TEORÍA DE LA SEÑAL Y COMUNICACIONES

Universidad Rey Juan Carlos

Topología y encaminamiento en Redes móviles y RAHI

Máster Oficial en Sistemas Telemáticos e Informáticos

Escuela Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

Curso 2012/13

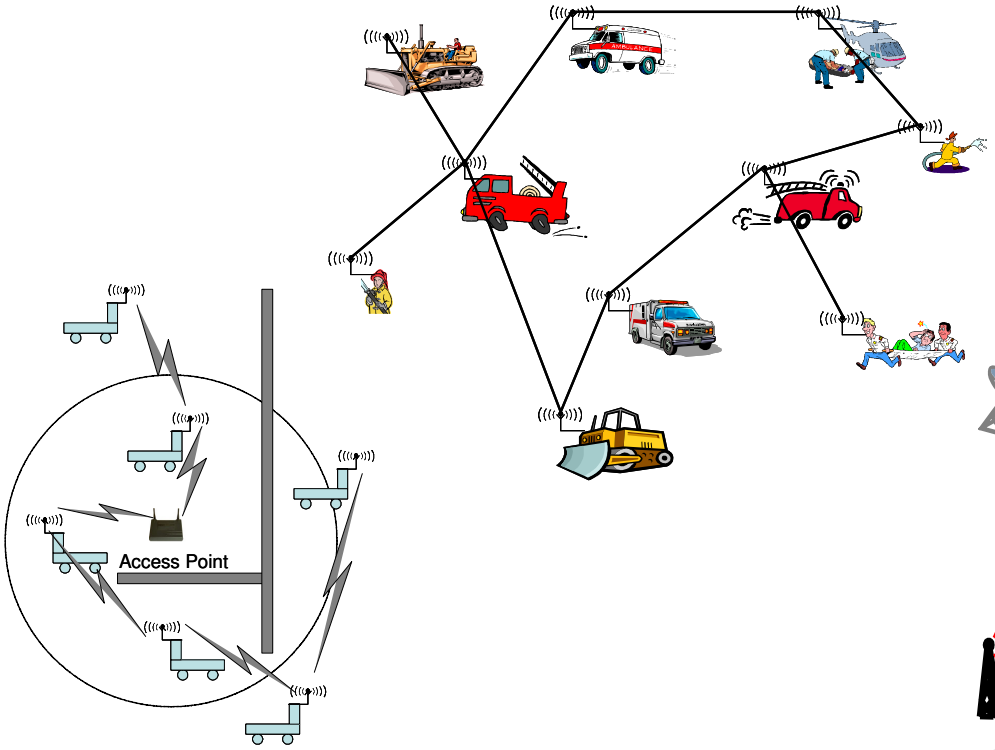
Esta presentación utiliza material cuya autoría corresponde a: Figuera; Marques; Karl&Willing; así como contribuciones menores de distintas autorías.

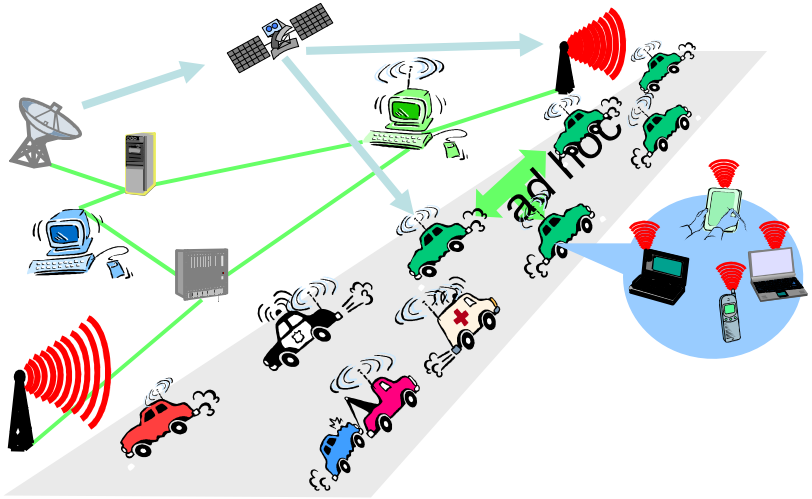
Índice

Índice

Escenarios básicos

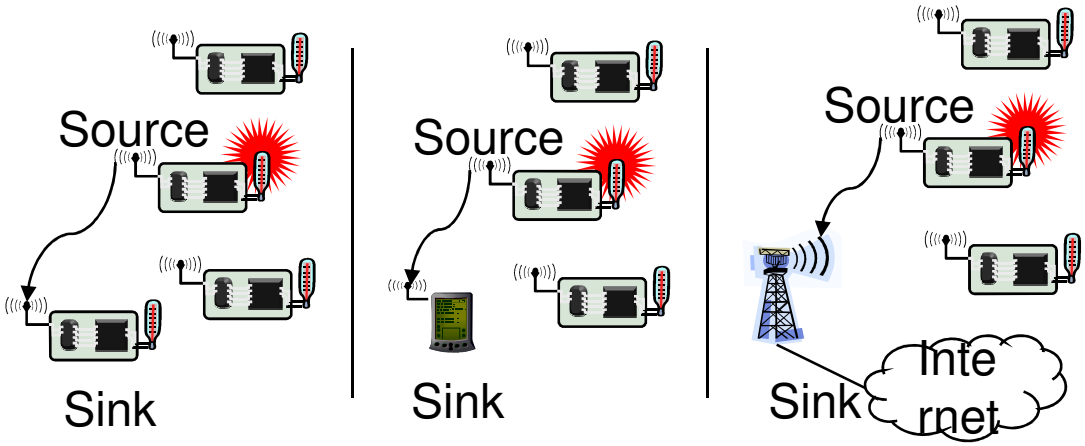
- Nodos hablando entre ellos.
- Nodos accediendo a un elemento en otra red.





