



# Testbench comparatiu del protocol RFC6550 storing mode envers no-storing mode

Implementat al sistema operatiu OpenWSN

**Marcos del Rio Sanz**

Màster Universitari en Enginyeria de Telecomunicació  
Telemàtica

**Jose Lopez Vicario**

**Xavi Vilajosana Guillen**

25 de Maig de 2016



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Testbench comparatiu del protocol RFC6550 -- storing mode envers no-storing mode</i> <i>Implementat al sistema operatiu OpenWSN</i>
<b>Nom de l'autor:</b>	<i>Marcos del Rio Sanz</i>
<b>Nom del consultor/a:</b>	<i>Jose Lopez Vicario</i>
<b>Nom del PRA:</b>	<i>Xavi Vilajosana Guillen</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>06/2016</i>
<b>Titulació o programa:</b>	<i>Pla d'estudis de l'estudiant</i>
<b>Àrea del Treball Final:</b>	<i>TFM-Telemàtica</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	RPL, IoT, OPENWSN
<p><b>Resum del Treball (màxim 250 paraules):</b> <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i></p>	
<p>Les xarxes IoT estan en procés d'evolució. En aquest treball es donarà un petit pas, implementant el protocol de routing RPL(RFC6550) en mode Storing, sobre el sistema OPENWSN amb equipament simulat.</p> <p>Actualment en el sistema OPENWSN només es troba implementat el mode Non-Storing, el qual utilitza com a routing el Source-Routing. Amb el nou mode de treball del protocol es vol establir un nou sistema de routing on tots els components de una xarxa IoT tindran una Taula de Routing amb la qual podran participar activament en el procés de routing.</p> <p>Amb un Testbench, dissenyat específicament per mesurar el rendiment d'un protocol d'enrutament, es faran mesures de valors tals com la latència, el ample de banda, el consum energètic, i demés utilitzat pel protocol. Amb aquestes dades es generarà un document on mitjançant taules comparatives i gràfiques de les dues DEMOS es farà un anàlisi dels resultats obtinguts.</p> <p>Finalment es plasmaran les conclusions obtingudes sobre les diferències de comportament entre els dos modes de treball del protocol RPL.</p>	

**Abstract (in English, 250 words or less):**

IoT networks are actually in evolution. This work will provide a small step, implementing RPL routing protocol (RFC6550) on Storing mode, on the OPENWSN system with simulated hardware.

Currently only Non-Storing mode is implemented on the OPENWSN platform, which uses Source-Routing for routing. With the new protocol working mode we want to set a new routing system where all the components of an IoT network will have a Routing Table with which they'll can actively participate in the routing proces.

With a testbench, specifically designed to measure the performance of a routing protocol, values such as latency, bandwidth, energy consumption, and others used by the protocol are mesured. With these data will generate a document using comparative tables ans graphs of the two DEMOS having an analysis of the results.

Finally conclusions are set to conclude with the differences in behavior between the two working modes RPL protocol.

# Índex

1. Introducció .....	3
1.1. Context i justificació del Treball .....	3
1.2. Objectius del Treball .....	3
1.3. Enfocament i mètode seguit .....	3
1.4. Planificació del Treball.....	4
1.5. Breu sumari de productes obtinguts .....	7
1.6. Breu descripció dels altres capítols de la memòria.....	7
2. Disseny conceptual de la solució.....	8
2.1. Xarxes IoT .....	8
2.3. Context del Treball dintre de les xarxes IoT .....	10
2.4. Protocol RPL .....	11
2.5. Processos per l'establiment de la xarxa DODAG .....	16
2.6. Disseny de la implementació .....	26
3. Disseny del Testbench .....	31
3.1. Incís al Testbench.....	31
3.2. Mesures generals a prendre.....	32
3.3. Mesures específiques pel protocol RPL.....	32
3.4. Efectes a simular en les topologies .....	33
3.5. Topologies escollides .....	33
3.6. Pla de treball.....	35
3.7. Mesures obtingues – Anàlisi del comportament de la xarxa.....	36
4. Documentació DEMO-1 Storing .....	36
4.1. Incís a la DEMO-1 .....	36
4.2. Tasques a realitzar en la DEMO.....	36
4.3. Implementació en el codi – Punts a tenir en compte .....	37
4.4. Anàlisi del codi.....	38
4.5. Codi desenvolupat - GitHub.....	38
4.6. Resolució dels punts - Resum .....	38
4.7. Resolució dels punts - Exposició .....	39
4.8. Testbench -- Topologia 1 .....	50
4.9. Testbench -- Topologia 2 .....	53
4.10. Testbench -- Anàlisi .....	55
5. Documentació DEMO-2 Non-Storing.....	57
5.1. Incís a la DEMO-2 .....	57
5.2. Tasques a realitzar en la DEMO.....	57
5.3. Testbench -- Topologia 1 .....	57
5.4. Testbench -- Topologia 2 .....	60
5.5. Testbench -- Anàlisi .....	62
6. Estudi comparatiu de rendiment.....	63
6.1. Incís al estudi comparatiu .....	63
6.2. Tasques a realitzar .....	63
6.3. Anàlisi comparatiu – Topologia 1 .....	64

6.4. Anàlisi comparatiu – Topologia 2.....	72
6.3. Anàlisi comparatiu – Conclusions.....	79
7. Conclusions.....	81
8. Glossari.....	82
9. Bibliografia.....	84

## Índex de Figures

Figura 1 - Primera Planificació.....	6
Figura 2 - Segona planificació.....	7
Figura 3 - Xarxa DODAG.....	12
Figura 4 - Topologia, estructura i relacions.....	13
Figura 5 - Missatge DIO.....	16
Figura 6 - Missatge DIO – Procés inicial.....	17
Figura 7 - Missatge DIO – Procés general.....	18
Figura 8 - Missatge DIO – Topologia establida.....	18
Figura 9 - Missatge DIO – Base del Missatge.....	19
Figura 10 - Missatge DAO.....	20
Figura 11 - Missatge DAO – Enviament.....	21
Figura 12 - Missatge DAO – Table-Routing.....	21
Figura 13 - Missatge DAO – Source-Routing.....	22
Figura 14 - Missatge DAO - Base del Missatge.....	22
Figura 15 - Missatge DAO – Transit Option.....	23
Figura 16 - Missatge DAO – Target Option.....	24
Figura 17 - Missatge DAO – No-Path.....	25
Figura 18 - Disseny de la implementació – Missatges DIO & DAO.....	26
Figura 19 - Disseny de la implementació – Missatges DIO – Procés inicial.....	27
Figura 20 - Disseny de la implementació – Missatges DIO – Procés general.....	28
Figura 21 - Registres de la Taula de Routing.....	29
Figura 22 - Disseny de la implementació – Missatges DAO – Procés.....	29
Figura 23 - Lògica de la taula de Routing – Missatges DAO.....	30
Figura 24 - Seqüència de Forwarding – Missatges DAO.....	31
Figura 25 - Lògica de Forwarding – Missatges DAO.....	31
Figura 26 - Testbench – Topologia 1.....	34
Figura 27 - Testbench – Topologia 2.....	35
Figura 28 - Conclusions estudi comparatiu.....	80

## Índex de Taules

Taula 1 - Estàndards OPENWSN.....	10
Taula 2 - Missatges RPL.....	13
Taula 3 - Conversió de Metrica a Rank.....	15

# 1. Introducció

## 1.1. Context i justificació del Treball

Aquest treball s'enfoca en la implementació del mode Storing del protocol RPL en la plataforma OPENWSN. Actualment la plataforma OPENWSN ja té implementat el protocol RPL, però només amb el mode Non-Storing. Per tant, al desenvolupar el nou mode de treball, OPENWSN passaria a ser una de les primeres plataformes en implementar els dos modes.

Per a poder tenir el detall tècnic del funcionament del nou mode es generarà un estudi comparatiu de rendiment entre els dos modes, que s'alimentarà dels valors mesurats amb un Testbench dissenyat específicament per aquest cas. D'aquesta manera es podran establir, els punts forts i dèbils de cada mode.

Fonamentalment la idea d'aquest treball es implementar un nou mode de Routing que permeti a tots els components de la xarxa IoT participar en les tasques d'enrutament, per tal de guanyar en rendiment, estabilitat i eficiència.

## 1.2. Objectius del Treball

Els objectius del treball es materialitzen en la següent llista:

- Disseny conceptual de la solució
- Disseny del Testbench a aplicar
- Documentació de la implementació, i mesures de la DEMO-1 Storing
- Documentació de la implementació, i mesures de la DEMO-2 Non-Storing
- Estudi comparatiu de rendiment dels resultats obtinguts en les dues Demos

## 1.3. Enfocament i mètode seguit

### Estratègia de desenvolupament del treball

Al tractar-se de una solució on ja han establert, la plataforma base i el hardware, no es pot escollir altre estratègia que treballar sobre el codi ja realitzat per tal de desenvolupar la solució.

Primerament es farà un estudi del document RFC6550 [1], per tal de comprendre l'abast dels canvis necessaris a implementar. Tot això caldrà plasmar-ho en un disseny conceptual que servirà de guia segons es vagi avançant en el treball.

Seguidament es realitzarà un anàlisi en profunditat del codi que compon la plataforma OPENWSN per tal de veure com s'ha implementat la solució del protocol RPL en mode Non-Storing. Per així posteriorment poder saber quins canvis aplicar en el codi per a obtenir la solució desitjada.

Pel disseny del Testbench només caldrà una mica de investigació sobre quines característiques són importants per un protocol de Routing, i en aquest cas un protocol que treballa en IoT. Amb el Testbench dissenyat es realitzaran les mesures intentant simular

situacions reals en una xarxa IoT, on hi poden haver pèrdues de connexió entre els nodes i canvis de topologia.

## Identificació i anàlisi dels punts claus per desenvolupar el TFM

### ■ Coneixement tècnic del funcionament del protocol RPL

*(detallat al RFC6550)*

Es realitza un estudi del protocol segons el document tècnic RFC6550 i s'exposa el funcionament específic de la solució en una presentació visual per tal de tenir clar els punts a seguir en el desenvolupament del TFM. [1] [2]

### ■ Coneixement tècnic del funcionament del sistema OPENWSN

*(entorn de implementació i simulació)*

Es realitza un aprenentatge tècnic sobre l'entorn i el funcionament del sistema OPENWSN amb els materials exposats a la web oficial. Es realitza la instal·lació i proves del sistema. [3] [4]

### ■ Coneixement tècnic avançat del llenguatges de programació C i Python

*(llenguatges base del sistema OPENWSN)*

Es va realitzant un aprenentatge tècnic a través de fonts de internet per tal de poder treballar amb el codi base del sistema OPENWSN. A més de cercar les eines necessàries per tal de poder dur a terme el treball de programació més eficient. [5] [6]

### ■ Altres coneixements enfocats a IPv6 i 6LoWPAN

*(la base de tot sistema telemàtic del TFM)*

Es va realitzant un aprenentatge tècnic a través de fonts de internet per tal d'entendre les possibles situacions en els enviaments de paquets amb els canvis que es realitzin en el protocol RPL. [7] [8]

### ■ Coneixement tècnic de mesures de rendiment per un protocol

*(coneixement necessari per tal de desenvolupar les mesures)*

Es realitza estudi dels valors necessaris per fer una valoració de rendiment del protocol RPL en els dos modes de treball. [9] [10] [11]

## 1.4. Planificació del Treball

### Activitats del Projecte

#### I. Disseny i programació del projecte

#### II. Disseny conceptual de la solució

- 1) Estudi introductiu del protocol RFC6550 – Mode Storing
- 2) Disseny conceptual a aplicar

#### III. Disseny del Testbench

- A. Elaboració del llistat de mesures importants a prendre.
- B. Disseny i elecció de topologia
- C. Disseny i planificació de les proves a realitzar

#### IV. DEMO-1: protocol RFC6550 Storing mode



- A. Implementació de la DEMO
- B. Presa de mesures amb el Testbench

#### V. DEMO-2: protocol RFC6550 Non-Storing mode

- A. Implementació de la DEMO
- B. Presa de mesures amb el Testbench

#### VI. Estudi comparatiu

- A. Creació de les taules comparatives de mesures
- B. Creació de les gràfiques representatives de la comparació
- C. Anàlisi comparatiu dels resultats en totes situacions aplicades
- D. Conclusions parcials
- E. Conclusions finals

#### VII. Memòria del Projecte

#### VIII. Presentació del Projecte

### Eines TIC escollides pel desenvolupament del TFM

Mes enllà de les eines ofimàtiques necessàries per desenvolupar el TFM, també ha calgut trobar eines TIC per tal de fer més fàcil les tasques de desenvolupament. Aquest projecte principalment ha estat aproximadament un 60-70% a nivell tècnic tasques de programació. La resta de tasques son enfocades a la cerca de informació i anàlisi tècnic de les mesures preses.

#### Eines TIC per les tasques de programació:

- **Notepad++**  
Editor generalista de codi de programació.
- **NetBeans**  
Editor de projectes de programació.
- **Whire shark**  
Software de captura i anàlisi de tràfic de xarxa.
- **GitHub**  
GitHub es una plataforma de desenvolupament col·laboratiu de software per tal d'allotjar els projectes utilitzant un sistema de control de versions Git.
- **TortoiseGit**  
Es un software enfocat amb el treballs de projectes amb un sistema de control de versions Git.
- **PythonWin**  
Es un software propi del paquet Python de desenvolupament, per tal de poder editar el codi.

#### Eines TIC per a l'anàlisi tècnic de mesures:

- **Microsoft Excel**  
Software per excel·lència escollit per tal de poder emmagatzemar mesures i realitzar comparatives i gràfiques.

## Eines TIC per la cerca de informació:

- **Biblioteca UOC**

Documentació tècnica bastant interessant per tal de trobar informació necessària del projecte.

- **Google**

Cercador mundialment reconegut, molt útil per trobar documentació de tot tipus.

## Primera planificació

S'exposa seguidament la primera planificació representada amb un diagrama de GANTT en la figura 1.

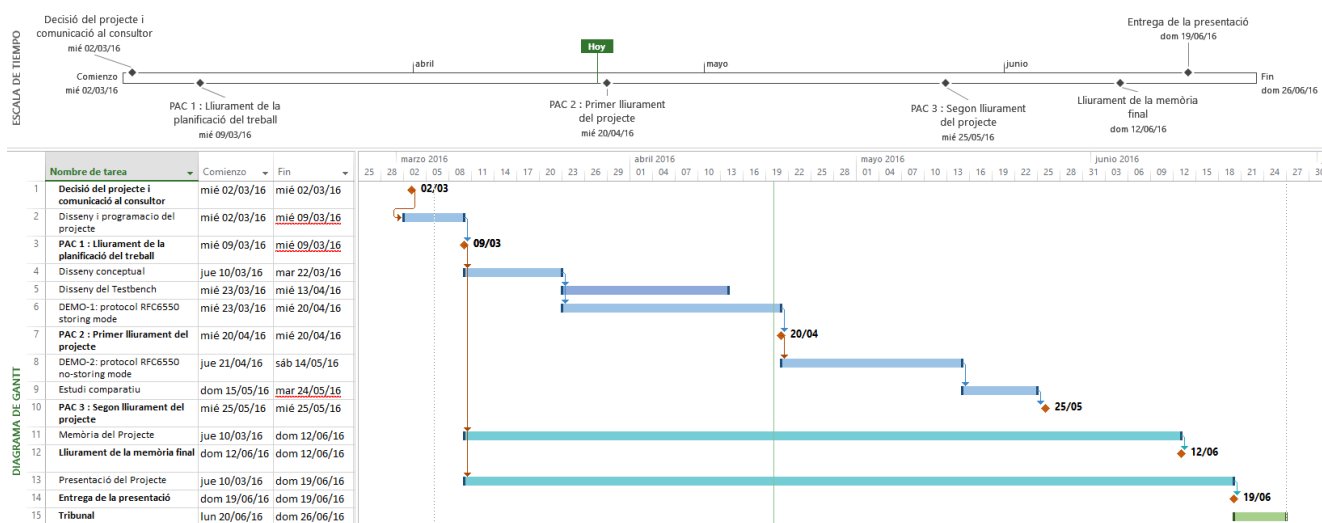


Figura 1 - Primera Planificació

## Segona planificació després de la PAC2

### Punts de inflexió trobats:

- **DEMO-1: protocol RFC6550 Storing mode**

El desenvolupament d'aquesta DEMO ha eclipsat la resta de tasques degut al volum necessari d'hores a invertir en el seu desenvolupament.

- **Disseny del Testbench**

Aquest punt s'ha desenvolupat, però queda encara informació necessària per a detallar-ho. Ha estat eclipsat per la DEMO-1.

### Nova planificació:

- **DEMO-1: protocol RFC6550 Storing mode**

S'allarga la tasca, tant per necessitat com per tractar-se de la pedra angular del TFM.

- **Disseny del Testbench**

S'allarga una mica mes enllà del planificat, situant-se al final de la DEMO-1.

- **DEMO-2: protocol RFC6550 Non-Storing mode**

Aquesta tasca s'escurça ja que la seva implementació es molt mes simple que a la primera. En aquest cas realment nomes caldrà fer el Testbench, ja que el mode Non-Storing ja es troba implementat a la solució.

S'exposa seguidament la segona planificació representada amb un diagrama de GANT en la figura 2.



Figura 2 - Segona planificació

## 1.5. Breu resumari de productes obtinguts

Aquí exposem els productes obtinguts en aquest treball:

### ■ Memòria del projecte

- 1) Introducció
- 2) Disseny conceptual de la solució
- 3) Disseny del Testbench a aplicar
- 4) Documentació de la implementació, i mesures de la DEMO-1 Storing
- 5) Documentació de la implementació, i mesures de la DEMO-2 Non-Storing
- 6) Estudi comparatiu de rendiment dels resultats obtinguts en les dues Demos

### ■ Presentació del Projecte

## 1.6. Breu descripció dels altres capítols de la memòria

La resta de capítols contindran els productes abans indicats:

### 2) Disseny conceptual de la solució

Estudi del protocol RPL en mode Storing i disseny conceptual que servirà de guia per la resta treball.

### 3) Disseny del Testbench a aplicar

Mesures clau a prendre i topologia escollida que s'aplicarà com a Testbench en les DEMOS a implementar

- 4) **Documentació de la implementació, i mesures de la DEMO-1 Storing**  
Primera DEMO on s'implementarà el mode Storing del protocol RPL
- 5) **Documentació de la implementació, i mesures de la DEMO-2 Non-Storing**  
Segona DEMO on s'implementarà el mode Non-Storing del protocol RPL
- 6) **Estudi comparatiu de rendiment dels resultats obtinguts en les dues Demos**  
Estudi comparatiu a partir de les mesures obtingudes en les dues DEMOS realitzades

## 2. Disseny conceptual de la solució

En aquest apartat primerament es farà un incís a les xarxes IoT, i els protocols de routing associats, per tal d'exposar el context on es desenvolupa aquest treball. S'aniran perfilant les característiques pròpies del treball, fins arribar a un disseny conceptual de la solució. Així aquest apartat servirà de guia, per tal de tenir un enfoc clar en la implementació del mode Storing.

### 2.1. Xarxes IoT

#### Incís a IoT

Internet de les coses fa referència a una xarxa d'objectes interconnectats. Es tracta de comunicacions M2M on els objectes s'anomenarien smart objects. Aquests objectes podran ser des de sensors situats en una ciutat o camp de cultiu fins a un objecte quotidià. El concepte que es planteja es un mon on tots els objectes estarien equipats amb dispositius que els dotarien d'un mitjà d'identificació, i donaria la possibilitat d'adquirir informació variada d'aquests objectes.

Aquestes xarxes d'objectes estableixen topologies canviant en el temps degut al possible moviment dels nodes dintre de la xarxa o pèrdues de connectivitat. Es tracta de xarxes ad-hoc on la majoria d'equips s'alimenten amb bateries, per tant es parla de xarxes de baix consum i pèrdues, conegut en anglès pel nom de LLN (Lossy Lowpower Networks).

#### Routing en una xarxa IoT

Com qualsevol altra xarxa, una xarxa IoT té els seus protocols de Routing per tal de garantir una comunicació eficient. El Routing es el procés de selecció del millor camí dintre de una xarxa, on tindrà que tenir en compte diverses característiques pròpies de la xarxa on treballa. Per tant, en una xarxa IoT, on les pèrdues de connexió de Nodes i el consum energètic es clau, es troben els següents protocols de Routing [12]:

##### ■ RPL

Protocol de Routing dissenyat específicament per xarxes LLN IoT. Suporta principalment tràfic MP2P, i un suport raonable al tràfic i funcions bàsiques de tràfic P2P. Construeix un gràfic per establir la xarxa anomenat DODAG (Destination Oriented Acyclic Graph). On tots els nodes participen proactivament enviant missatges DODAG Information Object (DIO) per tal d'explorar la xarxa. D'aquesta manera es pot classificar aquest protocol com un protocol proactiu. Mitjançant

missatges Destination Advertisement Object (DAO) es podrà establir el Routing a la xarxa. Existeixen dos modes de treball del protocol, el mode non-storing on tot el treball de Routing es centralitza en un Node amb més capacitats, o el mode storing on tots els nodes emmagatzemen informació de Routing i parlant de Routing distribuït.

### ■ OLSR i OLSRv2

El protocol Optimized Link-State Routing (OLSR) i el seu successor OLSRv2, són protocols proactius de Routing del tipus hop-by-hop i estat d'enllaç, tots dos especificats per la IETF. Es troben entre els protocols d'encaminament més populars per MANETs i per això no es poden deixar d'esmentar. OLSRv2 introdueix suport per a les mètriques alternatives com una de les majors millores de OLSR, permetre l'ús de mètriques que tinguin en compte el consum d'energia. Una extensió per OLSRv2 ha estat proposat per habilitar l'enrutament multipath, que va ser estudiada per OLSR en el passat. Tant OLSR i OLSRv2 són, tot i encara molt probablement inadequats per a les xarxes IoT ja que al tractar-se de protocols proactius generen un gran volum de tràfic per obtenir les dades de topologia, veïnatge i Routing generant gran consum d'energia en transmissions innecessàries. També cal tenir en compte el sobre ús de memòria dels Nodes per emmagatzemar totes aquestes dades.

### ■ AODV, LOADng i AODVv2

El protocol de Routing Ad hoc On-Demand Distance Vector (AODV) és un protocol reactiu d'enrutament del tipus hop-by-hop especificat per l'IETF en 2003. Fa ús d'una Route Request (RREQ)- Route Reply (RREP) -cicle, que s'activa cada vegada que un paquet a una destinació desconeguda ha de ser enviat.

Durant aquest cicle, una ruta és descoberta i emmagatzemada Hop-by-Hop: cada node només coneix el veí directe del següent salt cap a un determinat destí. Cada vegada que un vincle es trenca, això és comunicat corrent a baix de la mateixa manera.

A causa de que les rutes s'emmagatzemen només quan sigui necessari, AODV pot ser descrit com a memory-eficient. En la seva configuració més mínima, el protocol és pot ser que sigui petit en termes de mida de la imatge de codi, a causa de la seva simplicitat. Les extensions Multipath de AODV han estat proposats per l'autor original i d'altres.

Dos successors de AODV s'han desenvolupat des de la seva especificació: The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng) i AODVv, aquest últim ha estat adoptat pel grup de treball MANET de la Internet Engineering Task Force (IETF). Mentre que AODV només accepta Hop Count com a mètrica, els seus successors permeten mètriques alternatives, obrint la possibilitat per al desplegament d'una mètrica que tingui en compte el consum energètic.

### ■ CCNx/ CCNLite

CCNx és una implementació de la idea del CIE, creat per XEROX PARC. La seva adaptació més lleugera és el CCNlite, que ha estat adoptat per la IoT. CCN opera en base al sistema Hop-by-Hop. Cada vegada que un node està a la recerca de dades, es distribueix un missatge d'interès. Aquest interès es remet a través de la xarxa fins que pugui ser respost per un dels nodes participants. Cada node que rep els registres d'interès en el seu Pending Interest Table (PIT). Quan un interès es respon amb les dades, tots els nodes que fan forwarding de les dades les guarden a la seva cache, distribuint efectivament les dades per la xarxa. En fer-ho, CCN assegura que les dades són capaces de sobreviure fins i tot davant d'una partició de la xarxa. Addicionalment, les sol·licituds posteriors de les dades poden ser respostes pels nodes intermedis, distribuint la càrrega de la xarxa entre els veïns. CCN és més adequat pel tràfic MP2P o P2P.

### ■ PRoPHET

El protocol d'enrutament Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) es va publicar en 2012 com un protocol d'enrutament del tipus Hop-by-Hop per DTNs, i va ser descrit primerament en 2003 per IETF.

PRoPHET mesura els moviments d'una xarxa, tant física com en termes de trànsit de xarxa.

En base aquestes dades, la mètrica de previsibilitat de lliurament que indica la probabilitat d'una transferència de dades amb èxit es calcula per veí, caracteritzant a P<sub>RO</sub>PHET com un protocol probabilístic. Totes les dades cap a una certa ruta es guarden buffer fins que una ruta es pot establir. D'aquesta manera, P<sub>RO</sub>PHET és capaç de manejar les xarxes que no estan connectades totalment. Sempre que dos nodes es troben, ja sigui a través del moviment físic o un node de la connexió, que intercanvien la informació de previsibilitat que calculen i actualitzen les seves dades internes en conseqüència. Basant-se en aquesta informació, cada node decidirà si i quines dades faran forwarding a través d'un veí que acaba de conèixer. Un node pot enviar les seves dades a través de més d'un veí, fent de P<sub>RO</sub>PHET un protocol multipath també.

## 2.3. Context del Treball dintre de les xarxes IoT

### Tipus de xarxa

El tipus de xarxa IoT on s'enfoca el treball es del tipus LLN, Xarxes de Baix consum i Pèrdues. Específicament son xarxes de sensors sense fils (Wireless Sensor Networks, WSN). Aquestes comunitats de sensors autònoms i intel·ligents (sovint anomenats "MOTES"), alimentats per bateries o altres mitjans de baixa potencia, estableixen xarxes WSN.

L'objectiu d'aquestes xarxes es enviar les mesures preses a un punt central de recollida de dades des d'on les dades s'enviaran cap a un centre d'anàlisi i processat. Degut tant a la baixa potencia de treball com a les condicions ambientals es produeixen pèrdues totals o parcials aleatòries en la connexió entre nodes, generant la necessitat de protocols que mantinguin xarxes que s'adaptin dinàmicament a aquests canvis mantenint una comunitat entre rendiment i consum energètic.

### Sistema OPENWSN

El treball tindrà com a base de desenvolupament el sistema [3]. L'objectiu del projecte OPENWSN és proporcionar una implementació en codi obert d'una pila de protocols basada en el estàndard de IoT, en una varietat de plataformes de hardware i software. Així podrà ajudar al sistema educatiu i a la indústria per a comprovar l'aplicabilitat d'aquests estàndards de IoT, per a que aquelles xarxes siguin veritablement ubiqües.

Aquest sistema open-source, enfocat a xarxes WSN, treballa amb un conjunt d'estàndards agrupats segons les capes OSI [3] representat a la següent taula:

Aplicació	CoAP, HTTP
Transport	UDP, TCP
IP/Routing	IETF RPL
Adaptació	IETF 6LoWPAN
Access al Medi	IEEE802.15.4e
PHY	IEEE802.15.4-2006

Taula 1 - Estàndards OPENWSN

### Protocol de Routing utilitzat

El protocol de Routing utilitzat per el sistema OPENWSN es el RPL, això es degut a que el protocol RPL esta dissenyat específicament per xarxes IoT LLN, obtenint així millors resultats

que amb els altres protocols de Routing abans exposats. Tal vegada la resta de protocols podrien ser igual o mes eficients que el protocol escollit, però no son protocols que es puguin executar en Nodes amb hardware senzill i a la vegada garanteixin un sistema de Routing eficient.

## Routing utilitzat

Com ja s'ha exposat abans aquest treball s'enfoca principalment en la implementació del mode de Routing Storing del protocol RPL, ja que el mode Non-Storing ja esta actualment implementat al sistema OPENWSN. S'espera que amb el mode Storing els nodes de la xarxa participin activament en les tasques d'enrutament, deixant de ser dumb-routing nodes a ser smart-routing nodes que coneixen i saben cap on enrutar els paquets que reben, sense tenir que acudir a un node central.

La implementació del nou mode de Routing caldrà que vingui acompanyada d'un petit estudi comparatiu que serveixi per conèixer les bondats i defectes d'aquest nou mode de Routing respecte del mode Non-Storing.

## 2.4. Protocol RPL

### Descripció

El Protocol RPL (IPv6 Routing Protocol for Low Power and Lossy Networks) es un protocol de routing específicament dissenyat per a xarxes de de baix consum i amb pèrdues (Low power and Lossy Networks - LLN) compatible amb el protocol 6LoWPAN [13] [14].

La definició de 6LoWPAN [15] : "Els estàndards 6LoWPAN permeten l'ús eficaç de IPv6 amb baix consum d'energia, xarxes sense fils de baixa velocitat en dispositius encastats simples a través d'una capa d'adaptació i l'optimització dels protocols relacionats". Per tant, està clar que compleix amb els requisits per ser utilitzat a les xarxes IoT.

El grup de treball IETF 6LoWPAN (2005) es va crear per habilitar l'IPv6 per a ser utilitzada, en combinació amb IEEE 802.15.4., amb sistemes encastats sense fils. A més, per establir els requisits mínims per a la implementació d'una versió mes lleugera del IPv6 que s'ajusti amb els dispositius més mínims. Finalment, mitjançant el disseny d'una versió especifica de ND [16], tenint en compte xarxes sense fils amb baix consum.

La meta més alta de RPL és proporcionar camins d'enrutament eficient pels sistemes de trànsit P2MP i MP2P en les LLNs.

Tenim els següents punts claus del protocol RPL:

- RPL és un protocol d'enrutament vector-distància per LLNS que fa ús de IPv6.
- El protocol tracta d'evitar els bucles d'encaminament mitjançant el càlcul de la posició d'un node respecte a altres nodes utilitzant els Ranks, generant una posició relativa dintre de la l'arrel DODAG.
- L'especificació RPL defineix quatre tipus de missatges de control per al manteniment de la topologia i l'intercanvi d'informació.
- Un altre fet important sobre el disseny de protocols és el manteniment de la topologia.

El protocol RPL pot treballar en dos modes, en cada cas establirà un tipus diferent de Routing:

- **Mode Storing** (Table-Routing)  
Tots els nodes guarden informació i participen en el Routing activament. Routing distribuït.
- **Mode Non-Storing** (Source-Routing)  
El node root es el únic que emmagatzema la informació i porta el Routing. Routing centralitzat

## Xarxa DODAG

El protocol RPL construeix un arbre d'encaminament que defineix el DODAG (Destination-Oriented Directed Acyclic Graph). En un DAG (Directed Acyclic Graph) tots els camins estan formats per tal d'evitar bucles. Un DODAG no deixa de ser un DAG on tots els camins formen una topologia en arbre, i el centre d'aquesta topologia es el node root. S'anomena d'aquesta manera xarxa DODAG a la xarxa establerta en la topologia dibuixada pel DODAG. La xarxa DODAG es componrà en el seu inici de un conjunt de nodes inconnexos, sense cap topologia definida. Es pot veure a la figura 3 un exemple senzill.

