



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Experiments about a real use of mobile ad-hoc networks

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTOR: Núria Franco Rosselló

DIRECTOR: Roc Meseguer Pallarès

DATA: 24 d'abril de 2009

Títol: Experiments about a real use of mobile ad-hoc networks

Autor: Núria Franco Rosselló

Director: Roc Meseguer Pallarès

Data: 24 d'abril de 2009

Resum

En aquest treball es presenta un estudi sobre les xarxes ad-hoc mòbils (MANET), els diferents mecanismes d'encaminament i els procediments per al càlcul de rutes existents en aquest tipus de xarxes.

S'han estudiat dos protocols en concret: BATMAN (Better Approach To Mobile Ad-hoc Networking) i OLSR (Optimized Link State Routing).

S'ha analitzat el comportament d'aquests dos protocols en diferents escenaris, implementats sobre el mateix sistema operatiu i en les mateixes condicions de treball.

A partir d'aquest anàlisi s'han obtingut una sèrie de conclusions, aquestes es poden resumir en que les MANETs obtenen uns resultats acceptables si no hi ha més de 4 salts en les comunicacions i són eficients per a treball cooperatiu.

Title: Experiments about a real use of mobile ad-hoc networks

Author: Núria Franco Rosselló

Director: Roc Meseguer Pallarès

Date: April, 24th 2009

Overview

This document presents a study of Mobile Ad-hoc NETWORKS (MANET), the different routing mechanisms and the procedures to calculate routes in such networks.

Two specific protocols were studied: BATMAN (Better Approach To Mobile Ad-hoc Networking) and OLSR (Optimized Link State Routing).

The behaviour of these two protocols was analyzed in different scenarios, which are on the same operating system and under the same work conditions.

We was obtained a different conclusions from this analysis, these can be summarized in that the MANTEs obtain results acceptable if there aren't more than 4 jumps in communications and the MANETs are efficient for cooperative work.

ÍNDEX

INTRODUCCIÓ	6
CAPÍTOL 1. ENTORN DE TREBALL.....	8
1.1. Conceptes bàsics	8
1.1.1. Xarxes mesh.....	8
1.1.2. Xarxes ad-hoc.....	8
1.1.3. MANET (Mobile Ad-hoc Network)	9
1.2. Algoritmes d'encaminament en MANET	9
1.2.1. Protocols d'encaminament	9
1.2.2. Procediments per al càlcul de rutes	10
1.2.3. Mètriques per al càlcul de l'enllaç	12
1.3. Protocols implementats.....	12
1.3.1. BATMAN (Better Approach To Mobile Ad-hoc Networking).....	12
1.3.2. OLSR (Optimized Link State Routing).....	14
1.3.3. Comparació entre els dos protocols	15
CAPÍTOL 2. PROVES REALITZADES.....	16
2.1. Hipòtesis.....	16
2.2. Punts en comú dels diferents escenaris.....	16
2.3. Escenari estàtic	16
2.3.1. Resultats de les proves	17
2.4. Multicamí	20
2.4.1. Resultats de les proves	20
2.5. Comparació entre escenari estàtic i multicamí.....	22
2.6. Mobilitat.....	23
2.6.1. Resultats de les proves	25
2.7. Grups	26
2.7.1. Resultats de les proves	26
2.8. Sumari.....	31
CONCLUSIONS.....	33
BIBLIOGRAFIA	35
ANNEXOS.....	37
ANNEX A. Configuració OLSRd	37
A.1. Instal·lació de l'OLSRd	37

A.2.	Plug-ins.....	39
A.3.	Fitxer de configuració (olsrd.conf)	41

INTRODUCCIÓ

Avui dia el món de les tecnologies inal·làmbriques està evolucionant de manera exponencial. Dia a dia surten nous elements pensats per aquests tipus de tecnologies, així com nous tipus de configuracions. Una de les configuracions físiques que s'està estudiant i provant de millorar són les xarxes MANET (Mobile Ad-hoc NETwork).

Les xarxes MANET consisteixen en un grup de nodes, inal·làmbrics i mòbils que es comuniquen entre ells. La xarxa ha d'anar descobrint els nodes que entren a formar part d'ella i els que deixen d'estar-hi, de manera que es detecti qualsevol canvi en la topologia. També és la pròpia xarxa la que ha d'aprendre el camí pel que enviar un paquet d'un node origen a un node destí, per a això existeixen diferents algoritmes d'encaminament i diferents protocols que els implementen.

S'han dut a terme un nombre molt elevat d'estudis sobre el comportament dels protocols en xarxes MANETs, però la major part d'aquests s'han fet en simulacions, amb la qual cosa no es tenen en compte els problemes que es poden trobar en un escenari real.

Amb aquest treball es pretén analitzar el comportament de dos protocols per a MANETs en un entorn real, aquests són OLSR i BATMAN. El primer és un estàndard i el segon es va crear com a millora del primer. A més els algoritmes d'encaminament que implementen són els contraris.

Per a poder dur a terme aquest anàlisi es començarà per comprendre una sèrie de conceptes bàsics, com són els diferents tipus de xarxes inal·làmbriques de les que surten les MANETs, és a dir xarxes mesh i xarxes ad-hoc.

Després es procedirà a estudiar en profunditat com funcionen les MANETs, és a dir, els algoritmes d'encaminament que utilitzen. Dins d'algoritmes d'encaminament es troben els protocols d'encaminament, que poden ser de tres tipus (proactiu, reactiu i híbrid), els procediments per a càlculs de rutes, dels quals n'hi ha dos (per vector distància i per estat de l'enllaç) i les mètriques per al càlcul de l'enllaç més comunes (per nombre de salts i per ETX).

Fet això es farà un estudi teòric sobre el comportament dels dos protocols, veient els algoritmes d'encaminament i com treballen cadascun d'ells.

A partir de totes aquestes dades es començaran a dur a terme les proves experimentals amb ambdós protocols, aquestes es faran en diferents tipus d'escenaris, des d'un escenari estàtic, passant a mobilitat, després multicamí i per últim es provaran de fer proves amb grups, simulant un treball col·laboratiu. Aquests escenaris seran en un entorn real, és a dir, que hi haurà elements a l'ambient que podran interferir en la comunicació, de manera que les dades que s'obtidran són les que es trobarien a l'entorn en el que s'implementen aquests tipus de xarxes. Amb les dades obtingudes es farà un estudi analític per poder comparar-los.

I per últim s'intentarà arribar a alguna conclusió sobre quin dels dos protocols podria obtenir uns millors resultats en el tipus de xarxes en les que s'està treballant i per a cadascun dels escenaris estudiats.

CAPÍTOL 1. ENTORN DE TREBALL

1.1. Conceptes bàsics

1.1.1. Xarxes mesh

Xarxes en les que la informació s'envia entre nodes en forma de tots contra tots i en una jerarquia plana. Pot ser una topologia total, tots els nodes estan connectats entre ells, o parcial, on els nodes estan connectats amb alguns dels altres nodes.

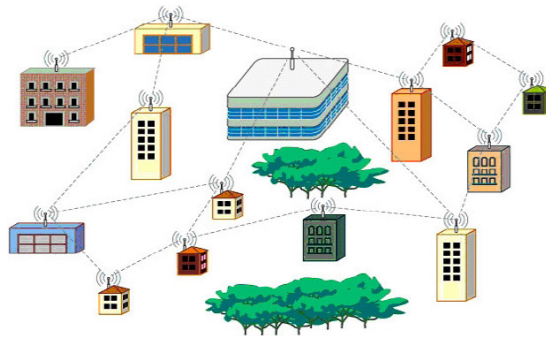


Fig. 1.1 Escenari típic d'una xarxa mesh

1.1.2. Xarxes ad-hoc

Xarxes autònomes i descentralitzades, on tots els nodes funcionen com a routers, aquests estan connectats per enllaços inal·làmbrics que s'organitzen de manera dinàmica i es comuniquen entre ells per formar una xarxa temporal. La topologia d'aquestes xarxes pot adoptar múltiples formes.

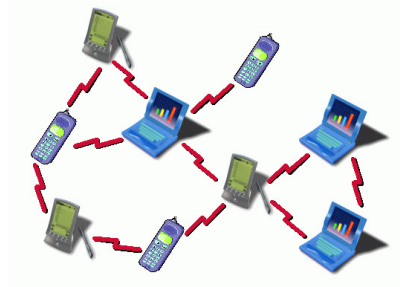


Fig. 1.2 Exemple de xarxa ad-hoc

1.1.3. MANET (Mobile Ad-hoc Network)

Xarxes mesh dinàmiques i mòbils. Consisteix en un grup de nodes mòbils que cooperen de manera dinàmica els uns amb els altres utilitzant enllaços inal·làmbrics amb multi-salt.



Fig 1.3 Exemple de MANET

1.2. Algoritmes d'encaminament en MANET

1.2.1. Protocols d'encaminament

1.2.1.1. Proactiu

Els nodes intercanvien la informació d'encaminament periòdicament i la mantenen actualitzada. Cada node guarda la informació d'encaminament de la resta de nodes de la xarxa en les seves taules. Quan canvia l'enllaç d'un node aquest ha d'informar a la xarxa i la resta de nodes actualitza les seves taules.

El principal avantatge d'aquest tipus d'enrutament és que se sap en tot moment quins nodes estan dins de la xarxa i quins a fora i no s'ha d'esperar fins a trobar una ruta. Però els principals inconvenients que presenta són la gran càrrega de CPU i l'alt tràfic de paquets.

1.2.1.2. Reactiu

El procediment de descobriment de rutes es fa només quan es sol·licita, és a dir, les rutes, per anar d'un node a un altre de la xarxa, es creen quan es necessiten i no s'emmagatzemen en tots els nodes. Quan un node emissor vol comunicar-se amb un altre node, el primer inicia el procediment de descobriment de rutes per trobar el camí per arribar al node destí. El node origen guarda aquest camí durant un cert temps, si en aquest temps algun

node del camí cau, el node origen per enviar dades al destí recalcula el camí a partir del node caigut.

1.2.1.3. *Híbrid*

Implementa característiques dels dos protocols anteriors. Les rutes d'alguns destins estan disponibles permanentment, en canvi les rutes per altres destins es descobreixen en el moment en que es necessiten. La necessitat dels protocols híbrids es deu a que per una banda els protocols proactius no són una solució òptima per a l'encaminament en MANETs de gran mobilitat, ja que les taules en aquestes xarxes van variant contínuament i això provoca retards importants en els càlculs de les rutes. Per altra banda els protocols reactius provoquen importants retards durant el temps de descobriment de les rutes.

El fet de que el protocol híbrid combina les millors característiques dels protocols proactiu i reactiu fa que sigui la millor solució per a l'encaminament en MANETs.

1.2.2. **Procediments per al càlcul de rutes**

1.2.2.1. *Vector distància*

Guarda la informació del següent salt al veí adjacent i el cost del camí de totes les destinacions que coneix. Utilitzant el vector distància d'encaminament, cada node manté una taula, anomenada vector distància, que conté la distància més curta i el següent salt al camí cap a cada destí de la xarxa.

Al començament, cada node només sap la distància dels veïns que estan directament connectats a ell, els nodes no tenen una visió global de la xarxa. Immediatament, cada node intercanvia el seu vector distància amb els veïns que hi estan directament connectats. Basant-se en els vectors distància que reben, cada node calcula el camí més curt cap a totes les destinacions que no són veïns adjacents. A continuació cada node envia les seves taules a la resta de nodes.

A la figura següent es pot veure un exemple de xarxa (formada per 12 nodes on un dels quals fa de router) que implementa vector distància, les taules que rep un dels nodes (J) de cadascun dels seus veïns i la taula que crea aquest node a partir de la informació rebuda.