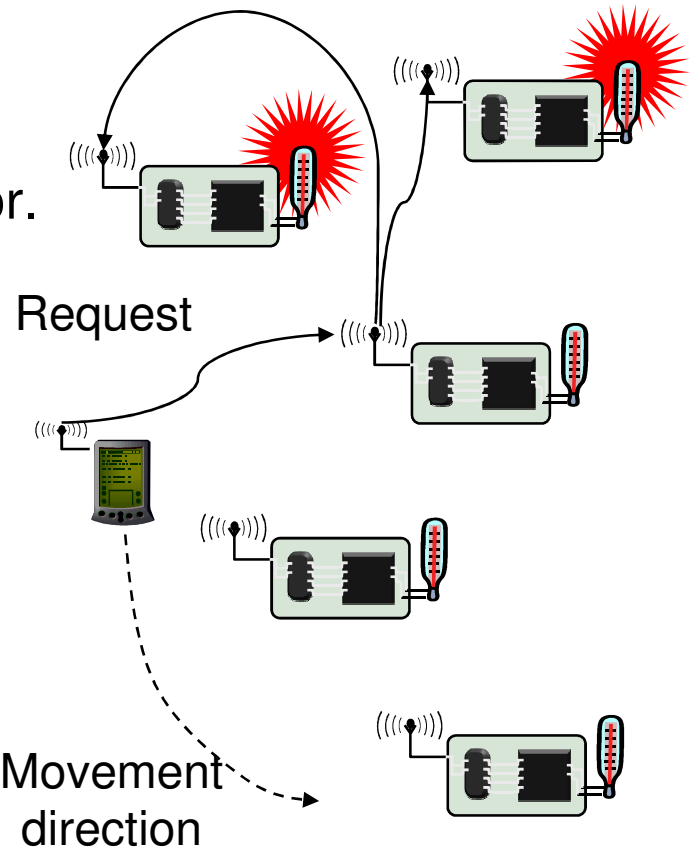
Escenarios básicos: redes de sensores

- 3 partes básicas:
  - Fuente de información.
  - Transporte.
  - Sumideros:
    - Nodos que pertenecen a la red y además a otra.
    - Aparecen y desaparecen, se mueven para recolectar información.



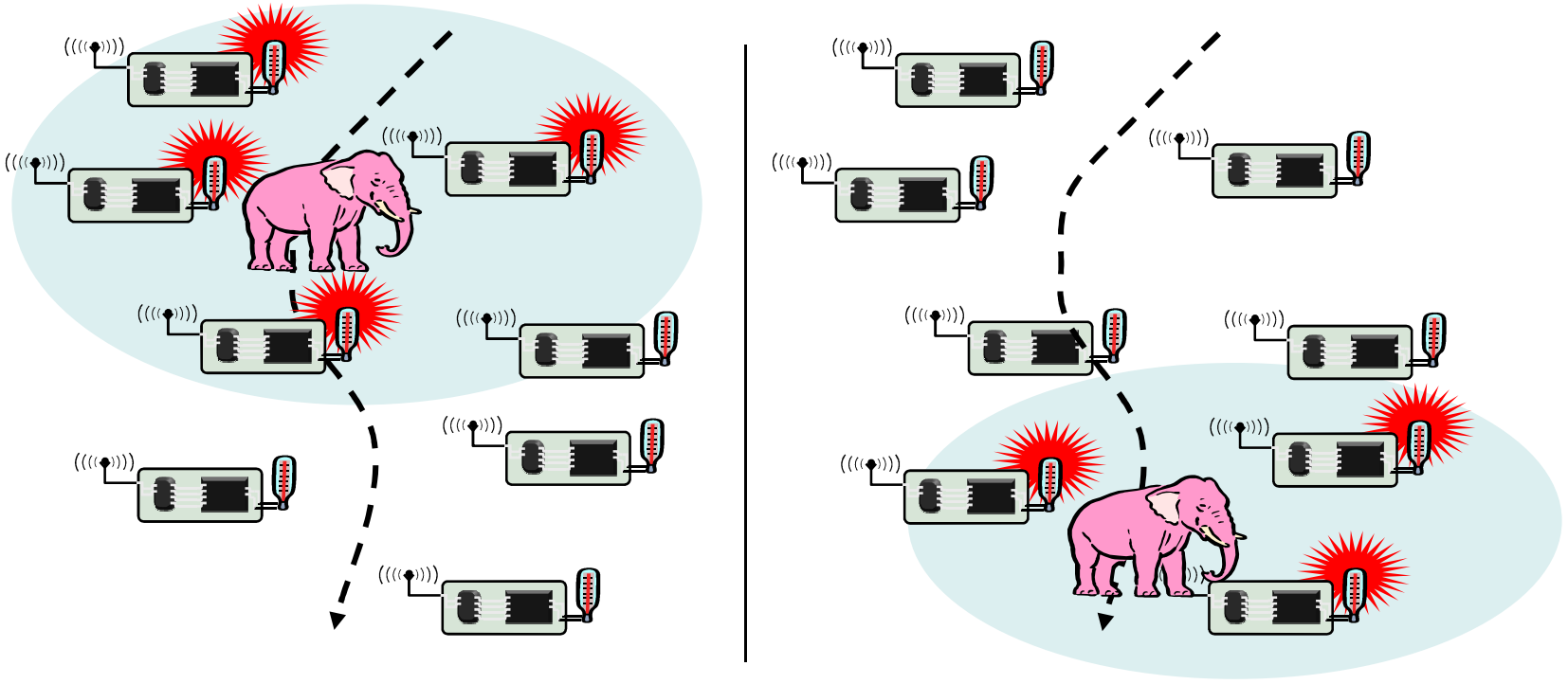
Movilidad

- ¿Qué elementos se pueden mover?
- Nodo
  - La fuente, el destino o un nodo repetidor.
  - Por voluntad propia o a la fuerza.
  - De forma determinista o aleatoria.
- Sumidero



Movilidad (II)

- Evento (es como si se moviera la fuente!!)