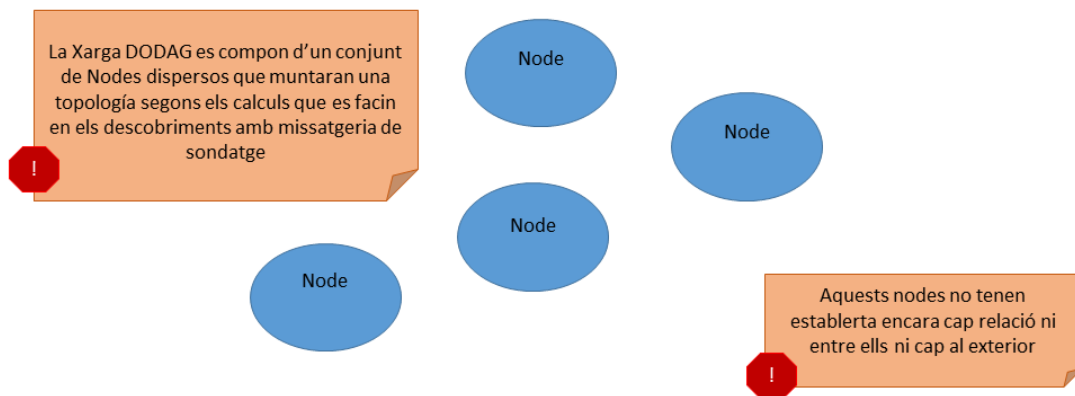


Figura 3 - Xarxa DODAG

## Xarxa DODAG -- Estructura i relacions

Primer de tot, per a engegar el procés de muntatge de la xarxa, caldrà escollir un node per a que prengui el rol de root, aquest node escollit passarà a ser el centre en la topologia, tenint el Rank 0 (el Rank mes baix de tots). En la topologia de la xarxa DODAG els Ranks van de menor a major, sent els mes petits els mes propers al node root i els mes grans els mes allunyats.

La construcció del DODAG s'inicia al node root mitjançant l'enviament de missatges DIO (DODAG Information Objects) a tots els nodes en mode broadcast. A traves dels missatges DIO, explicats mes endavant, els nodes aniran calculant el seu Rank i establint relacions amb la resta de nodes, aquestes relacions seran les següents:

- **Parent (Pare)** (situat per sobre del node) → Rank mes petit
- **Child (Fill)** (situat per sota del node) → Rank mes gran

La topologia es muntarà segons els Ranks establerts i les relacions entre els nodes. En la figura 4 es pot veure un exemple de topologia ja muntada, on s'exposa les funcionalitats i capacitats de cada node dintre de la topologia.



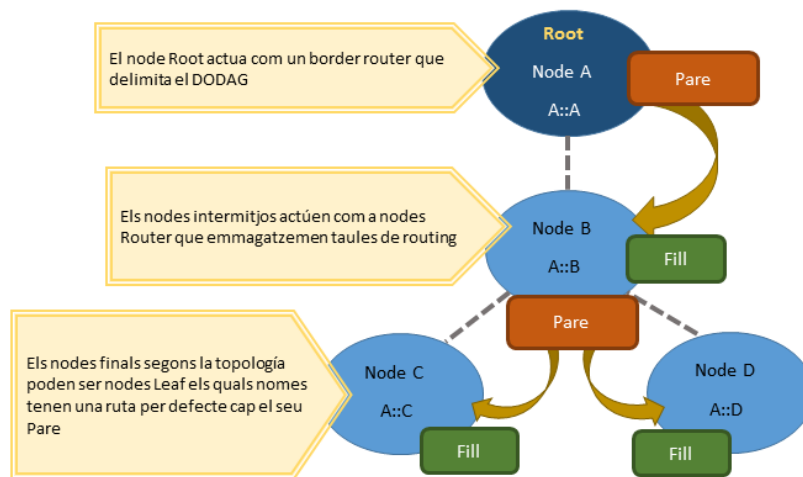


Figura 4 - Topologia, estructura i relacions

## Missatges RPL - Codificació i noms

En protocol RPL per tal de funcionar, necessita de un conjunt de missatges propis amb els quals establirà i mantindrà la informació de la xarxa. Aquesta missatgeria específica del protocol RPL tindrà les següents funcions:

- Establiment i manteniment de la topologia
- Establiment i manteniment del Routing
- Seguretat en les comunicacions

En la figura 5 es pot veure una taula on es llisten i s'exposen els missatges utilitzats pel protocol RPL.

Codificació dels missatges ICMPv6 en RPL		
Codi	Típus de Missatge RPL	Descripció
0x00	DODAG Information Solicitation (DIS)	Missatge per sol·licitar missatge DIO al node que ho enviem
0x01	DODAG Information Object (DIO)	Missatge que desencadena la construcció i actualització de la topologia
0x02	Destination Advertisement Object (DAO)	Missatge per a transmetre la informació des dels nodes cap endalt i muntar el sistema de routing
0x03	Destination Advertisement Object Acknowledgment (DAO-ACK)	Missatge de comprovació per saber que s'ha rebut el missatge DAO
0x80	Secure DODAG Information Solicitation	Missatge per establir seguretat en la comunicació
0x81	Secure DODAG Information Object (DIO)	Missatge DIO amb seguretat
0x82	Secure Destination Advertisement Object (DAO)	Missatge DAO amb seguretat
0x83	Secure Destination Advertisement Object Acknowledgment (DAO-ACK)	Missatge DAO-ACK amb seguretat

Taula 2 - Missatges RPL

## Càlcul del Rank

El Rank d'un node es una representació escalar de la localització del node dintre d'un DODAG. El Rank s'utilitza per tal de evitar i detectar loops. D'aquesta manera el Rank defineix la posició relativa d'un node relativament a d'altres nodes respecte al DODAG root. El Rank augmenta estrictament en la direcció Down (des de el DODAG root cap els nodes

Leaf), i estrictament disminueix en la direcció Up (des de els nodes Leaf cap el node DODAG root). El càlcul del valor del Rank dependrà de OF. Seguidament s'exposarà el procés pel càlcul del Rank a cada node, segons les següents RFC:

■ Es troba al RFC6552 [17], la següent explicació al càlcul del Rank:

Una implementació del OF0 primerament calcula la variable `step_of_rank` associada amb un parent determinat, segons les propietats rellevants de link i mètriques. La variable `step_of_rank` s'utilitza per calcular la quantitat que s'ha d'augmentar el Rank al llarg d'un link concret.

OF0 utilitza les següents variables:

`step_of_rank` (enter estrictament positiu): un càlcul intermig basat en les propietats de link amb un cert veí

`rank_increase` (enter estrictament positiu): delta entre el Rank del parent preferit i el d'un mateix

Calcular el `step_of_rank` basant-se en una mètrica fixa tal com el cost administratiu implica que la implementació del OF0 només considera als parents amb una connectivitat suficientment bona. Això es recomana que sigui la base de càlcul del `step_of_rank` en links dinàmics amb propietats tals com la mètrica ETX, tal com s'exposa al RFC6551.

OF0 permet una implementació per tal de donar elasticitat al `step_of_rank` per tal de permetre la selecció de com a mínim un successor viable i així mantenir diversitat de paths. Extendre el `step_of_rank` no està recomanat, perquè augmenta la distància aparent del node cap el root, distorsiona el DODAG fent que no obtingui una forma òptima, i pot causar inestabilitats degut als comportaments cobdiciosos amb el que els nodes augmenten els seus Ranks per utilitzar-se mútuament i generar un loop. Encara i això, la implementació cal que doni un marge de elasticitat al `step_of_rank` des de 0 i una constant fixada `MAXIMUM_RANK_STRETCH`.

La implementació té que mantenir la elasticitat entre dos constants fixades `MINIMUM_STEP_OF_RANK` i `MAXIMUM_STEP_OF_RANK`. Aquest rang esdevindrà en una àmplia variació de qualitats de link.

El marge entre `MINIMUM_STEP_OF_RANK` i `MAXIMUM_RANK_STRETCH` pot ser no suficient en cada cas per poder distingir clarament links de diferent tipus o categories per tal de afavorir, per exemple, nodes alimentats per corrent per davant dels de bateries, o dels connectats amb cablejat d'alta velocitat (wired) davant del connectats a baixa velocitat (wireless), dintre del mateix DAG. La implementació caldrà que permeti al operador configurar el `rank_factor` i aplicar el factor a tots els links i peers per tal de multiplicar el efecte de elasticitat del `step_of_rank` en el càlcul del `rank_increase` que s'exposarà seguidament.

Adicionalment, la implementació cal que reconeguin les categories dels peers i links, tals com els diferents tipus de links, per tant caldrà que pugui configurar un `rank_factor` més específic per aquestes categories. El `rank_factor` té que estar contingut entre el `MINIMUM_RANK_FACTOR` i el `MAXIMUM_RANK_FACTOR`.

La variable `rank_increase` es representa en unitats expressades per la variable `MinHopRankIncrease`, que per defecte es fixa amb la constant `MIN_HOP_RANK_INCREASE` descrita al RFC6550.

El `step_of_rank`  $S_p$  que es calcula per un link, es multiplica pel `rank_factor`  $R_f$ , i llavors es pot ajustar per un terme  $S_r$  que es menor o igual al `stretch_of_rank` configurat. El `rank_increase` resultant s'afegeix al Rank del parent preferit  $R(P)$  per tal d'obtenir el valor per aquest node  $R(N)$ :

$$R(N) = R(P) + \text{rank\_increase on:}$$

$$\text{rank\_increase} = (R_f * S_p + S_r) * \text{MinHopRankIncrease}$$

■ Es troba al RFC6719 [18], la següent explicació al càlcul del Rank:

El protocol de routing per xarxes LLN (RPL) construeix rutes utilitzant el OF, que optimitza o restringeix les rutes que selecciona o utilitza. En el MRHOF, OF selecciona rutes que minimitzen la mètrica i la histèresis redueix la inestabilitat en cas de petits canvis de mètrica. MRHOF treballa amb mètriques afegides a través de la ruta, i les mètriques que utilitza estan determinades per les mètriques que els missatges RPL DIO transmeten.

El DODAG root estableix el seu Rank com a `MinHopRankIncrease`. Valor mínim que fixat per l'increment del valor de Rank.

Una vegada un node no-root selecciona el seu conjunt de parents, pot utilitzar la següent taula 2 per a convertir el path cost d'un parent (escrit com Cost a la taula) a un valor de Rank:

Node / link Metric	Rank
Hop-Count	Cost
Latència	Cost / 65536
ETX	Cost

Taula 3 - Conversió de Metrica a Rank

Si s'utilitza el MRHOF amb altres mètriques, el Rank esta indefinit. Si el Rank esta indefinit, el node s'ha d'unir a un dels veïns com un node Leaf RPL, tal com s'exposa al RFC550.

El MRHOF utilitza aquest valor de Rank per a calcular el Rank associat amb el path per a cada membre del conjunt de parents. Aquest càlcul es el màxim de dos valors:

- El primer valor de Rank es calculat mitjançant la taula 2
- El segon es el Rank publicat pel node mes el `MinHopRankIncrease`

Un node estableix el seu Rank al màxim de tres valors:

- 1) El Rank calculat segons el path del parent preferit.
- 2) El Rank del membre del conjunt de parents amb el Rank publicat mes gran, aproximat al valor integral de Rank mes proper.

Exemple:

$$\text{MinHopRankIncrease} * (1 + \text{floor}(\text{Rank}/\text{MinHopRankIncrease}))$$

- 3) El Rank més gran calculat del conjunt de parents, menys `MaxRankIncrease`

El primer cas es el Rank associat amb el path del parent preferit. El segon cas cobreix el requeriment 5 del Rank publicat en la secció 8.2.1 del RFC65550. El tercer cas assegura que un node no publica un Rank, que s'oposa als utilitzats per conjunt de parents.

Cal tenir en compte que el tercer cas el node publica un valor conservatiu de Rank basat en els membres del conjunt de parents. Aquest valor conservatiu te que ser significativament major que el Rank calculat pel path del conjunt de parents.

En conseqüència, escollir un conjunt de parents amb paths que tenen una àmplia gamma de Ranks pot resultar molt probablement en enrutament suboptim: els nodes podrien no escollir bons paths, ja que s'anuncien com a molt pitjor del que realment són. La selecció exacta d'un conjunt dels parents és una decisió d'implementació.

## 2.5. Processos per l'establiment de la xarxa DODAG

### Missatge DIO – Broadcast-Info a la xarxa

Un dels missatges mes importants per l'establiment de la xarxa DODAG és el missatge DIO. Aquest missatge s'encarregarà principalment de informar a la resta de nodes de les dades del node que ho envia, per tal d'establir la topologia i les relacions entre els nodes.

Els missatges DIO s'enviaran des de tots els nodes en mode broadcast. Amb la informació rebuda cada node calcularà el seu Rank mitjançant un algorisme, i informarà llavors a la resta de nodes enviant un nou missatge DIO. Això portarà a un procés iteratiu que acabarà quan hagin convergit tots els càlculs de Rank en la xarxa. Els missatges DIO es seguiran enviant per tal de mantenir la xarxa actualitzada en cas de possibles canvis topològics. Es pot veure a la figura 5 quina informació i accions portarà el missatge DIO.

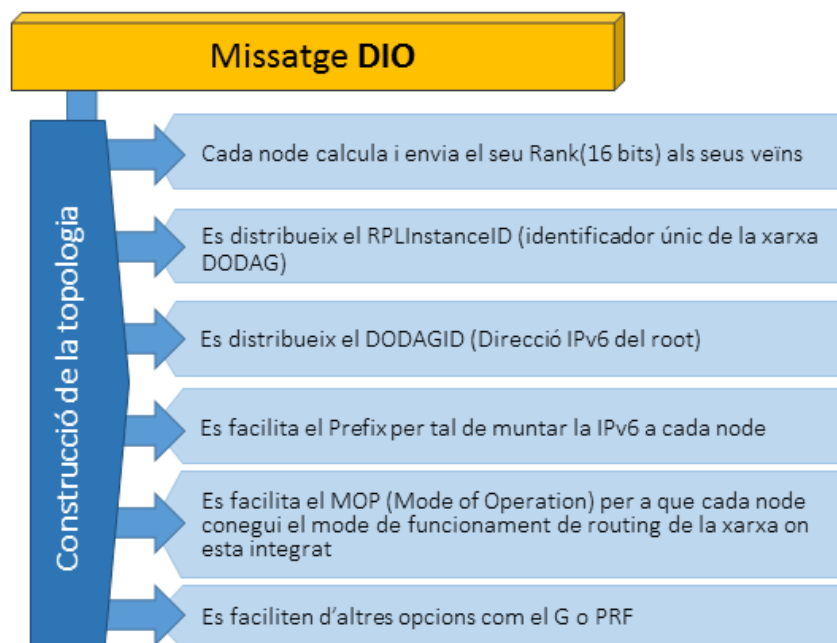


Figura 5 - Missatge DIO

### Missatge DIO - Construcció de la topologia

Tal com s'ha exposat abans els missatges DIO juguen un paper molt important en l'establiment de la topologia. El procés s'inicia amb els següent passos:

- 4) El node Root inicia l'enviament de missatges DIO en mode broadcast
- 5) Amb la informació rebuda en el missatge DIO els nodes mitjançant un algoritme trien el seu Pare i calculen el seu Rank
- 6) Els nodes envien un nou missatge DIO en mode broadcast per informar de les seves dades

Amb aquest procés inicial començarà el procés general d'enviament de missatgeria DIO, on ja tota la resta de nodes enviaran missatges DIO modificant el missatge DIO original rebut, afegint les seves dades. Es pot veure gràficament el procés inicial a la figura 6.

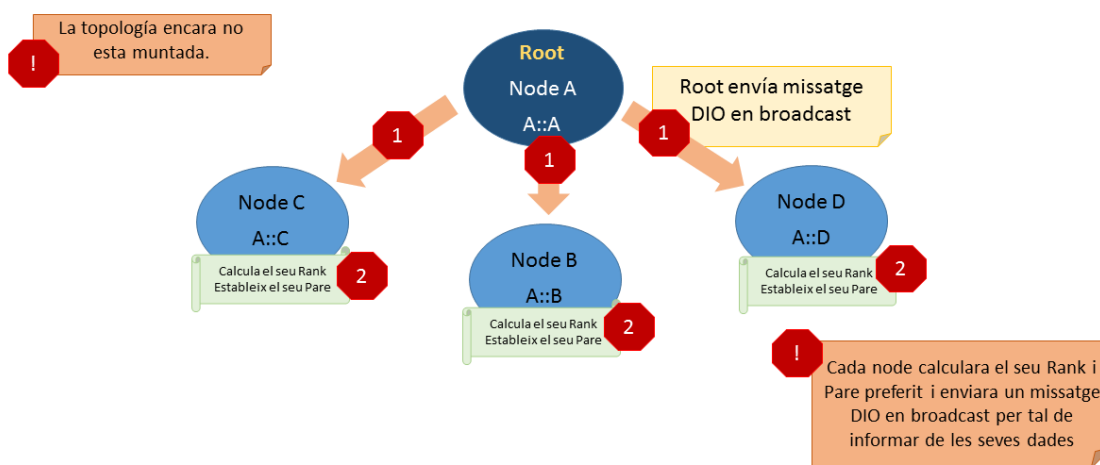


Figura 6 - Missatge DIO – Procés inicial

Un cop iniciat, el procés general es compondrà dels següents passos:

- 1) Tots els nodes envien missatges DIO en mode broadcast
- 2) Amb la informació rebuda en el missatge DIO els nodes mitjançant un algoritme trien el seu Pare i calculen el seu Rank
- 3) Els nodes envien un nou missatge DIO en mode broadcast per informar de les seves dades

Els passos del procés general es repetiran iterativament fins que convergeixin els càlculs dels Ranks donant finalment una topologia. Es pot veure gràficament el procés inicial a la figura 7.

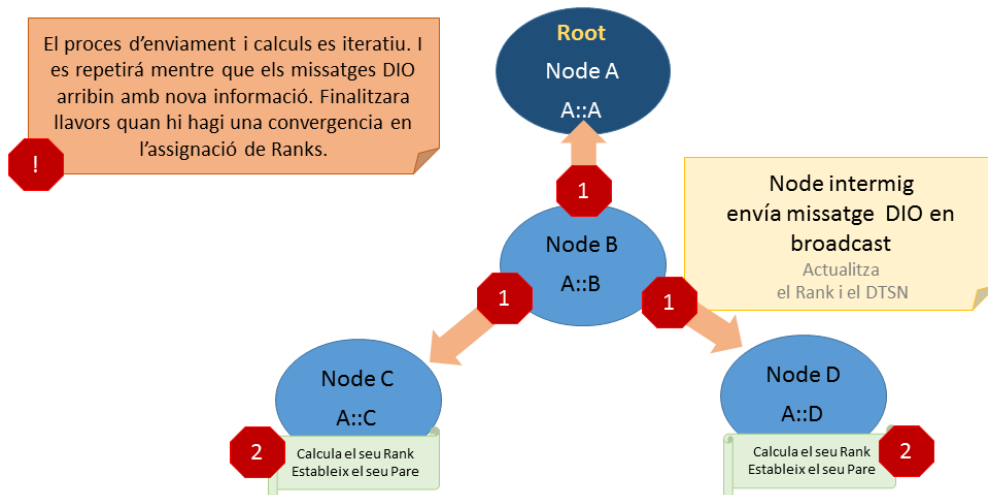


Figura 7 - Missatge DIO – Procés general

Amb el procés finalitzat, s'obté la topologia de la xarxa DODAG, on cada node té definit un Rank, i un conjunt de relacions amb la resta de nodes que els envolten. Es pot veure a la figura 8 una topologia exemple establida.

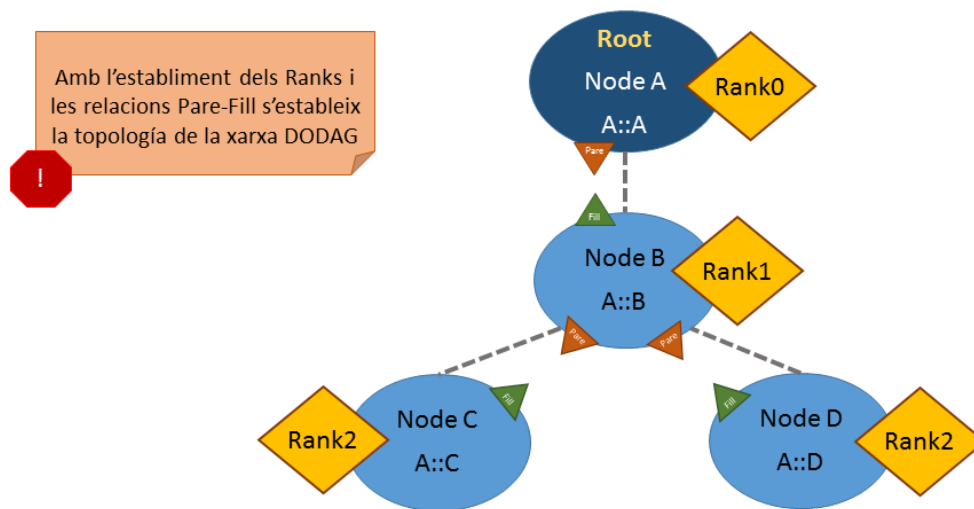


Figura 8 - Missatge DIO – Topologia establida

Quan es parla de la finalització del procés, es parla en relació amb el algoritme Trickle que controla l'enviament dels missatges DIO. On aquest algoritme mai cessa l'enviament de missatges DIO, sinó que disminueix el volum inicial necessari per muntar la topologia quan s'arriba a una convergència en la informació de la xarxa.

### Missatge DIO - Base del Missatge

Els missatges DIO es componen d'un conjunt de camps que formen la base del missatge. A la figura 9 es representa la base del missatge DIO gràficament indicant tots els camps que el formen i la seva distribució.

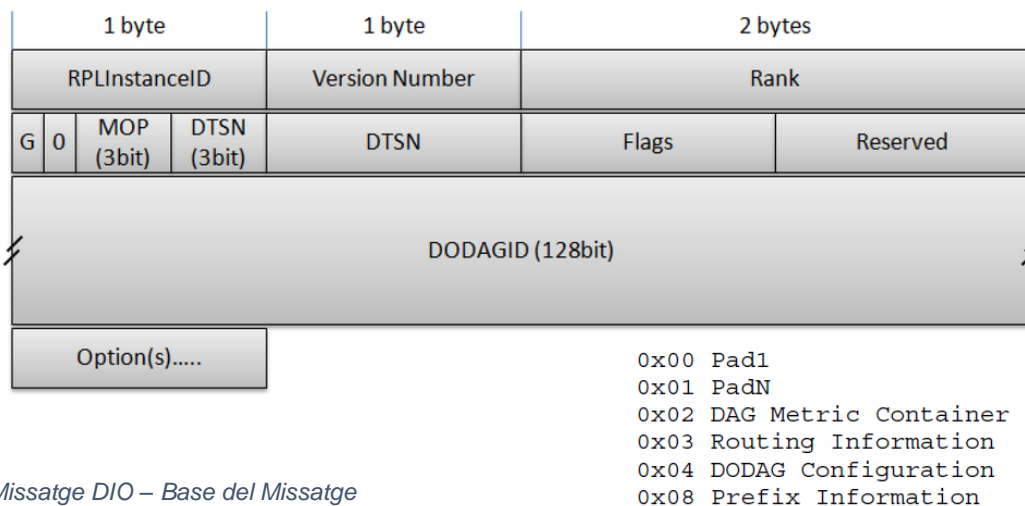


Figura 9 - Missatge DIO – Base del Missatge

Els camps que componen el missatge DIO base son:

**Grounded (G):** El flag 'G' (Grounded) indica si el DODAG que es publica pot satisfer la fita de la aplicació. Si el flag esta marcat, el DODAG esta "grounded". Si el flag esta desmarcat, el DODAG esta "floating".

**Mode of Operation (MOP):** El camp Mode of Operation (MOP) identifica el mode d'operació de la instancia RPL, tal com es publicarà a través del DODAG root. Les possibles codificacions del camp son:

- 0 → El protocol RPL no mante Downward routes
- 1 → Mode Non-Storing
- 2 → Mode Storing sense suport a Multicast
- 3 → Mode Storing amb suport a Multicast

**DODAGPreference (Prf):** Enter de 3-bit com es de preferit el node root d'aquest DODAG comparat a altres nodes root en la instancia. Els rangs compresos pel DAGPreference van de 0x00 (el menys preferit) a 0x07 (el mes preferit). El valor per defecte és 0.

**Version Number:** Enter de 8-bit fixat pel DODAG root a la DODAGVersionNumber.

**Rank:** Enter de 16-bit que indica el Rank DODAG del node que envia el missatge DIO.

**RPLInstanceID:** Camp de 8-bit fixat pel DODAG root que indica de quina Instancia RPL forma part el DODAG.

**Destination Advertisement Trigger Sequence Number (DTSN):** Enter de 8-bit fixat pel node que munta el missatge DIO. El flag Destination Advertisement Trigger Sequence Number (DTSN) s'utilitza com a part del procés de manteniment del routing.

**Flags:** Camp de 8-bit reservat per flags. El camp es te que inicialitzar a zero pel enviaador i te que ser ignorat pel receptor.

**Reserved:** Camp de 8-bit. El camp es te que inicialitzar a zero pel enviaador i te que ser ignorat pel receptor.

**DODAGID:** Adreça IPv6 de 128-bit IPv6 fixada pel DODAG root que identifica únicament al DODAG. El DODAGID te que ser una adreça IPv6 que pertanyi al DODAG root i sigui enrutable.

Tots aquells bits en la base del missatge DIO es troben reservats, i d'aquesta manera els bits sense assignar es tenen que posar a zero i ser ignorats en la recepció.

### Missatge DAO - Establiment del Routing

Al igual que el missatge DIO, el missatge DAO forma part del establiment de la xarxa DODAG. En aquest cas el missatge DAO s'encarregarà d'establir i mantenir el Routing en la xarxa DODAG.

Els missatges DAO aniran tipològicament dintre de l'arbre DODAG des dels nodes mes externs cap els nodes mes propers al node root, enviant-se de fills cap a pares. El node root no participarà en l'enviament de d'aquesta missatgeria. Es pot veure a la figura 10 quina informació i accions portarà el missatge DAO.

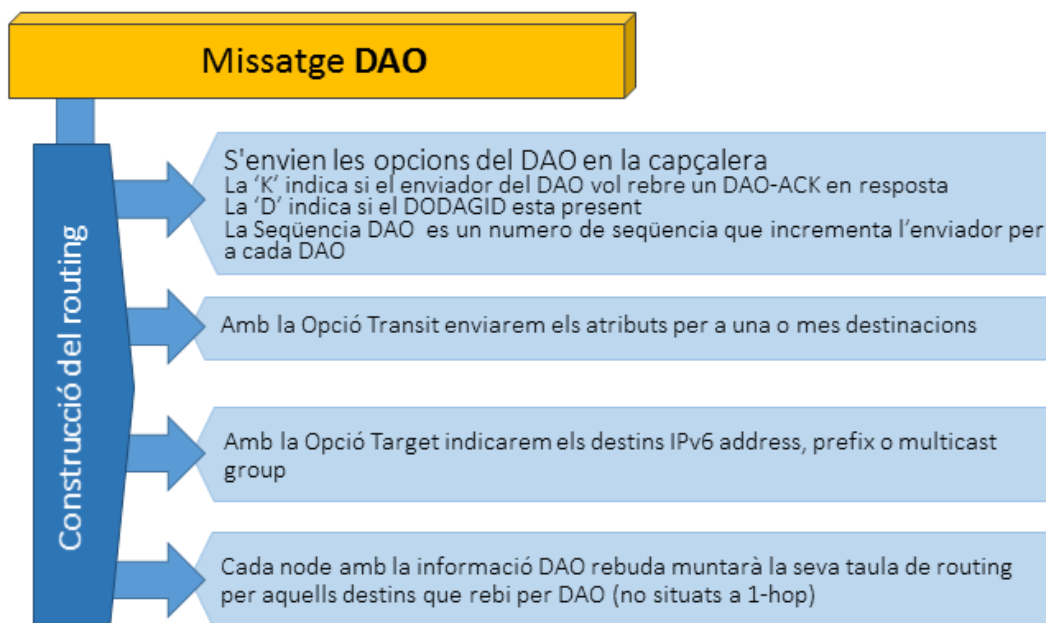


Figura 10 - Missatge DAO

El missatge DAO a diferencia del missatge DIO es tractarà diferent en l'enviament, segons en quin mode estigui treballant en protocol RPL. Aquí es presenten les diferències que es troben a cada Mode:

#### ■ Mode Storing:

- I. En aquest mode cada node enviarà els missatges DAO cap el seu Pares.
- II. No contindrà la direcció del Parent en la Transit Option.
- III. Es tindran que enviar els fills i rutes a través de la Target Option
- IV. Tots els nodes participaran en el Routing fent forwarding dels paquets amb les dades que tinguin de Routing.

Es pot veure gràficament el procés d'enviament dels missatges DAO, i els processos que desencadenen pel cas del Mode Storing a la figura 11.



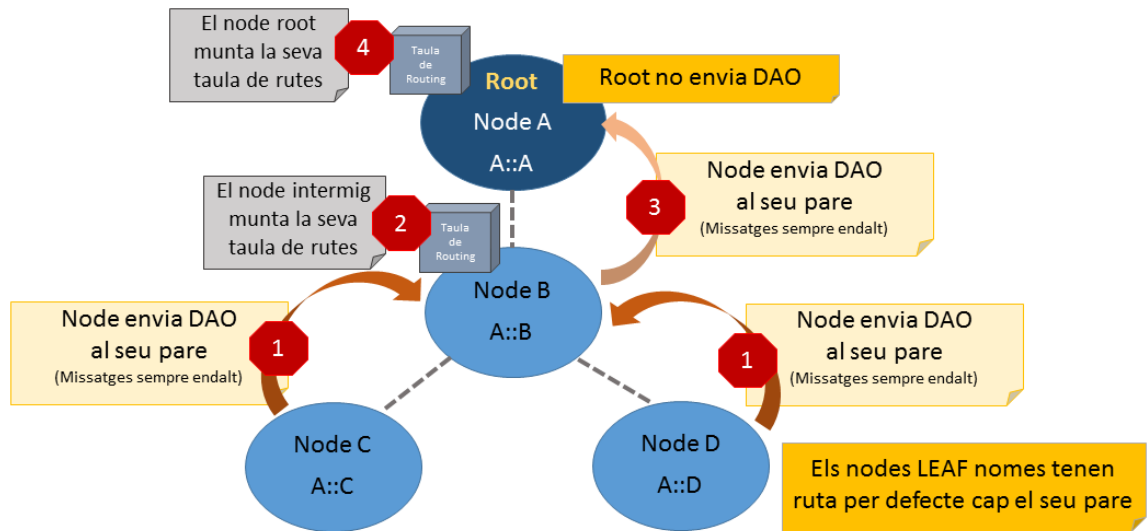


Figura 11 - Missatge DAO – Enviament

Una vegada establert el Routing tots els nodes contindran la informació necessària per tal de poder enviar els paquets a qualsevol node. En la figura 12 es pot veure un exemple gràfic de Table-Routing establert.

