

Redes de sensores inalámbricos (RSI)

Capa de red: RPL

(Routing Protocol for Low-Power and Lossy Networks)

Leonardo Steinfeld

Inst. de Ingeniería Eléctrica, Fac. de Ingeniería
Universidad de la República (Uruguay)



Co-funded by the
Erasmus+ Programme
of the European Union



Disclaimer: The European Commission support for the production of this website does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Objetivos

- Comprender el propósito de la capa de red.
- Comprender los conceptos RPL y su funcionamiento.
- Describir las características de RPL y los diferentes tipos de mensajes de control.
- Describir el algoritmo Trickle usado en RPL.

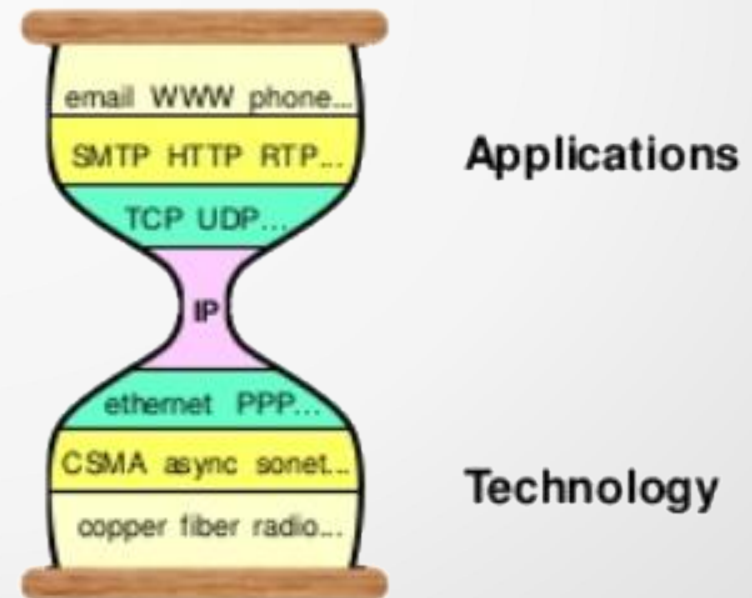
Introducción: ¿Qué es RPL (ripple)?

- RPL: **R**outing **P**rotocol for **L**ow power and lossy networks
 - Low power and lossy networks:
 - LoWPAN y otras redes
 - Múltiples medios físicos (IEEE 802.15.4, etc.)
 - Diseñado por IETF(*) sobre **IPv6**

(*) Internet Engineering Task Force

Introducción: ¿Por qué IP?

- provee conectividad “punta a punta”
- resuelve interoperatividad
- facilita el desarrollo e integración de aplicaciones (recolección de datos y configuración)
- suficientemente liviano para “pequeñas cosas”: se cayó un **mito**...



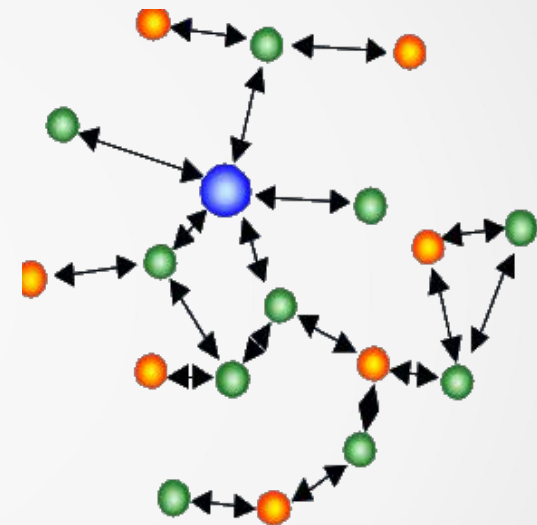
Introducción: ¿Qué son las LLN?

- Low power and lossy networks: Características
 - Nodos:
 - recursos limitados
 - procesamiento, memoria, energía
 - cantidad: ~10 a ~1000
 - Enlaces:
 - baja tasa de entrega de paquetes (low PDR)
 - baja velocidad (low data rates)
 - inestables (instability)



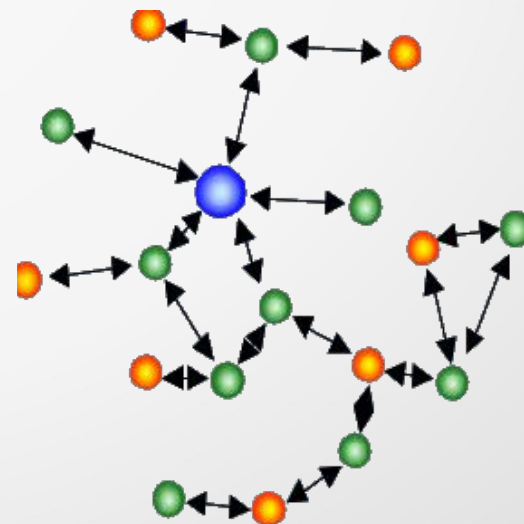
Introducción: ¿Qué son las LLN? (cont.)

- Low power and lossy networks: Características
 - Elementos de la red son:
 - sensores / actuadores
 - miden y envían sus datos
 - reciben comandos, etc.
 - routers
 - reenvían datos de otros nodos
 - sensores / actuadores + routers
- Necesidad **básica** de **encaminar** datos al nodo destino
- Característica común con LoWPAN (subconjunto de LLN)



Introducción: ¿Qué son las LLN? (cont.)

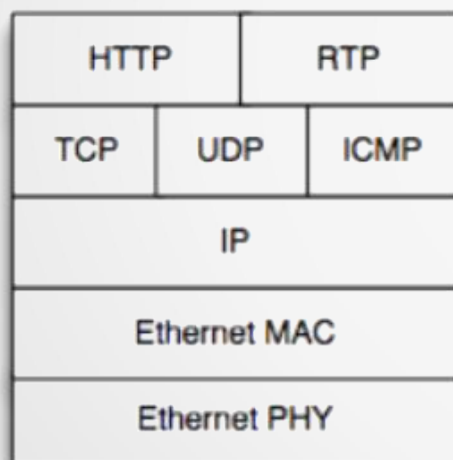
- Low power and lossy networks: Características
 - Tipos de tráfico:
 - M2P: multipoint-to-point (recolección de datos)
 - P2M: point-to-multipoint (distribución/diseminación)
 - P2P: point-to-point (entre nodos dentro de la red)



Modelo de capas en LLN

- Capa de ruteo: funciones
 - responsable de reenvío de paquetes de otros nodos

TCP/IP Protocol Stack



Application

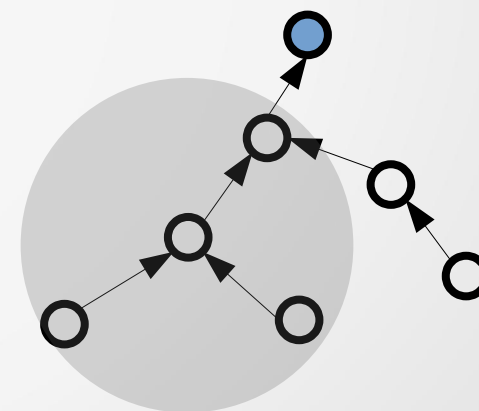
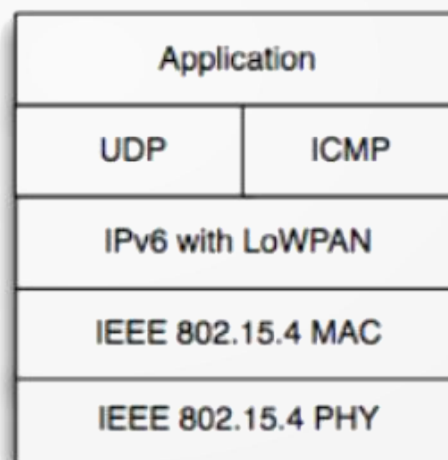
Transport