Objetivos de optimización/diseño

- Calidad de Servicio
  - A bajo nivel: tasa, retardo…
  - A alto nivel: calidad vista por la aplicación
  - WSN: probabilidad de detección, probabilidad de error de clasificación de evento, precisión de la medida, capacidad de seguimiento de móviles… (mezcla entre alto nivel y bajo nivel)
- Eficiencia energética
  - Energía por bit correctamente recibido
  - Energía por evento comunicado
  - Compromiso QoS/energía
  - Tiempo de vida de la red (t.fallo primer nodo, t. de vida al 50% de la red, t. de partición, t. fallo primer evento)
- Escalabilidad

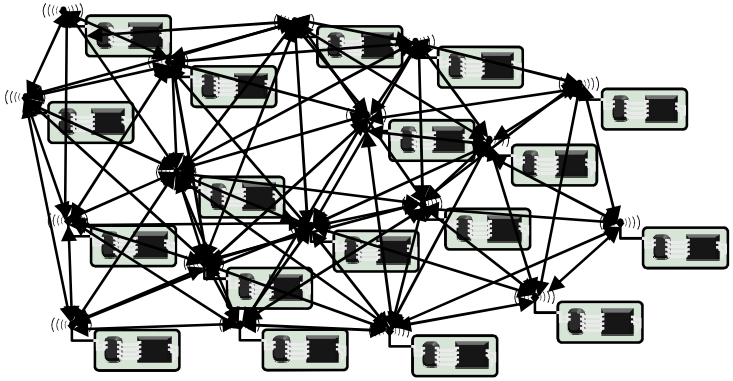
Principios de diseño

- Organización distribuida
  - MAC; enrutado…
  - Problema: eficiencia en comparación con centralizado
  - Opción: centralismo limitado (clustering) ➔ Jerarquía
- Procesado de información intra-red
  - Manipulamos, procesamos la información en la propia red
  - Agregación: media, máximo, mínimo del valor de un evento según recorre la red
  - Compresión de información
- Procesos basados en el dato
  - Habitualmente: IDs de transmisor y receptor importantes (communicaciones usuario-usuario)
  - En WSN: a veces no es relevante quién transmite, sino únicamente lo que se transmite
- Diseño cross-layer

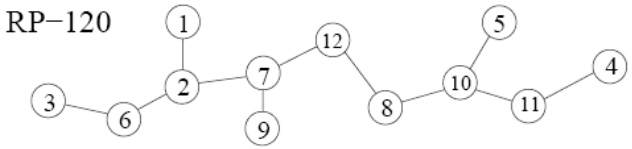
Índice

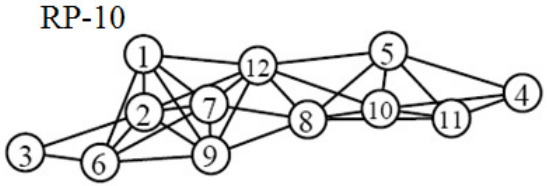
Motivación

- En una red densa existen muchas conexiones para cada nodo.



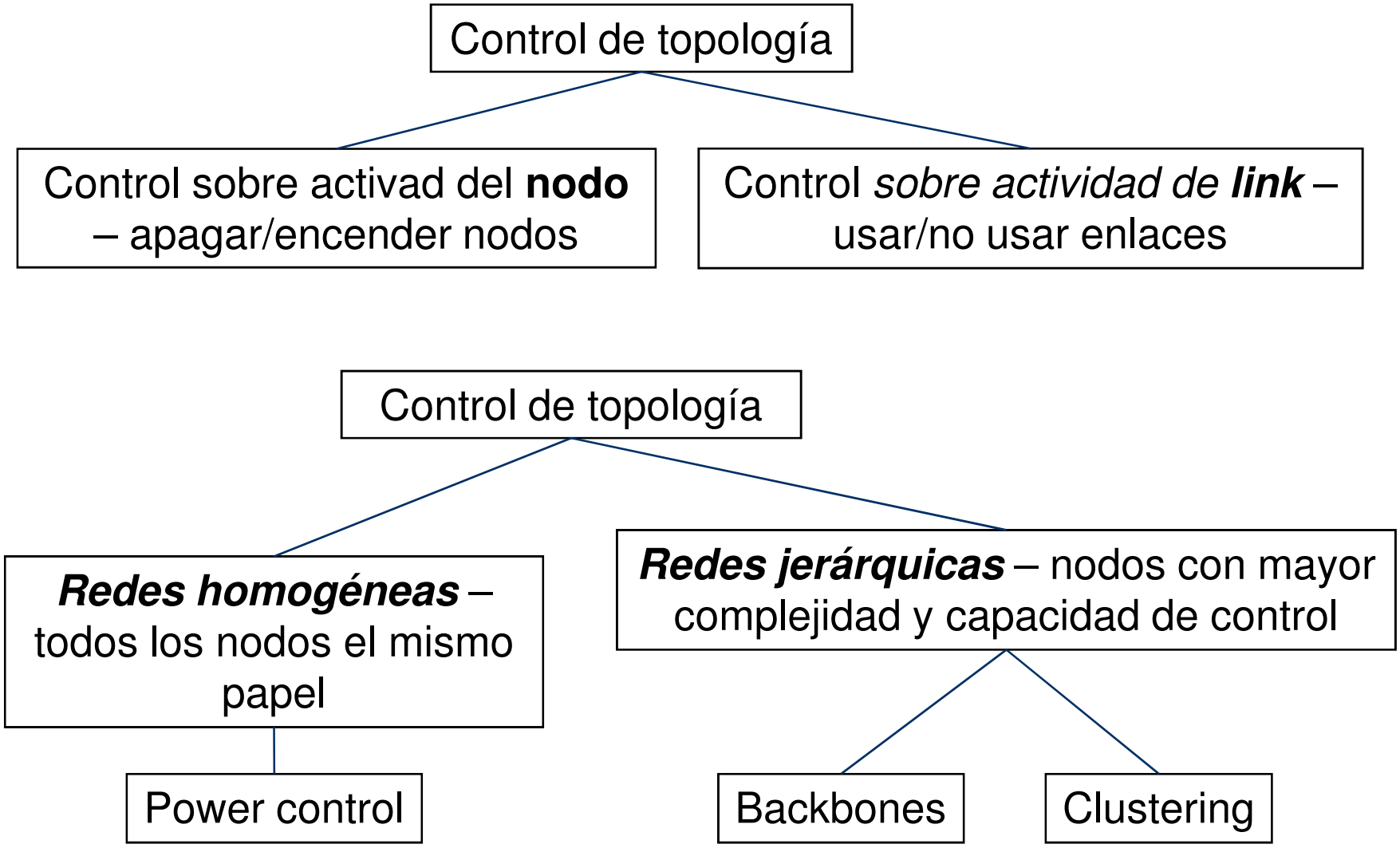
  - Esto puede complicar mucho el MAC y producir muchas colisiones ➔ RP-SNR_{min}





