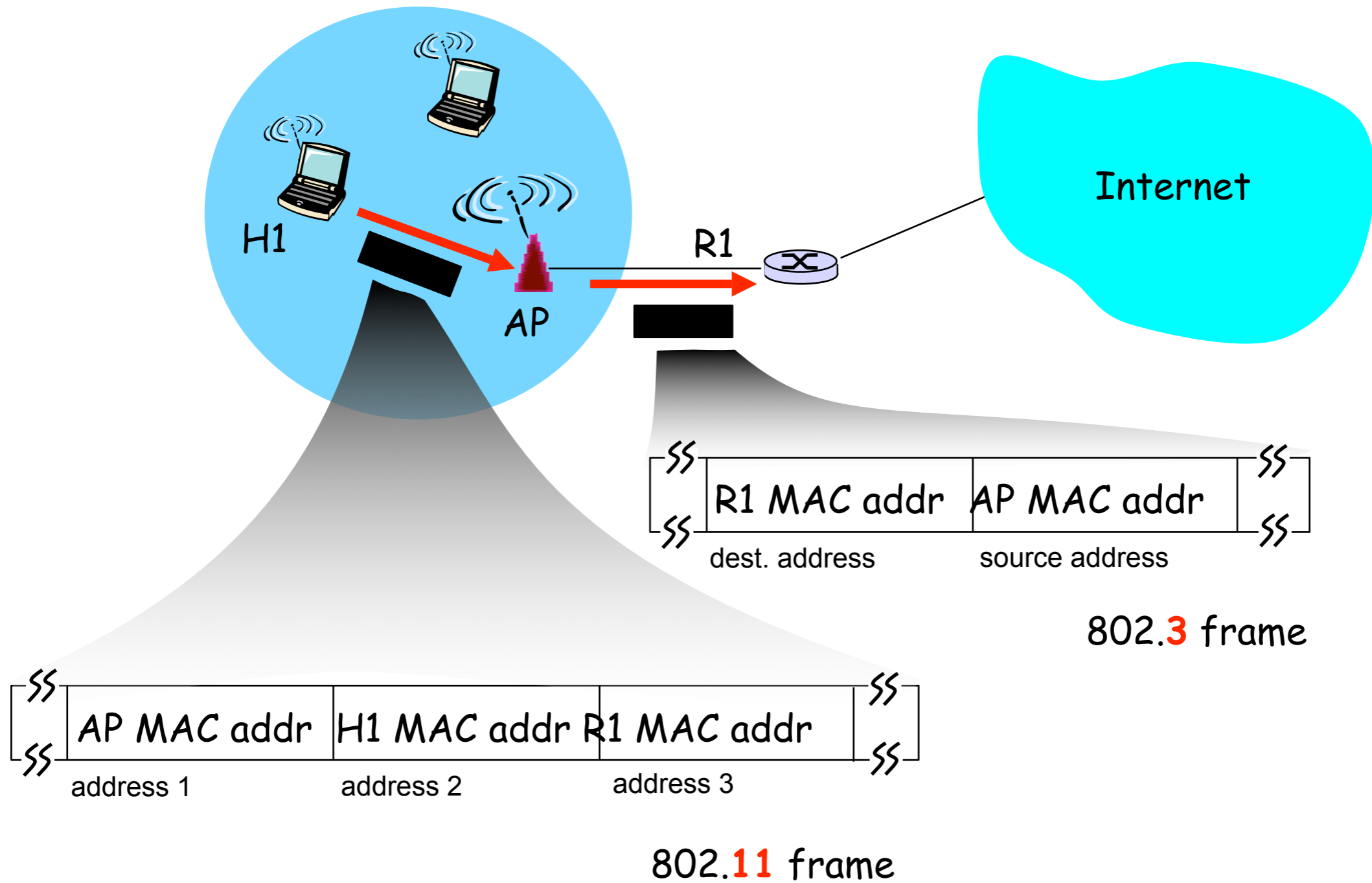
Por qué 3 direcciones?

- ▶ El access point es un dispositivo de nivel de enlace
- ▶ Para los dispositivos conectados al access point no debe haber diferencia entre hosts alámbricos o inalámbricos
- ▶ Como funcionaría el ARP aquí?

YO
pero
a que MAC
contesto?



Ejemplo



En resumen

- ▶ Medio inalámbrico compartido
- ▶ Las redes de área local inalámbricas siguen técnicas parecidas a las de cable
 - > CSMA
 - > Pero CSMA/CA en lugar de CD, colisiones costosas mejor evitar
 - > Se pueden usar técnicas de reserva de canal

Seguridad en redes 802.11

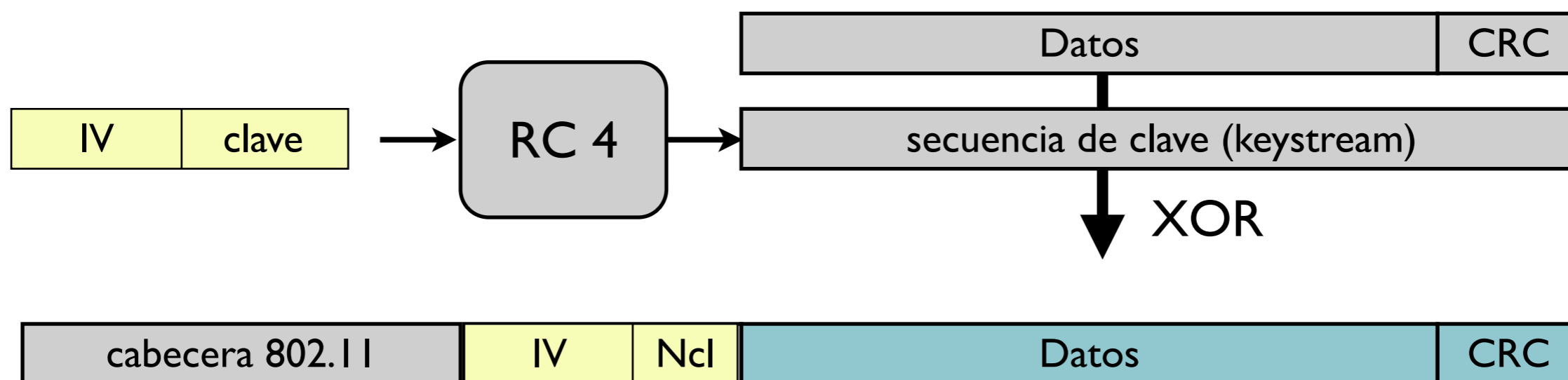
- ▶ **Wired Equivalen Privacy (WEP)**

Conseguir en la red inalámbrica el mismo nivel de privacidad que en una de cable

- ▶ Proteger la confidencialidad de los datos que se transmiten por el aire: cifrar las tramas de datos
- ▶ Proteger la integridad de los mensajes
- ▶ Se utiliza el algoritmo de cifrado RC4

WEP

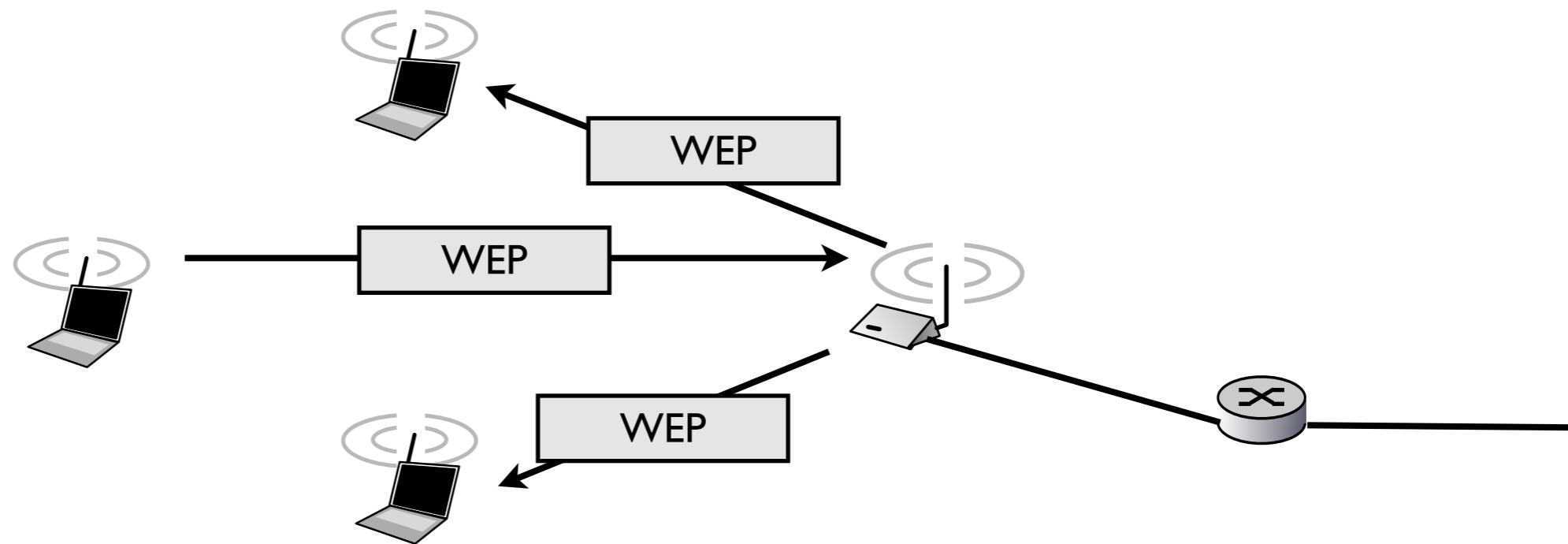
- ▶ A los datos de la trama se les añade un CRC para proteger la integridad y se cifran con RC4



- ▶ Se usan una clave de 64 o 128 bits
 - > Vector de inicialización de 24 bits
 - > Secreto compartido de 40 o 104 bits
- ▶ El vector de inicialización se cambia en cada paquete para cifrar cada paquete con diferente secuencia. Se envía en cada paquete para que el destinatario sea capaz de descifrar.

WEP

- ▶ Enviando con WEP
 - > El terminal calcula el CRC del paquete y cifra el paquete con WEP
 - > El paquete se envía al access point
 - > El access point descifra el paquete y si el CRC es inválido lo tira
 - > El access point puede cifrarlo con otro IV y enviarlo



- ▶ Un intruso
 - > No puede descifrar los paquetes que le llegan
 - > No puede generar paquetes válidos para otros

Ventajas

- ▶ Autenticación sencilla: los usuarios que conozcan la clave pueden usar la red inalámbrica
- ▶ Protección de integridad y confidencialidad “razonable”
 - > o no?

Seguridad en redes 802.11

▶ Primer intento: **Wired Equivalen Privacy (WEP)**

Conseguir en la red inalámbrica el mismo nivel de privacidad que en una de cable

▶ Se cifran las tramas con el algoritmo RC4

- Algoritmo de cifrado de tipo clave secreta

Se basa en generar una serie pseudo-aleatoria a partir de la clave secreta. El mensaje se cifra con una clave de la misma longitud que el mensaje pero que depende de la clave original (intento de hacer un cifrado de Vernan)

▶ Originalmente era un algoritmo propietario de RCA Security

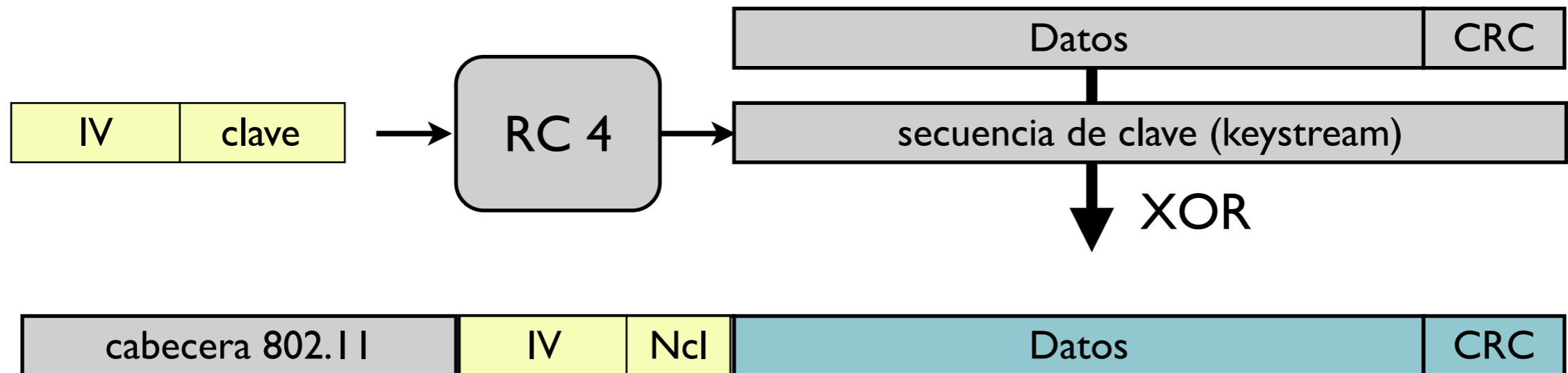
- Pero se publicó de forma anónima en Internet y se popularizó

El algoritmo cifra a gran velocidad y parecía muy seguro

- Con el tiempo se le han ido encontrando algunos

WEP

- ▶ A los datos de la trama se les añade un CRC para proteger la integridad y se cifran con RC4



- ▶ Se usan una clave de 64 o 128 bits
 - Vector de inicialización de 24 bits
 - Secreto compartido de 40 o 104 bits
- ▶ El vector de inicialización se cambia en cada paquete para cifrar cada paquete con diferente secuencia. Se envía en cada paquete para que el destinatario sea capaz de descifrar.

Ventajas

- ▶ Autenticación sencilla: los usuarios que conozcan la clave pueden usar la red inalámbrica
- ▶ Protección de integridad y confidencialidad “razonable”
 - o no?

Desventajas

- ▶ Múltiples vulnerabilidades del sistema
 - Contra la confidencialidad
 - La clave se reutiliza. El vector de inicialización de 24 bits solo hay que esperar 16777216 paquetes para que se repita y tener dos paquetes encriptados con la misma clave
 - RC4 tiene claves débiles. Algunos IVs generan claves en las que ciertas partes de la clave secuencia dependen solo de unos pocos bits de la clave original
 - Ataques de fuerza bruta (el secreto compartido depende de una clave introducida por el usuario)
 - Contra la integridad
 - El CRC que se usa fue diseñado para detectar errores no para integridad así que no es un buen hash
 - No hay protección contra inyección de paquetes
 - Si repito un paquete que veo en el canal sigue siendo un paquete válido
 - Contra la autenticación
 - Autenticación falsa
 - Ataques de desautenticación

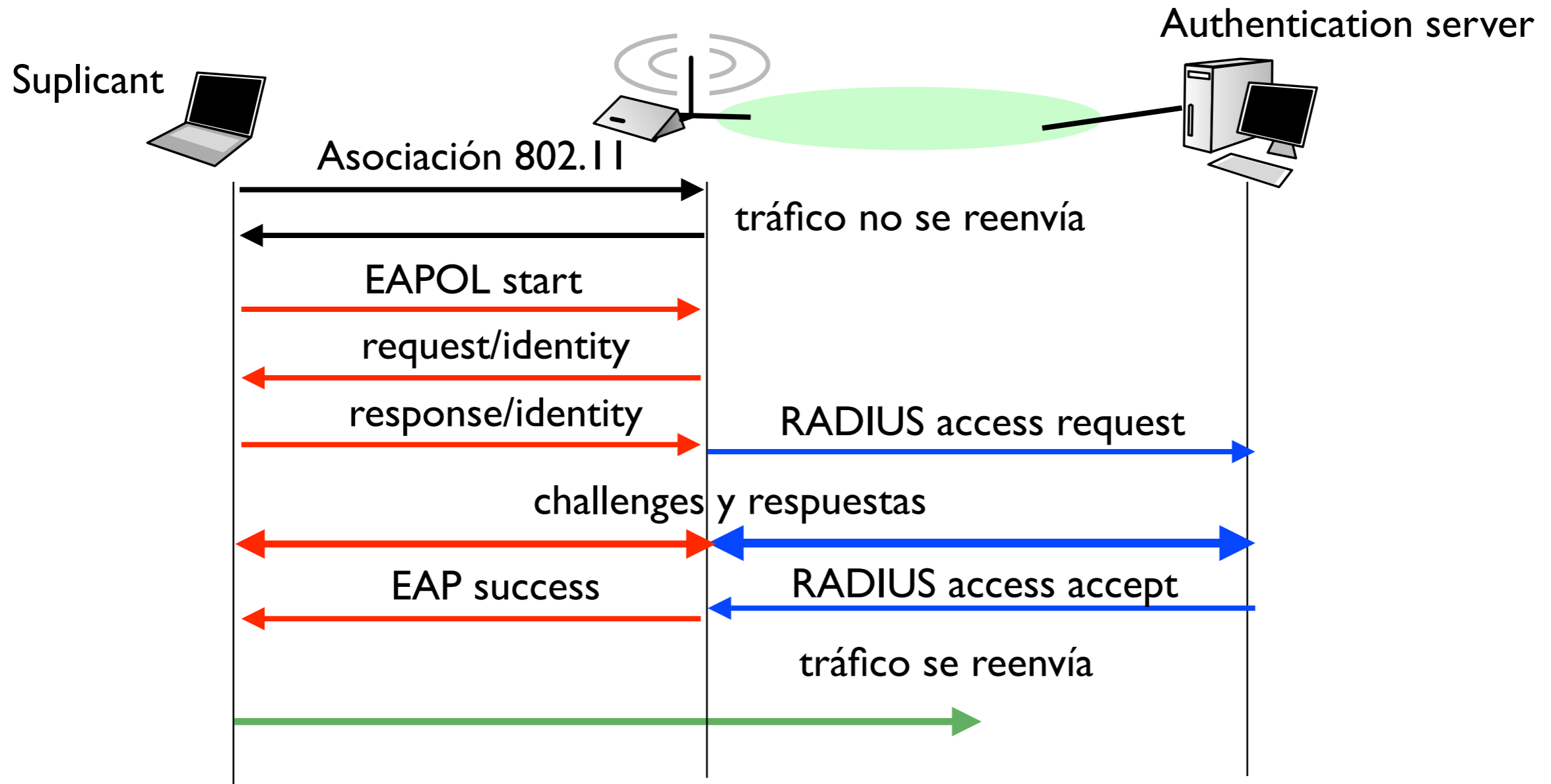
Mejorando confidencialidad de WEP

- ▶ 802.11i Estándar del IEEE sobre seguridad mejorada en redes 802.11

Añade:

- ▶ Autenticación basada en 802.1x
- ▶ 2 nuevos protocolos de cifrado para sustituir a WEP:
 - TKIP: protocolo basado en RC4 pero corrigiendo los problemas de WEP (iba a ser WEP2)
Fácil de cambiar en hardware que ya soporte WEP
 - CCMP: protocolo completamente rediseñado para nuevo hardware, basado en AES

802.1x en inalámbricas



- ▶ Se puede usar 802.1x en una red de acceso inalámbrica
- ▶ 802.1x autentifica al usuario aunque cambie de máquina
- ▶ Acceso protegido por 802.1x no importa que se averigüe la clave WEP
 - Salvo para confidencialidad

Comercialmente

- ▶ Nombres de la WiFi alliance para los equipos reales
- ▶ **WPA (WiFi protected access)** nombre comercial de TKIP. Se definió a partir del borrador de 802.11i cuando aun se trabajaba en el standar.
TKIP se implemento antes debido a que estaba basado en el hardware de WEP
- ▶ **WPA2** = 802.11i estandar. Con CCMP
- ▶ Ambos tienen dos formas de funcionamiento
 - **WPA personal**
Basado en secreto compartido (las claves se calculan a partir de una clave definida en los BSS y en los PCs)
 - **WPA enterprise**
Clave basada en TLS y certificados

Es WPA suficiente?