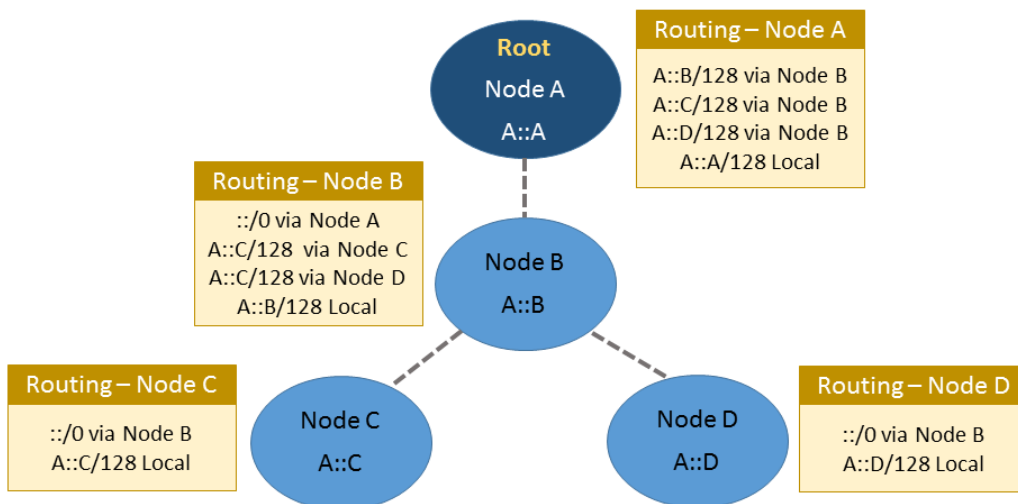


Figura 12 - Missatge DAO – Table-Routing

### ■ Mode Non-Storing:

- I. En aquest mode tots els nodes enviaran els missatges DAO cap el node Root
- II. Només el node Root participarà en el Routing dels paquets. Els nodes només aniran llegint la capçalera de Source-Routing per tal de fer forwarding dels paquets.

El procés d'enviament per a aquest Mode serà centralitzat, on tots els paquets aniran directament cap al node Root tal com es pot veure a la figura 13. Una vegada establert el Routing serà el node root que generarà amb les seves taules de Parents i Childs la capçalera de Source-Routing per tal d'enviar els paquets al destí.

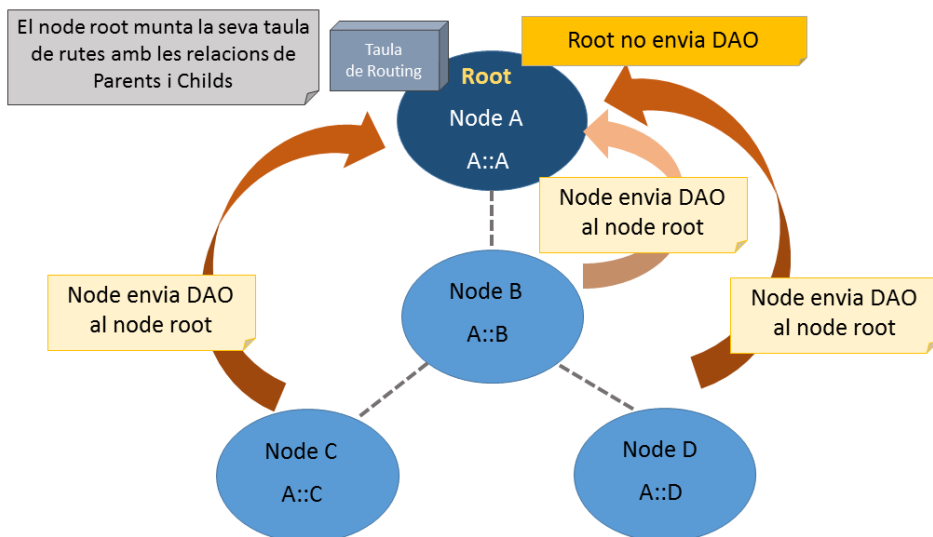


Figura 13 - Missatge DAO – Source-Routing

## Missatge DAO – Base del Missatge

Els missatges DAO es componen d'un conjunt de camps que formaran el paquet. A la figura 14 es representa el paquet DAO gràficament indicant tots els camps que el formen i la seva distribució.

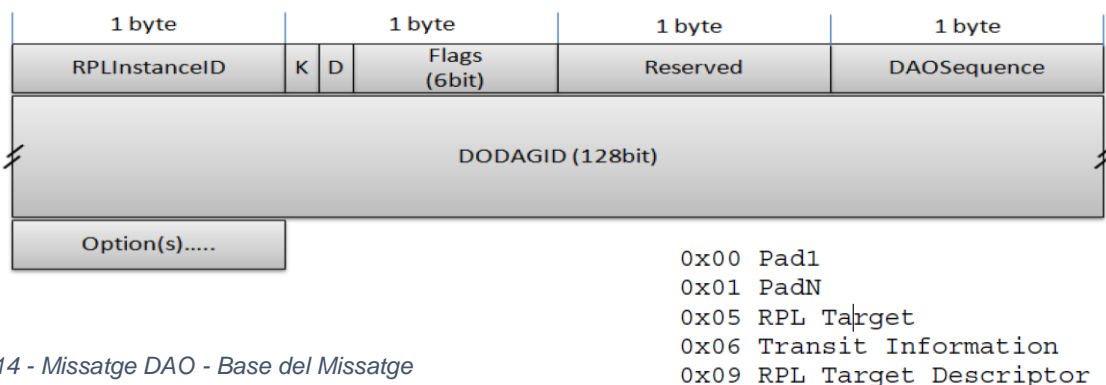


Figura 14 - Missatge DAO - Base del Missatge

Els camps que componen el missatge DAO base són:

**RPLInstanceID:** Camp 8-bit indica la instància de la topologia associada amb el DODAG, s'apren del missatge DIO.

**K:** El flag 'K' indica que el receptor del missatge tindrà que enviar un DAO-ACK en retorn. Es tracta de l'opció associada al protocol TCP, amb un missatge ACK rebre una confirmació. No s'exposarà el missatge DAO-ACK ja que no s'utilitzarà en aquest treball.

**D:** El flag 'D' indica que el camp DODAGID està present. Aquest flag es té que marcar quan s'utilitza una RPLInstanceID local.

**Flags:** Els 6 bits que queden sense ús en el camp Flags estan reservats per flags. El camp es té que inicialitzar a zero pel enviaador i té que ser ignorat pel receptor.

**Reserved:** Camp de 8-bit. El camp es té que inicialitzar a zero pel enviaador i té que ser ignorat pel receptor.

**DAOSequence:** S'incrementa amb cada nou missatge DAO des d'un node i serà reenviat en els missatges DAO-ACK.

**DODAGID (optional):** Enter de 128-bit fixat pel DODAG root que identifica únicament un DODAG. Aquest camp està únicament present quan el flag 'D' està marcat. Aquest camp està típicament només present quan s'utilitza una RPLInstanceID local, per tal de identificar el DODAGID que està associat amb el le RPLInstanceID. Quan s'utilitza una RPLInstanceID global, aquest camp no és necessari.

Tots aquells bits en la base del missatge DAO es troben reservats, i d'aquesta manera els bits sense assignar es tenen que posar a zero i ser ignorats en la recepció.

Tal com es pot veure en la figura 14, al final de la base del missatge, podrà contindre informació opcional. Aquesta informació opcional es divideix en les dues opcions següents:

### 1) Transit Option

La opció Transit Option es componrà d'un conjunt de camps que s'afegiran al paquet DAO. A la figura 15 es representa el gràficament la Transit Option indicant tots els camps que la formen i la seva distribució.

Amb aquesta opció es transmetrà informació bàsica per saber si les dades transmèses són noves, també es podrà saber en quin mode està treballant, ja que al mode Storing la Option Length es posarà 0 degut a que no es facilitarà cap Parent Address.

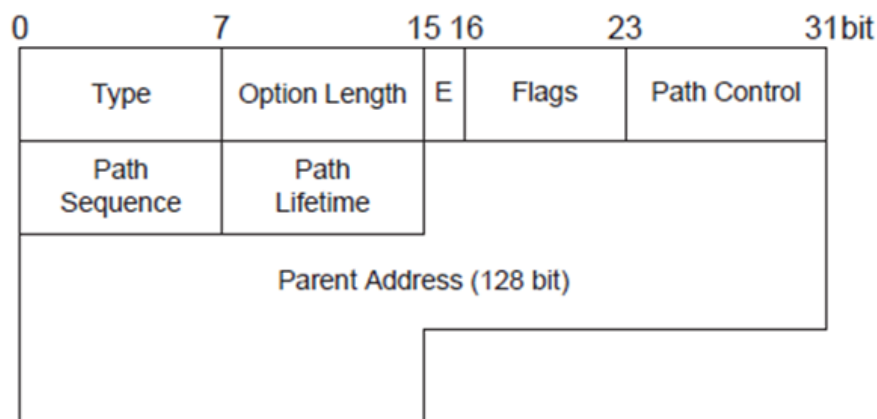


Figura 15 - Missatge DAO – Transit Option

Els camps que componen la opció Transit Option són:

**Option Type:** 0x06 (Valor prefixat per aquesta opció).

**Option Length:** Variable, depenent de si el camp DODAG Parent Address està present.

**External (E):** Flag de 1-bit. El flag 'E' es marca per indicar que parent router redistribueix destins externs cap a la xarxa RPL. Un destí extern és un destí que s'ha après a través d'altres protocols. Els destins externs es llisten a les Target option que immediatament succeeixen després de la Transit Information option. No s'espera que un destí extern suporti els missatges RPL o les opcions.

**Flags:** Els 7 bits que queden sense ús en el camp Flags estan reservats per flags. El camp es té que inicialitzar a zero pel enviaador i té que ser ignorat pel receptor.

**Path Control:** Camp de 8-bit. El camp Path Control limita el numero de DAO parents cap on un missatge DAO publica connectivitat cap una destinació que cal sigui especifica, i també proveir de indicacions de les preferències associades.

**Path Sequence:** Enter de 8-bit. Quan una Target option es crea pel node que conte el paix del destí (p.ex. en un missatge DAO), el node en qüestió fixa el Path Sequence i incrementa el Path Sequence cada vegada que munti una Target option amb informació actualitzada.

**Path Lifetime:** Enter de 8-bit. Temps de vida en Lifetime Units (obtingut de la Configuration option) que el prefix es fixarà per a que sigui valida una ruta. El període comença quan es rep un nova Path Sequence. Un valor de 0xFF representa infinit. Un valor de 0x00 indica la pèrdua d'accés. Un missatge DAO que contingui una Transit Information option amb un Path Lifetime de 0x00 cap al destí es tractarà com un DAO No-Path.

**Parent Address (optional):** Adreça IPv6 del DODAG parent del node originalment exposat a la Transit Information option. Aquest camp pot no estar present, depenent del mode de treball (Storing o Non-Storing) que estigui establert. La presència d'aquest camp s'indicarà amb la Option Length.

Tots aquells bits en la base de la Transit Information option es troben reservats, i d'aquesta manera els bits sense assignar es tenen que posar a zero i ser ignorats en la recepció.

## 2) Target Option

La opció Target Option es componrà d'un conjunt de camps que s'afegiran al paquet DAO. A la figura 16 es representa el gràficament la Target Option indicant tots els camps que la formen i la seva distribució.

Amb aquesta opció s'enviaran els Nodes que tingui per sota el Node que genera el missatge DAO, anomenats Childs. Al implementar el mode Storing aquesta opció també tindrà que servir per enviar les rutes emmagatzemades a la Taula de Routing, com si de propis Childs es tractes.

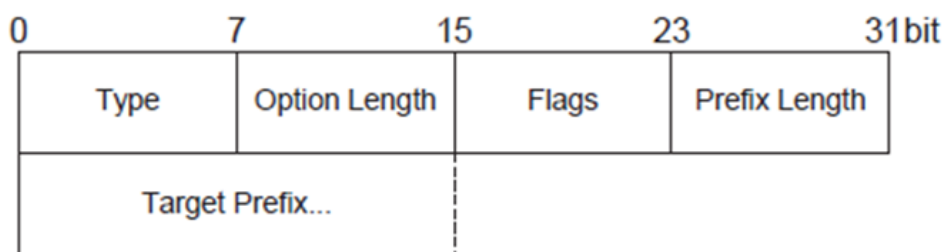


Figura 16 - Missatge DAO – Target Option

Els camps que componen la opció Target Option son:

**Option Type:** 0x05 (Valor prefixat per aquesta opció).

**Option Length:** Variable, indica la longitud del la opció en octets excloent els camps Type i Length.

**Flags:** Camp de 8-bit reservat per flags. El camp es te que inicialitzar a zero pel enviaador i te que ser ignorat pel receptor.

**Prefix Length:** Enter de 8-bit. Numero de bits inicials vàlids bits en el prefix IPv6.

**Target Prefix:** Camp variable per indicar la longitud d'una adreça IPv6, prefix, o grup multicast destí. El camp Prefix Length conte el numero de bits inicials vàlids bits en el prefix. Els bits

en el prefix després de prefix length estan reservats i es tenen que inicialitzar a zero pel emissor i tenen que ser ignorats pel receptor.

## Missatge DAO – Desconnexió d'un node – No-Path DAO

Aquí es presenta un tipus de missatge DAO que només s'utilitzarà en mode Storing. Si un node es vol desconnectar de la xarxa enviarà un missatge DAO anomenat No-Path. Amb aquest missatge es farà neteja del routing en la xarxa esborrant la resolució del node de les taules de routing. El missatge No-Path porta la opció Target i Transit amb un lifetime de 0, amb això s'assenyala la pèrdua d'accessibilitat al Target indicat. En la figura 17 es pot observar un procés exemple que seguiria una xarxa al rebre un missatge No-Path.

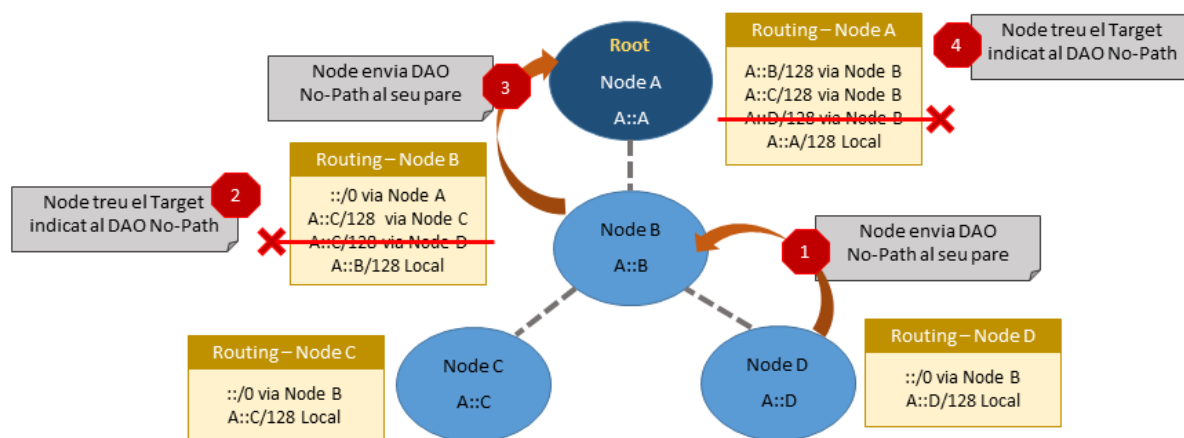


Figura 17 - Missatge DAO – No-Path

## Routing – Registres de la taula de routes

El llistat de registres que compondran la Taula de Routing serà la següent:

- **Advertising Neighbor Information**
- **IPv6 address**  
Direcció IPV6 on cal redirigir el paquet
- **Interface ID to which DAO parents has this entry been reported**  
Next hop MAC EUI64 on cal redirigir el paquet
- **Retry counter**  
Comptabilitza les vegades que s'ha provat l'accessibilitat a la ruta i a fallat
- **Logical equivalent of DAO Content:**  
Camps continguts en les capçaleres DAO que ens serviran per saber si cal renovar la ruta
  - **DAO-Sequence**  
Numero de seqüència per indicar que el DAO és nou
  - **Path Sequence**  
Numero de seqüència per indicar que el Path és nou
  - **DAO Lifetime**  
Temps de vida de la informació DAO
  - **DAO Path Control**  
Ens indica l'ordre de preferència dels successors

- **Destination Prefix (or address or Mcast Group)**

Camp que conté la Destinació on es vol accedir

## 2.6. Disseny de la implementació

En aquest apartat es plasmarà el disseny conceptual, per tal de tenir un enfoc clar de la implementació que caldrà realitzar. Analitzant i exposant primerament el funcionament dels diversos processos del protocol RPL, específicament en el mode Storing, degut a que es tracta del nou mode a implementar. Es plantejaran les estructures de dades necessàries i els processos lògics a desenvolupar per arribar a la solució final desitjada.

### Missatgeria DIO i DAO – Enviament i activadors

Tal com s'ha vist anteriorment el funcionament en l'enviament dels missatges DIO i DAO és diferent degut a les tasques que realitzen en cada cas. Tenint en compte només el Mode Storing, s'exposaran les diferències per a cada cas:

- **Missatge DIO**

Missatges en mode broadcast des de tots els nodes.

- **Missatge DAO**

Missatges en mode unicast cap el node Parent, de tots els nodes excepte el node root.

Caldrà establir sistemes activadors per cada tipus de missatge, per tal de llençar la tasca d'enviament amb uns temps establerts, o segons unes accions activadores. D'aquesta manera els activadors són:

- **Missatge DIO**

- ✓ Timer per activar enviament 100000ms

- **Missatge DAO**

- ✓ Timer per activar enviament 600000ms
- ✓ A la recepció d'un missatge DAO amb informació nova

Queda manifestat gràficament a la figura 18 les característiques, abans exposades, que tindran els missatges DIO i DAO.

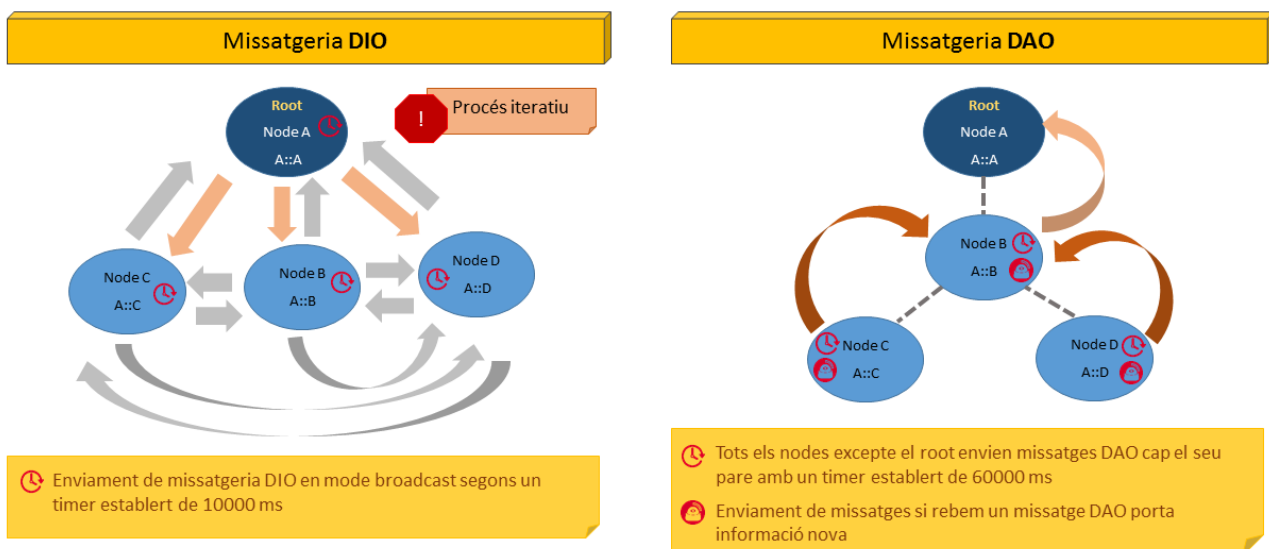


Figura 18 - Disseny de la implementació – Missatges DIO & DAO

## Missatgeria DIO – Implementació de la topologia

En aquest apartat s'exposarà tant el disseny de les dades necessàries per implementar-ho com els passos i processos que generarà la missatgeria DIO amb la seva creació, enviament i recepció.

Primer de tot es tindrà una llista de registres per tal d'emmagatzemar les dades necessàries bàsiques per que cada Node pugui conèixer en quina xarxa DODAG es troba i quina és la seva situació al respecte:

- **DODAGID** (Identificador del Node Root)
- **Prefix** (Prefix a la xarxa DODAG)
- **IPv6** (IP pròpia del Node)
- **Rank** (Numero identificador de posició relativa respecte la resta de nodes)
- **RPLInstanceID** (Identificador únic de la Xarxa DODAG)
- **MOP** (Flags del mode de treball del protocol RPL)
- **Pare Preferit** (MAC UEI64 o IPV6 del pare Preferit)

També serà necessari una taula multidimensional on cada registre de la taula contindrà les dades d'un veí:

- **ID** (identificador del registre)
- **MAC EUI64** (MAC del Veí)
- **Rank** (Rank del Veí)

A la figura 19 es pot veure gràficament els passos i processos d'inicialització des del node root de l'enviament de missatgeria DIO

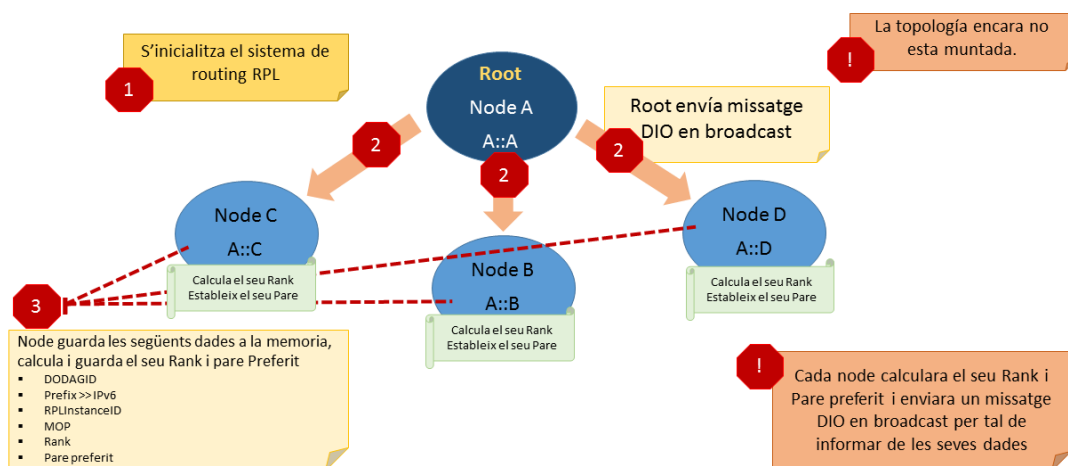


Figura 19 - Disseny de la implementació – Missatges DIO – Procés inicial

A la figura 20 es pot veure gràficament els passos i processos que es mantindran en l'enviament de missatgeria DIO per tots els nodes, una vegada succeït el procés inicial.

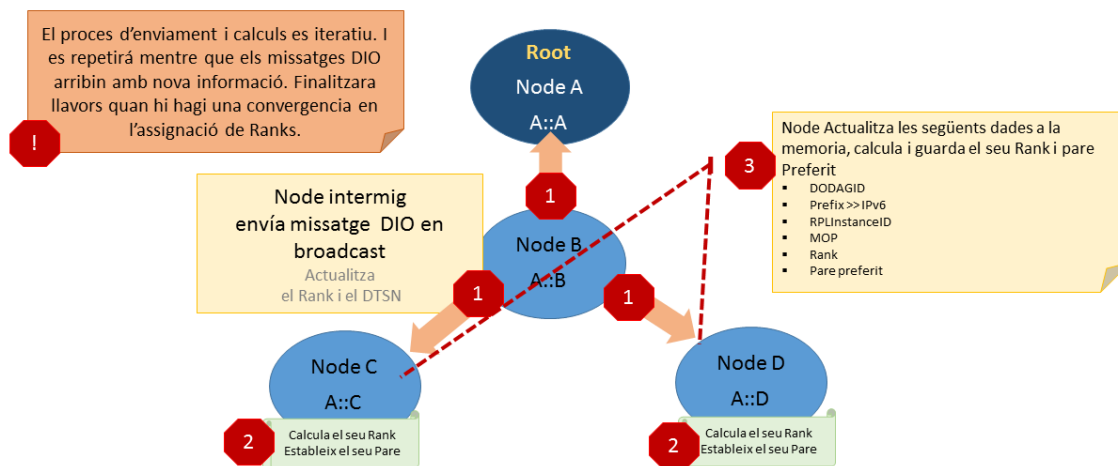


Figura 20 - Disseny de la implementació – Missatges DIO – Procés general

El procés general d'enviament de missatgeria DIO es un procés iteratiu on a través de la missatgeria DIO els Nodes aniran aconseguint informació de la xarxa i informant a la mateixa dels seus canvis de Rank per tal d'establir la topologia. Aquest procés iteratiu convergirà quan deixin d'haver canvis en els valors dels Ranks. El procés sempre estarà en marxa, amb enviaments constants programats de missatgeria DIO, per tal de controlar els canvis topològics.

## Missatgeria DAO – Implementació del Routing

En aquest apartat s'exposarà tant el disseny de les dades necessàries per implementar-ho com els processos que seguirà la missatgeria DAO a cada pas.

Primer de tot es tindrà una llista de registres per tal d'emmagatzemar les dades necessàries bàsiques per que cada Node pugi conèixer la situació del Routing a la xarxa DODAG on es troba:

- **DAO Timer** (Temps entre enviaments DAO)
- **DAO Sequence** (Numero de seqüència per indicar que el DAO és nou)
- **Opció K** (Opció habilitadora de l'enviament de respostes DAO-ACK)
- **Opció D** (Indica la presència del camp DODAGID)

Per tal de poder muntar la Taula de Routing tal com ja s'ha exposat amb anterioritat caldrà definir primerament una llista de registres que formaran cada registre de la taula. D'aquesta manera cada equip tindrà la seva Taula de Routing, on cada registre de la taula contindrà tots els registres necessaris:

- **IPv6 address** (IP del equip que publica la ruta)
- **Interface ID** (MAC EUI64 del equip que publica la ruta)
- **Retry counter** (Comptador de retries)
- **Logical equivalent of DAO Content** (Dades emmagatzemades extretes dels DAO):
  - **DAO-Sequence**
  - **Path Sequence**
  - **DAO Lifetime**
  - **DAO Path Control**



■ IPv6 Address Destination (IP del equip destí)

El muntatge final de la taula es pot observa en la figura 21.

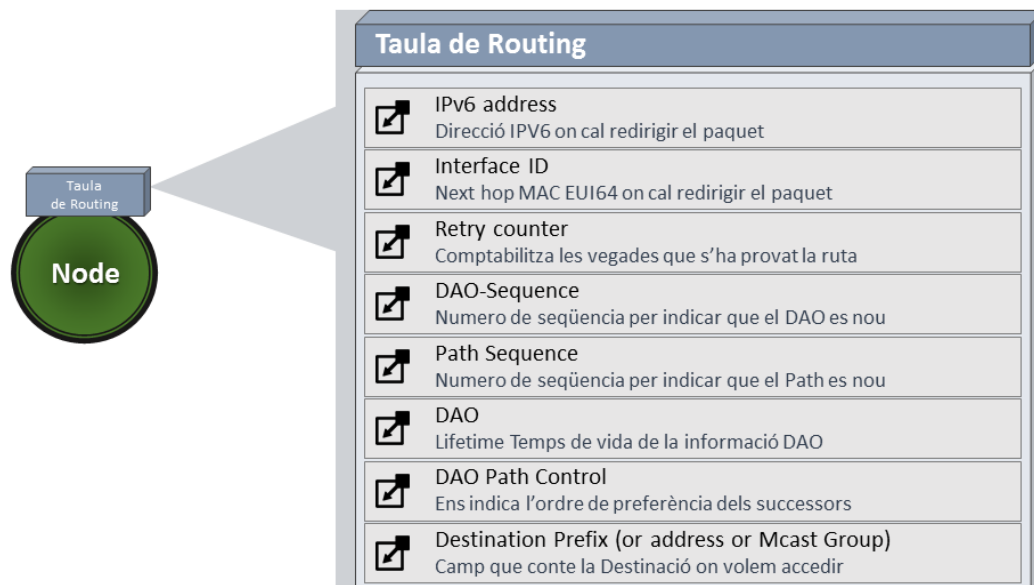


Figura 21 - Registres de la Taula de Routing

D'aquesta llista tal vegada no s'utilitzi el registre Retry counter, però es deixarà configurat per a possibles desenvolupaments futurs.

A la figura 22 s'exposa gràficament l'enviament i processos que es desencadenen amb els missatges DAO, i d'aquesta manera com es van omplint els registres abans exposats.

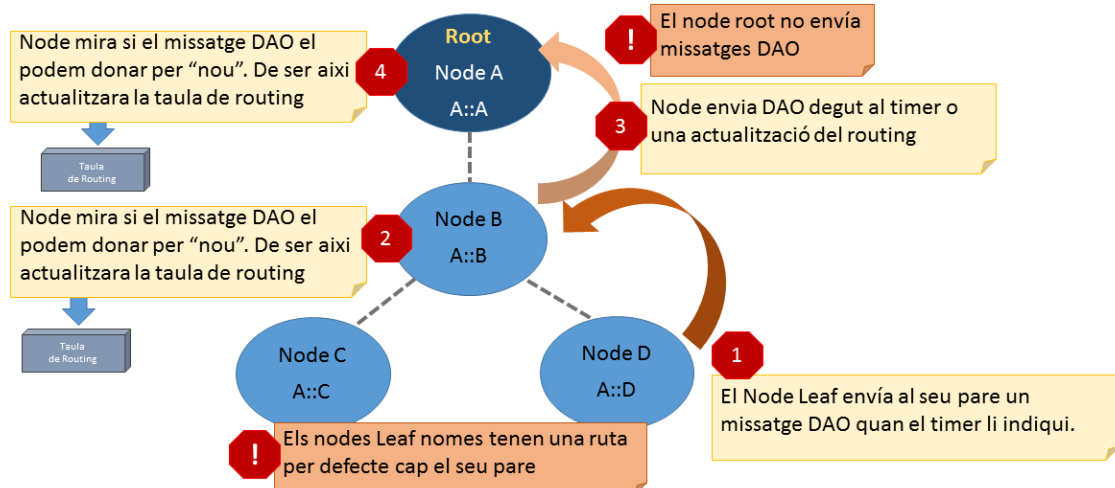


Figura 22 - Disseny de la implementació – Missatges DAO – Procés

Es tracta d'un procés continu on els missatges DAO transportaran la informació de Routing entre els Nodes. Amb aquesta informació es muntaran les Taules de Routing que posteriorment s'utilitzaran en els processos de Forwarding dels paquets rebuts. Per tal de mantenir el Routing correctament actualitzat en cas de desconexions de nodes, s'establiran

sistemes de neteja que faran que les rutes que no s'actualitzin s'acabin esborrant de la taula de rutes al finalitzar-se el seu Lifetime.