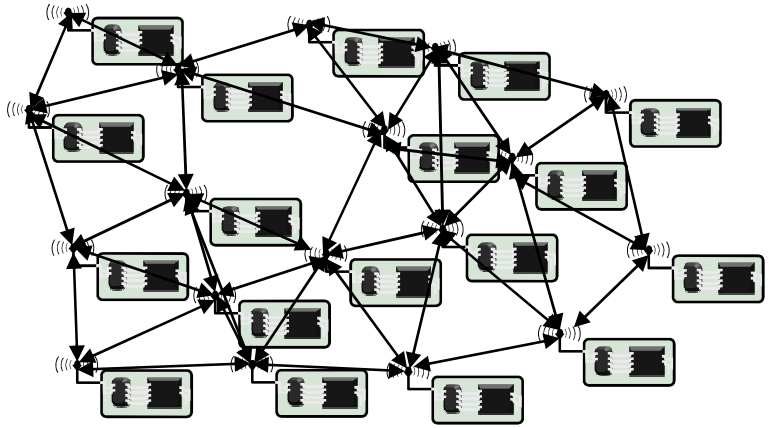
  - Hacer la topología más sencilla: la topología se encarga de establecer qué nodo se puede comunicar con qué otros nodos.
  - ¿Cómo lo controlamos?

Métodos básicos



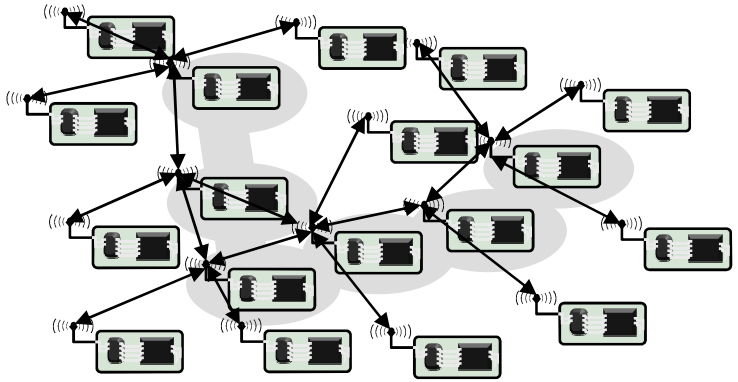
Control de topología en redes homogéneas

- Control de la potencia de transmisión:
  - No usar siempre el máximo de potencia.
  - Controlar el rango de transmisión.
  - Controlar el número de vecinos.
  - La topología aparece más despejada.
  - Menos interferencia.



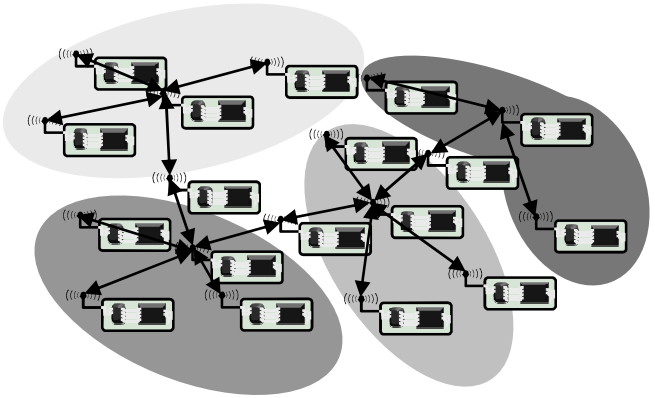
Control de topología en redes jerárquicas

- Se construye una subred principal (troncal)
  - Algunos nodos controlan a sus vecinos (conjunto dominante).
  - Cada nodo debe tener un nodo dominante.
  - Los nodos dominantes deben estar conectados.
  - Solamente se usan enlaces entre nodos dominantes y de nodo normal a nodo dominante.



Control de topología en redes jerárquicas (II)

- Se construyen sectores (clusters):
  - Se divide la red en clusters.
  - Cada nodo pertenece solamente a un cluster…
  - … excepto nodos puente.
  - Los grupos pueden tener nodos cabecera.
  - Todos los nodos de un cluster son vecinos inmediatos de su nodo cabecera.
  - Los nodos cabecera también son un conjunto dominante pero están alejados entre ellos.