- ▶ Es mucho más difícil de atacar aunque hay propuestas de ataques basados en fuerza bruta
- ▶ En WPA personal sigue pudiéndose hacer ataques de diccionario a la autenticación
- ▶ Se siguen pudiendo hacer ataques de bajo nivel
 - Inundación de paquetes de deautenticación o desasociación
 - Robo de ancho de banda
 - Denegación de servicio por Jamming/interferencia

Ad-hoc

Redes Adhoc

- ▶ **Mobile Ad-hoc Networks (MANET)**
 - Red formada entre dispositivos inalámbricos móviles para la ocasión
 - Enlaces inalámbricos
 - Alta movilidad
 - Cuestiones de ahorro de potencia... los dispositivos usan baterías

- ▶ **WMN wireless mesh networks**
 - Red de bajo coste basada en enrutamiento cooperativo sobre un backbone inalámbrico
 - Enlaces inalámbricos
 - Movilidad reducida (o solo de una parte de los dispositivos)
 - Los dispositivos tienen alimentación eléctrica (y entonces por que no tienen red de cable?)

Enlaces Ad-hoc

- ▶ Enlaces 802.11 de tipo Ad-hoc
 - punto a punto o multipunto
 - IP configurada

 - No asociacion?
 - Solo DCF
 - No access point

Enrutamiento en redes Adhoc

- ▶ Position aware
 - Protocolos que hacen uso de la posición geográfica
 - Enviar a los vecinos más o menos en la dirección del destino

- ▶ Position unaware
 - Como los protocolos de enrutamiento tradicionales
 - vecinos y grafo sin significado geográfico

Position-unaware Routing

Position-unaware Routing Protocols can be classified based on the way a protocol tries to find a route to a destination:

- ▶ Proactive Routing Protocol
- ▶ Reactive Routing Protocol

Proactive Routing

- ▶ Entire network topology is known to all nodes and maintained in a routing table
- ▶ Since each node knows the complete topology, a node can immediately find the best route to a destination.
- ▶ Routing messages are exchanged among the nodes periodically to update their routing tables
 - Routing Table Advertising

Protocols:

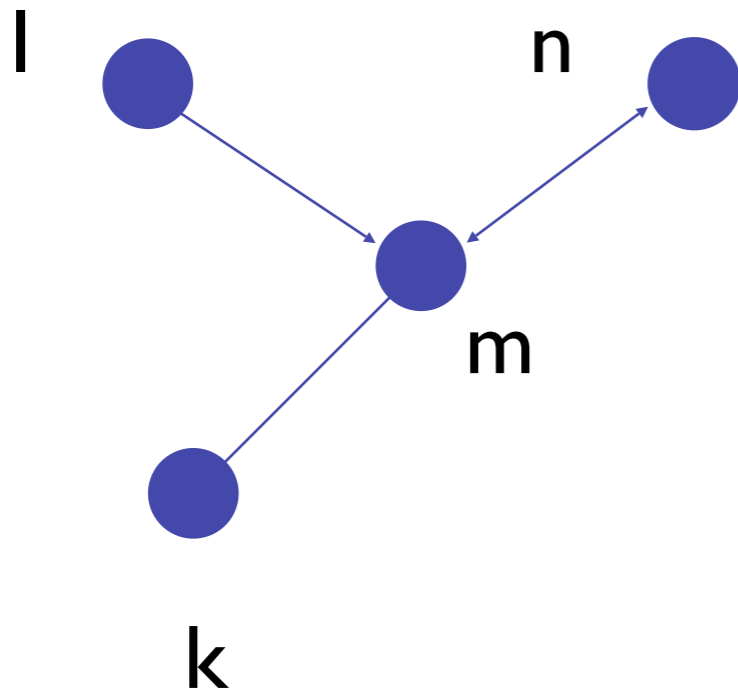
- ▶ Destination-Sequenced Distance Vector (DSDV)
- ▶ Fisheye State Routing (FSR)
- ▶ ...

Destination-Sequenced Distance Vector Protocol

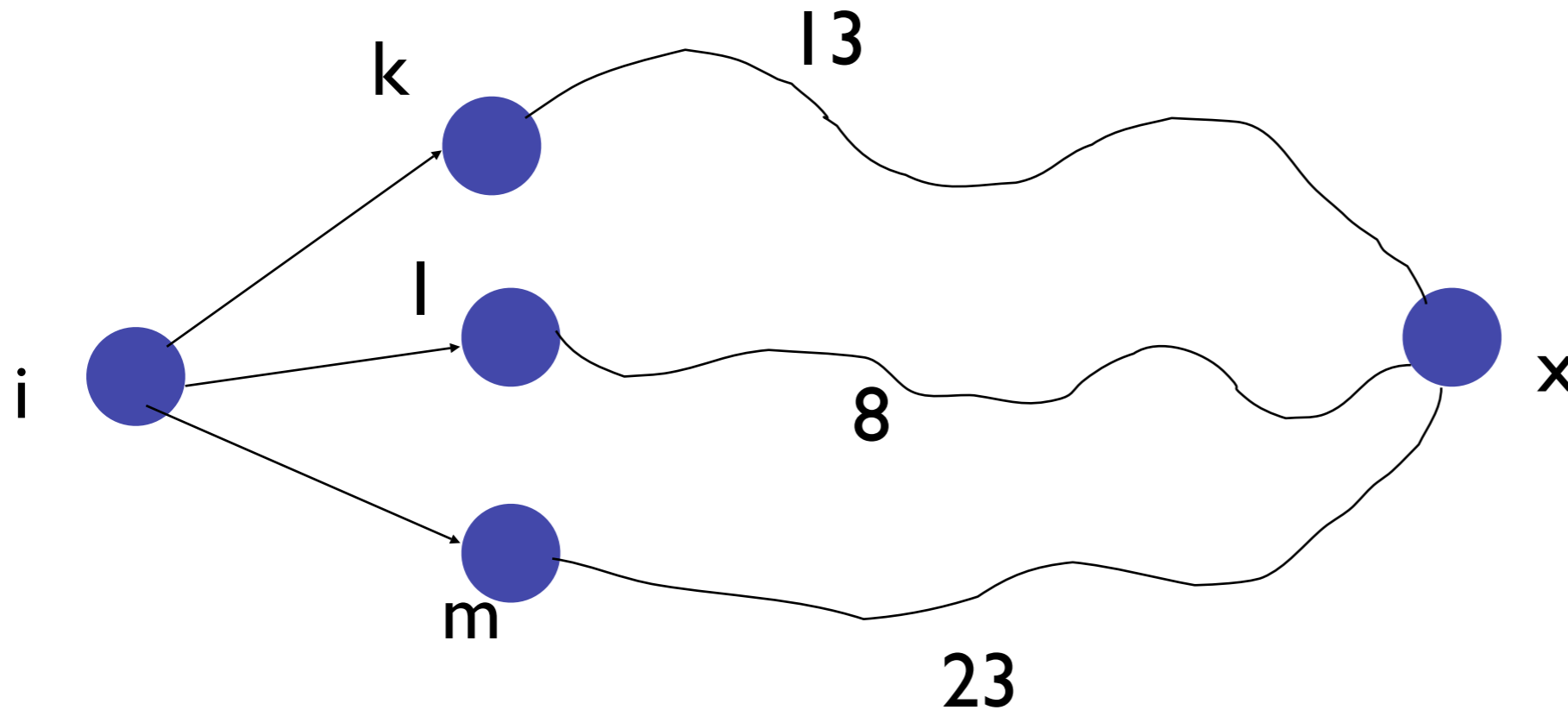
- ▶ Packets are transmitted between the nodes using route tables stored at each node.
- ▶ Each route table lists all available destinations and the number of hops to each destination.
- ▶ For each destination, a node knows which of its neighbours leads to the shortest path to the destination.

Routing Table Entries

- ▶ The destination's address
- ▶ Next Hop to Destination
- ▶ The number of hops to the destination
- ▶ The sequence number of the information received from that destination. This is the original sequence number assigned by the destination.



How the local Routing Table is Used



- Consider a node i. Suppose, i needs to send a message to node x.
- i can look up the best route to x from its routing table and forwards the message to the neighbour along the best route.
- The neighbour in turn checks the best route from its own table and forwards the message to its appropriate neighbour. The routing progresses this way.

Routing Table Advertising

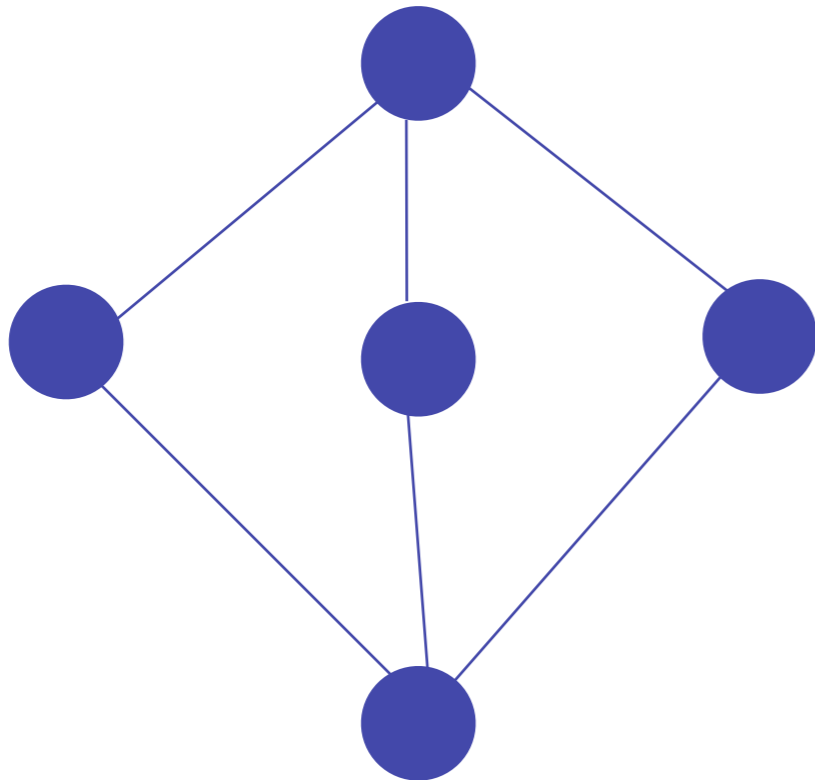
- The DSDV protocol requires each mobile node to advertise its own route table to all of its current neighbours.
- Each mobile node agrees to forward route advertising messages from other mobile nodes.
- This forwarding is necessary to send the advertisement messages all over the network.
- In other words, route advertisement messages help mobile nodes to get an overall picture of the topology of the network.

Responding to Topology Changes

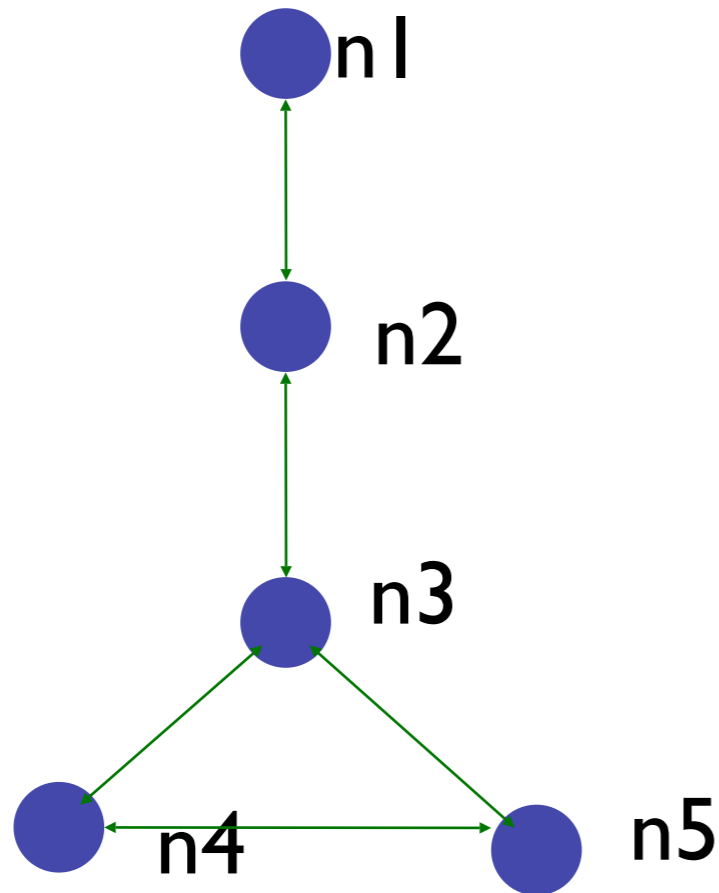
- ▶ It is necessary to avoid excessive control traffic (route update information). Otherwise, the bandwidth will be taken up by control traffic.
- ▶ The solution is to broadcast two types of updates:
 - Full Dump / Incremental Dump
- ▶ A full dump carries complete routing tables. A node broadcasts a full dump infrequently.
- ▶ An incremental dump carries minor changes in the routing table. This information contains changes since the last full dump.
- ▶ When the size of an incremental dump becomes too large, a full dump is preferred.

Responding to Topology Changes

- ▶ When a node i receives incremental dump or full dump from another node j , the following actions are taken :
 - The sequence number of the current dump from j is compared with previous dumps from j
 - If the sequence number is new, the route table at i is updated with this new information. (Reason for Sequence number: Loops)
 - Node i now broadcasts its new route table as an incremental or a full dump.



An Example of Route Update



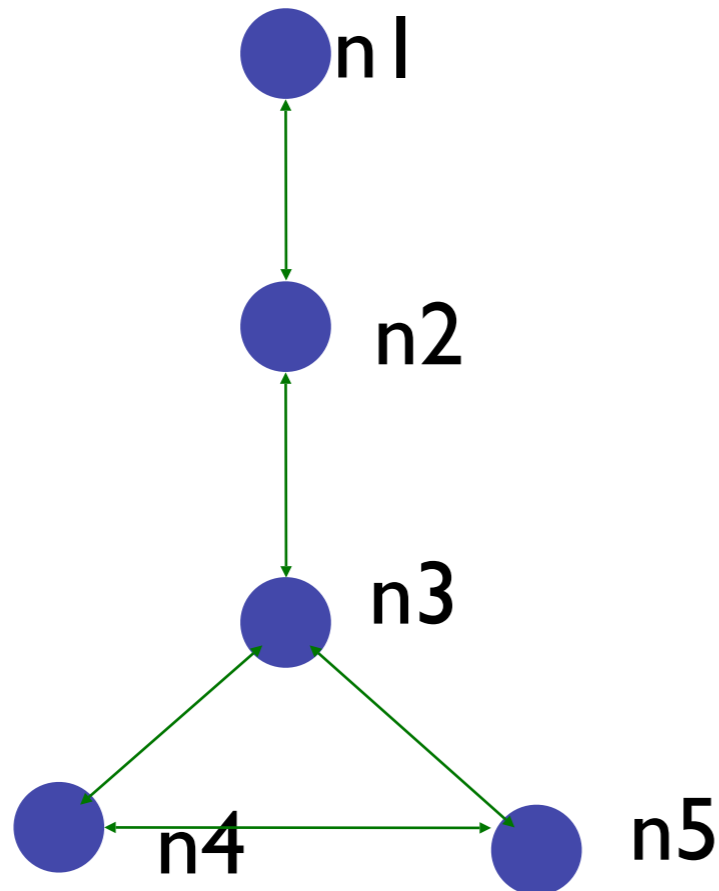
- ▶ At the start, each node gets route updates only from its neighbours.
- ▶ For **n4**, the distances to the other nodes are :

n5=1, n3=1, n2= ∞

n1 = ∞

All nodes broadcast with a sequence number **1**

An Example of Route Update



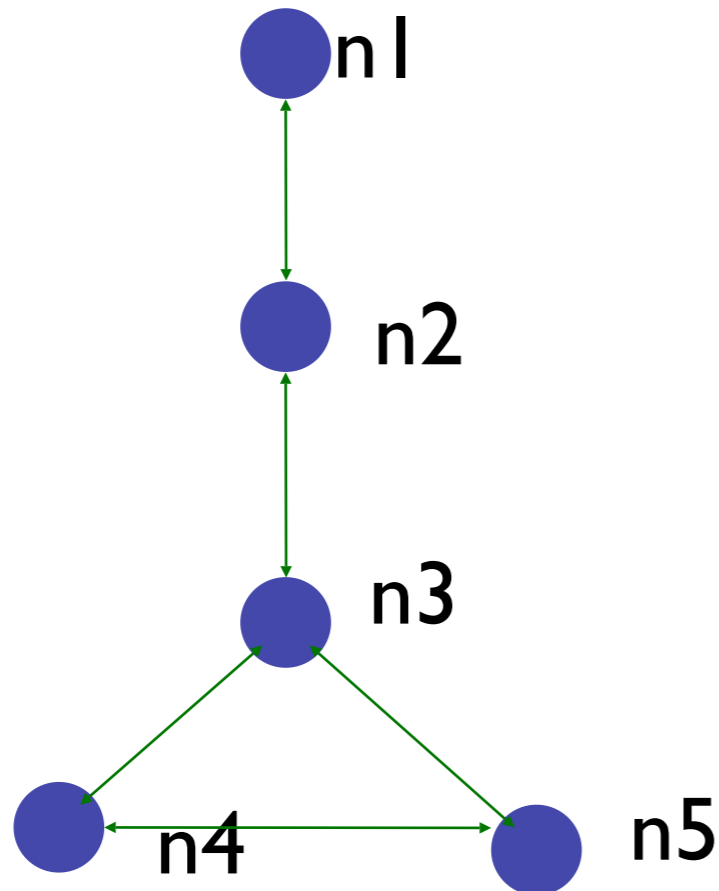
- ▶ After this, nodes forward messages that they have received earlier.
- ▶ The message that n2 sent to n3 is now forwarded by n3
- ▶ For n4, the distances are now :
 $n5=1, n3=1, n2=2, n1=\infty$

All messages have sequence number 1

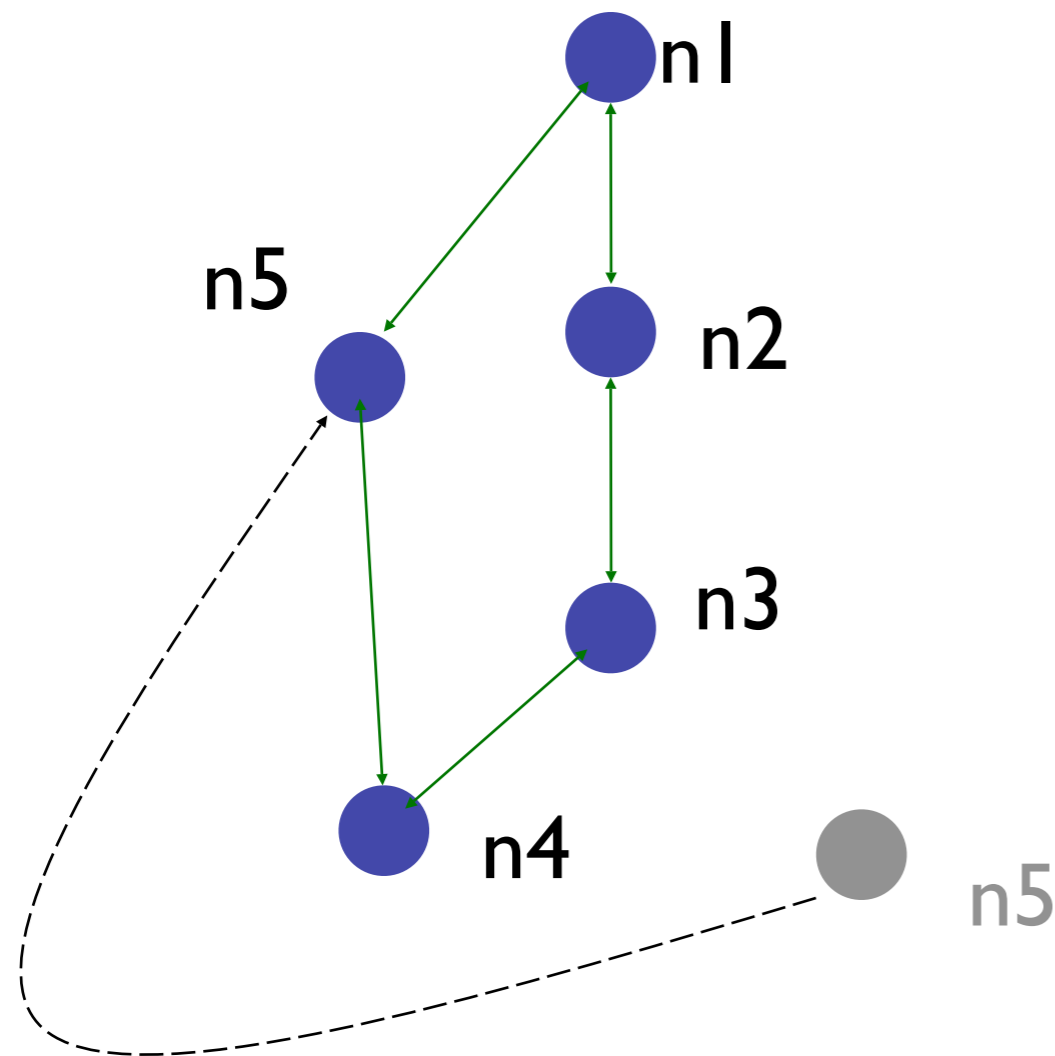
An Example of Route Update

- ▶ Finally, after second round of forwarding, **n4** gets the following distances :

n5=1, n3=1, n2=2, n1=3



An Example of Route Update



- ▶ Suppose **n5** has moved to its new location.
- ▶ Also, **n5** receives a new message from **n1** with a sequence number **2**
- ▶ This message is forwarded by **n5** to **n4**
- ▶ Two distances to **n1** in **n4**
- ▶ Distance **3** with sequence number **1**, and
- ▶ Distance **2** with sequence number **2**
- ▶ Since the latter message has a more recent sequence number, **n4** will update the distance to **n1** as **2**

How good is DSDV?