Network

Data Link

Physical

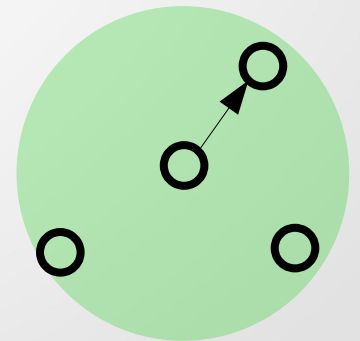
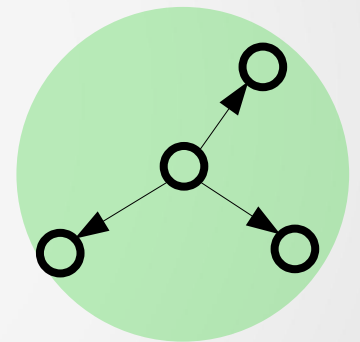
6LoWPAN Protocol Stack



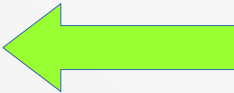
Fuente: <http://orbigo.net/2011/06/6lowpan-second-part-more-concepts-about-6lowpan/>

Capa de ruteo: premisas

- Capa enlace de datos: tipos de mensajes
 - broadcast:
 - recibidos por todos nodos vecinos (comparten link, i.e. "dentro del alcance de radio")
 - unicast
 - mensaje dirigido a un nodo específico



Tipos de ruteo

- Reactivo:
 - busca rutas cuando se necesitan
 - Ejemplos:
 - AODV: Ad hoc On Demand Distance Vector
 - DSR: Dynamic Source Routing
 - FCP: Flexible Control Protocol (sobre LPP en MAC)
- Proactivo:  RPL
 - ya tiene las rutas antes de que sean necesarias por los datos o nodos

Ruteo proactivo

- Funcionamiento
 - intercambio de mensajes de control (periódicos) para encontrar y propagar rutas en la red
 - envío de mensajes para compartir:
 - local (nivel de enlace): información de vecinos
 - global (toda la red): información de topología entre todos los nodos

Ejemplo



Ejemplo: fundamentos de ruteo

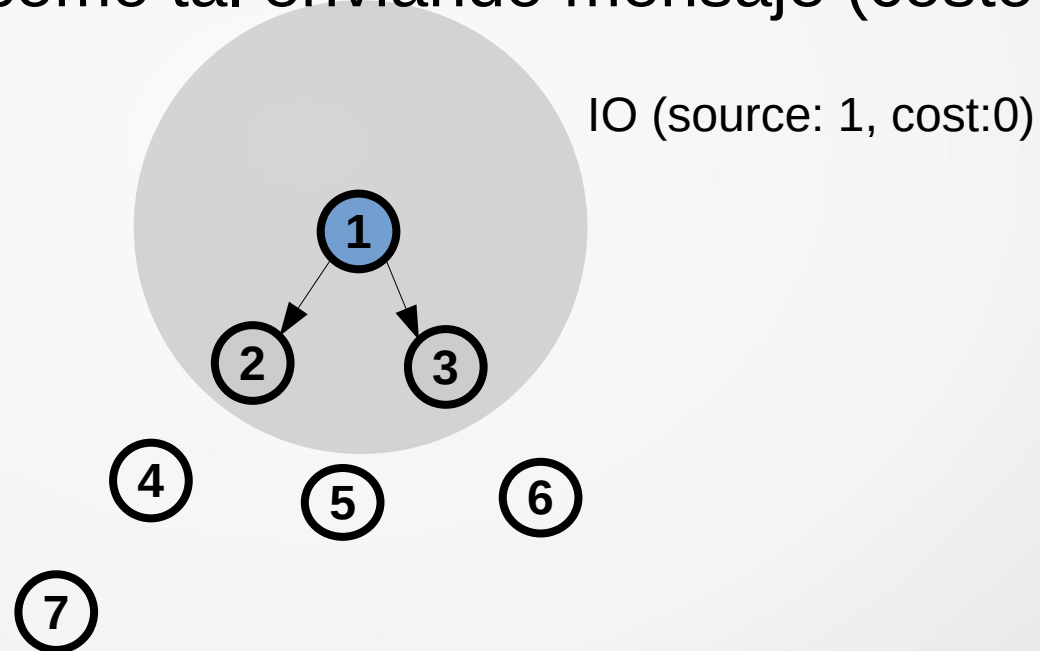
- Envío de datos a root (tipo de tráfico: M2P)
- Ruteo proactivo: basado en vector distancia (costo)
- Distancia: “costo” de envío
 - costo acumulado en el camino, e.g. número de saltos
- Mensajes intercambiados
 - IS solicitud de información (broadcast): nodo pide información de costo para llegar al root
 - IO objeto de información (broadcast): nodo anuncia su costo (se identifica), enviado en respuesta a solicitud o si tuvo cambios en su costo

Ejemplo: descripción de funcionamiento

- cada nodo
 - guarda **lista nodos vecinos** con costo asociado
 - **elige padre** para llegar al root con menor costo
- tabla de ruteo: único destino (una entrada)
 - próximo salto (padre) para llegar al root
- mensajes
 - reenviado a nodo padre

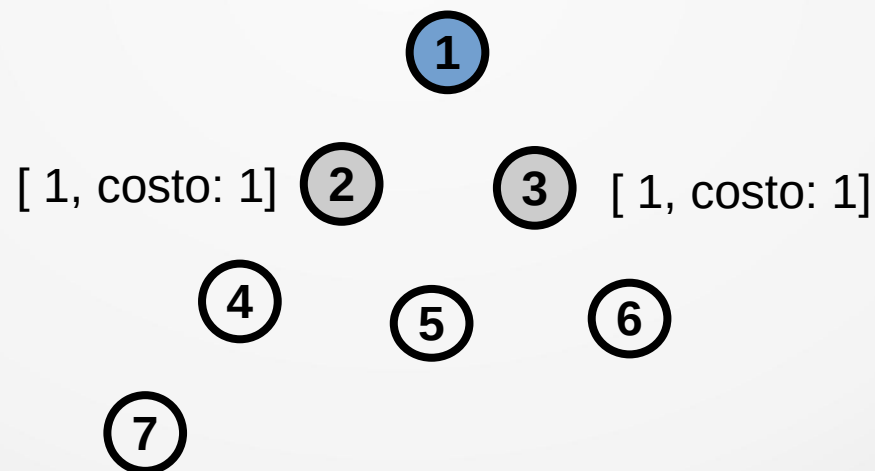
Ejemplo: fases

- Root (nodo 1)
 - manda IOs (broadcast):
 - se anuncia como tal enviando mensaje (costo cero)



