

Internet de las Cosas

Diego Dujovne

Enrutamiento

Enrutamiento Unicast

- Dada una red o un grafo,
 - Cada nodo tiene un identificador único (ID)
- El objetivo es derivar un mecanismo que permita a un paquete de un nodo cualquiera llegar a un nodo de destino específico.
 - Este es el problema de enrutamiento y reenvío.
 - Para enrutar: Construir estructuras de datos que contengan información de cómo un destino puede ser alcanzado.
 - Reenvío: Consultar estas estructuras de datos para reenviar un paquete dado el siguiente salto.
- Desafíos
 - Los nodos pueden moverse, las relaciones entre nodos cambian.
 - Las métricas de optimización puede ser más complicadas que el mínimo hop count, por ejemplo, eficiencia energética.

Enrutamiento Ad-hoc

- Debido a las restricciones de las redes de sensores, los métodos standard de enrutamiento no son realmente aplicables.
 - Demasiado overhead, demasiado lentas para reaccionar a cambios.
 - Ejemplo: Los algoritmos de Dijkstra y Bellman-Ford (DV)
- La solución más simple: Inundación.
 - No require información (tablas de ruteo). Es simple.
 - Los paquetes generalmente llegan a destino
 - El overhead es prohibitivo, y usualmente no aceptable.

Hacen falta protocolos de ruteo ad-hoc específicos.

Clasificación de protocolos Ad-Hoc

- La pregunta principal: Cuándo opera un protocolo de enrutamiento?
- Opción 1: El protocolo de ruteo siempre intenta mantener sus datos de ruteo actualizados
 - Entonces el protocolo es proactivo. (Activo antes de que las tablas sean realmente requeridas. También se denomina “manejado por tablas”)
- Opción 2: La ruta solamente es determinada cuando esta es requerida.
 - Se dice que el protocolo actúa bajo demanda
- Opción 3: Combinar las opciones anteriores.
 - Protocolos Híbridos

Clasificación de protocolos Ad-Hoc

- La red se puede considerar plana o jerárquica?
 - Se debe comparar el control de topología y el enrutamiento tradicional.
- Qué datos se utilizan para identificar nodos?
 - Un identificador arbitrario?
 - La posición de un nodo?
 - Puede ser utilizado para asistir en protocolos de enrutamiento geográfico, dado que la elección del siguiente salto puede ser calculada basada en la dirección de destino
 - Identificadores que no son arbitrarios pero que llevan cierta estructura?
 - Como en el enrutamiento tradicional
 - Estructurarlo de acuerdo a una posición o a un nivel lógico?

Protocolos Proactivos - DSDV

- Idea: Partir de un protocolo de enrutamiento standard y luego adaptarlo.
- DV (Distance-Vector) Adaptado: ***Destination Sequence Distance Vector (DSDV)***
 - Basado en un procedimiento de Bellman Ford distribuido
 - Agregar información de edad de la ruta para rutear información de publicidad por intercambios de Vector-Distancia. Ayuda a evitar ciclos
 - Enviar periódicamente actualizaciones completas de rutas
 - Frente a un cambio de topología, enviar cambios incrementales de rutas
 - Actualizaciones de rutas inestables son retardadas
 - ... + otros cambios menores

Protocolos Proactivos -OLSR

- Combina un protocolo de estado de enlace y control de topología
- ***Optimized Link State Routing (OLSR)***
- Componente de control de topología: Cada nodo elige el conjunto mínimo dominante para su vecindario de dos saltos
 - Se llaman relays multipunto (retransmisores multipunto?)
 - Solo estos nodos son utilizados para reenvío.
 - Permite la inundación eficiente (o restringida)
- Componente de estado de enlace: Esencialmente un algoritmo de estado de enlace para esta topología reducida
 - La idea clave es reducir el overhead de inundación, en este caso cambiando la topología

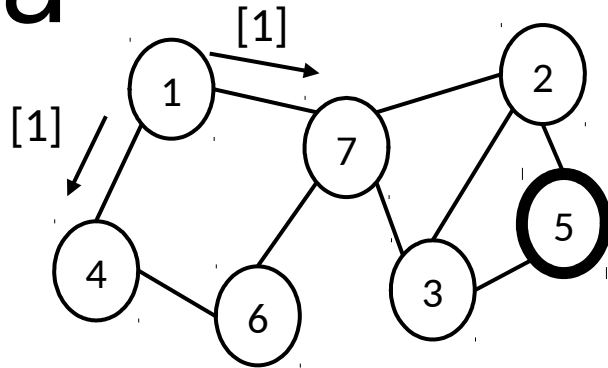
Protocolos Proactivos – Ojo de pescado (Fisheye)

- Fisheye State Routing (FSR) hace una observación básica: Cuando un destino es lejano, los detalles sobre el camino no son relevantes, solo en el vecindario los detalles hacen falta.
 - Ver a los gráficos como si se utilizaran lentes de ojo de pescado.
 - Las regions tienen distintos niveles de exactitud de información de ruteo
- En la práctica:
 - Cada nodo mantiene una tabla de topología de una red.
 - Solamente distribuye las actualizaciones de estado de enlace localmente
 - Usa actualizaciones de enrutamiento más frecuencias para nodos con menor alcance.

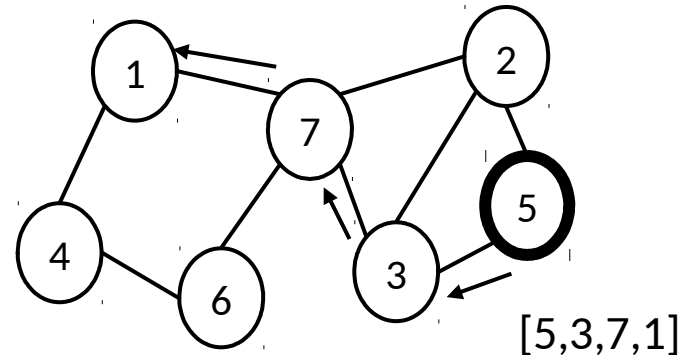
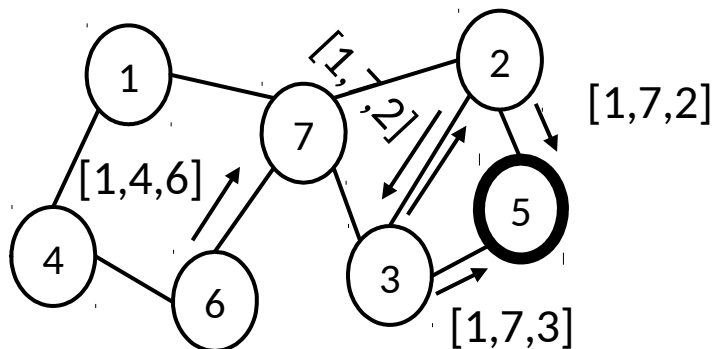
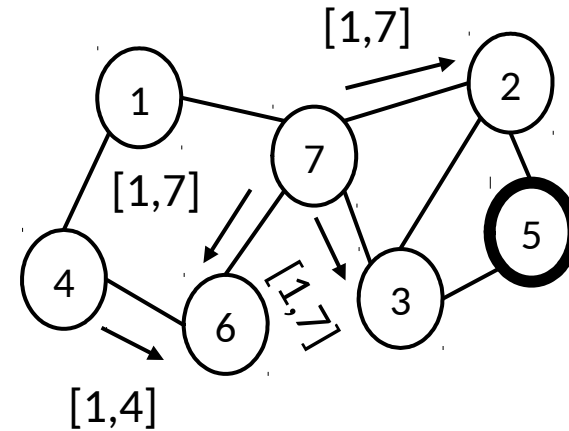
Protocolos Reactivos: DSR