- ▶ DSDV is an efficient protocol for route discovery. Whenever a route to a new destination is required, it already exists at the source.
- ▶ Hence, latency for route discovery is very low.
- ▶ However, DSDV needs to send a lot of control messages. These messages are important for maintaining the network topology at each node.
- ▶ This may generate high volume of traffic for high-density and highly mobile networks.

Reactive Routing

- In a reactive protocol, a route is discovered only on-demand, when it is necessary.
- These protocols generate much less control traffic at the cost of latency, i.e., it usually takes more time to find a route compared to a proactive protocol.

Protocols:

- Dynamic Source Routing (DSR)
- Ad Hoc On-Demand Distance-Vector (AODV)
- ...

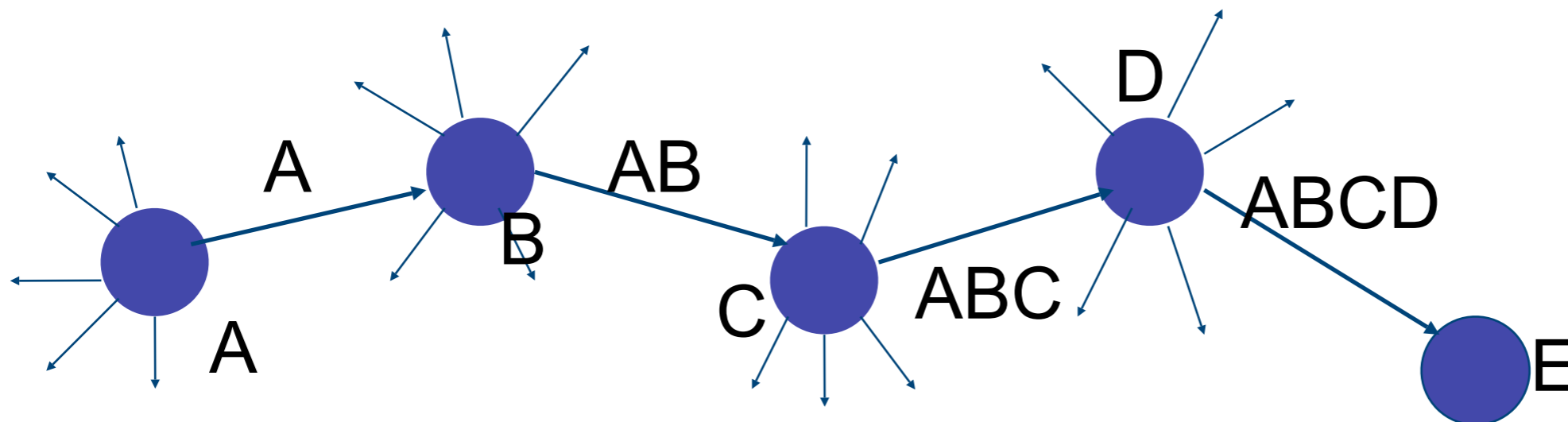
Dynamic Source Routing

- Each node maintains a route cache to remember routes that it has learnt about.
- A node may store multiple routes to a destination in its route cache.
- A node can react to changes in network topology much more rapidly by taking advantage of cached routes.
- For example, if one route to a destination is broken, the source node can choose another route to the destination from its route cache.

- The DSR protocol has two important mechanisms through which the protocol operates.
 - Route Discovery: A node A wishing to send a packet to node E obtains a route to E
 - Route Request
 - Route Reply
 - Route Maintenance: When A is using a discovered route to E, A may detect that the route is broken. In such cases, A may use an alternate route to E (if it is known), or start another route discovery phase to E.

Route Discovery

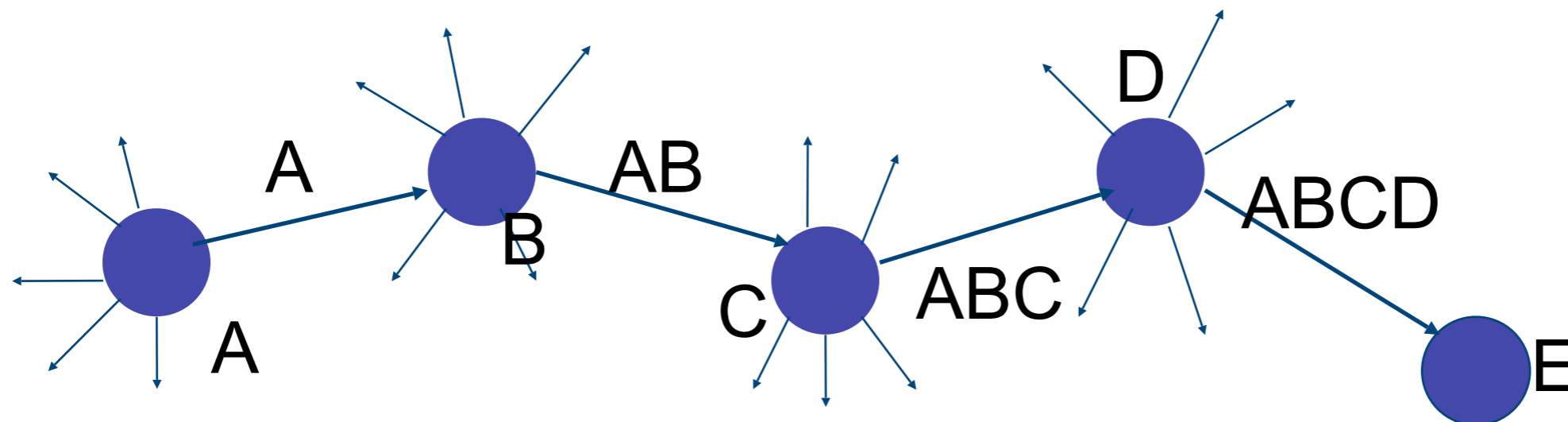
- Node A is trying to discover a route to node E.
- A broadcasts a route request message to its neighbours. This message is received by all nodes within the transmission range of A.
- Each route request message contains the source and target of the route discovery.
- Also, each route request is stamped with an unique ID assigned by the source.



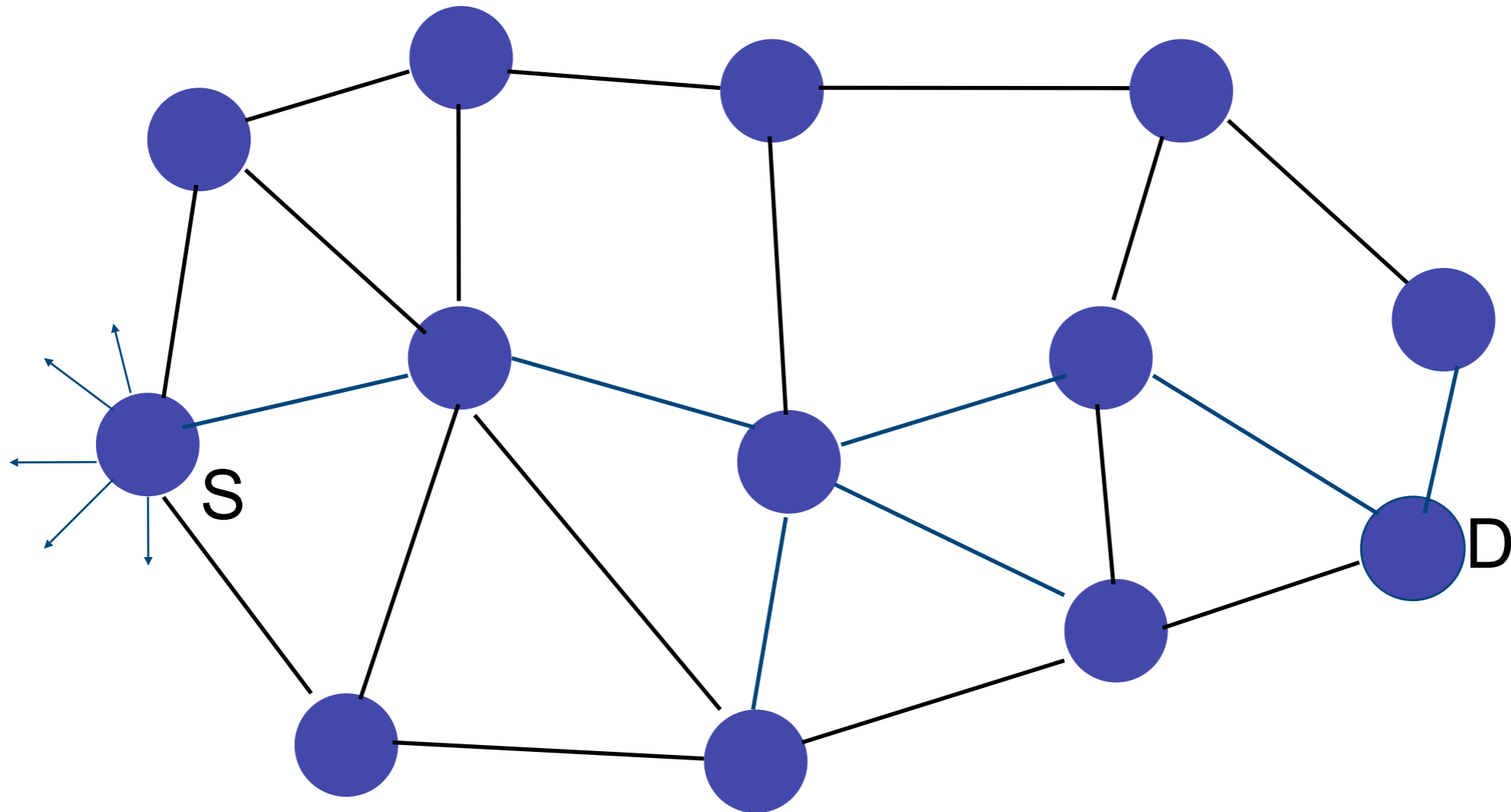
Route Discovery

Every node that receives a route request message, does one of the following:

- Check the unique ID of route request; already received: discard RReq
- A node like C first searches its route cache to see whether it has a stored route to E. If it has such a route, C sends that route to A. (Route Reply)
- If there is no such route in its route cache, C broadcasts the route request message to its neighbours. C attaches its own ID to the route request message

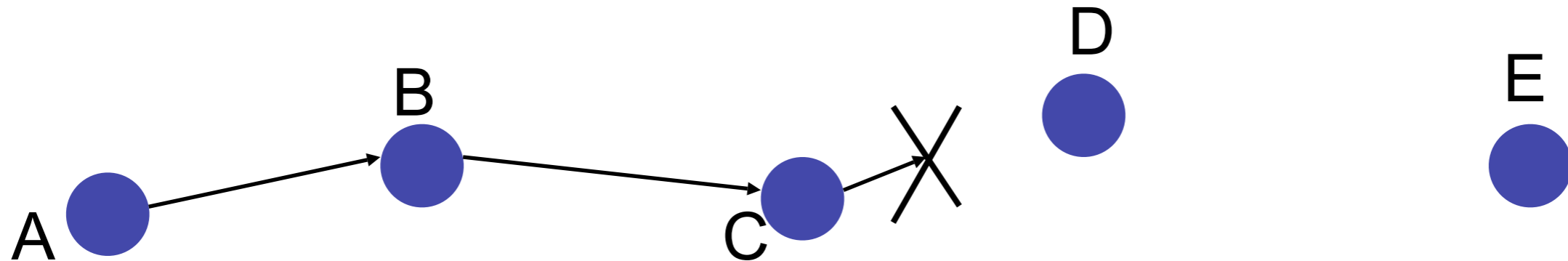


Example



- The DSR protocol has two important mechanisms through which the protocol operates.
 - Route Discovery: A node A wishing to send a packet to node E obtains a route to E
 - Route Request
 - Route Reply
 - Route Maintenance: When A is using a discovered route to E, A may detect that the route is broken. In such cases, A may use an alternate route to E (if it is known), or start another route discovery phase to E.

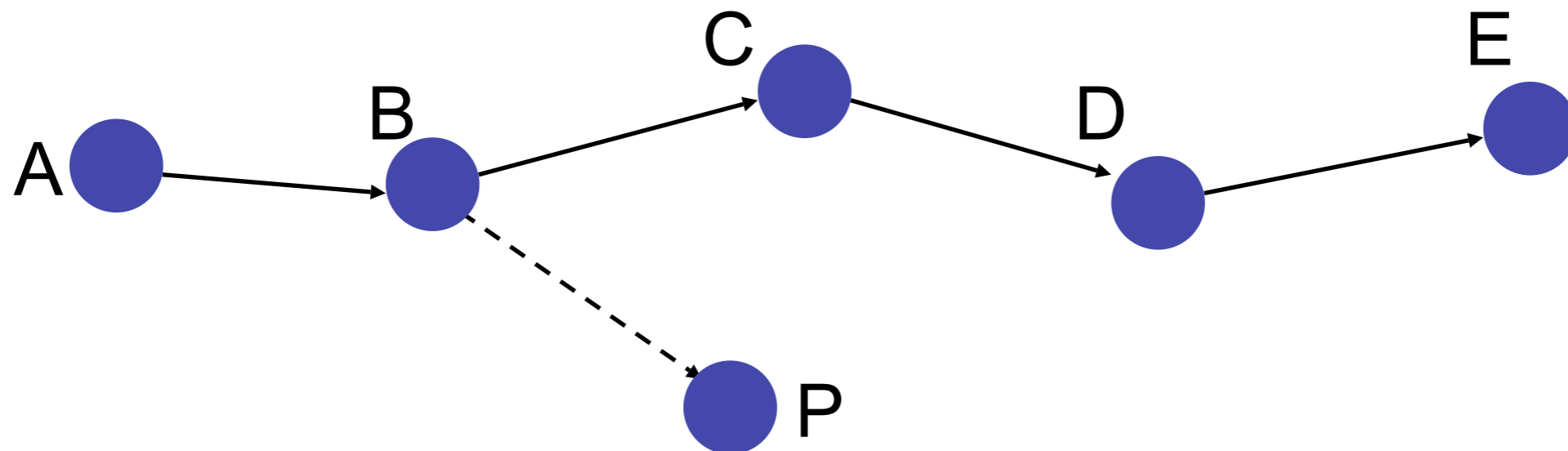
Route Error



- A node like C tries to forward the message and waits for acknowledgment. C will retransmit the message a fixed number of times if no acknowledgment arrives.
- After that, C will initiate a route error message.
- In this example, C will initiate a route error message back to A indicating that the link to D is currently broken.
- A will remove this route from its route cache and try another route to E, if it has one. Or, A may start a new route discovery.

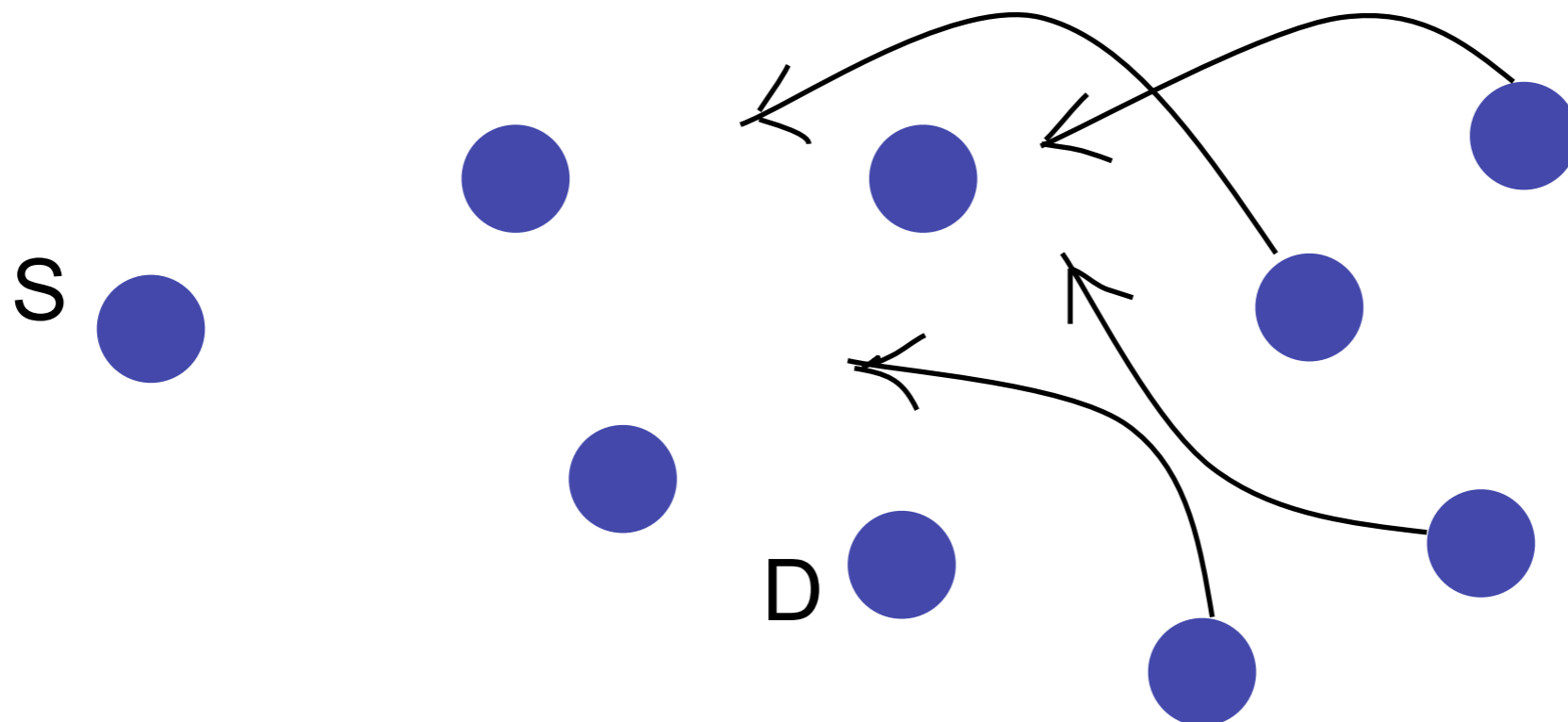
Caching Overheard Routing Information

- DSR extensively takes advantage of existing knowledge of the network topology.
- Each node gathers information about the network topology by overhearing other nodes' transmissions.