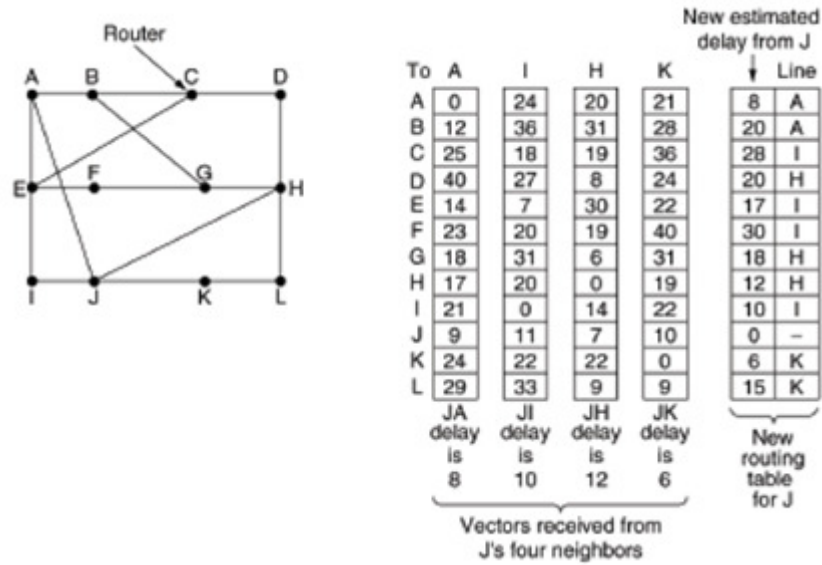


Fig 1.4 Exemple de xarxa implementant vector distància

1.2.2.2. Estat de l'enllaç

Guarda una còpia de la topologia de la xarxa i dels costos de tots els enllaços que coneix. Cada node té una visió completa de la topologia amb el cost de cadascun dels enllaços de la xarxa. La informació del cost de l'enllaç es transmet periòdicament a tots els nodes mitjançant un mecanisme d'inundació.

A la figura següent es pot veure un exemple de xarxa, formada per 6 nodes, que implementa estat d'enllaç i els paquets d'informació que envia cadascun dels nodes de la xarxa.

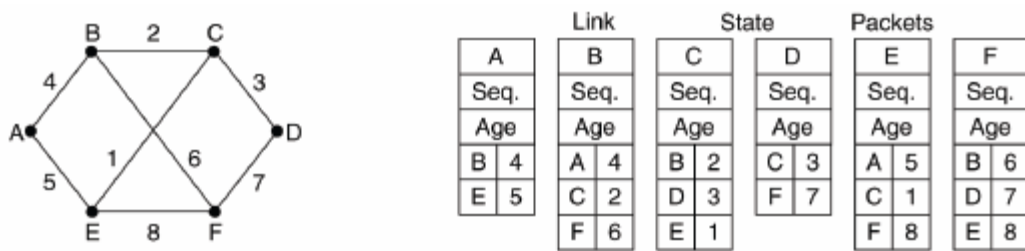


Fig 1.5 Exemple de xarxa implementant estat d'enllaç

1.2.3. Mètriques per al càlcul de l'enllaç

1.2.3.1. Nombre de salts

Amb aquesta mètrica es calcula el camí per el que hi haurà el mínim nombre de salts. La qualitat de l'enllaç per a aquest tipus de mètrica és un concepte binari, l'enllaç existeix o no existeix. L'avantatge principal d'aquest tipus de mètrica és la simplicitat. Una vegada coneguda la topologia, és fàcil comptar i minimitzar el nombre de salts entre un origen i un destí. A més, comptar el nombre de salts no requereix mesures addicionals.

El principal inconvenient d'aquesta mètrica és que no té en compte ni l'ample de banda ni les pèrdues de paquets, és a dir, la qualitat de l'enllaç. Una ruta en la que es redueix al mínim el nombre de salts no necessàriament maximitza el rendiment en el flux.

1.2.3.2. ETX (Expected Transmission Count)

Aquesta mètrica estima el nombre de retransmissions necessàries per enviar un paquet unicast, per fer-ho mesura la pèrdua de taxa enviant un nombre parell de paquets broadcast entre nodes veïns, analitzant els paquets rebuts en els últims 10 segons el node pot calcular la pèrdua de taxa en els enllaços, tant en el camí d'anada com en el camí de tornada, ja que coneix l'interval de temps entre dos paquets i segons arriben els ACKs es va coneixent el retard, l'estat de la xarxa...

Per calcular l'ETX cada node mesura la probabilitat de que un paquet arribi amb èxit al receptor (d_f) i la probabilitat que un ACK arribi amb èxit a l'emissor (d_r). El valor de ETX ve donat per:

$$ETX = \frac{1}{d_f \times d_r} \quad (1.1)$$

1.3. Protocols implementats

1.3.1. BATMAN (Better Approach To Mobile Ad-hoc Networking)

Es tracta d'un protocol proactiu que fa servir l'encaminament per vector distància i per calcular l'enllaç ho fa per nombre de salts. S'utilitza per a xarxes mesh ad-hoc inal·làmbriques i per a MANETs.

L'algoritme de BATMAN consisteix en dividir entre els nodes el que se sap del millor camí extrem a extrem de la xarxa. Cada node manté només la informació

sobre el millor següent salt cap a cadascun dels altres nodes. Quan hi ha algun canvi en la topologia de la xarxa els nodes envien paquets avisant d'aquests canvis.

Cada node transmet missatges broadcast (anomenats missatges originadors o OGMs) per informar als nodes veïns sobre la seva existència. Aquests veïns reenvien aquests missatges per informar als seus veïns de la existència de "l'originador" i així successivament, d'aquesta manera s'inunda la xarxa amb missatges "d'originador". Els missatges OGM contenen la direcció de l'originador, la direcció del node que transmet el paquet, un TTL i un nombre de seqüència.

Si els enllaços del camí pel qual es transmeten els paquets OGM són de mala qualitat o estan saturats es poden produir retards o pèrdues de paquets.

Per a saber si un paquet OGM s'ha rebut una o varies vegades, aquest conté un nombre de seqüència, donat per l'originador del paquet. Cada node reenvia tots els paquets rebuts més d'una vegada, però només envia el paquet que ha rebut del veí identificat com el millor salt al node originador del paquet.

Així doncs, s'inunda la xarxa de manera selectiva i s'informa a tots els nodes de l'existència d'un nou node a la xarxa. Si el nou node és considerat com el millor node veí s'han de reconfigurar les taules.

A la següent taula es mostra la taula d'encaminament del daemon batmand [10], on apareixen totes les IPs de la xarxa i el següent salt a través del que arriba a cadascun dels nodes.

Taula 1.1 Taula d'encaminament del daemon batmand

Originator (#/255)	Nexthop [outgoingIF]:	Potential nexthops
... [B.A.T.M.A.N. 0.3, MainIF/IP: wlan0/192.168.2.3, UT: 0d 3h19m]		
192.168.2.4 (165)	192.168.2.4 [wlan0]:	192.168.2.4 (165)
192.168.2.7 (59)	192.168.2.1 (75)	192.168.2.6 (33)
192.168.2.5 (52)	192.168.2.17 (87)	192.168.2.14 (54)
192.168.2.14 (130)	192.168.2.14 [wlan0]:	192.168.2.14 (130)
192.168.2.5 (59)	192.168.2.17 (118)	192.168.2.4 (79)
192.168.2.1 (63)	192.168.2.7 (68)	192.168.2.6 (59)
192.168.2.6 (106)	192.168.2.5 [wlan0]:	192.168.2.6 (106)
192.168.2.7 (104)	192.168.2.1 (70)	192.168.2.4 (71)
192.168.2.5 (106)	192.168.2.17 (49)	192.168.2.14 (58)
192.168.2.7 (138)	192.168.2.7 [wlan0]:	192.168.2.7 (138)
192.168.2.1 (79)	192.168.2.4 (93)	192.168.2.6 (68)
192.168.2.5 (106)	192.168.2.17 (47)	192.168.2.14 (48)
192.168.2.1 (182)	192.168.2.1 [wlan0]:	192.168.2.1 (182)
192.168.2.7 (58)	192.168.2.4 (82)	192.168.2.6 (48)
192.168.2.5 (69)	192.168.2.17 (53)	192.168.2.14 (59)
192.168.2.17 (162)	192.168.2.17 [wlan0]:	192.168.2.17

(162)	192.168.2.1 (82)	192.168.2.4 (92)	192.168.2.7 (
74)	192.168.2.6 (63)	192.168.2.5 (67)	192.168.2.14 (85)
192.168.2.5	(136)	192.168.2.5 [wlan0]:	192.168.2.5
(136)	192.168.2.7 (90)	192.168.2.1 (82)	192.168.2.4 (
89)	192.168.2.6 (64)	192.168.2.17 (55)	192.168.2.14 (41)

1.3.2. OLSR (Optimized Link State Routing)

El protocol OLSR utilitza l'enrutament proactiu i per estat d'enllaç. Està pensat específicament per a connexions sense fils, especialment per a MANETs, tot i que també es pot implementar sobre xarxes cablejades, com Ethernet.

Ell mateix permet generar rutes dinàmiques, que s'actualitzen constantment en funció de la qualitat de l'enllaç.

Un node OLSR envia missatges "Hello" cada cert interval de temps (per defecte dos segons) per a que els nodes veïns detectin la seva presència. Cada node computa una estadística del nombre de "Hellos" rebuts i perduts de cada veí per a obtenir informació sobre la topologia i la qualitat dels enllaços. Qualitat de l'enllaç s'entén com la probabilitat de que els paquets arribin correctament d'un node a la resta.

La informació de topologia obtinguda es difon amb els missatges TC (de control de topologia) i es reenvia a uns nodes determinats, anomenats "rellevadors multipunt", encarregats de reenviar els missatges de control.

Existeixen dos tipus més de missatges, el "HNA" que informa quan un node ofereix una passarel·la a altres xarxes, i el "MID", que informa quan un node té múltiples interfícies.

El software implementat (OLSRd) [11] pot funcionar sobre els sistemes operatius Windows, Linux, OS X, VxWorks, NetBSD, FreeBSD, OpenBSD, entre d'altres.

Pot treballar sobre múltiples interfícies i pot ampliar-se amb diferents plug-ins.

Per obtenir la mètrica de l'enllaç calcula l'ETX (expected transmission count) enviant paquets broadcast cada dos segons (configurables) i mesurant la seva recepció.

A la figura següent podem veure les dades que ensenya per pantalla el daemon OLSRd, on es pot veure que coneix totes les adreces que hi ha a la xarxa, l'estat de l'enllaç (LQ) i l'ETX cap a cada node.

```

File Edit View Terminal Tabs Help

Source IP addr  Dest IP addr      LQ      ETX
192.168.2.5     192.168.2.7     0.502/0.486  4.097
192.168.2.7     192.168.2.5     0.482/0.502  4.130
Processing TC from 192.168.2.7, seq 0x293d

--- 13:43:24.505439 ----- DIJKSTRA

--- 13:43:24.505458 ----- LINKS

IP address      hyst      LQ      ETX
192.168.2.7     0.000    0.502/0.486  4.097

--- 13:43:24.505496 ----- TWO-HOP NEIGHBORS

IP addr (2-hop) IP addr (1-hop) Total cost

--- 13:43:24.505510 ----- TOPOLOGY

Source IP addr  Dest IP addr      LQ      ETX
192.168.2.5     192.168.2.7     0.502/0.486  4.097
192.168.2.7     192.168.2.5     0.486/0.502  4.097
    
```

Fig 1.6 Dades que mostra el daemon OLSRd

1.3.3. Comparació entre els dos protocols

A la taula següent es pot veure una comparació entre els protocols BATMAN i OLSR segons els algorismes d'encaminament que implementen cadascun d'ells.

Taula 1.2. Taula comparativa entre els protocols

Mecanismes d'encaminament		
	BATMAN	OLSR
Protocol	proactiu	reactiu
Càlcul de rutes	vector distància	estat de l'enllaç
Mètriques	nombre de salts	ETX

CAPÍTOL 2. PROVES REALITZADES

2.1. Hipòtesis

Per estudiar els protocols es van fer una sèrie de proves en diferents escenaris, el primer va ser un escenari estàtic, on es volia analitzar el comportament segons el nombre de salts que s'havien de fer per enviar dades d'un node origen a un node destí, amb aquest escenari es volia comprovar si el fet d'afegir nodes al camí provocava que baixés el rendiment i quin era el màxim nombre de nodes amb el que es podia mantenir una comunicació acceptable.