LEACH

- Al algoritmo más "popular" para la creación de clusters es LEACH
  - Dos niveles de jerarquía: Cluster Heads (CH)
  - Comunicación Intra-cluster (one hop)
  - Comunicación Inter-cluster (one hop)
  - Los CH se seleccionan aleatoriamente y cambian a lo largo del tiempo (para no agotar energía)

- ¿Cómo se asocian los nodos?
  - Los CH lanzan un mensaje de difusión
  - Si un nodo sólo recibe mensaje de un único CH ➔ Se asocia a ese
  - Si más de uno ➔ Lo elige de acuerdo a algún criterio (típicamente menor energía de tranmisión)

- KO: los clusters pueden ser muy diferentes, la distnacia del CH al sink puede ser muy grande, …

Mejoras a LEACH

- Muchas mejoras:
  - (A) Mecanismos de múltiples saltos (árboles); (B) Reconfiguración, tolerancia a fallos,…; (C) Mecanismos **más inteligentes de selección de cluster head**

- Ejemplo de **(C)**:
  - La selección de CH sigue siendo aleatoria, pero tiene en cuenta otros aspectos
  - Objetivos de diseño: (1) minimizar el consumo de energía de la red, (2) hacer que ese consumo sea lo más homogéneo posible
  - Ideas básicas: (a) tener en cuenta la energía remanente (y la de nuestros vecinos), (b) tener en cuenta la distancia a la fuente
  - El nodo n generamos un num aleatorio x y es CH si $x<T(n)$, donde

$$T(n) = \alpha \frac{(d_{max} - d_{toBS})}{d_{max}} + \beta \frac{E_{rem}}{E_{mean}} p_{CH} + \gamma T_L(n)$$

Donde el umbral LEACH es:

$$T_L(n) = \begin{cases} \frac{p}{1-p\cdot\left(r \cdot mod \frac{1}{p}\right)} & , \text{si } n \in G \\ 0 & , \text{en cualquier otro caso} \end{cases}$$

Además, *n* será CH sólo si:

$$E_{rem} \geq \chi \cdot E_{mean}$$

Criterios de diseño en control de la topología**

- Partimos de una topología G y llegamos a una topología G'.
- Conectividad: si dos nodos están conectados en G, deben estar conectados en G'.
- Factor de estiramiento:
  - De salto: ¿cuánto más largos son los nuevos caminos en G' que en G?
  - De energía: ¿cuánta energía de más requiere el camino más eficiente en G' que en G?
- Robustez frente a movilidad.
- Sobrecarga de tráfico de gestión (el coste de mantenerla/controlarla puede ser muy alto).

Identificación de nodos (I)

- Nombres: denotan “cosas”, pueden ser únicos o no, centrados en nodo o en datos.
- Direcciones: datos necesarios para encontrar esas cosas.
- Nombres ←→ Direcciones: Mapeo (i.e. DNS)
- En redes grandes, ad-hoc, el problema consiste en:
  - Asignar direcciones.
  - Dar de baja direcciones.
  - Resolver conflictos.

Identificación de nodos (II)

- Técnicas distribuidas:
  - Opción 1. Asignación aleatoria: ¿probabilidad de conflicto?
  - Opción 2. Evitar direcciones ya usadas por los vecinos.
  - Opción 3. Reparar problemas: direcciones temporales con consulta sobre disponibilidad al resto de nodos.

- Nuevos "paradigmas":
  - Direcciones basadas en contenido (WSN): describe el contenido, no los nodos en sí. Por ejemplo, la dirección describe los intereses del nodo (información que está interesado en recibir).
  - Direcciones basadas en posición geográfica: útil para protocolos de enrutado geográficos.

Índice

Motivación y aspectos de diseño

- ¿Por qué el enrutado en RAHI es diferente?
  - En RAHI todos los nodos pueden ser routers
  - Cambios en la topología debido a movilidad
  - Canales inalámbricos de muy baja calidad
  - Interferencia
  - Entorno distribuido
  - Recursos limitados:
    - Energía
    - Velocidad (típicamente referido como ancho de banda)
    - Complejidad

Motivación y aspectos de diseño (II)

- En el diseño de un protocolo de enrutado hay que tener en cuenta los siguientes parámetros:
  - Aplicación (sistema de seguridad, medidas medioambientales, comunicaciones personales…)
  - Tipo de información a transportar (datos provenientes de un sensor, streaming de vídeo…)
  - Dimensión de la red
  - Movilidad de los nodos
  - Energía disponible
  - Complejidad de los nodos

Motivación y aspectos de diseño (III)

Objetivos de diseño:

- Auto-configuración distribuida.
- Mínimo consumo de energía.
- Mínimas necesidades computacionales.
- Tolerancia a fallos.
- Escalabilidad.
- Adaptabilidad al hardware.
- Poca sobrecarga de paquetes de gestión.
- Gestión de calidad de servicio.

Taxonomía:

- Reactivos frente a proactivos.
- Salto a salto frente a encaminamiento en origen.
- Centrado en el nodo - centrado en datos – centrado en la posición.

Encaminamiento en Origen vs Salto a Salto

- Encaminamiento salto a salto (Hop by Hop Routing)
  - Cada router decide sólo el siguiente salto
  - La información de enrutado la guardan los routers
- Encaminamiento en origen (Source Routing)
  - La ruta se establece al enviar el paquete
  - Cada paquete lleva incluida su ruta
  - La información de enrutado la guarda el paquete

Protocolos centrados en el nodo

- Es la manera tradicional de ver una red
- Cada nodo está identificado con una dirección única y se establecen jerarquías de forma sencilla
- El enrutado se hace transparente a los niveles superiores
- El descubrimiento de rutas se suele realizar por medio de inundación. Para evitar el gasto de recursos que esto supone se han propuestos diferentes métodos:
  - Location servers
  - Small world assumption

Protocolos centrados en los datos

- Cambio de filosofía: la red no se encarga de “transmitir del nodo 1 al nodo 3 los datos que el nodo 1 diga”, sino de “transporta al nodo destino los datos que satisfagan la condición X”.