- En un protocolo reactivo, cómo reenviar un paquete a un destino específico?
 - Inicialmente, no hay información sobre si el próximo salto está disponible
 - Solamente (?) un único recurso: Enviar el paquete a todos los vecinos – inundar la red -
 - Esperanza: En algún punto, el paquete va a llegar a destino, y la respuesta es enviada al origen. Se usa esta respuesta para aprender hacia atrás (backward learning) la ruta del destino a la fuente.
- En la práctica se llama: ***Dynamic Source Routing (DSR)***
 - Usa paquetes distintos para requerimiento de ruta y respuesta de ruta (***route request/route reply***) para descubrir la ruta.
 - Los paquetes de datos son enviados una vez que la ruta ha sido establecida
 - Los paquetes de descubrimiento son más pequeños que los paquetes de datos.
 - Se almacena la información de ruta en los paquetes de descubrimiento.

Procedimiento de descubrimiento de ruta



Buscar una ruta de 1 a 5



El nodo 5 usa información de ruteo guardada en RREQ para enviar de retorno, a través del source routing, una respuesta.

Modificaciones y extensiones al DSR

- Los nodos intermedios pueden enviar respuesta de rutas (route replies) si ya conocen la ruta.
 - Problema: las rutas pueden estar desactualizadas
- Funcionamiento promiscuo de los dispositivos de radio: los nodos pueden aprender sobre una topología escuchando los mensajes de control
- Retardos al azar para generar respuesta de rutas
 - Muchos nodos pueden tener una respuesta: tormentas de respuestas
 - NO necesariamente para acceso al medio: Es un tema de la MAC.
- Reparación local:
 - Cuando un error es detectado, usualmente el transmisor genera un timeout y construye una nueva ruta.
 - En su lugar: Intentar cambiar localmente la ruta designada por la fuente (source)
- Mecanismos de gestión de Cache:
 - Para remover los caches desactualizados rápidamente
 - Tiempo de vida fijo o adaptivo, mensajes para retirar una ruta del caché...

Protocolos Reactivos – AODV

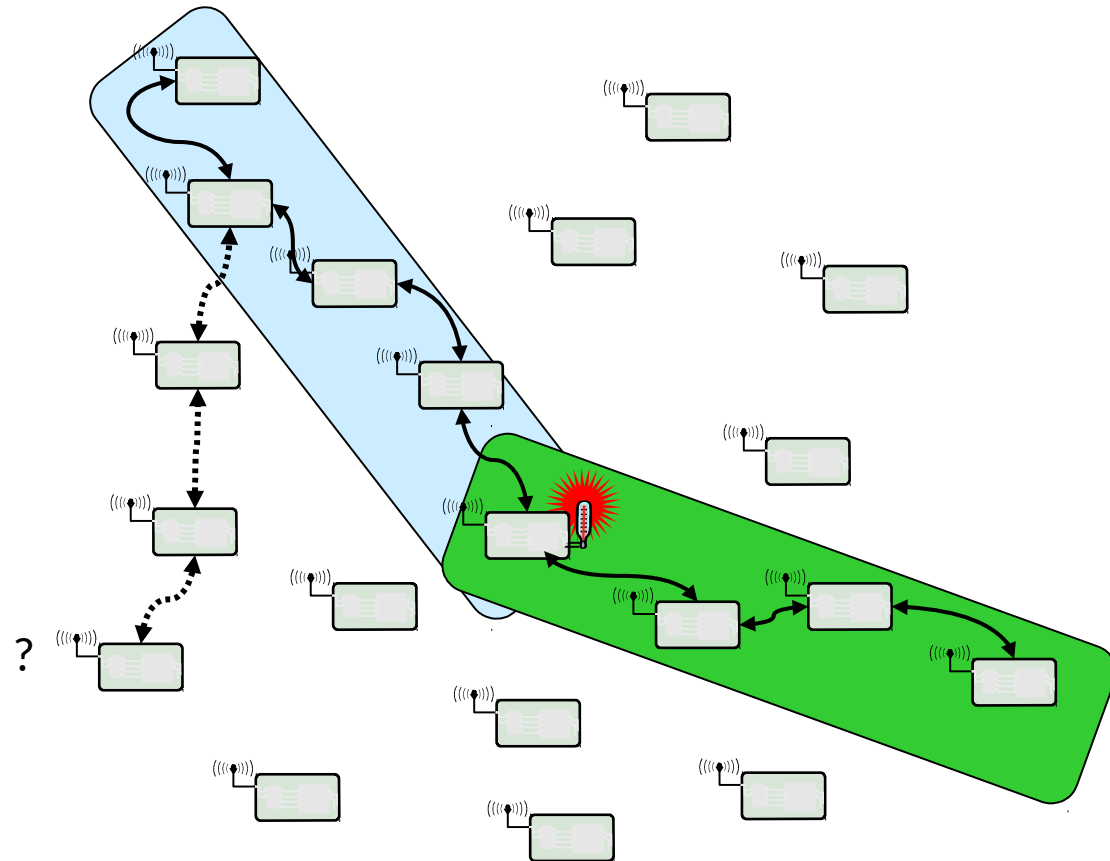
- ***Ruteo Ad hoc On Demand Distance Vector (AODV)***
 - Protocolo muy popular
 - Esencialmente la misma idea del DSR para el procedimiento de descubrimiento
 - Los nodos mantienen tablas de enrutamiento en lugar de enrutamiento a fuente (source)
 - Se agregan números de secuencia para manejar los caches desactualizados
 - Los nodos recuerdan de dónde vino un paquete y llenan sus tablas de enrutamiento con esa información.