El segon escenari que es va provar va ser multicamí, per analitzar si el fet de que un node tingués més d'una ruta per la que encaminar els paquets ajudava a que s'obtinguessin uns millors resultats.

La tercera prova es va fer amb mobilitat, es pretenia estudiar si el fet de que el camí, pel que s'enviaven els paquets, anés variant provocava que el rendiment donés uns resultats molt inferiors a les proves fetes amb el mateix nombre de salts, però només amb un camí. És a dir, analitzar si el canvi continu de topologia de la xarxa afecta molt negativament al protocol.

Per últim es va dur a terme una prova de treball en grup, amb aquest escenari es pretenia analitzar les possibles interferències que es poden produir al tenir varies màquines enviant o rebent alhora per un medi compartit.

2.2. Punts en comú dels diferents escenaris

Totes les proves es van fer amb ambdós protocols, BATMAN i OLSR, fent servir el sistema operatiu UBUNTU 8.04 i ordinadors HP NX6310 (excepte en el cas de les proves de grup, que s'hi van afegir ordinadors HP NX6110). Es va enviar tràfic simulat UDP, TCP i ICMP, amb les comandes ping i iperf.

2.3. Escenari estàtic

En aquest escenari teníem d'un a quatre salts, de manera que cadascun dels nodes només veia als seus nodes adjacents.

A la figura següent podem veure la topologia lògica de l'escenari amb cinc nodes.

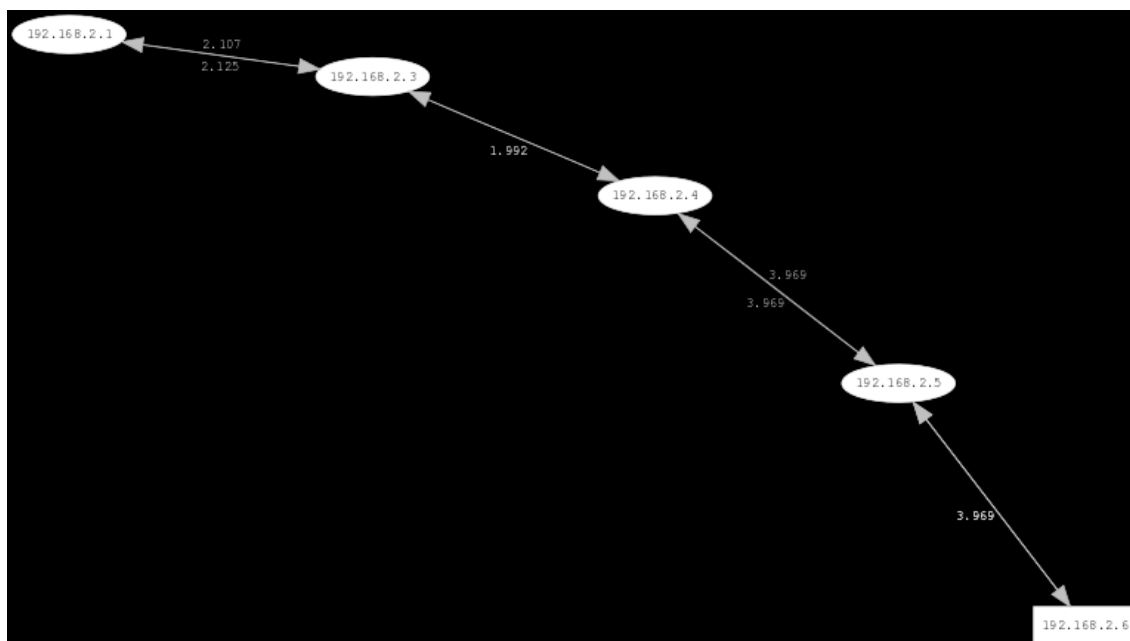


Fig 2.1. Topologia de l'escenari estàtic

2.3.1. Resultats de les proves

A la gràfica següent es pot veure el throughput rebut respecte de la càrrega oferta. En el cas de tres salts el comportament d'ambdós protocols és pràcticament el mateix. A partir de 1.5Mbps el throughput rebut deixa d'augmentar i passa a seguir una línia recta sobre els 1.5Mbps, amb la qual cosa comença a allunyar-se del cas ideal. Tot i això els valors obtinguts són acceptables.

Afegint un node a la xarxa, és a dir amb quatre salts, el comportament empitjora respecte al cas anterior, i es pot observar que OLSR funciona lleugerament millor que BATMAN. En aquest cas la línia no segueix en cap moment la del cas ideal.

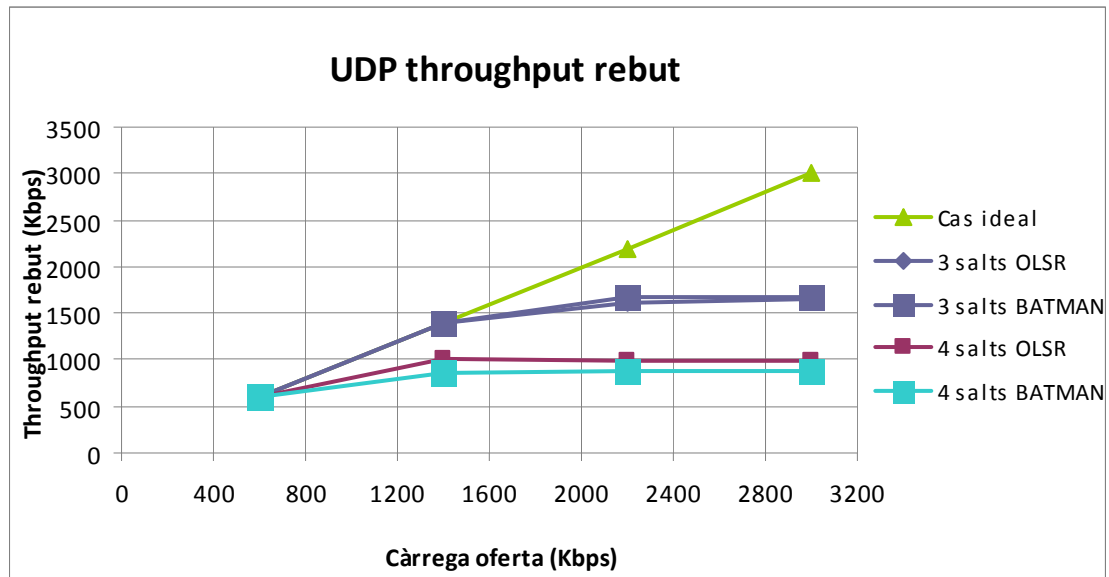


Fig 2.2. Avaluació del throughput rebut en UDP

La pèrdua de paquets en aquest escenari era pràcticament nul·la. En el cas de BATMAN per 3 salts es perdia un 0% dels paquets i per a 4 salts es perdia el 0.01% dels paquets. I per al cas de OLSR es perdia un 0% dels paquets, tant per a 3 com per a 4 salts.

La gràfica següent mostra el jitter obtingut respecte de la càrrega oferta, jitter s'entén com la variació del retard entre paquets. Per tant interessa un jitter petit. En la gràfica podem observar com aquest augmenta amb el nombre de salts. En aquest cas també té un millor comportament OLSR respecte de BATMAN.

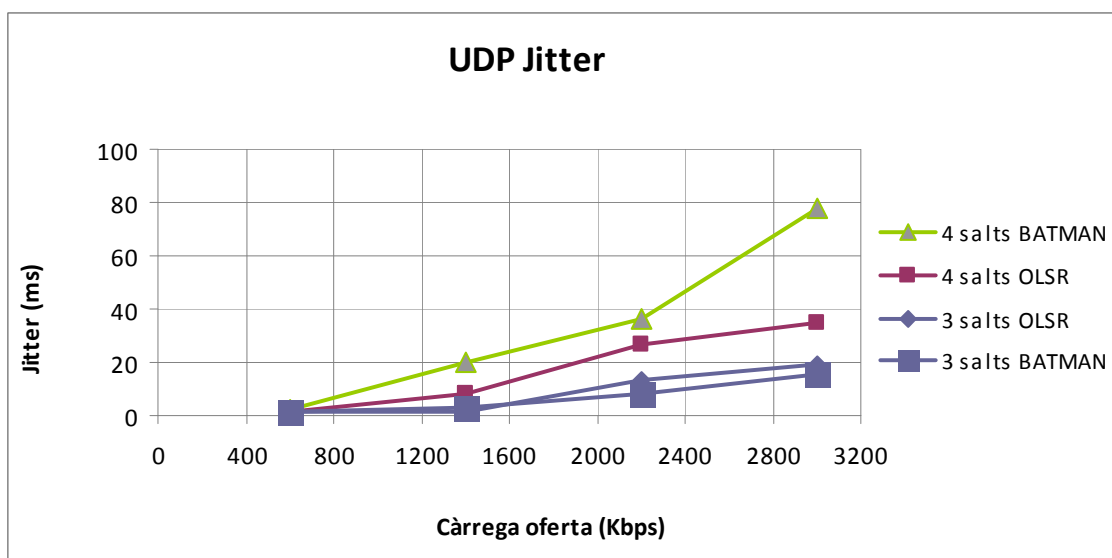


Fig 2.3. Estudi del Jitter

La gràfica següent mostra el RTT (Round-Trip delay Time) respecte del nombre de salts. El RTT indica el temps d'anada i tornada d'un paquet, és a dir, el retard d'aquest. Interessa que el valor sigui el més petit possible.

Es pot observar que si s'envien paquets de 64 bytes el comportament entre els dos protocols és pràcticament el mateix, en canvi si els paquets són més grans, de 1024 bytes, varia: BATMAN té un comportament millor fins als 3 salts, en 4 salts empitjora, i OLSR passa a obtenir millors resultats.

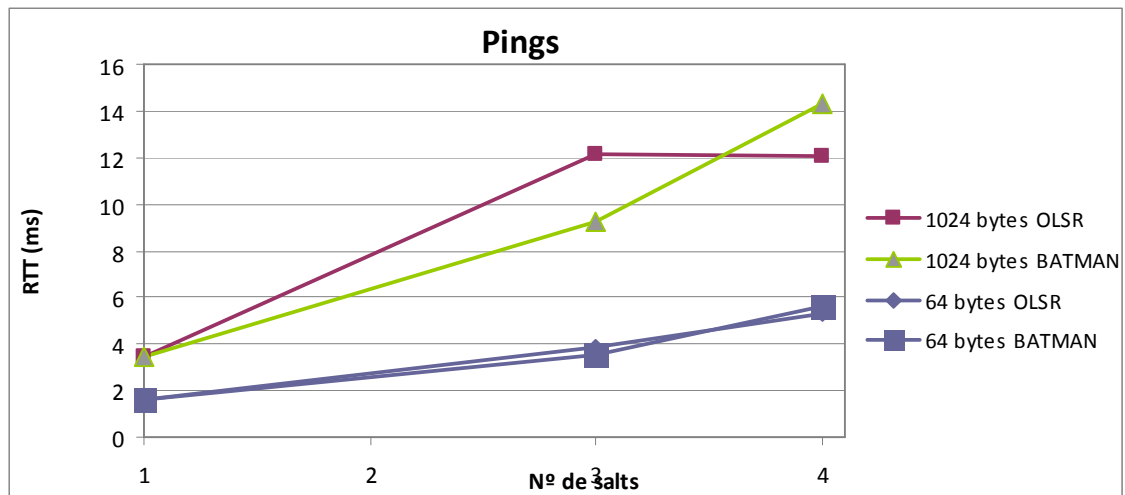


Fig 2.4. Estudi del RTT

El següents resultats mostren el throughput en TCP respecte del nombre de salts. En aquest cas OLSR i BATMAN tenen el mateix comportament.