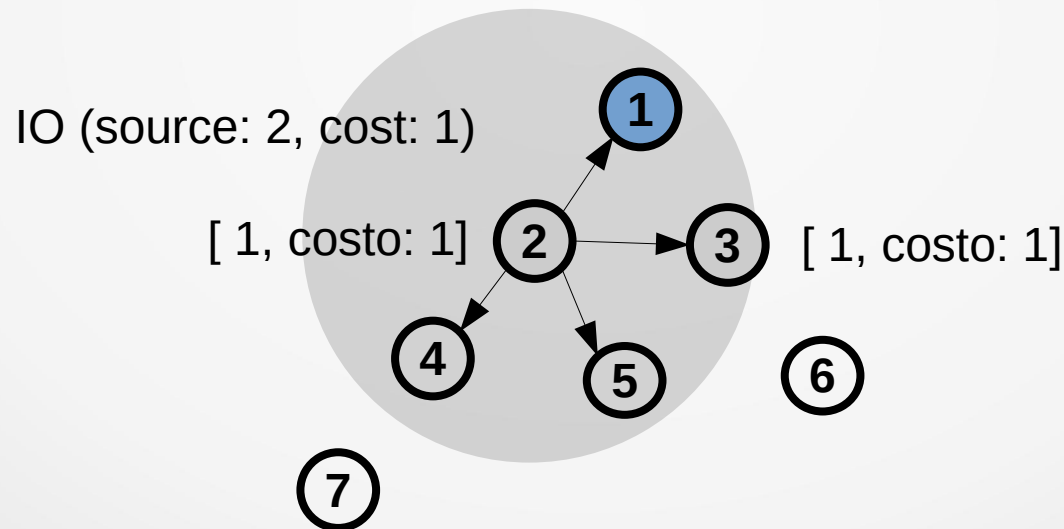
Ejemplo: fases

- Vecinos del root:
 - estiman el costo del enlace con el root (un salto)
 - agregan al root en sus tablas de ruteo con costo



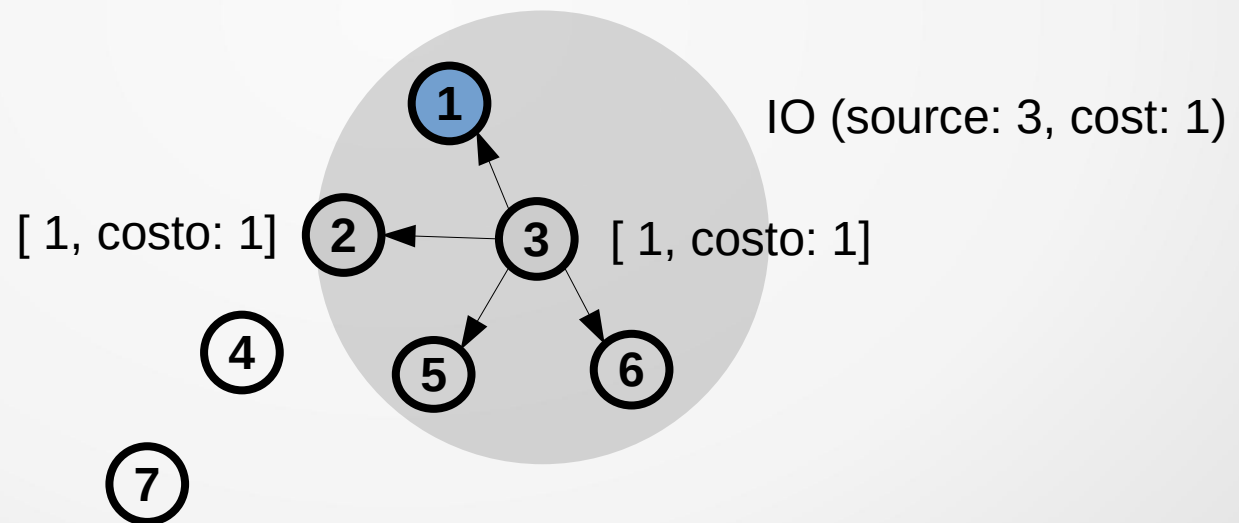
Ejemplo: fases

- Vecinos del root:
 - mandan IO (broadcast)
 - anunciando su costo de llegar al root



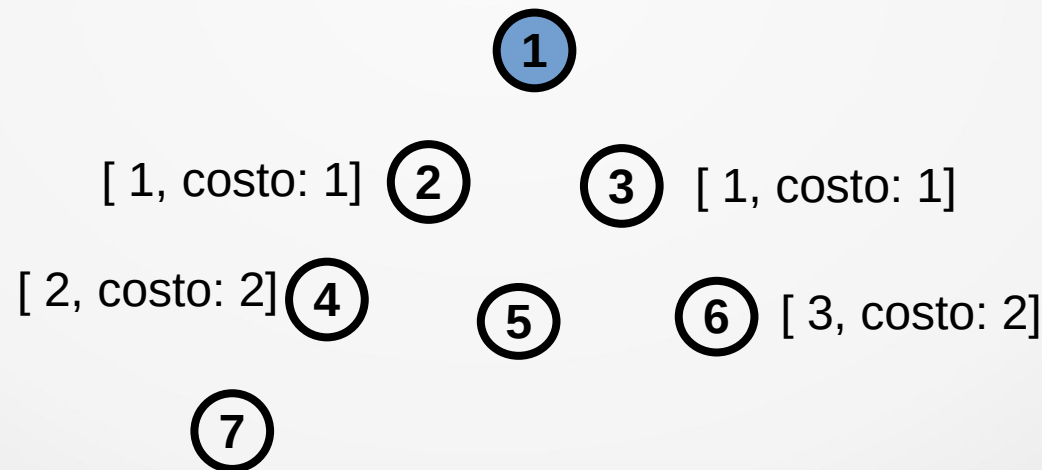
Ejemplo: fases

- Vecinos del root:
 - mandan IO (broadcast)
 - anunciando su costo de llegar al root



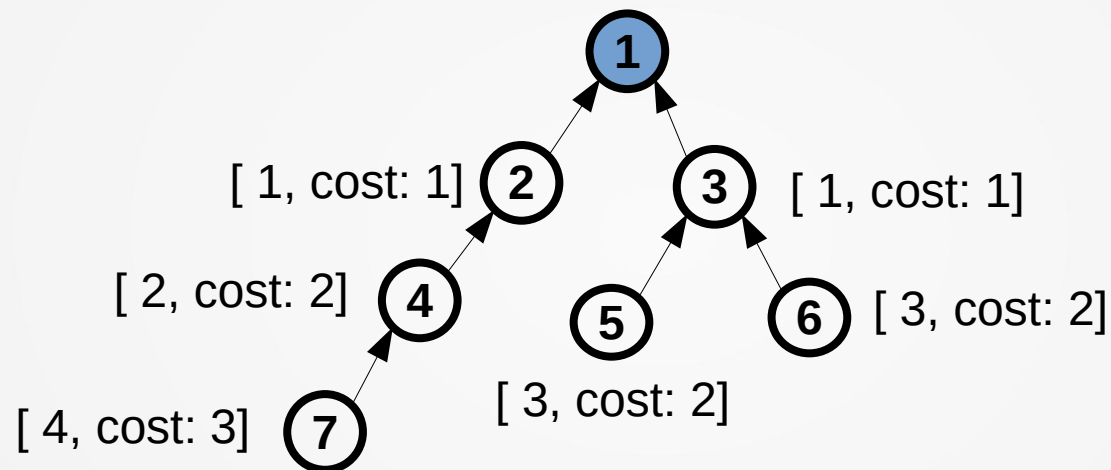
Ejemplo: fases

- Otros nodos:
 - reciben de vecinos con costos para llegar al root
 - calculan costo total (suman un salto)
 - eligen padre para mandar mensajes a root.