Protocolos reactivos – TORA

- Observación: En un terreno montañoso, ir hacia la desembocadura del río es simple: solo hace falta ir montaña abajo.
- Idea: Convertir la red en un terreno montañoso
 - Usar un “paisaje” para cada destino
 - Asignar “Alturas” a los nodos cuando van montaña abajo y así se alcanza el destino. En efecto: es como orientar las aristas entre los vecinos
 - Hace falta que el grafo orientado no tenga ciclos.
- Reacción a cambios de topología
 - Cuando el enlace es removido (considerando que fuera la única salida del nodo) Se revierte la dirección de todos los demás enlaces para incrementar la altura
 - Reaplicar continuamente hasta que cada nodo, a excepción del destino que tiene un único conector. Esto va a tener éxito en un gráfico conectado.

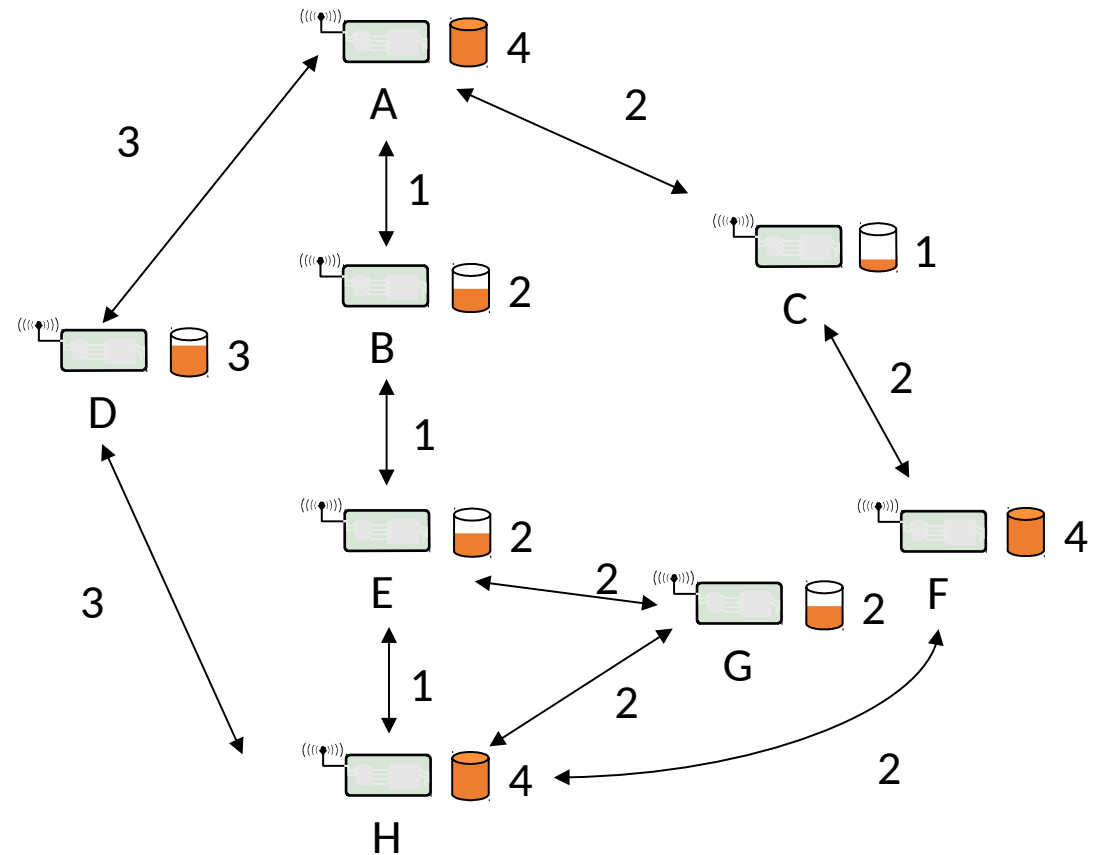
Propuesta alternativa: ruteo por chisme o rumor.

- Dar vuelta el problema: Pensar en un agente paseándose en la red buscando datos (eventos, ...)
- El agente inicialmente hace una caminata al azar (random walk)
- Deja “trazas” en la red
- Luego los agentes pueden utilizar estas trazas para encontrar datos
- Esencialmente trabaja debido a la alta probabilidad de intersección de líneas.



Unicast eficiente en energía

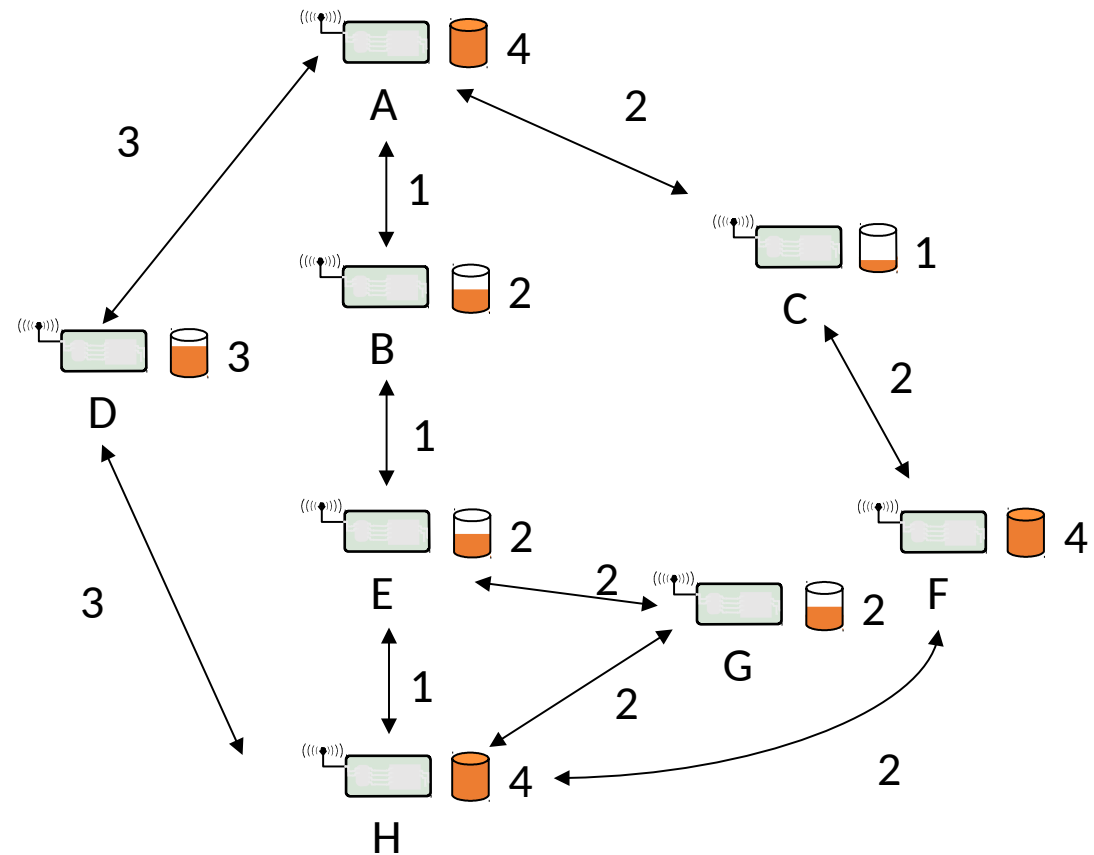
- Métrica: Eficiencia energética
- Objetivos
 - Minimizar la energía por bit
 - Ejemplo: A-B-E-H
 - Maximizar el tiempo de vida de la red
 - Tiempo hasta que el primer nodo falla, se pierda cobertura o suceda una partición
- Se ve trivial – usar métricas de enlace/camino (no cantidad de saltos y ruteo standard)



Ejemplo: Mandar datos de A a H.