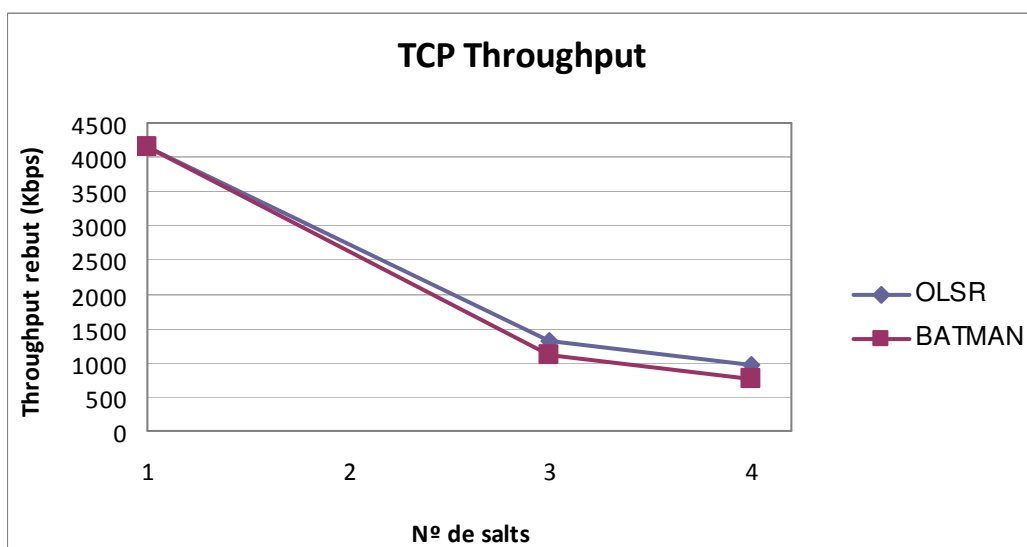


Fig 2.5. Throughput en TCP

2.4. Multicamí

En aquest escenari es va provar que un node pogués transmetre a un altre, amb la possibilitat de tenir dos camins per a fer-ho, les màquines que s'enviaven paquets eren la 192.168.2.1 i la 192.168.2.5, i els paquets podien passar a través de la màquina amb la IP 192.168.2.3 o de la màquina amb la IP 192.168.2.4, però mai directament; és a dir que sempre hi havia dos salts per arribar del node origen al node destí.

La topologia de l'escenari va ser la següent:

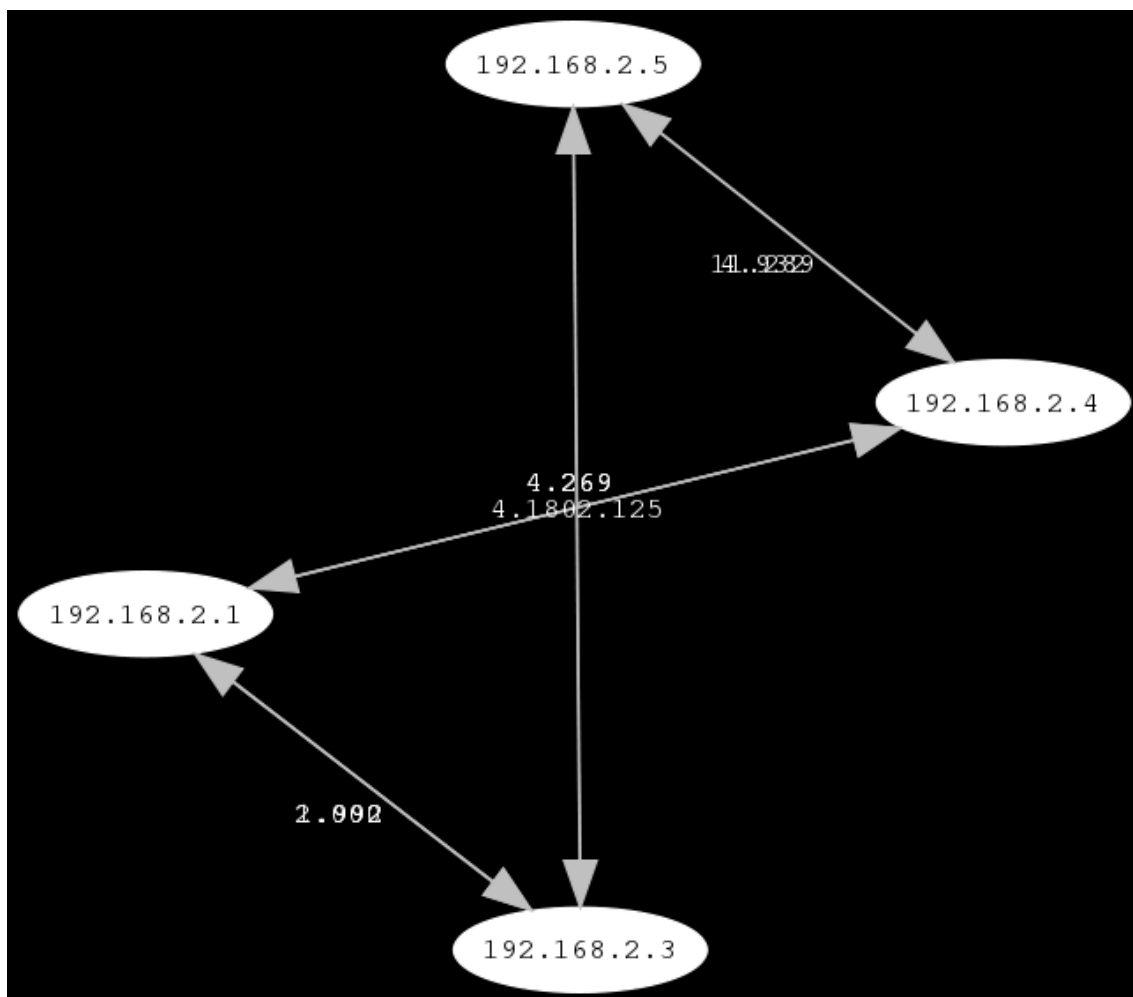


Fig 2.6. Topologia escenari multicamí

2.4.1. Resultats de les proves

Com es pot observar a la figura 2.11, on es mostra el throughput rebut respecte a la càrrega oferta, tots dos protocols tenen un comportament idèntic fins als 2Mbps, a 2.5Mbps (aproximadament) el comportament d'OLSR és

lleugerament millor que el de BATMAN, però a 3Mbps passa el contrari, ja que OLSR empitjora i BATMAN es manté constant.

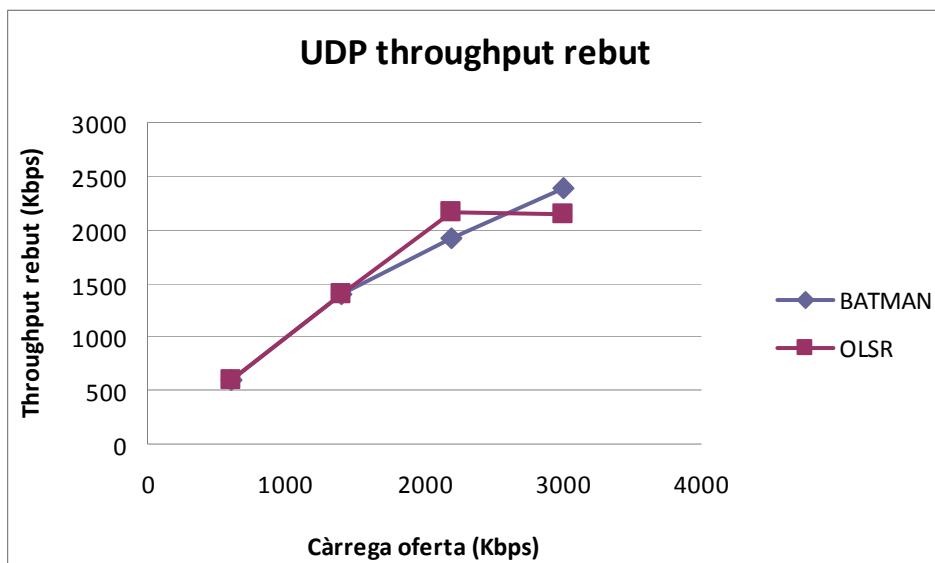


Fig 2.7. Comparació del throughput rebut en UDP

El jitter en aquest escenari presenta uns resultats similars per a ambdós protocols fins a 1.5Mbps (aproximadament), a partir d'aquest valor el comportament és millor per al cas d'OLSR respecte al de BATMAN, com es mostra a la gràfica següent.

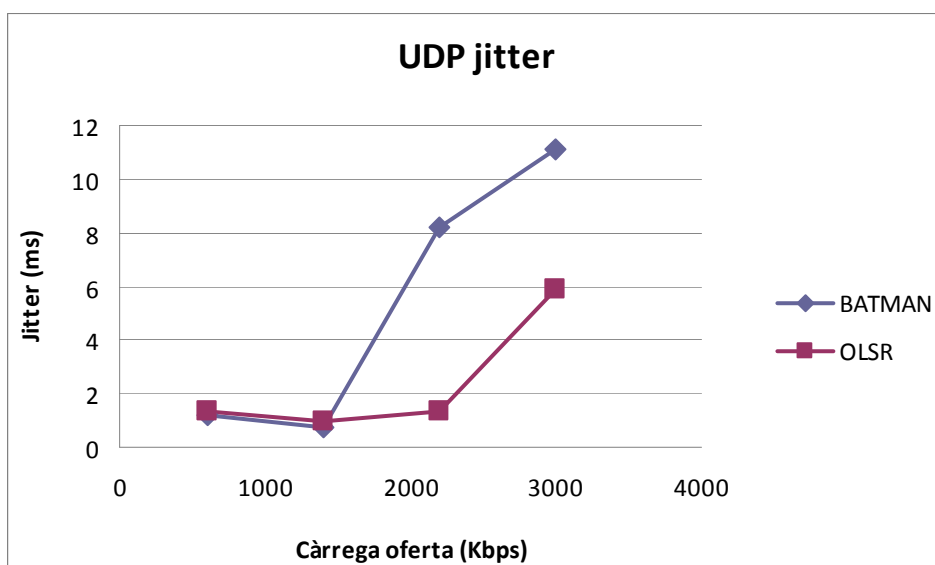


Fig 2.8. Avaluació del jitter en l'escenari estàtic

En la taula següent s'observa el throughput obtingut en TCP, el comportament d'ambdós protocols és molt similar, tot i que OLSR té uns resultats lleugerament inferiors als de BATMAN, en tots els casos excepte en 3 salts, que la diferència és una mica més notable.

Taula 2.1. Throughput en TCP

TCP		
Salts	BATMAN	OLSR
1	2180	2080
2	2190	2160
3	2400	1850
4	2230	2170
Mitjana	2250,00	2065,00

2.5. Comparació entre escenari estàtic i multicamí

A la gràfica següent es compara el throughput rebut en UDP entre el cas de l'escenari multicamí i els casos de 3 i 4 salts. El cas que té un pitjor comportament és en 4 salts, tant en BATMAN com en OLSR. El cas de 3 salts és una mica millor, i també es comporten pràcticament igual tots dos protocols. I com s'ha dit anteriorment, en multicamí té uns resultats millors OLSR fins a uns 2.5Mbps, però a partir d'aquest valor es comporta millor BATMAN.