Ejemplo: fases

- Estado final



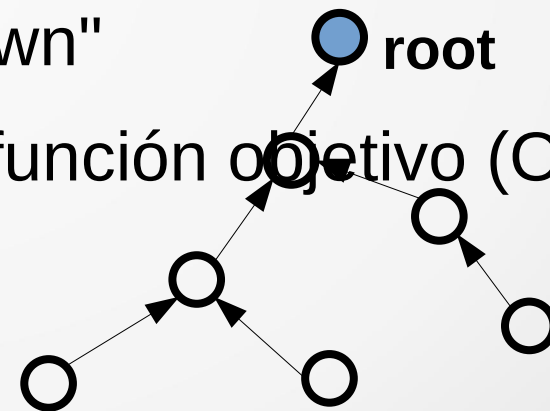
- Nota:
 - se envió un IO por nodo, condiciones estáticas de propagación
 - no consideramos IS (solicitud de información)

Observaciones

- Fundamentos del ejemplo (simplificado)
 - CTP (**C**ollection **T**ree **P**rotocol)
 - RPL (funcionalidad "route upwards", aguas arriba)
- RPL vs. CTP
 - diferencias:
 - nuevas funcionalidades: "route downwards", multicast
 - es IPv6, o sea "estándar" (IETF)
 - parecidos:
 - **algoritmo Trickle**
 - tiempo envío de broadcast adaptativo (adaptive beaconing)
 - validación de caminos (datapath validation)

RPL: Características

- Protocolo de ruteo para LLN sobre IPv6
- Medio físico: 802.15.4 u otros
- Proactivo: basado en vector distancia
- Topología tipo "árbol": construye grafo (distribuidamente)
- Trafico en direcciones "up" y "down"
- Selección de padres basado en función objetivo (OF)



RPL: Características

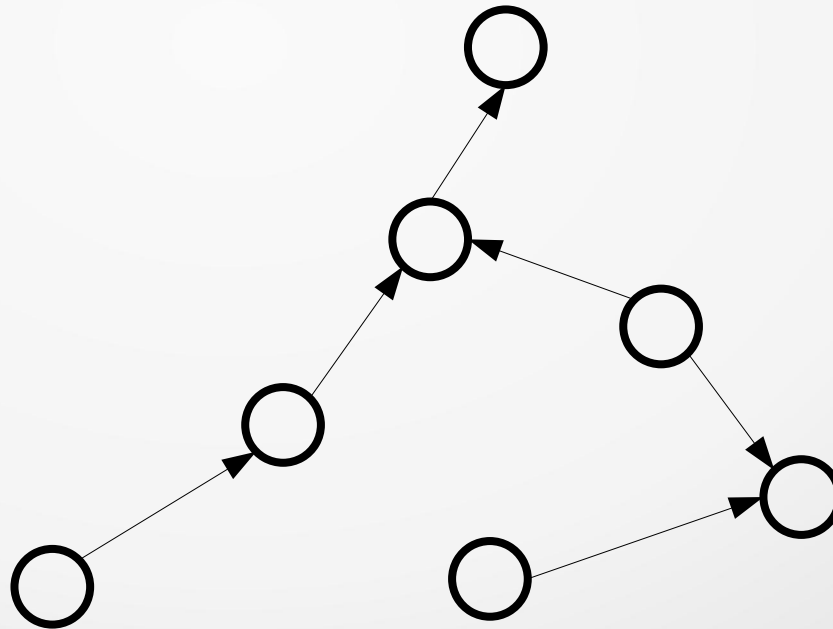
- arquitectura IP: separa las tareas de
 - procesado y reenvío de paquetes
 - encaminar paquetes en base a la tabla de ruteo
 - objetivo de optimización de ruteo
 - completar y mantener tabla de ruteo
- RPL: protocolo por vector distancia **genérico**
 - Función objetivo (OF)
 - combinación de métricas y restricciones para computar el "mejor camino".
 - permite adaptarlo a variedad de tipos de red

RPL: Definiciones y conceptos

- DAG: Directed Acyclic Graph
- DAG roots
- Destination-Oriented DAG (DODAG)
- Sub-DODAG
- Rank
- Función objetivo
- RPLInstanceID

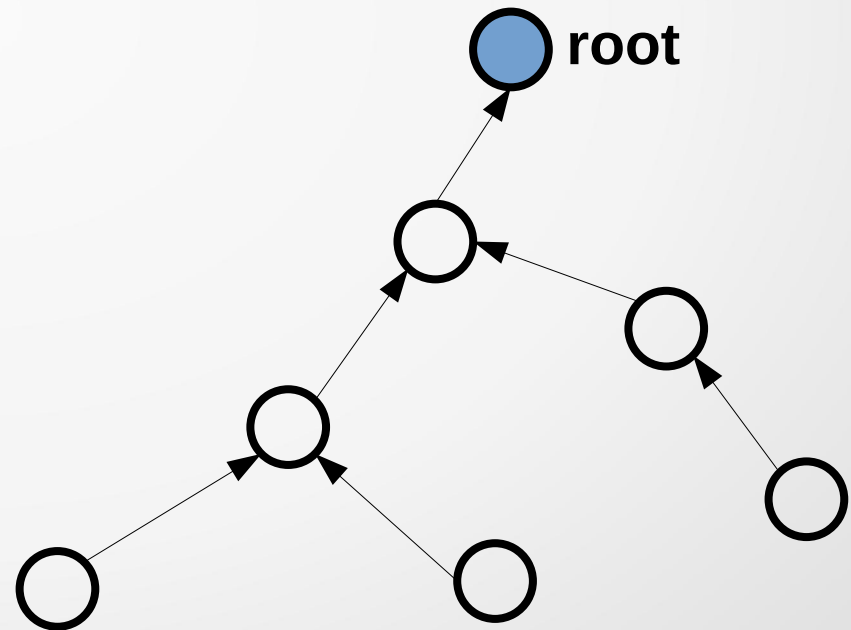
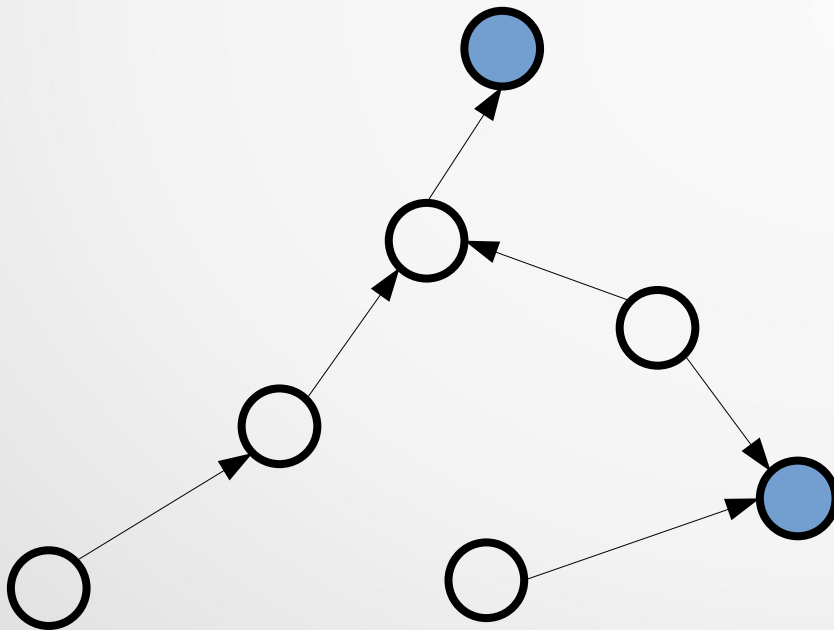
RPL: Definiciones y conceptos