Opciones básicas de métricas de camino

- Capacidad total disponible máxima
 - Métrica (Path): Suma de los niveles de batería
 - Ejemplo: A-C-F-H
- Ruteo por mínimo costo de batería
 - Métrica (Path): Suma de los niveles recíprocos de batería
 - Ejemplo: A-D-H
- Ruteo condicional con capacidad max-min
 - Solamente tomar el nivel de batería en cuenta cuando está por debajo de un nivel dado
- Minimiza la varianza en los valores de potencia
- Mínima potencia total de transmisión



Una métrica no trivial

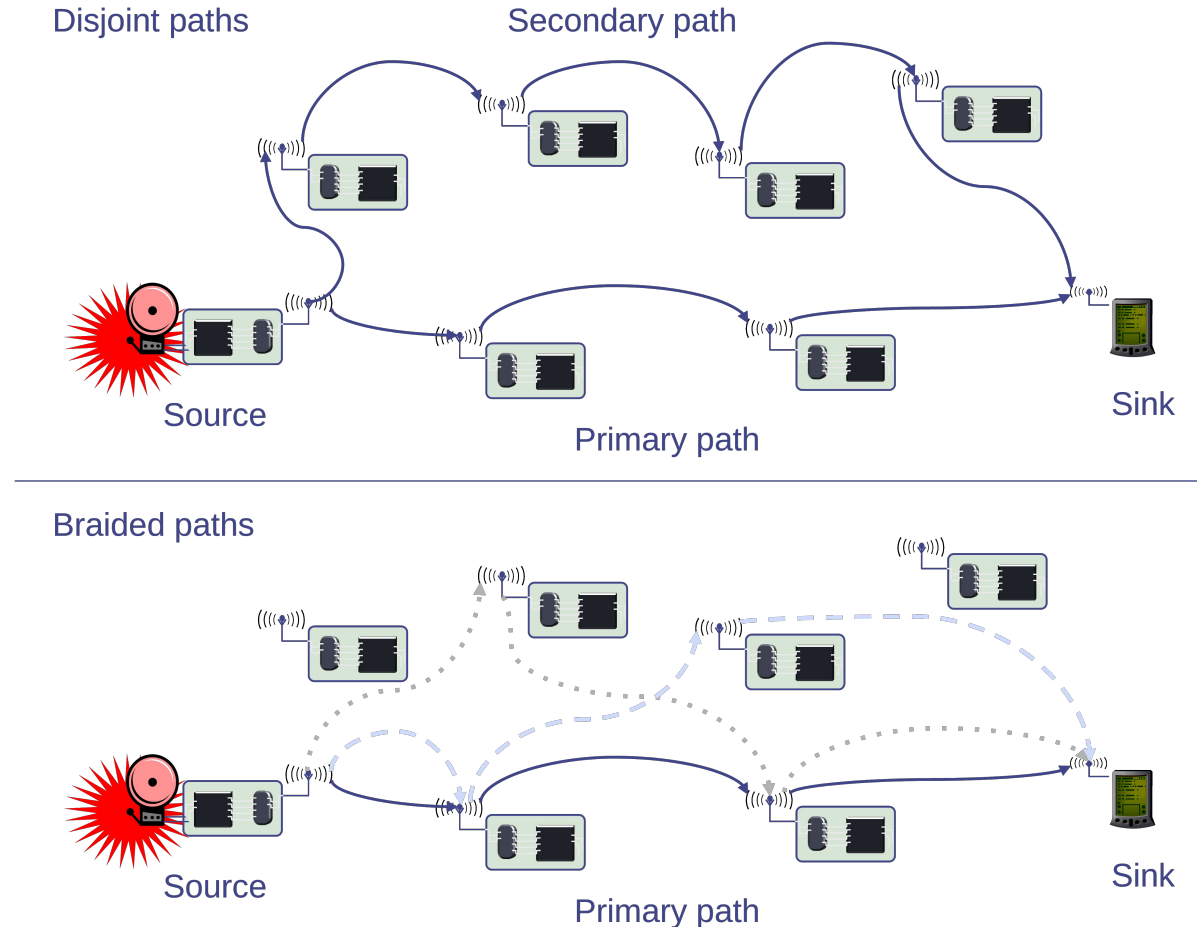
- Las métricas previas no corren particularmente bien

$$w_{ij} = e_{ij}(\lambda^{\alpha_i} - 1)$$

- Un peso de enlace no trivial:
 - w_{ij} es el pso del enlace entre el nodo i y el nodo j
 - e_{ij} energía requerida, λ una constante, α_i fracción de la batería del nodo i ya gastada
- Métrica de camino: Suma de los pesos de los enlaces
 - Usar el camino con la métrica más pequeña
- Propiedades: Muchos mensajes pueden ser enviados, tiempo de red alto.
 - Con control de admisión, incluso una relación logarítmica competitiva respecto del tamaño de la red puede ser utilizada.

Enrutamiento Multipath

- En lugar de un camino simple, puede ser útil calcular múltiples caminos entre un par origen-destino.
- Múltiples caminos pueden ser disjuntos o cruzados.
- Pueden ser usados simultáneamente, alternativamente, al azar...