**Missatgeria DAO – Lògica de la taula de Routing**

Amb tota aquesta informació, passos, i processos caldrà establir una lògica que serveixi de guia per a poder tenir una idea clara dels passos i decisions que els Nodes hauran de dur a terme al tractar les dades rebudes en els missatge DAO, i d'aquesta manera com muntar la seva Taula de Routing. A la figura 23 es pot veure la representació gràfica de la lògica dissenyada pel muntatge de la Taula de Routing amb la missatgeria DAO.

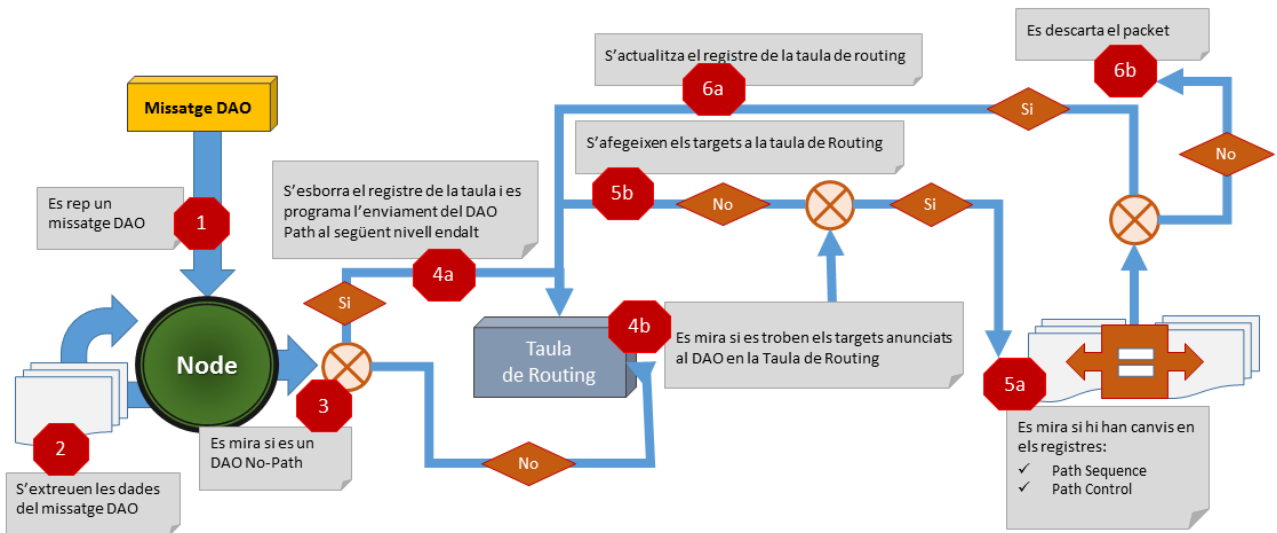


Figura 23 - Lògica de la taula de Routing – Missatges DAO

El concepte que es plasma a la lògica, es el comportament dels Nodes davant la recepció d'un missatge DAO. D'aquesta manera es planteja un seguit de passos i decisions per a comprovar si cal actualitzar la taula de rutes o si es farà cas omís de la informació rebuda. Dintre d'aquesta lògica es troben els processos que alimentaran la Taula de Routing.

Per tal d'evitar generar mes tràfic del necessari caldrà evitar l'enviament de informació de Routing sinó conté informació nova, d'aquesta manera es reduirà dràsticament el consum d'ample de banda i conseqüentment el consum energètic, tant important en aquest tipus de xarxes. Així que caldrà establir alguna mesura de control per tal d'enviar la major quantitat de informació nova, descartant d'enviar informació obsoleta.

**Missatgeria DAO - Routing – Lògica del Forwarding**

Una vegada establert el sistema d'emmagatzematge de les dades de Routing, caldrà centrar-se en el funcionament del Forwarding, on cada node tindrà que decidir que fer amb els paquets rebuts per tal de que aquests arribin a destinació. A la figura 24 es pot veure gràficament una seqüència d'enviament d'un paquet i quins passos i tasques es desencadenarien en els nodes en aquest procés.

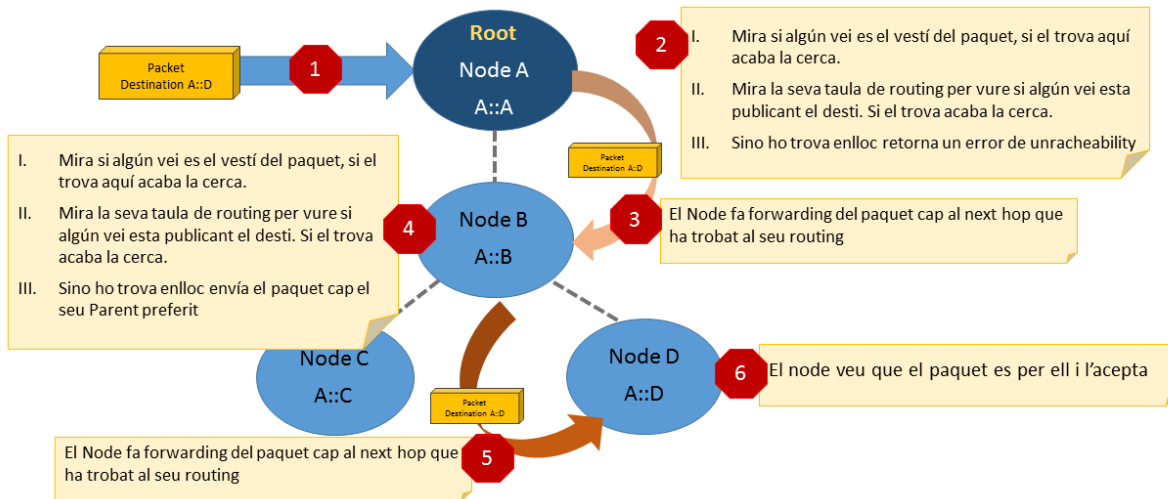


Figura 24 - Seqüència de Forwarding – Missatges DAO

Amb la informació, passos i processos es dissenya una lògica on es representen els passos i decisions que pren el Node davant l'arribada d'un paquet, per tal de poder fer Forwarding del paquet cap al seu destinatari. A la figura 25 es pot veure la representació gràfica de la lògica dissenyada de Forwarding.

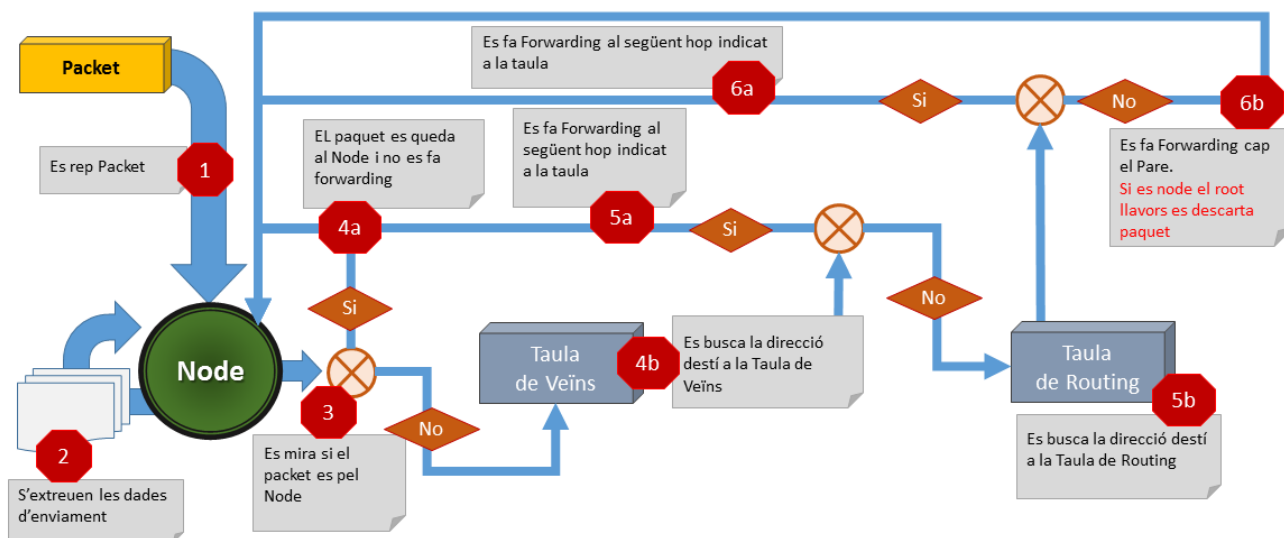


Figura 25 - Lògica de Forwarding – Missatges DAO

El concepte que es plasma a la lògica és el comportament de cada Node quan ha de realitzar les tasques de Forwarding. Sabent que s'implementa el mode Storing, els nodes deixen de llegir la capçalera de Source-Forwarding, on aquests no prenen decisions de routing sinó que envien el paquet al següent Node indicat pel Source-Forwarding. Ara caldrà plantejar com s'implementarà la Taula de Routing dintre de les tasques de Forwarding.

### 3. Disseny del Testbench

#### 3.1. Incís al Testbench

Cal tenir en compte que el Testbench que es dissenyarà per tal d'obtenir una imatge precisa del rendiment del protocol en cadascun dels modes de treball. D'aquesta manera les proves i mesures a seran específiques per aquest cas, i ens ajudaran a plantejar els comportament dels dos modes de treball del protocol.

En les topologies que s'establiran, es realitzaran simulacions d'efectes tals com:

- Pèrdua de paquets
- Desconnexions de nodes
- Moviments dels nodes entre les branques de la topologia

Amb aquests efectes es volen obtenir mesures més realistes. Sempre es tindrà en compte que el Testbench estarà limitat a allò que ens permeti les eines de simulació que es fan servir.

## 3.2. Mesures generals a prendre

Les mesures principals per a poder calcular la eficiència de un protocol son:

- **Temps de convergència del protocol**  
(inicial o rere un canvi)
- **Ample de banda utilitzat amb els missatges del protocol**  
(en el temps de convergència inicial)
- **Energia consumida amb els missatges del protocol**  
(en el temps de convergència inicial)
- **Latències en la comunicació entre nodes**

Coneixent aquestes mesures generalistes ens caldrà adaptar-les al cas concret del treball. Així, es proposaran mesures més adients, tenint sempre en compte les limitacions de les eines utilitzades.

## 3.3. Mesures específiques pel protocol RPL

Coneixent les mesures generals a prendre, caldrà establir pel cas del protocol RPL quines mesures específiques seran més adients. D'aquesta manera, s'estableixen les següents mesures específiques:

- **Temps de convergència del protocol**  
(inicial o rere un canvi)

Cal prendre varies mesures dels següents valors per tal d'obtenir el (MAX,MIN,MIT):

- ✓ Temps de convergència al iniciar el protocol
- ✓ Temps de convergència rere un canvi de topologia

- **Ample de banda utilitzat amb els missatges del protocol**  
(en el temps de convergència inicial)

Comptabilitzar missatges DIO i DAO:

- ✓ Quantitat de missatges DIO i DAO per iniciar el protocol
- ✓ Quantitat de missatges DIO i DAO rere un canvi de topologia

- **Energia consumida amb els missatges del protocol (en el temps de convergència inicial)**

Energia consumida pels missatges DIO i DAO:

- ✓ Energia consumida per iniciar el protocol
- ✓ Energia consumida rere un canvi de topologia

Aquests valors es calcularan directament multiplicant el numero de missatges pel consum estimat mig de 581,9  $\mu$ A [19]. Es fa referencia a un consum estimat mig, ja que per calcular el consum real de cada missatge enviat caldria no només mesurar la mida de cada missatge, sinó també tenir en compte moltes altres qüestions, cosa que ho fa impossible per un estudi com aquest.

- **Latències en la comunicació entre nodes**

Latències amb proves PING, o enviaments UDP o TCP des de fora del MESH.

### 3.4. Efectes a simular en les topologies

Per aconseguir unes mesures més reals caldria poder simular efectes en les topologies escollides, tals com:

- **PDR (Packet Delivery Ratio)**

S'han escollit dos valors de PDR, que es fixaran per a tots els enllaços en la simulació. Només s'utilitzen dos valors de PDR, per tal de no allargar les proves i presa de mesures. Ja que cal ajustar les proves a les més necessàries pel temps limitat d'aquest treball. Els valors i les argumentacions per la seva elecció són:

- ❖ **PDR al 100%**  
S'ha escollit aquest valor per tal de prendre mesures en una situació ideal on no hi ha pèrdua de paquets.
- ❖ **PDR al 75%**  
S'ha escollit aquest valor ja que es volia simular una xarxa amb pèrdues, establint un valor promig raonable, valor en base a un estudi de rendiment [20].

- **Desconnexions de nodes**

Es provocarà un tall en la connexió, esborrant la connexió en la topologia de xarxa, d'un node escollit per tal de simular desconnexió i així provocar que la xarxa refaci la seva topologia (Taula de Veïns, Taula de Parents i Taula de Routing).

- **Moviments dels nodes desconnectats a un altre branca**

Es tornarà a connectar el node a una altra branca de la topologia per tal de veure com es torna a associar el node en una altra posició. Aquesta prova es farà rere a una desconnexió del node, degut a que caldrà que el sistema i el propi node hagin fet neteja de les dades anteriors (Taula de Veïns, Taula de Parents i Taula de Routing).

### 3.5. Topologies escollides

S'han tingut que dissenyar topologies per tal de prefixar-les, enlloc de deixar que el sistema munti una topologia lliurement, degut a les següents raons:

- 1) **Limitació del OPENWSN**

El sistema OPENWSN, actualment, no accepta publicar més d'un Child als missatges DAO, així les topologies només poden ser lineals. Arbre centrat en el node root amb branques sense divisions.

## 2) Calen topologies idèntiques per poder realitzar la comparació

Per tal d'obtenir mesures comparables, caldrà que les topologies on es prenguin les mesures siguin iguals en els dos modes. Per això, cal prefixar les topologies i no deixar que sorgeixin aleatòriament.

Al final s'ha decidit per utilitzar un total de dues topologies. No s'han escollit més topologies, degut a que el conjunt de mesures a prendre i l'estudi deuen ajustar-se als temps establerts en el treball. Les topologies són:

### ■ Topologia 1

Topologia amb un total de 8 nodes distribuïts en dues branques. Aquesta topologia es va dissenyar pensant en una xarxa amb una profunditat de xarxa, que permetés tenir 4 capes. Per tal de veure el comportament dels modes Storing i Non-Storing, amb una xarxa amb suficients nodes, però no masses. Es pot veure la seva representació gràfica en la figura 26.

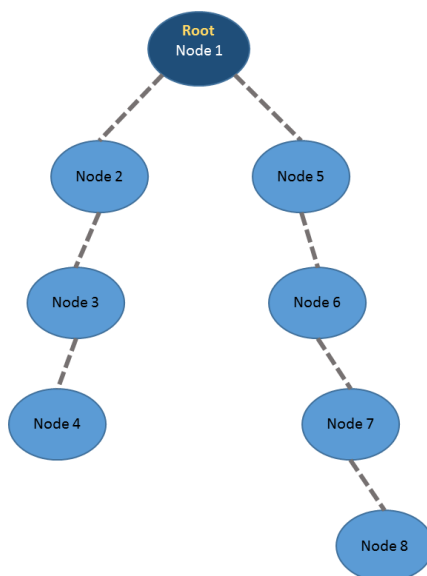


Figura 26 - Testbench – Topologia 1

### ■ Topologia 2

Topologia amb un total de 6 nodes distribuïts en dues branques, en aquesta topologia més senzilla s'aplicaran les simulacions de desconexió i reconexió de nodes, per tal de poder veure com es comportaran els modes Storing i Non-Storing al perdre i tornar a recuperar un node. Es pot veure la seva representació gràfica en la figura 27.

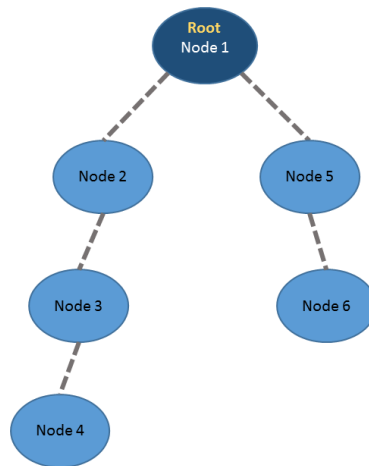


Figura 27 - Testbench – Topologia 2

### 3.6. Pla de treball

Amb les topologies ja establertes i coneixent quines mesures s'hauran de prendre, caldrà establir un pla de treball. D'aquesta manera s'estableix un pla de treball diferent per a cada topologia:

#### ■ Topologia 1

##### ➤ Cas PDR al 100% (Sense pèrdua de dades)

1. Prendre mesures del temps que triga el Routing en establir-se  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
2. Comptar els missatges DIO i DAO necessaris per establir el Routing  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
3. Realitzar Ping a tots els Nodes de la topologia per tal d'obtenir les latències  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
4. Desconnectar el Node 8 i prendre mesura de temps per saber quant triga en desaparèixer del Routing.  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
5. Connectar el Node 8, després de generar la seva total desconexió de la xarxa, sota el Node 4 i prendre mesura de temps per saber quant triga en aparèixer al Routing.  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*

##### ➤ Cas PDR al 75% (Amb 25% de pèrdua de dades)

1. Prendre mesures del temps que triga el Routing en establir-se  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
2. Comptar els missatges DIO i DAO necessaris per establir el Routing  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
3. Realitzar Ping a tots els Nodes de la topologia per tal d'obtenir les latències  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*

#### ■ Topologia 2

##### ➤ Cas PDR al 100% (Sense pèrdua de dades)

1. Prendre mesures del temps que triga el Routing en establir-se

*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*

2. Comptar els missatges DIO i DAO necessaris per establir el Routing  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
3. Realitzar Ping a tots els Nodes de la topologia per tal d'obtenir les latències  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
4. Desconnectar el Node 4 i prendre mesura de temps per saber quant triga en desaparèixer del Routing.  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
5. Connectar el Node 4, després de generar la seva total desconexió de la xarxa, sota el Node 6 i prendre mesura de temps per saber quant triga en aparèixer al Routing.  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*

➤ **Cas PDR al 75% (Amb 25% de pèrdua de dades)**

1. Prendre mesures del temps que triga el Routing en establir-se  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
2. Comptar els missatges DIO i DAO necessaris per establir el Routing  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
3. Realitzar Ping a tots els Nodes de la topologia per tal d'obtenir les latències  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
4. Desconnectar el Node 4 i prendre mesura de temps per saber quant triga en desaparèixer del Routing.  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*
5. Connectar el Node 4, després de generar la seva total desconexió de la xarxa, sota el Node 6 i prendre mesura de temps per saber quant triga en aparèixer al Routing.  
*Fer varies mesures per tal d'obtenir els valors en MAX/MIN/MIT*

### 3.7. Mesures obtingues – Anàlisi del comportament de la xarxa

Es farà un anàlisi concís amb el qual es plasmarà el comportament de la xarxa davant les diferents topologies i proves realitzades. Aquest curt però precís anàlisi servirà com a valor afegit en el estudi comparatiu més enllà de les mesures presses.

## 4. Documentació DEMO-1 Storing

### 4.1. Incís a la DEMO-1

En aquesta Demo s'implementarà i posteriorment s'aplicarà el Testbench dissenyat al protocol RPL treballant en Mode Storing. Com aquest mode de treball actualment no esta desenvolupat en el sistema OPENWSN, caldrà implementar-lo modificant el codi actual del sistema OPENWSN basant-se en el disseny conceptual proposat.

### 4.2. Tasques a realitzar en la DEMO

En la DEMO es desenvoluparan les següents tasques:

- **Implementació protocol RPL en mode Storing en una xarxa amb hardware simulat**
  - Anàlisi del codi de OPENWSN



- Modificació del codi de OPENWSN
- Proves de comprovació del codi implementat, i detecció de mancances

#### ■ Aplicació del Testbench a la solució final implementada

- Recull de mesures a la Topologia 1
- Recull de mesures a la Topologia 2

### 4.3. Implementació en el codi – Punts a tenir en compte

Caldrà tenir en compte els següents punts pel desenvolupament basant-nos en el disseny conceptual:

- **Cal establir un mitja per a poder canviar el mode de treball amb facilitat**
- **El nou mode necessita de canvis en la missatgeria del protocol (DIO i DAO)**
- **El nou mode necessita generar Taules de Routing en tots el Nodes**
- **El nou mode canvia el sistema de Routing de Source-Routing a Table-Routing**

També cal examinar el sistema OPENWSN, per saber com esta implementat en el codi. D'un primer anàlisi es pot definir que el sistema OPENWSN es compon de tres projectes diferenciats:

- **openwsn-fw**

<https://github.com/openwsn-berkeley/openwsn-fw>

OpenWSN firmware → Llenguatge C

Es tracta del firmware que funcionarà a cada Node Mote, virtual o real, que s'encarregarà de donar les següents capacitats dels nivells de la capa OSI:

- Capa 2 – Taula de veïns (MAC EUI64)
- Capa 3 – Routing (IPv6)
- Capa 4 – Transport (TCP, UDP)

- **openwsn-sw**

<https://github.com/openwsn-berkeley/openwsn-sw>

OpenWSN software → Llenguatge Python

Software base que s'encarregarà de simular les funcions del node Root.

- **coap**

<https://github.com/openwsn-berkeley/coap>

Llibreria CoAP per Python → Llenguatge Python

Aquest paquet integra el Constrained Application Protocol (CoAP) desenvolupat pel grup de treball IETF CORE, està definit al RFC7252. En particular s'implementa:

- Client COAP
- Servidor COAP amb un numero arbitrari de recursos

Dels projectes exposats que formen el sistema OPENWSN només caldrà fer canvis al codi en:

- **openwsn-fw** (OpenWSN firmware)  
Canvis per tal de canviar el comportament dels Nodes

- **openwsn-sw** (OpenWSN software)  
Canvis per tal de canviar el comportament del Node Root

#### 4.4. Anàlisi del codi

En aquest apartat s'identifiquen les tasques específiques que realitzen actualment, abans d'implementar cap canvi, en el software i el firmware del sistema OPENWSN.

- **openwsn-fw** (OpenWSN firmware)  
Tasques dels MOTES:
  - Missatgeria DIO
  - Missatgeria DAO
  - Forwarding + Routing
    - Forwarding dels paquets DAO directament cap el DODAG root (mode Non-Storing)
    - Source-Routing dels paquets cap un destí des del DODAG root (mode Non-Storing)
- **openwsn-sw** (OpenWSN software)  
Tasques del DODAG root:
  - Muntar la taula de Parents per tal de tenir un gràfic de la topologia
  - Tasques de Routing:
    - Source-Routing dels paquets cap un destí tret de la taula de Parents (mode Non-Storing)

#### 4.5. Codi desenvolupat - GitHub

El codi desenvolupat per aquesta DEMO es troba a GitHub per tal de poder realitzar un control de versions i mantenir una còpia de seguretat al núvol. S'indica els links per accedir al codi pujat a Github:

- **openwsn-fw** (OpenWSN firmware)  
<https://github.com/sekoto/openwsn-fw/tree/develop-STORING-MODE>
- **openwsn-sw** (OpenWSN software)  
[https://github.com/sekoto/openwsn-sw/tree/develop\\_STORING-MODE](https://github.com/sekoto/openwsn-sw/tree/develop_STORING-MODE)

#### 4.6. Resolució dels punts - Resum

En aquest apartat s'exposarà a mode resum les implementacions realitzades per solucionar els punts claus abans exposats.

- **Cal establir un mitja per a poder canviar el mode de treball amb facilitat**

Per tal de poder canviar el mode de treball del protocol RPL d'una forma centralitzada i senzilla, s'ha afegit una opció modificable que permetrà només canviant un valor escollir un mode o un altre.

- **El nou mode necessita de canvis en la missatgeria del protocol (DIO i DAO)**

S'han implementat els canvis necessaris en els missatges DIO i DAO segons el RFC6550.

- Els missatges DIO segons el estudiat només canvia el valor de MOP, on passa de 1 (001) en mode Non-Storing a 3 (011) en mode Storing.
- Els missatges DAO s'envien al Pare preferit enlloc del Node Root. Es manté la estructura base del missatge DAO, i els canvis es produeixen en les Option:
  - *Transit Option* – Ja no es publica el Parent, i s'indica amb la opció length a 0
  - *Target Option* – No hi ha canvis rellevants, més enllà del paper que pren aquesta opció convertint-se en el centre de comunicació del sistema Table-Routing.

- **El nou mode necessita generar taules de Routing en tots els Nodes**

S'ha implementat una Taula de Routing a cada Node MOTE. Aquesta Taula emmagatzema la informació rebuda a través dels missatges DAO dels nodes situats per sota:

- Childs del node que envia el missatge DAO
- Rutes emmagatzemades en el node envia el missatge DAO

El mode Storing seguirà la lògica exposada en la figura 23 per muntar la Taula de Routing.

- **El nou mode canvia el sistema de Routing de Source-Routing a Table-Routing**

S'ha aconseguit arribar al comportament esperat en el disseny conceptual, aplicant les següents modificacions segons el tipus de node:

- **Node Root**

S'ha modificat la taula de Parent per tal de que es comporti com a taula de Routing quan s'activi el mode Storing i pugui aplicar el Table-Routing sense afegir als paquets la capçalera de Source-Routing.

- **Node Genèric**

S'ha afegit una taula Routing tal com s'ha exposat en el punt anterior, que juntament amb la taula de veïns i una regla per defecte que enruta cap el Parent preferit sinó coneix el destí formen el procés de Table-Routing. Així els nodes ja no llegeixen la capçalera de Source-Routing sinó que prenen decisions de Routing segons les dades que tenen. Funcionarà tal com s'exposava al disseny conceptual en la figura 25 quan es faci Forwarding d'un paquet.

## 4.7. Resolució dels punts - Exposició

En aquest apartat s'explicarà, en base al disseny conceptual, com s'ha implementat en el codi de OPENWSN els punts claus plantejats anteriorment. Dintre dels punts claus s'exposarà on es realitzen els canvis, amb explicacions breus i concretes, i indicant els punts exactes de canvi per tal d'arribar a la solució.

S'exposarà molt breument, sense entrar en detall, els casos del mode Non-Storing per tal de diferenciar els comportaments implementats per a cada mode.

❖ Cal establir un mitja per a poder canviar el mode de treball amb facilitat

Opció per escollir Mode RPL

*openwsn-fw (OpenWSN firmware)*

S'ha implementat aquest punt en el mòdul "opendefs.h", on s'emmagatzemen les dades generals del funcionament del sistema, una variable global anomenada *RPLMODE* amb la

qual es controla en la resta de mòduls en quin mode s'està treballant. D'aquesta manera només canviant el valor de *RPLMODE* el firmware complet canvia de comportament.

Punt de canvi Variable global RPLMODE

- **RPLMODE = 1** → Storing Mode
- **RPLMODE = 0** → Non-Storing Mode

En primera instància es va pensar en duplicar els fitxers on caldria fer els canvis de codi, però després d'un segon anàlisi es va arribar a la conclusió de que caldria duplicar masses fitxers i que això només generaria més problemes quan es volguessin implementar noves funcions, ja que caldria fer-ho a més quantitat de fitxers, amb estructures de codi diferents.

Finalment es va decidir implementar-ho en els fitxers originals canviant el comportament específicament allà on calgués, a través d'estructures condicionals o d'opcions.

Opció per escollir Mode RPL *openwsn-sw (OpenWSN software)*

S'ha implementat un nou mòdul anomenat "confglobal.py" per tal de donar d'alta una variable global anomenada *rplmode* que farà que el comportament del software canviï. Així canviant el valor de *rplmode* canviarà automàticament el mode de treball de tot el software.

Punt de canvi Variable global rplmode

- **rplmode = 1** → Storing Mode
- **rplmode = 0** → Non-Storing Mode

També cal dir que s'ha implementat un sistema per tal que es faci automàticament el canvi de mode, a través de la recepció del primer missatge DAO, on canvia el mode de treball depenent del valor del camp Option Length en la Transit Option.

Punt de canvi Canvi automàtic de la variable rplmode

- **Option Length igual a 0** → rplmode = 1
- **Option Length diferent de 0** → rplmode = 0

❖ El nou mode necessita de canvis en la missatgeria del protocol (DIO i DAO)

Canvis en la missatgeria DIO *openwsn-fw (OpenWSN firmware)*

S'han implementat els següents canvis en la missatgeria DIO, que s'activaran segons el valor de la variable global RPLMODE.

Punt de canvi Valor de MOP

- **RPLMODE = 1 (Storing Mode)**  
Mode of Operation = 3 (011)
- **RPLMODE = 0 (Non-Storing Mode)**  
Mode of Operation = 1 (001)

Com es pot observar els canvis en la missatgeria DIO han sigut mínims, ja que només ha calgut fer que canviés una variable, amb la que justament s'indica el mode de funcionament del protocol RPL a la resta de la xarxa.

Canvis en la missatgeria DIO

*openwsn-sw (OpenWSN software)*

S'han implementat els següents canvis en la missatgeria DIO, que s'activaran segons el valor de la variable global `rplmode`.

Punt de canvi    Valor de MOP

- **rplmode = 1 (Storing Mode)**  
Mode of Operation = 3 (011)
- **rplmode = 0 (Non-Storing Mode)**  
Mode of Operation = 1 (001)

Com es pot observar els canvis en la missatgeria DIO han sigut mínims, ja que només ha calgut fer que canviés una variable que justament indica el mode de funcionament del protocol RPL.

Canvis en la missatgeria DAO

*openwsn-fw (OpenWSN firmware)*

S'exposaran els canvis amb explicacions generalistes, enfocades als canvis principals fets, ja que s'aprofundirà en aquests canvis en les tasques associades al Routing més endavant. S'han implementat els següents canvis en la missatgeria DAO, que s'activaran segons el valor de la variable global `RPLMODE`.