- DAG: Directed Acyclic Graph
 - grafo acíclico dirigido:
 - aristas tienen dirección definida y no tiene ciclos



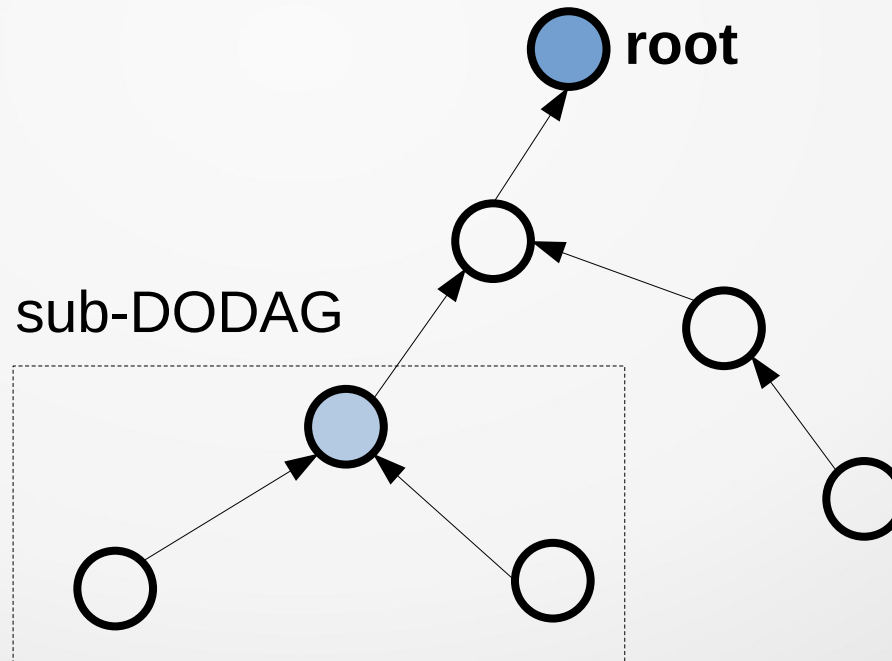
RPL: Definiciones y conceptos

- Destination-Oriented DAG (DODAG)
 - DAG con un solo root
 - border router (LBR, LowPAN BR: hacia otras redes, e.g. Internet)



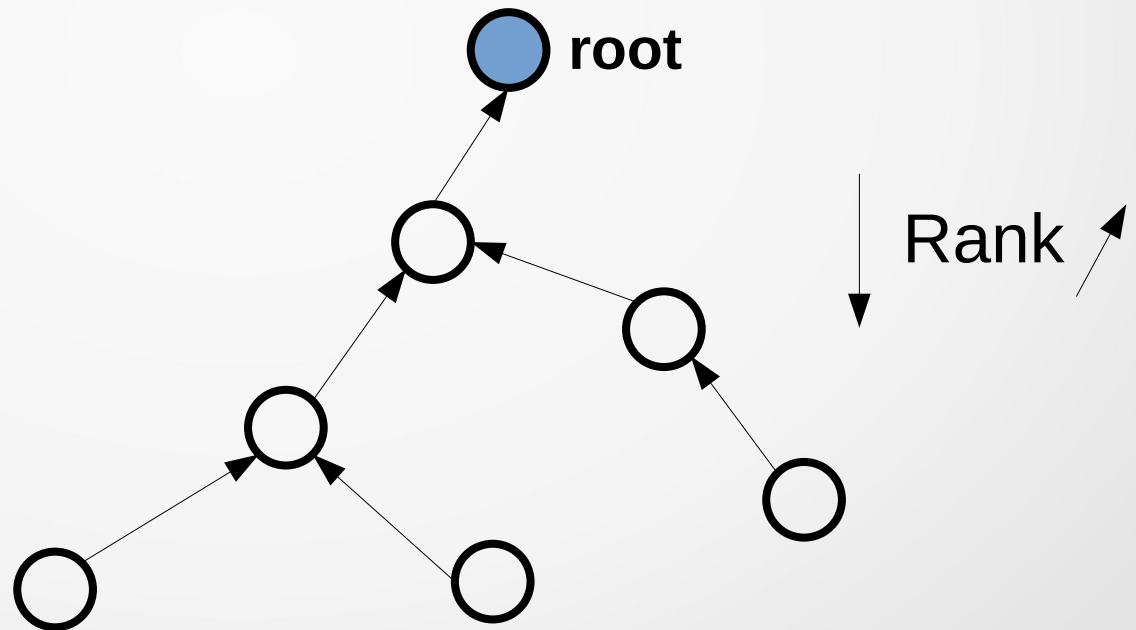
RPL: Definiciones y conceptos

- Sub-DODAG
 - definido para cada nodo
 - conjunto de nodos cuyos caminos hacia el root pasan por él.



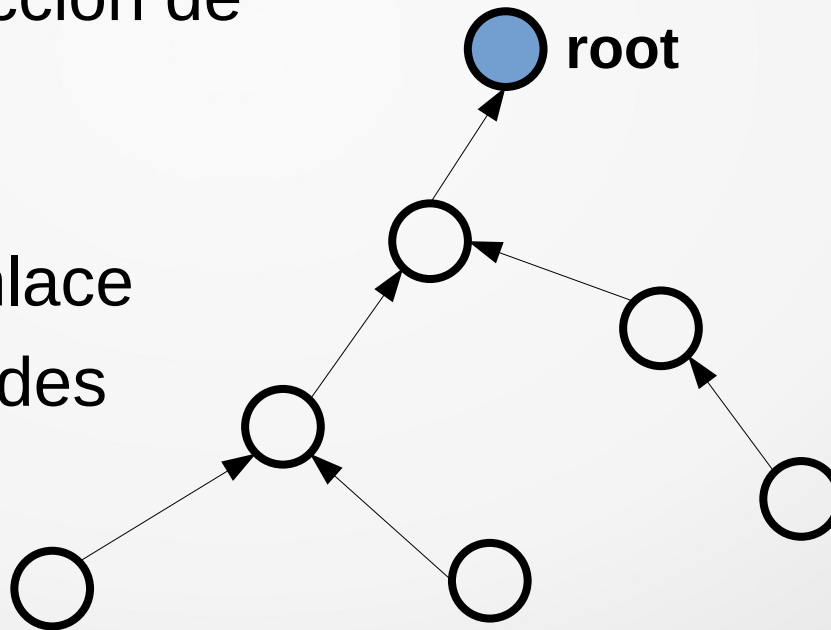
RPL: Definiciones y conceptos

- Rank (propiedad de un nodo)
 - define su posición en relación a otros respecto al root.
 - aumenta hacia abajo y decrece hacia arriba.
 - es la "distancia"