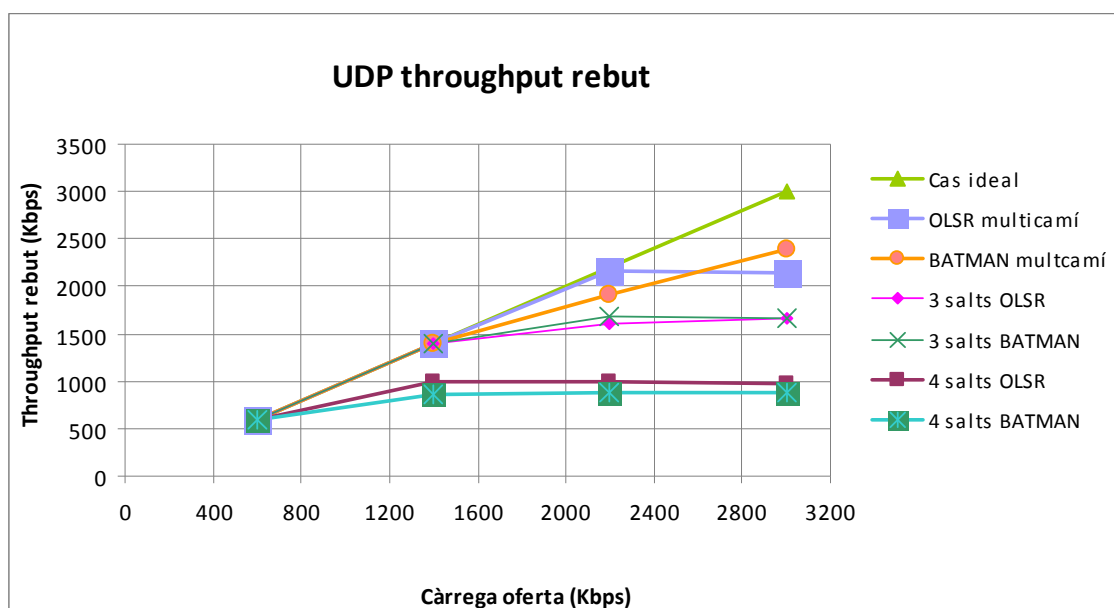


Fig 2.9. Comparació entre escenari estàtic i multicamí

2.6. Mobilitat

En aquest escenari teníem sis nodes, distribuïts igual que en el primer escenari (estàtic), en aquest cas, però, un dels nodes es desplaça des del primer fins a l'últim, de manera que s'esperava aconseguir que el nombre de salts entre el node transmissor i el receptor anés variant de cinc a un, tot i que degut a la potència de la Wi-Fi no es va poder aconseguir, ja que els nodes es veien a menys salts dels desitjats.

La transmissió anava de la màquina amb la IP 192.168.2.1 a la màquina amb la IP 192.168.2.6. La figura 2.10 mostra la topologia de l'escenari.

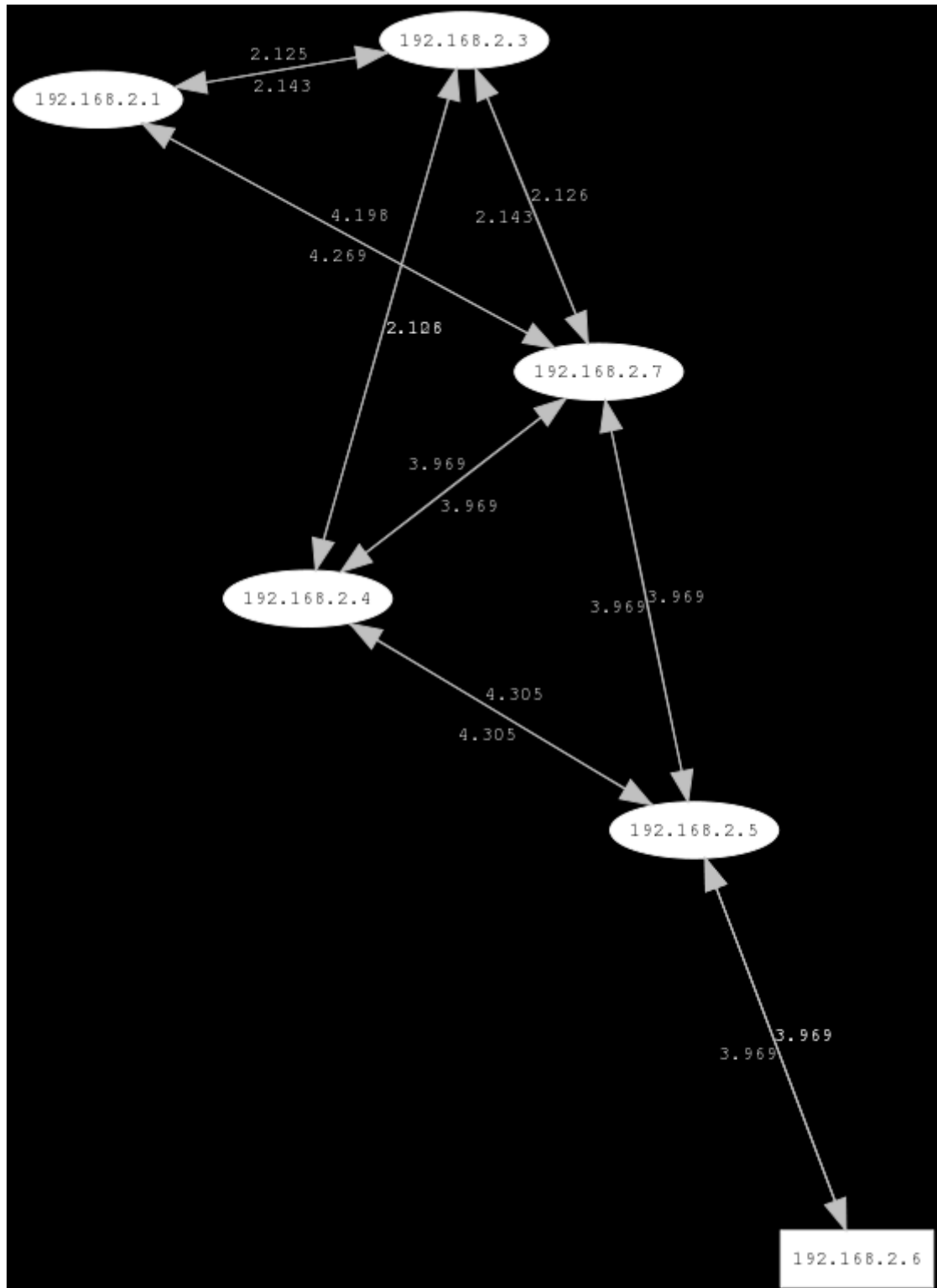


Fig 2.10. Topologia escenari mobilitat

2.6.1. Resultats de les proves

Com es pot observar a la següent gràfica el comportament de tots dos protocols en el cas de throughput rebut en UDP és molt similar i tenen uns resultats que s'aproximen significativament al cas ideal, és a dir que reben tanta càrrega com la que s'ofereix.

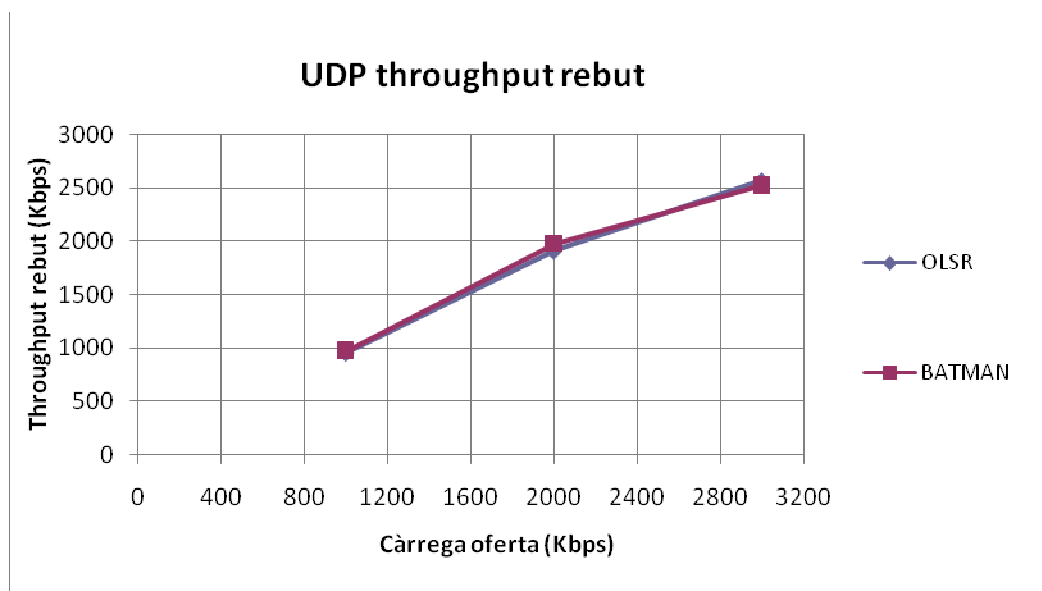


Fig 2.11. Throughput rebut en UDP

En el cas de TCP es pot veure que pràcticament segueixen una línia recta sobre els 2000Kbps, i tots dos protocols tenen aproximadament els mateixos resultats. En la gràfica podem observar que en el cas de BATMAN s'arriba a tres salts, en canvi en OLSR s'arriba a quatre salts, això pot ser degut a que el primer protocol triga menys a adonar-se que tenia un camí més curt per anar del node origen al node destí, mentre que el segon se n'adonava més tard.

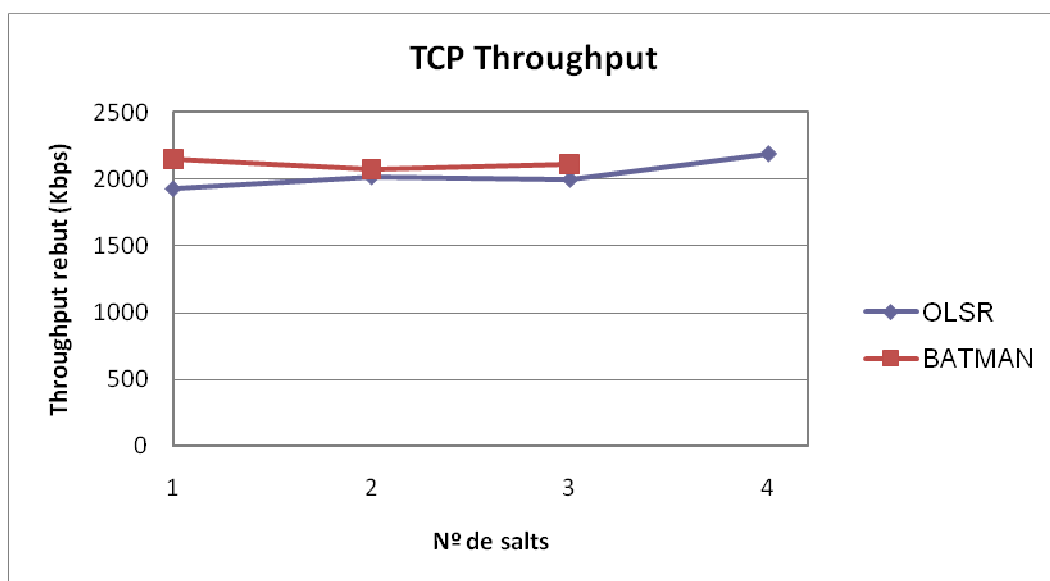


Fig 2.12. Throughput en TCP

2.7. Grups

En aquesta prova es van dividir els ordinadors en tres grups, un format per dos ordinadors HP NX6110 (grup 2) i dos formats per tres ordinadors HP NX6310 (grups 1 i 3).

Primer es van fer transmissions internes dins de cadascun dels grups, és a dir tots els grups enviaven alhora, transmetent cadascun d'ells dintre del seu propi grup, després només transmetien dos dels tres grups, també fent transmissions internes.

Per últim es transmetia d'un grup a l'altre, tots tres grups transmetent alhora, el grup 3 transmetia al grup 2, aquest transmetia al grup 1 i aquest al 3.

2.7.1. Resultats de les proves

2.7.1.1. Transmissió interna

A la següent gràfica es pot observar el throughput rebut respecte de la càrrega oferta en UDP per a cadascun dels grups i per a cada protocol. Com es pot veure els dos protocols tenen un comportament similar en cadascun dels grups, el que varia són els resultats dels diferents grups, el grup 2 té un comportament molt inferior a la resta.