- Existen varios ejemplos que motivan este tipo de filosofía:

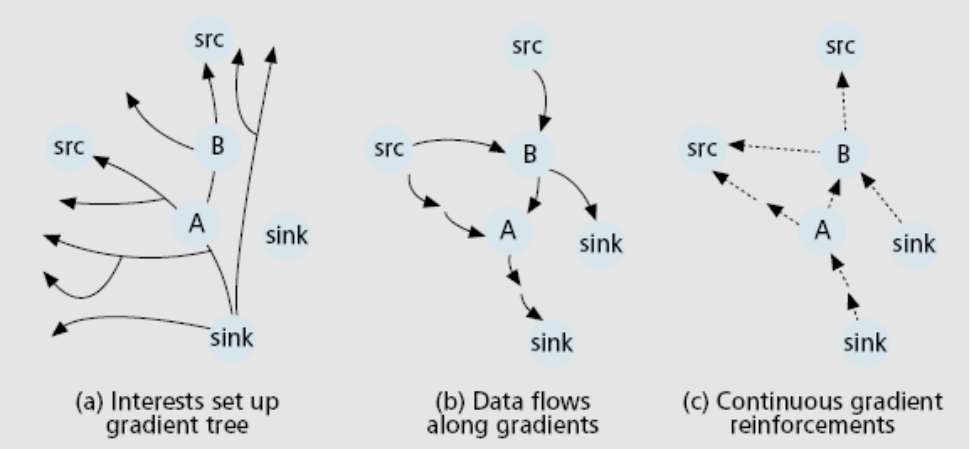
  - Base de datos de sensores: los sumideros realizan “queries” tipo SQL a los sensores, que pueden ser tan complejas como se quiera.

  - Difusión Dirigida: la red es vista como un mecanismo de “noticias por subscripción”.

  - Información correlada: si la información que un nodo transmite ya es conocida por otro (o por la que envió), el segundo nodo no la retransmite

Difusión dirigida

- El enrutamiento se hace en función del contenido: la identidad del nodo es irrelevante en este proceso.
- Un “sumidero” publica su interés por un tipo de datos determinado. Este interés llegará finalmente a la fuente de datos, creando ya la ruta (las entradas se llaman gradientes).
- El resultado es una estructura de árbol (o árboles superpuestos) que pueden llevar información de múltiples fuentes a múltiples destinos.
- La tabla de enrutado de los nodos intermedios contiene sólo el siguiente salto para cada gradiente.
- Cuando llega un paquete, si corresponde a uno o más de los gradientes de la tabla, se envía una copia a cada siguiente nodo.



Centrado en la posición

- Cuando la localización de los datos es tan esencial como los datos en sí mismos, conviene que ambas cosas estén asociadas
- En este caso tiene más utilidad identificar a un conjunto de nodos por su posición que por su dirección lógica
- El sistema GPS puede ser una solución para esta localización, si bien no es siempre viable
- El método básico es la “retransmisión cartesiana”: un nodo que tiene un paquete hacia otro nodo destino, cuya dirección conoce, lo retransmitirá hacia el vecino más cercano al destino
- Ventajas: no son necesarias las tablas, ni el conocimiento extensivo de las rutas
- Inconvenientes: es necesario conocer la posición de uno mismo y del destino