Punt de canvi    Enviament d'un missatge DAO

- **RPLMODE = 1 (Storing Mode)**
  - S'envien els missatges DAO cap al seu DODAG Parent en lloc del DODAGID (DODAG root), obtenint la adreça de destinació del paquet a partir d'una funció que extreu la adreça del Parent preferit.
  - Les opcions del missatge DAO poden contenir informació de Routing.
- **RPLMODE = 0 (Non-Storing Mode)**

- S'envien els missatges DAO cap el DODAGID (DODAG root).
- Les opcions del missatge DAO només inclouen el Parent i Childs del node que ho envia.

#### Punt de canvi | Recepció d'un missatge DAO

##### ■ RPLMODE = 1 (Storing Mode)

Al rebre un missatge DAO es tractaran les dades, segons la lògica exposada en el disseny conceptual (figura 23):

- A. S'extrauran les dades de la base del missatge DAO bolcant-les a les variables de memòria establertes
- B. Mitjançant un criteri d'opcions s'actuarà de forma diferent per a cada cas:
  - a. Transit Option  
S'extrauran les dades d'aquesta opció.
  - b. Target Option  
S'obtindrà d'aquesta opció les dades necessàries per muntar el Routing

##### ■ RPLMODE = 0 (Non-Storing Mode)

No s'han de rebre missatges DAO, així que retornarà un error

Com es pot observar els canvis en la missatgeria DAO han estat més profunds, fent que el sistema canviï la seva forma de tractar els missatges DAO tant en la seva recepció com enviament.

No s'han pogut implementar els missatges DAO No-Path ja que caldria que el firmware tingues un sistema de apagada mitjançant el qual abans de realitzar l'apagada del equip enviaria el missatge DAO No-Path per tal d'agilitzar la neteja de rutes. Encara que aquest missatge només agilitza la neteja en cas de que el node abandoni volent la xarxa DODAG, no serveix pas per les pèrdues de connexió amb els nodes.

#### Canvis en la missatgeria DAO | *openwsn-sw (OpenWSN software)*

S'exposaran els canvis amb explicacions generalistes, enfocades als canvis principals fets, ja que s'aprofundirà en aquests canvis en les tasques associades al Routing més endavant. S'han implementat els següents canvis en la missatgeria DAO, que s'activaran segons el valor de la variable global `rplmode`.

#### Punt de canvi | Recepció d'un missatge DAO

##### ■ `rplmode = 1 (Storing Mode)`

Al rebre un missatge DAO el tractarà de la següent forma:

1. S'extrauran les dades de la base del missatge DAO bolcant-les a les variables de memòria establertes
2. Mitjançant un sistema d'opcions mirarà quin tipus d'opció DAO esta llegint:
  - a. *Transit Option*  
S'extrauran les dades d'aquesta opció, les quals serviran per decidir en quin mode esta treballant la xarxa DODAG.
  - b. *Target Option*  
Amb aquesta opció es rebran tant els Childs dels nodes que es tenen per sota, com les rutes que tinguin aquests nodes.

■ **rplmode = 0 (Non-Storing Mode)**

El DODAG root rep tots els missatges DAO

Com es pot observar els canvis en la missatgeria DAO han estat mes profunds, en aquest cas la part de software no genera missatgeria DAO, per tant, només ha calgut canviar la forma de tractar els missatges DAO a la recepció.

❖ El nou mode necessita generar Taules de Routing en tots el Nodes

Taula de Routing

*openwsn-fw (OpenWSN firmware)*

Es crea una taula per tal d'emmagatzemar les dades de Routing en els nodes MOTE. En aquesta taula s'emmagatzemaran les dades següents indicades principalment al disseny conceptual.

Punt de canvi Taula de Routing

➤ **REGISTRE DE LA TAULA DE ROUTING**

- used** → Flag booleà per tal de controlar si el registre esta en us o lliure
- advertneighinf** → Advertising Neighbor Information (actualment sense us)
- addr\_128b** → IPv6 del node que anuncia la adreça
- addr\_64b** → MAC del node que anuncia la adreça
- retcount** → Retry counter (actualment sense us)
- DAOSequence** → DAO-Sequence
- PathSequence** → Path-Sequence
- PathLifetime** → Path-Lifetime
- destination** → IPv6 de la adreça destí
- tosend** → Flag booleà per saber si la ruta cal enviar-la
- scout** → Comptador de numero de cops que ja s'ha enviat la ruta

➤ **ESTRUCTURA DE LA TAULA DE ROUTING**

**tosend** → Flag booleà per saber si toca enviar una ruta  
**routes[*MAX\_ROUTE\_NUM*]** → Invocació als registre de la taula  
**timerIdRT** → Timer per establir la tasca de neteja de la  
**RTPeriod** → Duració en ms del timeout de timerIdRT  
**delayRT** → Numero d'esdeveniments timerIdRT abans d'executar la neteja

Ha calgut definir unes variables globals per tal de controlar els diversos aspectes relacionats amb el Routing que s'està implementant.

#### Punt de canvi Variables globals de Routing

**MAX\_ROUTE\_NUM** → Numero màxim de rutes que emmagatzemarà un Node  
**MAX\_ROUTE\_SEND** → Numero màxim de rutes que s'enviaran en un missatge DAO  
**RTAGING** → Valor d'envelliment que es restarà al Lifetime de les rutes emmagatzemades

Juntament amb la taula de Routing s'han creat un conjunt de funcions que permetran interactuar amb la taula, obtenint diferents paràmetres.

#### Punt de canvi Funcions de interacció Taula de Routing

**registerRoute** → Per afegir o actualitzar una ruta  
**isRoute** → Per comprovar si una adreça esta a la taula de rutes  
**posRoute** → Per obtenir la posició d'una adreça en la taula de rutes  
**removeRoute** → Per esborrar una ruta de la taula de rutes  
**routes\_getNumRoutes** → Per saber quantes rutes hi han emmagatzemades

Cal dir que s'ha implementat un nou timer i funcions associades per tal d'executar periòdicament un sistema de neteja en la taula de Routing.

#### Punt de canvi Tasca de neteja de Taula de Routing

**routetable\_read** → Funció que realitza la tasca de neteja:

- Si el Lifetime de la ruta encara es major de 0 li restarà el valor fixat amb la variable RTAGING. Si després de restar el Lifetime encara es positiu es mantindrà la ruta en la taula, sinó s'esborrarà la ruta inicialitzant tots els registres
- Si el Lifetime de la ruta es menor o igual a 0 s'esborrarà automàticament la ruta inicialitzant tots els registres

**routetable\_timer\_task** → Funció que invoca a la funció routetable\_read i estableix el període de la tasca

**routetable\_setRTPeriod** → Funció que estableix el schudeling de la funció routetable\_timer\_task.



Tal com s'ha explicat anteriorment es tornen a exposar els punts de canvi relacionats amb la missatgeria DAO, per tal d'aprofundir en els processos relacionats amb el Routing.

Punt de canvi    Enviament d'un missatge DAO

#### ■ RPLMODE = 1 (Storing Mode)

- Al enviar-se els missatge cap el DODAG Parent, el Routing deixa d'estar centralitzat i tots els nodes tenen que interpretar els missatges DAO.
- S'envien les següents dades, que establiran el Routing, amb els missatges DAO:
  - Childs (nodes inferiors)
  - Rutes de la Taula e Routing
- El timer d'enviament dels missatges DAO s'ajusta automàticament depenent del volum de rutes emmagatzemades a la Taula de Routing. D'aquesta manera es dona la capacitat d'enviar tota la informació necessària cap a els nodes superiors.

#### ■ RPLMODE = 0 (Non-Storing Mode)

- S'envien els missatges DAO cap el DODAGID (DODAG root).
- Les opcions del missatge DAO només inclouen el Parent i Childs del node que ho envia.

La limitació coneguda en la mida dels missatges de 127 Bytes màxim, esdevé de una limitació del buffer dels equips MOTE. Això fa que els missatges DAO únicament puguin portar una opció Transit i una opció Target.

Com actualment encara no s'ha desenvolupat un sistema de fragmentació de paquets per tal de pal·liar aquesta limitació, s'ha desenvolupat un sistema de balanceig per tal de permetre enviar totes les dades que els Nodes en aquest mode emmagatzemen.

Per tal d'implementar aquest balanceig s'han creat variables dintre de la taula de Routing.

Punt de canvi    Taula de Routing – Variables pel balanceig

#### ➤ REGISTRE DE LA TAULA DE ROUTING

**tosend** → Flag booleà per saber si la ruta cal enviar-la

**scount** → Comptador de numero de cops que ja s'ha enviat la ruta

#### ➤ ESTRUCTURA DE LA TAULA DE ROUTING

**tosend** → Flag booleà per saber si toca enviar una ruta

Amb aquestes variables es prendran decisions en l'enviament dels missatges DAO.

Punt de canvi    Balanceig en l'enviament de missatges DAO

### ■ RPLMODE = 1 (Storing Mode)

Es presenta un balanceig en dos nivells, per tal de decidir quines dades s'enviaran amb el següent missatge DAO.

1) El primer nivell de balanceig es farà entre les següents dades per tal d'escollir quina incloure:

- Childs (nodes inferiors)
- Rutes de la Taula e Routing

Aquest balanceig serà tant simple com controlar el flag *tosend* si esta marcat i anant alternant el seu estat segons el que s'envia:

- Enviament de Childs → Marcat flag *tosend*
- Enviament de Rutes → Desmarcat flag *tosend*

2) El segon nivell de balanceig s'activarà en cas de publicar una ruta, per tal d'escollir quina es publicarà en el missatge DAO:

- I. Es mirarà si han rutes per publicar, sinó hi han s'enviaran els Childs.
- II. Si hi han rutes es recorrerà tota la taula mirant quines tenen marcat el flag *tosend*, apuntant la posició. Al final s'obindrà la ruta per enviar que tingui el comptador *scout* mes petit, per tal de fer un balanceig entre totes les rutes equivalent.

### ■ RPLMODE = 0 (Non-Storing Mode)

No hi ha balanceig

Aquest balanceig garantirà que només s'enviïn rutes amb informació nova. Més endavant en la recepció dels missatges DAO s'exposarà com s'emplenaran les variables relacionades amb el balanceig.

#### Punt de canvi    Recepció d'un missatge DAO

### ■ RPLMODE = 1 (Storing Mode)

Al rebre un missatge DAO aplicant la lògica del disseny conceptual de la figura 23 es seguiran els següents processos:

1. S'extrauran les dades de la base del missatge DAO apuntant-les a les variables de memòria per després poder-les consultar i prendre decisions.
2. Mitjançant un criteri d'opcions s'actuarà de forma diferent per a cada cas:

**a.** *Transit Option*

S'extrauran les dades d'aquesta opció, les quals serviran per més endavant decidir si les dades de Routing rebudes son noves o ja estan emmagatzemades.

**b.** *Target Option*

Amb aquesta opció es rebran les publicacions dels nodes Child, que publicaran els seus nodes Child, que tenen per sota a 1-hop, i totes les rutes emmagatzemades a la seva taula de Routing.

Per a cada Target Option rebut es realitzarà una comprovació, seguint la lògica plantejada al disseny conceptual:

- I. En cas que la ruta existeixi i les dades obtingudes del DAO base i de la Transit Option siguin noves s'actualitzarà el registre i es marcarà el flag *tosend*, en cas contrari no es farà res amb les dades rebudes. Sinó es trobes la ruta a la taula es passarà al següent pas.
- II. En cas que no es trobi la adreça s'afegirà en la Taula de Routing marcant el flag *tosend*.

#### ■ RPLMODE = 0 (Non-Storing Mode)

No s'han de rebre missatges DAO, així que retornarà un error

Amb aquests canvis exposats s'estableix el sistema de Routing als Nodes. Garantint que la Taula de Routing estarà actualitzada i que no s'enviaran dades innecessàries de Routing a traves de la xarxa.

Taula de Routing

*openwsn-sw (OpenWSN software)*

Per tal d'implementar la Taula de Routing en el DODAG root s'ha aprofitat la taula de parents canviant la seva estructura per tal d'afegir un sistema de neteja de rutes.

Punt de canvi

Taula de Parents – Informe de canvis

En la versió antiga de la taula de Parents tenia la següent relació per recollir dades:

*Diccionari ( Adreça Node : Llista (Adreça Parents) )*

D'aquesta manera era impossible poder afegir cap variable mes a la relació muntada. Així que s'han realitzat canvis obtenint una nova relació de dades:

*Diccionari ( Adreça Node : Diccionari ( 'parents': Llista ( Diccionari ( 'parent' : Adreça Parent, 'timestamp' : Timestamp )*

Amb la nova relació creada no només s'ha pogut afegir una variable timestamp per tal de controlar la neteja de rutes, sinó que s'ha obert la possibilitat de poder afegir moltes mes variables amb gran facilitat.

Aquests canvis s'han aplicat a la funció ja existent *updateParents*, la cual s'encarrega d'actualitzar les dades de la Taula de Parents. A més, ha calgut actualitzar tant les invocacions a la funció *updateParents* com les consultes de la Taula de Parents.

Per tal de implementar el sistema de neteja ha calgut crear noves funcions que conjuntament formen la solució necessària.

**getstamp** → Funció que retorna el timestamp (segons)

**cleanParents** → Funció creada per realitzar la neteja de la Taula de Parents

En aquesta funció enlloc de tenir un comptador regressiu que quan arribi a zero s'esborrarà el registre associat, el funcionament anirà relacionat a un timestamp que s'actualitzarà amb la informació rebuda en els missatges DAO.

La lògica es basa en revisar tota la taula de Parents. Per cada cas resta el timestamp del registre del timestamp actual, i comparant-lo amb un temps màxim de permanència establert es decidirà d'esborrar-lo si es supera aquest valor.

**LoopCleaner** → Funció que executa periòdicament la funció cleanParents segons un temps periòdic establert.

Tal com s'ha explicat anteriorment es tornen a exposar els punts de canvi relacionats amb la missatgeria DAO, per tal d'aprofundir en els processos relacionats amb el Routing.

■ **rplmode = 1 (Storing Mode)**

Al rebre un missatge DAO el tractarà de la següent forma:

1. S'extrauran les dades de la base del missatge DAO bolcant-les a les variables de memòria establertes.
2. Mitjançant un sistema d'opcions mirarà quin tipus d'opció DAO està llegint:
  - a. *Transit Option*  
S'emmagatzemarà la adreça del DODAGID com a Parent del node que hagi enviat el paquet DAO, que serà emmagatzemat com destí.
  - b. *Target Option*  
S'emmagatzemaran les adreces dels Childs rebudes com a destinació i el node que ho envia com a Parent.

■ **rplmode = 0 (Non-Storing Mode)**

El DODAG root rep tots els missatges DAO

D'aquesta manera s'ha obtingut aprofitant el sistema ja muntat pel mode Non-Storing una simulació de la Taula de Routing. Cal dir que la implementació del sistema de neteja de Parents serveix pels dos modes de treball, afegint d'aquesta manera una nova funcionalitat al sistema OPENWSN que necessitava de bon grat.

❖ El nou mode canvia el sistema de Routing de Source-Routing a Table-Routing

Per tal de implementar el Table-Routing en el DODAG root ha calgut canviar el comportament del sistema segons la variable global rplmode.

Punt de canvi    Enrutament de paquets cap a la xarxa DODAG

■ **rplmode = 1 (Storing Mode)**

El DODAG root llegint el la Taula de Parents obtindrà el next-hop, i enviarà el paquet sense muntar la capçalera de Source-Routing.

■ **rplmode = 0 (Non-Storing Mode)**

El DODAG root llegint el la Taula de Parents muntarà la capçalera de Source-Routing i enviarà el paquet al next-hop.

D'aquesta manera es veu que el canvi principal en el comportament d'aquesta part serà la eliminació de la capçalera de Source-Routing [21] [22] en el mode Storing, per tal de que els Nodes al llegir el paquet prenguin les seves decisions.

Table-Routing

*openwsn-fw (OpenWSN firmware)*

S'han implementat canvis per tal de que els nodes MOTE apliquin Table-Routing modificant el comportament d'aquests en el Forwarding dels paquets rebuts, seguint la lògica dissenyada en la figura 25.

Punt de canvi    Forwarding

■ **RPLMODE = 1 (Storing Mode)**

Al rebre un paquet es seguiran els següents passos:

1. Es mirarà la adreça destí del paquet
2. Es buscarà segons la següent lògica per on encaminar el paquet per arribar al destí:
  - I. Si el paquet es reconeix propi no es passarà al següent salt i acaba la cerca
  - II. Si l'adreça destí es un veí a 1-hop es redirigeix el paquet cap el veí. Sinó ho es troba es passa al següent pas.
  - III. Si es troba el destí a la Taula de Routing, es redirigeix el paquet cap el veí a 1-hop que ha publicat la ruta. Sinó ho es troba es passa al següent pas.
  - IV. S'envia el paquet cap al DODAG Parent el paquet, pujant en la topologia de arbre, fins trobar un node que coneixerà la adreça destí.

■ **RPLMODE = 0 (Non-Storing Mode)**

Al rebre un paquet nomes llegirà la capçalera de Source-Routing treient l'ultima resolució, i enviarà el paquet cap el next-hop. Així seguirà fent Forwarding fins que arribi al destí.

Per tal d'evitar bucles el sistema tenia implementat una protecció que nomes tenia en compte que els paquets amb Source-Routing anaven cap a baix de la topologia, i que la resta de Routing era cap endalt de la topologia. Això nomes es correcte si el sistema

nomes te únicament el mode de treball Non-Storing. D'aquesta manera s'ha corregit aplicant una solució més genèrica pels dos mode.

**Punt de canvi**    Avoiding loops

Primer de tot caldrà utilitzar el flag O al enviar els paquets:

- Flag O marcat → Direcció del paquet Downstream
- Flag O desmarcat → Direcció del paquet Upstream

D'aquesta maner tenint el flag O i el Rank del Node que envia i del que rep es seguirà la següent lògica per tal de comprovar els loops:

Si el Flag O marcat (Downstream):

- a) Rank del node que envia major que el del receptor → Hi ha loop
- b) Rank del node que envia menor que el del receptor → No hi ha loop

Si el Flag O desmarcat (Upstream):

- a) Rank del node que envia major que el del receptor → No hi ha loop
- b) Rank del node que envia menor que el del receptor → Hi ha loop

Tots aquells paquets que no trobin el seu destí degut a una caiguda del node, però que a nivell de Routing encara es publica degut al temps necessari per la neteja, tornaran al node DODAG root que els acabarà descartant.

Tal com s'ha pogut veure els canvis realitzats en aquesta part han esdevingut dels canvis realitzats en la part de software del sistema OPENWSN, on s'ha tret la capçalera de Source-Routing. D'aquesta manera els nodes deixen de fer Sorce-Routing, i per tant ha calgut afegir a les resolucions de Forwarding la nova Taula de Routing.

La correcció en la detecció de loops ha sigut clau ja que no s'estava aplicant de forma genèrica. Així ara funcionarà controlant els possibles loops mirant les dades correctes per tots dos modes.

#### 4.8. Testbench -- Topologia 1

- **Temps de convergència del protocol al iniciar**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
02:44.93	02:34.85	02:39.86
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
03:52.49	01:57.03	03:01.07

- **Temps de convergència del protocol rere un canvi en la topologia (Desconnexió node)**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT

02:47.79	02:37.41	02:43.39
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
03:07.94	03:06.48	03:07.14

- **Temps de convergència del protocol rere un canvi en la topologia (Reconnexió node)**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:41.07	00:33.26	00:33.02
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
01:25.44	01:05.29	01:18.49

- **Ample de banda utilitzat amb els missatges del protocol (en el temps de convergència inicial)**

Cas PDR al 100% (Sense pèrdua de dades)			
<b>Missatges DIO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	31	30	30
2	31	30	30
3	31	30	30
4	31	30	30
5	31	30	30
6	31	30	30
7	31	30	30
8	31	30	30
<b>TOTAL</b>	248	240	240
<b>Missatges DAO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	5	5	5
2	7	7	7
3	5	5	5
4	5	5	5
5	8	8	8
6	5	5	5
7	5	5	5
8	5	5	5
<b>TOTAL</b>	45	45	45
<b>TOTAL Missatges</b>			
Numero MAX	Numero MIN	Numero MIT	
293	285	285	
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
<b>Missatges DIO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	43	25	36
2	43	25	36
3	43	25	36
4	43	25	36
5	43	25	36

6	43	25	36
7	43	25	36
8	43	25	36
<b>TOTAL</b>	344	200	288
<b>Missatges DAO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	7	4	5
2	11	4	8
3	7	4	6
4	7	4	6
5	10	5	8
6	10	4	7
7	7	4	6
8	7	4	6
<b>TOTAL</b>	66	33	52
<b>TOTAL Missatges</b>			
Numero MAX	Numero MIN	Numero MIT	
410	233	340	

- **Energia consumida amb els missatges del protocol**  
(en el temps de convergència inicial)

Cas PDR al 100% (Sense pèrdua de dades)		
<b>Missatges DIO</b>		
Consum MAX	Consum MIN	Consum MIT
144,31 mA	139,66 mA	139,66 mA
<b>Missatges DAO</b>		
Consum MAX	Consum MIN	Consum MIT
26,19 mA	26,19 mA	26,19 mA
<b>TOTAL Missatges</b>		
Consum MAX	Consum MIN	Consum MIT
170,5 mA	165,85 mA	165,85 mA
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
<b>Missatges DIO</b>		
Consum MAX	Consum MIN	Consum MIT
200,17 mA	116,38 mA	167,59 mA
<b>Missatges DAO</b>		
Consum MAX	Consum MIN	Consum MIT
38,41 mA	19,20 mA	30,26 mA
<b>TOTAL Missatges</b>		
Consum MAX	Consum MIN	Consum MIT
238,58 mA	135,58 mA	197,85 mA

- **Latències en la comunicació entre node**

Cas PDR al 100% (Sense pèrdua de dades)				
<b>Latència + Paquets rebuts</b>				
Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2	675 ms	257 ms	325 ms	10/10
3	635 ms	496 ms	536 ms	10/10



4	802 ms	653 ms	704 ms	10/10
5	304 ms	260ms	282 ms	10/10
6	541ms	451 ms	493 ms	10/10
7	1139 ms	662 ms	805 ms	10/10
8	1354 ms	857 ms	991 ms	10/10
Cas PDR al 75% (Amb 25% de pèrdua de dades)				
<b>Latència + Paquets rebuts</b>				
Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2	2252 ms	264 ms	1038 ms	10/10
3	3485 ms	1128 ms	2171 ms	7/10
4	3967 ms	1256 ms	2671 ms	8/10
5	1434 ms	298 ms	884 ms	9/10
6	3238 ms	551 ms	1643 ms	9/10
7	3092 ms	1156 ms	2248 ms	6/10
8	3511 ms	1734 ms	2590 ms	5/10

#### 4.9. Testbench -- Topologia 2

- **Temps de convergència del protocol al iniciar**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
01:01.94	01:01.84	01:01.89
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
01:01.96	01:00.06	01:01.56

- **Temps de convergència del protocol rere un canvi en la topologia (Desconnexió node)**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
02:26.85	02:08.27	02:15.71
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
02:26.75	02:11.20	02:20.06

- **Temps de convergència del protocol rere un canvi en la topologia (Reconnexió node)**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:33.50	00:25.10	00:28.19
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:28.72	00:21.86	00:26.39

- **Ample de banda utilitzat amb els missatges del protocol (en el temps de convergència inicial)**

Cas PDR al 100% (Sense pèrdua de dades)			
<b>Missatges DIO</b>			
Node	Numero MAX	Numero MIN	Numero MIT

1	13	12	12
2	13	12	12
3	13	12	12
4	13	12	12
5	13	12	12
6	13	12	12
<b>TOTAL</b>	78	72	72
<b>Missatges DAO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	3	3	3
2	3	3	3
3	3	3	3
4	3	3	3
5	3	3	3
6	3	3	3
<b>TOTAL</b>	18	18	18
<b>TOTAL Missatges</b>			
Numero MAX	Numero MIN	Numero MIT	
96	90	90	
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
<b>Missatges DIO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	19	19	19
2	19	19	19
3	19	19	19
4	19	19	19
5	19	19	19
6	19	19	19
<b>TOTAL</b>	114	114	114
<b>Missatges DAO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	3	3	3
2	3	3	3
3	3	3	3
4	3	3	3
5	3	3	3
6	3	3	3
<b>TOTAL</b>	18	18	18
<b>TOTAL Missatges</b>			
Numero MAX	Numero MIN	Numero MIT	
132	132	132	

- **Energia consumida amb els missatges del protocol**  
(en el temps de convergència inicial)

Cas PDR al 100% (Sense pèrdua de dades)		
<b>Missatges DIO</b>		
Consum MAX	Consum MIN	Consum MIT
45,39 mA	41,90 mA	41,90 mA
<b>Missatges DAO</b>		
Consum MAX	Consum MIN	Consum MIT

10,47 mA	10,47 mA	10,47 mA
<b>TOTAL Missatges</b>		
Consum MAX	Consum MIN	Consum MIT
55,86 mA	52,37 mA	52,37 mA
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
<b>Missatges DIO</b>		
Consum MAX	Consum MIN	Consum MIT
66,34 mA	66,34 mA	66,34 mA
<b>Missatges DAO</b>		
Consum MAX	Consum MIN	Consum MIT
10,47 mA	10,47 mA	10,47 mA
<b>TOTAL Missatges</b>		
Consum MAX	Consum MIN	Consum MIT
76,81 mA	76,81 mA	76,81 mA

- **Latències en la comunicació entre node**

Cas PDR al 100% (Sense pèrdua de dades)				
<b>Latència + Paquets rebuts</b>				
Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2	632 ms	193 ms	254 ms	10/10
3	836 ms	319 ms	425 ms	10/10
4	808 ms	470 ms	544 ms	10/10
5	280 ms	170 ms	218 ms	10/10
6	730 ms	321 ms	399 ms	10/10
Cas PDR al 75% (Amb 25% de pèrdua de dades)				
<b>Latència + Paquets rebuts</b>				
Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2	833 ms	176 ms	342 ms	10/10
3	2150 ms	331 ms	683 ms	9/10
4	3724 ms	601 ms	1836 ms	10/10
5	1028 ms	181 ms	496 ms	10/10
6	2053 ms	318 ms	816 ms	9/10

#### 4.10. Testbench -- Anàlisi

En aquest apartat es reflexa de forma concisa el comportament observat en cadascuna de les topologies i processos dintre del Testbench, ja que també serà un punt de referència en el estudi comparatiu.

##### Topologia 1

Segons el que s'ha observat la xarxa, tant al 100% de PDR com al 75% de PDR, s'ha mantingut estable sense flapeig en les connexions entre els nodes.

En les proves de Latència s'ha pogut veure que els paquets coincidents amb l'enviament de missatges DAO tenen una major latència, el cas on es nota mes es quan es prenen les mesures amb el 75% de PDR.

## Topologia 2

Segons el que s'ha observat la xarxa, tant al 100% de PDR com al 75% de PDR, s'ha mantingut estable sense flapeig en les connexions entre els nodes.

En les proves de Latència s'ha pogut veure que els paquets coincidents amb l'enviament de missatges DAO tenen una major latència, el cas on es nota més és quan es prenen les mesures amb el 75% de PDR, encara que l'efecte ha sigut molt menor en aquesta topologia.

En el cas de les proves de desconnexió i reconexió del node 4 de la topologia, s'ha pogut observar que el temps de desconnexió ha sigut molt proper als 120 segons de vida establerts en el sistema de neteja del DODAG root. No s'ha observat efectes molt nocius en aquestes mesures al canviar de 100% de PDR a 75% de PDR.

## Conclusions generals

En termes generals el comportament ha sigut molt similar en les dues topologies, diferenciant l'augment substancial de missatges enviats per establir la xarxa, i la major latència observada en la Topologia 1. Aquest augment de la latència primerament ha sigut degut al volum major de node, i com a efecte secundari l'augment de l'ús de l'ample de banda directament proporcional al numero de nodes extres. Sabent això es pot treure un aproximació de la relació entre el volum de nodes i l'augment de latència produït, fent servir les mesures de valor promig en el cas sense pèrdues, i indicant el numero de salts des de el equip root fins el node indicat

TOPOLOGIA 1			
Cas PDR al 100% (Sense pèrdua de dades)			
Latència + Paquets rebuts			
Node	Latència MIT	Salts	Increment per salt
2	325 ms	1	325 ms
3	536 ms	2	211 ms
4	704 ms	3	168 ms
5	282 ms	1	282 ms
6	493 ms	2	211 ms
7	805 ms	3	312 ms
8	991 ms	4	186 ms
Increment mitja per salt = 242,14 ms			

TOPOLOGIA 2			
Cas PDR al 100% (Sense pèrdua de dades)			
Latència + Paquets rebuts			
Node	Latència MIT	Salts	Increment per salt
2	254 ms	1	254 ms
3	425 ms	2	171 ms
4	544 ms	3	119 ms
5	218 ms	1	218 ms
6	399 ms	2	181 ms
Increment mitja per salt = 188,6 ms			

Amb la diferencia entre els increments mitjos per salt s'obté 53,54 ms de diferencia entre la topologia 1 i la topologia 2. Com sabem que entre les dues topologies l'augment de nodes es de 2, s'obté que l'augment en el increment mitja de latència per salt serà de 26,77 ms, per cada node afegit a la topologia.

## 5. Documentació DEMO-2 Non-Storing

### 5.1. Incís a la DEMO-2

En aquesta Demo s'implementa i posteriorment s'aplica el Testbench dissenyat al protocol RPL treballant en Mode Non-Storing. Com aquest mode de treball ja esta actualment desenvolupat en el sistema OPENWSN, només caldrà utilitzar el sistema implementat en l'anterior DEMO per gestionar el canvi de mode.