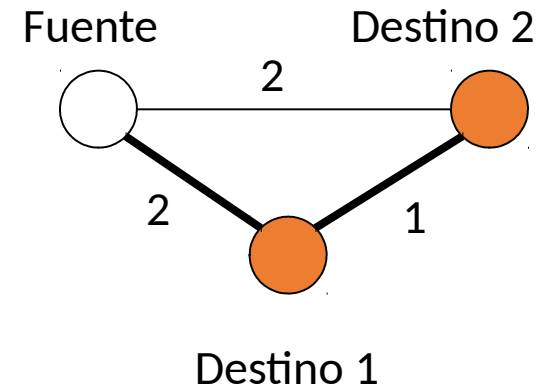
Broadcast & multicast (eficiente en energía)

- Distribuir un paquete a todos los nodos alcanzables (broadcast) o a un subgrupo (multicast)
- Opciones básicas:
 - Árbol basado en la fuente (source-based tree): Contruir un árbol (uno para cada fuente) para llegar a todas las direcciones
 - Minimiza el costo total (= suma de los pesos de los enlaces) del árbol
 - Minimiza el máximo costo a cada destino
 - Árboles compartidos, basados en el mismo tronco
 - Usa un mismo árbol para todas las fuentes
 - Cada fuente envía paquetes al árbol y luego son distribuidos
 - Malla
 - Los árboles tienen profundidad de 1: Usar mallas provee alta redundancia y robustez en ambientes móviles

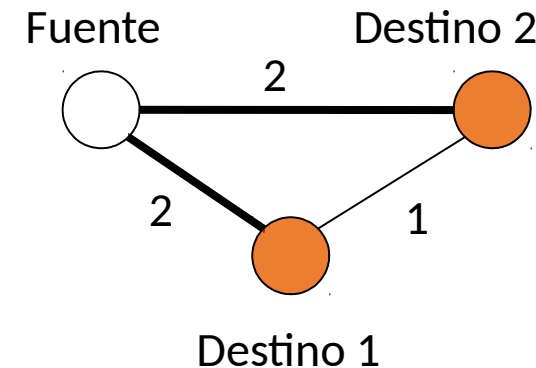
Objetivos de optimización para árboles orientados a fuente

- Para cada fuente, minimizar el costo total
 - Este es el problema del árbol de Steiner
- Para cada fuente, minimizar el máximo costo a cada destino
 - Esto se obtiene mediante la superposición de los caminos más cortos individuales calculados por un protocolo de ruteo normal.

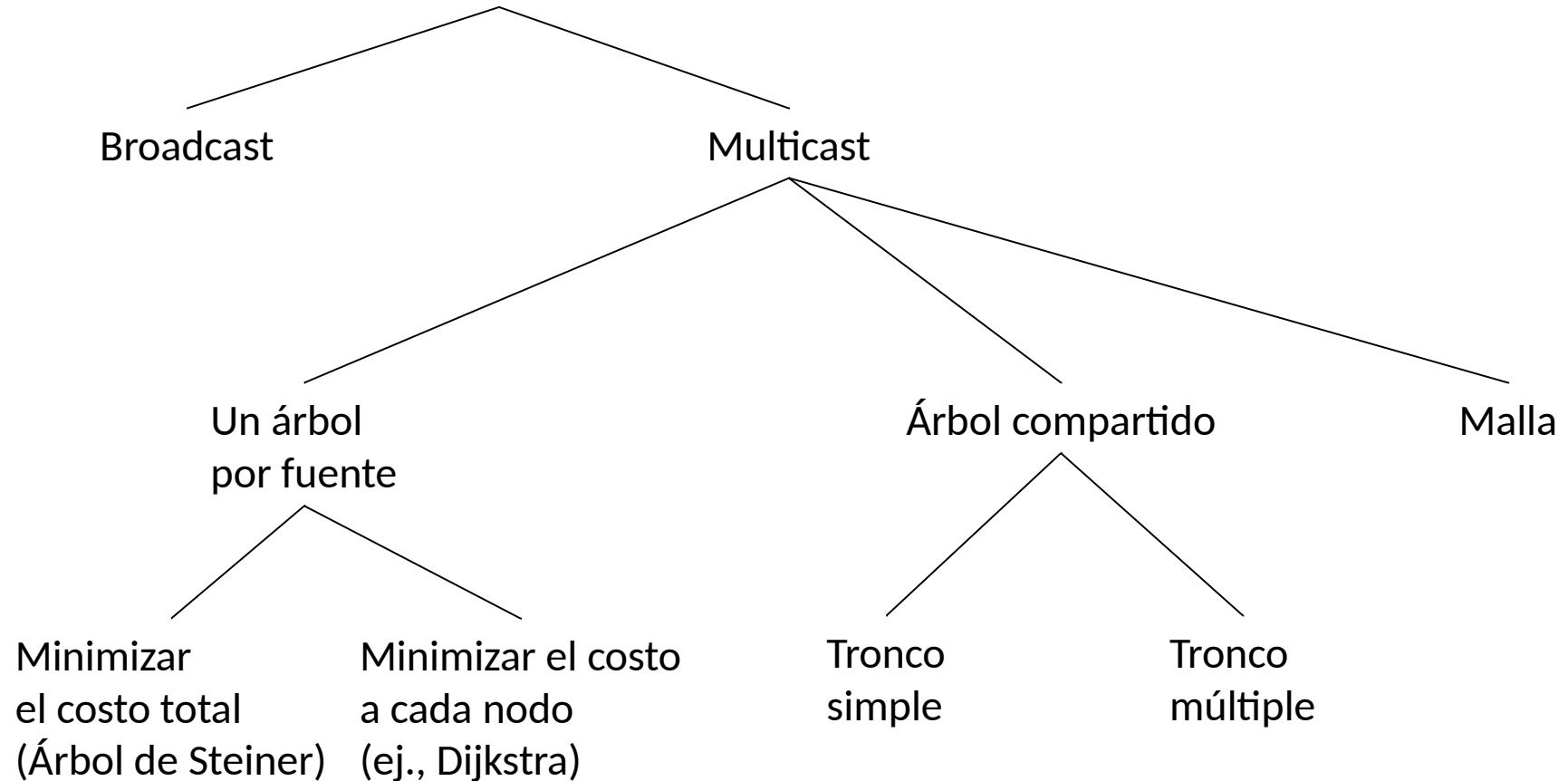
Árbol de Steiner



Árbol de camino más corto



Resumen de opciones (broadcast/multicast)



Ventaja del multicast inalámbrico

- Broad-/Multicast en el ámbito inalámbrico es distinto al cableado
 - Cableado: distribuir localmente un paquete a n vecinos: n veces el costo de un paquete unicast
 - Inalámbrico: Enviar a n vecinos puede generar costos
 - Tan altos como enviar a un vecino solo si no se consideran los costos de recibir
 - Tan altos como enviar una vez, recibir n veces, siempre que la recepción sea habilitada en el momento correcto
 - Tan alto como mandar n paquetes unicast, si el protocolo MAC no soporta multicast local

[illegible]

- Es una suposición realística.