Route Request Hop Limit

- Sometime it is not good to propagate a route request message throughout the network.
- In case D is in the neighbourhood of S, the route request message from S should not propagate too far away.
- If D is near S, propagating the route request message too far will result in too many unnecessary route reply messages in future.



Restricted Propagation of Route Request

- A better strategy is to propagate route request messages with increasing hop count.
- Initially, send the route request to a distance of 2 hops. If no route reply is received after sometime, send the route request to a distance of 4 hops and so on.
- This reduces network congestion by reducing the number of route reply messages.

Non-uniform Packet Size in DSR

- When a source node A sends a packet to a destination node E, A should send the entire route to E along with the packet.
- This is necessary for the intermediate nodes to forward the packet.
- Usually all media support packets of uniform size. If a packet is large, it has to be split into smaller packets.
- This may cause problems in the wireless medium as packets that are split into smaller parts may not arrive in correct order.
- Intermediate nodes may not be able to forward packets correctly.

Table-Driven

- Routing Tables (for saving information about topology)

On-Demand

- Route Discovery
 - Expanding Ring Search (Route Request Type)
 - Forward Path Setup (Saving Route Information in R' Tables)
- Route Maintenance (Methods to repair broken links)

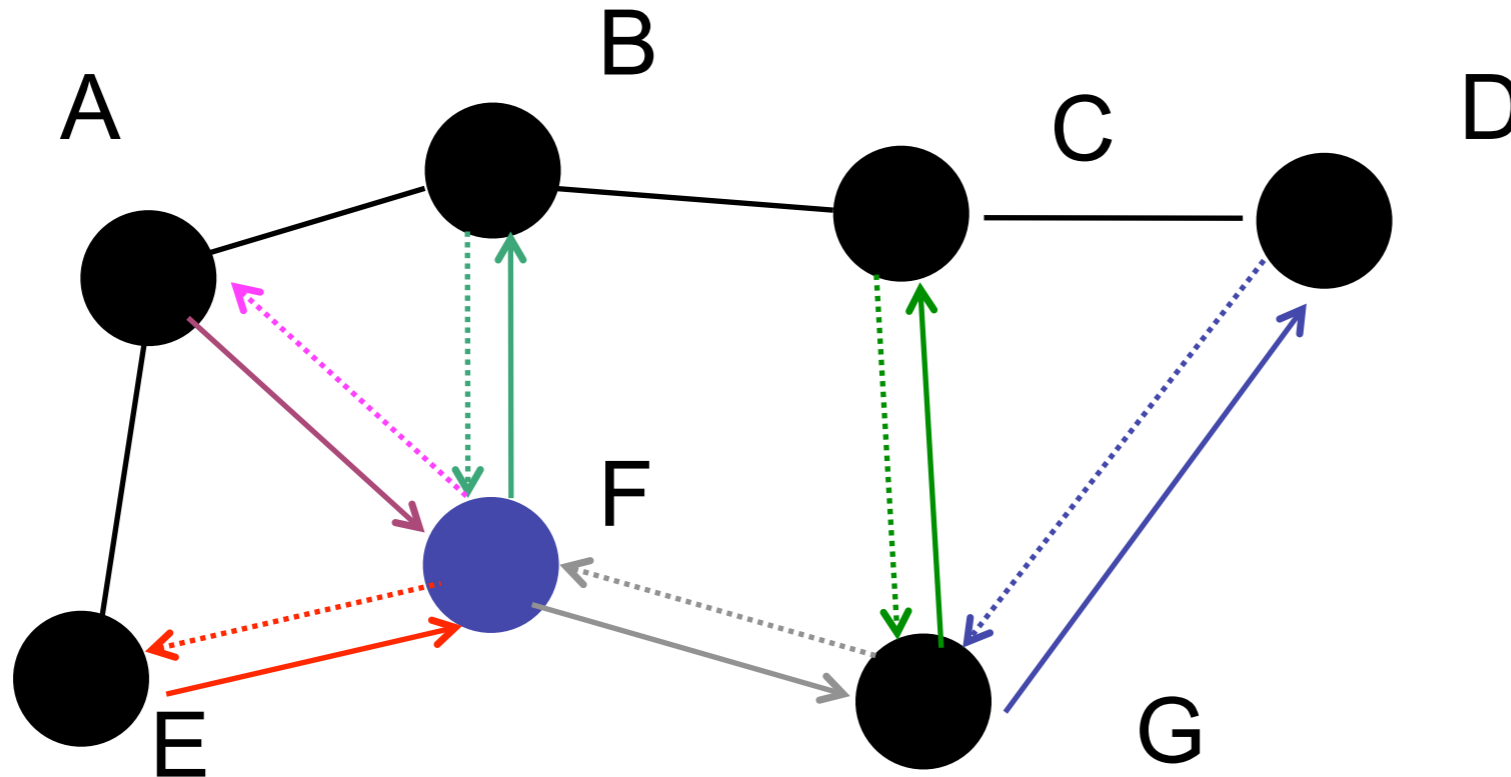
Entries:

- Destination IP
- Next Hop IP
- Destination Sequence Number
- Life Time
- List of Precursors
- Hop Count Number

Sequence Number:

The Seq. Number is monotonically increased each time the node learns of a change in the topology.

Example (Routing Table)



Routing Table of Node F:

Dest. IP	Next Hop IP	Dest. SeqNo.	Lifetime	Precursors	Hop Count
B	B	2	10	E	1
C	G	4	6	E	2
D	G	4	8	A, E	2

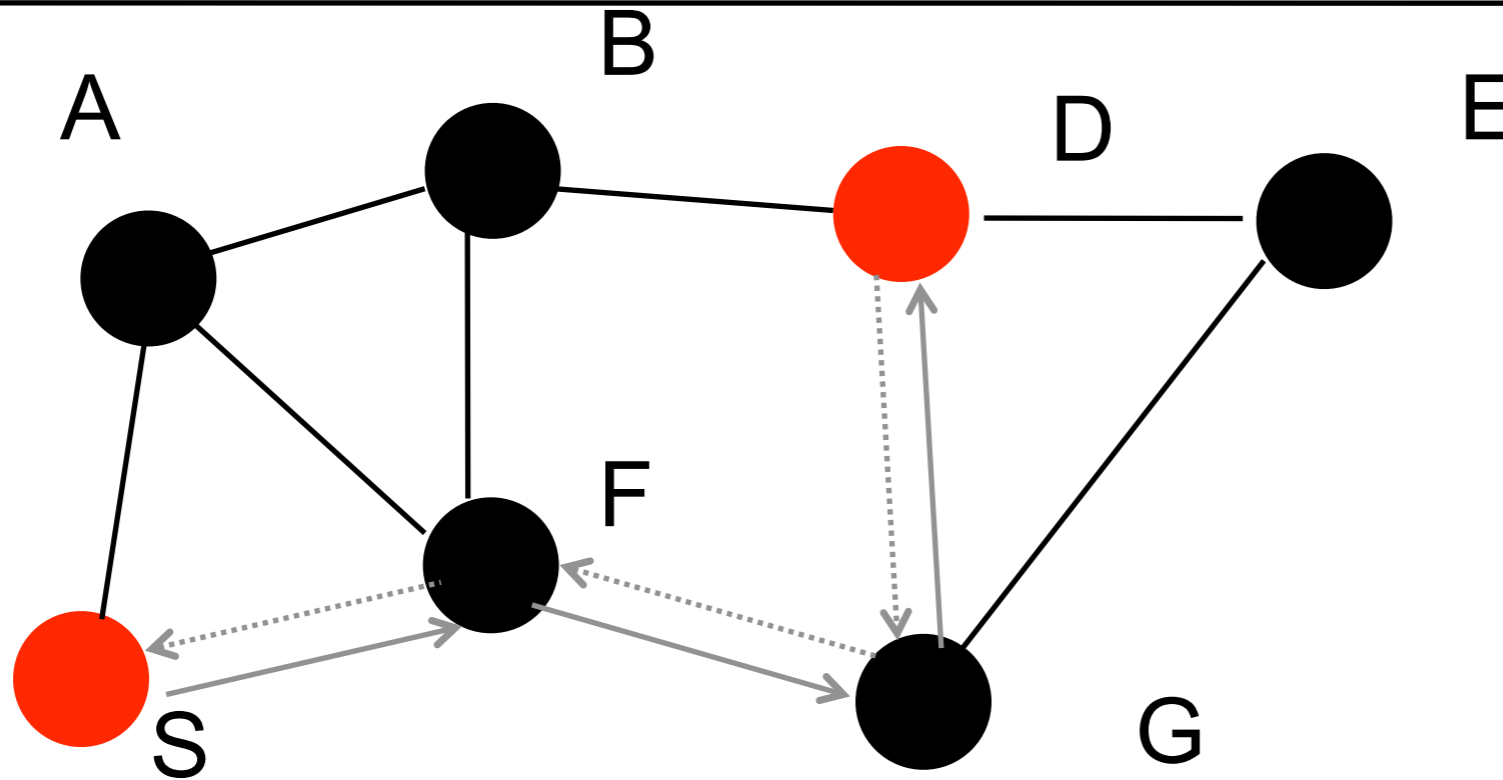
Like DSR, we use two types of messages, route request (RREQ) and route reply (RREP):

- Route Request Messages (RREQ)
- Route Request Processing (RREQ)
 - Reverse Route Entry
- Expanding Ring Search (RREQ)
- Responding to Route Request Messages (RREP)
 - Forward Path Setup

Route Request Message

- When node S wants to send a message to node D, S searches its routing table for a valid route to D.
- If there is no valid route, S initiates a RREQ message with the following components :
 - The IP addresses of S and D
 - The current sequence number of S and the last known sequence number of D
 - A broadcast ID from S. This broadcast ID is incremented each time S initiates a RREQ message.
 - Hop count
- The <broadcast ID, IP address> pair of the source S forms a unique identifier for the RREQ.

Example (Routing Request Message)



Routing Table of Node S:

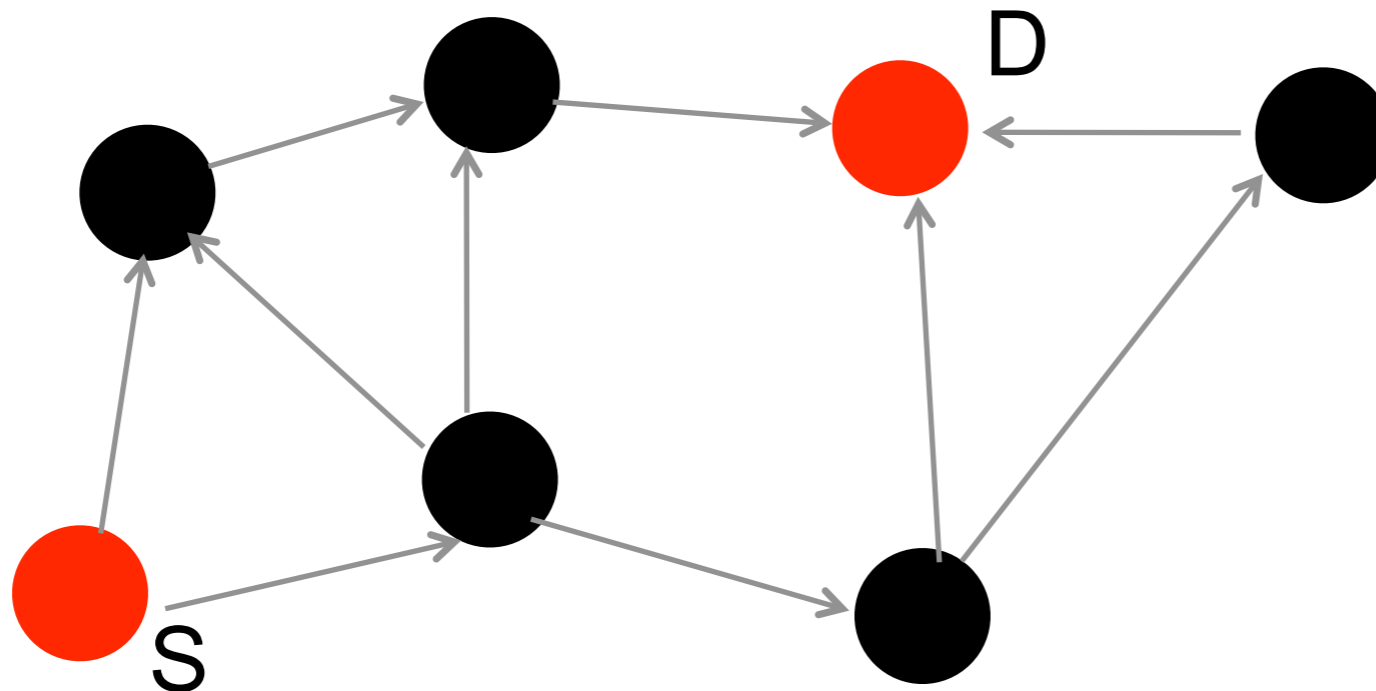
Dest. IP	Next Hop IP	Dest. SeqNo.	Lifetime	Precursors	Hop Count
D	F	4	0		3

Routing Request Message:

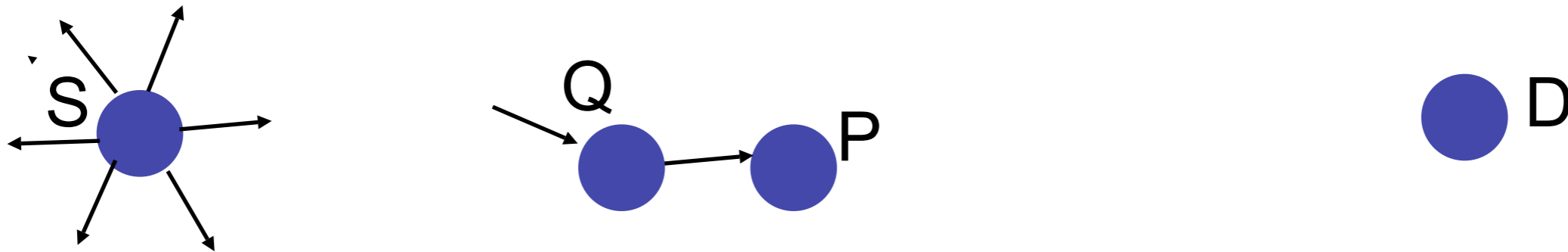
Source IP	Dest. IP	Source Seq. No.	Dest. Seq. No.	Broadcast ID	Hop Count
S	D	7	4	15	0 73

Route Request Processing

- Suppose a node P receives the RREQ from S. P first checks whether it has received this RREQ before.
- Each node stores the $\langle \text{broadcast ID, IPaddress} \rangle$ pairs for all the recent RREQs it has received. (for a specific amount of time)

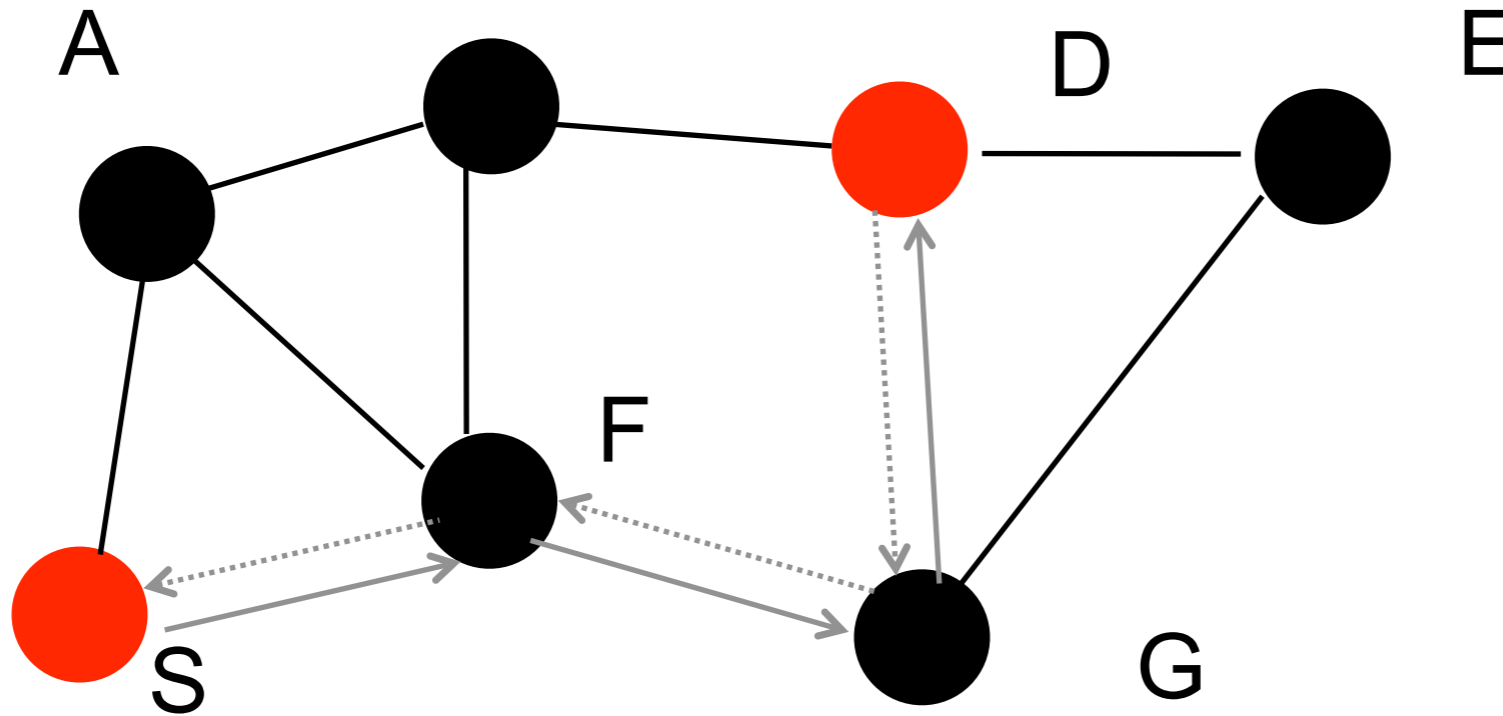


Processing a RREQ Message



- If P has seen this RREQ from S already, P discards the RREQ. Otherwise, P processes the RREQ :
 - P sets up a reverse route entry in its routing table for the source S.
 - This entry contains the IP address and current sequence number of S, number of hops to S and the address of the neighbour from whom P got the RREQ.