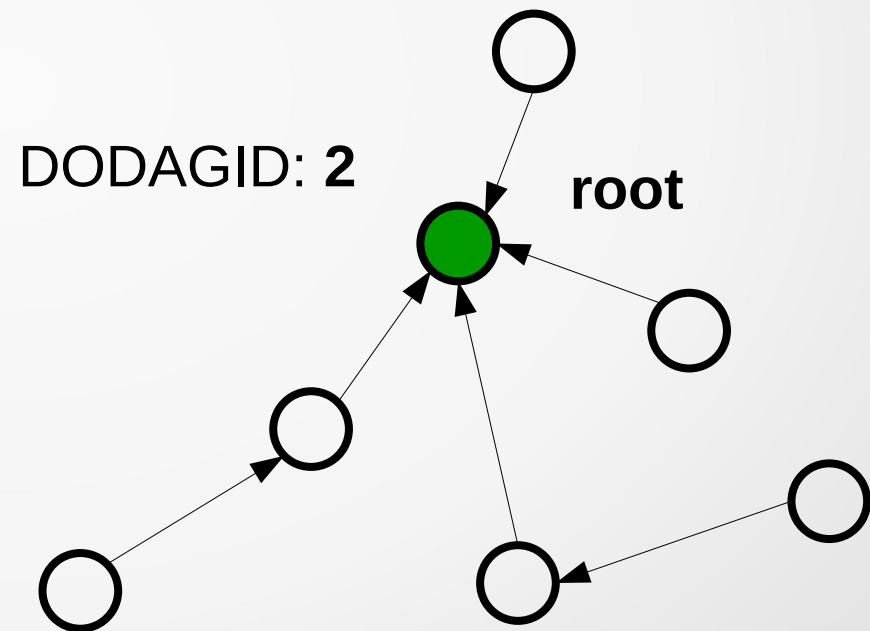
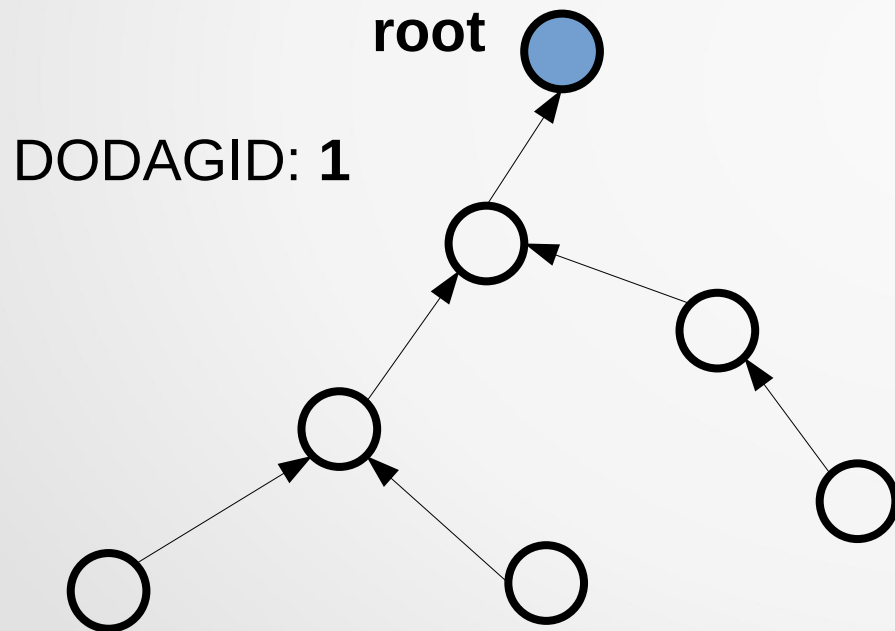
RPL: Definiciones y conceptos

- Función objetivo
 - Especifica:
 - cálculo de rank
 - reglas de selección de padres
 - Se basa en:
 - métricas de enlace
 - otras propiedades



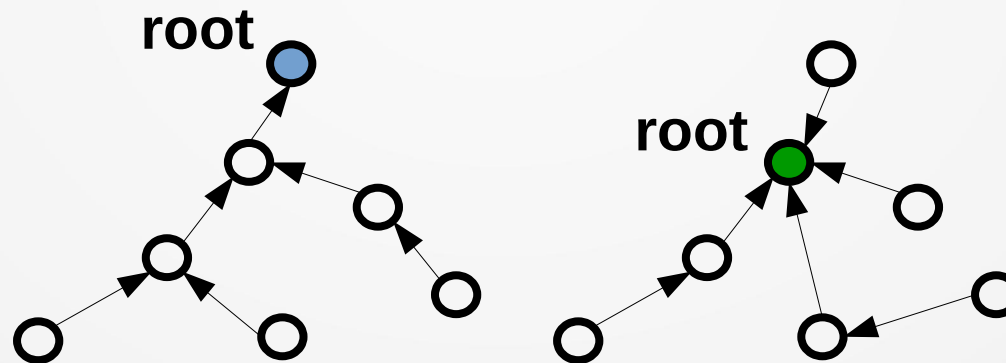
RPL: Definiciones y conceptos

- múltiples DODAG
 - **simultáneamente** en una misma red
 - identificadas por **DODAGID RPLInstanceID**



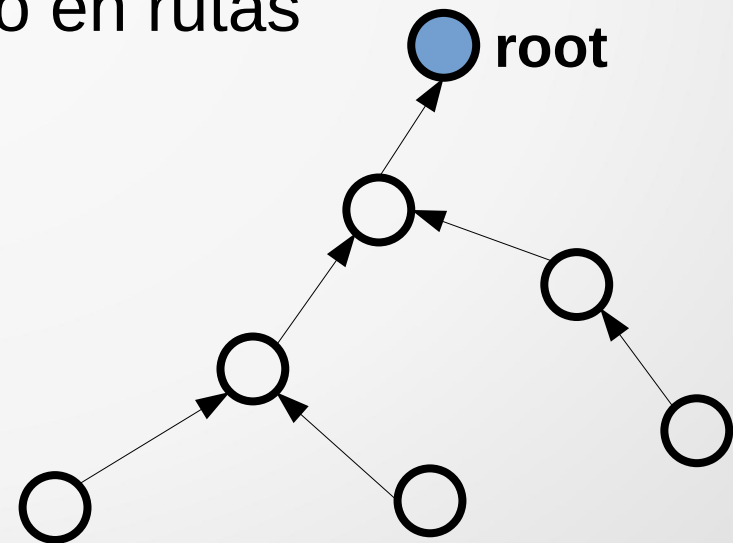
RPL: Definiciones y conceptos

- múltiples DODAG:
 - cada nodo se une y participa en un o más grafos
 - cada DODAG tiene su propio OF, métrica y root
 - para tráfico con requerimientos distintos



RPL: Características

- Paquetes de control: DIS, DIO, **DAO**
- Reglas
 - rutas hacia arriba o hacia abajo a través del DODAG
 - hacia arriba siempre hacia un rank menor
 - hacia abajo reenvía basado en rutas

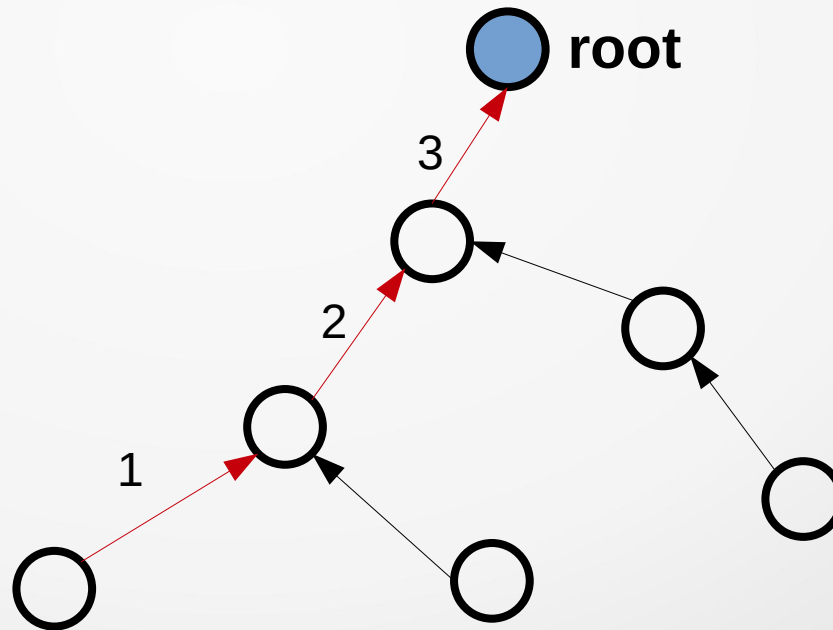


RPL: Características

- Modos de operación (MOP)
 - Non storing mode (modos 1 y 2)
 - Nodos no mantienen rutas hacia abajo
 - Storing modes (modos 3 y 4)
 - Nodo mantienen tabla de ruteo