### 5.2. Tasques a realitzar en la DEMO

En la DEMO es desenvoluparan les següents tasques:

- **Canvi a nivell de codi de les variables globals de control de mode RPL**
  - openwsn-fw (OpenWSN firmware)  
**RPLMODE = 0**
  - openwsn-sw (OpenWSN software)  
**rplmode = 0**
- **Aplicació del Testbench**
  - Recull de mesures a la Topologia 1
  - Recull de mesures a la Topologia 2

### 5.3. Testbench -- Topologia 1

- **Temps de convergència del protocol al iniciar**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
01:29.01	01:27.08	01:28.18
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
01:24.15	01:19.96	01:22.39

- **Temps de convergència del protocol rere un canvi en la topologia (Desconnexió node)**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
02:30.46	01:43.25	02:05.28
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT

02:12.37	01:39.42	01:57.92
----------	----------	----------

- **Temps de convergència del protocol rere un canvi en la topologia (Reconnexió node)**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:28.20	00:27.80	00:28.06
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:47.91	00:38.48	00:43.83

- **Ample de banda utilitzat amb els missatges del protocol (en el temps de convergència inicial)**

Cas PDR al 100% (Sense pèrdua de dades)			
Missatges DIO			
Node	Numero MAX	Numero MIN	Numero MIT
1	19	18	18
2	19	18	18
3	19	18	18
4	19	18	18
5	19	18	18
6	19	18	18
7	19	18	18
8	19	18	18
<b>TOTAL</b>	152	144	144
Missatges DAO			
Node	Numero MAX	Numero MIN	Numero MIT
1	3	3	3
2	3	3	3
3	3	3	3
4	3	3	3
5	3	3	3
6	3	3	3
7	3	3	3
8	3	3	3
<b>TOTAL</b>	24	24	24
TOTAL Missatges			
Numero MAX	Numero MIN	Numero MIT	
176	168	168	
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
Missatges DIO			
Node	Numero MAX	Numero MIN	Numero MIT
1	19	19	19
2	19	19	19
3	19	19	19
4	19	19	19
5	19	19	19
6	19	19	19
7	19	19	19
8	19	19	19

TOTAL	152	152	152
<b>Missatges DAO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	3	3	3
2	3	3	3
3	3	3	3
4	3	3	3
5	3	3	3
6	3	3	3
7	3	3	3
8	3	3	3
<b>TOTAL</b>	<b>24</b>	<b>24</b>	<b>24</b>
<b>TOTAL Missatges</b>			
Numero MAX	Numero MIN	Numero MIT	
176	176	176	

- **Energia consumida amb els missatges del protocol**  
(en el temps de convergència inicial)

Cas PDR al 100% (Sense pèrdua de dades)		
<b>Missatges DIO</b>		
Consum MAX	Consum MIN	Consum MIT
88,45 mA	83,79 mA	83,79 mA
<b>Missatges DAO</b>		
Consum MAX	Consum MIN	Consum MIT
13,97 mA	13,97 mA	13,97 mA
<b>TOTAL Missatges</b>		
Consum MAX	Consum MIN	Consum MIT
102,42 mA	97,76 mA	97,76 mA
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
<b>Missatges DIO</b>		
Consum MAX	Consum MIN	Consum MIT
88,45 mA	88,45 mA	88,45 mA
<b>Missatges DAO</b>		
Consum MAX	Consum MIN	Consum MIT
13,97 mA	13,97 mA	13,97 mA
<b>TOTAL Missatges</b>		
Consum MAX	Consum MIN	Consum MIT
102,42 mA	102,42 mA	102,42 mA

- **Latències en la comunicació entre node**

Cas PDR al 100% (Sense pèrdua de dades)				
<b>Latència + Paquets rebuts</b>				
Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2	927 ms	288 ms	419 ms	10/10
3	735 ms	504 ms	555 ms	10/10
4	1077 ms	703 ms	788 ms	10/10
5	718 ms	288 ms	364 ms	10/10
6	627 ms	503 ms	552 ms	10/10

7	1126 ms	709 ms	828 ms	10/10
8	1187 ms	979 ms	1071 ms	10/10
Cas PDR al 75% (Amb 25% de pèrdua de dades)				
<b>Latència + Paquets rebuts</b>				
Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2	1881 ms	281 ms	830 ms	10/10
3	3721 ms	756 ms	1820 ms	9/10
4	3510 ms	1194 ms	2034 ms	6/10
5	1940 ms	285 ms	764 ms	9/10
6	3304 ms	512 ms	1568 ms	8/10
7	3118 ms	819 ms	1907 ms	7/10
8	3540 ms	2427 ms	2983 ms	2/10

## 5.4. Testbench -- Topologia 2

- **Temps de convergència del protocol al iniciar**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:41.38	00:35.63	00:39.40
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:58.20	00:42.10	00:49.33

- **Temps de convergència del protocol rere un canvi en la topologia (Desconnexió node)**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
02:25.00	01:54.00	02:29.00
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
02:19.54	02:06.20	02:11.28

- **Temps de convergència del protocol rere un canvi en la topologia (Reconnexió node)**

Cas PDR al 100% (Sense pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:36.11	00:20.52	00:26.98
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Temps MAX	Temps MIN	Temps MIT
00:34.43	00:30.98	00:32.83

- **Ample de banda utilitzat amb els missatges del protocol (en el temps de convergència inicial)**

Cas PDR al 100% (Sense pèrdua de dades)			
<b>Missatges DIO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	13	13	13
2	13	13	13



3	13	13	13
4	13	13	13
5	13	13	13
6	13	13	13
<b>TOTAL</b>	<b>78</b>	<b>78</b>	<b>78</b>
<b>Missatges DAO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	2	2	2
2	2	2	2
3	2	2	2
4	2	2	2
5	2	2	2
6	2	2	2
<b>TOTAL</b>	<b>12</b>	<b>12</b>	<b>12</b>
<b>TOTAL Missatges</b>			
Numero MAX	Numero MIN	Numero MIT	
90	90	90	
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
<b>Missatges DIO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	19	13	17
2	19	13	17
3	19	13	17
4	19	13	17
5	19	13	17
6	19	13	17
<b>TOTAL</b>	<b>114</b>	<b>78</b>	<b>102</b>
<b>Missatges DAO</b>			
Node	Numero MAX	Numero MIN	Numero MIT
1	5	2	3
2	5	2	3
3	5	2	3
4	5	2	3
5	5	2	3
6	5	2	3
<b>TOTAL</b>	<b>30</b>	<b>12</b>	<b>18</b>
<b>TOTAL Missatges</b>			
Numero MAX	Numero MIN	Numero MIT	
144	90	120	

- **Energia consumida amb els missatges del protocol**  
(en el temps de convergència inicial)

Cas PDR al 100% (Sense pèrdua de dades)		
<b>Missatges DIO</b>		
Consum MAX	Consum MIN	Consum MIT
45,39 mA	45,39 mA	45,39 mA
<b>Missatges DAO</b>		
Consum MAX	Consum MIN	Consum MIT
6,98 mA	6,98 mA	6,98 mA
<b>TOTAL Missatges</b>		

Consum MAX	Consum MIN	Consum MIT
52,37 mA	52,37 mA	52,37 mA
Cas PDR al 75% (Amb 25% de pèrdua de dades)		
Missatges DIO		
Consum MAX	Consum MIN	Consum MIT
66,34 mA	45,39 mA	59,35 mA
Missatges DAO		
Consum MAX	Consum MIN	Consum MIT
17,46 mA	6,98 mA	10,47 mA
TOTAL Missatges		
Consum MAX	Consum MIN	Consum MIT
83,8 mA	52,37 mA	69,82 mA

- **Latències en la comunicació entre node**

Cas PDR al 100% (Sense pèrdua de dades)				
Latència + Paquets rebuts				
Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2	428 ms	180 ms	244 ms	10/10
3	689 ms	355 ms	393 ms	10/10
4	1129 ms	477 ms	648 ms	10/10
5	430 ms	176 ms	243 ms	10/10
6	422 ms	327 ms	357 ms	10/10
Cas PDR al 75% (Amb 25% de pèrdua de dades)				
Latència + Paquets rebuts				
Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2	1498 ms	171 ms	385 ms	9/10
3	2507 ms	315 ms	1018 ms	8/10
4	3755 ms	682 ms	2276 ms	6/10
5	3213 ms	191 ms	705 ms	10/10
6	3026 ms	407 ms	1564 ms	9/10

## 5.5. Testbench -- Anàlisi

En aquest apartat es reflexa de forma concisa el comportament observat en cadascuna de les topologies i processos dintre del Testbench, ja que també serà un punt de referència en el estudi comparatiu.

### Topologia 1

Segons el que s'ha observat la xarxa s'ha mantingut estable només amb el PDR al 100% , al prendre les mesures amb el PDR al 75% la xarxa ha tingut desconexions dels nodes més allunyats, degut a que el sistema de neteja veia que les rutes emmagatzemades superaven els 120 segons de vida sense haver rebut cap actualització.

D'aquesta manera l'única solució directa seria establir un valor superior al temps de vida de les rutes en el DODAG root, encara que això comportaria que el temps de desconexió de un node augmentaria notablement.

En les proves de Latència s'ha pogut veure que els paquets coincidents amb l'enviament de missatges DAO tenen una major latència, el cas on es nota més es quan es prenen les mesures amb el 75% de PDR.

## Topologia 2

Segons el que s'ha observat la xarxa, tant al 100% de PDR com al 75% de PDR, s'ha mantingut estable sense flapeig en les connexions entre els nodes.

En les proves de Latència s'ha pogut veure que els paquets coincidents amb l'enviament de missatges DAO tenen una major latència, el cas on es nota més es quan es prenen les mesures amb el 75% de PDR, encara que l'efecte ha sigut molt menor en aquesta topologia.

En el cas de les proves de desconexió i reconexió del node 4 de la topologia, s'ha pogut observar que el temps de desconexió ha sigut molt proper als 120 segons de vida establerts en el sistema de neteja del DODAG root. No s'ha observat efectes molt nocius en aquestes mesures al canviar de 100% de PDR a 75% de PDR.

## Conclusions generals

S'ha pogut observar que el comportament ha sigut notablement diferent en la Topologia 1, ja que al generar pèrdues en els enllaços del 25% la xarxa a començat a convertir-se en inestable. Cal dir que no s'ha variat el valor de vida de les rutes en el DODAG root per tal de tenir les mateixes condicions en les mesures pels dos modes de treball del protocol RPL.

# 6. Estudi comparatiu de rendiment

## 6.1. Incís al estudi comparatiu

En aquest estudi comparatiu s'aconsegueix exposar de forma clara els punts forts i dèbils, a nivell de rendiment, dels modes de treball Storing i Non-Storing del protocol RPL. Mitjançant les mesures obtingudes es muntaran taules i gràfiques comparatives, generant seguidament una exposició raonada del observat. Es farà una anàlisi general al final de cada topologia per tal de tenir un breu resum general. Al final d'aquest apartat s'exposarà un plantejament conclusiu dels comportaments observats per tal d'obtenir finalment un anàlisi comparatiu complet.

## 6.2. Tasques a realitzar

En l'estudi comparatiu es desenvoluparan les següents tasques:

- **Crear taules i gràfiques comparatives de les mesures obtingudes pels dos modes**

Es crearan taules i gràfiques comparatives per a cada objectiu plantejat anteriorment, per tal d'obtenir en les comparacions una idea clara de les diferències de rendiment dels modes del protocol RPL.

- **Realitzar anàlisi comparatiu de les mesures i altres dades obtingudes a cada Topologia**

Mitjançant les taules comparatives i els propis coneixements del funcionament de protocols, enfocant-ho al protocol RPL es plasmarà un anàlisi escrit comparatiu indicant les diferències.

- **Realitzar anàlisi comparatiu final**

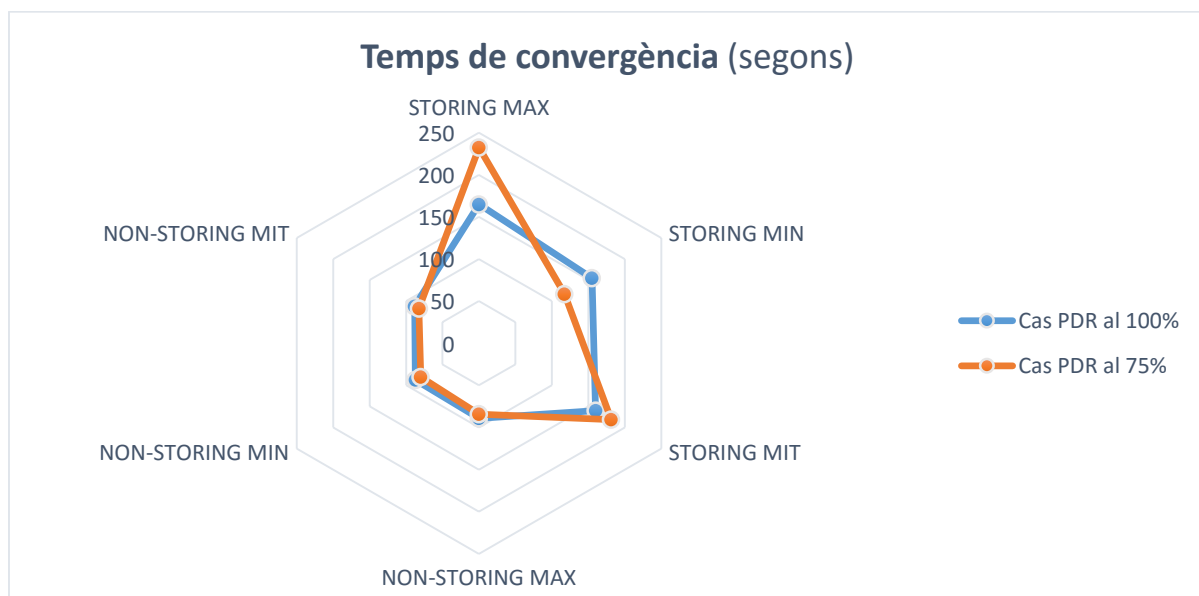
Amb les comparatives fetes a cada Topologia es desenvoluparà l'anàlisi comparatiu final.

### 6.3. Anàlisi comparatiu – Topologia 1

#### Anàlisi comparatiu – Punts específics

- **Temps de convergència del protocol al iniciar**

STORING		NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)			
<b>Temps de convergència</b>			
MAX	02:44.93	01:29.01	MAX
MIN	02:34.85	01:27.08	MIN
MIT	02:39.86	01:28.18	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
<b>Temps de convergència</b>			
MAX	03:52.49	01:24.15	MAX
MIN	01:57.03	01:19.96	MIN
MIT	03:01.07	01:22.39	MIT

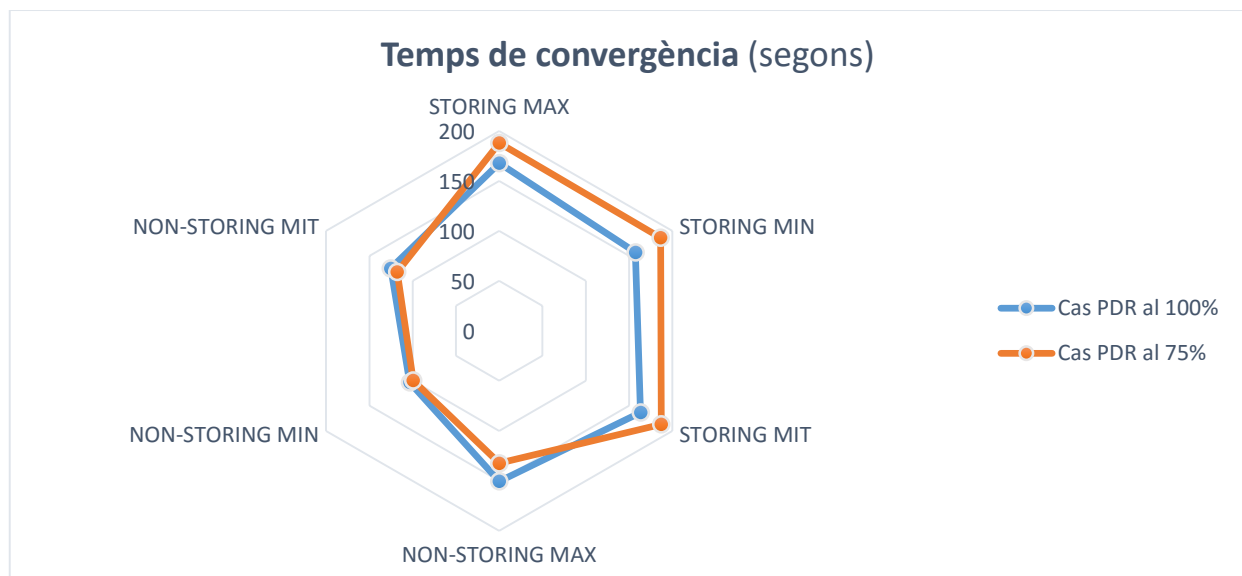


Es pot observar en els resultats que el temps de convergència per tal d'iniciar la xarxa en el mode Storing arriba a ser aproximadament el doble que en el mode Non-Storing. Això ens fa veure que el protocol RPL es molt més lent al iniciar en aquesta topologia amb el mode Storing.

Si es compara el cas sense pèrdues del cas amb pèrdues es pot veure que el mode mes afectat es el mode Storing, degut a que te un creixement de mitja del 13% sobre el temps mitja.

- **Temps de convergència del protocol rere un canvi en la topologia (Desconnexió node)**

STORING		NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)			
<b>Temps de convergència</b>			
MAX	02:47.79	02:30.46	MAX
MIN	02:37.41	01:43.25	MIN
MIT	02:43.39	02:05.28	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
<b>Temps de convergència</b>			
MAX	03:07.94	02:12.37	MAX
MIN	03:06.48	01:39.42	MIN
MIT	03:07.14	01:57.92	MIT



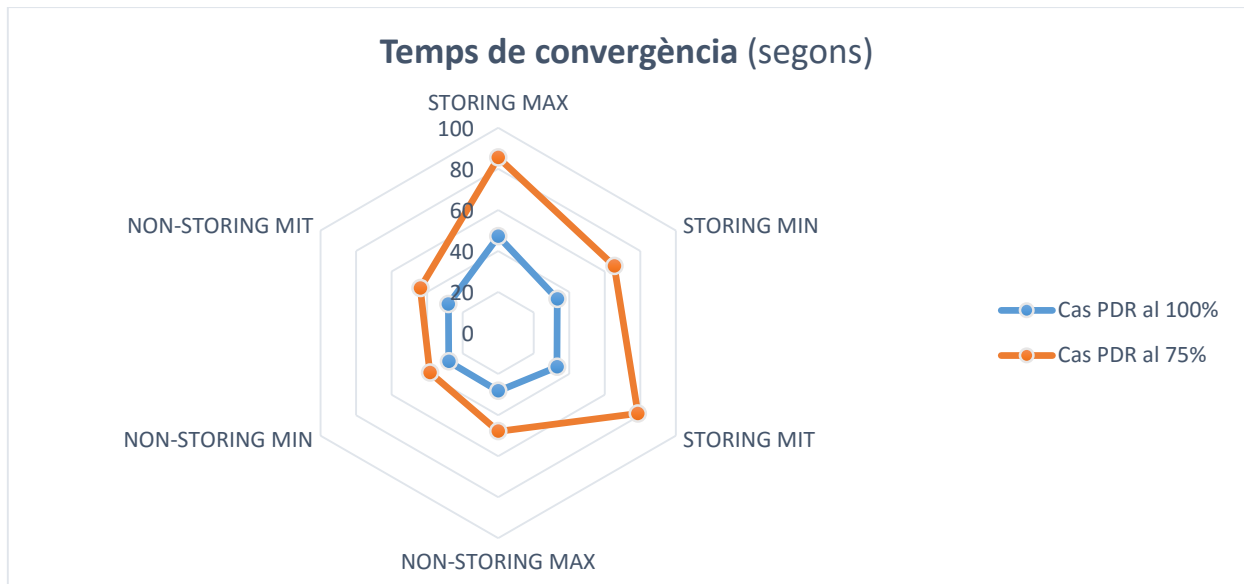
Es pot observar en els resultats que el temps en la desconnexió del node de la xarxa, en el mode Storing es major que en el mode Non-Storing.

Si es compara el cas sense pèrdues amb el cas amb pèrdues es pot veure que el mode mes afectat es el mode Storing, ja que augmenta el seu temps de convergència un 14%. Per tant augmenta més la diferència entre el mode Storing i Non-Storing, donant uns resultats encara millors en mode Non-Storing.

- **Temps de convergència del protocol rere un canvi en la topologia (Reconnexió node)**

STORING		NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)			
<b>Temps de convergència</b>			
MAX	00:41.07	00:28.20	MAX
MIN	00:33.26	00:27.80	MIN

MIT	00:33.02		00:28.06	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)				
<b>Temps de convergència</b>				
MAX	01:25.44		00:47.91	MAX
MIN	01:05.29		00:38.48	MIN
MIT	01:18.49		00:43.83	MIT



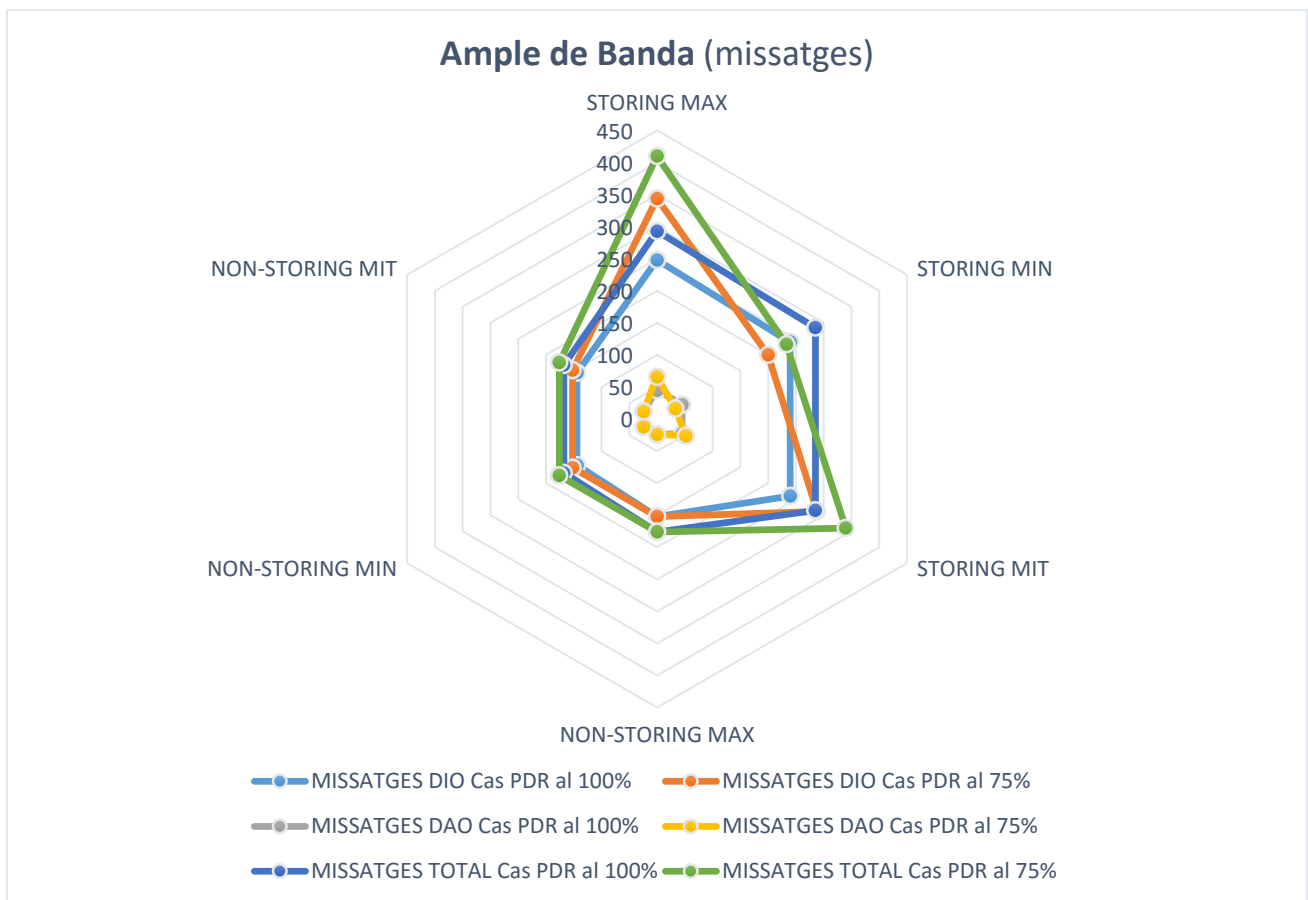
Es pot observar en els resultats que el temps en la reconexió del node a la xarxa, en el mode Storing es aproximadament de mitja el mateix que en el mode Non-Storing.

Si es compara el cas sense pèrdues del cas amb pèrdues es pot veure que el mode mes afectat es el mode Storing, ja que acaba trigant mes el mode Non-Storing.

- **Ample de banda utilitzat amb els missatges del protocol (en el temps de convergència inicial)**

STORING			NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)				
<b>Missatges DIO Total</b>				
MAX	248		152	MAX
MIN	240		144	MIN
MIT	240		144	MIT
<b>Missatges DAO Total</b>				
MAX	45		24	MAX
MIN	45		24	MIN
MIT	45		24	MIT
<b>Missatges Totals</b>				
MAX	293		176	MAX
MIN	285		168	MIN
MIT	285		168	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)				
<b>Missatges DIO Total</b>				

MAX	344		152	MAX
MIN	200		152	MIN
MIT	288		152	MIT
<b>Missatges DAO Total</b>				
MAX	66		24	MAX
MIN	33		24	MIN
MIT	52		24	MIT
<b>Missatges Totals</b>				
MAX	410		176	MAX
MIN	233		176	MIN
MIT	340		176	MIT



Es pot observar que en els resultats de ample de banda:

#### ■ Missatges DIO

En el cas sense pèrdues el mode Storing el consum d'ample de banda es superior al mode Non-Storing. En el cas amb pèrdues la diferència en el consum d'ample de banda arriba a ser prop del doble en el mode Storing que en el mode Non-Storing.

#### ■ Missatges DAO

En el cas sense pèrdues el mode Storing el consum d'ample de banda es superior al mode Non-Storing. En el cas amb pèrdues la diferència augmenta encara mes entre el mode Storing i el mode Non-Storing.

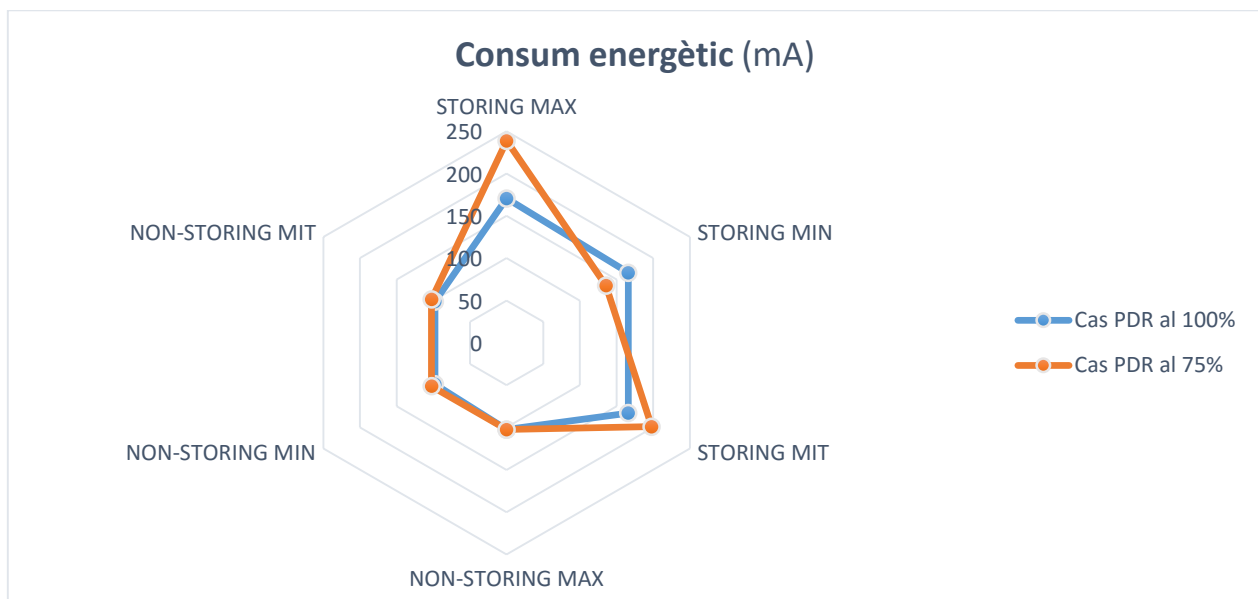
#### ■ Total Missatges

En el cas sense pèrdues el mode Storing el consum d'ample de banda es superior al mode Non-Storing. En el cas amb pèrdues la diferencia en el consum d'ample de banda arriba a ser prop del doble en el mode Storing que en el mode Non-Storing.

Així es pot dir que l'ample de banda utilitzat pel protocol RPL ha sigut, en termes generals, prop del doble en el mode Storing que en el mode Non-Storing.

- **Energia consumida amb els missatges del protocol (en el temps de convergència inicial)**

STORING		NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)			
<b>Consum energètic (Missatges Totals)</b>			
MAX	170,5 mA	102,42 mA	MAX
MIN	165,85 mA	97,76 mA	MIN
MIT	165,85 mA	97,76 mA	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
<b>Consum energètic (Missatges Totals)</b>			
MAX	238,58 mA	102,42 mA	MAX
MIN	135,58 mA	102,42 mA	MIN
MIT	197,85 mA	102,42 mA	MIT



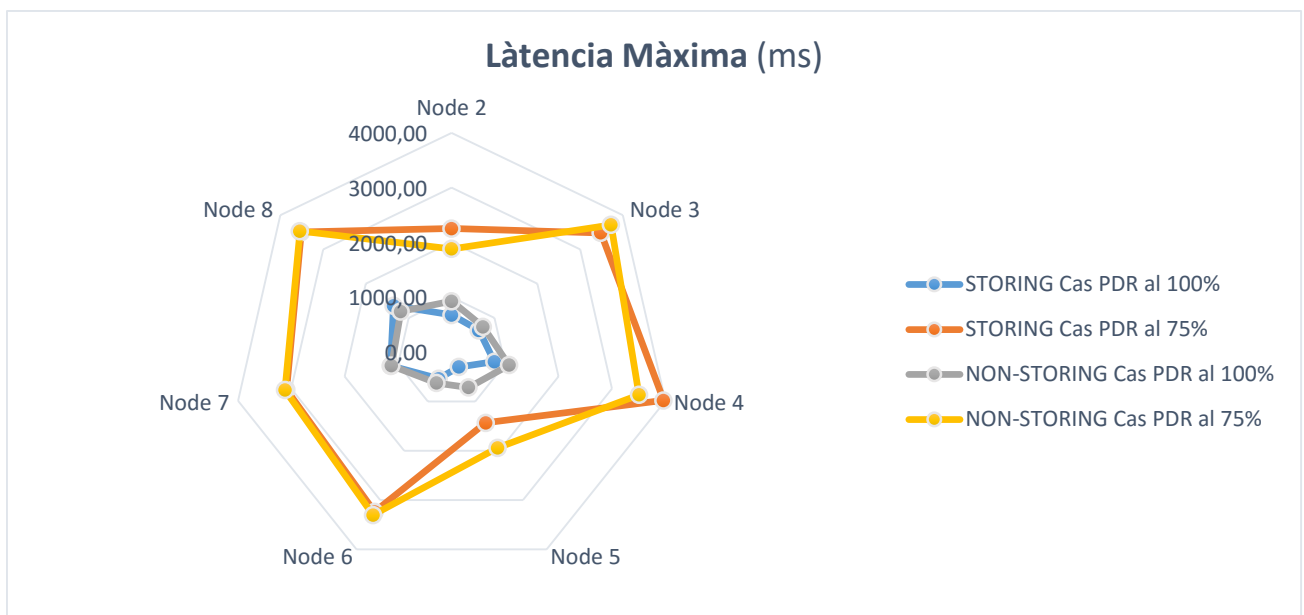
Tal com calia esperar, degut a que el càlcul del consum energètic va directament relacionat al volum de missatges enviats, es veu que el consum energètic es prop el doble en el mode Storing que en el mode Non-Storing.