Aproximaciones al árbol de Steiner

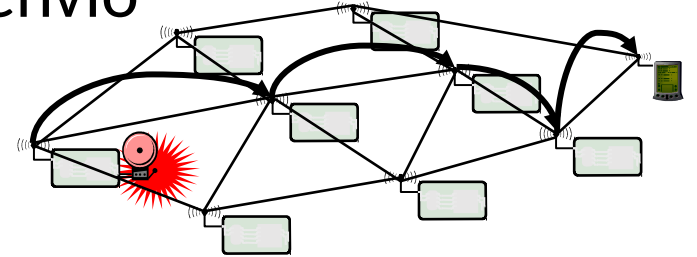
- Calcular un árbol de Steiner es un problema NP completo.
- Una aproximación simple:
 - Tomar un orden arbitrario para todos los nodos de destino + el nodo de origen.
 - Sucesivamente agregar estos nodos al árbol: Para cada nodo, construir el camino más corto a otro nodo dentro del árbol.
 - Funciona razonablemente bien en la práctica.
- Heurística de Takahashi Matsuyama
 - Similar, pero deja al algoritmo decidir cuál es el próximo nodo que será agregado
 - Empezar con un nodo origen, sumar el nodo destino al árbol que tenga el camino más corto
 - Iterar, tomando el destino que tiene el camino más corto a algún nodo que ya esté en el árbol
- Problema: La ventaja de multicast inalámbrico no se aprovecha.
 - Y además no se ajusta realmente a la definición del árbol de Steiner.

Enrutamiento Geográfico

- Las tablas de ruteo contienen información hacia dónde el siguiente salto debe ser reenviado.
- Está construido explícitamente
- Alternativa: Inferir implícitamente esta información del lugar físico de los nodos
 - Posición del nodo actual, de los vecinos, conociendo el destino. Enviar a un vecino en la dirección correcta como próximo salto.
- Opciones
 - Mandar a cualquier nodo en un área determinada **geocasting**
 - Usar información de posición para ayudar en el ruteo **position-based routing**
 - Puede necesitar un Sistema de ubicación para mapear un ID de nodo a una posición

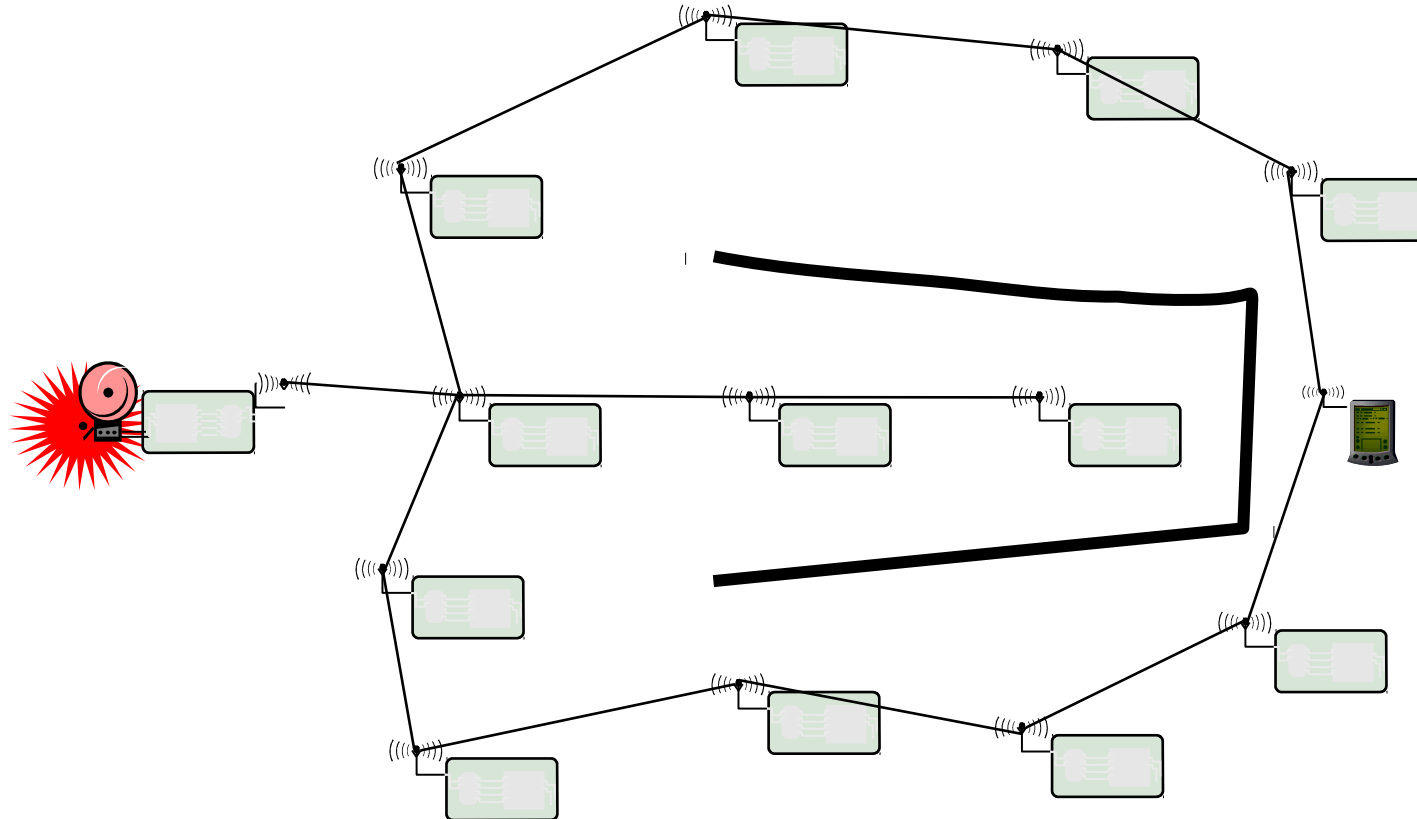
ABC del enrutamiento por posición

- “La mayoría reenvía dentro del rango r ”
 - Enviar al nodo que realiza la mayor cantidad de reenvíos hacia el destino
 - NO: El más alejado del transmisor
- El nodo más cercano con cualquier estadística de reenvío
 - Idea: Minimizar la potencia de transmisión
- Ruteo direccional:
 - Elegir el próximo salto que está angularmente cerca de su destino
 - Ellegir el próximo salto que está cerca de la línea de conexión con su destino
 - Problema: Puede generar ciclos



Problema: Calles sin salida

- Las estrategias simples pueden enviar un paquete a una calle sin salida.