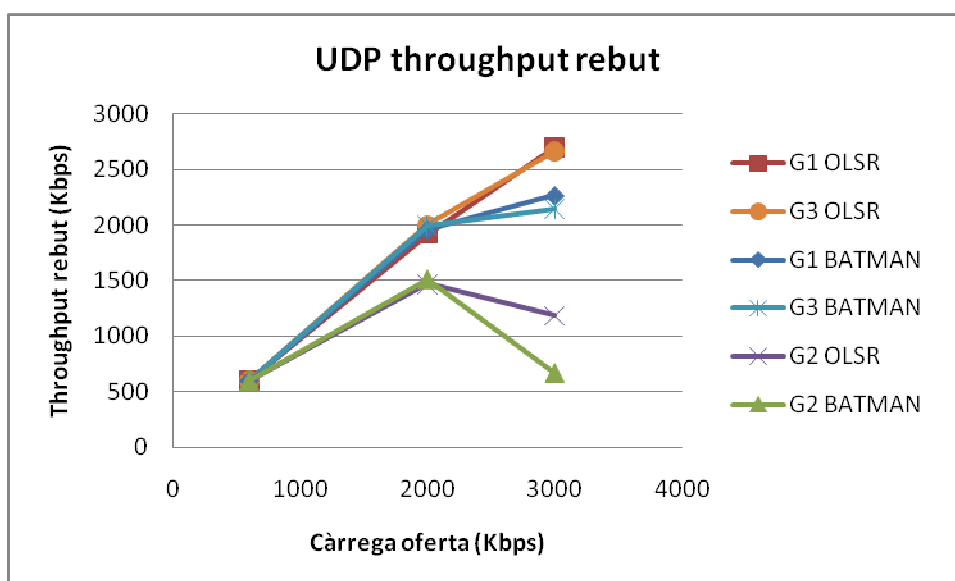


Fig 2.13. Throughput rebut UDP

En la figura 2.14 podem veure el jitter respecte de la càrrega oferta. En aquest cas és el grup 2 el que té el pitjor resultat. La resta de grups obtenen comportaments molt propers al cas ideal, com s'ha dit en apartats anteriors convé un jitter el més petit possible.

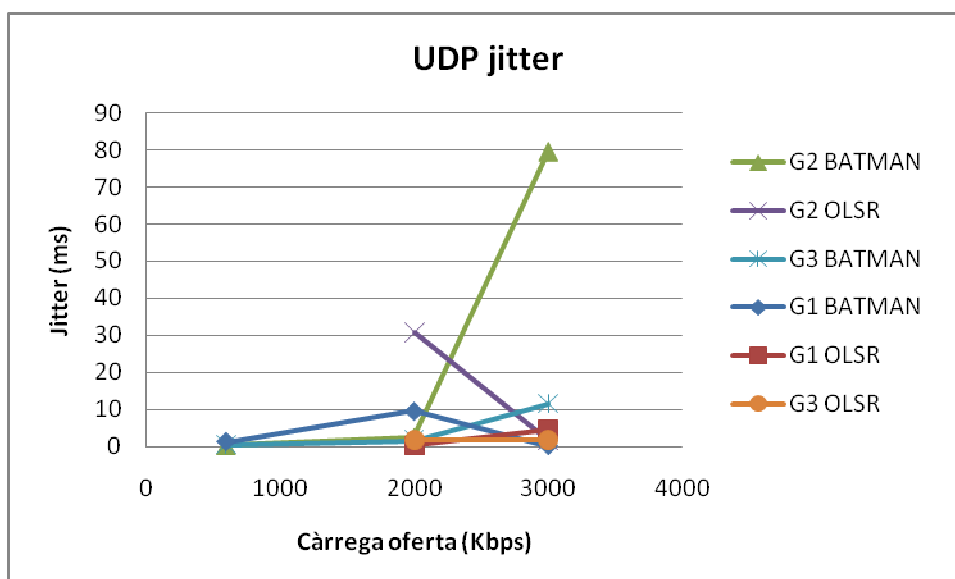


Fig 2.14. UDP jitter

A la taula següent tenim els resultats de les proves de transmissió de paquets TCP. Igual que en el cas de UDP el grup 2 és el que té un comportament notablement inferior pel que fa al throughput rebut. També podem veure el nombre de paquets rebuts fora d'ordre, en cap dels casos n'hi ha hagut, i el nombre de retransmissions necessàries per a cadascun dels grups i protocols,

per als grups 1 i 2 OLSR necessita més retransmissions que BATMAN, en canvi per al grup 3 passa al contrari.

Taula 2.2. Paquets TCP

TCP:	G1		G2		G3	
	BATMAN	OLSR	BATMAN	OLSR	BATMAN	OLSR
Throughput rebut (Kbps)	2210	2297	393	502	1730	2373
Paquets rebuts fora d'ordre	-	0,00	0,00	0,00	0,00	0,00
Nº de retransmissions	-	349,33	0,00	22,33	173,50	0,00

2.7.1.2. Transmissió entre grups

En la gràfica següent es mostra la relació entre el throughput rebut respecte de la càrrega oferta en UDP. Com es pot observar el comportament d'OLSR en tots els grups és pitjor que el de BATMAN. Tots tres grups, en BATMAN, es comporten de manera molt similar fins als 2Mbps, a partir d'aquest punt el grup 1 baixa considerablement el seu rendiment. Pel que fa a OLSR el grup 1 torna a ser el que té pitjor rendiment, i el grup 2 el que millor comportament té.

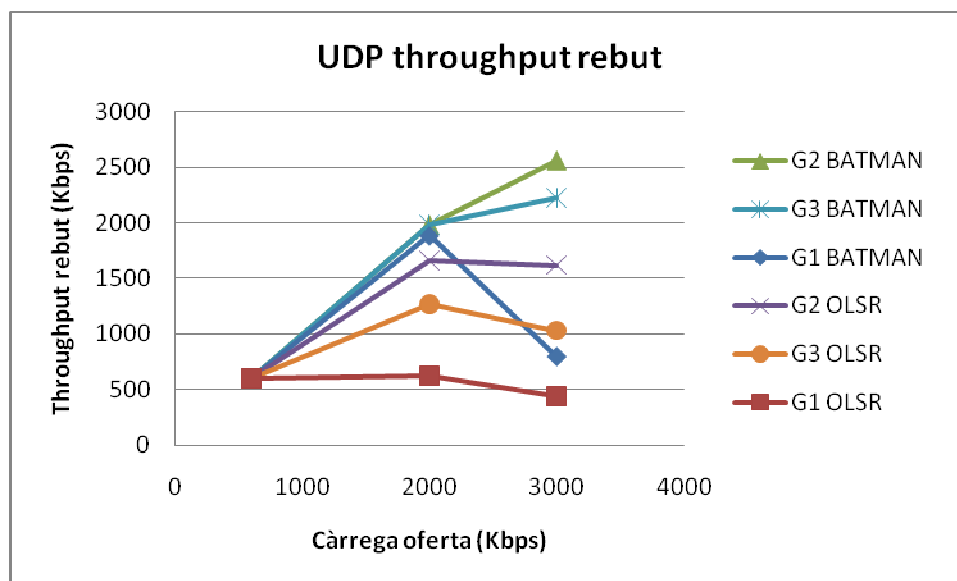


Fig 2.15. UDP throughput rebut

En el cas del jitter els grups 1 i 3 treballant sobre OLSR són els que pitjors resultats tenen. La gràfica següent mostra aquests resultats.

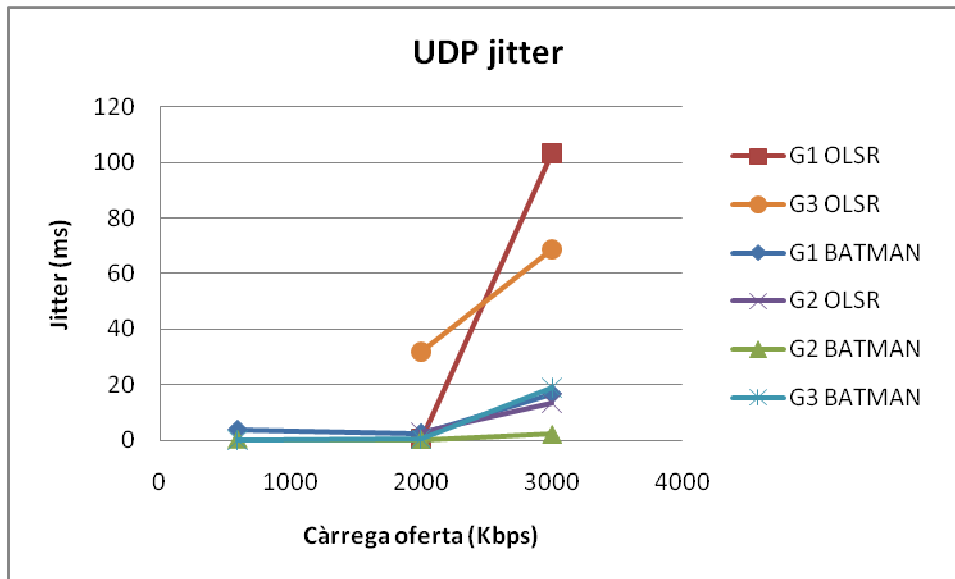


Fig 2.16. UDP jitter

En la transmissió de paquets TCP es pot observar a la taula següent que el grup 1 té pitjors resultats pel que fa al throughput rebut. En aquest cas veiem que sí que hi ha hagut paquets rebuts fora d'ordre, sobretot en el grup 3 treballant sobre OLSR. I pel que fa al nombre de retransmissions el grup 3 és el que n'ha necessitat més.

Taula 2.3. Paquets TCP

TCP:	G1		G2		G3	
	BATMAN	OLSR	BATMAN	OLSR	BATMAN	OLSR
Throughput rebut (Kbps)	751,75	580,33	902,00	1300,00	2587,50	2236,67
Paquets rebuts fora d'ordre	2,50	0,00	705,50	1,50	265,50	2900,67
Nº de retransmissions	729,00	80,50	729,00	288,00	1069,50	587,00

2.7.1.3. Comparació entre transmissió interna i entre grups

Fent la mitjana entre les transmissions internes i entre grups s'han obtingut una sèrie de resultats.

A la gràfica següent es mostra la relació entre el throughput rebut respecte de la càrrega oferta. El que obté pitjors resultats és OLSR en transmissió interna. BATMAN obté bons resultats, tant per transmissió interna com per transmissió entre grups, fins als 2Mbps de càrrega oferta, a partir d'aquest valor els resultats empitjoren. Pel que fa a OLSR entre grups obté uns resultats acceptables i lineals.

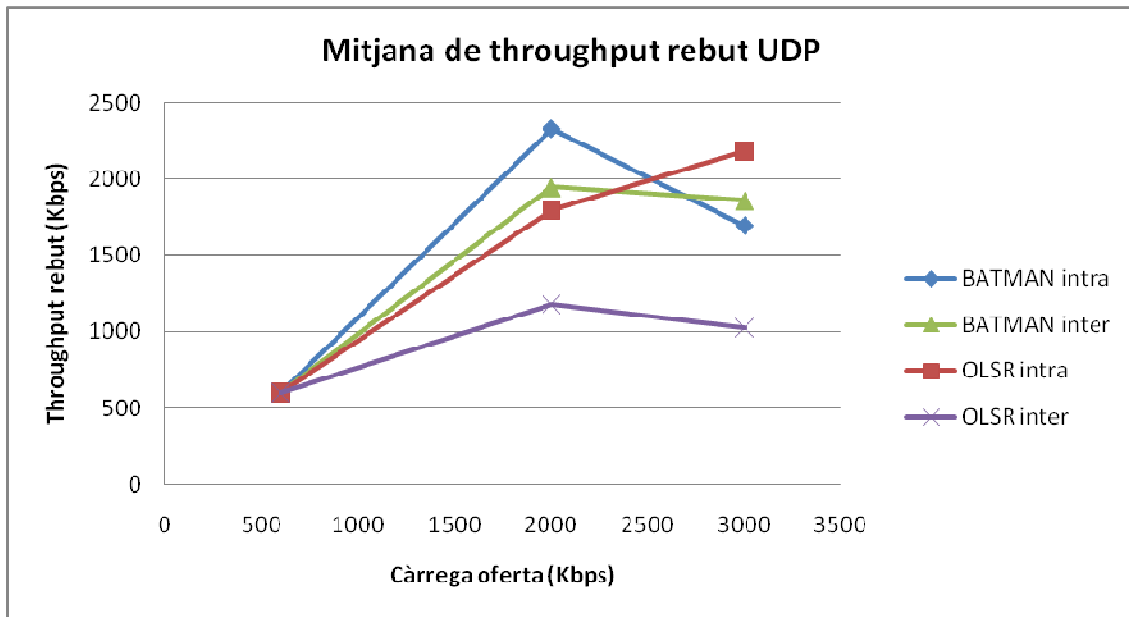


Fig 2.17. Mitjana de throughput rebut en UDP

Igual que en el cas anterior, els pitjors resultats de jitter els té OLSR en transmissió interna, seguit de BATMAN en transmissió entre grups, en canvi OLSR en transmissió entre grups i BATMAN en interna obtenen uns bons resultats.