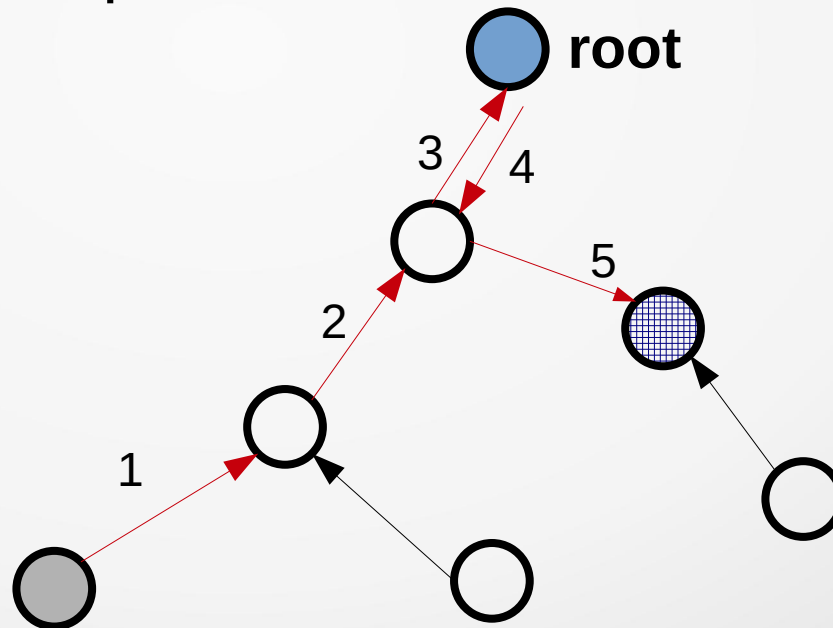
Modos de operación: non storing mode

- No downward routes maintained (modo 1)
 - solo trafico hacia el root (y otras redes a través de él)
 - una sola ruta: ruta por defecto



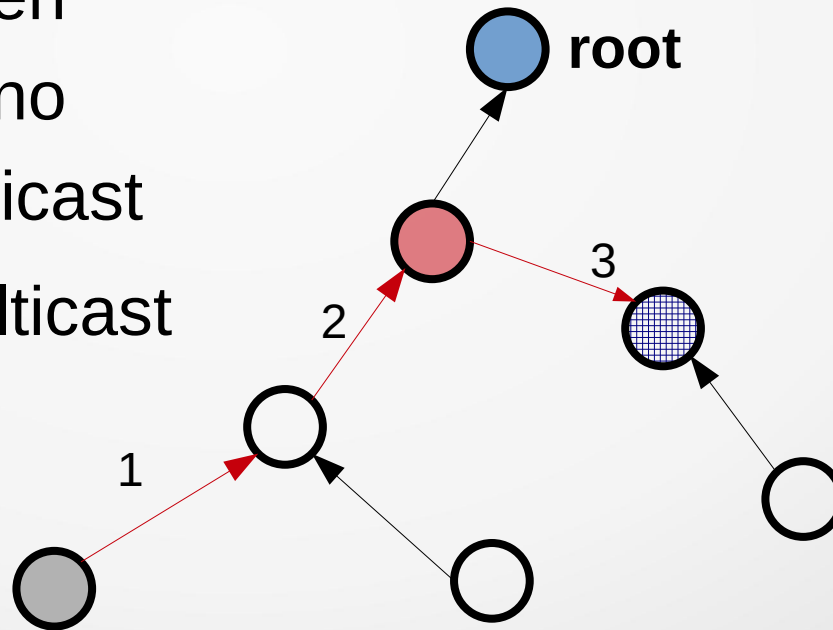
Modos de operación: non storing mode

- Non-storing mode of operation (modo 2)
 - permite tráfico hacia abajo pero desde el root y especificando la ruta completa
 - tráfico siempre pasa por root



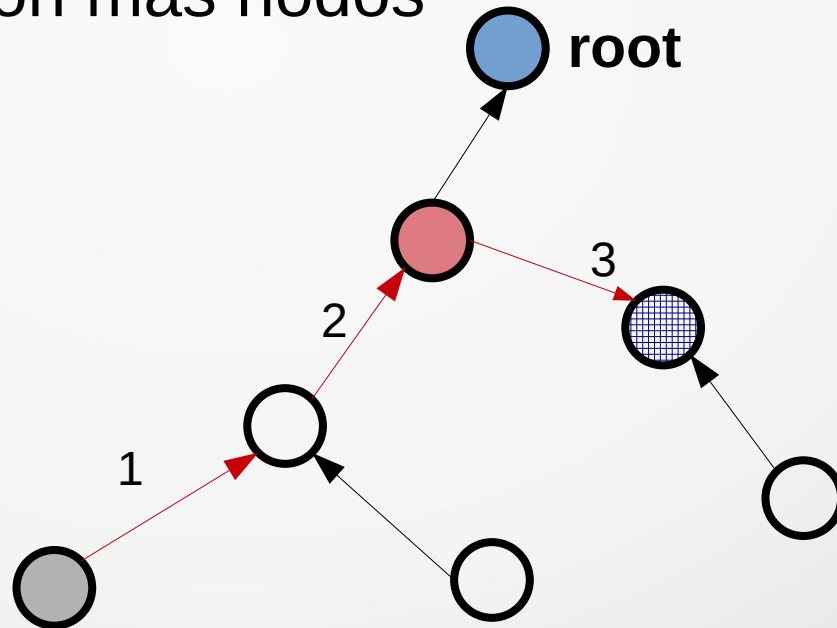
Modos de operación: storing mode

- Storing mode of operation (modo 3)
 - cada nodo guarda tabla de ruteo
 - “sube” hasta ancestro común
 - abuelo de origen
 - padre de destino
- Storing mode w/multicast
 - Idem ant. con multicast



Modos de operación: storing mode

- Observaciones (Storing mode of operation):
 - entradas para todos los nodos del sub-DODAG
 - cercanos al root tienen más entradas
 - sub-DODAG con más nodos



Métricas de ruteo

- Métricas de nodo:
 - Node State Attribute (NSA), Node Energy
 - Hop count
- Métricas de enlace:
 - Throughput, Latencia
 - Nivel de calidad de enlace (Link Quality Level)
 - ETX (expected number of transmissions)
 - número de veces que se envía un mensaje hasta ser recibido