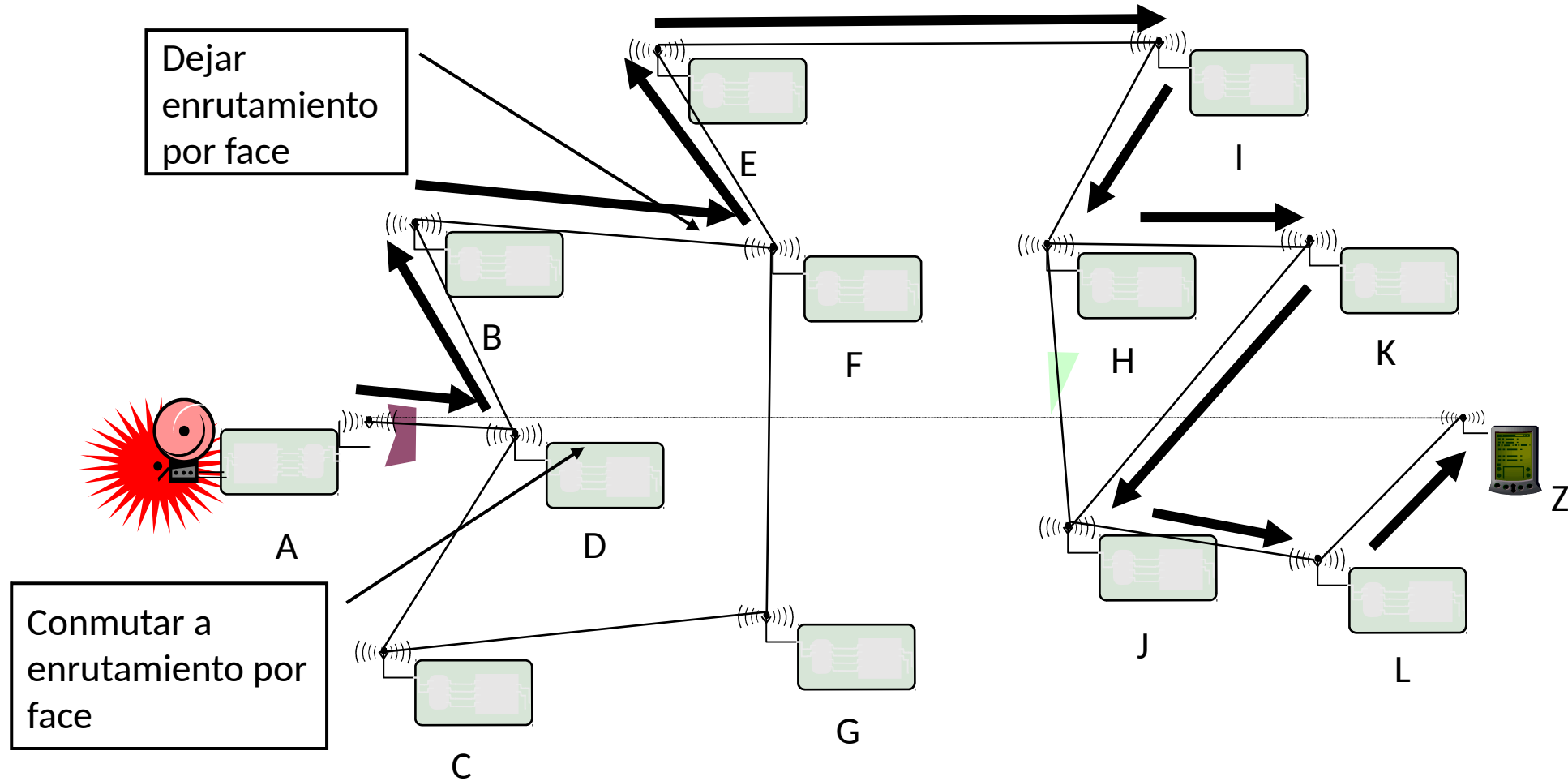


La regla de la mano derecha para evitar calles sin salida – GPSR

- Idea básica de la calle sin salida: Ponga la mano derecha en la pared, siga la pared.
 - No funciona si es una pared interna: va a caminar en círculos
 - Necesita reglas adicionales para detectar estos círculos.
- ***Geometric Perimeter State Routing (GPSR)***
 - Versiones tempranas: Compass (Brújula) Routing II, face-2 routing
 - Usa ruteo codicioso “más reenviado” tanto como sea posible
 - Si no es possible, conmuta a enrutamiento “face”
 - Face: Región lo más amplia possible del plano que no es cortada por una arista del grafo, puede ser interior o exterior.
 - Enviar el paquete alrededor de la “Face” usando la regla de la mano derecha.
 - Usar una posición donde la “Face” fue dato y la posición de destino para determinar cuándo la “Face” puede ser abandonada. En ese momento, conmutar a ruteo codicioso.
 - Requiere: un gráfico plano, algo que se puede asegurar con control de topología.