Reverse Route Entry



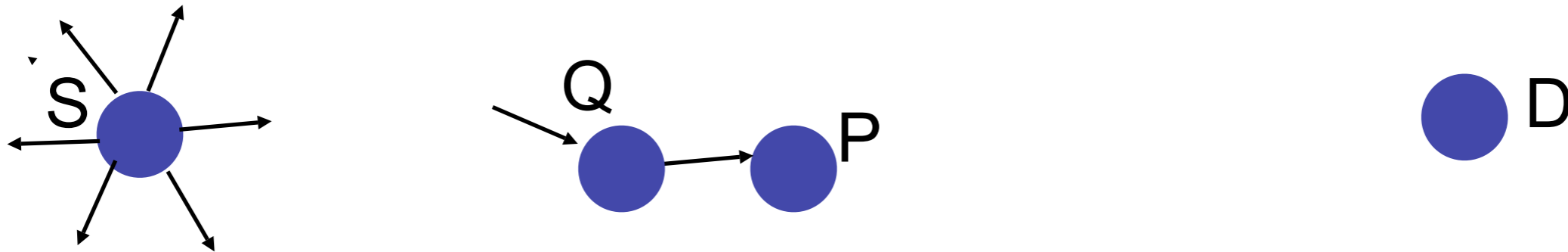
Routing Table of Node G:

Dest. IP	Next Hop IP	Dest. SeqNo.	Lifetime	Precursors	Hop Count
S	F	7	10		2

Routing Request Message:

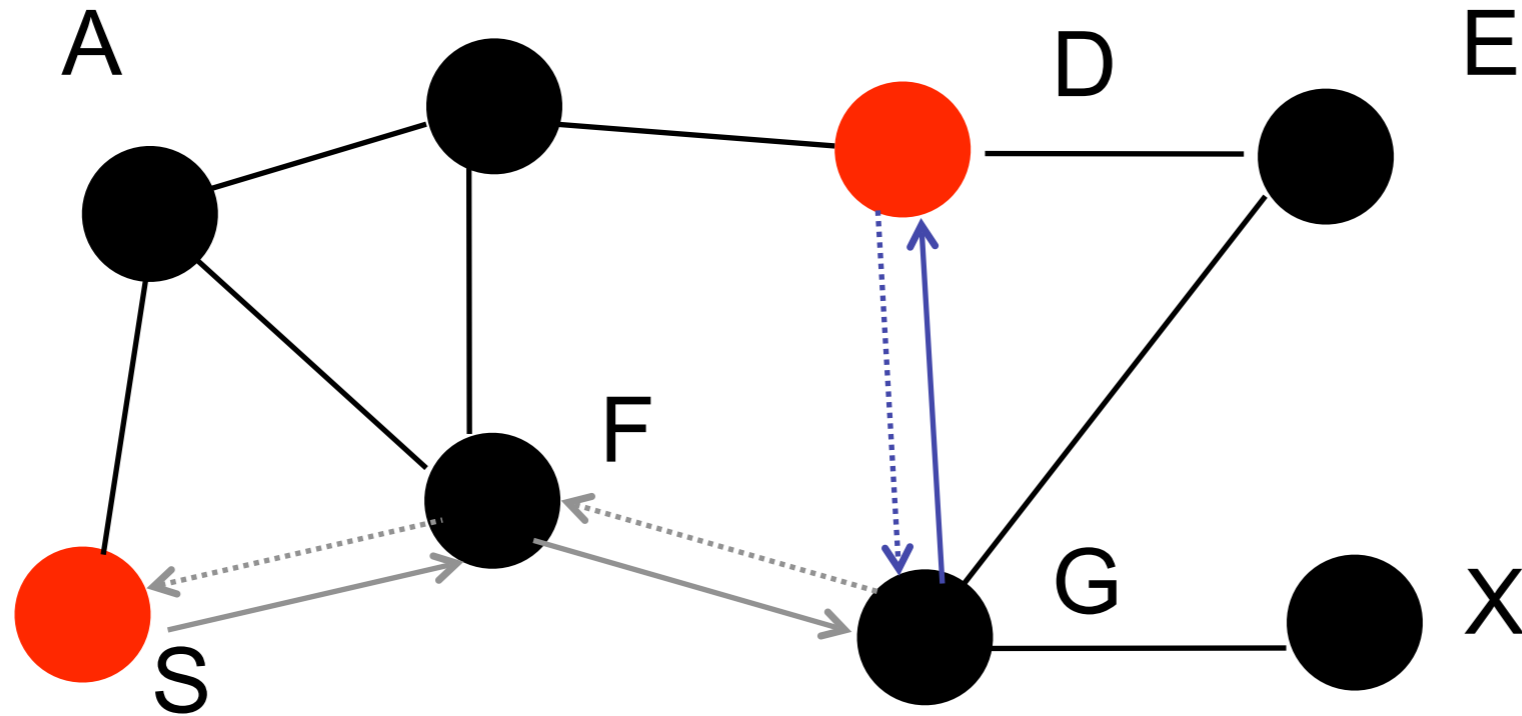
Source IP	Dest. IP	Source Seq. No.	Dest. Seq. No.	Broadcast ID	Hop count
S	D	7	4	15	2 76

Responding to a RREQ Message



- P can respond to the RREQ from S if P has an unexpired entry for D in its routing table.
- Moreover, the sequence number from D that P has, must not be less than the sequence number of D that was in the RREQ from S.
- This ensures that there is no loop in the route.
- If P satisfies both of these requirements, it sends a RREP message back to S.
- If P cannot reply to the RREQ from S,
- P increments the hop-count of the RREQ and broadcasts it to its neighbours.

Responding to a RREQ Message



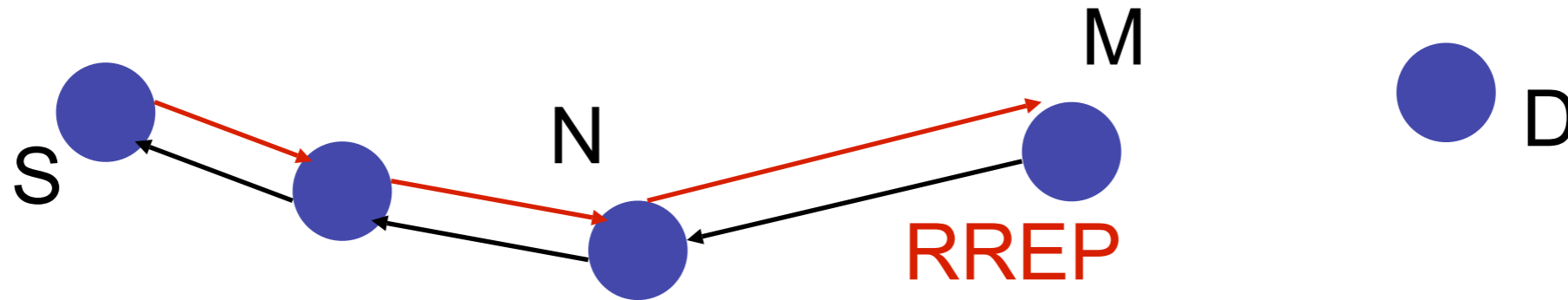
Routing Table of Node G:

Dest. IP	Next Hop IP	Dest. SeqNo.	Lifetime	Precursors	Hop Count
D	D	6	10	X	1

Routing Request Message:

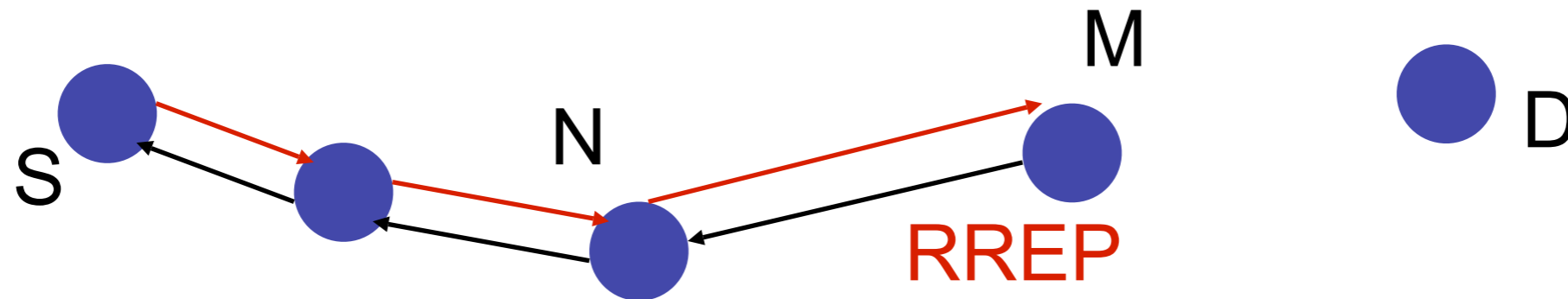
Source IP	Dest. IP	Source Seq. No.	Dest. Seq. No.	Broadcast ID	Hop Count
S	D	7	4	15	2 78

Forward Path Setup (Sending a RREP)



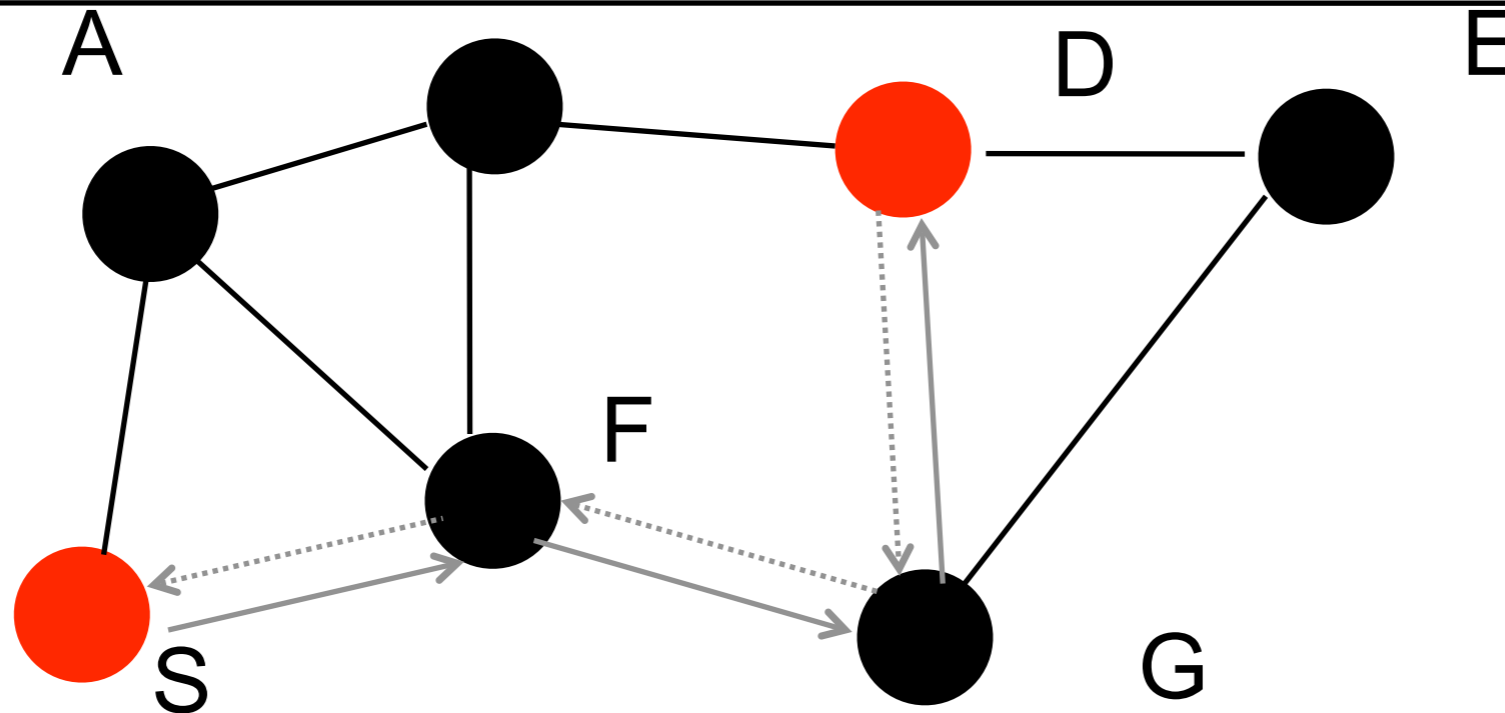
- A RREP message has several fields :
 - The IP address of both source and destination
 - If the destination is sending the RREP, it sends its current sequence number, a lifetime for the route and sets the hop-count to 0
 - If an intermediate node is responding, it sends the last known sequence number from the destination, sets the hop-count equal to distance from the destination and a lifetime for the route.

Forward Path Setup



- A node (here Node M) sends a RREP back to a neighbour from whom it received the RREQ.
- When an intermediate node (here Node N) receives a RREP, it sets up a forward path to the destination in its route table.
- This contains the IP addresses of the neighbour and the destination, hop-count to the destination and a lifetime for the route.

Forward Path Setup



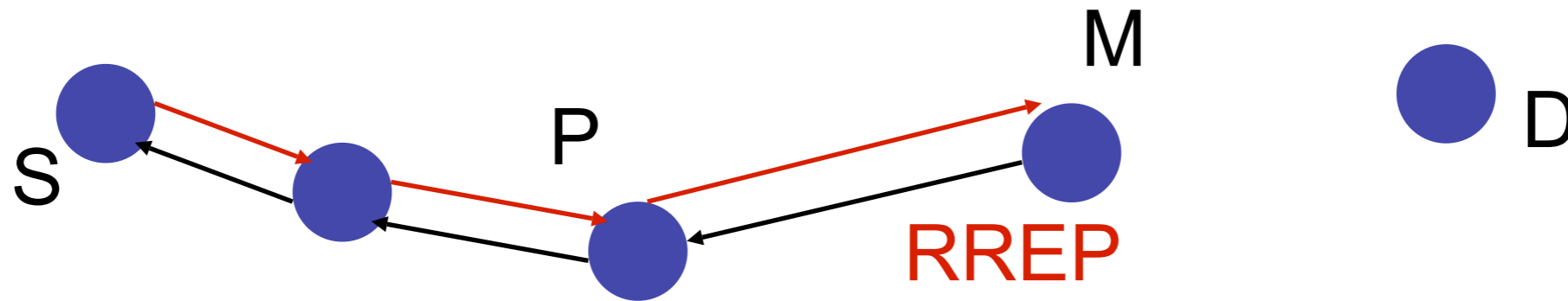
Routing Table of Node F:

Dest. IP	Next Hop IP	Dest.Seq. No	Lifetime	Precursors	Hop Count
S	S	7	10	G	1
D	G	6	4	S	2

Routing Reply Message:

Source IP	Dest. IP	Source Seq. No.	Dest. Seq. No.	Lifetime	Hop Count
S	D	7	6	4	1 82

Handling more than one RREP

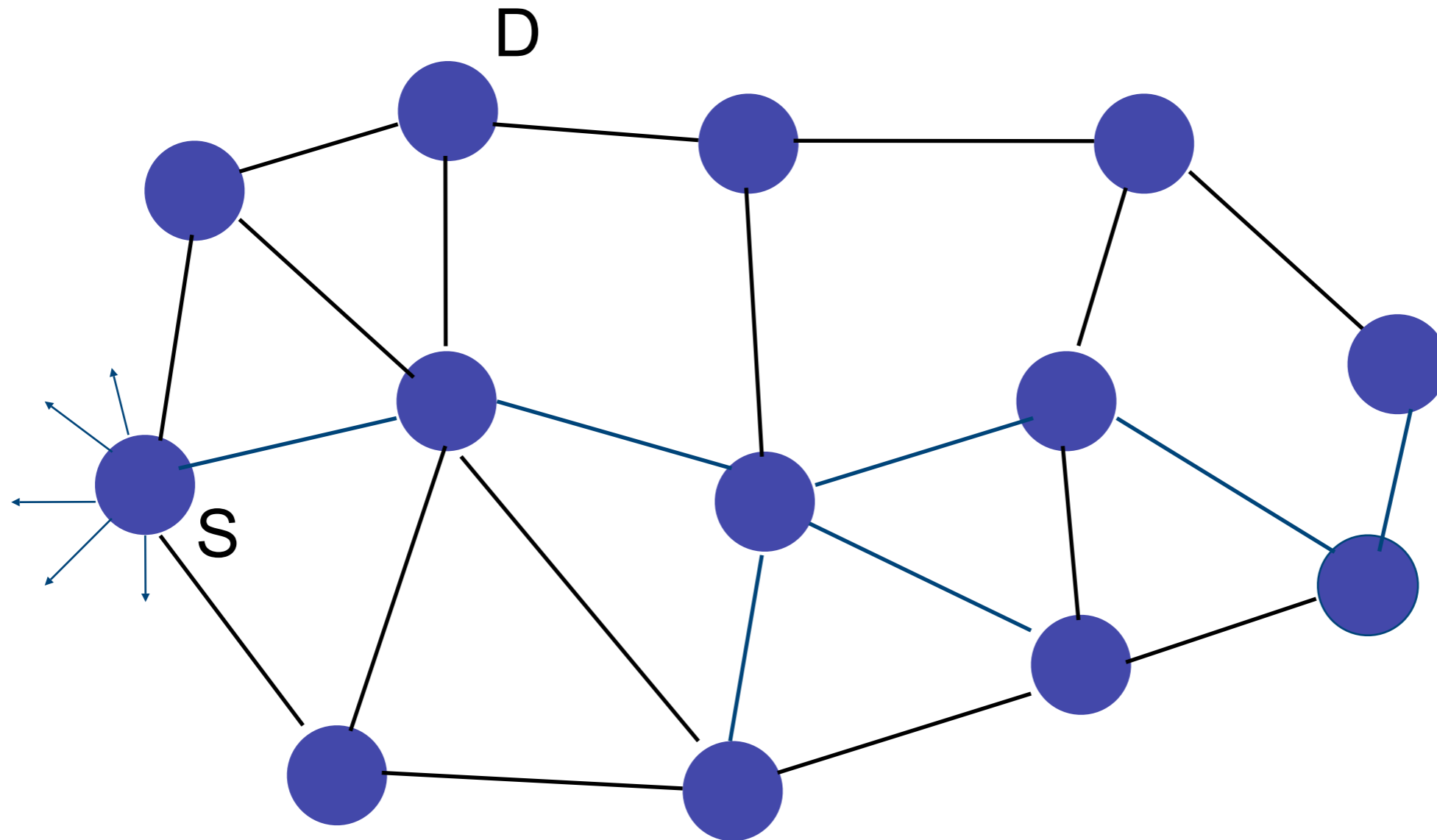


- An intermediate node P may receive more than one RREP for the same RREQ.
- P forwards the first RREP it receives and forwards a second RREP later only if :
 - The later RREP contains a greater sequence number for the destination, (RREP for later RREQ)
 - The hop-count to the destination is smaller in the later RREP
 - Otherwise, it does not forward the later RREPs. This reduces the number of RREPs propagating towards the source.

Expanding Ring Search

- For route discovery, a source node broadcasts a RREQ across the network. This may create a lot of messages in a large network.
- A source node uses an expanding ring search strategy. With a ring diameter K , a RREQ dies after its hop-count exceeds K .
- If a RREQ fails, the source node increases the value of K incrementally.