De la mateixa forma al comparar el cas sense pèrdues amb el cas amb pèrdues s'observa que el mode més afectat ha sigut el mode Storing, degut a que té un augment mitjà del 19% de consum, donant encara més consum en el mode Storing que en el mode Non-Storing.

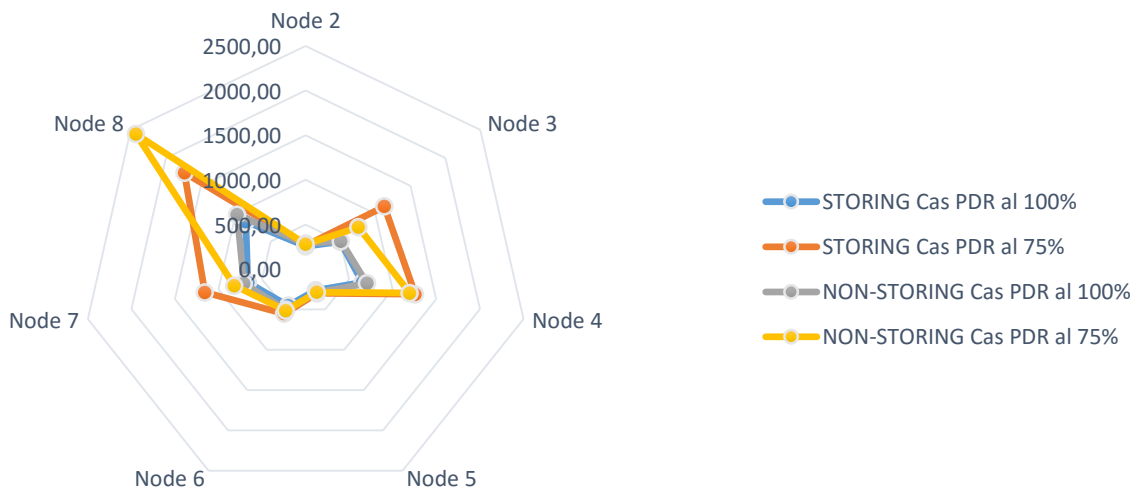
- **Latències en la comunicació entre node**



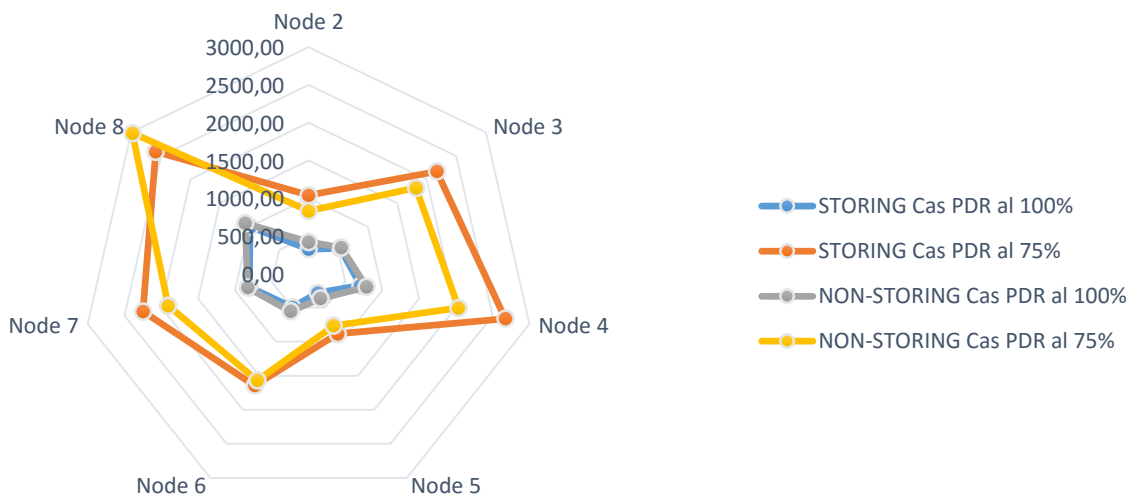
STORING					NON-STORING			
Cas PDR al 100% (Sense pèrdua de dades)								
Latència + Paquets rebuts								
Latència MAX	Latència MIN	Latència MIT	Paquets rebuts	Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
675 ms	257 ms	325 ms	10/10	2	927 ms	288 ms	419 ms	10/10
635 ms	496 ms	536 ms	10/10	3	735 ms	504 ms	555 ms	10/10
802 ms	653 ms	704 ms	10/10	4	1077 ms	703 ms	788 ms	10/10
304 ms	260 ms	282 ms	10/10	5	718 ms	288 ms	364 ms	10/10
541 ms	451 ms	493 ms	10/10	6	627 ms	503 ms	552 ms	10/10
1139 ms	662 ms	805 ms	10/10	7	1126 ms	709 ms	828 ms	10/10
1354 ms	857 ms	991 ms	10/10	8	1187 ms	979 ms	1071 ms	10/10
Cas PDR al 75% (Amb 25% de pèrdua de dades)								
Latència + Paquets rebuts								
Latència MAX	Latència MIN	Latència MIT	Paquets rebuts	Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
2252 ms	264 ms	1038 ms	10/10	2	1881 ms	281 ms	830 ms	10/10
3485 ms	1128 ms	2171 ms	7/10	3	3721 ms	756 ms	1820 ms	9/10
3967 ms	1256 ms	2671 ms	8/10	4	3510 ms	1194 ms	2034 ms	6/10
1434 ms	298 ms	884 ms	9/10	5	1940 ms	285 ms	764 ms	9/10
3238 ms	551 ms	1643 ms	9/10	6	3304 ms	512 ms	1568 ms	8/10
3092 ms	1156 ms	2248 ms	6/10	7	3118 ms	819 ms	1907 ms	7/10
3511 ms	1734 ms	2590 ms	5/10	8	3540 ms	2427 ms	2983 ms	2/10



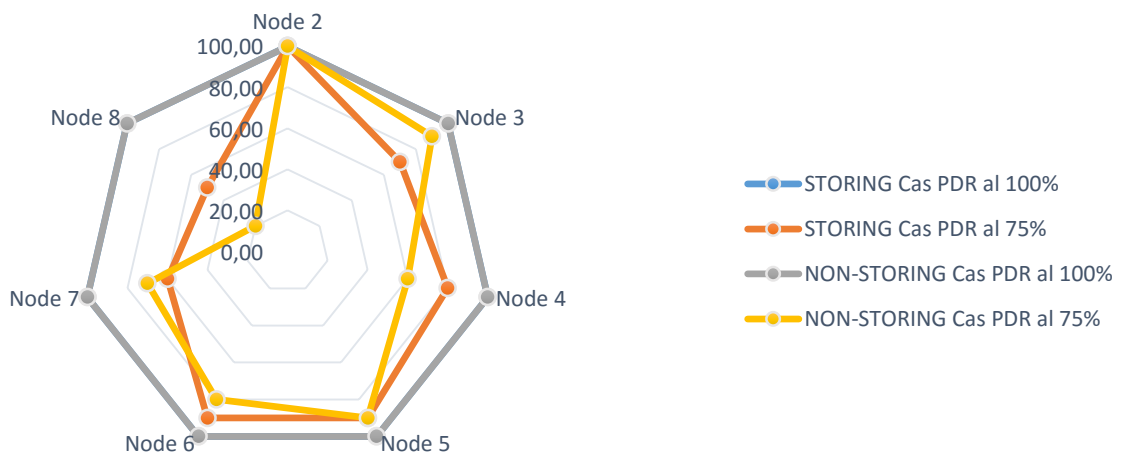
### Làtencia M nima (ms)



### L tencia Mitja (ms)



### Paquets Rebuts (%)



Es pot observar que en els resultats de latència:

■ **Latència Màxima**

En el cas sense pèrdues el mode Storing dona de mitja resultats una mica millors que el mode Non-Storing. En el cas amb pèrdues els dos modes arriben a tenir de mitja uns resultats propers.

■ **Latència Mínima**

En el cas sense pèrdues els dos modes donen resultats de mitja propers. En el cas amb pèrdues el mode Non-Storing dona de mitja millors resultats que el mode Storing.

■ **Latència Mitja**

En el cas sense pèrdues els dos modes donen resultats propers. En el cas amb pèrdues el mode Non-Storing dona de mitja millors resultats que el mode Storing.

■ **Pèrdua de Paquets**

En el cas sense pèrdues els dos modes es comporten igual servint tots els paquets sense problemes. En el cas amb pèrdues el mode Storing dona de mitja millors resultats que el mode Non-Storing.

En termes generals després de fixar-se en la topologia, i els salts que es requereixen per arribar a cada node. Es pot veure que pel cas sense pèrdues els dos modes donen resultats propers de Latència, però pel cas amb pèrdues el mode Storing dona Latències majors que el mode Non-Storing. En quant a la pèrdua de paquets es pot veure que el mode Storing garanteix millor l'arribada de paquets en cas de pèrdues, ja que en el cas sense pèrdues els dos modes donen els mateixos resultats.

#### Anàlisi comparatiu – Conclusiu de la topologia

Tal com s'ha pogut veure a l'anàlisi comparatiu fet en aquesta topologia es poden obtenir els següents punts resolutius:

- **El Temps de Convergència en el mode Storing es significativament major que en el mode Non-Storing**
- **L'Ample de Banda utilitzat en el mode Storing ha sigut prop del doble del mode Non-Storing**
- **El Consum Energètic utilitzat en el mode Storing ha sigut prop del doble del mode Non-Storing**
- **La Latència en mode Storing ha sigut propera al mode Non-Storing en el cas sense pèrdues, i pitjor en el cas amb pèrdues**
- **La Pèrdua de Paquets de paquets ha sigut menor en el mode Storing**

També cal tenir en compte el exposat als anàlisis finals de cada bloc de mesures, ja que es pot observar que el mode Non-Storing es torna inestable en el cas amb pèrdues. Això es deu a que necessita un temps major de vida en les rutes emmagatzemades en la Taula de Routing del DODAG root, perquè el temps necessari per actualitzar-les es molt major, ja que els missatges DAO donen mes salts que en el cas del mode Storing.

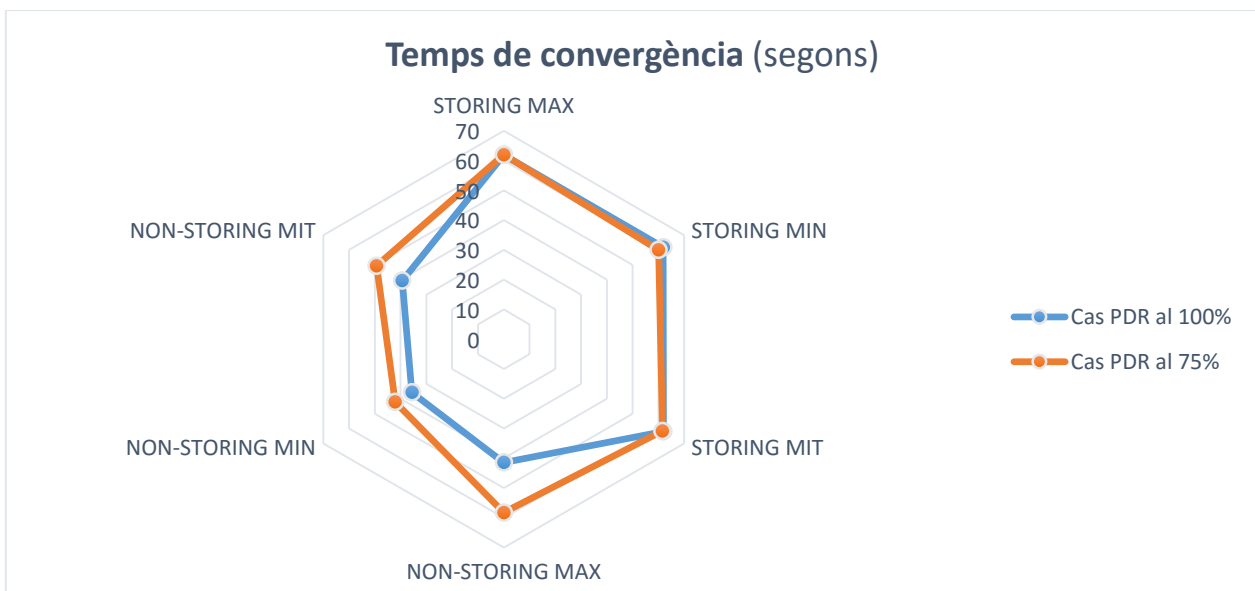
Així es pot arribar a la conclusió de que el mode Non-Storing dona en termes generals un rendiment millor en aquest tipus de topologia, encara que el mode Storing ha demostrat donar un nivell d'estabilitat i fiabilitat superior en la comunicació.

## 6.4. Anàlisi comparatiu – Topologia 2

### Anàlisi comparatiu – Punts específics

- **Temps de convergència del protocol al iniciar**

STORING		NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)			
<b>Temps de convergència</b>			
MAX	01:01.94	00:41.38	MAX
MIN	01:01.84	00:35.63	MIN
MIT	01:01.89	00:39.40	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
<b>Temps de convergència</b>			
MAX	01:01.96	00:58.20	MAX
MIN	01:00.06	00:42.10	MIN
MIT	01:01.56	00:49.33	MIT



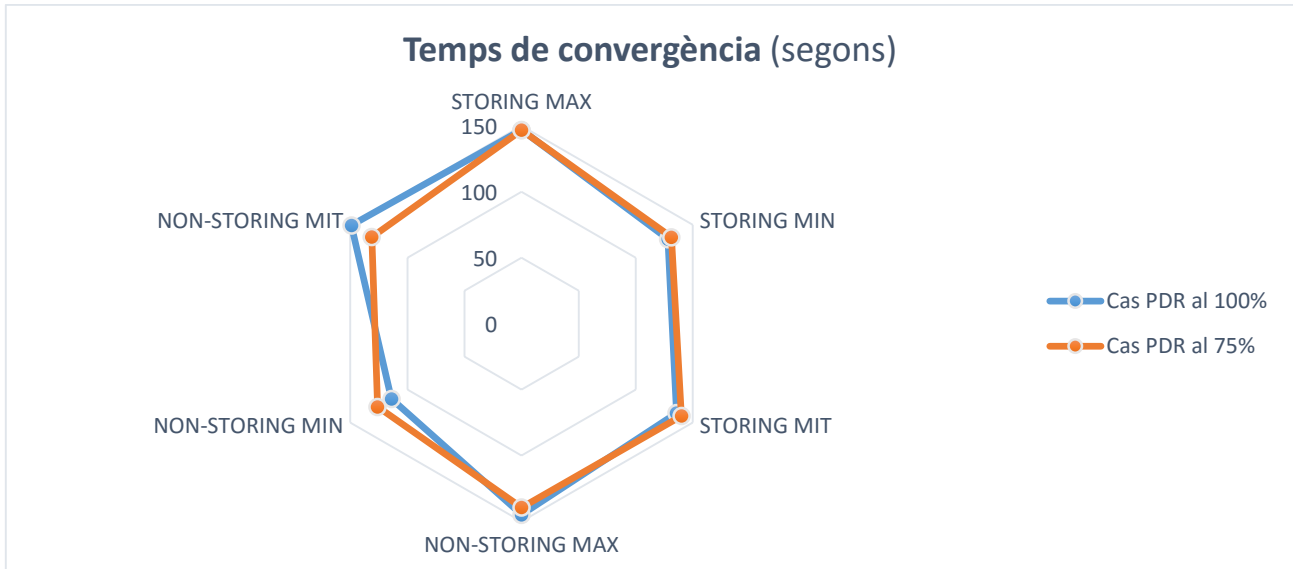
Es pot observar que els resultats de temps de convergència per tal d'iniciar la xarxa en el mode Storing son superiors als del mode Non-Storing.

Si es compara el cas sense pèrdues del cas amb pèrdues es pot veure que el mode mes afectat es el mode Non-Storing, ja que ha augmentat un 25%. Encara i això el mode Storing segueix tenint temps de convergència superiors..

- **Temps de convergència del protocol rere un canvi en la topologia (Desconnexió node)**

STORING		NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)			
<b>Temps de convergència</b>			
MAX	02:26.85	02:25.00	MAX

MIN	02:08.27		01:54.00	MIN
MIT	02:15.71		02:29.00	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)				
<b>Temps de convergència</b>				
MAX	02:26.75		02:19.54	MAX
MIN	02:11.20		02:06.20	MIN
MIT	02:20.06		02:11.28	MIT

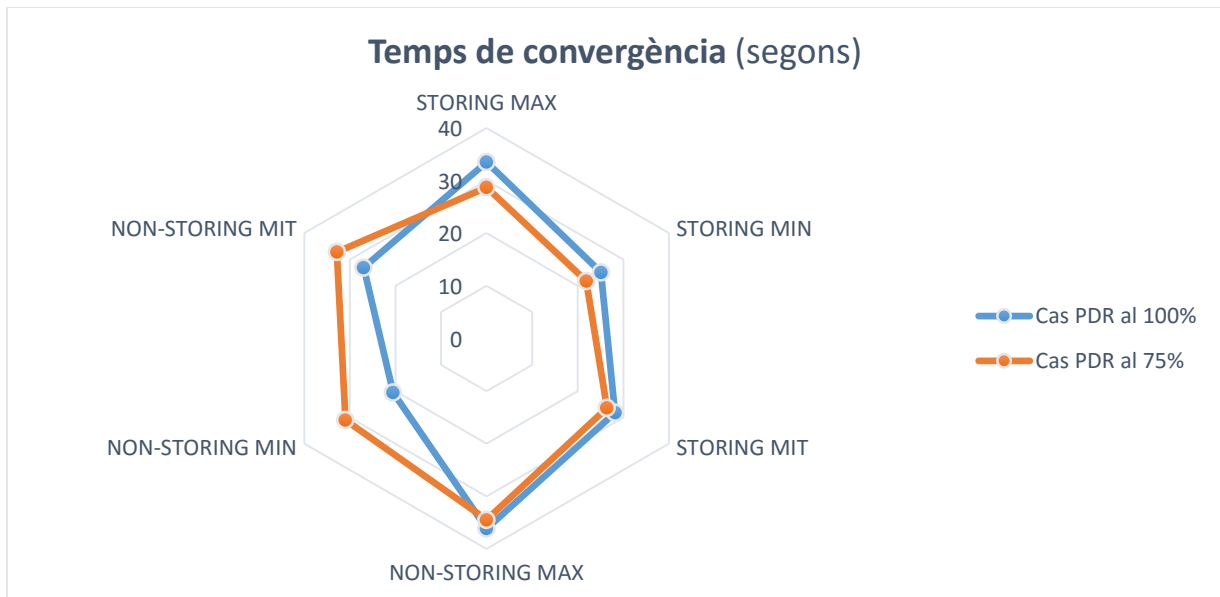


Es pot observar que els resultats en el temps de desconexió del node en els dos modes son propers, encara que de mitja el mode Non-Storing es una mica superior. Això es degut a que depenen del temps de vida que s'estableix en el DODAG root de 120 segons.

Si es compara el cas sense pèrdues del cas amb pèrdues es pot veure que els dos modes han sofert una molt petita variació mantenint resultats propers, i en aquest cas el mode Storing es una mica superior de mitja.

- **Temps de convergència del protocol rere un canvi en la topologia (Reconnexió node)**

STORING			NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)				
<b>Temps de convergència</b>				
MAX	00:33.50		00:36.11	MAX
MIN	00:25.10		00:20.52	MIN
MIT	00:28.19		00:26.98	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)				
<b>Temps de convergència</b>				
MAX	00:28.72		00:34.43	MAX
MIN	00:21.86		00:30.98	MIN
MIT	00:26.39		00:32.83	MIT



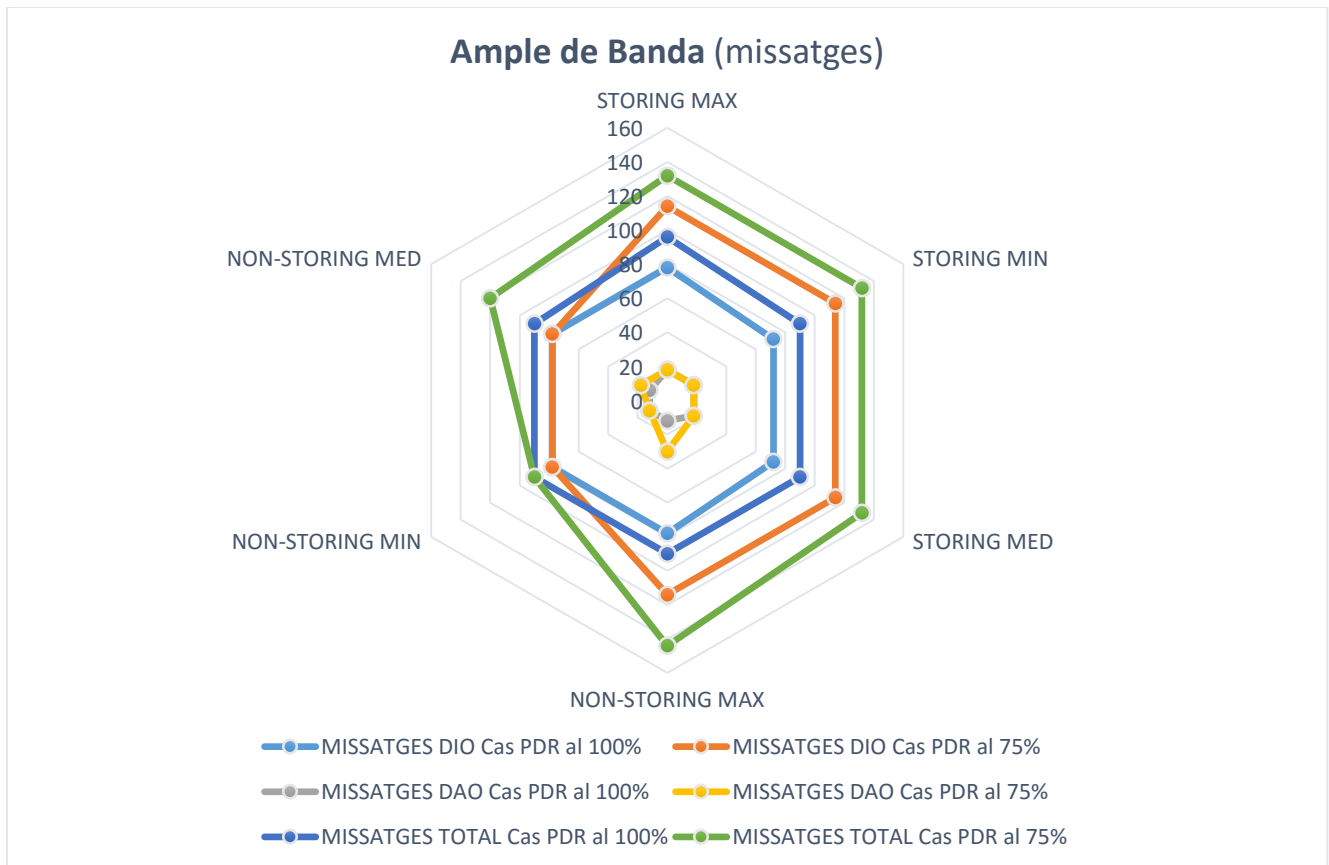
Es pot observar en els resultats que el temps en la reconexió del node a la xarxa, en el mode Storing es aproximadament de mitja el mateix que en el mode Non-Storing.

Si es compara el cas sense pèrdues del cas amb pèrdues es pot veure que el mode mes afectat es el mode Non-Storing, ja que acaba trigant mes que en el mode Storing.

- **Ample de banda utilitzat amb els missatges del protocol (en el temps de convergència inicial)**

STORING		NON-STORING	
Cas PDR al 100% (Sense pèrdua de dades)			
<b>Missatges DIO Total</b>			
MAX	78	78	MAX
MIN	72	78	MIN
MIT	72	78	MIT
<b>Missatges DAO Total</b>			
MAX	18	12	MAX
MIN	18	12	MIN
MIT	18	12	MIT
<b>Missatges Totals</b>			
MAX	96	90	MAX
MIN	90	90	MIN
MIT	90	90	MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)			
<b>Missatges DIO Total</b>			
MAX	114	114	MAX
MIN	114	78	MIN
MIT	114	78	MIT
<b>Missatges DAO Total</b>			
MAX	18	30	MAX
MIN	18	12	MIN
MIT	18	18	MIT

Missatges Totals				
MAX	132		144	MAX
MIN	132		90	MIN
MIT	132		120	MIT



Es pot observar que en els resultats de ample de banda:

■ **Missatges DIO**

En el cas sense pèrdues els dos modes donen consums propers d'ample de banda, siguent una mica millor els resultats del mode Non-Storing. En el cas amb pèrdues el consum d'ample de banda en el mode Storing augmenta, fent que consumeixi més ample de banda que al mode Non-Storing

■ **Missatges DAO**

En el cas sense pèrdues els dos modes donen consums propers d'ample de banda. En el cas amb pèrdues el consum d'ample de banda en el mode Storing augmenta, fent que consumeixi més ample de banda que al mode Non-Storing

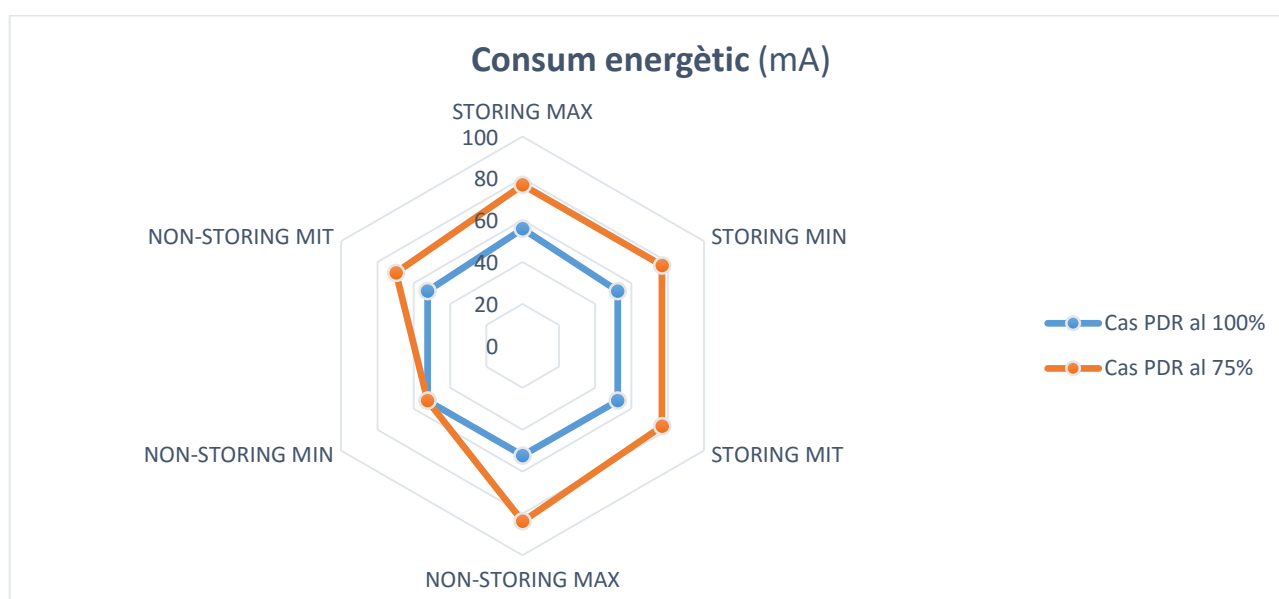
■ **Total Missatges**

En el cas sense pèrdues els dos modes donen consums propers d'ample de banda. En el cas amb pèrdues el consum d'ample augmenta en els dos modes, fent que el mode Storing tingui un consum mitja superior.

Així es pot dir que l'ample de banda utilitzat pel protocol RPL ha sigut en el cas sense pèrdues igual en els dos modes. En el cas amb pèrdues el mode Non-Storing ha tingut un consum d'ample de banda de mitja menor.

- **Energia consumida amb els missatges del protocol (en el temps de convergència inicial)**

STORING			NON-STORING		
Cas PDR al 100% (Sense pèrdua de dades)					
<b>Consum energètic (Missatges Totals)</b>					
MAX	55,86 mA		52,37 mA		MAX
MIN	52,37 mA		52,37 mA		MIN
MIT	52,37 mA		52,37 mA		MIT
Cas PDR al 75% (Amb 25% de pèrdua de dades)					
<b>Consum energètic (Missatges Totals)</b>					
MAX	76,81 mA		83,8 mA		MAX
MIN	76,81 mA		52,37 mA		MIN
MIT	76,81 mA		69,82 mA		MIT



Tal com calia esperar, degut a que el càlcul del consum energètic va directament relacionat al volum de missatges enviats, es veu que el consum energètic es aproximadament igual pels dos modes en el cas sense pèrdues.

De la mateixa forma al comparar el cas sense pèrdues amb el cas amb pèrdues s'observa que el mode més afectat ha sigut el mode Storing, passant a tenir un consum major de mitja.