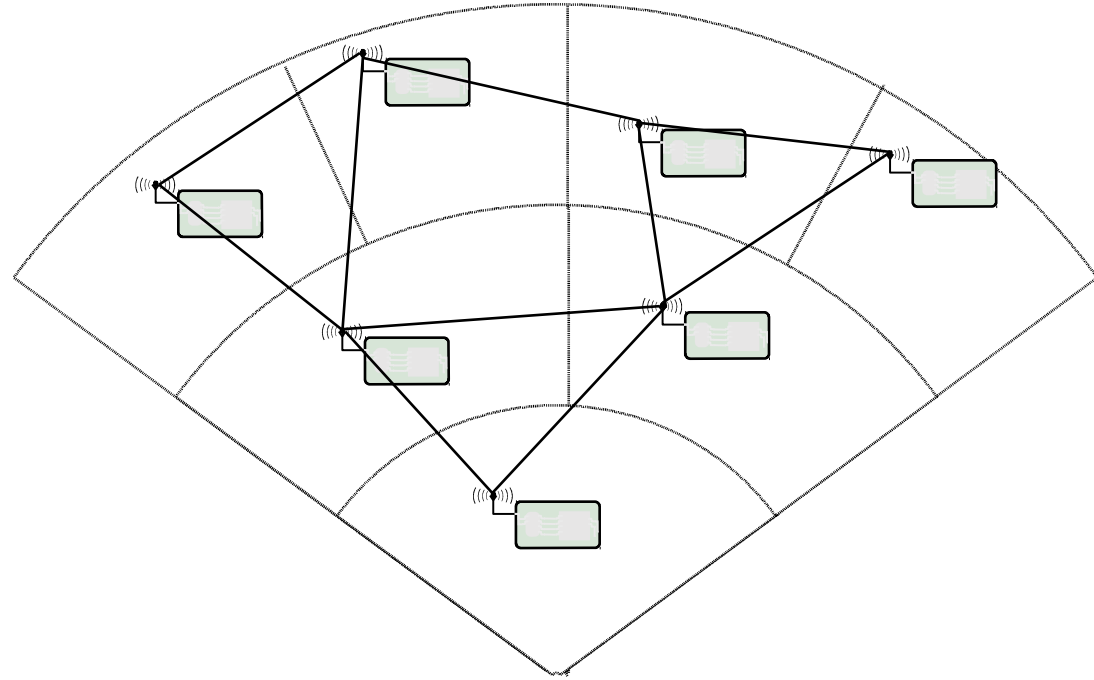
GPSR - Ejemplo

- Enrutar un paquete de A a Z



Geographic routing sin posiciones – GEM

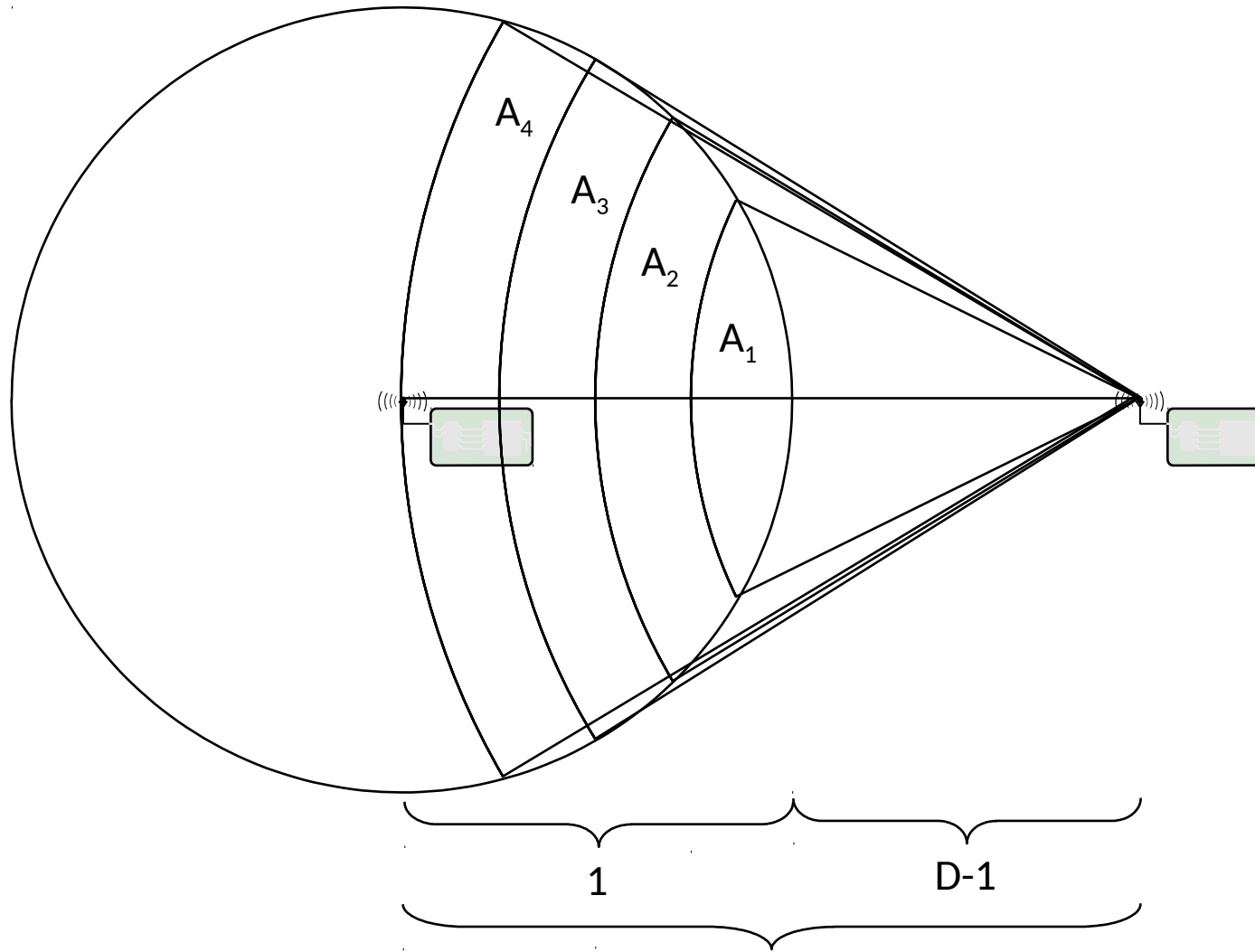
- Contradicción: Geográfico pero sin posición?
- Construir coordenadas virtuales que presean suficiente información de vecindad para ser útiles en ruteo geográfico pero que no requiere determinación real de la posición. Usar coordenadas plares desde un punto central.
- Asigna un “rango angular virtual” a los vecinos de un nodo, con un radio mayor
- Los ángulos son recursivamente distribuidos a los nodos hijos



GeRaF

- Cómo combinar el conocimiento de la posición con nodos encendiéndose y apagándose?
 - Objetivo: Transmitir el mensaje sobre múltiples saltos al nodo de destino, gestionar una topología en constant cambio debido al encendido y apagado de los nodos
 - Idea: Reenvío iniciado por el Receptor: ***Receiver-initiated*** forwarding
 - El nodo de reenvío S simplemente envía un paquete por broadcast sin especificar el nodo siguiente.
 - Algún nodo T va a tomarlo (idealmente el más cercano a la fuente) y reenviarlo.
- Problema: Cómo gestionar multiples reenviadores?
 - Aleatorización de la posición informada: Cuanto más cerca del destino está el nodo que reenvía, lo menos que duda de reenviar un paquete.
 - Usar varios anillos puede simplificar el problema, se deben agrupar los nodos de acuerdo a la distancia, aunque aún pueden existir colisiones.

GeRaF - Ejemplo



Multicast basado en la ubicación(LBM)

- Geocasting mediante inundación geográficamente restringida
- Definir una zona de “forwarding” (reenvío) – los nodos en esta zona van a reenviar el paquete para lograr que llegue a la zona de destino
 - La zona de reenvío puede estar especificada en el paquete o recalculada en el camino
 - Zona estática – El rectángulo más pequeño contiene la fuente original y la zona de destino.
 - Zona adaptive – el rectángulo más pequeño contiene al nodo de reenvío y a la zona de destino
 - Otra vez, pueden haber calles sin salida.
 - Distancias adaptivas – el paquete es reenviado por un nodo u si el nodo u está más cerca del dentro de la zona de destino que su predecesor (nodo v) (el paquete ha progresado)
- El paquete siempre es reenviado por nodos dentro de la zona de destino

Geocasting usando ruteo ad hoc – GeoTORA

- Recordar TORA: Los nodos calculan un DAG cuyo destino es un solo Sink.
- Observación: Reenvío a través del DAG todavía funciona si multiples nodos son el destino. (El grafo tiene múltiples Sinks)
- GeoTORA: Todos los nodos de la region de destino actúan como Sinks.
 - Se reenvía a través del DAG, todos los Sinks también reenvían como broadcast el paquete en la region de destino
- Nota: Esto funciona también para anycasting, donde los nodos de destino no necesariamente son vecinos.
 - El paquete es enviado a alguno (no necesariamente el más cercano) de los miembros del grupo.

Reenvío basado en trayectoria (TBF)

- Piense en término de un “agente”: Debe viajar a través de la red recolectando medidas.
 - El reenvío al azar puede tomar mucho tiempo.
- Idea: Proveer al agente con una cierta trayectoria a sobre la cual Viajar
- Descripta, por ejemplo, por una curva simple.
 - Reenvío al nodo más cercano a esta trayectoria.