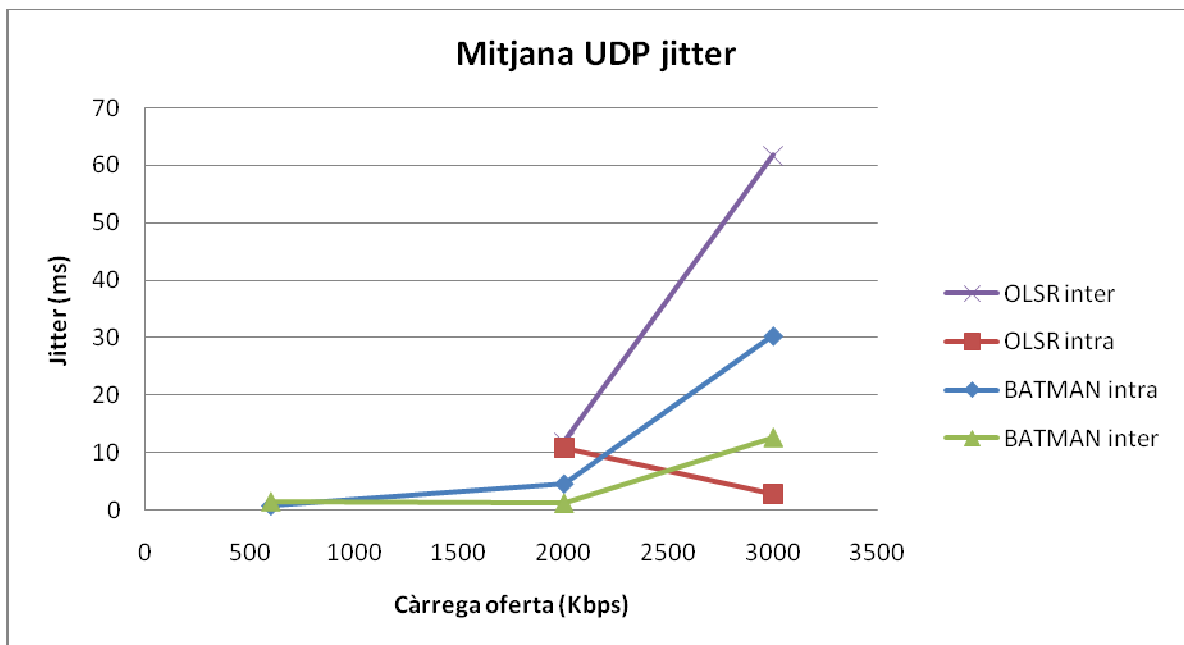


Fig 2.18. Mitjana UDP jitter

Pel que fa als paquets TCP tots dos protocols tenen un comportament similar respecte al throughput rebut, tot i que OLSR es comporta lleugerament millor

que BATMAN. En el cas de transmissió entre grups no s'han produït pèrdues de paquets per a cap dels dos protocols i el nombre de retransmissions també ha estat bastant similar, tot i que OLSR n'ha necessitat més que BATMAN. Pel que fa a les transmissions internes el throughput rebut també és bastant similar entre els dos protocols, però en aquest cas és BATMAN qui té un comportament lleugerament superior a OLSR. En aquest cas sí que hi ha hagut recepció de paquets fora d'ordre, en OLSR tres vegades més que en BATMAN, i el nombre de retransmissions no varia massa entre els dos protocols.

Taula 2.4. Paquets TCP

	TCP:			
	Intra		Inter	
	BATMAN	OLSR	BATMAN	OLSR
Throughput rebut (Kbps)	1444	1724	1414	1372
Paquets rebuts fora d'ordre	0	0	324,50	967,39
Nº de retransmissions	87	124	842,50	318,50

2.8. Sumari

Una vegada obtingudes i analitzades les dades de les proves s'ha pogut comprovar si les hipòtesis inicials eren certes.

En el cas de l'escenari estàtic es compleix el fet de que com més nodes hi ha a la xarxa pitjor rendiment s'obté, fins arribar a un màxim de cinc, a partir d'aquí els valors obtinguts eren massa baixos, per tant no es pot obtenir una comunicació amb més de quatre salts.

En multicamí s'ha pogut observar que s'obtenen millors resultats que en l'escenari anterior tenint el mateix nombre de salts, per tant el fet de tenir més d'una ruta per encaminar els paquets des d'un mateix origen fins a un mateix destí sí que provoca una millora en l'eficiència de la comunicació.

En el cas de mobilitat no s'han complert les hipòtesis inicials, ja que el rendiment de la xarxa era molt similar al cas ideal. Per tant podem suposar que els paquets que s'envien pel canvi de topologia no afecten de manera negativa en la transmissió.

Per últim, en les proves de treball col·laboratiu s'han obtingut uns resultats no esperats, el que produïa més pèrdues era la qualitat de les targes dels diferents ordinadors, el fet de treballar amb màquines de models diferents va mostrar que les que tenien les targes de menor qualitat produïen una baixada en l'eficiència de la xarxa.

En totes les proves també s'ha notat un altre efecte que influeix en els resultats, que és la presència de la gent en els escenaris, les mateixes proves fetes en

moments diferents donaven resultats inferiors si hi havia molta gent a prop de les màquines respecte a si no n'hi havia.

CONCLUSIONS

Per dur a terme aquest projecte s'ha partit d'una sèrie de conceptes bàsics, com ara saber què són les xarxes mesh, les xarxes ad-hoc i les MANETs, a partir d'aquests conceptes s'ha profunditzat en el tema de MANETs i s'han estudiat els algorismes d'encaminament en aquests tipus de xarxes. Concretament hi ha tres tipus d'algorismes, que són els protocols d'encaminament, els procediments per a càlculs de rutes i les mètriques per al càlcul de l'enllaç. De protocols, com s'ha dit anteriorment n'hi ha de tres tipus: proactiu, reactiu i híbrid. De procediments per a càlculs de rutes en trobem dos, vector distància i estat de l'enllaç. Les mètriques per al càlcul de l'enllaç més comunes són per nombre de salts i per ETX.

A partir d'aquests conceptes es van triar dos protocols per ser estudiats i analitzar el seu comportament. Aquests protocols són BATMAN i OLSR.

Van ser triats aquests dos en concret per dos motius. Per una banda es volien comparar dos protocols que fessin servir algorismes d'encaminament totalment diferents. BATMAN fa servir protocol proactiu i encaminament per vector distància, al contrari que OLSR, que fa servir enrutament proactiu i encaminament per estat de l'enllaç, d'aquesta manera s'ha pogut estudiar si influeix de manera notable la diferència entre els algorismes d'encaminament a l'hora d'establir la comunicació i per a la transmissió.

L'altre motiu pel que es van triar aquests dos protocols va ser que OLSR és un protocol estandaritzat, el que significa que es garanteix la interoperativitat entre les màquines. Mentre que BATMAN es va crear com a millora als problemes que tenia OLSR. D'aquesta manera s'ha pogut comprovar si realment BATMAN oferia millors resultats que OLSR.

Després de tenir clars els conceptes necessaris per al treball i escollir els dos protocols que serien estudiats, es va procedir a la realització de les proves. Aquestes es van dur a terme en un escenari real, provant els protocols en diferents circumstàncies, primer es van implementar en un escenari estàtic, després es va analitzar en multicamí, tot seguit es va provar amb mobilitat i per últim es van fer proves de treball en grup. A partir d'aquí es van obtenir una sèrie de resultats.

Una vegada analitzats els resultats de les proves s'ha arribat a una sèrie de conclusions.

En general els dos protocols es comporten de manera molt similar i obtenen resultats prou acceptables fins a 3 salts, a partir de 4 salts les transmissions comencen a baixar el seu rendiment.

El comportament dels protocols varia en funció de les targetes dels ordinadors. En les proves de grup s'ha pogut observar que s'obtenen pitjors resultats quan la transferència es fa entre ordinadors HP NX6110 i quan aquests són els que

transmeten la informació a ordinadors HP NX6310, la qual cosa es pot atribuir a la qualitat de les targetes dels HP NX6110 que és inferior a la dels HP NX6310. I això s'ha de tenir en compte a l'hora de formar els grups de treball.

Durant la realització de les proves, es va veure que “la gent afecta a la qualitat de l'enllaç”, ja que realitzant les mateixes proves, el mateix dia s'obtenien resultats molt més baixos si hi havia molta gent al voltant dels ordinadors, en canvi si a l'escenari no hi havia gent el comportament millorava notablement.

A partir d'aquest comportament es pot concloure que les MANETs són útils per a transmissions amb pocs salts, 4 com a molt, i per a treball cooperatiu, ja que amb aquestes dues característiques en els escenaris el rendiment dels protocols és prou bo.

BIBLIOGRAFIA

Llibres:

[1] Rob Flickenger, *Redes inalámbricas en los países en desarrollo: Una guía práctica para planificar y construir infraestructuras de telecomunicaciones de bajo costo*, Limehouse Book Sprint Team, 2006

Articles:

[2] *Routing Metrics and Protocols for Wireless Mesh Networks* [en línia]. Brasil: Universidade Federal do Rio de Janeiro, Departamento de Engenharia de Telecomunicações –TET Universidade Federal Fluminense, Universidade do Estado do Rio de Janeiro [Consulta: novembre de 2008]. Disponible a: <http://www.gta.ufrj.br/ftp/gta/TechReports/CPE08.pdf>

[3] *Studying Wireless Routing Link Metric Dynamics* [en línia]. Tittsburgh: School of ECE, Purdue University and Intel Research [Consulta: novembre de 2008]. Disponible a: <http://www.imconf.net/imc-2007/papers/imc191.pdf>

[4] Roberto Riggio, Daniele Miorandi, “JANUS: a framework for distributed management of wireless mesh networks”, CREATE-NET

[5] *A Multi-Commodity Flow Approach for Globally Aware routing in Multi-Hop Wireless Networks* [en línia]. New York: CS. Department, Binghamton University [Consulta: desembre de 2008]. Disponible a: <http://www.cs.binghamton.edu/~nael/research/papers/percom06.pdf>

[6] *Medium Access control and Routing Protocols for Wireless Mesh Networks* [en línia]. Illinois: Department of Computer Science, University of Illinois, Urbana, IL 61801 [Consulta: desembre de 2008]. Disponible a: http://lion.cs.uiuc.edu/~kjp/papers/07-MeshBookChapter_corrected.pdf

[7] *Comparison of Routing Metrics for Static Multi-Hop Wireless Networks* [en línia]. Microsoft Research [Consulta: desembre de 2008]. Disponible a: <http://research.microsoft.com/pubs/73158/metrics.pdf>

Projectes/treballs final de carrera:

[8] Alexandra Villagrassa Batalla, Esunly Medina Medina, “A real-world implementation and parametrization of mobile ad-hoc networks”, Universitat Politècnica de Catalunya.

[9] Alejandro Pascual Abascal, Miguel Ángel Pérez Ruiz, “Diseño e implantación de una red Wi-Fi mesh en un entorno campus”, Universidad de Deusto.