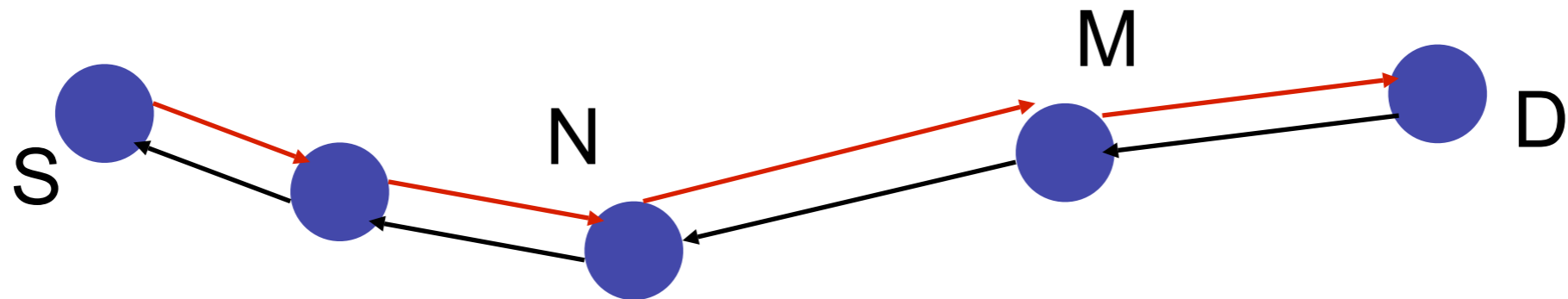
Example (Expanding Ring Search)



Routing Request Message:

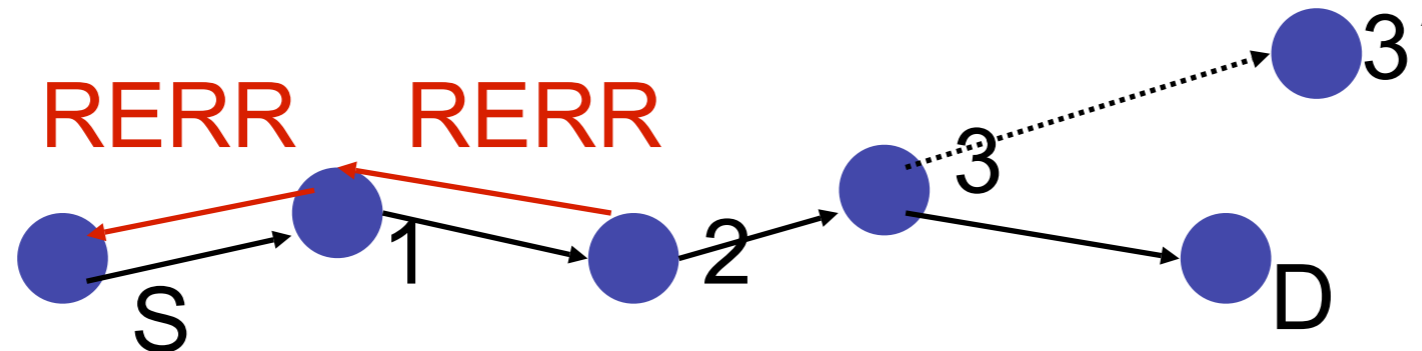
Source IP	Dest. IP	Source Seq. No.	Dest. Seq. No.	Broadcast Id	Hop Count	Time To Live
S	D	5	4	3	0	185

Route Maintenance



- Once a route has been established between two nodes S and D, it is maintained as long as S (source node) needs the route.
- If S moves during an active session, it can reinitiate route discovery to establish a new route to D.
- When D or an intermediate node moves, a route error (RERR) message is sent to S.

Route Error

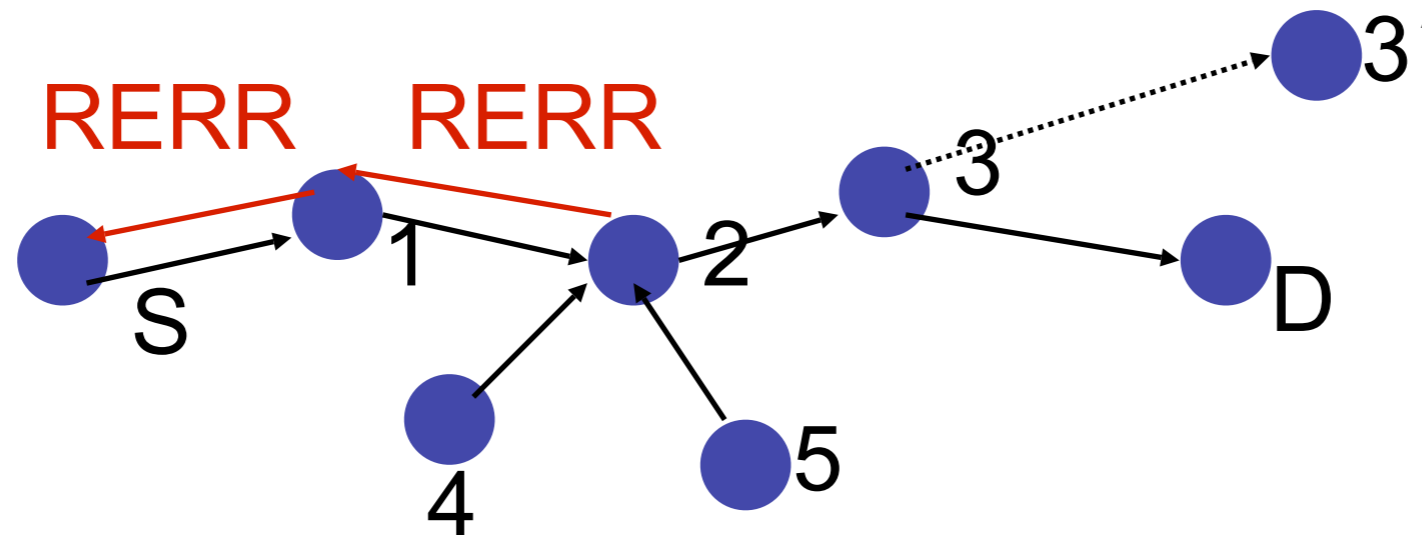


- If S moves during an active session, it can reinitiate route discovery to establish a new route to D.
- When D or an intermediate node moves, a route error (RERR) message is sent to S.

Example:

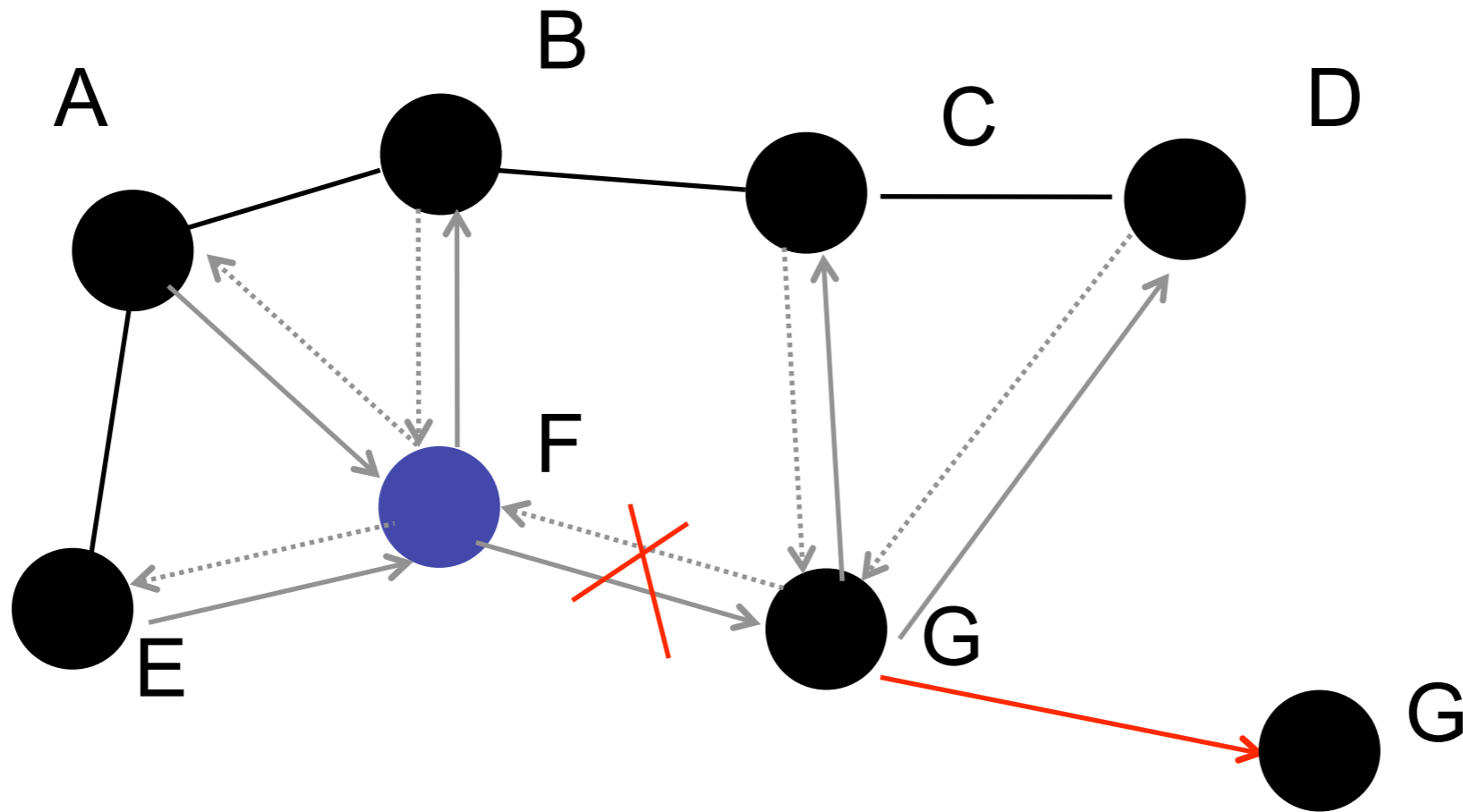
- The link from node 3 to D is broken as 3 has moved away to a position 3'.
- Node 2 sends a RERR message to 1 and 1 sends the message in turn to S.
- S initiates a route discovery if it still needs the route to D.

Updating Routing Tables



- Suppose neighbours 4 and 5 route through 2 to reach D. Node 2 broadcasts RERR to all such neighbours.
- Each neighbour marks its route table entry to D as invalid by setting the distance to infinity.
- Each neighbour in turn propagates the RERR message.

Example (Routing Error)



Routing Table of Node F:

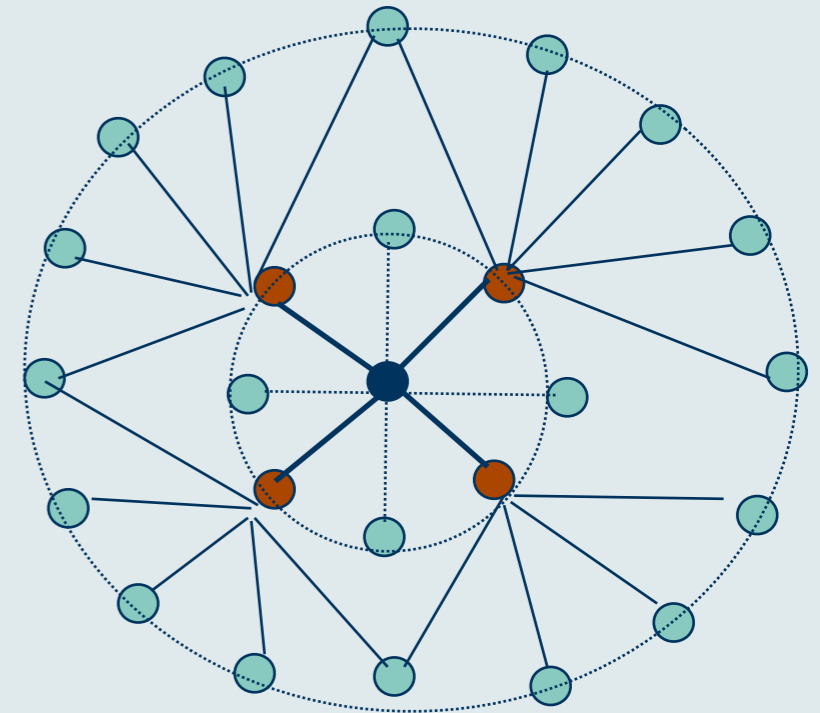
Dest. IP	Next Hop IP	Dest. SeqNo.	Lifetime	Precursors	Hop Count
B	B	2	10	E	1
C	G	4	6	E	2
D	G	4	8	A, E	2

- AODV does not retransmit data packets that are lost and hence does not guarantee packet delivery.
- However, the packet delivery percentage is close to 100 with relatively small number of nodes.
- The packet delivery percentage drops with increased mobility.
- The overhead packets in AODV are due to RREQ, RREP and RERR messages.
- AODV needs much less number of overhead packets compared to DSDV.
- The number of overhead packets increases with increased mobility, since this gives rise to frequent link breaks and route discovery.
- The route discovery latency in AODV is low compared to DSR and DSDV.

Overview

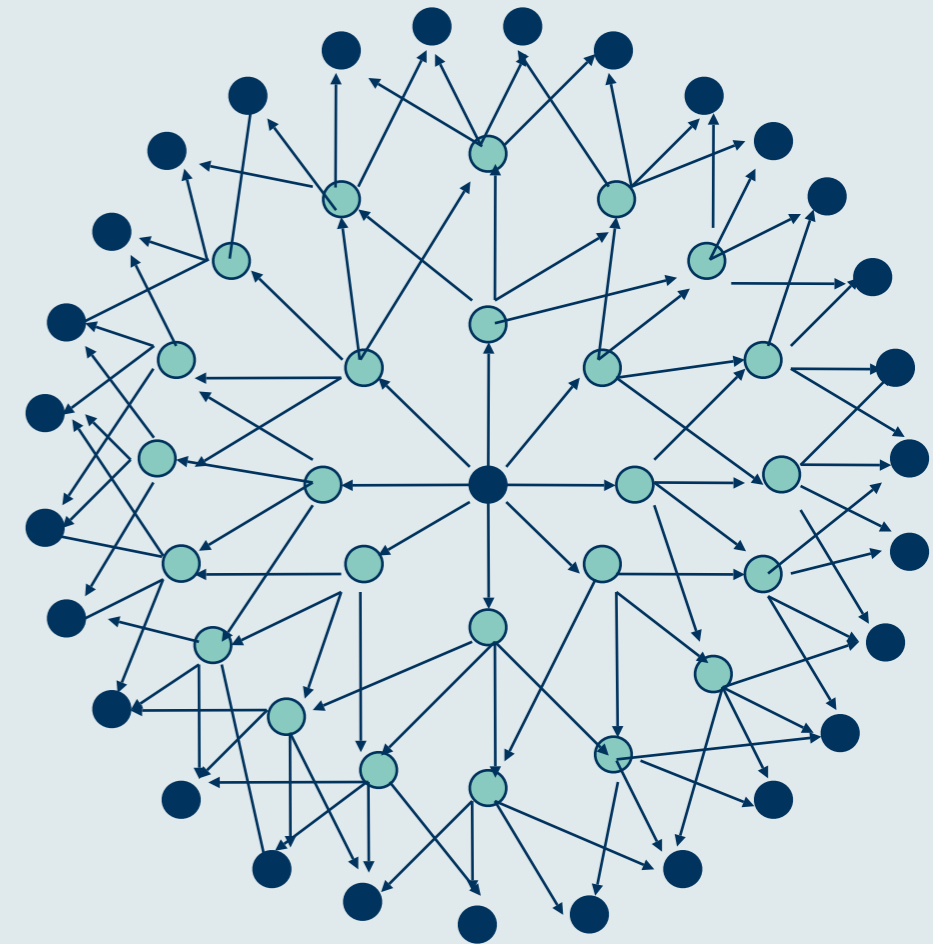
□ OLSR

- Developed by IETF
- Table driven
- Inherits Stability of Link-state protocol
- Selective Flooding
- Periodic Link State Information generated only by MPR
- MPRs employed for optimization



Link State Routing (eg, OSPF)

- Each node periodically floods status of its links
- Each node re-broadcasts link state information received from its neighbour
- Each node keeps track of link state information received from other nodes
- Each node uses above information to determine next hop to each destination



24 retransmissions to diffuse a message up to 3 hops

● Retransmission node

OLSR Overview

□ In LSR

- protocol a lot of control messages unnecessary duplicated

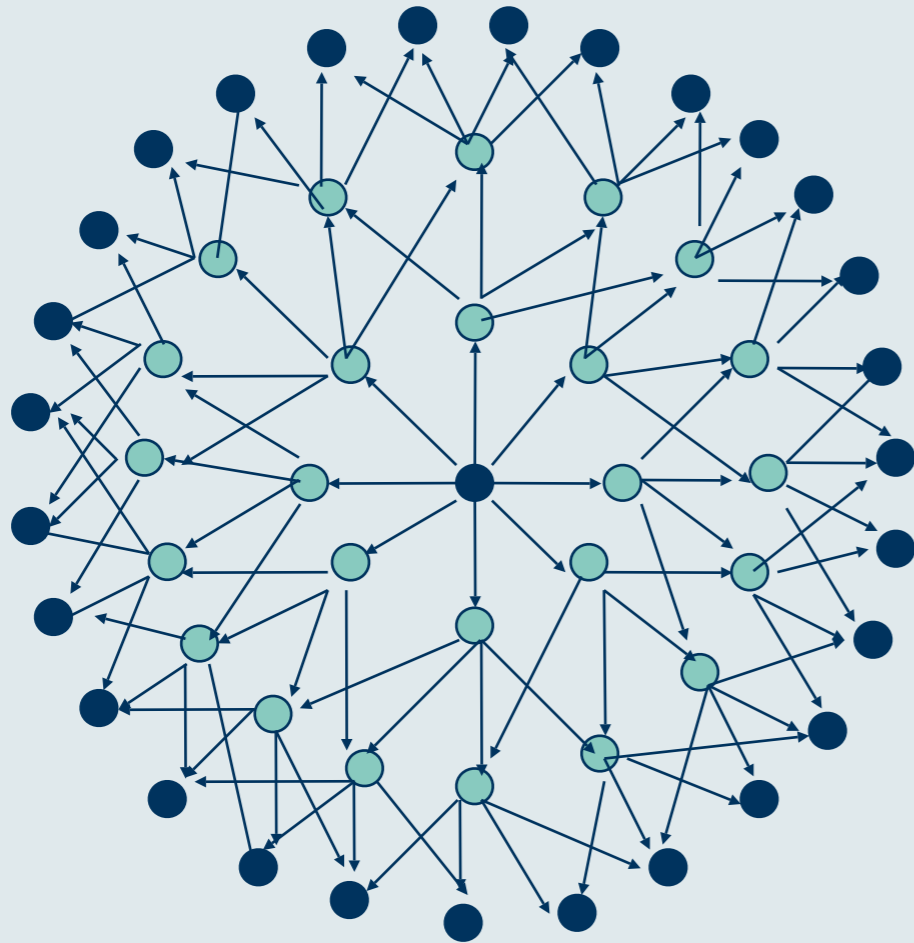
□ In OLSR

- **only** MPR retransmit control messages:
 - **Reduce size of control message;**
 - **Minimize flooding**

□ Other advantages (the same as for LSR):

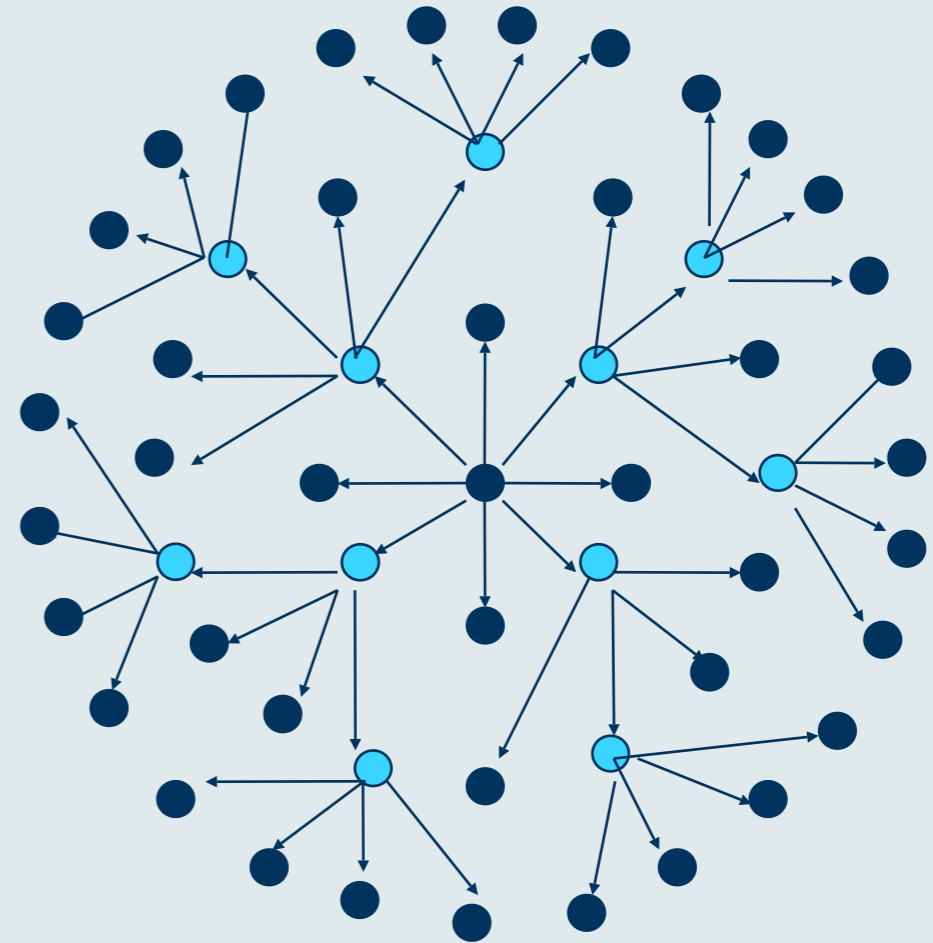
- As **stable** as LSR protocol;
- **Proactive** protocol(routes already known);
- Does not depend upon any **central entity**;
- **Tolerates** loss of control messages;
- Supports nodes **mobility**.
- Good for **dense network**