- **Latències en la comunicació entre node**

STORING					NON-STORING			
Cas PDR al 100% (Sense pèrdua de dades)								
<b>Latència + Paquets rebuts</b>								
Latència MAX	Latència MIN	Latència MIT	Paquets rebuts	Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
632 ms	193 ms	254 ms	10/10	2	428 ms	180 ms	244 ms	10/10
836 ms	319 ms	425 ms	10/10	3	689 ms	355 ms	393 ms	10/10



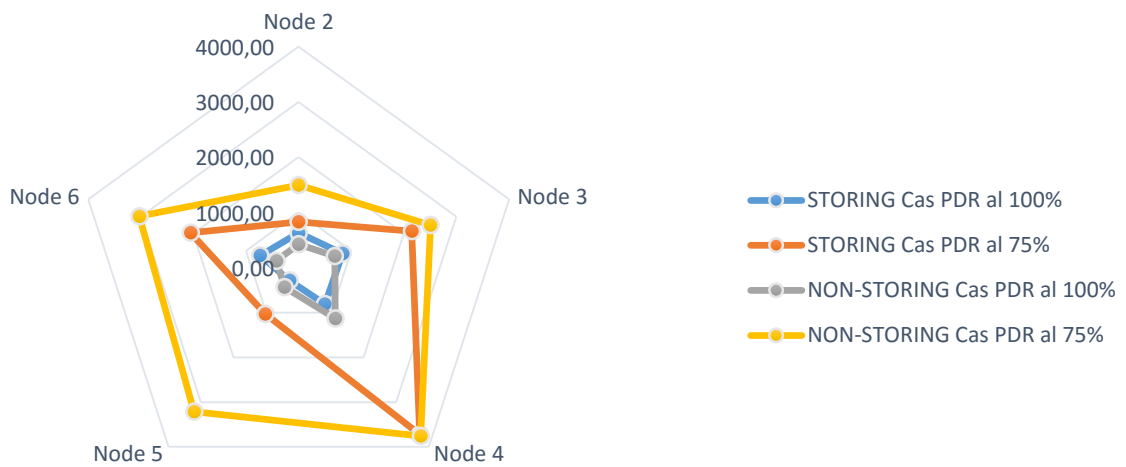
808 ms	470 ms	544 ms	10/10	4	1129 ms	477 ms	648 ms	10/10
280 ms	170 ms	218 ms	10/10	5	430 ms	176 ms	243 ms	10/10
730 ms	321 ms	399 ms	10/10	6	422 ms	327 ms	357 ms	10/10

Cas PDR al 75% (Amb 25% de pèrdua de dades)

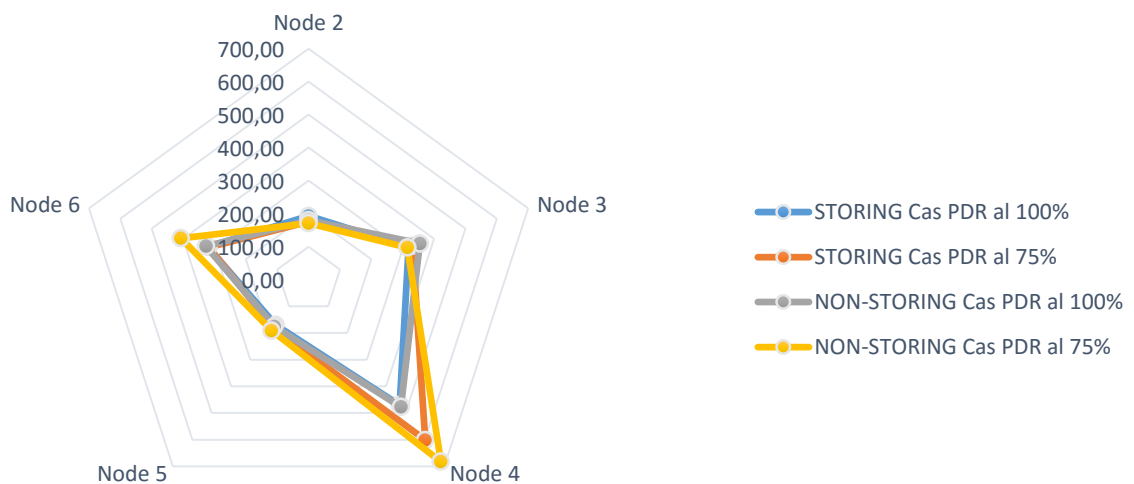
### Latència + Paquets rebuts

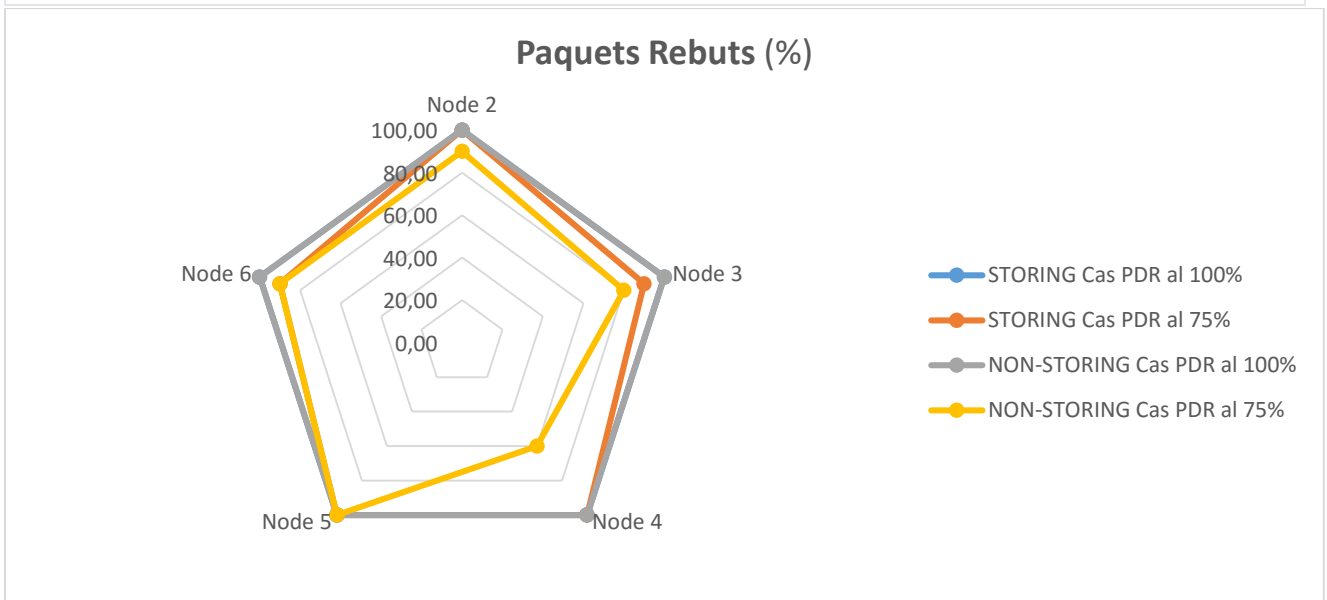
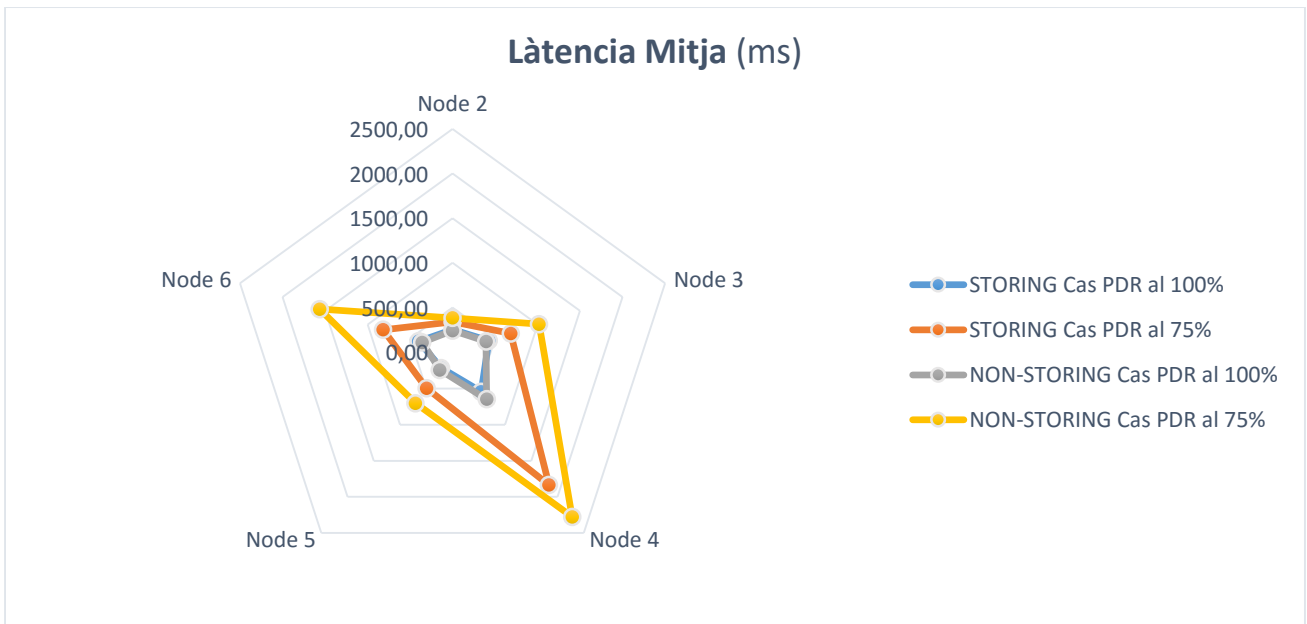
Latència MAX	Latència MIN	Latència MIT	Paquets rebuts	Node	Latència MAX	Latència MIN	Latència MIT	Paquets rebuts
833 ms	176 ms	342 ms	10/10	2	1498 ms	171 ms	385 ms	9/10
2150 ms	331 ms	683 ms	9/10	3	2507 ms	315 ms	1018 ms	8/10
3724 ms	601 ms	1836 ms	10/10	4	3755 ms	682 ms	2276 ms	6/10
1028 ms	181 ms	496 ms	10/10	5	3213 ms	191 ms	705 ms	10/10
2053 ms	318 ms	816 ms	9/10	6	3026 ms	407 ms	1564 ms	9/10

### Làtència Màxima (ms)



### Làtència Mínima (ms)





Es pot observar que en els resultats de latència:

■ **Latència Màxima**

En el cas sense pèrdues els dos modes donen uns resultats de mitja propers. En el cas amb pèrdues el mode Storing dona millors resultats que el mode Non-Storing.

■ **Latència Mínima**

En el cas sense pèrdues els dos modes donen uns resultats propers. En el cas amb pèrdues el mode Storing dona uns resultats una mica millors que el mode Non-Storing, però molt propers de mitja.

■ **Latència Mitja**

En el cas sense pèrdues els dos modes donen uns resultats propers. En el cas amb pèrdues el mode Storing dona millors resultats que el mode Non-Storing.

■ **Pèrdua de Paquets**

En el cas sense pèrdues els dos modes es comporten igual servint tots els paquets sense problemes. En el cas amb pèrdues el mode Storing dona de mitja millors resultats que el mode Non-Storing.

En termes generals després de fixar-se en la topologia, i els salts que es requereixen per arribar a cada node. Es pot veure que el mode Storing pel cas sense pèrdues té resultats molt propers de Latència al mode Non-Storing, inclús una mica millors. Pel cas amb pèrdues el mode Storing dona millors resultats que el mode Non-Storing. En quant a la pèrdua de paquets es pot veure que el mode Storing garanteix millor l'arribada de paquets en cas de pèrdues, ja que en el cas sense pèrdues els dos modes donen els mateixos resultats.

### Anàlisi comparatiu – Conclusiu de la topologia

Tal com s'ha pogut veure a l'anàlisi comparatiu fet en aquesta topologia es poden obtenir els següents punts resolutius:

- **El Temps de Convergència en els dos modes es aproximadament igual en el cas sense pèrdues, en el cas amb pèrdues ha sigut major en el mode Storing.**
- **L'Ample de Banda utilitzat ha sigut en els dos modes aproximadament idèntic en el cas sense pèrdues, en el cas amb pèrdues ha sigut una mica major en el mode Storing.**
- **El Consum Energètic utilitzat ha sigut en els dos modes aproximadament idèntic, encara que en el cas amb pèrdues ha sigut una mica major en el mode Storing.**
- **La Latència en el mode Storing ha sigut propera al mode Non-Storing en el cas sense pèrdues, en el cas amb pèrdues els resultats han estat millor en el mode Storing.**
- **La Pèrdua de Paquets de paquets ha sigut menor en el cas del mode Storing**

Després de llegir l'exposat als anàlisis finals de cada bloc de mesures es troba que els dos modes han restat estables en el cas amb pèrdues, fent així que només es puguin arribar a conclusions amb les mesures preses.

Així es pot arribar a la conclusió de que els dos modes en termes generals han donat un rendiment similar en aquest tipus de topologia, encara que el mode Storing ha demostrat donar un nivell d'estabilitat i fiabilitat superior en la comunicació.

### 6.3. Anàlisi comparatiu – Conclusions


Al posar en comú els anàlisis obtinguts per a cada topologia, es pot arribar als següents punts conclusius reflectits en la figura 28.


Topologia gran (topologia 1)	Temps de convergència		Ample de Banda		Consum Energètic		Latència		Pèrdua de paquets		Estabilitat	
	100% PDR	75% PDR	100% PDR	75% PDR	100% PDR	75% PDR	100% PDR	75% PDR	100% PDR	75% PDR	100% PDR	75% PDR
<b>STORING</b>	∇	∇	∇	∇	∇	∇	=	∇	=	∧	=	∧
<b>NON-STORING</b>	∧	∧	∧	∧	∧	∧	=	∧	=	∇	=	∇

Topologia petita (topologia 2)	Temps de convergència		Ample de Banda		Consum Energètic		Latència		Pèrdua de paquets		Estabilitat	
	100% PDR	75% PDR	100% PDR	75% PDR	100% PDR	75% PDR	100% PDR	75% PDR	100% PDR	75% PDR	100% PDR	75% PDR
<b>STORING</b>	=	∇	=	∇	=	∇	=	∧	=	∧	=	=
<b>NON-STORING</b>	=	∧	=	∧	=	∧	=	∇	=	∇	=	=

 Resultats Millors

 Resultats Pitjors


 Resultats Equivalents

Figura 28 - Conclusions estudi comparatiu

Després d'aquests punts conclusius cal explicar certes característiques de cada mode que cal tenir en compte, més enllà del obtingut en les mesures.

En el cas del mode Storing al deixar de treballar amb Source-Routing, les dades de Routing ja no estan incloses en els paquets i passen a ocupar la memòria dels Nodes. Això sota un anàlisi estricte faria veure que el mode Storing encara que necessita més missatges per establir i mantindrà la xarxa, aquests paquets i la resta de paquets no vinculats al protocol de Routing son mes petits que els generats en el mode Non-Storing. Per tant, els missatges consumiran per sota del valor mig estipulat, fent així que el mode Storing no estigui tant distant del mode Non-Storing en quant a consum d'ample de banda i energia. En contrapartida al utilitzar la memòria dels Nodes pel Routing, la Taula de Routing tindrà un límit de rutes a emmagatzemar, limitant així la mida màxima de la xarxa [23].

En el cas del mode Non-Storing al treballar amb Source-Routing, com ja s'ha explicat, fa que les dades de Routing tinguin que estar contingudes en els paquets enviats. Fent així que s'augmenti l'ample de banda, i el consum energètic utilitzat tant pel tràfic propi del protocol de Routing com pel tràfic genèric de la xarxa. Coneixent les limitacions de buffer dels equips MOTE, esdevé en una limitació en la mida màxima d'aquesta informació de Routing, limitant també les dades màximes contingudes en cada paquet que no tinguin funció associada al Routing.

Amb tota aquesta informació es pot arribar a establir els següents punts, que serviran d'indicador per a escollir el mode del protocol RPL que s'ajustarà millor a la xarxa que es vol muntar:

- Mida de la Xarxa a implementar
- Característiques del Hardware dels Nodes (Memòria, CPU i Energia) que conformen la xarxa
- Necessitat d'estabilitat en la xarxa i temps de reacció davant la desconexió de nodes
- Necessitat de fiabilitat en les comunicacions davant la pèrdua de paquets

Finalment, rere les conclusions que s'han arribat cal indicar que amb possibles evolucions en el sistema OPENWSN es podrà millorar el consum d'ample de banda i energètic. Així es poden plantejar com a treballs futurs de millora els següents punts:

- Fragmentació de paquets. Amb aquesta evolució s'estalviaria bastant ample de banda ja que no caldria enviar repetidament la base del missatge DAO per redistribuir la informació de Routing.
- Control en el enviament de missatges DIO per tal que apliqui el algorisme Trickle, ja que actualment s'estan enviant aplicant una periodicitat fixe. Això faria que el volum de paquets descendís notablement al establir-se la topologia.
- Control en el enviament de missatges DAO, seguint més de prop les característiques exposades en el RFC6550. Que l'activació de l'enviament dels missatges DAO sigui actiu enlloc de només periòdic.

## 7. Conclusions

Les xarxes IoT, i en particular les WSN LLN, son tecnologia d'un futur proper. Aquesta tecnologia formarà part de la vida quotidiana, fent que la informació que ens envolta sigui digitalitzada i utilitzada. Així generant un mon intel·ligent, capaç de ser conscient del entorn, i mes proper a les necessitats humanes. Entre els diversos sistemes que estan sorgint per tal de dur a terme aquest nou paradigma es pot trobar el sistema OPENWSN, un sistema basat en el enfocament open-source que avança cap aquest futur proper gracies a l'ajuda del treball comú de molts.

Amb aquest treball s'ha posat un petit gra de sorra en aquest projecte comú, aportant una nova funcionalitat i algunes correccions al sistema OPENWSN. Es pot dir amb certesa que els objectius plantejats al principi del TFM s'han assolit, arribant a obtenir els productes plantejats al inici del treball. Encara que amb més temps s'hauria aconseguit avançar més en el desenvolupament, per tal de treure algunes limitacions importants, que mes endavant es plantejaran com a treballs a desenvolupar. També l'estudi comparatiu ha sofert aquesta limitació temporal, on s'han tingut que fer concessions. Fent servir valors mitjos pels càlculs, per tal d'arribar a un resultat final esclaridor en el temps pactat.

La planificació inicial va tenir errors en el seu plantejament, degut principalment al desconeixement tant del caire del projecte com de la temàtica associada. Hauria calgut tenir molt mes temps per fer un estudi previ del projecte, i prendre una visió global amb tots els factors per a que la planificació hagués sigut mes precisa. En el desenvolupament del TFM a traves del propi seguiment es van poder plantejar correccions amb gran encert, per a reenfocar la planificació, alineant-la correctament al procés de desenvolupament que estava succeint. Això va ser així, degut a que el propi estudi previ que abans s'ha plantejat com necessari, es va realitzar durant el desenvolupament del TFM.

La gran dificultat associada al projecte ha vingut principalment de dos fronts. Un primer front ha sigut estudiar i entendre la definició del protocol RPL, descrita en el document RF6550, per a poder implementar ens canvis necessaris. El segon front, i mes constitutiu de la dificultat intrínseca al projecte ha sigut la interpretació del codi que implementa la solució del sistema OPENWSN. Això ha sigut imprescindible per trobar els punts de canvi necessaris, per així dur a terme la implementació del disseny conceptual, esdevingut del estudi del protocol RPL. Tenint així que interpretar gran part del codi de OPENWSN, per tal d'entendre com s'havia implementat el protocol RPL, resultant així no sols en un treball d'implementació sinó més en un estudi i adaptació a una solució ja implementada.

En quant a la metodologia associada a la part didàctica d'aquest treball, s'han pogut experimentar els avantatges i els inconvenients intrínsecs en l'ensenyament a distancia. Per

una banda s'ha tingut via lliure per a poder planificar les sessions de treball. Però per altra banda ha faltat un accés més directe al professorat, ja que degut als temps de resposta propis d'aquest sistema educatiu s'han desencadenat a vegades temps invertits perduts, al trobar-se el desenvolupament del treball en punt mort, degut a la necessitat d'assistència.

Encara i les situacions negatives viscudes en aquest treball m'alegro d'haver aconseguit arribar a bon port i la sensació de haver finalitzat una tasca important que m'ha servit d'aprenentatge. No només l'aprenentatge lligat a la temàtica pròpia del TFM, sinó també l'aprenentatge propi introduït amb molta cura pels professors per tal de generar, o millorar les capacitats d'autoaprenentatge i autoresolució en els alumnes.

Tal com s'han exposat en les conclusions de l'estudi comparatiu, s'han plantejat un conjunt de punts de treball immediats que caldria tenir en compte per tal de millorar la eficiència de la implementació del protocol RPL:

- ❖ Fragmentació de paquets. Per poder així solucionar la limitació actual en l'enviament d'informació amb els missatges DAO. Així paral·lelament s'estalviaria bastant ample de banda, i conseqüentment energia.
- ❖ Control en el enviament de missatges DIO aplicant el algorisme Trickle, enlloc de l'actual enviament controlat amb un sistema fixe periòdic. Així s'aconseguiria un descens notable en el consum d'ample de banda al establir-se la topologia.
- ❖ Control en el enviament de missatges DAO, seguint més de prop les característiques exposades en el RFC6550.

També caldria tenir en compte de solucionar a curt-mig termini els següents punts:

- ❖ Solucionar la limitació que estableix un numero màxim de fills o parents. D'aquesta manera es podrien establir topologies mes complexes.
- ❖ Fer a la xarxa autoconscient de les limitacions reals del hardware, memòria, processador, etc. Per tal de que actuï dinàmicament, establint per si mateixa les limitacions (no fixades en el codi), i eviti problemes adaptant-se d'una forma mes genèrica a la possible evolució del hardware utilitzat en aquest tipus de xarxes.
- ❖ Aplicar els sistemes de seguretat exposats en el RFC6550, per tal de garantir que les dades no han estat modificades o hi ha algun tipus d'enverinament. Ja que el mode de Source-Routing es propens a possibles enverinaments al modificar els paquets enviats en la capçalera de Source-Routing.

## 8. Glossari

6LoWPAN: Xarxa de nodes amb interfície IEEE 802.15.4 (RFC4919).

ACK: Acknowledgement, és un missatge que s'envia per confirmar l'arribada d'un o varis missatges.

CoAP: Constrained Application Protocol (CoAP) és un protocol de transferència web especialitzat per utilitzar-se amb els nodes i les xarxes de IoT.

DAG: Directed acyclic graph. Un gràfic amb la propietat de que tots els cantons estan orientats de manera que no existeixin bucles. Tots els cantons estan continguts en camins orientats cap i finalitzats en un o mes nodes root.

DIO: DODAG Information Object.

DIS: DODAG Information Solicitation.

DODAG: Destination Oriented DAG. Topologia que es forma a nivell d'IP, és part del protocol RPL.

ETX: Expected number of transmissions. Numero de vegades que s'envia un missatge fins a ser rebut.

HTTP: HyperText Transfer Protocol (Protocol de transferència d'hipertext) estableix el protocol per a l'intercanvi de documents d'hipertext i multimèdia al web.

IEEE 802.15.4-2006: Segona revisió del estàndard de 2003. L'estàndard que defineix els nivells físics (capa 1 o PHY) i de control d'accés al medi (capa 2 o MAC). Dissenyat per les xarxes WPAN, ha esdevingut el estàndard per WSN.

IEEE802.15.4e és un grup de treball de la IEEE enfocat per definir una modificació de MAC cap al estàndard existent 802.15.4-2006, per donar millor suport als mercats industrials.

IoT: The Internet of Things (Internet de les Coses)

LLNs: Low-Power and Lossy Networks.

M2M: Machine to Machine (maquina a maquina). Es un concepte genèric referit al intercanvi d'informació o comunicació en format de dades entre dues màquines remotes.

MP2P: Multipoint-to-Point.

MRHOF: Minimum Rank with Hysteresis Objective Function

ND: Neighbor Discovery - IPv6 Neighbor Discovery (RFC4861)

OF: Objective Function. Defineix com les mètriques de Routing, els objectius de optimització, i les funcions relacionades s'utilitzen per tal de calcular el Rank. També dicta com s'han de seleccionar el Parents, per tal de formar el DODAG.

OF0: Objective Function 0 correspon al Objective Code Point 0 (OCP0). OF0 nomes requereix de la informació base continguda en el RPL DIO, tal com Rank i el camp DODAGPreference que descriu la preferència administrativa.

OLSR: Optimized Link-State Routing

P2MP: Point-to-Multipoint.

P2P: Point-to-Point.

PPP: Point-to-Point Protocol.

PDR: Packet Delivery Ratio. La relació entre el nombre de paquets rebuts i el nombre de paquets enviats

PHY: Capa de nivell físic.

PRoPHET: Probabilistic Routing Protocol using History of Encounters and Transitivity

RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. Protocol d'encaminament per a xarxes de sensors.

Trickle: Algorisme que utilitza el protocol RPL per a regular l'enviament dels missatges DIO.

TCP/IP: Transmission Control Protocol / Internet Protocol.

UDP: User Datagram Protocol.

Xarxa ad-hoc: és un tipus de xarxa sense fils descentralitzada, que no depèn d'una infraestructura pre-existent (routers, punts d'accés). Enlloc d'això cada node participa de l'encaminament.

WPAN: Wireless Personal Area Network. Xarxes de comunicació entre dispositius molt propers al punt d'accés. Tenen un abans d'uns pocs metres i per a ús personal.

WSN: Wireless Sensor Networks.

## 9. Bibliografia

### [1] RFC6550 - RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks

Internet Engineering Task Force (IETF)

<https://tools.ietf.org/html/rfc6550>

### [2] RPL: IPv6 Routing Protocol for Low Power and Lossy Networks

Chung-Yi Chao

[https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjh59Lcze7MAhXESRoKHSgUASoQFggyMAI&url=http%3A%2F%2Fnetlab.csie.ntut.edu.tw%2Fseminar%2Fyear2015%2F20151008\\_CYC\\_RPL%2520IPv6%2520Routing%2520Protocol%2520for%2520Low%2520Power%2520and%2520Lossy%2520Networks.pptx&usq=AFQjCNEBBIpPvfw2ydTnTNqiO7kqZnbiKQ&sig2=5o-dnn0VguCuE5Ad8c6bWw](https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjh59Lcze7MAhXESRoKHSgUASoQFggyMAI&url=http%3A%2F%2Fnetlab.csie.ntut.edu.tw%2Fseminar%2Fyear2015%2F20151008_CYC_RPL%2520IPv6%2520Routing%2520Protocol%2520for%2520Low%2520Power%2520and%2520Lossy%2520Networks.pptx&usq=AFQjCNEBBIpPvfw2ydTnTNqiO7kqZnbiKQ&sig2=5o-dnn0VguCuE5Ad8c6bWw)

### [3] OPENWSN

Web Oficial

<https://openwsn.atlassian.net/wiki/display/OW/Home>

Protocol Stack

<https://openwsn.atlassian.net/wiki/pages/viewpage.action?pageId=688187>

### [4] OpenWSN: A Standards-Based Low-Power Wireless Development Environment

Thomas Watteyne, Xavier Vilajosana, Branko Kerkez, Fabien Chraim, Kevin Weekly, Qin Wang, Steven Glaser, Kris Pister

[https://www.researchgate.net/publication/271325478\\_OpenWSN\\_A\\_Standards-Based\\_Low-Power\\_Wireless\\_Development\\_Environment](https://www.researchgate.net/publication/271325478_OpenWSN_A_Standards-Based_Low-Power_Wireless_Development_Environment)



- [5] **Curso Aprender a Programar en C desde Cero**  
[https://www.youtube.com/playlist?list=PLw8RQJQ8K1ySN6bVHYEpDoh-CKVkl\\_uOF](https://www.youtube.com/playlist?list=PLw8RQJQ8K1ySN6bVHYEpDoh-CKVkl_uOF)
- [6] **Algoritmos y programación con lenguaje Python**  
<https://pythonizame.s3.amazonaws.com/media/Book/algoritmos-y-programacion-con-python/file/0e443016-5986-11e5-964d-04015fb6ba01.pdf>
- [7] **RFC2460 - Internet Protocol, Version 6 (IPv6) – Specification**  
Network Working Group  
<http://www.rfc-editor.org/rfc/pdf/rfc2460.txt.pdf>
- [8] **RFC4919 - IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals**  
Network Working Group  
<https://tools.ietf.org/pdf/rfc4919>
- [9] **Performance Evaluation of RPL Routing Protocol in 6lowpan**  
Haofei Xie , Guoqi Zhang , Delong Su, Ping Wang and Feng Zeng  
<http://0-ieeeexplore.ieee.org.cataleg.uoc.edu/stamp/stamp.jsp?tp=&arnumber=6933646>
- [10] **A Performance Evaluation Study of RPL: Routing Protocol for Low Power and Lossy Networks**  
J. Tripathi, J. C. de Oliveira, J. P. Vasseur  
<http://0-ieeeexplore.ieee.org.cataleg.uoc.edu/stamp/stamp.jsp?tp=&arnumber=5464820>
- [11] **RPL in a nutshell: A survey**  
Olfa Gaddour, Anis Koubâa  
[http://ac.els-cdn.com/S1389128612002423/1-s2.0-S1389128612002423-main.pdf?\\_tid=6395f10e-21bc-11e6-be5c-00000aacb361&acdnat=1464100549\\_ea161ccb07194bf9705ca9bc5084c02a](http://ac.els-cdn.com/S1389128612002423/1-s2.0-S1389128612002423-main.pdf?_tid=6395f10e-21bc-11e6-be5c-00000aacb361&acdnat=1464100549_ea161ccb07194bf9705ca9bc5084c02a)
- [12] **Routing in the Internet of Things**  
Lotte Steenbrink  
[https://inet.haw-hamburg.de/teaching/ss-2014/master-projekt/aw1\\_lotte\\_steenbrink.pdf](https://inet.haw-hamburg.de/teaching/ss-2014/master-projekt/aw1_lotte_steenbrink.pdf)
- [13] **Some Considerations on Routing in Particular and Lossy Environments**  
Thomas Clausen, Ulrich Herberg  
<https://www.iab.org/wp-content/IAB-uploads/2011/03/Herberg.pdf>
- [14] **Redes de sensores inalámbricos (RSI) RPL: Routing Protocol for Low-Power and Lossy Network**  
Leonardo Steinfeld  
[https://eva.fing.edu.uy/pluginfile.php/97273/mod\\_resource/content/1/LSteinfeld-RPL-2015.pdf](https://eva.fing.edu.uy/pluginfile.php/97273/mod_resource/content/1/LSteinfeld-RPL-2015.pdf)
- [15] **MASTER THESIS - Internet of things implementation with Raspberry Pi**  
Alex Nasarre Ramírez  
<http://upcommons.upc.edu/bitstream/handle/2099.1/24256/memoria.pdf>
- [16] **RFC4861 - Neighbor Discovery for IP version 6 (IPv6)**  
Network Working Group  
<https://tools.ietf.org/pdf/rfc4861>

**[17] Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)**

Internet Engineering Task Force (IETF)

<https://tools.ietf.org/html/rfc6552#section-4.1>

**[18] The Minimum Rank with Hysteresis Objective Function**

Internet Engineering Task Force (IETF)

<https://tools.ietf.org/html/rfc6719#section-3.3>

**[19] A Realistic Energy Consumption Model for TSCH Networks**

Xavier Vilajosana, Member, IEEE, Qin Wang, Fabien Chraim, Thomas Watteyne, Member, IEEE, Tengfei Chang, and Kristofer S. J. Pister

<http://0-ieeeexplore.ieee.org.cataleg.uoc.edu/stamp/stamp.jsp?tp=&arnumber=6627960>

**[20] A Performance Evaluation of RPL in Contiki - Page 54 - Fig.6.11**

Hazrat Ali

<https://www.diva-portal.org/smash/get/diva2:833572/FULLTEXT01.pdf>

**[21] 6LoWPAN Routing Header**

Internet-Draft - 6lo

<https://tools.ietf.org/pdf/draft-ietf-6lo-routing-dispatch-05.pdf>

**[22] RFC6554 - An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)**

Internet Engineering Task Force (IETF)

<https://tools.ietf.org/pdf/rfc6554.pdf>

**[20] MERPL: A More Memory-efficient Storing Mode in RPL**

Wei Gan, Zhiqiang Shi, Chen Zhang, Limin Sun, Dan Ionescu

[http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6781985&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D6781985](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6781985&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6781985)