Altres:

[10] Open-mesh.net. [En línia] <https://www.open-mesh.net/batman>

[11] Ad-hoc wireless mesh routing daemon. [En línia] <http://www.olsr.org>

[12] Dr. Nasser Yazdani. *Special Topics on Wireless Ad-hoc Networks*. [En línia] University of Tehran [Consulta: novembre-desembre de 2008]. Disponible a: ece.ut.ac.ir/Classpages/F86/ECE637/FILES/lectures/Routing-lec10.ppt

[13] The care, feeding and use of Netperf. [En línia] <http://www.netperf.org/netperf/NetperfTraining.html>

[14] RFC Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N). Network Working Group. [En línia] <http://tools.ietf.org/id/draft-wunderlich-openmesh-manet-routing-00.txt>

[15] RFC Optimezed Link State Routing Protocol (OLSR). Network Working Group. [En línia] <http://www.ietf.org/rfc/rfc3626.txt>

ANNEXOS

ANNEX A. Configuració OLSRd

A.1. Instal·lació de l'OLSRd

En el sistema operatiu Windows per instal·lar l'olsrd només cal descarregar-se el fitxer executable de la web d'olsrd i seguir els passos.

En el sistema operatiu Linux cal descarregar el paquet "olsrd-0.5.6-r2.tar.gz" de la mateixa web que en el cas de Windows, i en consola s'han d'utilitzar les comandes següents:

```
tar xzvf olsrd-0.5.6-r2.tar.gz
cd olsrd-0.5.6-r2
make
make install
```

Una vegada instal·lat el daemon de l'olsr, s'ha de modificar el fitxer de configuració (que s'ha creat a la carpeta /etc/), de manera que s'ha de canviar la interfície per la que es vol connectar, afegir les diferents direccions IPs dels ordinadors que formen part de la xarxa i els plug-ins que es necessiten, en aquest cas s'ha afegit el plug-in dot_draw, que serveix per obtenir un dibuix de la topologia lògica de la xarxa.

En Linux un cop modificat el fitxer de configuració de l'olsr es pot executar el daemon fent servir la comanda "olsrd" en consola. En el cas de Windows només cal clicar sobre la icona de l'aplicació per accedir a la interfície gràfica. En tots dos casos apareixen una sèrie de camps:

Links: on apareix la direcció IP de l'ordinador en el que estem executant l'olsrd.

Two-hop neighbors: on s'observen els veïns que té la màquina a dos salts i el proper salt per arribar a aquests veïns

Topology: on es mostren totes les adreces IP que hi ha a la xarxa

A la figura següent podem observar una captura del terminal executant el programa en Linux, en aquest exemple només hi havia dues màquines connectades, una amb la IP 192.168.2.5 i l'altra amb la IP 192.168.2.7 (des d'on es va fer la captura).

```

File Edit View Terminal Tabs Help

Source IP addr  Dest IP addr      LQ      ETX
192.168.2.5     192.168.2.7     0.502/0.486  4.097
192.168.2.7     192.168.2.5     0.482/0.502  4.130
Processing TC from 192.168.2.7, seq 0x293d

--- 13:43:24.505439 ----- DIJKSTRA

--- 13:43:24.505458 ----- LINKS

IP address      hyst      LQ      ETX
192.168.2.7     0.000    0.502/0.486  4.097

--- 13:43:24.505496 ----- TWO-HOP NEIGHBORS

IP addr (2-hop) IP addr (1-hop) Total cost

--- 13:43:24.505510 ----- TOPOLOGY

Source IP addr  Dest IP addr      LQ      ETX
192.168.2.5     192.168.2.7     0.502/0.486  4.097
192.168.2.7     192.168.2.5     0.486/0.502  4.097

--- 13:43:24.755972 ----- LINKS

IP address      hyst      LQ      ETX
192.168.2.7     0.000    0.502/0.490  4.064

--- 13:43:24.756043 ----- TWO-HOP NEIGHBORS

IP addr (2-hop) IP addr (1-hop) Total cost

--- 13:43:24.756077 ----- TOPOLOGY

Source IP addr  Dest IP addr      LQ      ETX
192.168.2.5     192.168.2.7     0.502/0.486  4.097
192.168.2.7     192.168.2.5     0.486/0.502  4.097

```

Terminal de l'OLSRd

Si s'executa el programa en Windows les dades que es mostren són les mateixes que en Linux, la figura següent és un fragment del fitxer "log" que guarda tot l'historial de la connexió, en aquest cas hi ha tres màquines a la xarxa:

```

|-- 11:51:11.02375000 ----- TWO-HOP NEIGHBORS
IP addr (2-hop) IP addr (1-hop) TLQ
192.168.2.8      192.168.2.6      0.053
192.168.2.6      192.168.2.8      0.064

--- 11:51:11.37 ----- TOPOLOGY
Source IP addr  Dest IP addr    LQ      ILQ      ETX
192.168.2.3     192.168.2.6     0.200   0.957   5.23
192.168.2.3     192.168.2.8     0.458   0.498   4.38
192.168.2.6     192.168.2.3     0.200   0.957   5.23
192.168.2.6     192.168.2.8     0.294   0.949   3.58
192.168.2.8     192.168.2.3     0.451   0.498   4.45
192.168.2.8     192.168.2.6     0.945   0.298   3.55
Processing TC from 192.168.2.6, seq 0x0034
Processing TC from 192.168.2.8, seq 0x0038
Processing TC from 192.168.2.6, seq 0x0034
Processing TC from 192.168.2.8, seq 0x0038
Processing TC from 192.168.2.6, seq 0x0034

--- 11:51:22.59 ----- DIJKSTRA

--- 11:51:22.59 ----- LINKS
IP address      hyst    LQ      lost    total  NLQ      ETX
192.168.2.6     0.000  0.200  0       12     0.957   5.23
192.168.2.8     0.000  0.417  2       12     0.498   4.82

--- 11:51:22.02593000 ----- TWO-HOP NEIGHBORS
IP addr (2-hop) IP addr (1-hop) TLQ
192.168.2.8      192.168.2.6      0.053
192.168.2.6      192.168.2.8      0.054

```

Fitxer "log"

A.2. Plug-ins

Per poder fer servir els plug-ins, prèviament a afegir-los al fitxer de configuració de l'olsr, s'han d'instal·lar, s'ha d'accedir a la carpeta on es troben "/olsrd-0.5.6-r2/lib" i utilitzar la comanda "make" i "make install" per a cadascun dels plug-ins que interessin.

A.2.1 *Plug-in dot_draw*

En el cas del plug-in dot_draw també és necessari instal·lar els paquets graphviz i imagemagick.

Per comprovar si funciona correctament es pot fer un telnet al port 2004 amb l'olsrd en execució. En cas que funcioni apareixerà la següent resposta al telnet:

```
nuria@nuria-laptop:/etc/olsrd$ telnet 192.168.2.5 2004
Trying 192.168.2.5...
Connected to 192.168.2.5.
Escape character is '^]'.
digraph topology
{
"192.168.2.5" -> "192.168.2.7"[label="4.305", style=solid];
"192.168.2.5" -> "192.168.2.7"[label="4.305"];
"192.168.2.7" -> "192.168.2.5"[label="4.305"];
}

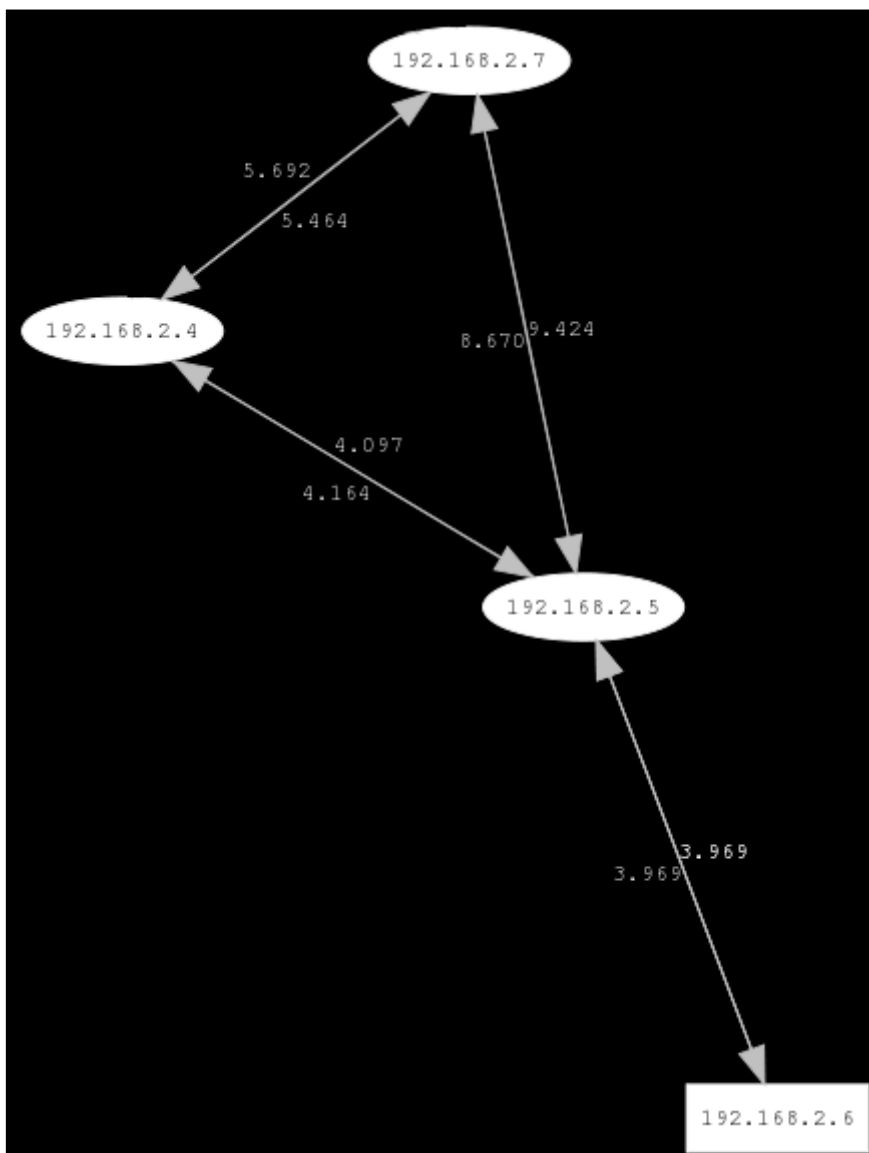
Connection closed by foreign host.
```

Resposta al telnet

En aquest cas hi havia dues màquines a la xarxa.

Per a executar el plug-in `dot_draw` s'ha d'accedir al directori `"olsrd-0.5.6-r2/lib/dot_draw/misc"` i fer servir la comanda `"./olsr-topology-view.pl"`.

La figura següent mostra un exemple, amb quatre màquines a la xarxa, del dibuix de la topologia fent servir aquest plug-in:



Dibuix de la topologia lògica de la xarxa

A.3. Fitxer de configuració (olsrd.conf)

El fitxer de configuració utilitzat en aquest cas ha estat el següent:

```
DebugLevel 2
IpVersion 4
FIBMetric "flat"
Hna4
{
}
AllowNoInt yes
```

```
TosValue      16

RtTable                256

IpcConnect
{
    MaxConnections      1
    Host                127.0.0.1
    Net                 192.168.1.0/24
}

UseHysteresis      no

Pollrate          0.05

NicChgsPollInt     2.50

TcRedundancy       2

MprCoverage        7

LinkQualityLevel   2

LinkQualityFishEye 1

LinkQualityWinSize 12

ClearScreen        no

LoadPlug-in "/home/olsrd-0.5.6-r2/lib/dot_draw/olsrd_dot_draw.so.0.3"
{
    PlParam "accept" "127.0.0.1"
    PlParam "port" "2004"
}

Interface "wlan0"{
    HelloInterval          2.00
    HelloValidityTime      20.00
    TcInterval             3.00
    TcValidityTime         375.00
    MidInterval            25.00
    MidValidityTime        375.00
    HnaInterval            25.00
    HnaValidityTime        375.00
    LinkQualityMult        192.168.2.1 0.5
    LinkQualityMult        192.168.2.4 0.5
    LinkQualityMult        192.168.2.5 0.5
    LinkQualityMult        192.168.2.6 0.5
    LinkQualityMult        192.168.2.7 0.5
    LinkQualityMult        default 0.8
}
```