Optimized Link state routing (OLSR)



24 retransmissions to diffuse a message up to 3 hops

● Retransmission node

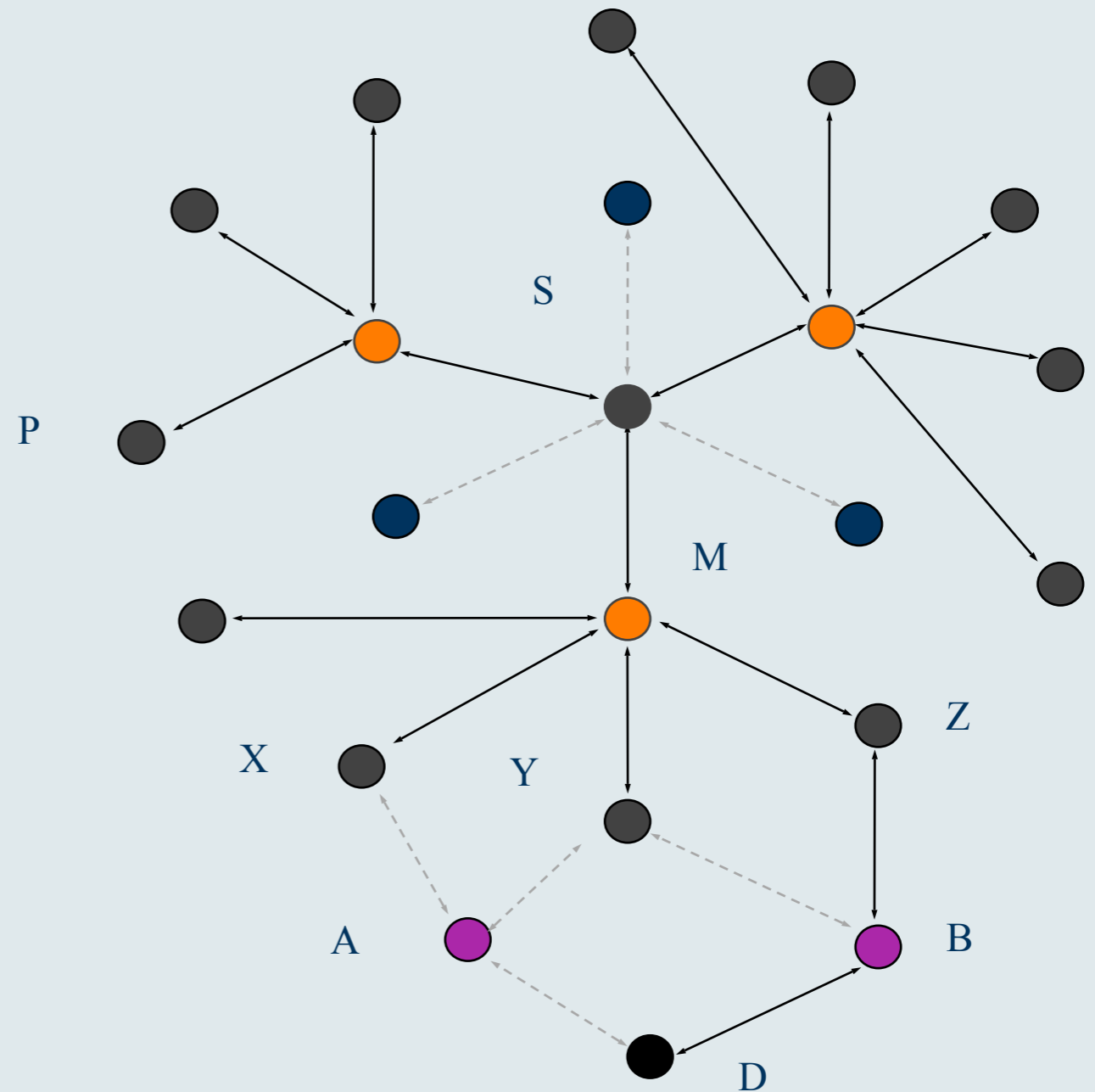


11 retransmission to diffuse a message up to 3 hops

● Retransmission node

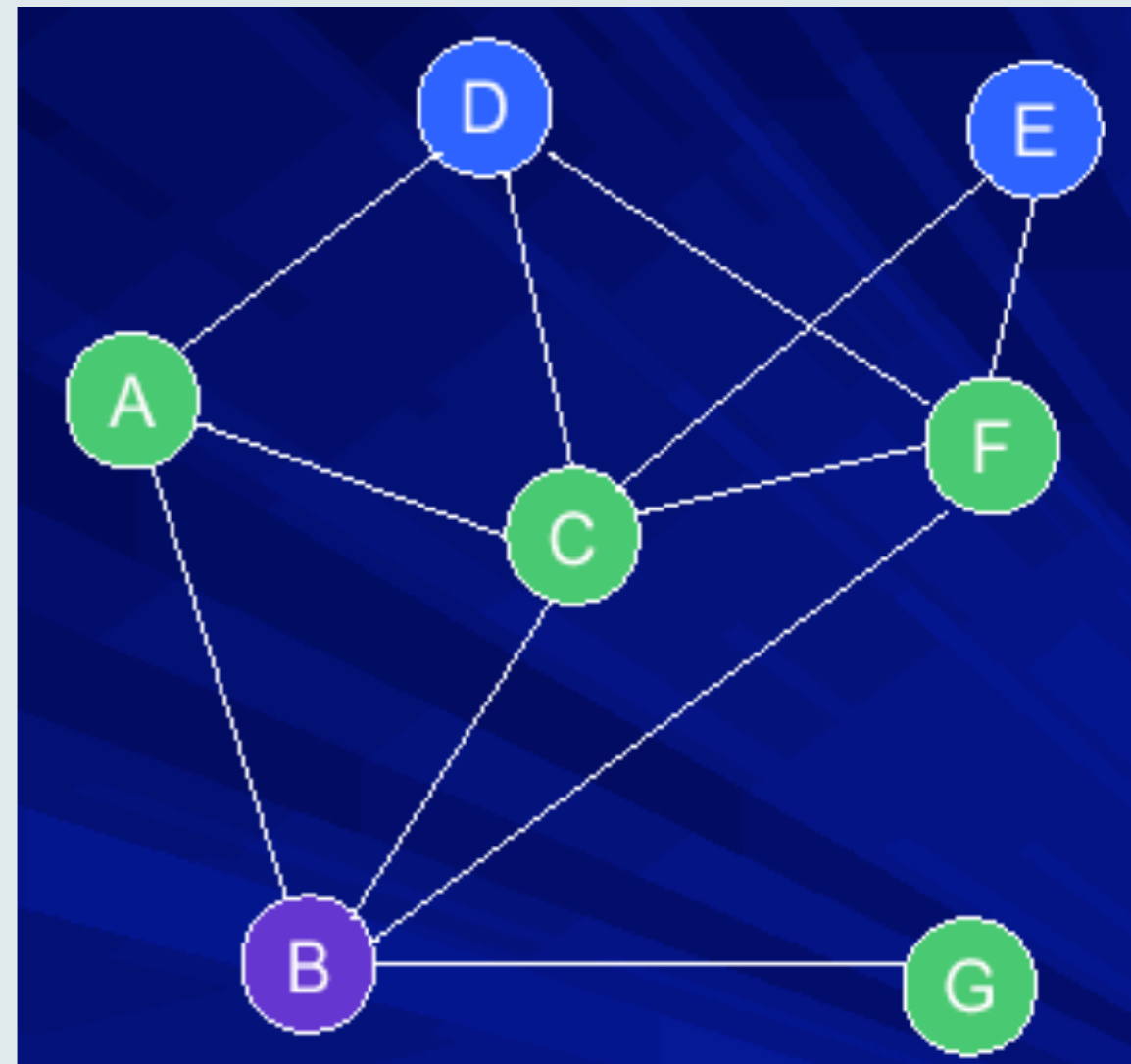
Description of OLSR

- MPR (Multipoint relays)
- MPR selector
- Symmetric 1-hop neighbours
- Symmetric strict 2-hop neighbours



Neighbor sensing

- Each node periodically broadcasts Hello message:
 - List of neighbors with bi-directional link
 - List of other known neighbors.
- Hello messages permit each node to learn topology up to 2 hops
- Based on Hello messages each node selects its set of MPR's



Example of neighbor table

One-hop neighbors

Neighbor's id	State of Link
B	Bidirectional
G	Unidirectional
C	MPR
...	...

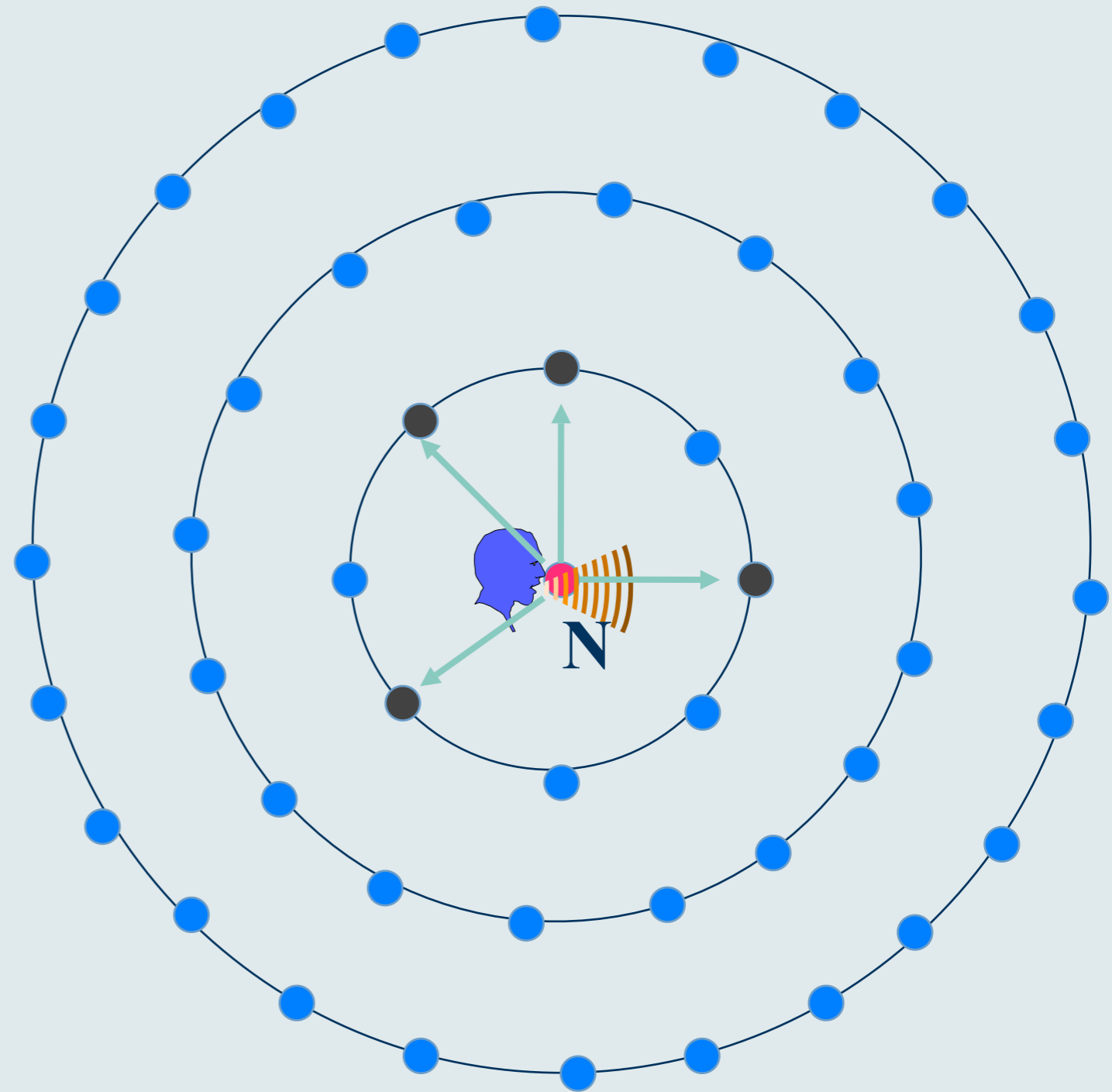
Two-hop neighbors

Neighbor's id	Access through
E	C
D	C
...	...

Also every entry in the table has a timestamp, after which the entry is not valid

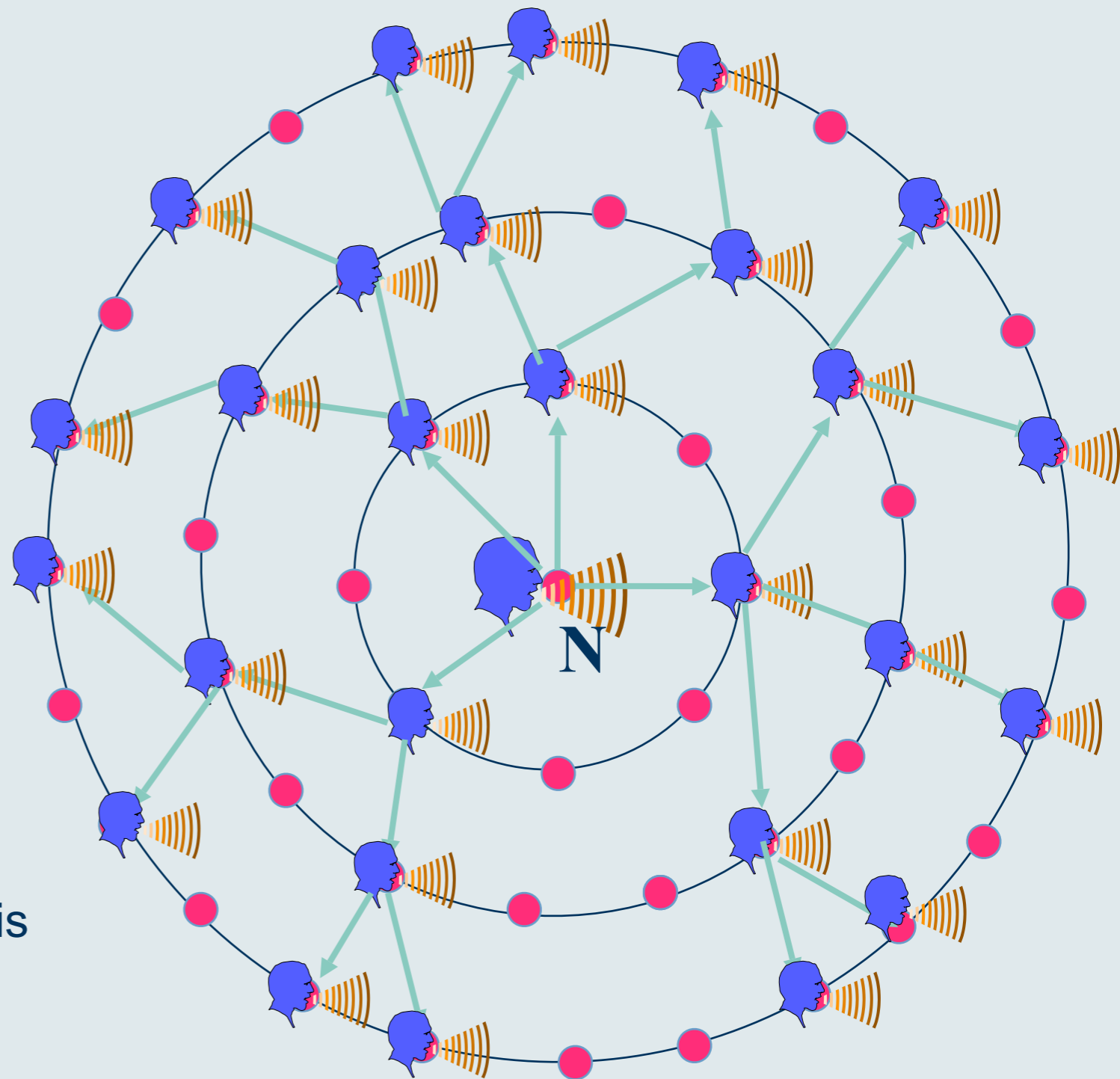
Multipoint Relays (MPR)

- **Reduce** re-transmission in the same region
- Each node select a set of ***MPR Selectors***
- MPR Selectors of node **N - MPR(N)**
 - one-hop neighbors of N



Multipoint Relays (MPR)

- **Reduce** re-transmission in the same region
- Each node select a set of **MPR Selectors**
- MPR Selectors of node **N - MPR(N)**
 - one-hop neighbors of N
- MPR set of Node N
- Set of MPR's is able to transmit to all two-hop neighbors
- Link between node and it's MPR is **bidirectional**.