Clasificación de protocolos de enrutado

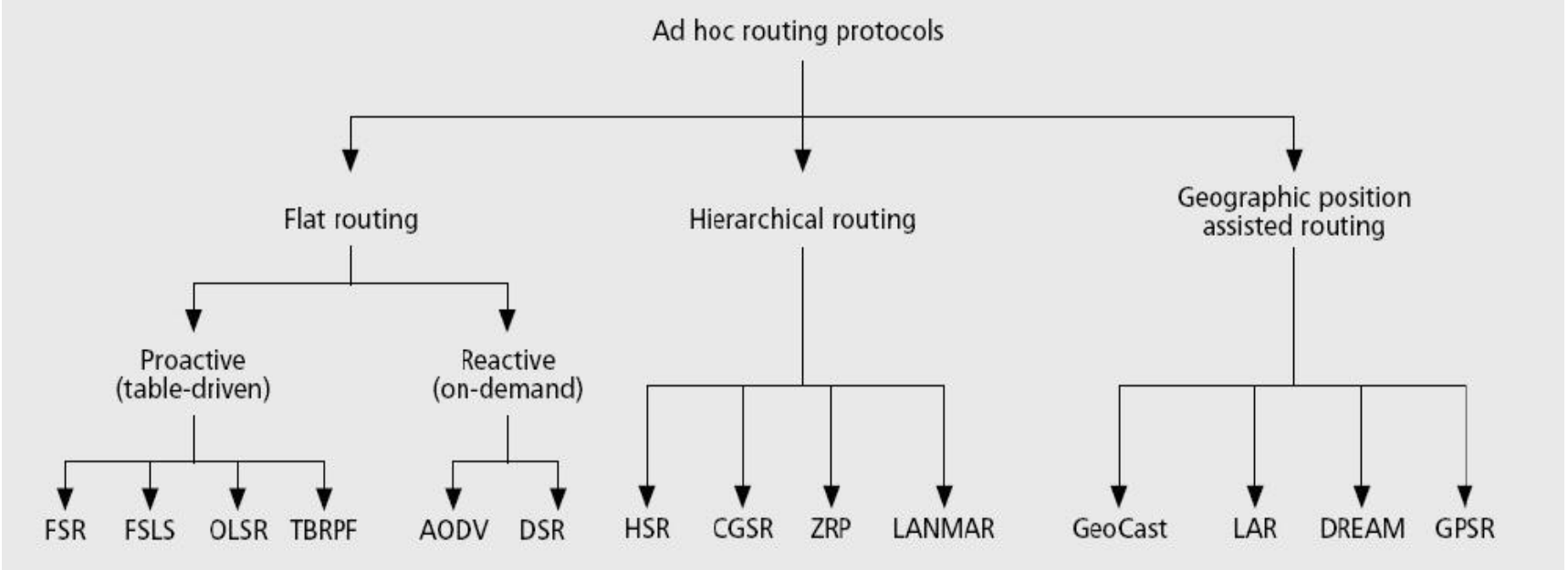


Table-driven: protocolos proactivos

- Reactivo (hacer sólo cuando se necesita)
- Cada nodo mantiene una o más tablas con la información de cómo llegar a cualquier otro nodo de la red
- Las tablas tienen que ser consistentes a lo largo de la red
- Los cambios deben propagarse a través de la red
- Se necesita una visión/topología de la red
- Ejemplo: Destination-Sequenced Distance Vector Protocol (DSDV)
  - Basado en el algoritmo de Bellman-Ford
  - Cada nodo tiene una lista con todos los destinos y # de saltos
  - Las tablas se envían periódicamente a los vecinos para consistencia (distintas estrategias)

Un ejemplo para entender DSVD

El nodo B cambia de posición, ¿cómo afecta este cambio a la tabla de rutas del nodo A?

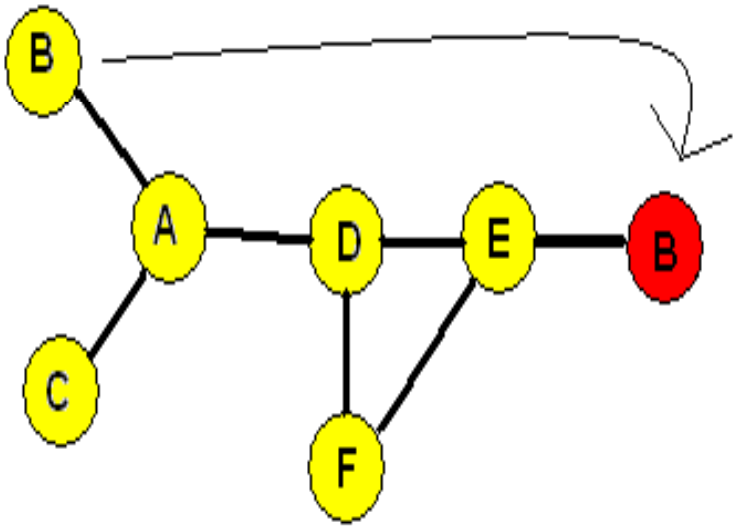


Tabla de rutas en A antes del cambio

Destination	Next Hop	Distance	Sequence Number
A	A	0	S205_A
B	B	1	S334_B
C	C	1	S198_C
D	D	1	S567_D
E	D	2	S767_E
F	D	2	S45_F

Tabla de rutas en A después del cambio

Destination	Next Hop	Distance	Sequence Number
A	A	0	S304_A
B	D	3	S424_B
C	C	1	S297_C
D	D	1	S687_D
E	D	2	S868_E
F	D	2	S164_F

On-demand: protocolos reactivos

- Reactivo (hacer después de –solo si- necesitar)
- Bajo demanda: las rutas se crean sólo cuando se necesitan por parte de la fuente
- Cuando un nodo necesita conocer la ruta hacia un destino, inicia un proceso de "descubrimiento de ruta" (route discovery process)
- La ruta se conserva hasta que el destino desaparece o la fuente pierde el interés

- Ejemplo 1: Ad hoc On-demand Distance Vector Routing (AODV)
  - No necesita mantener info de nodos si no se comunica con ellos
  - Dos mecanismos básicos: route discovery route maintenance

- Ejemplo 2: Dynamic Source Routing (DSR)
  - Como AODV pero con mejor "route maintenance" (tiene una caché donde se almacenan las rutas por si luego hay fallos)

Descubrimiento y mantenimiento de red

- Descubrimiento: La fuente difunde mensaje RREQ. Los nodos intermedios graban en sus tablas la dirección del vecino que ha enviado el RREQ (para considerarla a la vuelta). Cuando RREQ llega al destino, se responde con un RREP a su vecino que lo propaga hacia atrás
- Mantenimiento: Si la fuente se mueve, se reestablece el path. Si es el destino o un nodo intermedio el que se mueve, se envía notificación de fallo y se reinicia el descubrimiento de ruta
- Enrutado en fuente: al final el origen sabe el destino. En AODV los nodos intermedios también lo guardan

