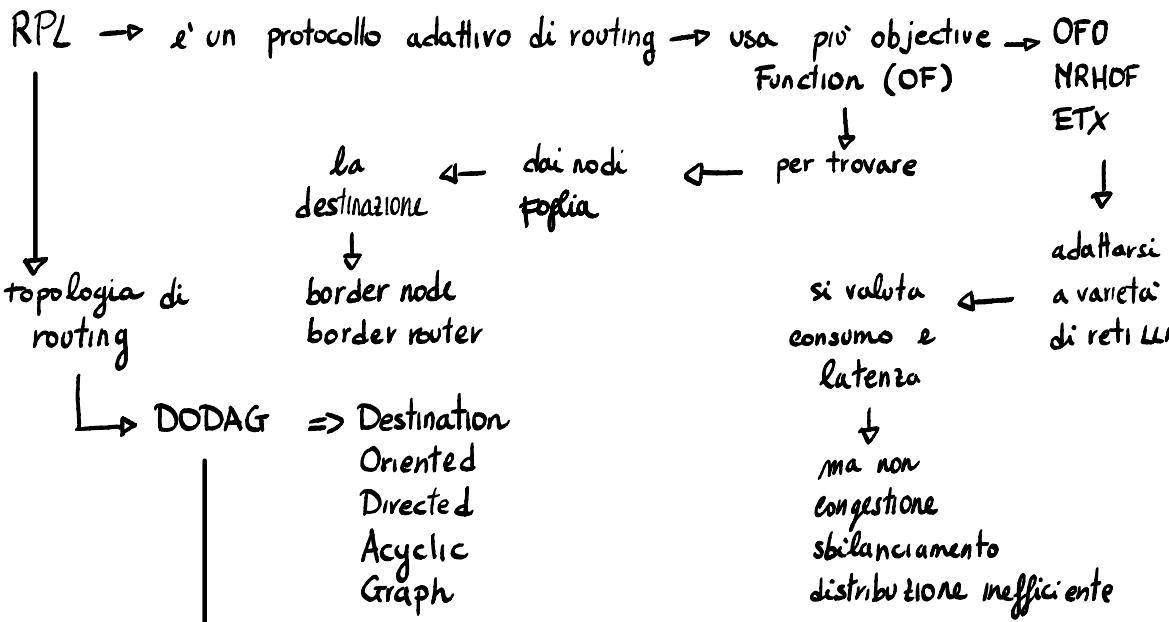
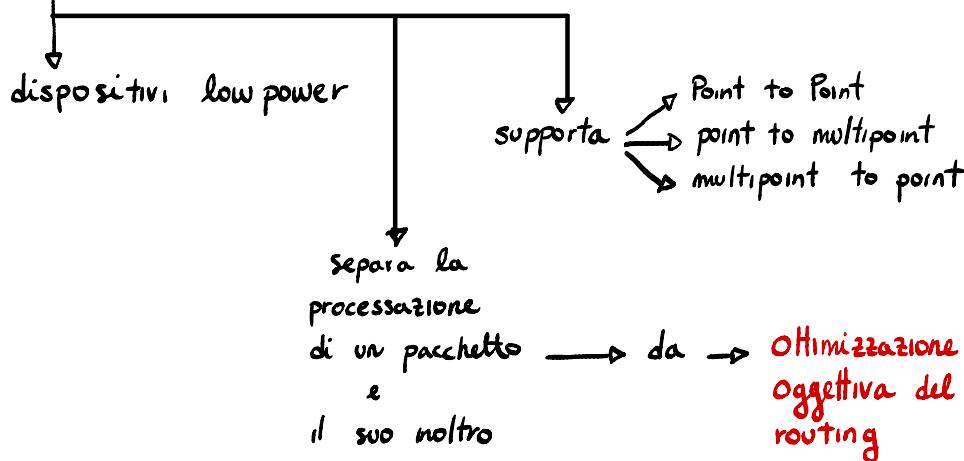


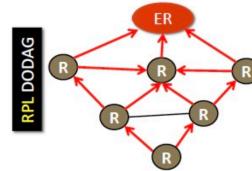
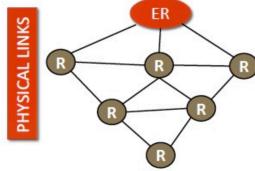
RPL

ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORK

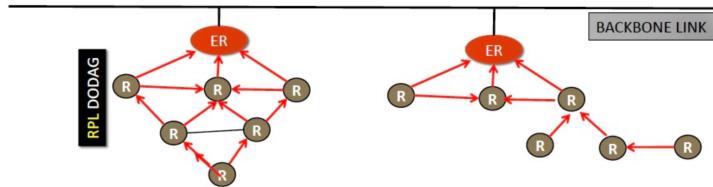
ROLL \Rightarrow serve a standardizzare protocolli di routing per app embedded come reti industriali su automazione, domotica.



grafo diretto senza cicli



In caso di extended LowPAN \Rightarrow RPL può creare più DODAG disgiunte, indirizzate ad un ER diverso



Il DODAG introduce i seguenti pacchetti di controllo:

- DIO: usato per stabilire il percorso verso l'alto.
"Ehi sono nel DODAG, ecco chi sono e come raggiungo la root"
- DAO: usato da un nodo per comunicare con i genitori che lo è raggiungibile da una certa direzione
"Se vuoi raggiungermi passa da questo nodo"
- DIS: un nodo lo invia per richiedere info sulla rete, sollecitando i nodi vicini ad inviare DIO
- DAO-ACK

DIO \Rightarrow vengono mandati in broadcast dal nodo radice e contiene il suo rank

\hookrightarrow vicini ricevono DIO, selezionano parente e ogni nodo si calcola il rango (posizione nel DODAG)

ai nodi vicini
selezionando quello con rango minore

Successivamente il nodo manda DIO → emissione di questo messaggio

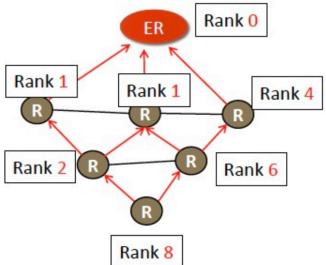
ridurre consumo potenza e

Algoritmo Trickle

DAO ⇒ viene mandato per mantenere informazioni percorso inverso

messaggi ridondanti

DIS ⇒ sollecita di avere pacchetti DIO dagli altri



numero astratto che indica → rank si ottiene la sua relativa posizione tramite Objective Function

ogni nodo contiene info routing per ogni destinazione

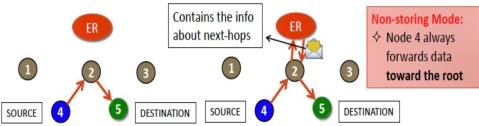
RPL → due tipi di routing

Storing Mode

Non Storing Mode

mentre le info vengono contenute nella root.

Storing Mode:
✧ Node 4 forwards data toward Node 2
✧ Node 4 stores routing info for all its subgraph (nodes 4 and 5)

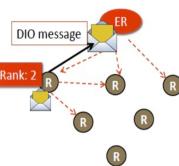


Creatione dei percorsi verso l'alto

1. The Edge router creates the DIO message, containing its rank and DODAG id, and sends it in multicast.

RECEIVING NODES

2. Each node establishes the upward link toward the sender.
3. Each node computes its own rank value, based on the root's rank and on the Objective Function.
4. Each node rebroadcasts the DIO message (following the Trickle algorithm), by including its own computed rank.



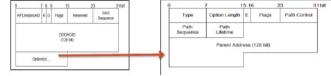
Creazione dei percorsi verso il basso:

NON-STORING MODE

→ root ha tutte info

1. Each node periodically generates a DAO message and sends it to the destination, by using the upward path established through the DIO message.

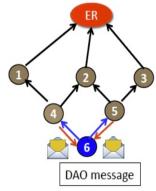
2. All the intermediate parents extend the DAO message by adding their IPv6 address in the **Transit Information Option**.



STORING MODE

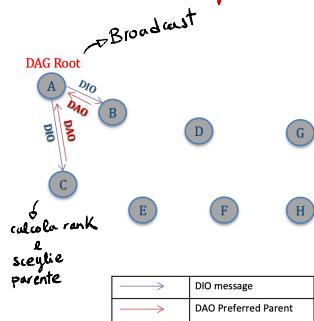
1. Each node periodically generates a DAO message and sends it to all parents node (differently to the previous case, the message is not forwarded toward the root).