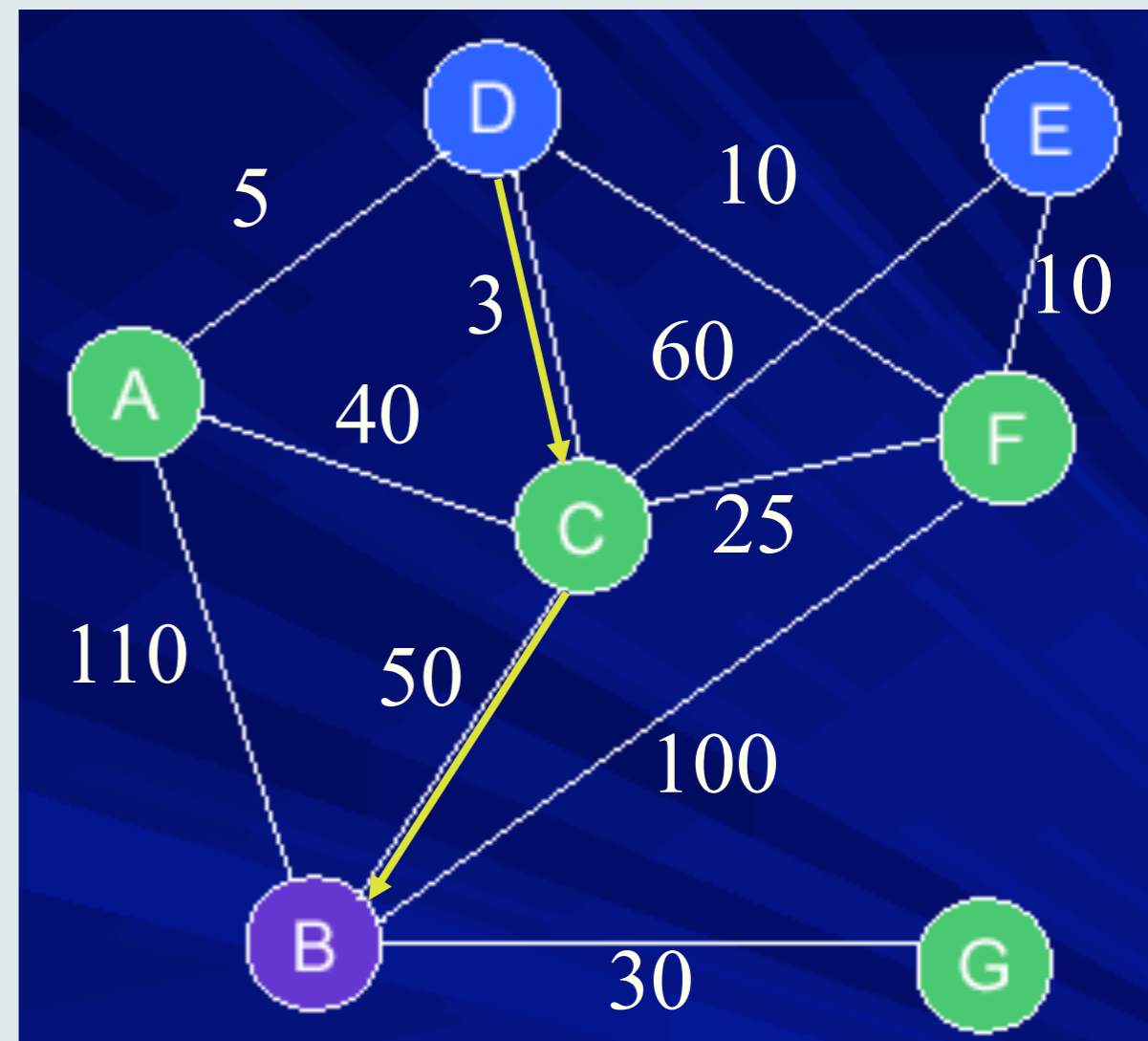
Multipoint Relays (MPR)

- ❑ Every node keeps a table of routes to all known destination through its MPR nodes
- ❑ Every node periodically broadcasts list of its MPR Selectors (instead of the whole list of neighbors).
- ❑ Upon receipt of MPR information each node recalculates and updates routes to each known destination

MRP selection in OLSR

Node	1 Hop Neighbors	2 Hop Neighbors	MPR(s)
B	A,C,F,G	D,E	C

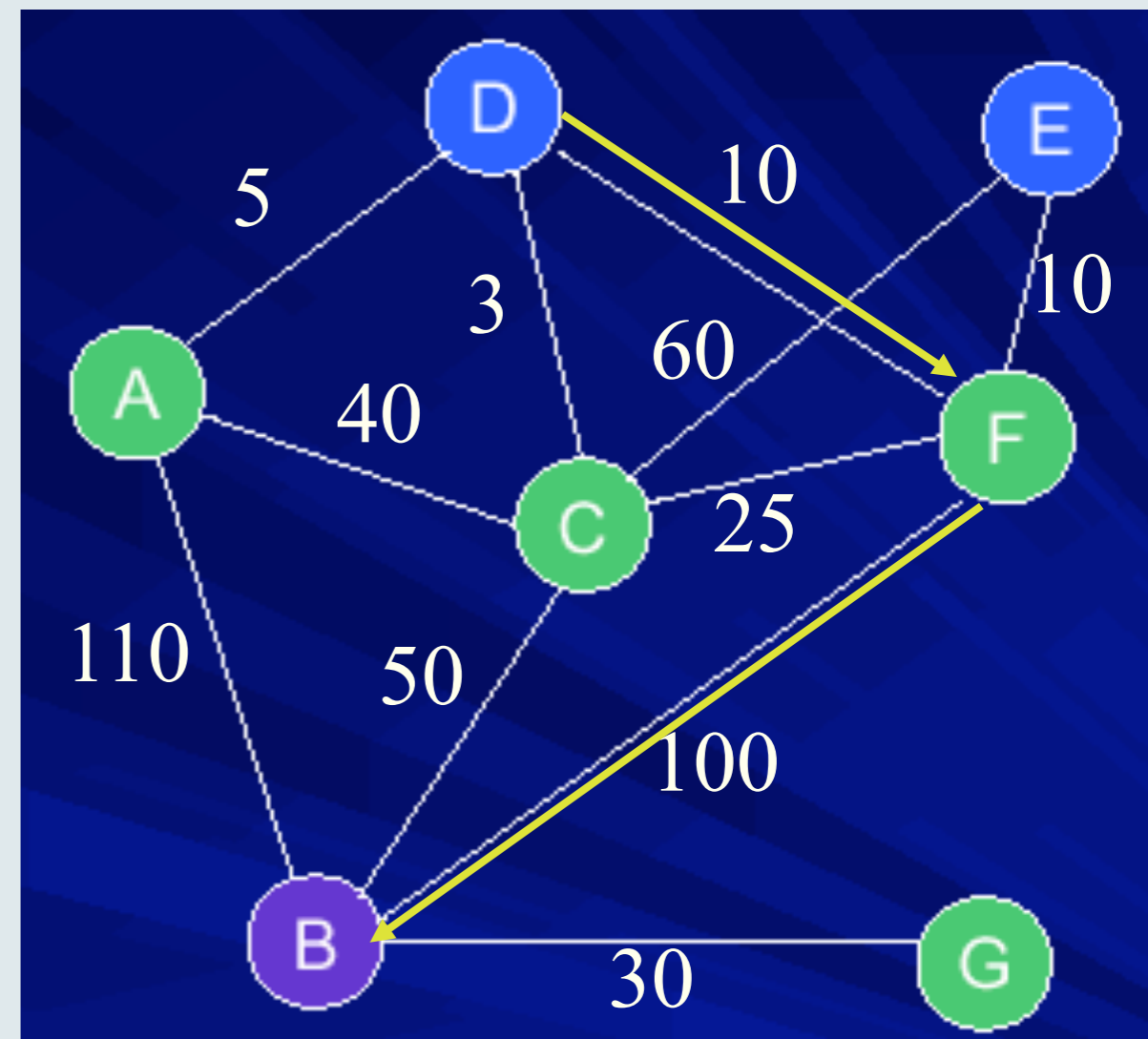
- Available BW
- OLSR: node B will select C as its MPR So all the other nodes know that they can reach B via C
- D->B route is D-C-B, whose bottleneck BW is 3



MRP selection in OLSR

Node	1 Hop Neighbors	2 Hop Neighbors	MPR(s)
B	A,C,F,G	D,E	C

- Available BW
- OLSR: node B will select C as its MPR So all the other nodes know that they can reach B via C
- D->B route is D-C-B, whose bottleneck BW is 3
- Optimal route (i.e., path with maximum bottleneck bandwidth: D-F-B (bottleneck bandwidth of 10)



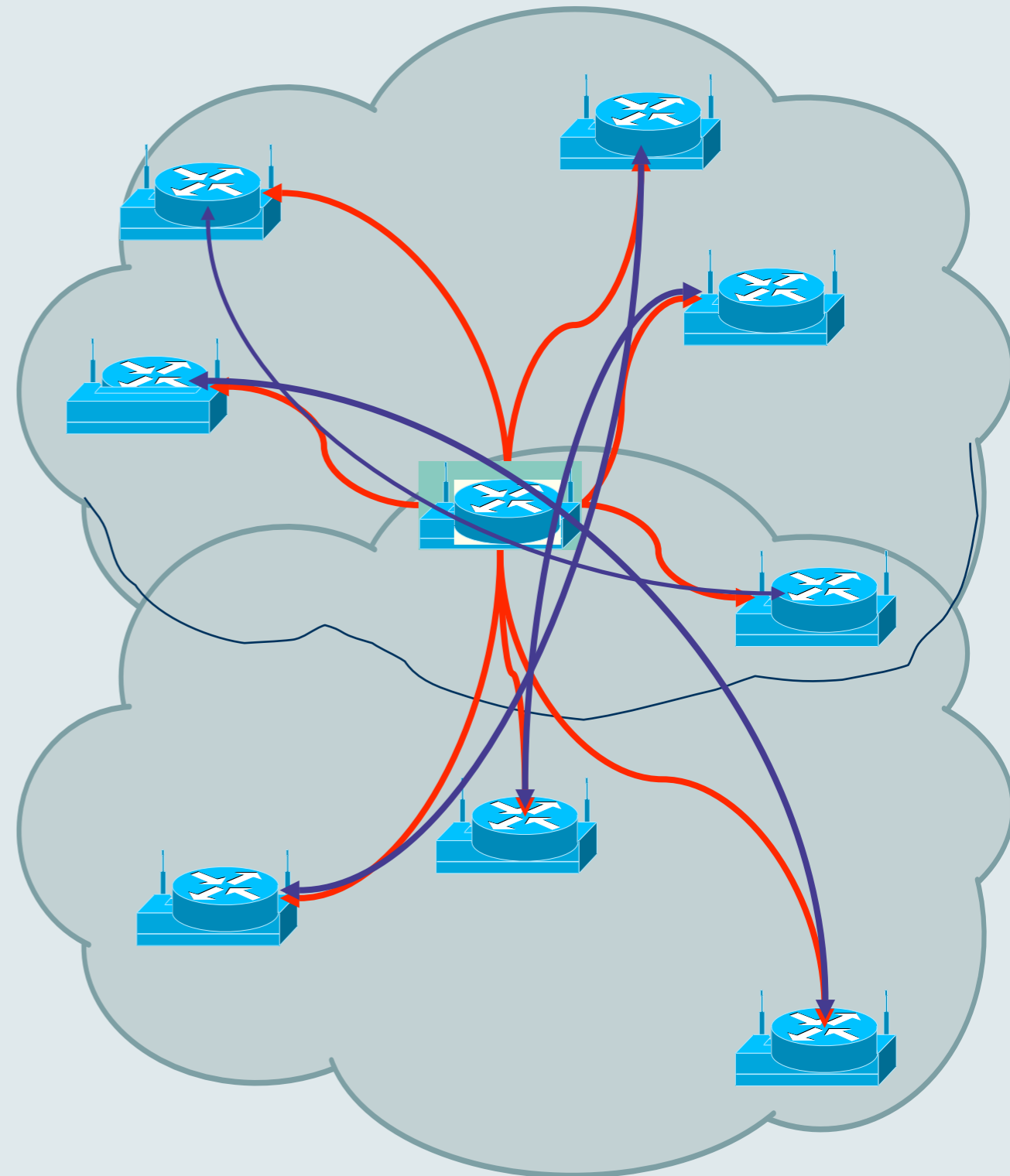
Multi-Point Relays/routers

Passes Topology Information

Acts as router between hosts

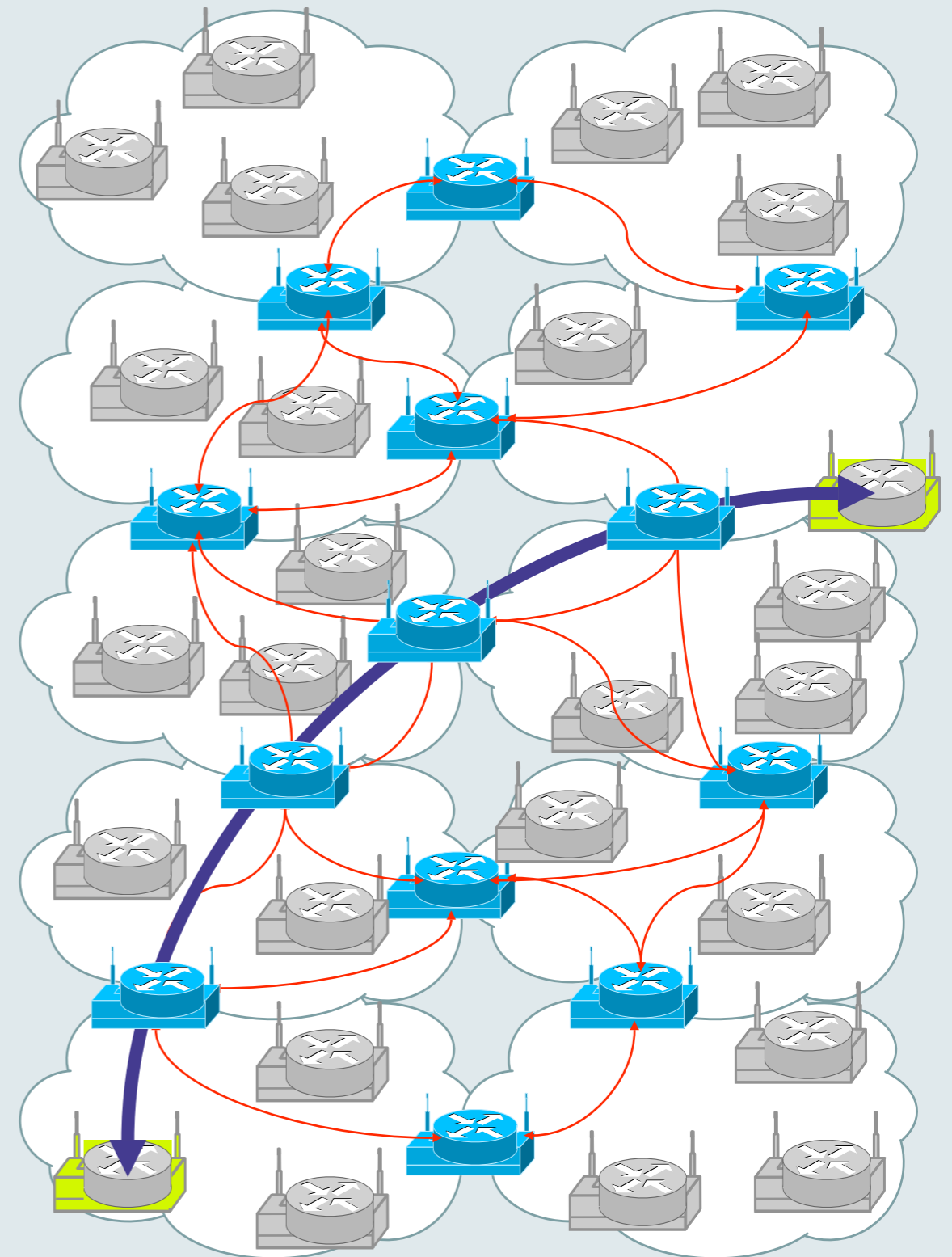
Minimizes information retransmission

Forms a routing backbone



Structure of an OLSR Network

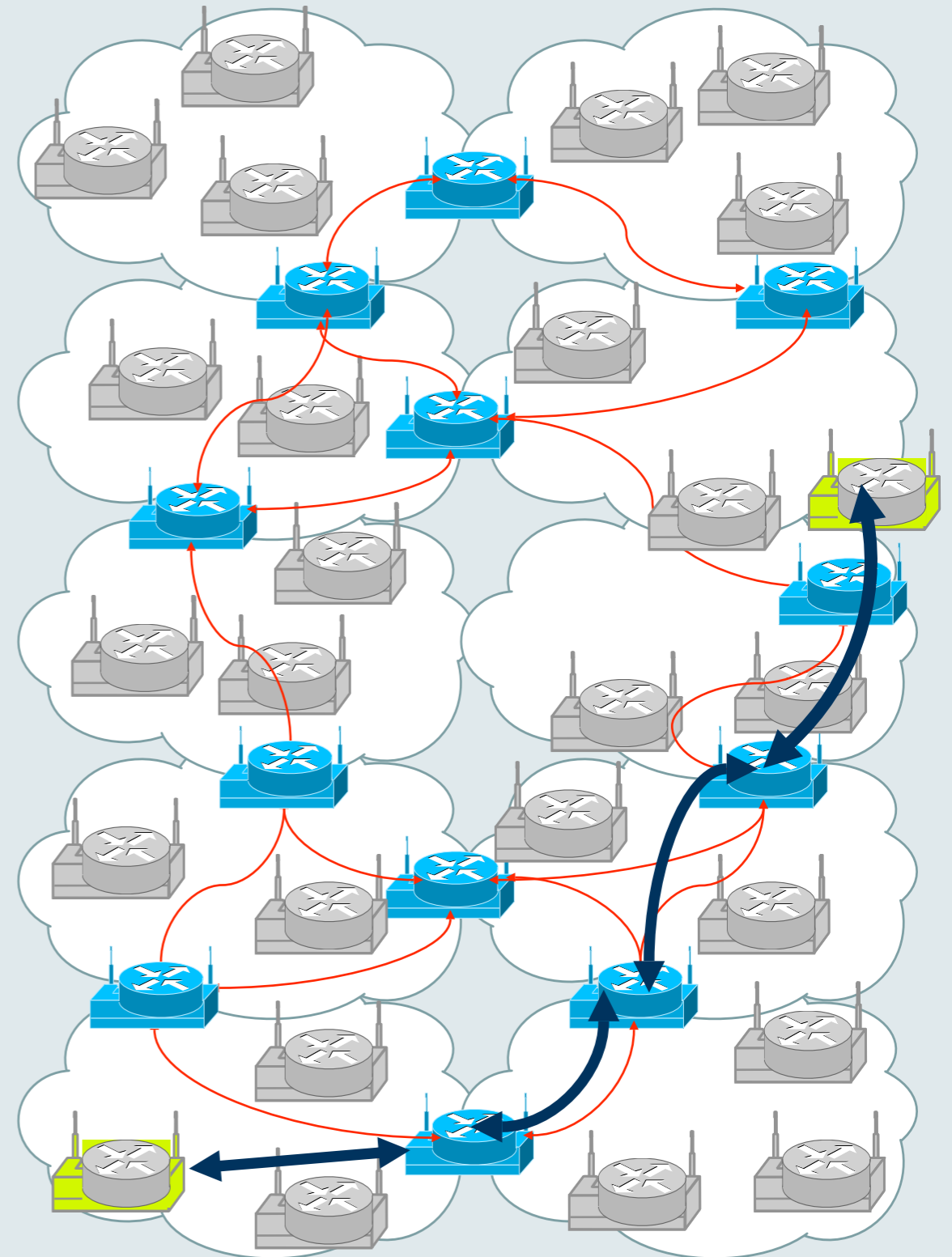
MPRs form **routing backbone**
Other nodes act as “hosts”



Structure of an OLSR Network

MPRs form **routing backbone**
Other nodes act as “hosts”

As devices **move**
Topological relationships change
Routes change
Backbone shape and composition changes



MPR information declaration

- TC – Topology control message:
 - Sent **periodically**. Message might not be sent if there are no updates and sent earlier if there are updates
 - **Contains:**
 - MPR Selector Table
 - Sequence number

- Each node **maintains** a *Topology Table* based on TC messages
 - **Routing Tables** are calculated based on Topology tables

Topology Table

Destination address	Destination's MPR	MPR Selector sequence number	Holding time
---------------------	-------------------	------------------------------	--------------

MPR Selector in the received TC message

Last-hop node to the destination.

Originator of TC message

Topology Table (cont)

- Upon receipt of TC message:
 - If there exist some entry to the same destination with **higher** Sequence Number, the TC message is ignored
 - If there exist some entry to the same destination with **lower** Sequence Number, the topology entry is removed and the new one is recorded
 - If the entry is the **same** as in TC message, the holding time of this entry is refreshed
 - If there are **no** corresponding entry – the new entry is recorded

Routing Table

- ❑ **Each node** maintains a routing table to all known destinations in the network
- ❑ Routing table is **calculated** from Topological Table, taking the connected pairs
- ❑ Routing table:
 - **Destination address**
 - **Next Hop address**
 - **Distance**
- ❑ Routing Table is **recalculated** after every change in neighborhood table or in topological table

802.11s

- ▶ Extension de 802.11 para definir soporte para redes Ad-hoc en 802.11
 - Aun esta en draft pero podria aprobarse muy pronto
- ▶ Dispositivos Mesh Points (MPs)
- ▶ Enrutamiento obligatorio por defecto HMMP (Hybrid Wireless Mesh Protocol)
 - Basado en AODV y enrutamiento basado en arboles
- ▶ Enrutamiento alternativo OLSR
- ▶ Los MPs pueden comunicarse entre si
- ▶ Los MPs pueden ser access points que dan acceso a redes 802.11 de tipo infraestructura
- ▶ Los MPs pueden ser gateways a la red cableada

- ▶ El proyecto OLPC dice soportar 802.11s