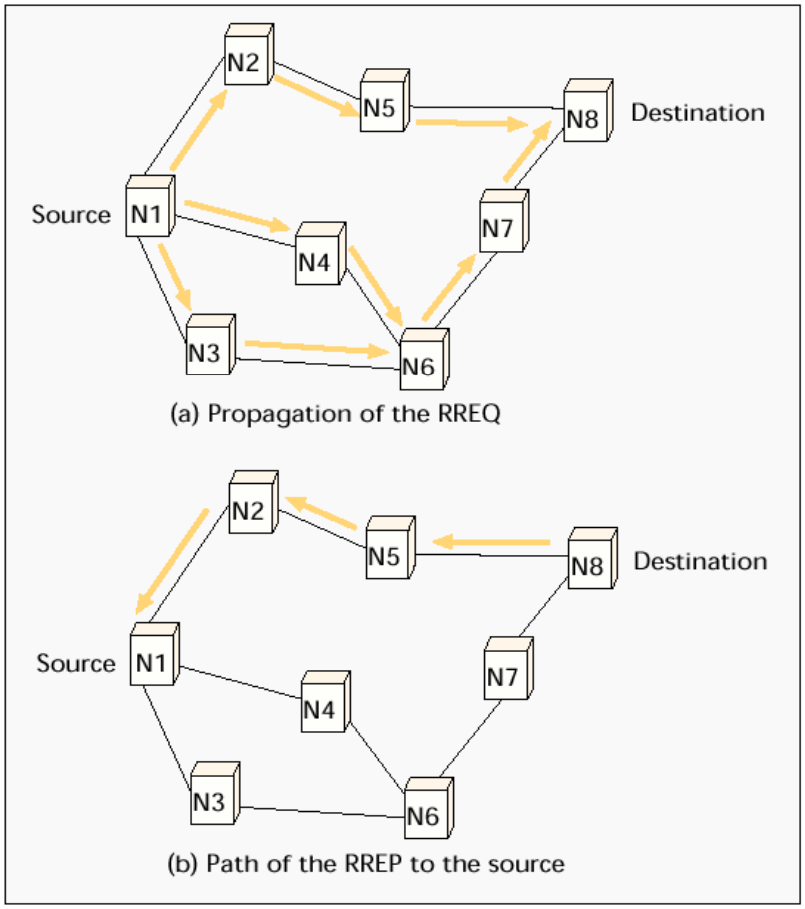
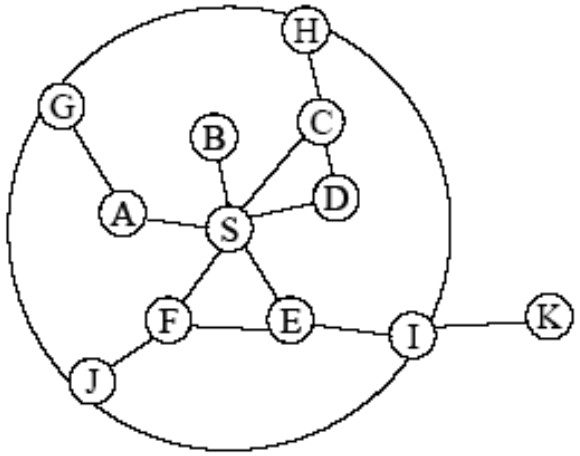


Figure 3. *AODV route discovery.*

Protocolos híbridos/jerárquicos

- Híbridos: mezcla de table-driven y bajo demanda
- Ejemplo: Zone Routing Protocol (ZRP)
- Asociado a cada nodo, existe el concepto de "zona"
  - Dentro de cada zona: table-driven (proactivo)
  - Para enrutado entre zonas distintintas: bajo demanda (reactivo)
  - Los nodos no intentan manentener información global sobre enrutado



- Tres tipos de nodos:
  - Nodos interiores
  - Nodos de frontera
  - Nodos periféricos

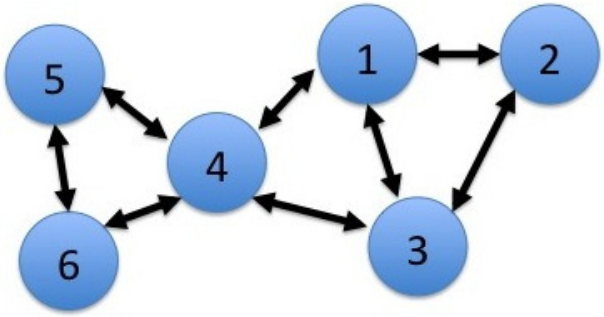
Comparación

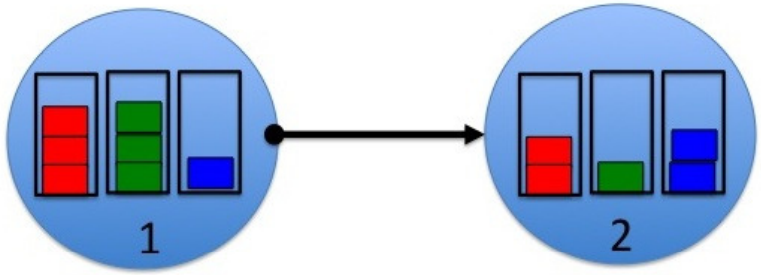
Parámetros	Bajo demanda	Table-Driven
Disponibilidad de información de enrutado	Se obtiene cuando se necesita	Siempre disponible (independientemente de cuándo se necesite)
Actualización periódica de las rutas	No es necesario	Necesario
Solución para la movilidad	Utiliza descubrimiento de rutas localizado	Comunicación entre nodos para conseguir tablas de enrutado consistente
Tráfico de señalización generado	Aumenta si los nodos se mueven más rápido	Mayor que en el caso bajo demanda

- Conclusión: table-driven para redes pequeñas (también quasi-estáticas o muy exigentes en QoS) y on-demand para redes grandes y/o con mucha movilidad

QoS: Dynamic Backpressure

- Algoritmo inventado por Tassiulas y Efremides en 1992





  - Si μ es la capacidad de cada enlace, queremos $(Eq.1)\ \text{Maximize:}\ \sum_{a=1}^{N}\sum_{b=1}^{N}\mu_{ab}(t)W_{ab}(t)$
  - El peso (calidad) de cada enlace es $W_{ab}(t) = \max\left[Q_a^{(c_{ab}^{opt}(t))}(t) - Q_b^{(c_{ab}^{opt}(t))}(t), 0\right]$
  - Tiene en cuenta tanto el nodo como la info!!

- Para RAHI: los enlaces no tienen tasa fija, la QoS no es solo tasa, también energía, etc.

Bibliografía

- H. Karl & A. Willing, “Protocols and Architectures for Wireless Sensor Networks”, Wiley 2005. (Chapters 10, 11 and references therein.)