2. Each parent maintains additional **routing tables** for all the nodes of its sub-DODAG.

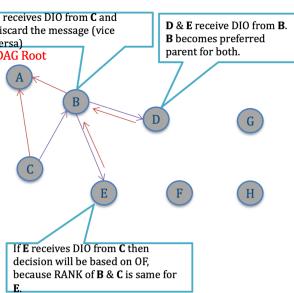


Iterazione per creare percorso:

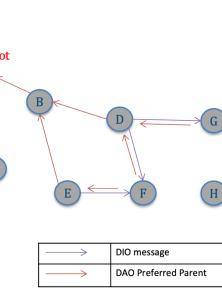
1)



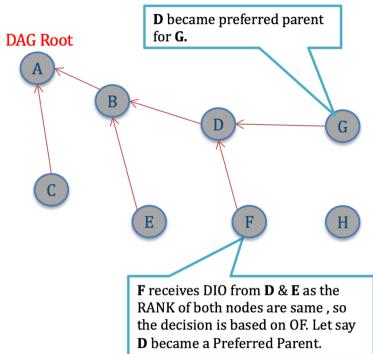
2)



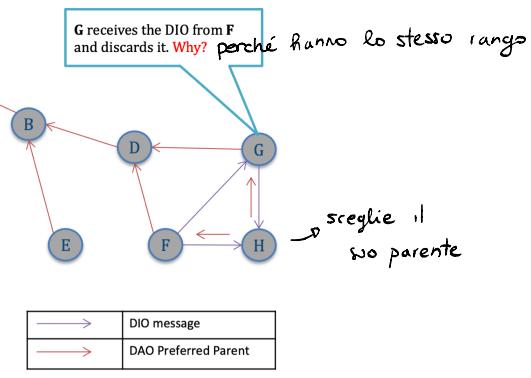
3)



4)



5)



Algoritmo Trickle \Rightarrow i nodi RPL trasmettono i messaggi DIO usando un timer DIO

es arrivo
di messaggio DIO
con DODAG num
nuovo

↓
incrementa l'intervallo o resta stabile se → variabili:
si scopre transmissioni consistenti

ci sono 3 parametri

→ dim intervallo minimo I_{min}
→ dim intervallo massimo I_{max}
→ costante ridondante K

I: dim intervallo corrente

t: tempo nell'intervallo corrente

D (contatore)

segue 5 regole:

1. All'inizio si setta un valore tra $[I_{\min}, I_{\max}]$, $c=0$ e t inizia random da $\frac{I}{2}$ e I
2. Se si ha una trasmissione consistente $c=c+1$
3. Al tempo t , Trickle trasmette DIO se $c < K$ (serve a limitare ridondanza)
4. Se intervallo scade \Rightarrow raddoppia $I \Rightarrow$ vuol dire che rete è stabile
5. Se Trickle ascolta trasmissione inconsistente e I è più grande di I_{\min} fa quello allo step 1.

Objective Function \Rightarrow definisce specifiche metriche e vincoli per trovare i percorsi a costo minimo

→ queste si occupano di:

- calcolare rank
- selezionare parenti
- calcolare il costo del cammino

→ 2 tipo di Objective function sono:

~ OF0: Object function Zero \Rightarrow conta gli hop come metrica

~ OF1: Minimum Rank with Hysteresis Objective Function \Rightarrow seleziona i percorsi che minimizzano una metrica \Rightarrow ETX

usa range per decidere \rightarrow più basso è
hop successivo \rightarrow più vicino
alla radice

$$PRR(\rho) = \frac{\text{Number of received packets}}{\text{Number of sent packets}}$$

$$ETX = \frac{1}{PRR_{down} \cdot PRR_{up}}$$

→ più è alta
peggio è

↓
funziona quando
la qualità della
connessione è critica

↓
metrica di
link su tentativi
di trasmissione
si prende
il minimo

Se arriva un altro DIO \Rightarrow piccole variazioni
sulla metrica
↓
Hysteresis \Leftarrow threshold \rightarrow per evitare di
rifare la selezione
della route