Métricas de ruteo: ContikiRPL

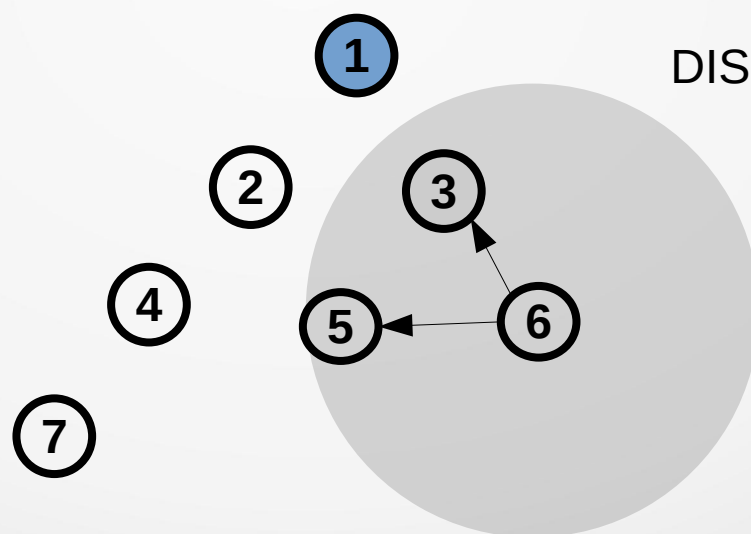
- Rank
 - costo de un camino como suma de ETX de todos sus enlaces
- Ejemplos: costo
 - 3 enlaces con PDR de 100% **3**
 - 2 enlaces con PDR de 50% (ETX: 2) **4**
- Funciones objetivos implementadas:
 - RPL's objective function 0
 - Minimum Rank with Hysteresis Objective Function (MRHOF)

Mensajes de control: propósitos

- DIO: DODAG Information Object
 - crear/formar, mantener y descubrir el DODAG
 - contiene información de la configuración del DODAG (sirve para seleccionar padres)
- DIS: DODAG Information Solicitation
 - solicitar explícitamente mensajes DIO de sus vecinos
- DAO: DODAG Destination Advertisement Object
 - propagar info de nodo hacia los nodos ancestros (padres, etc.) para soportar tráfico hacia abajo.

Mensajes de control: cuándo enviar

- DIS: DODAG Information Solicitation
 - periódicamente se chequea si conoce un DODAG, sino se envía DIS



Mensajes de control: cuando enviar

- DIO: DODAG Information Object:
 - Ejemplos:
 - se recibe un mensaje DIS
 - cambia el rank del nodo
 - el nodo cambia de padre preferido
 - repetidamente (cada tanto)...

Mensajes de control: procesamiento

- **Alogritmo Trickle:** para envío de DIO
 - Se envía repetidamente
 - cada t aleatorio, $t \in [I/2, I]$, $I \in [I_{\min}, I_{\max}]$
 - sólo si al llegar t se cumple $c < k$, k cte redundancia
 - I elegido:
 - cuando timer expira $I = I * 2$, $I < I_{\max}$
 - si algún evento indica que la red debe actualizarse $I = I_{\min}$



Mensajes de control: procesamiento

- DIO: DODAG Information Object
 - Al recibir un DIO
 - Si no conocía el DODAG
 - Se agrega remitente como padre preferido
 - Si nodo no tiene dirección IP se crea
 - Se resetea timer envío de DIO
 - Si conocía el DODAG
 - Se agrega el remitente como padre
 - Se reelige padre preferido
 - Si el remitente es nuevo padre preferido se le envía DAO

Mensajes de control: procesamiento

- DAO: DODAG Destination Advertisement Object
 - Se envía cuando:
 - cambio de padre preferido (al viejo y al nuevo)
 - se recibe
 - DIO del padre (para que actualice lifetime)
 - DAO de hijo, se reenvía a padre
 - Cuando se recibe:
 - Se actualiza tabla de ruteo (agrega o quita entrada)
 - Si se notifica que es nuevo padre.
- Observación: DAO son reenviados hacia arriba hasta el root, llenando tablas de ruteo en el camino.

Problemas a resolver

- Inestabilidad
 - origen: *greediness* (avaricia) de nodos
 - solución se limita procesamiento de DIO nodos más profundos
- Loop avoidance: chequeo de consistencias
 - Paquetes incluyen información de RPL (usando IPv6 hop-by-hop header extension)
 - Rank del origen (para comparación con receptor)
 - flag indicando dirección (upward or downward) and the Rank

Mantenimiento: mecanismos de reparación

- Local repair
 - ante una inconsistencia se busca encontrar rápidamente una ruta alternativa
 - envenena sus links (manda DIO con INFINITE_RANK)
 - vuelve a unirse
- Global repair
 - operación disparada por root
 - se incrementa DODAGVersionNumber

ICMPv6

- IPv6: next header indica ICMPv6
- ICMPv6
 - Type: RPL Control Message (155)
 - Code field:
 - DIS, DIO, DAO etc. (más versiones seguras)

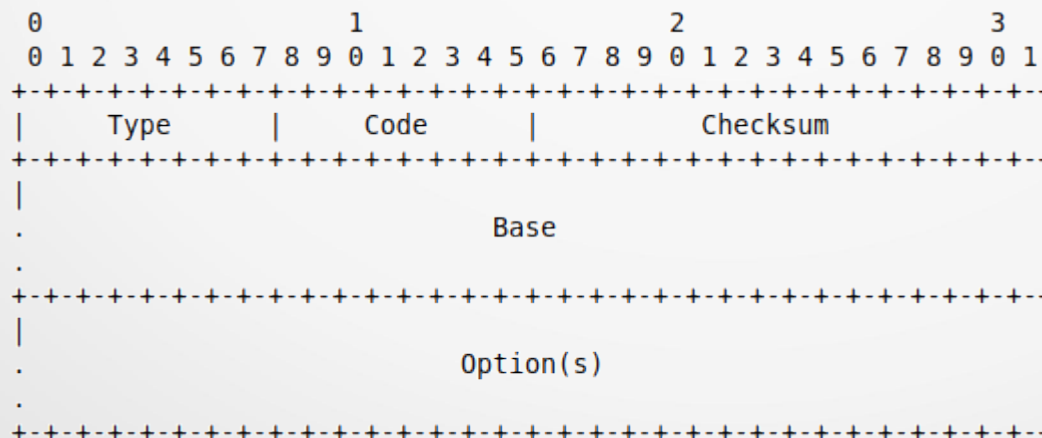


Figure 6: RPL Control Message
(fueente: RFC 6550)

IPv6: direcciones

- Repaso: multicast addresses
 - FF02::1 All nodes (link-local)
 - FF02::2 All routers (link-local)
 - FF02::1A All-RPL-nodes (link-local)

RPL en ContikiOS

- contiki/core/net/rpl/
 - rpl-conf.h parametros de configuración
 - rpl-private.h private declarartions
 - rpl.{h,c} estructuras, definciones, etc
 - rpl-dag.c mantenimiento del grafo
 - rpl-timers.c timers managements
 - rpl-ext-header.c formato de mensajes RPL
 - rpl-icmp6.c ICMPv6 RPL control messages.
 - rpl-of0.c objective function zero
 - rpl-mrhof.c MRH objective function

Normalización

- Routing over Low-power and Lossy networks (ROLL)
 - RFC 6550: RPL
 - RFC 6551: Routing Metrics for path calculation in LLN
 - RFC 6552: Objective Function Zero for RPL
 - RFC 6206: The Trickle Algorithm

Planificación clases

- 1. Introducción RSI
- 2. IPv6
- 3. Plataforma de hardware
- 4. Plataforma de software: Contiki OS I
- 5. Plataforma de software: Contiki OS II
- 6. Capa de aplicación: CoAP
- **7. Capa de red: RPL**
- 8. Subcapa MAC
- 9. IEEE 802.15.4 / 6lowpan
- 10. Capa Fisica & antenas
- 11. IoT y las RSI



FIN... ¿más preguntas?